902 56

รายงานวิจัยฉบับสมบูรณ์
ระบบแนะนำสถานที่ท่องเที่ยวอัตโนมัติด้วยพิกัดจีพีเอส
Automation Tourist Guide with GPS

สัญญา ผาสุข
ศุภชัย อรุณพันธ์
จินดา สามัคคี

ว 004
ส 213
2559

สาขาวิชาวิศวกรรมโทรคมนาคม สาขาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

ได้รับการสนับสนุนวิจัยจากมหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย
งบประมาณรายได้ ประจำปี พ.ศ.2559

คำนำ

ปัจจุบันความก้าวหน้าเทคโนโลยีด้านภูมิศาสตร์ได้ก้าวหน้าไปอย่างมาก สามารถรู้พิกัดตำแหน่งบนพื้นที่ผิวของโลกได้ทุกๆตำแหน่ง โดยการระบุเป็นตำแหน่งละติจูด ลองติจูด เปรียบเสมือนการระบุตำแหน่ง x,y บนกระดาษกราฟ ทำให้ง่ายต่อการระบุตำแหน่งและกำหนดจุดพื้นที่หรือบริเวณบนผิวโลกด้วยเครื่องบอกพิกัดจีพีเอส ซึ่งปัจจุบันได้นำมาใช้กันอย่างแพร่หลายในงานด้านต่างๆ ซึ่งในงานวิจัยนี้ได้ประยุกต์กับการนำมาใช้สำหรับบรรยายข้อมูลของสถานที่ท่องเที่ยวต่างๆในรูปแบบเสียง ซึ่งเก็บไว้เป็นไฟล์ MP3 ในหน่วยความจำ SD การ์ด โดยระบบถูกควบคุมด้วยไมโครคอนโทรลเลอร์ ทำหน้าที่ตรวจสอบพิกัด และควบคุมการเล่นไฟล์เสียง

ผู้เขียนหวังว่ารายงานฉบับนี้จะมีประโยชน์ต่อผู้สนใจ สามารถนำไปต่อยอดทั้งในงานวิจัยด้านอื่นๆ หรือสามารถนำไปพัฒนาต่อยอดเป็นชิ้นงานนวัตกรรม เพื่อประโยชน์แก่สังคมต่อไป และต้องขอขอบคุณคณะวิศวกรรมศาสตร์ มทร. ศรีวิชัย สำหรับการให้ทุนสนับสนุนการวิจัยในครั้งนี้

สัญญา ผาสุข

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

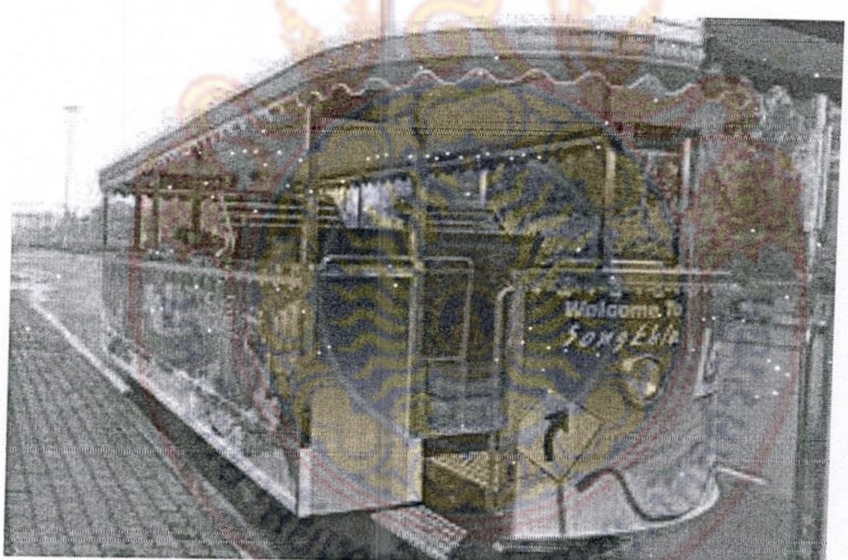
สารบัญ

บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของการวิจัย	3
1.2 วัตถุประสงค์	3
1.3 ขอบเขตงานวิจัย	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎี	4
2.1 การแบ่งเส้นละติจูดและลองติจูด	7
2.2 การระบุจีพีเอส	6
2.3 ระบบจีพีเอส	8
บทที่ 3 การออกแบบระบบควบคุม	16
3.1 การรับข้อมูลพิกัดจีพีเอส	16
3.2 การควบคุมเครื่องเล่น MP3	20
3.3 การแสดงผล	22
3.4 การป้อนพิกัดสถานที่ท่องเที่ยว	23
3.5 โปรแกรมป้อนพิกัดสถานที่ท่องเที่ยว	26
3.6 การทำงานของโปรแกรม	27
ทที่ 4 ผลการทดลอง	73
4.1 การทดสอบพิกัดจีพีเอส	74
4.2 การทดสอบเล่นไฟล์เสียง	75
4.3 การทดสอบบริเวณขอบเขตของพิกัดสถานทีท่องเที่ยว	76
4.4 การทดลองกับสถานที่ท่องเที่ยว	78
บทที่ 5 สรุปผลการทดลอง	84
ภาคผนวก	85

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของการวิจัย

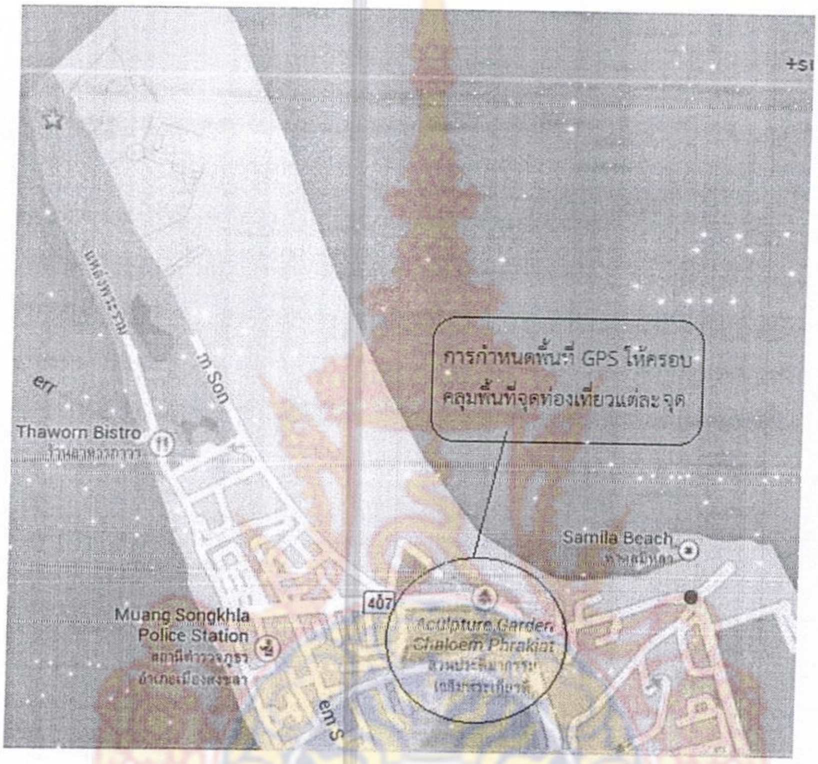
ประเทศไทยมีสถานที่ท่องเที่ยวเป็นจำนวนมาก เป็นที่ดึงดูดนักท่องเที่ยวทั้งในประเทศและต่างประเทศให้เดินทางไปท่องเที่ยวในสถานที่ต่างๆ ทำให้เศรษฐกิจมีความรุ่งเรือง สามารถดึงเม็ดเงินเข้าในประเทศได้เป็นจำนวนมาก แต่ประเทศไทยยังขาดจำนวนมัคคุเทศ หรือที่เรียกกันติดปากกว่าไกด์ (Guide) ซึ่งจะทำหน้าที่แนะนำ บรรยายให้ความรู้ ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวในแต่ละแหล่ง ทำให้นักท่องเที่ยวขาดข้อมูล ความรู้ในการไปเที่ยวตามสถานที่ต่างๆ บางพื้นที่บนรถนำเที่ยวอาจจะใช้การบรรยายแหล่งท่องเที่ยวจากรูปแบบของมัลติมิเดียร์จากเครื่องเล่น เช่นจุดท่องเที่ยวในสวนสัตว์ หรือสวนสนุกต่างๆ ซึ่งต้องอาศัยผู้ควบคุมเครื่องเล่นคอยกดปุ่ม ซึ่งไม่เป็นไปโดยอัตโนมัติ ดังนั้นหากมีการพัฒนารูปแบบของเครื่องบรรยายสถานที่ท่องเที่ยวให้เป็นรูปแบบอัตโนมัติเมื่อเดินทางมาถึงสถานที่ต่างๆ จะทำให้เกิดความสะดวกต่อการท่องเที่ยวมากยิ่งขึ้น



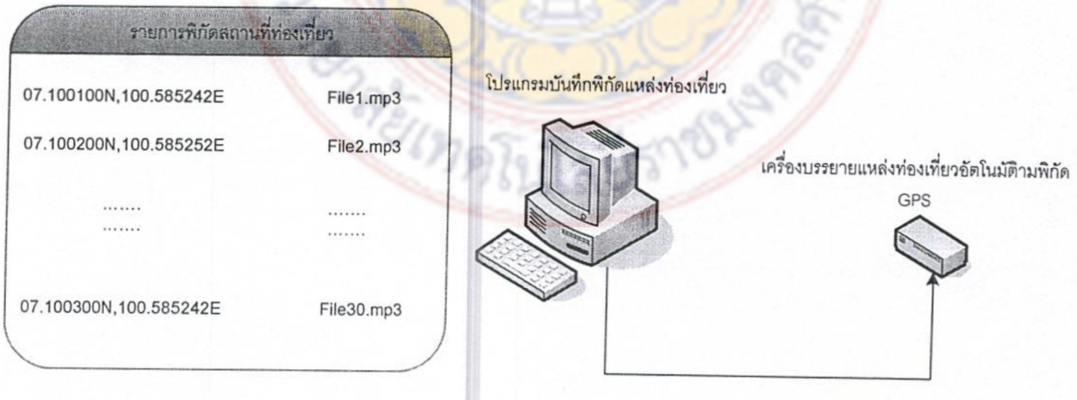
รูปที่ 1.1 รถนำเที่ยวตามจุดท่องเที่ยวต่างๆ

การบรรยายให้ข้อมูลและความรู้เกี่ยวกับแหล่งท่องเที่ยวจะบรรยายโดยอัตโนมัติเมื่อรถนำเที่ยวได้เดินทางมาถึงพื้นที่ของสถานที่ท่องเที่ยว จะใช้พิกัดของจีพีเอส ซึ่งเป็นตำแหน่งลองติจูด และละติจูด ที่กำหนดให้ครอบคลุมพื้นที่ของแหล่งท่องเที่ยวพื้นที่นั้นๆ โดยที่สามารถเข้าถึงบริเวณจุดท่องเที่ยวทางทิศใดๆ ก็ได้ เพียงแต่เมื่อรถนำเที่ยวมาถึงพื้นที่ดังกล่าว ระบบก็จะเริ่มการบรรยายอย่าง

อัตโนมัติ โดยเสียงบรรยายจะถูกเก็บไว้ในหน่วยความจำประเภท SD Card ซึ่งจะบึกทึกเก็บพิกัดตำแหน่งละติจูด ลองติจูด และเสียงบรรยายของแหล่งท่องเที่ยวแต่ละจุดด้วยการสร้างโปรแกรมป้อนข้อมูลด้วยคอมพิวเตอร์ ทำหน้าที่เก็บเสียงบรรยายและจุดพิกัดลองติจูดและละติจูดแหน่งใหม่ๆ แล้วจึงถ่ายโอนไปยังหน่วยความจำ SD การ์ดดังลักษณะในรูปที่ 3 เพื่อนำไปใช้กับเครื่องบรรยายสถานที่ท่องเที่ยวที่ได้สร้างขึ้น



รูปที่ 1.2 การกำหนดตำแหน่ง GPS ให้ครอบคลุมพื้นที่บริเวณจุดท่องเที่ยว



รูปที่ 1.3 การถ่ายโอนพิกัดและเสียงบรรยายแหล่งท่องเที่ยวไปยังเครื่องบรรยาย

1.2 วัตถุประสงค์

1.2.1 ศึกษาการควบคุมการเล่นไฟล์เสียง MP3 อัตโนมัติ ตามพิกัดของจีพีเอส

1.2.2 ออกแบบระบบการบรรยายให้ความรู้ ข้อมูลสถานที่ท่องเที่ยวโดยอัตโนมัติ

1.3 ขอบเขตงานวิจัย

1.3.1 สามารถบรรยายข้อมูลแหล่งท่องเที่ยว ด้วยเสียง MP3 อย่างอัตโนมัติ ตามพิกัดจีพีเอส

1.3.2 สามารถเก็บข้อมูลของสถานที่ท่องเที่ยวได้ไม่น้อยกว่า 20 จุด

1.3.3 การกำหนดตำแหน่งของสถานที่ท่องเที่ยวด้วยระบบจีพีเอส (GPS)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้เครื่องต้นแบบสำหรับการบรรยายแหล่งท่องเที่ยวอัตโนมัติตามพิกัดของ GPS

1.4.2 ลดปัญหาการขาดแคลนไกด์บรรยายแหล่งท่องเที่ยว

1.4.3 การเผยแพร่ในการประชุมวิชาการทั้งใน และต่างประเทศ

บทที่ 2
ทฤษฎี

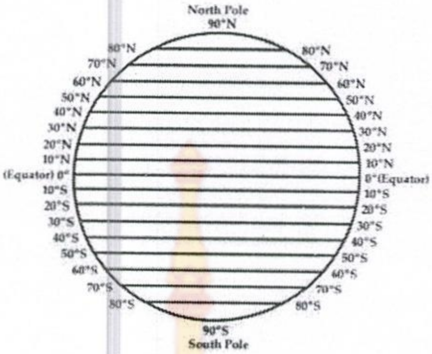
โลก (Earth) มีลักษณะเป็นรูปวงรี (Oblate Ellipsoid) คือมีลักษณะป่องตรงกลาง ขั้วเหนือ-ใต้ แบนเล็กน้อย แต่พื้นผิวโลกที่แท้จริงมีลักษณะขรุขระ สูง ต่ำ ไม่ราบเรียบสม่ำเสมอ พื้นผิวโลกจะมีพื้นที่ประมาณ 509,450,000 ตารางกิโลเมตร ความยาวรอบโลกประมาณ 40,000 กิโลเมตร มีเส้นผ่านศูนย์กลางที่ศูนย์สูตร ยาว 12,757 กิโลเมตร มีเส้นผ่านศูนย์กลางจากขั้วโลกเหนือถึงขั้วโลกใต้ 12,714 กิโลเมตร จะเห็นว่าระยะทางระหว่างแนวนอน (เส้นศูนย์สูตร) ยาวกว่าแนวตั้ง (ขั้วโลกเหนือ-ใต้) จากลักษณะดังกล่าวนี้ ทำให้ไม่สามารถใช้รูปทรงเรขาคณิตอย่างง่ายแสดงขนาดรูปร่างของโลกได้อย่างถูกต้อง ดังนั้นเพื่อความสะดวกต่อการพิจารณารูปทรงของโลกและในกิจการของแผนที่ จึงมีการใช้รูปทรงของโลก 3 แบบ คือ ทรงกลม (Spheroid), ทรงรี (Ellipsoid) และยีออยต์(Geoid) ดังแสดงในรูปที่ 2.1

รูปที่ 2.1 รูปทรงของโลกแต่ละแบบ

2.1 การแบ่งเส้นละติจูดและลองติจูด

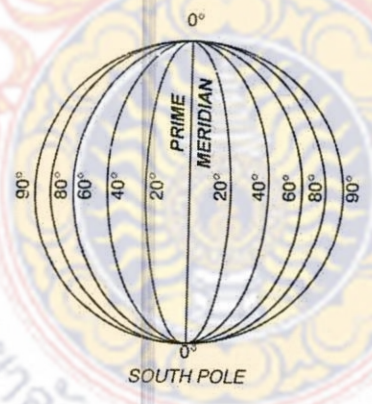
การหาพิกัดตำแหน่งบนพื้นโลกด้วยเครื่องค้นหาตำแหน่งจีพีเอสนิยม ใช้การระบุด้วยค่าละติจูดและค่าลองติจูดซึ่งเปรียบเหมือนการหาค่าพิกัด x,y ในกราฟ โดยการแบ่งเส้นของค่าทั้งสองเป็นดังนี้

2.1.1 เส้นละติจูด (Latitude) หรือเรียกว่าเส้นรุ้ง ลากไปทางแนวเส้นศูนย์สูตร (Equator) ซึ่งเป็นเส้นสมมติแบ่งกึ่งกลางโลกตามขวาง แบ่งไปทางเหนือและใต้อย่างละ 90 เส้น รวมทั้งหมด 180 เส้น โดยลำดับของเส้น 0 จะอยู่ที่เส้นศูนย์สูตร ซึ่งเป็นจุดกึ่งกลาง และลำดับถัดไป จะเรียงลำดับไปทางทิศเหนือ และทางทิศใต้ ของเส้นศูนย์สูตร ดังนั้นหากมีการระบุตำแหน่งของเส้นละติจูด จะต้องมีการกำกับทิศเหนือ หรือทิศใต้ของเส้น 0 ร่วมกับลำดับเส้นละติจูดด้วยเสมอ เช่น ละติจูดที่ 7.000N หมายถึงลำดับเส้นละติจูดที่ 7 องศาเหนือ เป็นต้น



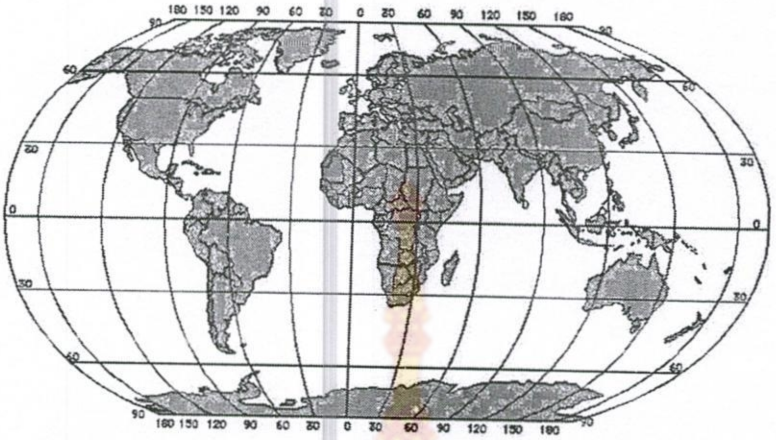
รูปที่ 2.2 การแบ่งเส้นละติจูด

2.1.2 เส้นลองติจูด (Longtitude) หรือเส้นแวง เป็นเส้นที่ลากจากขั้วโลก มาตามพื้นผิวพิภพไปยังขั้วโลกใต้ โดยเส้น 0 ทางสากลถือว่าเป็นเส้นสมมติศูนย์ จะลากจากขั้วโลกเหนือ มายังขั้วโลกใต้ ผ่านเมืองกรีนิช (Greenwich) ประเทศอังกฤษ และมีการแบ่งเส้นๆละเท่าๆกันมาทางทิศตะวันออกจำนวน 180 เส้นและแบ่งไปทางทิศตะวันตกของเส้น 0 อีก 180 เส้น รวมทั้งหมด 360 เส้น ลักษณะดังรูปที่ 2.3 การระบุตำแหน่งเส้นลองติจูดใดๆ จะต้องระบุทิศร่วมด้วย เช่นเส้นลองติจูดที่ 100E หมายถึงเส้นลองติจูดที่ 100 องศาตะวันออก เพราะหากไม่มีการระบุทิศ จะทำให้ไม่แน่ชัดว่าเป็นลำดับเส้นที่ 100 ในทิศตะวันออก หรือ ทิศตะวันตกของเส้น 0



รูปที่ 2.3 การแบ่งเส้นลองติจูด

ดังนั้นการระบุตำแหน่งบนพื้นโลก สามารถใช้จุดตัดของเส้นละติจูด และเส้นลองติจูด เพื่อบอกตำแหน่ง ณ ตำแหน่งใดๆบนโลก คล้ายๆ กับการระบุจุด X,Y บนกราฟ นั่นเอง



รูปที่ 2.4 การตัดกันของเส้นละติจูดและเส้นลองติจูด

การระบุตำแหน่ง จะต้องใช้ทั้งค่าละติจูดและลองติจูด เหมือนการกำหนดจุด x,y บนกราฟ โดยกำหนดค่าละติจูดแทนตำแหน่งในแนวนอน และกำหนดค่าลองติจูดแทนตำแหน่งในแนวตั้ง เช่น พิกัดตำแหน่งของประติมากรรมพญานาคพ่นน้ำของเมืองสงขลาตั้งอยู่ที่พิกัด 7.227223N,100.576994E ซึ่งหมายความว่าตำแหน่งดังกล่าวอยู่ที่พิกัดละติจูด 7.227223 องศาเหนือ และ ลองติจูดที่ 100.576991 องศาตะวันออก โดยเป็นการระบุในรูปแบบทศนิยม

รูปที่ 2.5 ประติมากรรมพญานาคพ่นน้ำของจังหวัดสงขลา

รูปที่ 2.6 ตำแหน่งพิกัดละติจูด ลองติจูด ของประติมากรรมพญานาคพ่นน้ำ จังหวัดสงขลา

2.2 การระบุพิกัดจีพีเอส

การแสดงพิกัดบนเครื่อง GPS ที่ใช้ในประเทศไทยส่วนใหญ่จะใช้ 2 ระบบ คือ พิกัดภูมิศาสตร์ และพิกัด UTM (Universal Transverse Mercator) การอ่านในระบบพิกัด UTM จะง่ายกว่าเพราะอ่านตัวเลขตามค่า East (X) และค่า North (Y) คล้ายกับการอ่านพิกัด X,Y บนกราฟ แต่การอ่านค่าระบบภูมิศาสตร์นั้นยุ่งยากกว่า เนื่องจากมีการระบุเป็น องศาลิปดา ฟิลิปดา (DMS : Degree Minute Second) ดังนั้นเมื่อต้องการใช้งานรูปแบบใดแบบหนึ่ง จึงต้องมีการแปลงค่าหน่วย DMS เป็น DD หรือ DD เป็น DMS

1) ค่าพิกัดภูมิศาสตร์ DMS (Degree Minute Second) ซึ่งเป็นการบอกหน่วยคล้ายกับหน่วยเวลา ชั่วโมง นาที วินาที โดยค่าหน่วยพิกัดภูมิศาสตร์ดังตารางที่ 1

ตารางที่ 1 หน่วยวัดพิกัดภูมิศาสตร์

1 องศา (Degree)	เท่ากับ 60 ลิปดา
1 ลิปดา (Minute)	เท่ากับ 60 ฟิลิปดา
1 ฟิลิปดา (Second)	มีค่าระยะทาง 30.48 ม หรือ 100 ฟุต

ตัวอย่างเช่น ประติมากรรมพญานาคพ่นน้ำ ของจังหวัดสงขลา ตั้งอยู่ ณ ตำแหน่งพิกัดภูมิศาสตร์ ละติจูด 7 องศา 13 ลิปดา 38 ฟิลิปดา เหนือ (7° 13’ 38”N) , ลองติจูด 100 องศา 34 ลิปดา 37 ฟิลิปดา ตะวันออก (100° 34’ 37”E)

2) พิกัดแบบ UTM จะบอกค่าในลักษณะแบบ DD (Decimal Degree) เป็นการบอกพิกัดแบบค่าตัวเลขทศนิยมฐานสิบ ซึ่งมีลักษณะคล้ายๆกับการบอกพิกัด x,y บนกราฟ เช่น ประติมากรรมพญานาคพ่นน้ำ ของจังหวัดสงขลา ตั้งอยู่ ณ ตำแหน่ง ละติจูด ลองติจูดที่ 7.227223N,100.576994E เป็นต้น

การบอกพิกัดตำแหน่งแต่ละแบบสามารถแปลงค่าซึ่งกันและกันได้ เพื่อความสะดวกและความคุ้นเคยของผู้ใช้งาน โดยการแปลงค่าจากรูปแบบ DMS เป็นรูปแบบ DD สามารถใช้สมการดังนี้

ละติจูด,ลองติจูด = องศา + (ลิปดา x 60 + ฟิลิปดา) / 3600

จากตัวอย่างตำแหน่งประติมากรรมพญานาคพ่นน้ำอยู่ที่พิกัด 7° 13’ 38”N, 100° 34’ 37”E สามารถแปลงเป็นพิกัดละติจูด ลองติจูดในรูปแบบ DD ได้เป็น

ละติจูด = 7 + (13 x 60 +38) / 3600
= 7.227222 N
ลองติจูด = 100 + (34 x 60 +37) / 3600
= 100.576944 E

การแปลงพิกัดจากรูปแบบ DD เป็น DMS เพื่อหาค่าองศา ลิปดา และ ฟิลิปดา เช่นที่พิกัดละติจูด ลองติจูด 7.227223N, 100.576994E มีวิธีการแปลงเป็นรูปแบบ DMS ดังนี้

ละติจูด
องศา = 7
ลิปดา = 0.227223 x 60 = 13.63338 เอาเฉพาะจำนวนเต็ม คือ 13
ฟิลิปดา = 0.63338 x 60 = 38.0028 เอาเฉพาะจำนวนเต็ม คือ 38

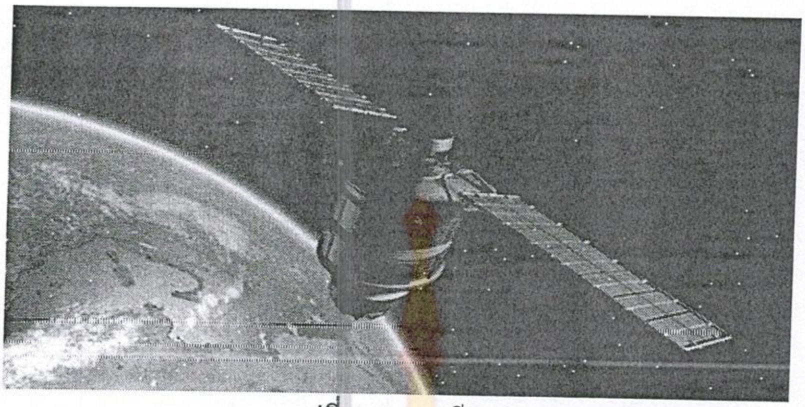
นั่นคือ ละติจูดที่ 7 องศา 13 ลิปดา 38 ฟิลิปดาเหนือ

ลองติจูด
องศา = 100
ลิปดา = 0.576994 x 60 = 34.61964 เอาเฉพาะจำนวนเต็ม คือ 34
ฟิลิปดา = 0.61964 x 60 = 37.1784 เอาเฉพาะจำนวนเต็ม คือ 37

นั่นคือ ลองติจูดที่ 100 องศา 34 ลิปดา 37 ฟิลิปดาตะวันออก

2.3 ระบบจีพีเอส

GPS (ย่อมาจาก Global Positioning System) คือ ระบบกำหนดตำแหน่งบนพื้นโลกผ่านดาวเทียม โดยพิกัดบนพื้นโลกจะอ้างอิงจากตำแหน่งของดาวเทียมแต่ละดวงที่โคจรเหนือพื้นโลก มาคำนวณหาพิกัดละติจูดและลองติจูดด้วยเครื่องรับสัญญาณ GPS นอกจากนี้ยังสามารถคำนวณหาระดับความสูง และค่าความเร็วของการเคลื่อนที่ได้อีกด้วย



รูปที่ 2.7 ดาวเทียม

ดาวเทียมของระบบ GPS มีจำนวน 24 ดวง แบ่งเป็น 6 วงโคจร วงโคจรละ 4 ดวง ซึ่งครอบคลุมพื้นผิวรอบโลกของแต่ละวงโคจร โดยวงโคจรจะเอียงทำมุมเอียง 55 องศากับเส้นศูนย์สูตร (Equator) ในลักษณะสานกันคล้ายลูกตะกร้อ มีรัศมีวงโคจรจากพื้นโลก 20,162.81 กม.หรือ 12,600 ไมล์ ดาวเทียมแต่ละดวงใช้เวลาการโคจรรอบโลกเพียง 12 ชั่วโมง ซึ่งเร็วกว่าการหมุนของโลก แต่หากดาวเทียมดวงไหนโคจรพ้นขอบท้องฟ้าไป จะมีดาวเทียมดวงถัดไปขึ้นมาทำหน้าที่สอดรับแทนที่ ทำให้ทุกๆตำแหน่งบนโลกจะมีดาวเทียมของแต่ละวงคอยส่งสัญญาณเพื่ออ้างอิงพิกัดแก่เครื่องรับสัญญาณ GPS บนท้องฟ้าตลอดเวลาอย่างน้อย 4 ดวง จึงทำให้สามารถบอกตำแหน่งได้ทุกแห่งบนโลกตลอด 24 ชั่วโมง ดังลักษณะในรูปที่ 2.8

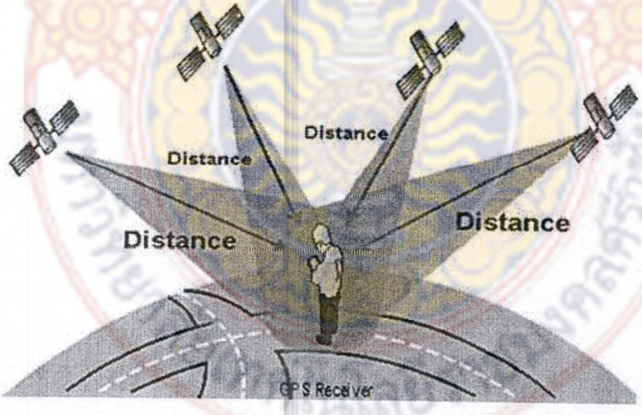
รูปที่ 2.8 ลักษณะการโคจรของดาวเทียม

ดาวเทียม GPS ทุกดวงมีรหัสประจำตัว ทำให้เครื่องรับสัญญาณรู้ได้ว่ารับสัญญาณนั้นๆ มาจากดาวเทียมดวงใด ซึ่งดาวเทียม GPS ที่กระโทรงกลาโหมอเมริกัน DoD (Department of Defense) ปล่อยขึ้นไปนั้นมีทั้งหมด 24 ดวงเพื่อใช้งานอยู่ในปัจจุบัน และในจำนวนนี้จะเป็นดาวเทียมเพื่อใช้สำรอง 3 ดวง แต่ละดวงมีอายุการใช้งาน 10 ปี โดยดวงแรกถูกปล่อยขึ้นไปตั้งแต่ปี 1978 และมีการปล่อยขึ้นเพื่อชดเชยดวงที่หมดอายุการใช้งาน ดาวเทียม GPS ไม่อยู่กับที่ แต่ละดวงจะโคจร

รอบโลกทุกๆ 12 ชั่วโมง ซึ่งมีวงโคจร 6 วงๆละ 4 ดวง รวม 24 ดวงครอบคลุมกระจายอยู่ทั่วโลก ทำให้ไม่ว่าจะเป็นเวลาใด ณ ตำแหน่งใดบนพื้นโลก จะมีดาวเทียมอยู่เหนือเส้นขอบฟ้าอย่างน้อย 4 ดวงคอยส่งสัญญาณให้แก่เครื่องรับ GPS โดยเราสามารถอาศัยตำแหน่งของดาวเทียมในอวกาศเป็นจุดอ้างอิง แล้ววัดระยะทางจากดาวเทียมทั้ง 4 ดวง และใช้หลักการทางเรขาคณิตในการคำนวณหากตำแหน่งบนพื้นโลกดังลักษณะในรูปที่ 2.9 เครื่องรับ GPS จะสามารถบอกตำแหน่งได้นั้น จะต้องได้รับสัญญาณจากดาวเทียมอย่างน้อย 3 ดวง และหากรับสัญญาณดาวเทียม 4 ดวงจะสามารถบอกระดับความสูงได้ และได้ข้อมูลในสี่มิติคือ X,Y,Z,T ซึ่งทำให้มีความแม่นยำมาก ดังนั้นยิ่งระบบจีพีเอสรับสัญญาณดาวเทียมมากเท่าใดความแม่นยำในการระบุตำแหน่งก็จะแม่นยำตามไปด้วย นอกจากนี้ตำแหน่งของดาวเทียมที่เครื่องรับ GPS รับสัญญาณได้ก็จะมีความสำคัญต่อความแม่นยำมากขึ้น หากรับสัญญาณได้จากดาวเทียมที่อยู่ในตำแหน่งใขว้กันมากๆ ไม่กระจุกตัว โดยการวัดระยะทางระหว่างเครื่องรับ GPS กับดาวเทียมนั้นจะใช้วิธีการวัดระยะเวลาของคลื่นวิทยุ ที่เดินทางจากดาวเทียมมายังเครื่องรับ โดยคลื่นวิทยุมีความเร็วเท่ากับความเร็วของแสง คือ 300,000 กิโลเมตรต่อวินาที (km/s) ซึ่งสามารถคำนวณหาระยะทางได้ดังสมการ

$$\text{ระยะทาง} = \text{ความเร็วคลื่นวิทยุ} * \text{เวลาที่ใช้เดินทาง}$$

การวัดระยะเวลาที่คลื่นวิทยุ จะต้องใช้นาฬิกาที่แม่นยำมากถ้าหมายเลขดาวเทียม (PRN CODE) จากดาวเทียมมีข้อมูลเวลาที่คลื่นเริ่มออกเดินทาง ซึ่งเครื่องรับ GPS จะต้องมีการ Synchronize และจะต้องใช้ Atomic Clock ในการวัดเวลา ซึ่งทำให้มีความแม่นยำถึง 10 นาโนวินาที โดยคลื่นสัญญาณดาวเทียม GPS จะใช้ความถี่ 2 ความถี่คือ L1 Code ซึ่งใช้ความถี่ 1575.42 MHz และ L2 Code ซึ่งใช้ความถี่ 1227.6 MHz



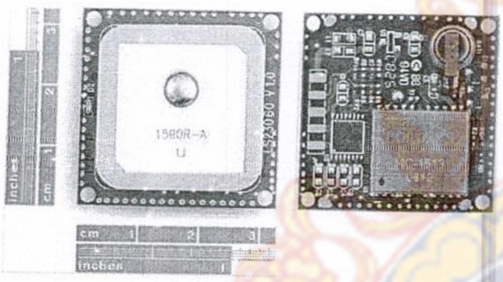
รูปที่ 2.9 จำนวนดาวเทียมของการคำนวณหาพิกัดตำแหน่งจีพีเอส

2.3.1 เครื่องรับสัญญาณจีพีเอส

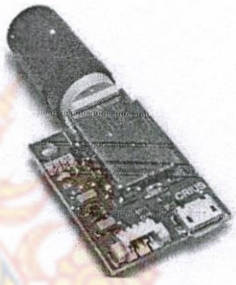
เครื่องรับสัญญาณ GPS มีหลายรูปแบบ ทั้งการบอกพิกัดตำแหน่งบนพื้นโลก บางรุ่นยังสามารถค้นหาเส้นทาง นำทางได้อีกด้วย แต่หากต้องการใช้สำหรับการนำข้อมูลจาก GPS มาพัฒนาต่อยอดจะต้องใช้ GPS ที่มีลักษณะของโมดูลจีพีเอส ซึ่งมีลักษณะส่วนประกอบดังนี้

1) Accuracy หมายถึง ค่าความถูกต้อง หรือความแม่นยำในการคำนวณระบุตำแหน่งบนพื้นโลกของเครื่องรับ สำหรับเครื่องรับที่ราคาไม่แพงมาก โดยทั่วไปแล้วจะมีค่าความถูกต้องอยู่ที่ประมาณ +/- 3 ถึง 5 เมตร แต่ถ้าหากต้องการความถูกต้องสูง ราคาของเครื่องรับจะสูงขึ้น และเพื่อให้ได้ความถูกต้องที่ดีที่สุด เครื่องรับสัญญาณควรจะต้องอยู่ในมุมมองที่ชัดเจนของท้องฟ้า

2) Antenna หมายถึง สายอากาศของเครื่องรับ เนื่องจากดาวเทียมอยู่ห่างออกไปไม่ต่ำกว่า 20,00 กิโลเมตร เพื่อประสิทธิภาพที่ดีที่สุดในการรับสัญญาณ สายอากาศของเครื่องรับสัญญาณจึงต้องมีประสิทธิภาพสูง ซึ่งมีให้เลือกหลากหลายชนิดด้วยกัน เช่น สายอากาศแบบแพทช์ (Patch Antenna) สายอากาศเฮลิคอล (Helical Antenna) สายอากาศภายนอก (Ext Antenna) ดังแสดงในรูปที่ 2.10



ก) สายอากาศแบบแพทช์



ข) สายอากาแบบเฮลิคอล

ค) สายอากาศภายนอก

รูปที่ 2.10 โมดูลเครื่องรับจีพีเอสและสายอากาศแบบต่างๆ

3) Baud Rate หมายถึง รูปแบบของการสื่อสารข้อมูลในการรับสัญญาณจากจีพีเอสจะเป็นแบบอนุกรม (UART) ในอัตราบิตเฉพาะ เช่น 9600 bps , 57600 bps, etc. ซึ่งตรวจสอบได้จากดาต้าชีท

4) Channel หมายถึง จำนวนช่องสัญญาณของเครื่องรับ เป็นคุณสมบัติที่ผลต่อเวลาของการกำหนดตำแหน่งในครั้งแรกที่เริ่มเปิดเครื่อง จำนวนช่องที่มากกว่าจะช่วยให้ใช้เวลาที่สั้นกว่า แต่หลังจากที่กำหนดตำแหน่งได้แล้ว ช่องสัญญาณบางช่องอาจจะต้องปิดเพื่อประหยัดพลังงาน

5) TTFF (Time To First Fix) หมายถึง เวลาที่ถูกใช้ตั้งแต่เริ่มเปิดเครื่องจนกระทั่งสามารถคำนวณหาตำแหน่งที่ถูกต้องได้ ซึ่งจะต้องใช้ข้อมูลจากดาวเทียมอย่างน้อย 4 ดวง เวลา TTFF นี้ยังขึ้นอยู่กับสภาพอากาศด้วย

6) Lock & Fix หมายถึง การค้นหาและติดตามตำแหน่งของดาวเทียม โดยถ้าหากรับสัญญาณจากดาวเทียมได้ตั้งแต่ 4 ดวงขึ้นไป จึงจะประมวลผลหาพิกัดตำแหน่งและเวลาที่ถูกต้องได้

7) Start-up Times (Hot, Warm, and Cold) – หมายถึง เวลาตั้งแต่เริ่มจ่ายไฟเข้าโมดูลจนกระทั่งสามารถ Fix & Lock ได้ โมดูลบางชนิดได้เพิ่มซุปเปอร์คาปาซิเตอร์ หรือ แบตเตอรี่ขนาดเล็กเพื่อเลี้ยงหน่วยความจำสำหรับเก็บข้อมูลตำแหน่งดาวเทียมหลังจากปิดเครื่อง ทำให้ลดเวลา TTFF และประหยัดพลังงาน

8) Update Rate หมายถึง ความถี่ในการรายงานพิกัดตำแหน่งของโมดูล (หน่วยเป็น Hz) ซึ่งโดยทั่วไปจะอยู่ที่ 1 ครั้งต่อวินาที หรือ 1 Hz ส่วน พวกยูเอวีหรือยานอัตโนมัติที่มีความเร็วสูงต้องการอัตราอัพเดตที่มากขึ้น อาจจะเลือกใช้โมดูล 5 Hz หรือ 10 Hz

2.3.2 การอ่านค่าพิกัดจากโมดูล GPS

เครื่องรับสัญญาณ GPS (GPS Module Receive) จะมีการส่งข้อมูลออกมาตลอดเวลาผ่านทางพอร์ตสื่อสารอนุกรม โดยใช้มาตรฐานของ NMEA (The National Marie Electronics Association) ไปยังเครื่องคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ ซึ่งข้อมูลประกอบด้วย PVT (Position Velocity Time) อยู่ในลักษณะประโยคข้อความรหัสแอสกี้ โดยแต่ละประโยคขึ้นต้นด้วยเครื่องหมาย $ และแต่ละรายการขั้นด้วยเครื่องหมายคอมม่า (,) และตามด้วยข้อมูลต่างๆ โดยประโยคข้อมูลสำคัญที่สามารถนำมาใช้ในการระบุพิกัดตำแหน่งได้นั้นจะเป็นประโยคที่ขึ้นต้นด้วย $GPGGA , $SGPGSA , $GPRMC และ $GPVTG ดังตัวอย่างในรูปที่ 2.11

รูปที่ 2.11 ประโยคข้อมูลที่ส่งออกมาจากโมดูลจีพีเอส

ข้อกำหนดของประโยค NMEA โดยทั่วไปมีดังนี้

1) ในแต่ละประโยค NMEA จะต้องขึ้นต้นด้วยเครื่องหมาย $ ก่อน

2) แต่ละประโยค NMEA จะต้องมีความยาวไม่เกิน 80 อักขระ

3) รายการข้อมูลจะถูกแยกด้วยเครื่องหมายคอมมา (,)

4) ข้อมูลจะมีลักษณะเป็นรหัส ASCII

5) มีการ Check Sum ที่ท้ายประโยค ซึ่งอาจจะเช็คหรือไม่เช็ค โดยการอ่านข้อมูล

6) การ Check Sum ประกอบด้วยเครื่องหมาย * และอีก 2 ตัวเลขฐาน 16 (Hex) แสดงการ Exclusive OR ของอักขระทั้งหมด

การเชื่อมต่อวงจรของฮาร์ดแวร์ เป็นการเชื่อมต่อแบบอนุกรมระหว่าง GPS และคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ โดยใช้มาตรฐาน RS232 ด้วยความเร็วในการส่งข้อมูลขึ้นอยู่กับคุณสมบัติของเครื่องรับ GPS แต่ละเครื่อง ทั้งนี้โปรโตคอลที่ใช้เป็นมาตรฐาน NMEA ที่ส่งออกมาจากโมดูล GPS มีลักษณะเป็นประโยคข้อความดังนี้

1) $GPGGA

เป็นประโยคที่เกี่ยวข้องกับตำแหน่งละติจูด และลองติจูด ซึ่งข้อมูลและความหมายดังนี้
$GPGGA,095733.705,0659.6165,N,10030.5223,E,1,07,1.0,8.4,M,-16.2,M,0000*7D

GGA ประเภทข้อมูลสำคัญ

095733.705 เวลา 09:57:33.705 UTC

0659.6165,N ละติจูด 06 องศาเหนือ 59.6165 ลิปดา

10030.5223,E ลองติจูด 100 องศาตะวันออก 30.5223 ลิปดา

1 = มีคุณภาพ 0 = ผิดพลาด

07 จำนวนของดาวเทียมที่ติดต่อ

1.0 ความเที่ยงตรงของตำแหน่งในแนวดิ่ง

8.4,M ความสูงเหนือระดับน้ำทะเล (เมตร)

-16.2,M ความสูงเหนือระดับน้ำทะเล แบบ WGSS584 (เมตร)

*7D Check sum

2) $GPGSA

เป็นประโยคเกี่ยวข้อกับข้อมูลของดาวเทียม ตัวอย่างประโยคเช่น
$GPGSA,A,3,02,28,10,04,27,05,17,,,,,2.7,1.0,2.5*3C จะมีความหมายดังนี้

A เลือกโดยอัตโนมติ 2D หรือ 3D fix (M = ควบคุมเอง)

3 3D fix

1 คือ (Not fix)

2 คือ 2 มิติ (2D fix)

3 คือ 3มิติ (3D fix)

02,28,100,04,...,17 คือ รหัส PRNs ของดาวเทียม

2.7 คือ ความเที่ยงตรง (PSOP)

1.0 คือ ความเที่ยงตรงในแนวราบ (HDOP)

2.5 คือ ความเที่ยงตรงในแนวดิ่ง (VDOP)

*3C คือ CheckSum ของข้อมูล ขึ้นต้นด้วย * เสมอ

3) $GPRMC

เป็นประโยคเกี่ยวกับตำแหน่งลองติจูด ละติจูดและความเร็วในการเคลื่อนที่ของตัวรับ GPS ซึ่งมีลักษณะประโยคและความหมายดังนี้

$GPRMC,095743,095743.705,A,0659.6166,N,10030.5223,E,000.0,000.0,171009,,,A *67

095743.705 เวลา 09:57:43.705 UTC

A สถานะ A = ทำงาน V = ไม่ทำงาน

0659.6166,N ละติจูด(Latitude) 06 องศาเหนือ 59.6166 ลิปดา

10030.5223,E ลองติจูด (Longtitude) 100 องศาตะวันออก 30.5223 ลิปดา

000.0 ความเร็วบนพื้นโลก (Knots)

000.0 มุมขอติดตามดาวเทียมในหน่วยองศา

171009 วันที่ 17 เดือนตุลาคม คศ. 2009

A Autonomous

*67 ตรวจผลรวมของข้อมูล CheckSum

4) $GPGSV

เป็นประโยคเกี่ยวกับ รายละเอียดของดาวเทียม GPS ที่รับได้ ซึ่งมีลักษณะประโยคและความหมายดังนี้

$GPGSV,3,1,12,42,44,095,37,02,38,320,38,28,34,150,45,10,54,210,47*7D

3 จำนวนประโยคสำหรับข้อมูลทั้งหมด

1 ประโยคที่ 1

12 จำนวนดาวเทียมที่รับได้

42 จำนวนดาวเทียม PRN

44 มุมเงย (Evaluation) องศา

095 มุมกวาด (Azimuth) องศา

37 ค่า SNR Signal to Noise Ratio

*7D CheckSum

5) $GPVTG

เป็นประโยคเกี่ยวกับความเร็วในการเคลื่อนที่ของตัวรับ GPS ซึ่งมีลักษณะประโยคและความหมายดังนี้

$GPVTG,000.0,T,000.0,M,000.0,N,000.0,K,A*0D

000.0,T ผลการติดตามวงโคจรดาวเทียม

000.0,M ผลการติดตามวงโคจรดาวเทียมแบบแม่เหล็ก

000.0,N	ความเร็วบนพื้นโลก หน่วยน๊อต (knots)
000.0,K	ความเร็วบนพื้นโลก กิโลเมตร/ชั่วโมง
*0D	Check Sum

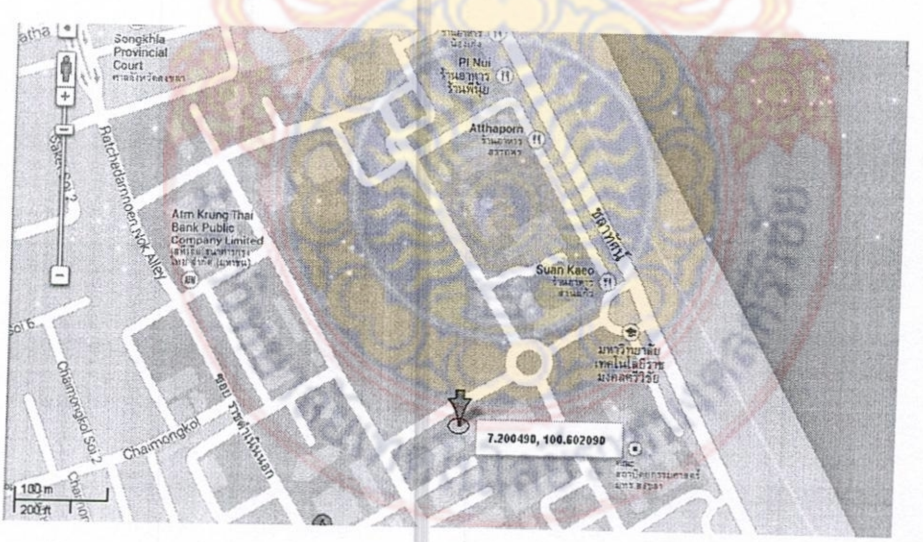
การหาค่าพิกัดละติจูดและลองติจูดเพื่อระบุพิกัดตำแหน่งนั้นสามารถนำข้อมูลจากประโยคของ $GPGGA มาแปลงเป็นค่าพิกัดลองติจูดและละติจูดตามดังสมการที่ 2

พิกัดละติจูด, ลองติจูด = องศา+(หน่วยลิปดา/60) (2)

เช่นหากได้รับข้อมูลเป็น $GPGGA,100952.412,0712.0294,N,10036.1254,E,1,07,1.8,26.4,M,-16.2,M,,0000*4E

ละติจูด = 07 +(12.0294/60)
= 07.200490 N
ลองติจูด = 100 + (36.1254/60)
= 100.602090E

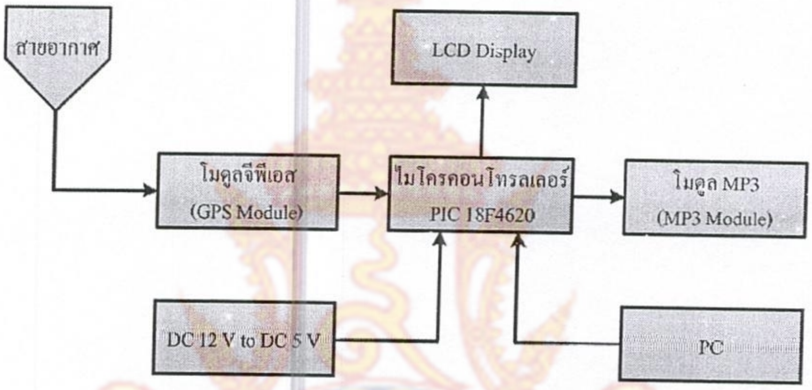
ซึ่งสามารถนำค่าพิกัดละติจูดและลองติจูดนี้ไปป้อนค่าค้นหาตำแหน่งในเวปไซต์แผนที่ Google map เพื่อหาตำแหน่งดังกล่าวได้ดังรูปที่ 2.11



รูปที่ 2.11 ตำแหน่งบนเวปไซต์แผนที่ พิกัด 07.200490N,100.602090E

บทที่ 3
การออกแบบระบบควบคุม

การออกแบบระบบบรรยายให้ข้อมูลและความรู้เกี่ยวกับแหล่งท่องเที่ยวจะบรรยายโดยอัตโนมัติเมื่อรถนำเที่ยวได้เดินทางมาถึงพื้นที่ของสถานที่ท่องเที่ยว ด้วยการตรวจสอบพิกัดตำแหน่งละติจูด ลองติจูด จากเครื่องรับสัญญาณจีพีเอส แล้วนำข้อมูลการบรรยายรายละเอียดของสถานที่ท่องเที่ยวต่างๆ จากหน่วยความจำ SD การ์ด รูปแบบ MP3 มาเล่นเสียงและเข้าสู่เครื่องขยายเสียง มีระบบดังในรูปที่ 3.1

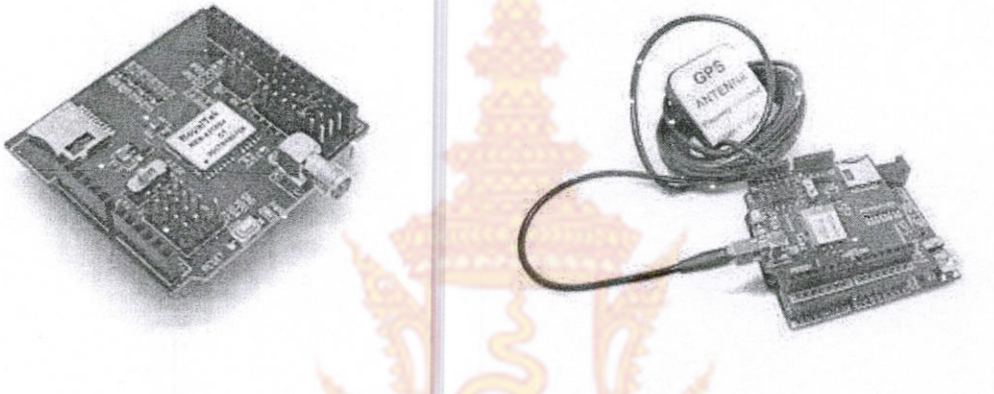


รูปที่ 3.1 ระบบควบคุมการทำงาน

3.1 การรับข้อมูลพิกัดจีพีเอส

การตรวจสอบตำแหน่งพิกัดละติจูด ลองติจูด ณ ตำแหน่งปัจจุบันนั้น ต้องอาศัยโมดูลจีพีเอส โดยเลือกใช้โมดูล GPS ของ Royaltek REB-4216 ซึ่งจะส่งข้อมูลเป็นประโยคข้อความมาตรฐาน NMEA ทุกๆ วินาที มายังขา RC2 ของไมโครคอนโทรลเลอร์เบอร์ PIC18F420 ทำหน้าที่รับสัญญาณอนุกรม ด้วยความเร็ว 9600 bps ดังลักษณะในรูปที่ 3.2

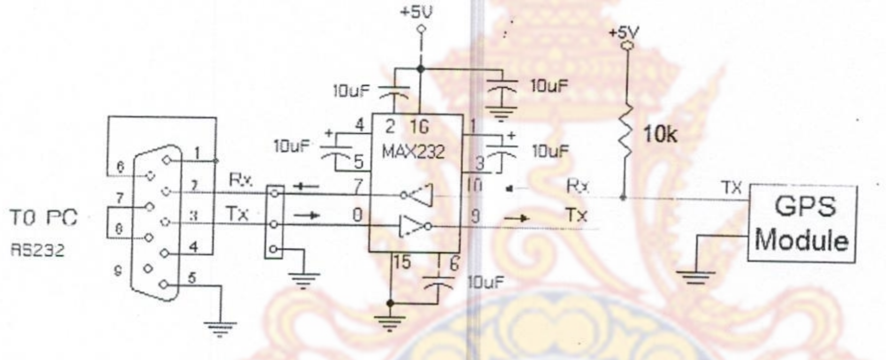
การอ่านข้อมูลจากเครื่องรับจีพีเอสต้องใช้การสื่อสารแบบอนุกรม ระหว่างเครื่องรับจีพีเอสกับไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม โดยโมดูลจีพีเอสจะมีการส่งประโยคข้อความทางขา Tx ซึ่งใช้ความต้านทาน 10 k ต่อพูลอัพเพื่อรักษาสถานะลอจิก 1 เมื่อไม่มีการสื่อสารข้อมูล โดยการสื่อสารข้อมูลจะใช้อัตราความเร็ว 9,600 bps ซึ่งเครื่องรับสัญญาณจีพีเอสจะส่งข้อมูลเป็นประโยคข้อความรหัสแอสกี้ไปยังไมโครคอนโทรลเลอร์ทุกๆ 1 วินาที เพื่อนำข้อมูลดังกล่าวไปใช้ในการประมวลผลพิกัดตำแหน่งละติจูดและลองติจูดต่อไป ซึ่งระบบฮาร์ดแวร์วงจรสื่อสารกับโมดูลจีพีเอสมีการต่อวงจรดังในรูปที่ 3.3



รูปที่ 3.2 โมดูลจีพีเอสและสายอากาศ

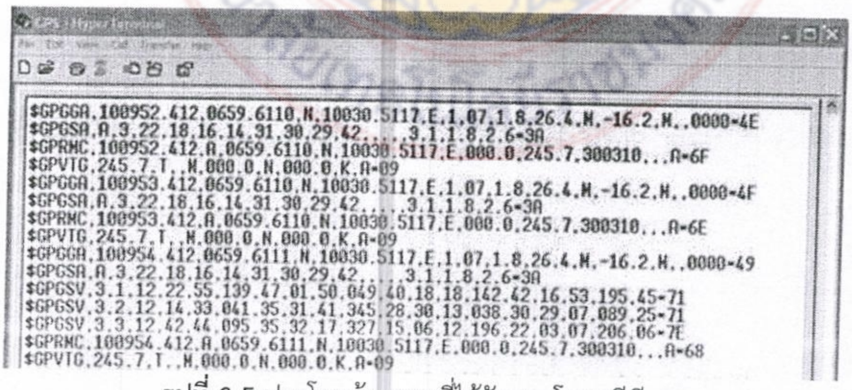
รูปที่ 3.3 วงจรสื่อสารกับโมดูลจีพีเอส

โมดูลจีพีเอสจะมีรอบการส่งข้อมูลที่ได้รับสัญญาณมาจากดาวเทียม เป็นลักษณะประโยคข้อความด้วยรหัสแอสกี้ (ASCII Code) ออกมายังพอร์ตอนุกรม หรือขา Tx ทุกๆ 1 วินาที ดังนั้นเราจึงทำให้สามารถรู้ตำแหน่งพิกัดลองติจูด หรือข้อมูลความเร็วในการเคลื่อนที่ที่เปลี่ยนแปลงไปทุกๆ 1 วินาที โดยระดับแรงดันที่ส่งออกมาจากขา Tx ของโมดูลจีพีเอสมีลักษณะเป็นสัญญาณมาตรฐาน TTL ดังนั้นจึงสามารถต่อเข้าโดยตรงกับขารับอินพุตแบบอนุกรมของไมโครคอนโทรลเลอร์ PIC18F420 โดยจะใช้ขา RC2 ทำหน้าที่สื่อสารข้อมูลกับโมดูลจีพีเอสในลักษณะการสื่อสารแบบอนุกรม ที่ใช้อัตราการสื่อสาร 9600 bps พาริตี้บิตเป็น None บิตหยุดเป็น 1 ตามการกำหนดค่าพื้นฐานของการใช้พอร์ต RS232 ทั้งนี้ทุกๆ วินาที ที่โมดูลจีพีเอส ส่งข้อมูลเป็นประโยคข้อความออกมานั้น หากต้องการให้ทราบว่าข้อความรหัสแอสกี้ที่ส่งออกมานั้นมีลักษณะอย่างไร สามารถใช้คอมพิวเตอร์รับค่าทางพอร์ตอนุกรมผ่านมาทางไอซีแปลงระดับสัญญาณ Max232 ดังลักษณะวงจรในรูปที่ 3.4



รูปที่ 3.4 วงจรรับข้อมูลจากโมดูลจีพีเอสแสดงผลบนจอภาพผ่านทางพอร์ตอนุกรม RS232

และใช้โปรแกรม Hyperterminal ของ Windows XP จะต้องมีการกำหนดค่าอัตราการสื่อสาร พาริตี้บิต และบิตหยุด ให้ตรงกันกับการส่งข้อมูลจากโมดูลจีพีเอส มิฉะนั้นจะทำให้การส่งข้อมูลเกิดข้อผิดพลาด และเมื่อสามารถเชื่อมต่อกับโปรแกรม Hyperterminal แล้ว ทุกๆวินาทีก็จะมีลักษณะการแสดงผลของประโยคข้อความรหัสแอสกี้บนหน้าต่างของโปรแกรม Hyperterminal ดังในรูปที่ 3.5



รูปที่ 3.5 ประโยคข้อความที่ได้รับจากโมดูลจีพีเอส

ในระบบรับข้อมูลจากโมดูลจีพีเอสได้เลือกใช้ขา RC2 ทำหน้าที่รับข้อมูลอนุกรมที่มีประโยคข้อความไหลเข้ามาทางขาดังกล่าว จึงต้องมีการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์รับสัญญาณที่ขา RC2 และตรวจสอบประโยคข้อความที่ขึ้นต้นด้วย $GPGGA เช่นหากได้รับข้อมูลเป็น $GPGGA,100952.412,0712.0294,N,10036.1254,E,1,07,1.8,26.4,M,-16.2,M,,0000*4E ก็จะทำสำเนามาเก็บไว้ในหน่วยความจำแรม เพื่อใช้สำหรับการดำเนินการขั้นตอนต่อไป

โดยประโยคที่ขึ้นต้นด้วย $GPGGA จะมีข้อมูลตำแหน่งละติจูด และลองติจูด โดยหากสามารถระบุค่าของละติจูด และลองติจูดได้แล้ว ก็จะนำมาคำนวณหาค่าลองติจูดและละติจูดตามสมการที่ (1) โดยมีกระบวนการทำงานดังโฟลวชาร์ทในรูปที่ 3.6

พิกัดละติจูด, ลองติจูด = องศา+(หน่วยลิปดา/60) (1)

จะสามารถคำนวณหาค่าลองติจูดและละติจูดดังนี้

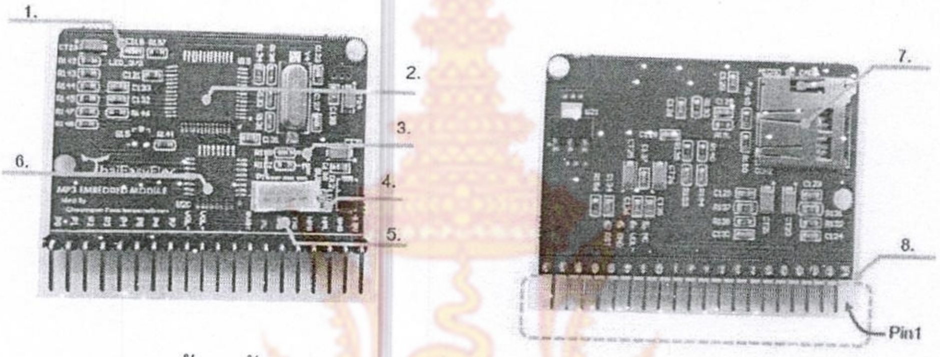
ละติจูด = 07 +(12.294/60)
= 07.200490 N
ลองติจูด = 100 + (36.1254/60)
= 100.602090E

รูปที่ 3.6 โฟลวชาร์ทการจับข้อมูลตำแหน่งละติจูด ลองติจูดจากโมดูลจีพีเอส

90256

3.2 การควบคุมเครื่องเล่นเสียง MP3

การควบคุมเครื่องเล่นเสียง MP3 ได้ใช้ตัวเล่นเสียงแบบโมดูล ยี่ห้อ ThaiEasyElec MP3 Embedded Module เนื่องจากหาซื้อง่ายและราคาถูก สามารถควบคุมได้ทั้งโหมดขนาน (Parallel) และแบบอนุกรม (Serial) รองรับความจุของ micro SD Card ได้ถึง 32 GByte อัตราการสุ่มสัญญาณเสียง (Sampling Rate) ที่ 8 kHz – 48 kHz ใช้แหล่งจ่ายไฟ 5 โวลท์ และมีแผ่นวงจรขนาดเล็ก ดังรูปที่ 3.7



ด้านหน้า ด้านหลัง

รูปที่ 3.7 โมดูลเล่นเสียง MP3

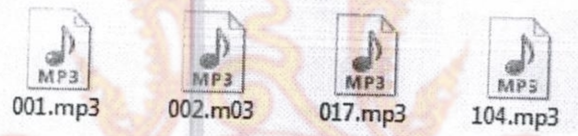
โดยบอร์ดโมดูล MP3 มีส่วนประกอบดังนี้
หมายเลข 1 คือ LED แสดงผลเมื่อมีไฟเลี้ยง 5 โวลท์
หมายเลข 2 คือ ไอซี ถอดรหัส (Decode) MP3 เบอร์ BU94502AKS2
หมายเลข 3 คือ จุดบัดกรีเลขโหมด ระหว่าง โหมดขนาด หรือ อนุกรม
หมายเลข 4 คือ LED แสดงสถานะ BUSY
หมายเลข 5 พอร์ต DEBUG MCU
หมายเลข 6 คือ MCU เบอร์ STM8S103
หมายเลข 7 คือ ช็อกเก็ตสำหรับใส่ Mini SD Card
หมายเลข 8 คือ ขาสัญญาณควบคุม ซึ่งประกอบด้วยชื่อขาและหน้าที่ดังตารางที่ 3.1

ตารางที่ 3.1 ตำแหน่งขาของโมดูลเล่นเสียง MP3

ขาที่	ชื่อ	หน้าที่
1 - 8	P0 – P7	ขาเลือกลำดับไฟล์เสียง
9	VOL-	ลดระดับเสียง
10	VOL+	เพิ่มระดับเสียง
11	RESERVE	ไม่ใช้

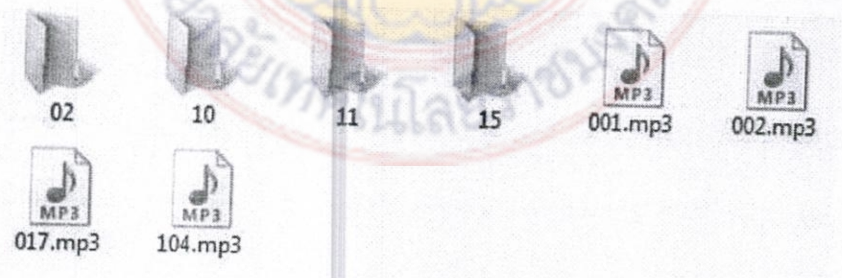
12	RESERVE	ไม่ใช้
13	BUSY	แสดงสถานะการเล่นไฟล์
14	PL	เริ่มเล่นไฟล์เสียง (Active Low)
15	RXD	ขา RX สำหรับพอร์ตอนุกรม
16	TXD	ขา TX สำหรับพอร์ตอนุกรม
17	HPR	สัญญาณเสียงด้านขวา
18	HPL	สัญญาณเสียงด้านซ้าย
19	GND	กราวน์
20	+5V	ไฟเลี้ยง 5 โวลท์

การเตรียมไฟล์บน์ SD Cardจะต้องตั้งชื่อไฟล์เป็นลำดับตัวเลข เพื่อให้การเลือกจะเลือกตามลำดับตัวเลข ที่ระบุลำดับด้วยขา P0 – P7 ซึ่งสามารถเล่นไฟล์ได้ตั้งแต่ลำดับที่ 1 ถึงลำดับไฟล์ที่ 199 โดยไฟล์เสียงลำดับที่ 1 ต้องตั้งชื่อไฟล์เป็น 001. MP3 และเรียงลำดับมาเป็น 002.MP3 และเรียงลำดับไปเรื่อยๆจนถึง 199.MP3ดังรูปที่ 3.8



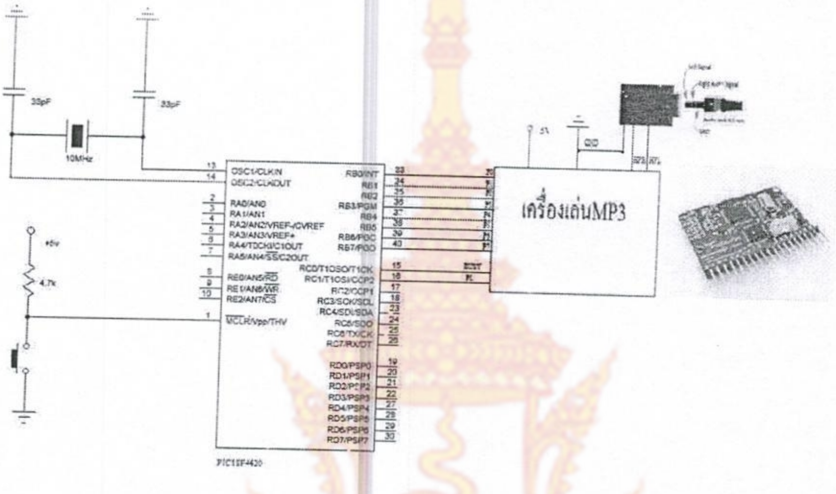
รูปที่ 3.8 การเรียงลำดับของไฟล์เสียง

โมดูลเล่นเสียง MP3 นี้สามารถเล่นไฟล์ได้สูงสุด 199 ไฟล์ต่อโฟลเดอร์ ซึ่งสามารถมีโฟลเดอร์ได้ 15 โฟลเดอร์ ดังนั้นจึงทำให้สามารถควบคุมการเล่นไฟล์เสียงได้ทั้งหมดเป็น 199 x 15 เท่ากับ 2,985 ไฟล์เสียง และการตั้งชื่อโฟลเดอร์ก็ต้องตั้งเป็นลำดับตัวเลข โดยโฟลเดอร์แรกจะเป็นรากโฟลเดอร์ (Root Directory) นั่นหมายความว่า โฟลเดอร์ลำดับแรกสุดต้องตั้งชื่อโฟลเดอร์เป็น 02 ดังลักษณะในรูปที่ 3.9



รูปที่ 3.9 การตั้งชื่อโฟลเดอร์ของไฟล์เสียง

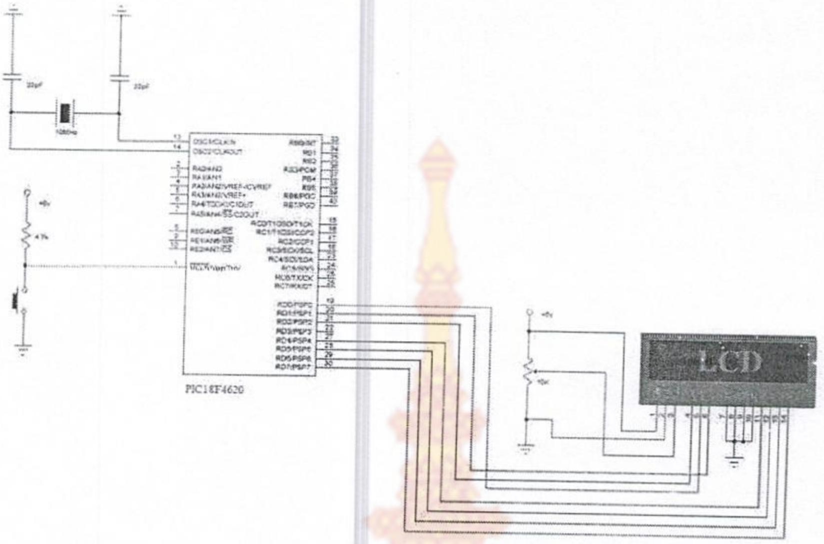
การควบคุมโมดูล MP3 ให้เล่นเสียงด้วยไมโครคอนโทรลเลอร์ PIC18F4620 ได้เลือกการควบคุมด้วยโหมดขนาน โดยใช้พอร์ต B (ขา RB0 – RB7) ของไมโครคอนโทรลเลอร์ทำหน้าที่เลือกไฟล์เสียงที่ต้องการเล่น และใช้ขา RC0 ทำหน้าที่ตรวจสอบสถานะ BUSY ของโมดูล MP3 และหากเลือกลำดับของไฟล์เสียงเรียบร้อยแล้ว จะใช้ขา RC1 ทำหน้าที่สั่งเล่นไฟล์เสียง โดยมีวงจร ดังในรูปที่ 3.10



รูปที่ 3.10 วงจรควบคุมการเล่นไฟล์เสียง MP3

3.3 การแสดงผล

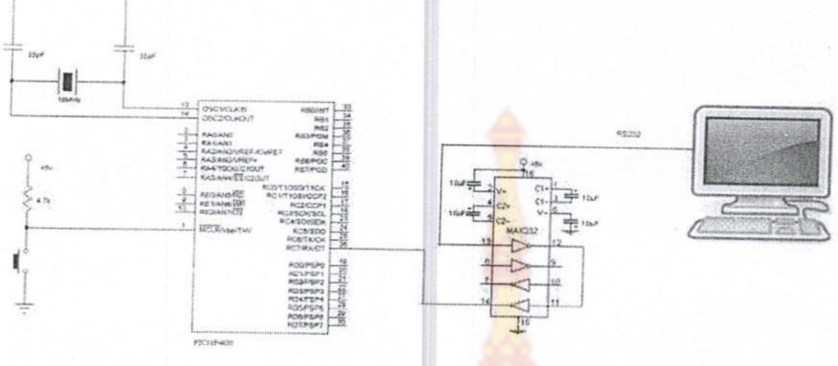
การแสดงผลค่าตำแหน่งพิกัดละติจูด ลองติจูดได้ออกแบบให้ใช้จอ LCD ขนาด 16 ตัวอักษร 2 บรรทัด ซึ่งสามารถแสดงได้ทั้งค่าตัวเลขและตัวอักษรภาษาอังกฤษ โดยใช้พอร์ต C ของไมโครคอนโทรลเลอร์ทำหน้าที่ควบคุมการแสดงผลค่าละติจูด ลองติจูด ณ ตำแหน่งปัจจุบัน ดังนั้นหากมีการเปลี่ยนพิกัดตำแหน่ง ก็จะทำให้ทราบพิกัดละติจูดลองติจูดที่เปลี่ยนแปลงไป โดยมีวงจรดังรูปที่ 3.11



รูปที่ 3.11 วงจรการแสดงผลด้วย LCD

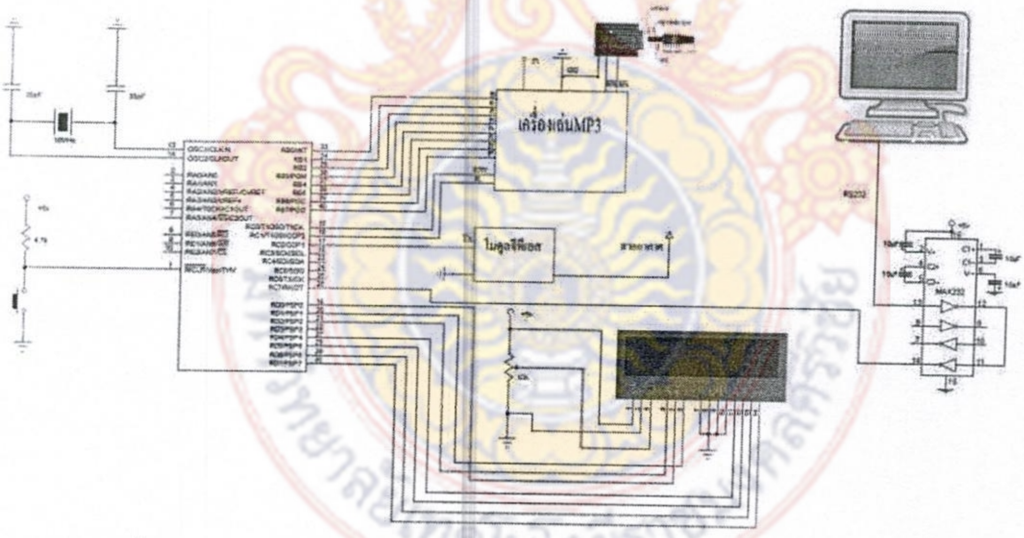
3.4 การป้อนพิกัดสถานที่ท่องเที่ยว

การบันทึกพิกัดสถานที่ท่องเที่ยวลงในหน่วยความจำ EEPROM ของไมโครคอนโทรลเลอร์ได้ทำการออกแบบให้ ไมโครคอนโทรลเลอร์รับข้อมูลผ่านทางพอร์ตอนุกรม โดยรับข้อมูลมาจากไมโครคอมพิวเตอร์ด้วยโปรแกรมที่ได้พัฒนาขึ้นจาก วิสชวลเบสิค 6 (Visual Basic 6) ผ่านทางพอร์ตอนุกรม RS232 ด้วยความเร็ว 9600 bps แล้วนำพิกัดละติจูด ลองติจูด ของสถานที่ท่องเที่ยวแต่ละที่มาเก็บไว้ในหน่วยความจำ EEPROM เพื่อใช้สำหรับการตรวจสอบเปรียบเทียบกับพิกัดตำแหน่งละติจูด ลองติจูดปัจจุบัน ที่ได้มาจากโมดูล GPS โดยวงจรการโหลดพิกัดข้อมูล ละติจูด ลองติจูดของสถานที่ท่องเที่ยวมีลักษณะวงจรดังในรูปที่ 3.12

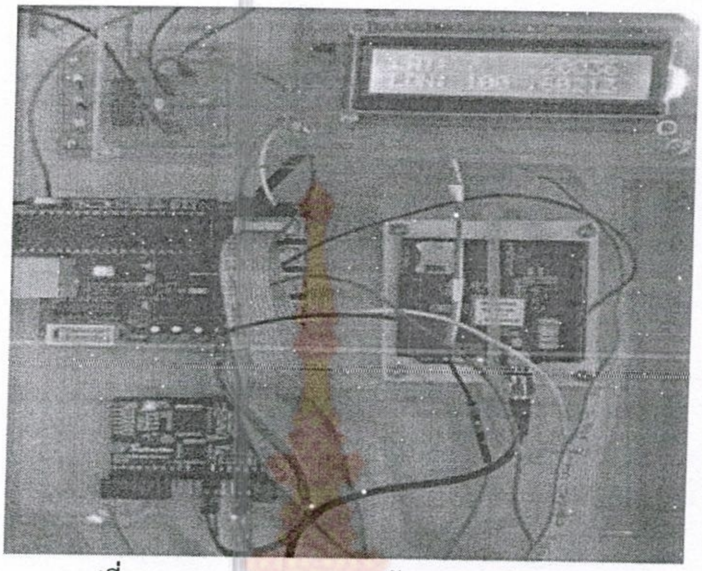


รูปที่ 3.12 วงจรการบันทึกพิกัดสถานที่ท่องเที่ยว

เมื่อนำวงจรการทำงานแต่ละส่วนมารวมกันเป็นวงจรเดียวกัน เพื่อให้ทำงานสอดคล้องกันตามที่ได้ออกแบบไว้ จะมีลักษณะดังแสดงในรูปที่ 3.13 ลักษณะการต่อวงจรจริงดังรูปที่ 3.14



รูปที่ 3.13 วงจรระบบบรรยายข้อมูลสถานที่ท่องเที่ยวอัตโนมัติด้วยพิกัดจีพีเอส



รูปที่ 3.14 การประกอบวงจรด้วยอุปกรณ์

หลังจากทดสอบการทำงานของอุปกรณ์ต่างๆ ทำงานได้ดังที่เขียนโปรแกรมควบคุมการทำงานไว้แล้ว นั้น จึงได้ดำเนินการบรรจุลงในกล่อง ป้องกันสิ่งที่จะมากระทบกระเทือนแก่อุปกรณ์ โดยมีลักษณะการประกอบดังในรูปที่ 3.15

รูปที่ 3.15 การประกอบลงกล่อง

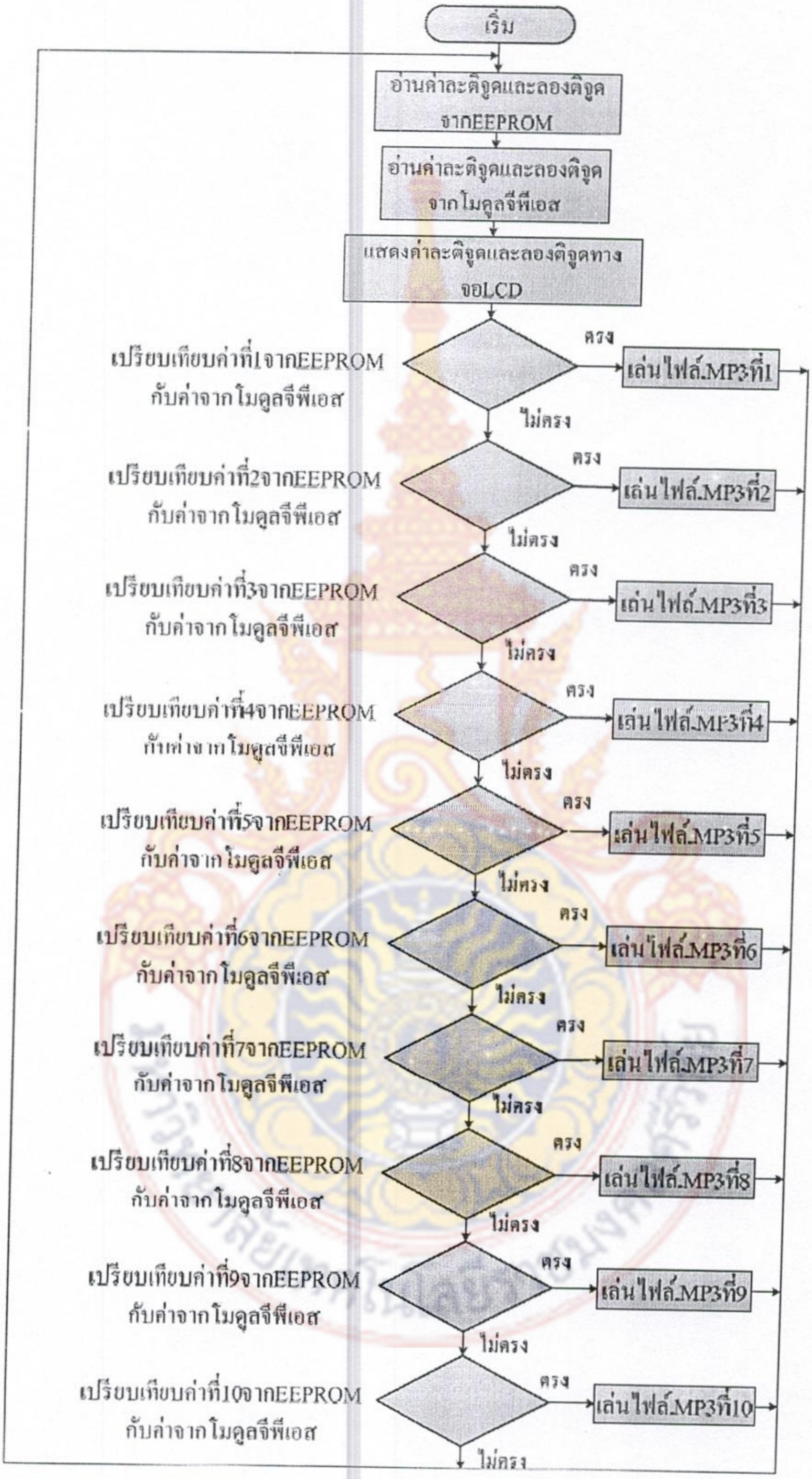
3.5 โปรแกรมป้อนพิกัดสถานที่ท่องเที่ยว

การออกแบบซอฟแวร์สำหรับการป้อนค่าพิกัดสถานที่ท่องเที่ยว ได้พัฒนาขึ้นมาจาก Visual Basic6 เป็นภาษาที่นิยมใช้สร้างโปรแกรมประยุกต์อย่างแพร่หลาย มีสื่อและเครื่องมือสำหรับการเรียนรู้จำนวนมาก โดยได้ออกแบบให้ทำหน้าที่รับพิกัดตำแหน่งละติจูด ลองติจูด จากผู้ใช้งานผ่านทางคีย์บอร์ด ซึ่งจะทำให้การป้อนข้อมูลมีความสะดวก ง่ายต่อการใช้งาน เนื่องจากการป้อนข้อมูลบนเครื่องคอมพิวเตอร์นั้น มีหน้าจอแสดงผล และคีย์บอร์ดรองรับการใช้งานอยู่แล้ว เพียงแต่ต้องพัฒนาโปรแกรมที่ทำหน้าที่รับค่าพิกัดละติจูด ลองติจูด ของสถานที่ท่องเที่ยวต่างๆจากเจ้าหน้าที่ผู้ใช้งานเครื่องมือดังกล่าว โดยโปรแกรมจะสร้างการเชื่อมโยงไปยังชื่อไฟล์ของเสียงบรรยายให้อย่างอัตโนมัติ เพื่อให้การบรรยายข้อมูลเป็นได้อย่างถูกต้อง หลังจากป้อนพิกัดละติจูด ลองติจูดของสถานที่ท่องเที่ยวแล้ว ก็สามารถโหลดข้อมูลพิกัดของสถานที่ท่องเที่ยว ผ่านทางพอร์ตอนุกรม ไปเก็บไว้ยังหน่วยความจำ EEPROM ของไมโครคอนโทรลเลอร์ PIC18F420 เพื่อใช้เป็นพิกัดสำหรับการตรวจสอบเปรียบเทียบกับตำแหน่งปัจจุบันจากโมดูลจีพีเอสต่อไป

รูปที่ 3.16 โปรแกรมการป้อนพิกัดละติจูด ลองติจูด ของสถานที่ท่องเที่ยว

3.6 การทำงานของโปรแกรมไมโครคอนโทรลเลอร์

การออกแบบระบบควบคุมให้โมดูล MP3 สามารถเล่นเสียงบรรยายข้อมูลของสถานที่ท่องเที่ยวแต่ละแหล่งนั้น ไมโครคอนโทรลเลอร์จะอ่านค่าข้อมูลที่รับมาจากโมดูล GPS โดยกรองเอาเฉพาะประโยคข้อความที่มีข้อมูล ละติจูด ลองติจูด มาทำการแสดงผลตำแหน่งบนหน้าจอ LCD พร้อมกับเปรียบเทียบกับ ตำแหน่งละติจูด ลองติจูดของสถานที่ท่องเที่ยวแต่ละจุด หากตำแหน่งละติจูด ลองติจูด ตำแหน่งปัจจุบันไม่ตรงกับพิกัดละติจูด ลองติจูดในหน่วยความจำ EEPROM ก็ให้ทำการวนตรวจสอบในรอบถัดไป แต่ หากไปตรงกับพิกัดตำแหน่งของสถานที่ท่องเที่ยวใดไมโครคอนโทรลเลอร์ ก็จะควบคุมให้โมดูล MP3 ทำการเล่นไฟล์เสียงของสถานที่ท่องเที่ยวนั้น ๆ อย่างอัตโนมัติ โดยมีกระบวนการทำงานของโปรแกรมดังโฟลวชาร์ทในรูปที่ 3.17 และมีโปรแกรมที่เขียนขึ้นโดยทำงานบนไมโครคอนโทรลเลอร์ PIC18F120 ดังนี้



รูปที่ 3.17 โฟลวชาร์ทการทำงานของระบบบรรยายสถานที่ท่องเที่ยวอัตโนมัติด้วย GPS

```
#include <18f4620.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#use delay(clock=10000000)
#fuses HS,NOPROTECT,NOWDT,NOBROWNOUT
#use rs232(baud=9600,xmit=pin_c4,rcv=pin_c2,stream=HOSTPC,errors)
#use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7,stream=PROGRAMPC,errors)
#define use_portb_lcd
#include <lcd.c>
char buffer[180];
float Lat,Lon;
char c,k,d;
char user1[5],user2[5],user3[5],user4[5];
float tu1,tu2,tu3,tu4,tu5,tu6 ;
char lati[9],lont[10] ;
int1 busy,pl;
float latstation01, latstation02, latstation03, latstation04, latstation05,
latstation06, latstation07, latstation08, latstation09, latstation10,
logstation01, logstation02, logstation03, logstation04, logstation05,
logstation06, logstation07, logstation08, logstation09, logstation10 ;
///////////////////////////////////////////////////////////////////////GPS
void Readmessage ()
{
float f0,f1;
int16 i0,i1;
char sentent[]="GPRMC";
do
{
while ( fgetc(HOSTPC) != '$' );
```

```
for (k=0;k<6;k++)
buffer[k]=fgetc(HOSTPC);
}
while (strncmp(buffer,sentent,5)!= 0);
k=0;
c=0;
while (c !='*'&&k<79)
{
c = fgetc(HOSTPC);
buffer[k++]= c;
}
for (k = 13; k <=21; k++)
lati [k-13] = buffer[k] ;
for (k = 25; k <=34; k++)
{
lont[k-25] = buffer[k];
}
f0 = atof(lati);
f1 = atof(lont);
f0 = (f0 / 100) ;
f1 = (f1 / 100) ;
i0 = (int16)f0 ;
i1 = (int16)f1 ;
Lat = i0+((f0 -i0 )*100)/60 ;
Lon = i1+((f1 -i1 )*100)/60 ;
fprintf(HOSTPC,"gps\n\r");
{
fprintf(HOSTPC,"%.5f\n",Lat);
fprintf(HOSTPC,"\n\r");
fprintf(HOSTPC,"%.5f\n",Lon);
```

```
fprintf(HOSTPC,"\n\r");
}
{
fprintf(HOSTPC,"\n\r");
}
}
////////////////////////////////////////////////////////////////////////mp3
void mp()
{
set_tris_c(0x01);
set_tris_d(0x00);
busy = input(PIN_c0);
int1 cmd_stop = 0;
int u = 3;
output_high(pin_c1);
{
/////////////////////////////////////////////////////////////////////1
if( ((Lat>=(latstation01-
0.00024))&&(Lat<=(latstation01+0.00024)))&&((Lon>=(logstation01-
0.00024))&&(Lon<=(logstation01+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x01);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 1\n\r");
```

```
}
}
//////////////////////////////////////////////////////////////////////////2
else if( ((Lat>=(latstation02-
0.00024))&&(Lat<=(latstation02+0.00024)))&&((Lon>=(logstation02-
0.00024))&&(Lon<=(logstation02+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x02);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 2\n\r");
}
}
//////////////////////////////////////////////////////////////////////////3
else if( ((Lat>=(latstation03-
0.00024))&&(Lat<=(latstation03+0.00024)))&&((Lon>=(logstation03-
0.00024))&&(Lon<=(logstation03+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x03);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
```

```
cmd_stop = true;
fprintf(HOSTPC,"mp3 3\n\r");
}
}
//////////////////////////////////////////////////////////////////////4
else if( ((Lat>=(latstation04-
0.00024))&&(Lat<=(latstation04+0.00024)))&&((Lon>=(logstation04-
0.00024))&&(Lon<=(logstation04+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x04);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 4\n\r");
}
}
else if( ((Lat>=(latstation05-
0.00024))&&(Lat<=(latstation05+0.00024)))&&((Lon>=(logstation05-
0.00024))&&(Lon<=(logstation05+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x05);
delay_us(200);
output_low(pin_c1);
delay_us(200);
```

```
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 5\n\r");
}
}
///////////////////////////////////////////////////////////////////////////6
else if( ((Lat>=(latstation06-
0.00024))&&(Lat<=(latstation06+0.00024)))&&((Lon>=(logstation06-
0.00024))&&(Lon<=(logstation06+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x06);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 6\n\r");
}
}
///////////////////////////////////////////////////////////////////////////7
else if( ((Lat>=(latstation07-
0.00024))&&(Lat<=(latstation07+0.00024)))&&((Lon>=(logstation07-
0.00024))&&(Lon<=(logstation07+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x07);
delay_us(200);
```

```
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 7\n\r");
}
}
//////////////////////////////////////////////////////////////////////////8
else if( ((Lat>=(latstation08-
0.00024))&&(Lat<=(latstation08+0.00024)))&&((Lon>=(logstation08-
0.00024))&&(Lon<=(logstation08+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
output_d(0x08);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 8\n\r");
}
}
//////////////////////////////////////////////////////////////////////////9
else if( ((Lat>=(latstation09-
0.00024))&&(Lat<=(latstation09+0.00024)))&&((Lon>=(logstation09-
0.00024))&&(Lon<=(logstation09+0.00024))) ) //latitude longitude +- 0.00024
{
if ((busy==1) && (cmd_stop==0))
{
```

```
output_d(0x09);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 9\n\r");
}
}
//////////////////////////////////////////////////////////////////////////////10
else if( ((Lat>=(latstation10-
0.00024))&&(Lat<=(latstation10+0.00024)))&&((Lon>=(logstation10-
0.00024))&&(Lon<=(logstation10+0.00024))) ) //latitude longitude +- 0.00024
{
if((busy==1) && (cmd_stop==0))
{
output_d(0x0A);
delay_us(200);
output_low(pin_c1);
delay_us(200);
output_high(pin_c1);
cmd_stop = true;
fprintf(HOSTPC,"mp3 10\n\r");
}
}
//////////////////////////////////////////////////////////////////////////////else
{
cmd_stop = true;
fprintf(HOSTPC,"mp else\n\r");
}
```

```
}
fprintf(HOSTPC,"mp3\n\r");
}
////////////////////////////////////////////////////////////////////////////////lcd
void lcd()
{
char a[17];
char b[17];
char c[17];
char d[17];
int32 str1 , str2 , str3 , str4 ;
int i ;
int x ;
int y ;
{
/////////////////lat//////////////////////////////
lcd_init();
lcd_send_byte(0,0x01);
a[4] = ".";
str1 = Lat;
str2 = (Lat-str1)*1000000;
itoa(str1,10,a);
itoa(str2,10,b);
for(x=0;x<5;x++)
{
a[x+5] =b[x];
}
lcd_send_byte(0,0x80);
lcd_putc("LAT:\n");
lcd_gotoxy(6,1);
```

```
for (i = 0; i <=16; i++)
{
lcd_send_byte(1, a[i]);
}
lcd_gotoxy(16,1);
lcd_putc(" \n");
lcd_gotoxy(7,1);
lcd_putc(" \n");
/////////////////Lon///////////////////////////
c[4] = ".";
str3 = Lon;
str4 = (Lon-str3)*1000000;
itoa(str3,10,c);
itoa(str4,10,d);
for(x=0;x<5;x++)
{
c[x+5] =d[x];
}
lcd_send_byte(0,0xc0);
lcd_putc("LON:\n");
lcd_gotoxy(6,2);
for (i = 0; i <=16; i++)
{
lcd_send_byte(1, c[i]);
}
lcd_gotoxy(16,2);
lcd_putc(" \n");
lcd_gotoxy(9,2);
lcd_putc(" \n");
}
```

```
}
////////////////////////////////////////////////////////////interrub1
void writeeeprom ()
{
char sentent0[]="AAAAA";
do
{
while ( fgetc(PROGRAMPC) != '$' );
for (k=0;k<6;k++)
buffer[k]=fgetc(PROGRAMPC);
}
while (strncmp(buffer,sentent0,5)!= 0);
k=0;
c=0;
while (c !='*'&&k<180)
{
c = fgetc(PROGRAMPC);
buffer[k++]= c;
}
/////////////////////////////
{
set_tris_a(0x00);
output_high(pin_a1);
delay_ms(10);
write_eeprom(0, buffer[0]);
delay_ms(10);
write_eeprom(1, buffer[1]);
delay_ms(10);
write_eeprom(2, buffer[2]);
delay_ms(10);
```

```
write_eeprom(3, buffer[3]);
delay_ms(10);
write_eeprom(4, buffer[4]);
delay_ms(10);
write_eeprom(5, buffer[5]);
delay_ms(10);
writc_ccprom(6, buffcr[6]);
delay_ms(10);
write_eeprom(8, buffer[8]);
delay_ms(10);
write_eeprom(9, buffer[9]);
delay_ms(10);
write_eeprom(10, buffer[10]);
delay_ms(10);
write_eeprom(11, buffer[11]);
delay_ms(10);
write_eeprom(12, buffer[12]);
delay_ms(10);
write_eeprom(13, buffer[13]);
delay_ms(10);
write_eeprom(14, buffer[14]);
delay_ms(10);
write_eeprom(15, buffer[15]);
delay_ms(10);
write_eeprom(16, buffer[16]);
delay_ms(10);
write_eeprom(18, buffer[18]);
delay_ms(10);
write_eeprom(19, buffer[19]);
delay_ms(10);
```

```
write_eeprom(20, buffer[20]);
delay_ms(10);
write_eeprom(21, buffer[21]);
delay_ms(10);
write_eeprom(22, buffer[23]);
delay_ms(10);
write_eeprom(23, buffer[23]);
delay_ms(10);
write_eeprom(24, buffer[24]);
delay_ms(10);
write_eeprom(26, buffer[26]);
delay_ms(10);
write_eeprom(27, buffer[27]);
delay_ms(10);
write_eeprom(28, buffer[28]);
delay_ms(10);
write_eeprom(29, buffer[29]);
delay_ms(10);
write_eeprom(30, buffer[30]);
delay_ms(10);
write_eeprom(31, buffer[31]);
delay_ms(10);
write_eeprom(32, buffer[32]);
delay_ms(10);
write_eeprom(33, buffer[33]);
delay_ms(10);
write_eeprom(34, buffer[34]);
delay_ms(10);
write_eeprom(36, buffer[36]);
delay_ms(10);
```

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

```
write_eeprom(37, buffer[37]);
delay_ms(10);
write_eeprom(38, buffer[38]);
delay_ms(10);
write_eeprom(39, buffer[39]);
delay_ms(10);
write_eeprom(40, buffer[40]);
delay_ms(10);
write_eeprom(41, buffer[41]);
delay_ms(10);
write_eeprom(42, buffer[42]);
delay_ms(10);
write_eeprom(44, buffer[44]);
delay_ms(10);
write_eeprom(45, buffer[45]);
delay_ms(10);
write_eeprom(46, buffer[46]);
delay_ms(10);
write_eeprom(47, buffer[47]);
delay_ms(10);
write_eeprom(48, buffer[48]);
delay_ms(10);
write_eeprom(49, buffer[49]);
delay_ms(10);
write_eeprom(50, buffer[50]);
delay_ms(10);
write_eeprom(51, buffer[51]);
delay_ms(10);
write_eeprom(52, buffer[52]);
delay_ms(10);
```

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

```
write_eeprom(54, buffer[54]);
delay_ms(10);
write_eeprom(55, buffer[55]);
delay_ms(10);
write_eeprom(56, buffer[56]);
delay_ms(10);
write_eeprom(57, buffer[57]);
delay_ms(10);
write_eeprom(58, buffer[58]);
delay_ms(10);
write_eeprom(59, buffer[59]);
delay_ms(10);
write_eeprom(60, buffer[60]);
delay_ms(10);
write_eeprom(62, buffer[62]);
delay_ms(10);
write_eeprom(63, buffer[63]);
delay_ms(10);
write_eeprom(64, buffer[64]);
delay_ms(10);
write_eeprom(65, buffer[65]);
delay_ms(10);
write_eeprom(66, buffer[66]);
delay_ms(10);
write_eeprom(67, buffer[67]);
delay_ms(10);
write_eeprom(68, buffer[68]);
delay_ms(10);
write_eeprom(69, buffer[69]);
delay_ms(10);
```

```
write_eeprom(70, buffer[70]);
delay_ms(10);
write_eeprom(72, buffer[72]);
delay_ms(10);
write_eeprom(73, buffer[73]);
delay_ms(10);
write_eeprom(74, buffer[74]);
delay_ms(10);
write_eeprom(75, buffer[75]);
delay_ms(10);
write_eeprom(76, buffer[76]);
delay_ms(10);
write_eeprom(77, buffer[77]);
delay_ms(10);
write_eeprom(78, buffer[78]);
delay_ms(10);
write_eeprom(80, buffer[80]);
delay_ms(10);
write_eeprom(81, buffer[81]);
delay_ms(10);
write_eeprom(82, buffer[82]);
delay_ms(10);
write_eeprom(83, buffer[83]);
delay_ms(10);
write_eeprom(84, buffer[84]);
delay_ms(10);
write_eeprom(85, buffer[85]);
delay_ms(10);
write_eeprom(86, buffer[86]);
delay_ms(10);
```

```
write_eeprom(87, buffer[87]);
delay_ms(10);
write_eeprom(88, buffer[88]);
delay_ms(10);
write_eeprom(90, buffer[90]);
delay_ms(10);
write_eeprom(91, buffer[91]);
delay_ms(10);
write_eeprom(92, buffer[92]);
delay_ms(10);
write_eeprom(93, buffer[93]);
delay_ms(10);
write_eeprom(94, buffer[94]);
delay_ms(10);
write_eeprom(95, buffer[95]);
delay_ms(10);
write_eeprom(96, buffer[96]);
delay_ms(10);
write_eeprom(98, buffer[98]);
delay_ms(10);
write_eeprom(99, buffer[99]);
delay_ms(10);
write_eeprom(100, buffer[100]);
delay_ms(10);
write_eeprom(101, buffer[101]);
delay_ms(10);
write_eeprom(102, buffer[102]);
delay_ms(10);
write_eeprom(103, buffer[103]);
delay_ms(10);
```

```
write_eeprom(104, buffer[104]);
delay_ms(10);
write_eeprom(105, buffer[105]);
delay_ms(10);
write_eeprom(106, buffer[106]);
delay_ms(10);
write_eeprom(108, buffer[108]);
delay_ms(10);
write_eeprom(109, buffer[109]);
delay_ms(10);
write_eeprom(110, buffer[110]);
delay_ms(10);
write_eeprom(111, buffer[111]);
delay_ms(10);
write_eeprom(112, buffer[112]);
delay_ms(10);
write_eeprom(113, buffer[113]);
delay_ms(10);
write_eeprom(114, buffer[114]);
delay_ms(10);
write_eeprom(116, buffer[116]);
delay_ms(10);
write_eeprom(117, buffer[117]);
delay_ms(10);
write_eeprom(118, buffer[118]);
delay_ms(10);
write_eeprom(119, buffer[119]);
delay_ms(10);
write_eeprom(120, buffer[120]);
delay_ms(10);
```

```
write_eeprom(121, buffer[121]);
delay_ms(10);
write_eeprom(122 , buffer[122]);
delay_ms(10);
write_eeprom(123, buffer[123]);
delay_ms(10);
write_eeprom(124, buffer[124]);
delay_ms(10);
write_eeprom(126, buffer[126]);
delay_ms(10);
write_eeprom(127, buffer[127]);
delay_ms(10);
write_eeprom(128, buffer[128]);
delay_ms(10);
write_eeprom(129, buffer[129]);
delay_ms(10);
write_eeprom(130, buffer[130]);
delay_ms(10);
write_eeprom(131, buffer[131]);
delay_ms(10);
write_eeprom(132, buffer[132]);
delay_ms(10);
write_eeprom(134, buffer[134]);
delay_ms(10);
write_eeprom(135, buffer[135]);
delay_ms(10);
write_eeprom(136, buffer[136]);
delay_ms(10);
write_eeprom(137, buffer[137]);
delay_ms(10);
```

```
write_eeprom(138, buffer[138]);
delay_ms(10);
write_eeprom(139, buffer[139]);
delay_ms(10);
write_eeprom(140 , buffer[140]);
delay_ms(10);
write_eeprom(141, buffer[141]);
delay_ms(10);
write_eeprom(142, buffer[142]);
delay_ms(10);
write_eeprom(144, buffer[144]);
delay_ms(10);
write_eeprom(145, buffer[145]);
delay_ms(10);
write_eeprom(146, buffer[146]);
delay_ms(10);
write_eeprom(147, buffer[147]);
delay_ms(10);
write_eeprom(148, buffer[148]);
delay_ms(10);
write_eeprom(149, buffer[149]);
delay_ms(10);
write_eeprom(150, buffer[150]);
delay_ms(10);
write_eeprom(152, buffer[152]);
delay_ms(10);
write_eeprom(153, buffer[153]);
delay_ms(10);
write_eeprom(154, buffer[154]);
delay_ms(10);
```

```
write_eeprom(155, buffer[155]);
delay_ms(10);
write_eeprom(156, buffer[156]);
delay_ms(10);
write_eeprom(157, buffer[157]);
delay_ms(10);
write_eeprom(158 , buffer[158]);
delay_ms(10);
write_eeprom(159, buffer[159]);
delay_ms(10);
write_eeprom(160, buffer[160]);
delay_ms(10);
write_eeprom(162, buffer[162]);
delay_ms(10);
write_eeprom(163, buffer[163]);
delay_ms(10);
write_eeprom(164, buffer[164]);
delay_ms(10);
write_eeprom(165, buffer[165]);
delay_ms(10);
write_eeprom(166, buffer[166]);
delay_ms(10);
write_eeprom(167, buffer[168]);
delay_ms(10);
write_eeprom(168, buffer[168]);
delay_ms(10);
write_eeprom(170, buffer[170]);
delay_ms(10);
write_eeprom(171, buffer[171]);
delay_ms(10);
```

```
write_eeprom(172, buffer[172]);
delay_ms(10);
write_eeprom(173, buffer[173]);
delay_ms(10);
write_eeprom(174, buffer[174]);
delay_ms(10);
write_eeprom(175, buffer[175]);
delay_ms(10);
write_eeprom(176, buffer[176]);
delay_ms(10);
write_eeprom(177 , buffer[177]);
delay_ms(10);
write_eeprom(178, buffer[178]);
delay_ms(10);
output_low(pin_a1);
}
}
/////////////////////////////////////////////////////////////////main
void main()
{
ENABLE_INTERRUPTS(GLOBAL); // Enable Interrupts
ENABLE_INTERRUPTS(INT_RDA); // Enable Serial Interrupts
fprintf(HOSTPC," \n\r Automation Tour Guide System Positioning with GPS \n\r");
/////////////////////////////////////////////////////////////////read EEPROM
{
delay_ms(10);
user1[0] = read_eeprom(0);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
```

```
////////////////////
delay_ms(10);
user2[0] = read_eeprom(2);
delay_ms(10);
user2[1] = read_eeprom(3);
delay_ms(10);
user2[2] = read_eeprom(4);
delay_ms(10);
user2[3] = read_eeprom(5);
delay_ms(10);
user2[4] = read_eeprom(6);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation01 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation01);
}
/////////////////////////////////////////////////////////////////////read EEPROM
{
delay_ms(10);
user3[0] = read_eeprom(8);
delay_ms(10);
user3[1] = read_eeprom(9);
delay_ms(10);
user3[2] = read_eeprom(10);
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
```

```
delay_ms(10);
user4[0] = read_eeprom(12);
delay_ms(10);
user4[1] = read_eeprom(13);
delay_ms(10);
user4[2] = read_eeprom(14);
delay_ms(10);
user4[3] = read_eeprom(15);
delay_ms(10);
user4[4] = read_eeprom(16);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation01 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation01);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
/////////////////////////////////////////////////////////////
{
delay_ms(10);
```

```
user1[0] = read_eeprom(18);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
////////////////////
delay_ms(10);
user2[0] = read_eeprom(20);
delay_ms(10);
user2[1] = read_eeprom(21);
delay_ms(10);
user2[2] = read_eeprom(22);
delay_ms(10);
user2[3] = read_eeprom(23);
delay_ms(10);
user2[4] = read_eeprom(24);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation02 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation02);
}
//////////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(26);
delay_ms(10);
user3[1] = read_eeprom(27);
delay_ms(10);
user3[2] = read_eeprom(28);
```

```
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
delay_ms(10);
user4[0] = read_eeprom(30);
delay_ms(10);
user4[1] = read_eeprom(31);
delay_ms(10);
user4[2] = read_eeprom(32);
delay_ms(10);
user4[3] = read_eeprom(33);
delay_ms(10);
user4[4] = read_eeprom(34);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation02 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation02);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
```

```
}
///////////////////////////////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(36);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
////////////////////
delay_ms(10);
user2[0] = read_eeprom(38);
delay_ms(10);
user2[1] = read_eeprom(39);
delay_ms(10);
user2[2] = read_eeprom(40);
delay_ms(10);
user2[3] = read_eeprom(41);
delay_ms(10);
user2[4] = read_eeprom(42);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation03 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation03);
}
///////////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(44);
```

```
delay_ms(10);
user3[1] = read_eeprom(45);
delay_ms(10);
user3[2] = read_eeprom(46);
delay_ms(10);
for (k=0;k<1;k++)
tu4 - atof(user3);
////////////////////
delay_ms(10);
user4[0] = read_eeprom(48);
delay_ms(10);
user4[1] = read_eeprom(49);
delay_ms(10);
user4[2] = read_eeprom(50);
delay_ms(10);
user4[3] = read_eeprom(51);
delay_ms(10);
user4[4] = read_eeprom(52);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation03 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation03);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
```

```
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
////////////////////////////////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(54);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
/////////////////////
delay_ms(10);
user2[0] = read_eeprom(56);
delay_ms(10);
user2[1] = read_eeprom(57);
delay_ms(10);
user2[2] = read_eeprom(58);
delay_ms(10);
user2[3] = read_eeprom(59);
delay_ms(10);
user2[4] = read_eeprom(60);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation04 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation04);
}
```

```
////////////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(62);
delay_ms(10);
user3[1] = read_eeprom(63);
delay_ms(10);
user3[2] = read_eeprom(64);
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
delay_ms(10);
user4[0] = read_eeprom(66);
delay_ms(10);
user4[1] = read_eeprom(67);
delay_ms(10);
user4[2] = read_eeprom(68);
delay_ms(10);
user4[3] = read_eeprom(69);
delay_ms(10);
user4[4] = read_eeprom(70);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation04 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation04);
user1[5] = 0;
user2[5] = 0;
```

```
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
/////////////////////////////////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(72);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
////////////////////
delay_ms(10);
user2[0] = read_eeprom(74);
delay_ms(10);
user2[1] = read_eeprom(75);
delay_ms(10);
user2[2] = read_eeprom(76);
delay_ms(10);
user2[3] = read_eeprom(77);
delay_ms(10);
user2[4] = read_eeprom(78);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
```

```
tu3 = tu2*0.00001;
latstation05 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation05);
}
////////////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(80);
delay_ms(10);
user3[1] = read_eeprom(81);
delay_ms(10);
user3[2] = read_eeprom(82);
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
delay_ms(10);
user4[0] = read_eeprom(84);
delay_ms(10);
user4[1] = read_eeprom(85);
delay_ms(10);
user4[2] = read_eeprom(86);
delay_ms(10);
user4[3] = read_eeprom(87);
delay_ms(10);
user4[4] = read_eeprom(88);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
```

```
logstation05 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation05);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
/////////////////////////////////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(90);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
////////////////////
delay_ms(10);
user2[0] = read_eeprom(92);
delay_ms(10);
user2[1] = read_eeprom(93);
delay_ms(10);
user2[2] = read_eeprom(94);
delay_ms(10);
user2[3] = read_eeprom(95);
delay_ms(10);
```

```
user2[4] = read_eeprom(96);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation06 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation06);
}
/////////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(98);
delay_ms(10);
user3[1] = read_eeprom(99);
delay_ms(10);
user3[2] = read_eeprom(100);
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
delay_ms(10);
user4[0] = read_eeprom(102);
delay_ms(10);
user4[1] = read_eeprom(103);
delay_ms(10);
user4[2] = read_eeprom(104);
delay_ms(10);
user4[3] = read_eeprom(105);
delay_ms(10);
user4[4] = read_eeprom(106);
```

```
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation06 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation06);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
/////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(108);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
////////////////////
delay_ms(10);
user2[0] = read_eeprom(110);
delay_ms(10);
user2[1] = read_eeprom(111);
```

```
delay_ms(10);
user2[2] = read_eeprom(112);
delay_ms(10);
user2[3] = read_eeprom(113);
delay_ms(10);
user2[4] = read_eeprom(114);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation07 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation07);
}
///////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(116);
delay_ms(10);
user3[1] = read_eeprom(117);
delay_ms(10);
user3[2] = read_eeprom(118);
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
///////////////////
delay_ms(10);
user4[0] = read_eeprom(120);
delay_ms(10);
user4[1] = read_eeprom(121);
delay_ms(10);
```

```
user4[2] = read_eeprom(122);
delay_ms(10);
user4[3] = read_eeprom(123);
delay_ms(10);
user4[4] = read_eeprom(124);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation07 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation07);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
///////////////////////////////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(126);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
////////////////////
```

```
delay_ms(10);
user2[0] = read_eeprom(128);
delay_ms(10);
user2[1] = read_eeprom(129);
delay_ms(10);
user2[2] = read_eeprom(130);
delay_ms(10);
user2[3] = read_eeprom(131);
delay_ms(10);
user2[4] = read_eeprom(132);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation08 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation08);
}
//////////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(134);
delay_ms(10);
user3[1] = read_eeprom(135);
delay_ms(10);
user3[2] = read_eeprom(136);
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
delay_ms(10);
```

```
user4[0] = read_eeprom(138);
delay_ms(10);
user4[1] = read_eeprom(139);
delay_ms(10);
user4[2] = read_eeprom(140);
delay_ms(10);
user4[3] = read_eeprom(141);
delay_ms(10);
user4[4] = read_eeprom(142);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation08 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation08);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
//////////////////////////////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(144);
```

```
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
///////////////////
delay_ms(10);
user2[0] = read_eeprom(146);
delay_ms(10);
user2[1] = read_eeprom(147);
delay_ms(10);
user2[2] = read_eeprom(148);
delay_ms(10);
user2[3] = read_eeprom(149);
delay_ms(10);
user2[4] = read_eeprom(150);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation09 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation09);
}
//////////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(152);
delay_ms(10);
user3[1] = read_eeprom(153);
delay_ms(10);
user3[2] = read_eeprom(154);
delay_ms(10);
```

```
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
delay_ms(10);
user4[0] = read_eeprom(156);
delay_ms(10);
user4[1] = read_eeprom(157);
delay_ms(10);
user4[2] = read_eeprom(158);
delay_ms(10);
user4[3] = read_eeprom(159);
delay_ms(10);
user4[4] = read_eeprom(160);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation09 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation09);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
```

```
//////////////////////////////////
{
delay_ms(10);
user1[0] = read_eeprom(162);
delay_ms(10);
for (k=0;k<1;k++)
tu1 = atof(user1);
///////////////////
delay_ms(10);
user2[0] = read_eeprom(164);
delay_ms(10);
user2[1] = read_eeprom(165);
delay_ms(10);
user2[2] = read_eeprom(166);
delay_ms(10);
user2[3] = read_eeprom(167);
delay_ms(10);
user2[4] = read_eeprom(168);
delay_ms(10);
for (k=0;k<5;k++)
tu2 = atof(user2);
tu3 = tu2*0.00001;
latstation10 = tu1+tu3 ;
fprintf(HOSTPC,"%.5f\n\r",latstation10);
}
//////////////////////////////////////////////////////
{
delay_ms(10);
user3[0] = read_eeprom(170);
delay_ms(10);
```

```
user3[1] = read_eeprom(171);
delay_ms(10);
user3[2] = read_eeprom(172);
delay_ms(10);
for (k=0;k<1;k++)
tu4 = atof(user3);
////////////////////
delay_ms(10);
user4[0] = read_eeprom(174);
delay_ms(10);
user4[1] = read_eeprom(175);
delay_ms(10);
user4[2] = read_eeprom(176);
delay_ms(10);
user4[3] = read_eeprom(177);
delay_ms(10);
user4[4] = read_eeprom(178);
delay_ms(10);
for (k=0;k<5;k++)
tu5 = atof(user4);
tu6 = tu5*0.00001;
logstation10 = tu4+tu6 ;
fprintf(HOSTPC,"%.5f\n\r",logstation10);
user1[5] = 0;
user2[5] = 0;
user3[5] = 0;
user4[5] = 0;
tu1 = 0;
tu2 = 0;
tu3 = 0;
```

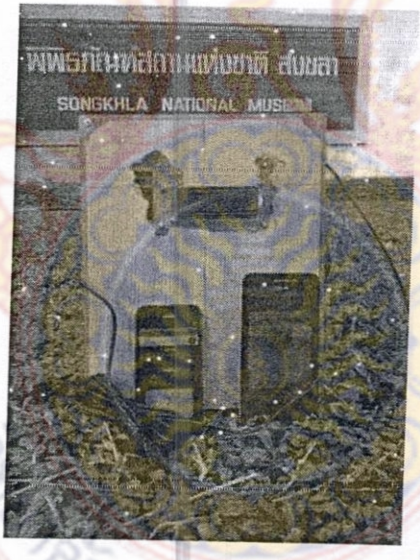
```
tu4 = 0;
tu5 = 0;
tu6 = 0;
}
/////////////////////////////////////
set_tris_a(0x01);
int1 sw;
sw = input(PIN_c0);
while(true)
{
if (sw==1)
{
Readmessage(); //gps
delay_ms(100);
lcd();
delay_ms(100);
mp();
delay_ms(100);
}
else if (sw==0)
{
writeeeprom ();
delay_ms(100);
}
}
}
```

บทที่ 4
ผลการทดลอง

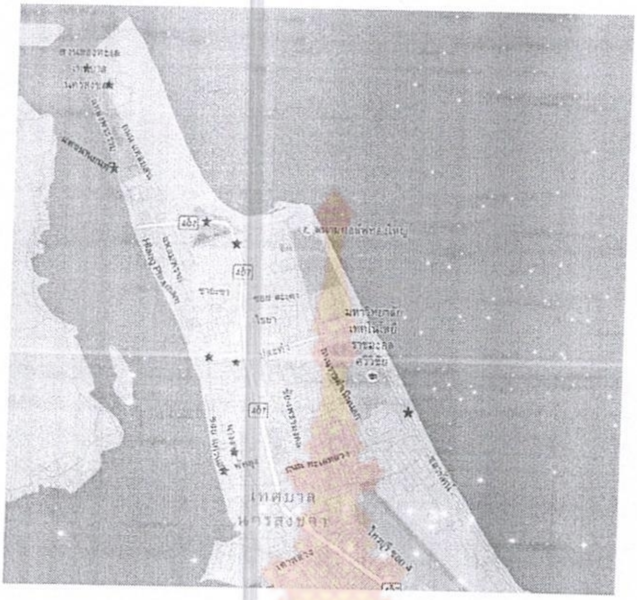
การทดลองระบบบรรยายข้อมูลสถานที่ท่องเที่ยวอัตโนมัติด้วยจีพีเอสได้แบ่งการทดลองออกเป็นส่วนๆ และการทดลองระบบรวมดังนี้

4.1 การทดสอบพิกัดจีพีเอส

การทดสอบพิกัดจีพีเอส เป็นการตรวจสอบพิกัดละติจูด ลองติจูดที่จากเครื่องรับสัญญาณจีพีเอสที่สร้างขึ้น แล้วเปรียบเทียบกับค่าจีพีเอสจากแอปพลิเคชั่นของ Google Map บนระบบปฏิบัติการแอนดรอยน์ (Android) ของสมาร์ทโฟน เพื่อดูค่าความแตกต่างของพิกัด ณ ตำแหน่งเดียวกัน โดยใช้ค่าจาก Google Map ของสมาร์ทโฟน เป็นค่าอ้างอิง ซึ่งทำการทดลองวัดค่าและเปรียบเทียบจำนวน 10 จุด กระจายในพื้นที่ เทศบาลนครสงขลา ตำบลบ่อยาง อำเภอเมือง จังหวัดสงขลา มีข้อมูลดังตารางที่ 4.1



รูปที่ 4.1 ลักษณะการวางตำแหน่งสมาร์ทโฟนและจีพีเอสที่สร้างขึ้น



รูปที่ 4.2 ตำแหน่งจุดพิกัดทดสอบในพื้นที่เทศบาลนครสงขลา

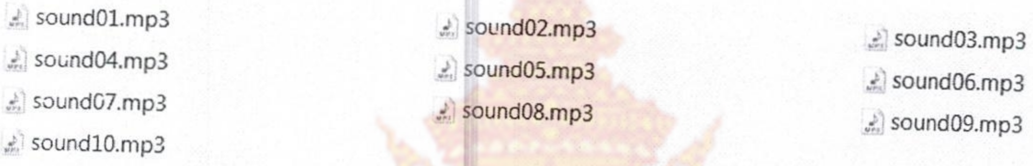
ตารางที่ 4.1 การเปรียบเทียบพิกัดละติจูด ลองติจูด กับจีพีเอส Google Map บนแอนดรอยน์

จุด	จีพีเอสสมาร์ทโฟน		จีพีเอสที่สร้างขึ้น		ความผิดพลาด (เมตร)	
พิกัดที่	ละติจูต	ลองติจูด	ละติจูด	ลองติจูด	ละติจูด	ลองติจูด
1	7.22669	100.57721	7.22662	100.57720	7.68	1.09
2	7.22470	100.57918	7.22469	100.57920	1.09	2.19
3	7.21039	100.59108	7.21041	100.59107	2.19	1.09
4	7.20208	100.59051	7.20216	100.59058	8.77	7.68
5	7.20243	100.58846	7.20238	100.58847	5.48	1.09
6	7.19296	100.59006	7.19292	100.59013	4.38	7.68
7	7.19633	100.59061	7.29640	100.59060	7.68	1.09
8	7.20954	100.59945	7.20955	100.59946	1.09	1.09
9	7.21260	100.58910	7.21261	100.58911	1.09	1.09
10	7.21756	100.58038	7.21756	100.58039	0	1.09

ผลการทดลองเปรียบเทียบพิกัดตำแหน่งละติจูด ลองติจูด ที่อ่านได้จากชุดที่สร้างขึ้น กับ สมาร์ทโฟนบนระบบปฏิบัติการแอนดรอยน์ พบว่ามีค่าความแตกต่างกัน ทั้งในส่วนของละติจูด และลองติจูด โดยเมื่อนำค่าความแตกต่างที่ได้ไปคำนวณเป็นระยะทาง แล้วพบว่าระยะคลาดเคลื่อนไม่เกิน 10 เมตร

4.2 การทดสอบเล่นไฟล์เสียง

การทดสอบเล่นไฟล์เสียงเป็นการทดสอบการเล่นไฟล์เสียงบรรยายข้อมูลของสถานที่ท่องเที่ยวแต่ละแหล่งให้ตรงกับพิกัดละติจูด ลองติจูด ของสถานที่ท่องเที่ยวนั้นๆ จะต้องทำการกำหนดสถานที่ท่องเที่ยวเรียงเป็นลำดับ เนื่องจากโมดูล MP3 จะควบคุมให้เล่นไฟล์เสียงตามลำดับ ดังนั้นการตั้งชื่อไฟล์เสียงบรรยาย ซึ่งเก็บไว้ในหน่วยความจำ SD การ์ดจะต้องตั้งชื่อไฟล์เป็นลำดับตัวเลข จึงจะไม่ทำให้เกิดความสับสนในการเรียงลำดับไฟล์ หากขึ้นต้นด้วยคำว่า Sound แล้วตามด้วยตัวเลข เช่น Sound00 ไฟล์ลำดับถัดไปต้องตั้งชื่อเป็น Sound01 จึงจะทำให้การเรียงลำดับไฟล์เป็นไปตามเลขกำกับด้านหลัง หากตั้งชื่อไฟล์ไม่ใช้ตัวเลขกำกับท้าย จะทำให้การเรียงลำดับเกิดความไม่แน่นอนของลำดับไฟล์ที่จัดเก็บในหน่วยความจำ SD การ์ดเนื่องจากค่าตัวเลขประจำตัวอักษรแต่ละตัวมีความแตกต่างกัน



รูปที่ 4.3 ไฟล์เสียงบรรยายสถานที่ท่องเที่ยว

ตารางที่ 4.2 พิกัดละติจูด ลองติจูดของสถานที่ท่องเที่ยว

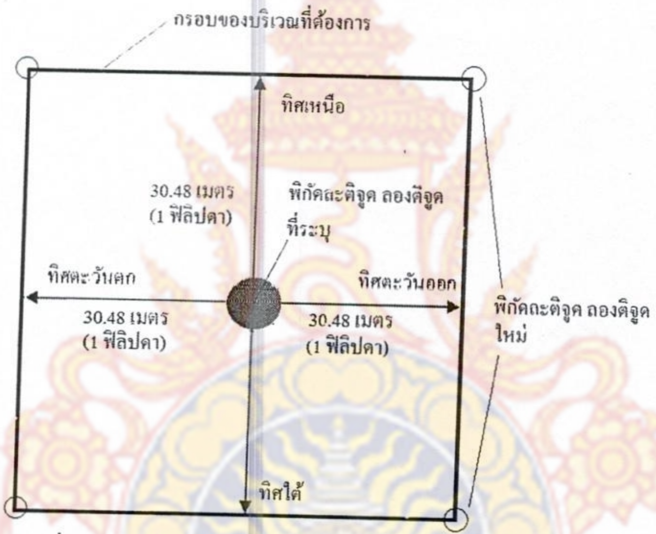
สถานที่ท่องเที่ยว จุดที่	จีพีเอสที่สร้างขึ้น		ชื่อไฟล์เสียงบรรยาย
	ละติจูด	ลองติจูด	
1	7.22662	100.57720	Sound00
2	7.22469	100.57920	Sound01
3	7.21041	100.59107	Sound3
4	7.20216	100.59058	Sound4
5	7.20238	100.58847	Sound5
6	7.19292	100.59013	Sound6
7	7.29640	100.59060	Sound7
8	7.20955	100.59946	Sound8
9	7.21261	100.58911	Sound9
10	7.21756	100.58039	Sound10

จากนั้นนำเครื่องบรรยายสถานที่ท่องเที่ยวเข้าไปยังพิกัดละติจูด ลองติจูด เพื่อทดสอบว่าไมโครคอนโทรลเลอร์ตรวจสอบพิกัดละติจูด ลองติจูด จากโมดูลจีพีเอส แล้วนำมาเปรียบเทียบกับพิกัดละติจูด ลองติจูด ของสถานที่ท่องเที่ยวที่กำหนดไว้ แล้วควบคุมให้โมดูล MP3 เล่นไฟล์เสียงบรรยายข้อมูลสถานที่ท่องเที่ยว พบว่า สามารถเล่นไฟล์เสียงได้ตามที่ได้ออกแบบโปรแกรมไว้ได้ แต่มีความยากต่อการเข้าถึงจุดพิกัดดังกล่าว เนื่องจากพิกัดดังกล่าวเป็นพิกัดจุดเล็กๆ ทำให้เสียเวลาในการ

ค้นหาเพื่อให้ เข้าถึงพิกัดละติจูด ลองติจูดดังกล่าว ดังนั้นจึงต้องขยายพิกัดละติจูด ลองติจูด ให้มีลักษณะเป็นบริเวณล้อมรอบเป็นขอบเขตของพิกัดละติจูด ลองติจูดที่กำหนดเป็นสถานที่ท่องเที่ยว

4.3 การทดสอบบริเวณขอบเขตของพิกัดสถานที่ท่องเที่ยว

จากปัญหาความยากต่อการเข้าถึงพิกัดละติจูด ลองติจูด ที่กำหนดไว้เพียงจุดเดียว ซึ่งจะทำให้โมดูล MP3 ไม่เล่นไฟล์เสียงบรรยายข้อมูลเลย หากตำแหน่งไม่ตรงกันกับพิกัดที่กำหนด ดังนั้นจึงดำเนินการขยายจุดพิกัดดังกล่าว ให้มีลักษณะเป็นพื้นที่ หรือบริเวณขอบเขตล้อมรอบพิกัดละติจูด ลองติจูดของสถานที่ท่องเที่ยวแต่ละแห่ง มีลักษณะเป็น กรอบสี่เหลี่ยมล้อมรอบจุดพิกัดเดิม โดยกำหนดให้มีรัศมีกว้างเป็นระยะ 1 ฟิลิปดา หรือ 30.48 เมตร ดังลักษณะในรูปที่ 4.4



รูปที่ 4.4 ลักษณะการกำหนดกรอบขอบเขตของพื้นที่

เช่นหากกำหนดจุดพิกัดเพื่อเริ่มเล่นไฟล์เสียงของสถานที่ท่องเที่ยวเป็นพิกัด 7.201124N,100.602E ซึ่งจะต้องดำเนินการขยายขอบเขตของพื้นที่ที่ระบบบรรยายข้อมูลสามารถเข้าถึงจากทิศทางใดๆก็ได้เพียงแต่ให้อยู่ภายในบริเวณ ดังนั้นการคำนวณในส่วนของด้านทิศตะวันตกของพิกัดละติจูดจึงต้องลบด้วย 30.48 เมตร ส่วนด้านทิศตะวันออกของพิกัดละติจูดจึงต้องบวกเพิ่มอีก 30.48 เมตร และทางด้านลองติจูดดำเนินการในลักษณะเดียวกัน คือด้านทิศเหนือของพิกัดลองติจูดทำการบวกเพิ่มอีก 30.48 เมตร ส่วนด้านทิศใต้ของพิกัดลองติจูดต้องทำการลบค่า 30.48 เมตร โดยโปรแกรมจะดำเนินการคำนวณหาพิกัดใหม่เพื่อตีกรอบล้อมรอบพิกัดเดิม ซึ่งจะได้พิกัดละติจูด ลองติจูดใหม่จำนวน 4 จุด เพื่อใช้เป็นของเขตล้อมรอบพิกัดเดิม ดังข้อมูลในตารางที่ 4.3

ตารางที่ 4.3 ขอบเขตบริเวณการขยายพื้นที่ของพิกัด

จุดพิกัดใหม่	ละติจูด	ลองติจูด
ตะวันออกเฉียงเหนือ	7.201394	100.603024
ตะวันออกเฉียงใต้	7.200854	100.603024
ตะวันตกเฉียงใต้	7.200854	100.602484
ตะวันตกเฉียงเหนือ	7.201394	100.602484

จึงทำให้ พิกัดละติจูด ลองติจูดของสถานที่ท่องเที่ยวมีลักษณะเป็นกรอบบริเวณสี่เหลี่ยม ดังแสดงใน รูปที่ 4.5

รูปที่ 4.5 พิกัดละติจูด ลองติจูดใหม่

โดยหากพิกัดตำแหน่งปัจจุบันของเครื่องบรรยายเคลื่อนที่มาอยู่ภายในระหว่างค่าละติจูดและลองติจูด ดังกล่าวแล้ว จะทำให้ระบบเริ่มต้นบรรยายเสียงข้อมูลของสถานที่ท่องเที่ยวทันที

7.200854 > ละติจูดปัจจุบัน < 7.201394

100.602484 > ลองติจูดปัจจุบัน < 100.603024

จากนั้นได้ทำการทดสอบความแม่นยำของระยะ 1 ฟิลิปดา หรือ 30.48 เมตร โดยใช้จุดพิกัด 7.201124N,100.602E เป็นจุดอ้างอิง กับการวัดไปยังพิกัดละติจูด ลองติจูดที่คำนวณได้มาใหม่ของทิศ เหนือ ใต้ ตะวันออก ตะวันตก ซึ่งได้ระยะและความคลาดเคลื่อนดังตารางที่ 4.4

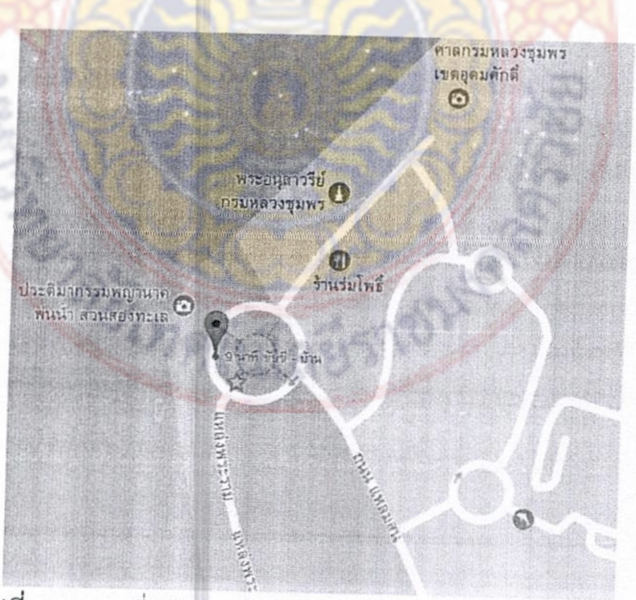
ตารางที่ 4.4 ความคลาดเคลื่อนจากระยะ 30.48 เมตร

ทิศ	ละติจูด	ลองติจูด	ระยะทาง	คลาดเคลื่อน
เหนือ	7.201394	100.602754	29.79	0.69 เมตร
ตะวันออก	7.201124	100.603024	29.69	0.79 เมตร
ใต้	7.200854	100.602754	30.12	0.36 เมตร
ตะวันตก	7.201124	100.602484	29.81	0.67 เมตร

4.4 การทดลองกับสถานที่ท่องเที่ยว

การทดลองนี้ได้นำข้อมูลของสถานที่ท่องเที่ยวในพื้นที่เทศบาลสงขลาจำนวน 10 แห่งมาแปลงเป็นเสียงบรรยายเก็บไว้ในไฟล์รูปแบบ MP3 ที่มีการตั้งชื่อเรียงลำดับไว้แล้วในหน่วยความจำ SD การ์ด จากนั้นป้อนค่าพิกัดละติจูด ลองติจูด ลงในโปรแกรมที่พัฒนาขึ้นมาบนคอมพิวเตอร์ แล้วดำเนินการดาวน์โหลดค่าพิกัดสถานที่ท่องเที่ยวไปเก็บไว้ในหน่วยความจำใช้โปรแกรมอีอีพรอม จากนั้นเปิดสวิทซ์เครื่องมาที่โหมด รัน แล้วจึงนำเครื่องบรรยายสถานที่ท่องเที่ยวติดตั้งบนรถยนต์ โดยติดตั้งสายอากาศไว้บนหลังคารถ แล้วขับรถเข้าไปพื้นที่บริเวณของสถานที่ท่องเที่ยวแต่ละแห่ง ดังนี้

1) ประติมากรรมพญานาคพ่นน้ำ ซึ่งได้กำหนดพิกัดละติจูด ลองติจูด ให้ระบบเริ่มบรรยายคือ พิกัด โดยมีข้อมูลการบรรยายดังนี้

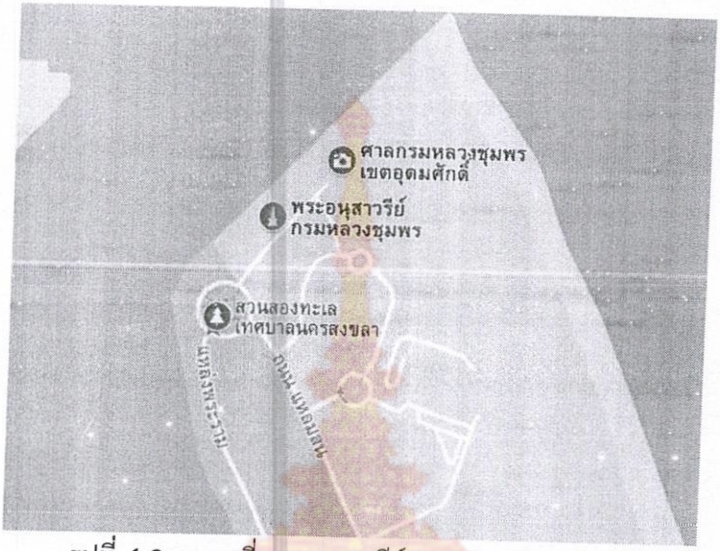


รูปที่ 4.6 แผนที่พิกัดตำแหน่งของประติมากรรมพญานาคพ่นน้ำ

รูปที่ 4.7 ประติมากรรมพญานาคพ่นน้ำ สงขลา ณ พิกัด 7.226756,100.577205

ประติมากรรมพญานาคพ่นน้ำ ถือเป็นสัญลักษณ์หนึ่งของจังหวัดสงขลา เป็นโครงการที่เทศบาลนครสงขลา สรรค์สร้างขึ้นเพื่อ ปรับปรุงภูมิทัศน์บริเวณ ชายหาดสมิหลาให้เป็นสถานที่พักผ่อนของ นักท่องเที่ยวและชาวสงขลา โดยนำเอา คติความเชื่อเกี่ยวกับ พญานาคที่เชื่อว่า "พญานาค" เป็นสัญลักษณ์ของการ กำเนิดน้ำและความอุดมสมบูรณ์ ชาวใต้จึงนับถือพญานาคเป็นสิ่งศักดิ์สิทธิ์ และกราบไหว้ขอพร เพื่อเป็นสิริมงคลแก่ชีวิต นับ เป็นแหล่งการเรียนรู้ทาง วัฒนธรรมแห่งใหม่ที่เกิดขึ้นของ ชาวสงขลา ประติมากรรมพญานาคพ่นน้ำ เป็นสัญลักษณ์หนึ่งของ จังหวัดสงขลา มีลักษณะแบบลอยตัว สามารถมองเห็นได้รอบด้านเนื้อ วัตถุเป็นโลหะทองเหลืองรมสนิมเขียว ออกแบบ โดย อาจารย์มนตรี สังข์มุสิกานนท์ (รองอธิการบดีมหาวิทยาลัยทักษิณคนปัจจุบัน) มีการสร้างขึ้นเป็น 3 ส่วน คือ ส่วนที่หนึ่ง หัวพญานาค ตั้งอยู่บริเวณสวนสองทะเล ปลายแหลมสนอ่อน มีขนาดเส้นผ่าศูนย์กลาง ของลำตัว 1.20 เมตร ความสูงจากฐานลำตัวจนถึงปลายยอดสุด ประมาณ 9 เมตร พ่นน้ำลงสู่ปากอ่าวทะเลสาบสงขลา ส่วนที่สอง สะดือพญานาค ตั้งอยู่บริเวณลานชมดาว สนามสระบัว แหลมสมิหลา ขนาดเส้นผ่าศูนย์กลางของลำตัว 1.20 เมตร ความยาว 5.00 เมตร ความสูง 2.50 เมตร ลักษณะลำตัวโค้งครึ่งวงกลม เพื่อให้นักท่องเที่ยวได้ลอด ใต้สะดือพญานาคให้เกิดความเป็นสิริมงคล แก่ตนเอง ส่วนที่สาม หางพญานาค ตั้งอยู่บริเวณชายหาดสมิหลา ริมถนนสะเดา (หลังสนามกอล์ฟ) ขนาดเส้นผ่าศูนย์กลาง 1.20 เมตร ความยาว 4.00 เมตร ความสูง 4.50 เมตร ปัจจุบันประติมากรรมพญานาคพ่นน้ำได้รับความนิยมจากนักท่องเที่ยวทั้งชาวไทย และชาวต่างประเทศที่เข้ามาเยี่ยมเยือน จังหวัดสงขลา

2) พระอนุสาวรีย์กรมหลวงชุมพร



รูปที่ 4.8 แผนที่พระอนุสาวรีย์กรมหลวงชุมพร

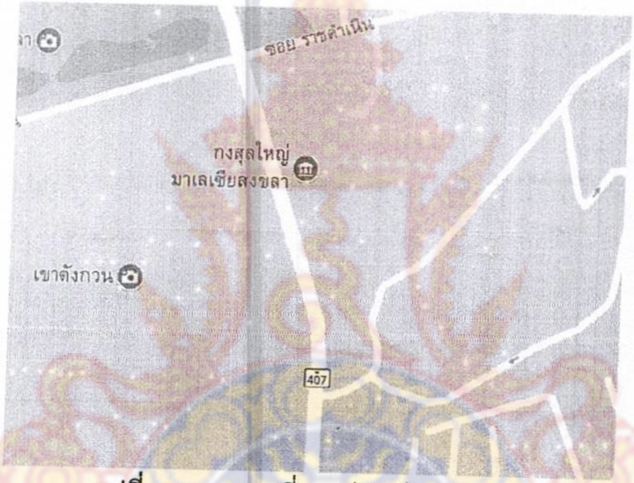
รูปที่ 4.9 พระอนุสาวรีย์กรมหลวงชุมพร ณ พิกัด 7.228185,100.578179

พลเรือเอก พระเจ้าบรมวงศ์เธอ พระองค์เจ้าอาภากรเกียรติวงศ์ กรมหลวงชุมพรเขตอุดมศักดิ์ พระราชโอรสในพระบาทสมเด็จพระจุลจอมเกล้าเจ้าอยู่หัวและเจ้าจอมมารดาโหมด พระนามเดิม พระองค์เจ้าอาภากรเกียรติวงศ์ ประสูติวันอาทิตย์ที่ 19 ธันวาคม พุทธศักราช 2423 สิ้นพระชนม์ วันเสาร์ที่ 19 พฤษภาคม พุทธศักราช 2466 ทรงศึกษาสำเร็จวิชาทหารจากโรงเรียนนายเรือ ประเทศ

อังกฤษ ทรงเข้ารับราชการในกระทรวงทหารเรือในรัชกาลที่ 5 ได้วางรากฐานปรับปรุงราชนาวีไทย ให้มีสมรรถภาพทัดเทียมอารยประเทศ ทรงดำรงตำแหน่งเสนาบดีกระทรวงทหารเรือในรัชกาลที่ 6 กองทัพเรือไทยรุ่งเรืองเป็นปึกแผ่นสืบมาถึงปัจจุบันด้วยพระอัจฉริยะปรีชาชาญ จึงน้อมถวายพระสมัญญานามว่า องค์บิดาของทหารเรือไทย กองทัพเรือร่วมกับชาวจังหวัดสงขลาและกลุ่มอาสาป้องกันชาติในทะเลจังหวัด สงขลาสำนึกในพระกรุณาจึงพร้อมใจกันสร้างพระอนุสาวรีย์นี้เพื่อเป็นที่เคารพ สักการะสืบไป

http://songkhlask.blogspot.com/2011/05/blog-post_18.html

3) เขาตังกวน

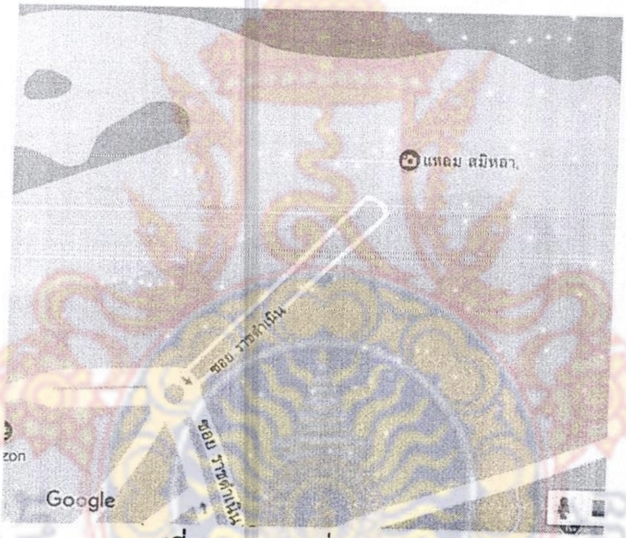


รูปที่ 4.10 แผนที่สถานีลิฟท์เขาตังกวน

รูปที่ 4.11 ลิฟท์ขึ้นเขาตังกวน ณ พิกัด 7.210832,100.590919

เขาตังกวน เป็นสถานที่ท่องเที่ยวที่สำคัญแห่งหนึ่งในอ.เมือง จ.สงขลา เป็นเนินเขาสูง จากระดับน้ำทะเลประมาณ 2,000 ฟุต จากยอดเขาตังกวนนี้สามารถมองเห็นทิวทัศน์ของเมืองสงขลาได้โดยรอบ บนยอดเขาตังกวนเป็นที่ประดิษฐานเจดีย์พระธาตุคู่เมือง สงขลาซึ่งสร้างในสมัยอาณาจักรนครศรีธรรมราช เป็นศิลปะสมัยทวาราวดี(อยู่บนยอดเขาสูงจากระดับน้ำทะเลประมาณ 2,000 ฟุต) โดยพระบาทสมเด็จพระจอมเกล้าเจ้าอยู่หัว (ร.4) ได้พระราชทานเงินหลวงให้เป็นทุนในการบูรณะปฏิสังขรณ์ และในเดือนธันวาคม พ.ศ. 2539 พระบาทสมเด็จพระเจ้าอยู่หัวรัชกาลปัจจุบัน (ร.9) ได้ทรงพระราชทานพระบรมสารีริกธาตุให้มาบรรจุในองค์พระเจดีย์ในทุกๆ ปีในเดือนตุลาคม จะมีงานพิธีห่มผ้าองค์พระเจดีย์ และประเพณีตักบาตรเทโวและลากพระของสงขลา
http://www.paiduaykan.com/province/south/songkhla/tangkouan.html

4) แหลมสมิหลา



รูปที่ 4.12 แผนที่ หาดสมิหลา

รูปที่ 4.13 หาดสมิหลา ณ พิกัด 7.213943,100.594788

ถ้ามาถึงสงขลาแล้วไม่ได้เยือนหาดสมิหลาย่อมถือว่ามาไม่ถึง เพราะที่นี่คือไอคอนของเมืองสงขลาที่ทุกคนต้องมาแวะถ่ายภาพไว้เป็นที่ระลึก หาดสมิหลาเป็นชายหาดที่มีโขดหินขนาดย่อมยื่นลงทะเล ทรายขาวละเอียดมากที่เรียกว่า "ทรายแก้ว" ร่มรื่นด้วยป่าสน และจากหาดสมิหลายังสามารถมองเห็นทิวทัศน์อันงดงามของเกาะหนูเกาะแมว จนมีคำกล่าวว่าใครมาเยือนสงขลาแล้วไม่มาเยือนสมิหลาก็เหมือนมาไม่ถึงสงขลา อีกทั้งบริเวณหาดยังมีสัญลักษณ์ที่มีชื่อเสียง นั่นก็คือ รูปปั้นนางเงือกทองที่ทุกคนต้องแวะมาถ่ายภาพเก็บไว้ บรรยากาศโดยรอบของชายหาดเต็มไปด้วยความเงียบสงบ มีร้านอาหาร รีสอร์ต และร้านขายของที่ระลึกมากมาย รวมทั้งมีกิจกรรมทางน้ำเติมความสนุกให้นักท่องเที่ยว เช่น บานาน่าโบ๊ต เจ็ตสกี แล่นเรือใบ และชายหาดแห่งนี้ยังสามารถลงเล่นน้ำทะเลได้เนื่องจากเป็นชายหาดที่ไม่ลาดชัน และมียามรักษาการณ์จากเทศบาลเมืองสงขลาคอยดูแลความปลอดภัยตลอดทั้งวัน

5) ประตูเมืองเก่าสงขลา

รูปที่ 4.14 ประตูเมืองเก่า ณ พิกัด 7.192857 ,100.590127

จังหวัดสงขลามีประวัติศาสตร์ความเป็นมาอย่างต่อเนื่อง มีหลักฐานปรากฏในการสร้างเมือง สมัยเจ้าพระยาคีรี (เถี้ยนเส้ง ณ สงขลา) ได้เริ่มสร้างป้อม กำแพงเมืองยาว 1200 เมตร และประตูเมือง สิบประตู ตั้งแต่ พ.ศ. 2379 เป็นประตูเมืองที่สวยงาม เป็นซุ้มใหญ่โดยรอบ 10 ประตู กว้างประมาณ 6 ศอก สูง 3 วา ซึ่งปัจจุบันนี้เทศบาลนครสงขลา ได้จำลองไว้บริเวณทางเข้าเมืองสงขลา และย่านเมืองเก่า ถนนนครนอก ถนนนครใน เมืองสงขลา เพื่อเป็นสัญลักษณ์เมืองสงขลา จากเรื่องราวดังกล่าวช่างฝีมือจึงได้จัดทำประตูเมืองสงขลาจำลอง เป็นของที่ระลึกสำหรับผู้มาเยือนสงขลา

บทที่ 5
สรุป

ระบบบรรยายข้อมูล ความรู้เรื่องสถานที่ท่องเที่ยวตามพิกัด จีพีเอสที่กำหนดไว้ในหน่วยความจำ สามารถเริ่มเล่นไฟล์เสียงอย่างอัตโนมัติเมื่อนำเครื่องที่พัฒนาขึ้นมาเข้าภายในกรอบบริเวณของพื้นที่ที่คำนวณไว้ด้วยโปรแกรม และไมโครคอนโทรลเลอร์สามารถควบคุมให้เล่นไฟล์เสียงบรรยายจนกระทั่งจบการบรรยายข้อมูลของสถานที่นั้นๆ โดยจะไม่เล่นซ้ำแม้จะยังอยู่ภายในกรอบบริเวณ จนกว่าจะออกไปจากพื้นที่ ซึ่งหากนำเครื่องบรรยายเข้าสู่บริเวณพื้นที่ของพิกัดจีพีเอสของสถานที่ท่องเที่ยวใดระบบก็จะ เริ่มบรรยายข้อมูลของสถานที่ท่องเที่ยวนั้นๆได้อย่างถูกต้อง ซึ่งสามารถนำไปสร้างเป็นระบบบรรยายสถานที่ท่องเที่ยว บนรถนำเที่ยว หรือสามารถนำไปประยุกต์ใช้กับการบอกตำแหน่งสถานี หรือป้ายจอดของรถบัส หรือ รถไฟ ได้อีกด้วย

ภาคผนวก

The 9th Conference of Electrical Engineering Network of Rajamangala University of Technology

งานประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 9

EENET2017

"การพัฒนานวัตกรรมเพื่ออุตสาหกรรมและการเกษตรอย่างยั่งยืน"

Sustainable Development of Innovation for Industry and Agriculture

2-4 May 2017, K.P. Grand Hotel Chanthaburi

Conference Topics

- Electrical Power (PW)
- Power Electronics (PE)
- Energy and Energy Saving (ES)
- Control Systems and Instrumentation (CT)
- Computer and Information Technology (CP)
- Electric Communication (CM)
- Electronics (EL)
- Digital Signal Processing (DS)
- Innovation and Invention (IN)
- General Electrical Engineering (GN)

BETA ENERGY SOLUTION COMPANY LIMITED
AI SOLUTION
UNITIS CO.,LTD.
บริษัท นีโอ ไดแด็กติก จำกัด NEO DIDACTIC CO.,LTD.
phelps dodge International Corp
a General Cable company
PTS Combination
ThaiEasyElec enable your design Education
PRIMUS COMPANY LIMITED
SAT

ระบบบรรยายข้อมูลสถานที่ท่องเที่ยวอัตโนมัติด้วยพิกัดจีพีเอส

Automatic tourist Guide System by GPS Coordinates

สัญญา ผาสุข ศุภชัย อรุณพันธ์ จินดา สามัคคี

[1]สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

ถนนราชดำเนินนอก ตำบลบ่อยาง อำเภอเมือง จังหวัดสงขลา โทรศัพท์ 07-4312167 E-mail: sunya.p@rmutsv.ac.th

บทคัดย่อ

บทความนี้นำเสนอการประยุกต์ใช้ตำแหน่งจีพีเอสเพื่อบรรยายข้อมูลของสถานที่ท่องเที่ยวโดยอัตโนมัติตามพิกัดตำแหน่งละติจูด ลองติจูด โดยระบบจะรับพิกัดละติจูด ลองติจูดจากโมดูลจีพีเอสมาเปรียบเทียบกับพิกัดของสถานที่ท่องเที่ยวซึ่งบรรจุไว้ในหน่วยความจำอีอีพรอมของไมโครคอนโทรลเลอร์ หากพิกัดตำแหน่งอยู่ในบริเวณที่กำหนดไว้ ซึ่งมีรัศมี 30.48 เมตรของพิกัดที่ระบุ ไมโครคอนโทรลเลอร์จะสั่งให้ เล่นไฟล์เสียง MP3 ซึ่งบรรจุไว้ใน SD การ์ด บรรยายข้อมูลของสถานที่ท่องเที่ยวนั้นๆ อย่างอัตโนมัติ ได้อย่างถูกต้อง

Abstract

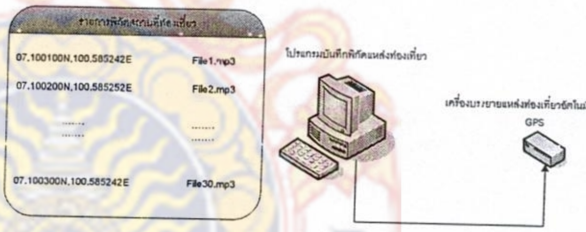
This article presents the application of the GPS data to describe the sights, automatically coordinates latitude longitude system to latitude. Longitude from a GPS module. Compared to a range of attractions which are contained in the EEPROM memory of the microcontroller. If the position is in the defined area, which has a 30.48-meter radius of the specified limits. Microcontroller instructs Play MP3 audio files on the SD card that contains data describing the sights that. automatically Correctly

Keywords: Auto Guide GPS

1. บทนำ

ประเทศไทยมีสถานที่ท่องเที่ยวต่างๆ เป็นจำนวนมาก ซึ่งการดึงดูดนักท่องเที่ยวให้เดินทางไปท่องเที่ยวในสถานที่ต่างๆ จะทำให้เศรษฐกิจขยายตัว มีความรุ่งเรืองสามารถดึงเม็ดเงินเข้าในประเทศได้เป็นจำนวนมาก แต่ประเทศไทยยังขาดจำนวนมัคคุเทศ หรือที่เรียกกันติดปากกว่าไกด์ (Guide) ทำหน้าที่แนะนำ บรรยายให้ความรู้ ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวในแต่ละแหล่ง ทำให้นักท่องเที่ยวขาดข้อมูล ความรู้ในการท่องเที่ยวตามสถานที่ต่างๆ บางพื้นที่บนรถนำเที่ยวอาจจะใช้การบรรยายในรูปแบบของมัลติมิเดียร์จากเครื่องเล่น เช่นจุดท่องเที่ยวในสวนสัตว์ หรือสวนสนุกต่างๆ ซึ่งต้องอาศัยผู้ควบคุมเครื่องเล่นคอยกดปุ่ม ไม่เป็นไปโดยอัตโนมัติ และไม่ทันสมัย ดังนั้นหากมีการพัฒนารูปแบบของเครื่องบรรยายข้อมูลสถานที่ท่องเที่ยวให้เป็นรูปแบบอัตโนมัติ หากเมื่อเดินทางมาถึงสถานที่ต่างๆ จะทำให้เกิดความสะดวก และดูทันสมัยต่อการท่องเที่ยวมากยิ่งขึ้น

การบรรยายให้ข้อมูลและความรู้เกี่ยวกับแหล่งท่องเที่ยวจะเริ่มบรรยายโดยอัตโนมัติเมื่อรถนำเที่ยวได้เดินทางมาถึงพื้นที่ของสถานที่ท่องเที่ยว ซึ่งจะใช้พิกัดของจีพีเอสกำหนดให้ครอบคลุมเป็นบริเวณพื้นที่ของแหล่งท่องเที่ยวพื้นที่นั้นๆ โดยสามารถเข้าถึงบริเวณจุดท่องเที่ยวได้ทุกทิศทาง ระบบก็จะเริ่มการบรรยายอย่างอัตโนมัติ ด้วยเสียงบรรยายที่จัดเก็บไว้ในหน่วยความจำ SD การ์ด โดยการบันทึกเก็บพิกัดจีพีเอส และเสียงบรรยายของแหล่งท่องเที่ยวแต่ละจุด ด้วยการป้อนข้อมูลลงในโปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นมาด้วยวิสชวลเบสิค แล้วจึงถ่ายโอนไปยังหน่วยความจำอีอีพรอมและ SD การ์ดดังลักษณะในรูปที่ 1 เพื่อนำไปใช้กับเครื่องบรรยายสถานที่ท่องเที่ยวที่ได้สร้างขึ้นมา



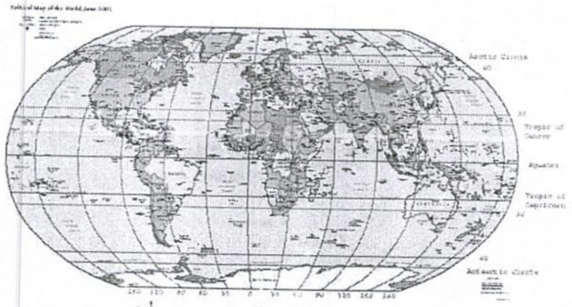
รูปที่ 1 การถ่ายโอนพิกัดและเสียงบรรยาย

2. การตรวจสอบและระบุตำแหน่งจีพีเอส

2.1 พิกัดจีพีเอส

พิกัดจีพีเอส คือพิกัดตำแหน่งละติจูด ลองติจูดบนผิวโลก โดยทุกๆพื้นที่ของผิวโลกนี้สามารถระบุตำแหน่งได้ด้วยพิกัดละติจูด ลองติจูด ซึ่งค่าละติจูดคือเส้นที่ใช้แบ่งจากเส้นศูนย์สูตรไปยังขั้วโลกเหนือ (North) 90 เส้น(องศา) และไปทางขั้วโลกใต้ (South) อีก 90 เส้น ส่วนค่าลองติจูดเป็นเส้นแบ่งจากเส้นเมริเดียน(Meridian) ซึ่งเป็นจุดเส้นที่ 0 ไปทางทิศตะวันออก (East) 180 เส้น และแบ่งไปทางทิศตะวันตก (West) อีก 180 เส้น ดังแสดงในรูปที่ 2 การระบุค่าตำแหน่งละติจูลองติจูด สามารถระบุค่าเป็นพิกัดภูมิศาสตร์ (DMS:Degree Minute Second) โดยระบุเป็น องศา ลิปดา ฟิลิปดา และสามารถระบุเป็นเลขทศนิยม

(DD:Decimal Degree) เช่น ประติมากรรมพญานาคพ่นน้ำของสวนสองทะเล จังหวัดสงขลา ตั้งอยู่ที่พิกัด 7 องศาเหนือ 13 ลิปดา 37 ฟิลิปดา 100 องศาตะวันออก 34 ลิปดา 36 ฟิลิปดา (7°13'37"N 100°34'36"E) ซึ่งหมายความว่าพิกัดดังกล่าวอยู่เหนือขึ้นมาจากเส้นศูนย์สูตรอยู่ที่ 7 องศา และมาทางทิศตะวันออกของเส้นเมริเดียนที่ 100 องศา



รูปที่ 2 การแบ่งเส้นละติจูดและลองติจูด

การแปลงหน่วยพิกัดภูมิศาสตร์เป็นระบบพิกัดทศนิยมสามารถใช้สมการดังนี้

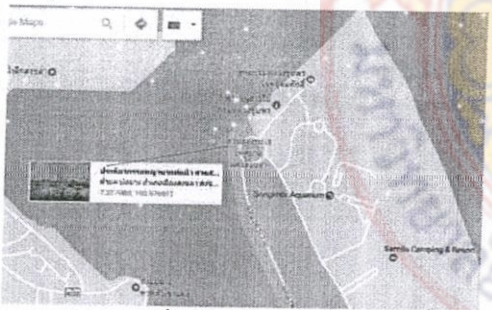
$$\mathrm{DD} = Degree + \frac{\mathrm{arcminute} \times 60 + \mathrm{arcsecond}}{3,600} \tag{1}$$

1 องศา(degree) เท่ากับ 60 ลิปดา

1 ลิปดา (arcminute) เท่ากับ 60 ฟิลิปดา

1 ฟิลิปดา(arcsecond) เท่ากับระยะทาง 30.48 เมตร

จากพิกัดของประติมากรรมพญานาคพ่นน้ำจึงสามารถคำนวณหาค่าพิกัดแบบระบบทศนิยมได้เป็น 7.227088N,100.576911E เมื่อนำพิกัดแบบทศนิยมไปป้อนค่าในโปรแกรมแผนที่ google จะได้ที่ตั้งดังแสดงในรูปที่ 3



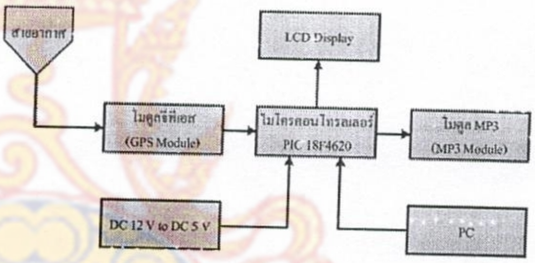
รูปที่ 3 พิกัดตำแหน่งสถานที่ท่องเที่ยว

2.2 การระบุพิกัด

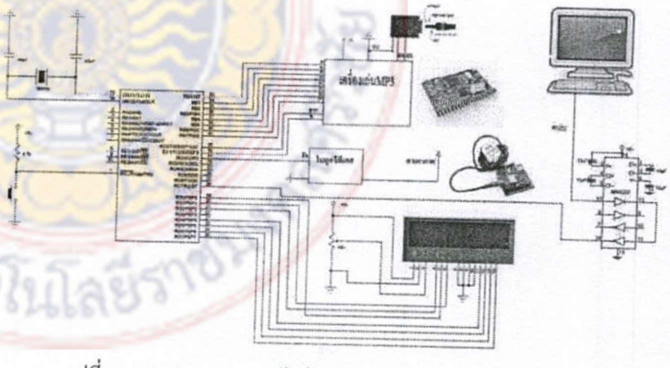
การระบุพิกัดตำแหน่งละจูด ลองติจูดสามารถใช้ข้อมูลจากระบบจีพีเอส (Global Positioning System) ซึ่งเป็นระบบที่สามารถระบุตำแหน่งบนโลกด้วยกลุ่มดาวเทียมจำนวน 24 ดวงโคจรรอบโลกระดับความสูงประมาณ 20,000 กิโลเมตร ดาวเทียมของระบบจีพีเอสแต่ละดวงจะโคจรรอบโลกทุกๆ 12 ชั่วโมง ทำให้ทุกๆพื้นที่บนโลกมีดาวเทียมอยู่เหนือท้องฟ้าอย่างน้อย 4 ดวงตลอดเวลา จึงสามารถระบุพิกัดตำแหน่งบนโลกได้ทุกๆจุด เครื่องรับสัญญาณจีพีเอส สามารถรู้พิกัดตำแหน่งของตนเองได้ตลอดเวลาด้วยวิธีการนำเอาเวลาที่ได้รับสัญญาณจากดาวเทียมแต่ละดวงซึ่งใช้ระยะเวลาเดินทางมาถึงไม่เท่ากันมาคำนวณหาระยะทางเพื่อให้ได้ค่าพิกัดตำแหน่งละติจูด และลองติจูดและข้อมูลอื่นๆ ได้อีกด้วยเช่นเวลาแท้จริง ความเร็วและความสูง เป็นต้น

3. การออกแบบระบบและทดลอง

การออกแบบระบบบรรยายให้ข้อมูล และความรู้เกี่ยวกับแหล่งท่องเที่ยวจะบรรยายโดยอัตโนมัติเมื่อรถนำเที่ยวได้เดินทางมาถึงพื้นที่ของสถานที่ท่องเที่ยวด้วยการตรวจสอบพิกัดตำแหน่งละติจูด ลองติจูดจากเครื่องรับสัญญาณจีพีเอส แล้วสั่งเล่นเสียงบรรยายข้อมูลรายละเอียดของสถานที่ท่องเที่ยวจากหน่วยความจำในรูปแบบ MP3 และขยายสัญญาณออกสู่เครื่องขยายเสียงต่อไป มีระบบดังในรูปที่ 4 และวงจรในรูปที่ 5



รูปที่ 4 ไดอะแกรมการทำงานของระบบ



รูปที่ 5 วงจรระบบควบคุมด้วยไมโครคอนโทรลเลอร์

3.1 ระบบตรวจสอบพิกัดตำแหน่งจีพีเอส

การรับสัญญาณจีพีเอสได้ใช้โมดูลจีพีเอส ทำหน้าที่รับสัญญาณจากดาวเทียม แล้วส่งออก (Tx) มายังขา C7 ของ

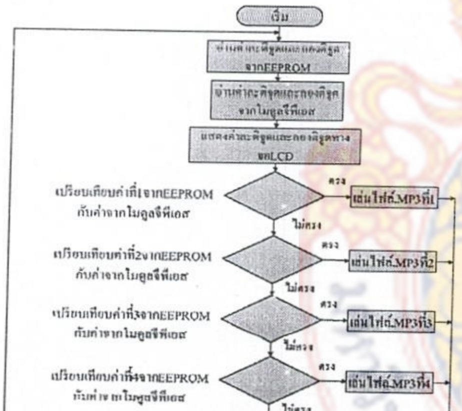
IN

ไมโครคอนโทรลเลอร์ในลักษณะการสื่อสารแบบอนุกรมด้วยอัตราบอร์ต 9,600 บิตต่อวินาที ซึ่งโมดูลจะมีการส่งสัญญาณดังกล่าวให้แก่ไมโครคอนโทรลเลอร์ทุกๆ วินาที เพื่อให้ไมโครคอนโทรลเลอร์ได้แปลงเป็นพิกัดจีพีเอส(ค่าละติจูด ลองติจูด)แล้วนำไปเปรียบเทียบกับตำแหน่งของสถานที่ท่องเที่ยวที่เก็บอยู่ในหน่วยความจำอีอีพรอมต่อไป

3.2 ระบบเล่นเสียง

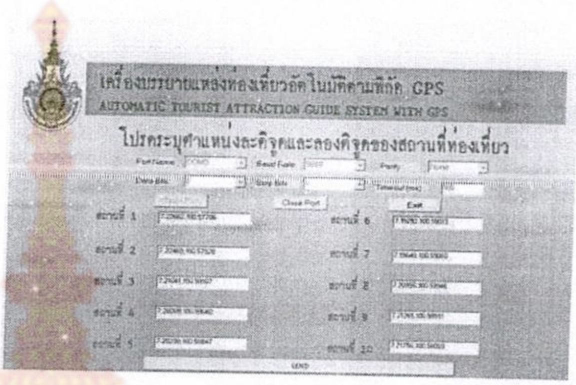
การเล่นเสียงได้ใช้ MP3 Embedded โมดูลของ ThaiEasyelec มีอัตราสุ่ม 8 kHz – 48 kHz ทำหน้าที่เล่นไฟล์เสียงบรรยายของสถานที่ท่องเที่ยวต่างๆ ซึ่งไฟล์ถูกบรรจุไว้ใน SD การ์ด โดยจะถูกควบคุมการเล่นแบบพอร์ตขนานด้วยขา B0-B7 ของไมโครคอนโทรลเลอร์ และขา C1 เพื่อตรวจสอบสถานะ สัญญาณเสียงที่ได้จากโมดูลดังกล่าวจะถูกส่งเข้าไปยังเครื่องขยายเสียงต่อไป

การทำงานของระบบนั้นจะรับสัญญาณพิกัดตำแหน่งจีพีเอสจากดาวเทียมมาแสดงผลตำแหน่งปัจจุบันบนหน้าจอ LCD และเปรียบเทียบ หรือตรวจสอบกับพิกัดในหน่วยความจำอีอีพรอม หากพิกัดไม่ตรงกัน ก็จะวนตรวจสอบต่อไป แต่หากตรงกับตำแหน่งของบริเวณที่ระบุของสถานที่ท่องเที่ยว ก็จะควบคุมให้โมดูล MP3 เล่นไฟล์เสียงบรรยายของสถานที่ท่องเที่ยวนั้นอย่างอัตโนมัติดังการทำงานตามโฟลวชาร์ทในรูปที่ 6



รูปที่ 6 ขั้นตอนการทำงานของระบบ

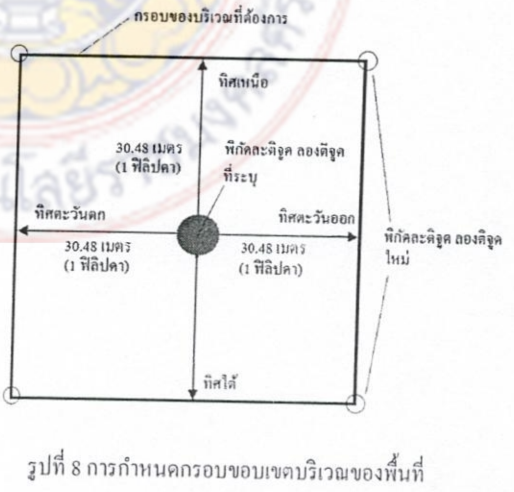
การบันทึกตำแหน่งของสถานที่ท่องเที่ยวจะใช้โปรแกรมที่พัฒนาขึ้นด้วยโปรแกรม Visual Basic รับค่าพิกัดละติจูด ลองติจูด แล้วบันทึกผ่านทางการสื่อสารอนุกรมไปยังขาC7 ของไมโครคอนโทรลเลอร์เพื่อนำไปเก็บไว้ยังหน่วยความจำอีอีพรอม



รูปที่ 7 โปรแกรมรับค่าพิกัดจีพีเอสของสถานที่ท่องเที่ยว

3.3 การระบุบริเวณ

การระบุบริเวณของสถานที่ท่องเที่ยวแต่ละที่นั้น จะต้องออกแบบให้มีลักษณะบริเวณที่เป็นกรอบหรือขอบเขต เนื่องจากหากกำหนดเป็นจุดพิกัดใดพิกัดหนึ่ง จะทำให้แคบเกินไป ยากต่อการเข้าถึง ซึ่งระบบจะไม่เล่นเสียงหากไม่ได้อยู่ในตำแหน่งของพิกัดนั้นๆ เพราะขณะบันทึกพิกัดจะระบุเพียงจุดเดียว จึงต้องนำจุดพิกัดดังกล่าวไปคำนวณหาค่าละติจูด ลองติจูดใหม่ 4 จุด (มุม) เพื่อสร้างกรอบหรือขอบเขตของบริเวณล้อมจุดพิกัดดังกล่าว โดยมีรัศมีห่างออกไปจากจุดประมาณ 30.48 เมตรหรือเท่ากับ 1 ฟิลิปดา เมื่อเครื่องรับสัญญาณจีพีเอสเข้ามาอยู่ภายในกรอบของบริเวณดังกล่าว ไม่ว่าจะเข้าถึงจากทิศทางใด ก็สามารถเล่นเสียงบรรยายของสถานที่ท่องเที่ยวนั้นๆได้ โดยไม่จำเป็นต้องเข้าถึงเฉพาะจุดพิกัดที่ระบุว่าเป็นสถานที่ท่องเที่ยว



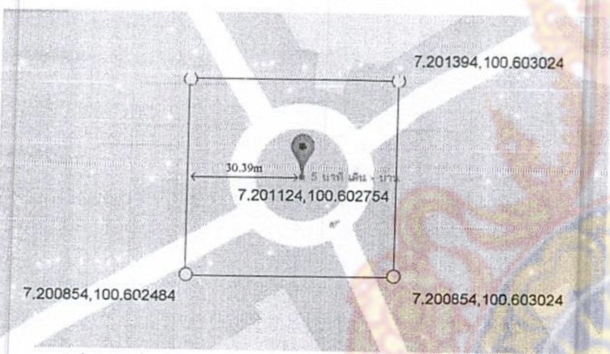
รูปที่ 8 การกำหนดกรอบขอบเขตบริเวณของพื้นที่

4. การทดสอบ

การทดสอบเพื่อหากรอบบริเวณพื้นที่ของพิกัดจีพีเอสที่ระบุเป็นสถานที่ท่องเที่ยวนั้นด้วยการคำนวณหาค่าละติจูด ลองติจูดในระบบทศนิยมขึ้นมาใหม่ ซึ่งจะกำหนดให้มีขอบเขตมีรัศมี 1 ฟิลิปดาหรือ 30.48 เมตร หากแปลงให้เป็นระบบทศนิยมจะได้เป็น 0.00027 เพื่อนำไปใช้เป็นค่าสำหรับเพิ่มและลดค่าจากพิกัดที่กำหนดไว้เพื่อให้ได้เป็นบริเวณพื้นที่ครอบคลุมจุดพิกัด เช่น กำหนดจุดสถานที่ท่องเที่ยวเป็นพิกัด 7.201124N,100.602754E จะสามารถหาขอบเขตของพื้นที่เป็นค่าละติจูด ลองติจูดใหม่ที่ครอบคลุมพิกัดเดิมดังในตารางที่ 1

ตารางที่ 1 พิกัดขอบเขตของพื้นที่บริเวณของพิกัดที่ระบุ

มุมกรอบ	ละติจูด	ลองติจูด
ตะวันออกเฉียงเหนือ	7.201394	100.603024
ตะวันออกเฉียงใต้	7.200854	100.603024
ตะวันตกเฉียงใต้	7.200854	100.602484
ตะวันตกเฉียงเหนือ	7.201394	100.602484



รูปที่ 7 พิกัดละติจูด ลองติจูด ของกรอบพื้นรอบจุดพิกัด

การทดสอบระยะทางกับตำแหน่งพิกัดจีพีเอสพบว่ามีความคลาดเคลื่อนระหว่างระยะทางกับพิกัดละติจูด ลองติจูด บ้างเล็กน้อย ซึ่งวัดระยะจากพิกัดละติจูด ลองติจูด 7.201124,100.602754 ออกไปยังทิศต่างๆในแนวเส้นละติจูด ลองติจูด เพื่อนำมาเปรียบเทียบกับระยะทาง 1 องศาฟิลิปดาหรือระยะทาง 30.48 เมตร ผลดังตารางที่ 2 ซึ่งไม่ได้เป็นปัญหาต่อการใช้งาน เนื่องจากลักษณะงานต้องการเพียงแค่ขอบเขตของพื้นที่ ที่สามารถรองรับการเข้าถึงบริเวณพื้นที่ครอบคลุมพิกัดโดยสะดวก

ตารางที่ 2 เปรียบเทียบระยะทางที่วัดได้จากพิกัดจีพีเอส

ทิศ	ละติจูด	ลองติจูด	ระยะทาง	คลาดเคลื่อน
เหนือ	7.201394	100.602754	29.79	0.69 เมตร
ตะวันออก	7.201124	100.603024	29.69	0.79 เมตร
ใต้	7.200854	100.602754	30.12	0.36 เมตร
ตะวันตก	7.201124	100.602484	29.81	0.67 เมตร

5. สรุปผล

ระบบบรรยายข้อมูล ความรู้เรื่องสถานที่ท่องเที่ยวตามพิกัดจีพีเอสที่กำหนดไว้ในหน่วยความจำ สามารถเริ่มเล่นไฟล์เสียงอย่างอัตโนมัติเมื่อนำเครื่องที่พัฒนาขึ้นมาเข้าภายในกรอบบริเวณของพื้นที่ที่คำนวณไว้ด้วยโปรแกรม และไมโครคอนโทรลเลอร์สามารถควบคุมให้เล่นไฟล์เสียงบรรยายจนกระทั่งจบการบรรยายข้อมูลของสถานที่นั้นๆ โดยจะไม่เล่นซ้ำแม้จะยังอยู่ภายในกรอบบริเวณ จนกว่าจะออกไปจากพื้นที่ ซึ่งหากนำเครื่องบรรยายเข้าสู่บริเวณพื้นที่ของพิกัดจีพีเอสของสถานที่ท่องเที่ยวใดระบบก็จะ เริ่มบรรยายข้อมูลของสถานที่ท่องเที่ยวนั้นๆได้อย่างถูกต้อง

4. กิตติกรรมประกาศ

ขอขอบคุณคณะวิศวกรรมศาสตร์ มทร.ศรีวิชัย สำหรับการสนับสนุนทุนการวิจัยเงินรายได้ ปีงบประมาณ 2559

เอกสารอ้างอิง

[1] Todd Simcock, Stephen Peter Hillenbrand and Bruce H.Thomass, Developing a Location Based Tourist Guide Application, Australian Computer Society Inc. 2003

[2] สัญญา ผาสุข.(2558).การแจ้งตำแหน่งอุบัติเหตุของยานยนต์ด้วยพิกัด GPS ผ่านเครือข่ายโทรศัพท์เคลื่อนที่. การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้ามหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่7. เชียงใหม่: มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา.

ผศ. สัญญา ผาสุข สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย ต. บ่อยาง อ.เมือง จ.สงขลา 90000. Email: sunya.p@rmutsv.ac.th

นายศุภชัย อรุณพันธ์ สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย ต. บ่อยาง อ.เมือง จ.สงขลา 90000. Email: supachai.a@rmutsv.ac.th

ดร. จินดา สามัคคี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย ต. บ่อยาง อ.เมือง จ.สงขลา 90000. Email: chinda.s@rmutsv.ac.th

IN