

รถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส

AUTONOMOUS CAR WITH GPS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

รถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส

AUTONOMOUS CAR WITH GPS



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTONOMOUS CAR WITH GPS



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2556

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง รถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส
AUTONOMOUS CAR WITH GPS

ผู้จัดทำ	นายอดิศร	ธรรมดี	53011825
	นายอภิวัฒน์	วิษณุวรรณกุล	53011870
	นายอัศวินท์	แก้วมีศรี	53011908



..... อาจารย์ที่ปรึกษา

(รองศาสตราจารย์สุเชียร เกียรติสุนทร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส

โดย

นายอดิศร	ธรรมดี	53011825
นายอภิวัฒน์	วิษณุวรรณกุล	53011870
นายอัศวินท์	แก้วมีศรี	53011908

อาจารย์ที่ปรึกษา

รองศาสตราจารย์สุเชีร เกียรติสุนทร

ปีการศึกษา 2556

บทคัดย่อ

รถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส (Autonomous car with GPS) เป็นรถที่ขับเคลื่อนได้ด้วยตัวเองผ่านระบบจีพีเอส โดยการใช้ GPS module เป็นตัวรับสัญญาณจากดาวเทียม รถจะวิ่งไปตามค่าจีพีเอส ที่เราได้ทำการระบุตำแหน่งไว้ซึ่งพอถึงจุดหมายรถจะหยุดเองโดยอัตโนมัติ ถือเป็นนวัตกรรมความก้าวหน้าทางเทคโนโลยีอย่างหนึ่ง ที่สามารถนำไปประยุกต์ใช้ได้ในอนาคต

โครงการรถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส (Autonomous car with GPS) ได้จัดทำขึ้นเพื่อศึกษาผลตอบสนองของรถต่อค่าจีพีเอสที่เป็นระบบควบคุมแบบป้อนกลับว่ารถนั้นมีปฏิกิริยาเป็นอย่างไรเมื่อรับสัญญาณจากดาวเทียมที่มีค่าเปลี่ยนไป

AUTONOMOUS CAR WITH GPS

By

Mr. Adisorn Thamdee 53011825

Mr. Aphiwat Vitchayawannakul 53011870

Mr. Akarawin Kaewmesri 53011908

Advisor

Assoc. Prof. Suthian Kiatsunthorn

Academic year 2013

ABSTRACT

Autonomous car with GPS is an automatic car which is driving itself by GPS system. By using GPS module as a signal receiver to detect signal from satellites. The car will be driving itself follow to GPS Values which is set. When it approaches to the point, it will stop automatically. This is an innovation that we can apply in the future.

Autonomous car with GPS project has created for studying car's respond to GPS values which is an feedback control system and learn how it reaction when GPS values is changing.

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดีนั้น เพราะทางคณะผู้จัดทำได้รับความช่วยเหลือจาก รองศาสตราจารย์สุเชียร เกียรติสุนทร ที่ท่านได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น ทางผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบคุณ อาจารย์เทพจิตร เขยโสภา ที่คอยให้ความช่วยเหลือเกี่ยวกับเรื่องแบตเตอรี่ และเรื่องเกี่ยวกับ GPS มาโดยตลอดตั้งแต่ตอนที่เกิดปัญหาจนสิ้นสุดปัญหา

ขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ อีกทั้งสนับสนุนอุปกรณ์ที่ขาดเหลือ ตกเดือนตอนทำงาน รวมทั้งการได้สอบถามถึงความคืบหน้าของโครงการอย่างห่วงใยอยู่เสมอ

สุดท้ายนี้ทางคณะผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องเงินของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการชิ้นนี้ได้เสร็จสมบูรณ์ลงได้

ผู้จัดทำ
นายอดิศร ธรรมดี

นายอภิวัฒน์ วิชญวรณกุล

นายอัศวินท์ แก้วมีศรี

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	1
1.3 ขอบเขตการศึกษา	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 องค์ประกอบของโครงการ	2
บทที่ 2 ทฤษฎีหรือหลักการ	3
2.1 จีพีเอส	3
2.1.1 หลักการทำงานพื้นฐานของจีพีเอส	3
2.1.2 การอ่านข้อมูลจาก GPS Module	3
2.2 ไมโครคอนโทรลเลอร์	4
2.2.1 โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์	5
2.3 Compass Module	5
2.3.1 การเชื่อมต่ออุปกรณ์แบบ I ² C Bus	6
2.3.2 การเขียน-อ่านข้อมูลแบบ I ² C Bus	6
2.3.3 รหัสควบคุม I ² C Bus	7
2.3.4 ช่วงเวลารับส่งบิตข้อมูล I ² C Bus	8
2.4 องค์ประกอบของเซอร์โวมอเตอร์	9
2.4.1 ประเภทของเซอร์โวมอเตอร์	10
2.4.2 เซอร์โวมอเตอร์ที่เลือกใช้	11
2.5 DC Motor	12
2.5.1 หลักการทำงานของ DC Motor	12
2.5.2 การใช้งานมอเตอร์	13
2.6 IC driver L293	13

สารบัญ(ต่อ)

	หน้า
2.7 Wireless Transceiver Module	13
2.7.1 หลักการทำงาน Wireless Transceiver Module	14
2.8 แบตเตอรี่และเครื่องชาร์จแบตเตอรี่	15
บทที่ 3 การคำนวณและการสร้าง	16
3.1 วิธีที่ใช้ในการศึกษาค้นคว้า	16
3.2 ลักษณะข้อมูล	16
3.2.1 ลักษณะข้อมูลเชิงตัวเลข	16
3.2.2 ข้อมูลเชิงกายภาพ	16
3.2.3 การเลือกข้อมูล	16
3.3 เครื่องมือ	16
3.4 ขั้นตอนในการสร้าง	17
3.4.1 การคิดหัวข้อและการออกแบบเบื้องต้น	17
3.4.2 การศึกษาข้อมูล	17
3.4.3 การทดลองการทำงานของอุปกรณ์ต่างๆ	17
3.4.4 การทดลองวิ่งช่วงที่หนึ่ง	17
3.4.5 การพัฒนาโครงงานครั้งที่หนึ่ง	17
3.4.6 การทดลองวิ่งช่วงที่สอง	18
3.4.7 การปรับแต่งรถ	18
3.4.8 การพัฒนาส่วนของการแสดงผล	18
3.4.9 การทดลองวิ่งช่วงที่สาม	18
บทที่ 4 การทดลอง	20
4.1 การทดลอง Servo Motor	20
4.2 การทดลองสัญญาณเข็มทิศ	21
4.3 การทดลอง GPS Module	22
4.4 การทดลอง DC Motor	23
4.5 การทดลอง Wireless Transceiver Module	24
4.6 การทดลองสวิทซ์ในการสั่งการวิ่ง	25
บทที่ 5 วิจารณ์และสรุป	28
5.1 สรุปผลการทดลอง	28
5.2 ปัญหาที่พบและแนวทางแก้ไข	28

สารบัญ(ต่อ)

	หน้า
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	29
เอกสารอ้างอิง	30
ภาคผนวก	32
ภาคผนวก ก โปรแกรมและการประมวลผล	33
ก.1 โค้ดแสดงการรับค่าจีพีเอส 2 ค่า	33
ก.2 โค้ดแสดงการรับค่าจีพีเอสทุกค่า	34
ก.3 โค้ดการขับ DC Motor	34
ก.4 โค้ดการขับ Servo Motor	36
ก.5 โค้ดแสดงเข็มทิศ	37
ก.6 โค้ดแสดงสวิตช์	39
ก.7 โค้ดลอง Wifi ตัวรับ	41
ก.8 โค้ดลอง Wifi ตัวส่ง	44
ก.9 โค้ดจริง Wifi ตัวรับ	47
ก.10 โค้ด Run แบบสมบูรณ์	50
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	84
ข.1 คู่มือ Arduino Mega 2560 R3	84
ข.2 คู่มือ nRF905	92
ข.3 คู่มือ GPS Module	133

สารบัญรูป

รูปที่	หน้า
2.1 GPS Module ที่ใช้ในโครงการ	4
2.2 Arduino Mega 2560 R3	5
2.3 HMC5883L: Compass Module	6
2.4 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I ² C Bus	6
2.5 แบบการเขียน/อ่านข้อมูลแบบ I ² C Bus	6
2.6 แสดงสถานะเริ่มและหยุดของ I ² C Bus	7
2.7 แสดงการส่งสัญญาณ	7
2.8 แสดงช่วงเวลาการรับส่งข้อมูล	8
2.9 แสดงชิ้นส่วนต่างๆ ของเซอร์โวมอเตอร์	9
2.10 แสดงการทำงานของ Servo Motor	10
2.11 เซอร์โวมอเตอร์ในงานวิจัย	11
2.12 มอเตอร์กระแสตรง	12
2.13 สัญลักษณ์มอเตอร์กระแสตรง	12
2.14 ส่วนประกอบภายในมอเตอร์กระแสตรง	12
2.15 แสดงคุณสมบัติของ IC Driver L293	13
2.16 Wireless Transceiver Module	14
2.17 การติดต่อสื่อสารด้วย SPI: Serial Peripheral Interface	14
2.18 แบตเตอรี่ลิโธ	15
2.19 เครื่องชาร์จแบตเตอรี่	15
3.1 รถที่เสร็จสมบูรณ์	19
4.1 แสดงการเลียวซ้าย	20
4.2 แสดงมุมจาก Servo Motor เมื่อเลียวซ้าย	20
4.3 แสดงการเลียวขวา	21
4.4 แสดงมุมจาก Servo Motor เมื่อเลียวขวา	21
4.5 แสดงเข็มทิศชี้ไปทางทิศใต้	21
4.6 ข้อมูลจาก GPS Module	22
4.7 การทดลอง DC Motor	23
4.8 ลักษณะการใช้งาน Wireless Module	24

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.9 ข้อมูลที่ได้จากระบบ Wireless Module	24
4.10 ลักษณะสวิตช์ที่ใช้งาน	25
4.11 การทดลองสวิตช์ที่ 1 ถึง 3	26
4.12 การทดลองสวิตช์ที่ 4 และ 5	26
4.13 ตำแหน่งต่างๆ	27
4.14 ถนนภายในสถาบัน	27



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบัน ความก้าวหน้าทางการประยุกต์ใช้เทคโนโลยีเกี่ยวกับการสื่อสารผ่านดาวเทียมได้มีมากขึ้น ทางคณะผู้จัดทำจึงได้เล็งเห็นถึงประโยชน์ของการสื่อสารผ่านดาวเทียมที่สามารถนำไปต่อยอดหรือสามารถนำไปพัฒนาทางด้านการคมนาคมในอนาคตได้คือ การใช้เครื่องรับสัญญาณจีพีเอส โดยการนำเครื่องรับสัญญาณจีพีเอสไปติดตั้งบนรถบังคับ เพื่อทำการนำข้อมูลที่ได้จากเครื่องรับสัญญาณจีพีเอสเป็นตัวบังคับรถแบบอัตโนมัติ เพื่อให้รถนั้นไปในจุดที่ต้องการ โดยเราเพียงแค่ระบุจุดที่ต้องการจะไปเท่านั้น เราจะใช้เทคโนโลยีนี้ในการแก้ปัญหาสำหรับผู้ที่มาติดต่อสถานที่ต่างๆ ในสถาบัน ซึ่งปัญหาที่พบคือบุคคลภายนอกไม่รู้ตำแหน่งของอาคารทำให้เสียเวลาในการหาอาคารนั้นๆ อีกทั้งยังไม่มีผู้ที่จะให้คำแนะนำได้อย่างถูกต้องได้เท่ากับการที่หาบุคคลภายนอกไปยังจุดหมายด้วยตัวเอง

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาการเขียนโปรแกรมไมโครคอนโทรลเลอร์ในการควบคุมรถ
2. เพื่อนำประโยชน์ของการสื่อสารผ่านดาวเทียมมาใช้งาน
3. เพื่อควบคุมการแสดงผลโดยการส่งงานผ่านไมโครคอนโทรลเลอร์
4. เพื่อสร้างงานที่ควบคุมแบบอัตโนมัติโดยอาศัยการสื่อสารผ่านดาวเทียม
5. เพื่อหาผลตอบแทนของรถต่อค่าที่ได้จากสัญญาณ GPS

1.3 ขอบเขตการศึกษา

1. ศึกษาผลตอบแทนของจีพีเอสโมดูล คอมแพสโมดูล และไวไฟโมดูล
2. ศึกษาผลตอบแทนของเซอร์โวมอเตอร์ต่อมุมมองศาที่เปลี่ยนไป
3. ศึกษาผลตอบแทนโดยรวมของรถต่อค่าจีพีเอสที่เปลี่ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถวิ่งไปตามค่าพิกัด GPS ที่เราได้โปรแกรมไว้
2. ค่า GPS ต้องเปลี่ยนตลอดทุกๆ การเคลื่อนไหว
3. ได้รับความรู้เกี่ยวกับการโปรแกรมผ่านตัวไมโครคอนโทรลเลอร์
4. สามารถนำความรู้ที่ได้จากการทำงานไปต่อยอดได้ในอนาคต
5. เป็นต้นแบบให้กับการพัฒนาทางด้านรถเคลื่อนที่อัตโนมัติด้วย GPS ได้

1.5 องค์ประกอบของโครงการ

บทที่ 1 บทนำ ซึ่งจะกล่าวถึงความเป็นมา และความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา ขอบเขตการศึกษา ประโยชน์ที่คาดว่าจะได้รับ องค์ประกอบของโครงการ

บทที่ 2 ทฤษฎีหรือหลักการที่เกี่ยวข้อง กล่าวถึงหลักการของอุปกรณ์ต่างๆ ที่ใช้ประกอบกัน เพื่อสร้างรถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส และการนำเอาความรู้ไปประยุกต์ใช้ในการจัดทำโครงการ

บทที่ 3 การคำนวณและการสร้าง กล่าวถึงวิธีที่ใช้ในการศึกษาค้นคว้าข้อมูล อุปกรณ์ที่ใช้ในการทำงานวิจัย ขั้นตอนในการสร้างตั้งแต่เริ่มศึกษาจนเสร็จสมบูรณ์ ซึ่งรวมถึงปัญหาที่พบเจอในแต่ละขั้นตอน

บทที่ 4 การทดลอง เป็นส่วนของการทดสอบองค์ประกอบส่วนต่างๆ ในระบบ ตลอดจนการทดลองการวิ่งของรถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส

บทที่ 5 วิจารณ์และสรุป เป็นการสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป

บทที่ 2

ทฤษฎีหรือหลักการ

2.1 หลักการทำงานของจีพีเอส

GPS ย่อมาจาก Global Positioning System คือ ระบบที่ระบุตำแหน่งทุกตำแหน่งบนโลก จากกลุ่มดาวเทียมมากกว่า 24 ดวงที่โคจรรอบโลก ซึ่งถูกปล่อยโดยกระทรวงกลาโหมของอเมริกา

2.1.1 หลักการทำงานพื้นฐานของ GPS : Satellites Triangulation

กล่าวคือดาวเทียมทุกดวงมีรหัสประจำตัว ทำให้เครื่องรับสัญญาณ GPS รู้ว่ากำลังรับสัญญาณ จากดาวเทียมดวงไหน บนผิวโลก ณ ตำแหน่งใดตำแหน่งหนึ่งนั้นจะมีดาวเทียมอยู่เหนือเส้นขอบฟ้า อย่างน้อยสี่ดวงคอยรับส่งสัญญาณนำทางให้แก่เครื่องรับสัญญาณ GPS โดยเราอาศัยตำแหน่ง ดาวเทียมในอวกาศเป็นจุดอ้างอิง แล้ววัดระยะจากดาวเทียมทั้ง 4 ดวง และใช้หลักการทางเรขาคณิต ในการคำนวณหาตำแหน่งบนผิวโลก โดยการวัดระยะทางระหว่างเครื่องรับสัญญาณ GPS กับ ดาวเทียมที่ส่งสัญญาณมา สามารถหาตำแหน่งได้จากสมการ ระยะทาง = ความเร็ว \times เวลาที่ใช้ เดินทาง ซึ่งคลื่นวิทยุจะมีความเร็วประมาณ 186,000 ไมล์ต่อวินาที

2.1.2 การอ่านค่าข้อมูลจาก GPS Module

เป็นการอ่านค่าข้อมูลจากเครื่องรับสัญญาณ GPS (GPS Module Receiver) ผ่าน พอร์ทอนุกรม (Serial Port) โดยใช้มาตรฐานของ NMEA (The National Marine Electronics Association) เป็นมาตรฐานในการอ่านข้อมูล ซึ่ง NMEA เป็นมาตรฐานที่ยอมรับในการส่งข้อมูล Marine Electronics ไปยังเครื่องคอมพิวเตอร์หรืออุปกรณ์อื่นๆ ข้อมูลที่ส่งมาจะมีลักษณะเป็น ประโยคที่เป็นมาตรฐานของ NMEA และทุกๆ ประโยค NMEA จะต้องมีย่อชื่อย่อขึ้นต้น (Prefix) เป็นการกำหนดชนิดของประโยค NMEA นั้นๆ ข้อกำหนดของประโยค NMEA โดยทั่วไปมีลักษณะดังนี้

- ในแต่ละประโยค NMEA จะต้องขึ้นต้นด้วยเครื่องหมาย \$ ก่อน Prefix
- แต่ละประโยค NMEA จะต้องมีความยาวไม่เกิน 80 อักขระ
- รายการของข้อมูลจะถูกแยกด้วยเครื่องหมายคอมมา (,)
- ข้อมูลจะมีลักษณะเป็นรหัส ASCII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อมูลจะเปลี่ยนแปลงตามความเที่ยงตรงที่บรรจุอยู่ในข้อความ
- มีการ Checksum ที่ท้ายประโยค ซึ่งจะเช็คหรือไม่ก็ได้
- การ Checksum ประกอบด้วยเครื่องหมาย * และตัวเลขฐาน 16 (HEX) อีก 2 ตัว แสดงการ Exclusive OR ของอักขระทั้งหมด
- การเชื่อมต่อฮาร์ดแวร์ของชุดอุปกรณ์ GPS จะเป็นการเชื่อมต่อแบบอนุกรม โดยใช้มาตรฐานของ RS-232 เชื่อมต่อกับคอมพิวเตอร์หรือ Microcontroller โดยทั่วไปจะใช้สายสัญญาณเพียง 2 เส้น คือเส้นส่งสัญญาณออกจาก GPS Module และ สาย Ground ความเร็วในการส่งสัญญาณทั่วไปคือแบบ NMEA 0183 [4800 baud rate, 8 bits of data, no parity, และ 1 stop bit] ซึ่งจะทวนสัญญาณทุกๆ 1 วินาที



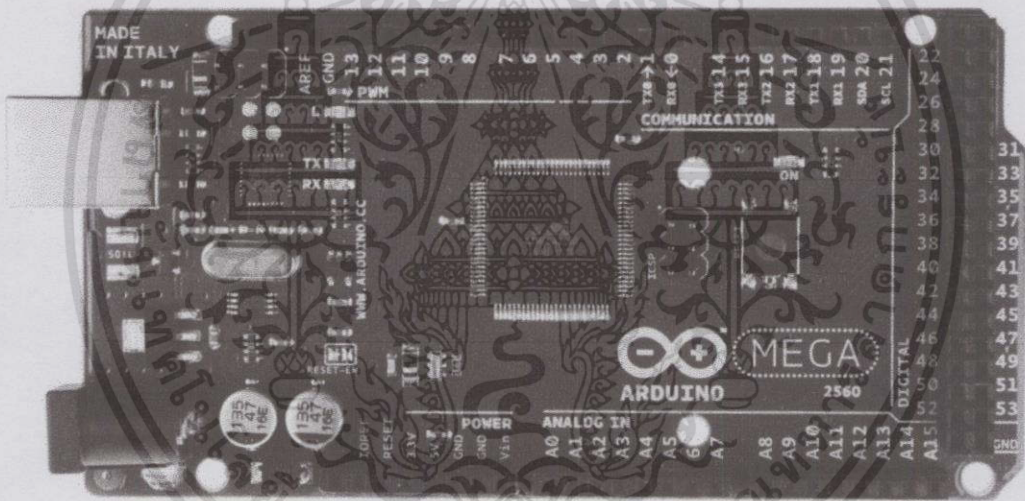
รูปที่ 2.1 GPS Module ที่ใช้ในโครงการ

2.2 ไมโครคอนโทรลเลอร์

(Microcontroller หรือ MCU) คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน

2.2.1 โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์ นั้นสามารถแบ่งออกได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

- หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)
- หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ
 - หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory)
 - หน่วยความจำข้อมูล (Data Memory) คล้ายกับ RAM
- ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port)
- เส้นทางเดินสัญญาณ หรือบัส (BUS)
- วงจรกำเนิดสัญญาณนาฬิกา



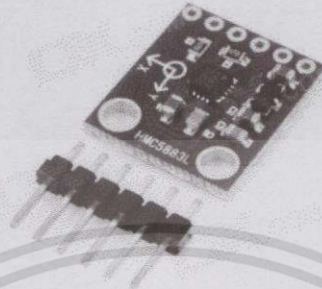
รูปที่ 2.2 Arduino Mega 2560 R3 เป็นไมโครคอนโทรลเลอร์ที่ใช้ในการทำโครงงานครั้งนี้

2.3 Compass Module

สนามแม่เหล็กโลกเกิดจากกระบวนการไดนาโมของโลก กล่าวคือโลหะหนักที่มีสถานะเป็นของเหลวที่อยู่ในแกนโลกมีการหมุนวน ทำให้เกิดสนามแม่เหล็กที่เอียงทำมุมประมาณ 10 องศาจากแกนหมุนของโลก ที่บริเวณผิวโลกมีความเข้มของสนามแม่เหล็กโลกประมาณ 30,000 - 60,000 นาโนเทสลา และความเข้มจะค่อยๆ ลดลงเมื่ออยู่ห่างจากผิวโลกมากขึ้น ซึ่งเมื่อนำแท่งแม่เหล็กที่ไร้แรงเสียดทานอันหนึ่งมาตั้งไว้ จะทำให้แท่งแม่เหล็กนั้นเบนชี้ไปทางทิศเหนือเสมอ ซึ่งสามารถนำหลักการนี้มาใช้สร้าง Compass Module โดยภายใน Compass Module จะมีเหล็กที่สามารถก่อให้เกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

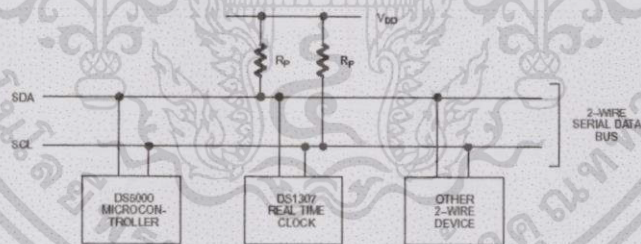
สนามแม่เหล็กอยู่ภายใน เมื่อหมุนตัว Compass Module ไป จะทำให้ค่าสนามแม่เหล็กภายในเปลี่ยนแปลงตามแกน X Y และ Z ซึ่งสามารถนำค่าแต่ละแกนมาคำนวณหาทิศทางของการหมุนนั้นได้



รูปที่ 2.3 HMC5883L: Compass Module

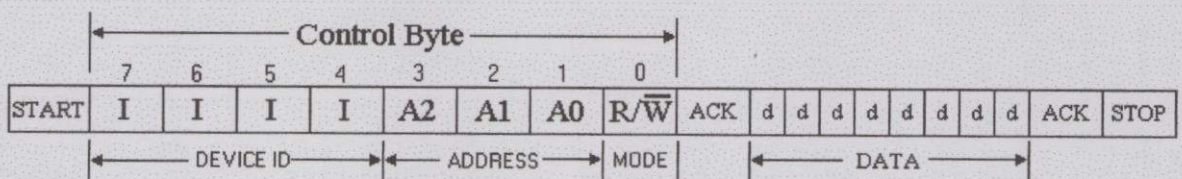
2.3.1 การเชื่อมต่ออุปกรณ์แบบ I²C Bus

ใช้สายสัญญาณ 2 เส้น คือ SCL, SDA สำหรับใช้ติดต่อกับอุปกรณ์แบบ 2 ทิศทาง โดยที่ขาสัญญาณทั้ง 2 จะต้องต่อกับตัวต้านทานแบบ Pull Up 2-10K เนื่องจากเอาต์พุตมีลักษณะเป็นแบบ Open Drain หรือเป็นแบบ Open Collector เพื่อให้เอาต์พุตเชื่อมต่อกันได้หลายตัว



รูปที่ 2.4 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I²C Bus

2.3.2 การเขียนอ่านข้อมูลกับอุปกรณ์แบบ I²C Bus



รูปที่ 2.5 แบบการเขียน/อ่านข้อมูลแบบ I²C Bus

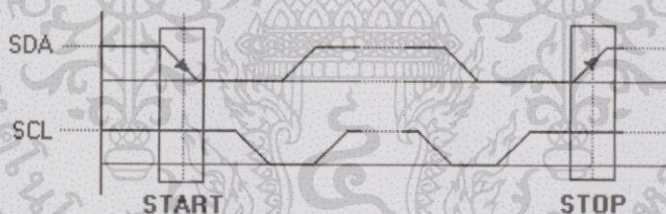
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งข้อมูลแบบ I²C Bus ส่วน MCU จะเริ่มการส่งข้อมูลด้วยวิธีการดังนี้

- ส่งสถานะเริ่มต้น (Start Condition) เพื่อแสดงการขอใช้ Bus
- ตามด้วยรหัสควบคุม (Control Byte) เพื่อแสดงการขอใช้บัส
- เมื่ออุปกรณ์รับทราบว่า MCU ต้องการที่จะติดต่อสื่อสารก็จะส่งสถานะรับรู้ด้วยรหัสประจำตัว Device ID, Device Address, และ Mode ในการเขียนหรืออ่านข้อมูล
- และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Condition) เพื่อบอกอุปกรณ์ว่าสิ้นสุดการใช้บัส

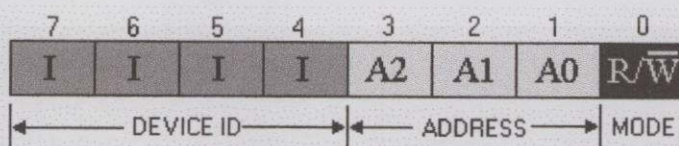
การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I²C Bus (Start and Stop condition)

- เมื่อต้องการส่งข้อมูล MCU จะต้องส่งสถานะเริ่มต้น (Start Condition) คือให้ SDA เปลี่ยนจาก 1 มาเป็น 0 ในขณะที่ SCL มีค่าเป็น 1
- เมื่อสิ้นสุดการใช้บัส MCU จะต้องส่งสถานะสิ้นสุด (Stop Condition) คือให้ SDA เปลี่ยนจาก 0 มาเป็น 1 ในขณะที่ SCL มีค่าเป็น 1



รูปที่ 2.6 แสดงสถานะเริ่มและหยุด ของ I²C Bus

2.3.3 รหัสควบคุมของ ของ I²C Bus (Control Byte)



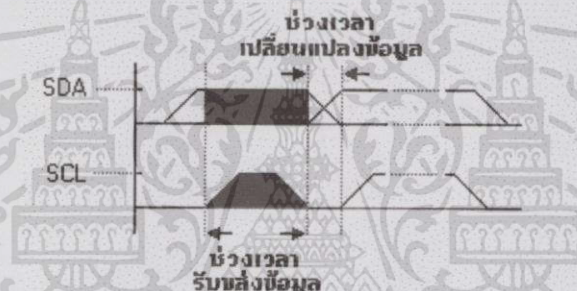
รูปที่ 2.7 แสดงการส่งสัญญาณส่วนรหัสควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสควบคุมของ I²C Bus ประกอบด้วยรหัสประจำตัวของอุปกรณ์ (Device ID) ประกอบด้วยบิตที่ 1-7 และบิต 0 เป็นบิตควบคุมการเขียนอ่าน

- รหัสประจำตัวของอุปกรณ์ประกอบด้วยรหัสประจำตัวจากผู้ผลิต (Product ID) 4 บิต (บิต 4-7) ที่เปลี่ยนแปลงแก้ไขไม่ได้ และ Device Address 3 บิต (บิต 1-3) ซึ่งผู้ใช้สามารถกำหนดเองได้ รวมแล้วเป็นรหัส 7 บิต ใช้ระบุอุปกรณ์ที่ต่ออยู่บนบัส ซึ่งจะมีค่าซ้ำกันไม่ได้
- บิตควบคุมการเขียนอ่าน (Mode) บิต 0 เมื่อ MCU ต้องการเขียนข้อมูลไปยังอุปกรณ์ก็กำหนดให้บิตนี้เป็น 0 และเมื่อต้องการอ่านข้อมูลจากอุปกรณ์ก็กำหนดให้บิตนี้เป็น 1

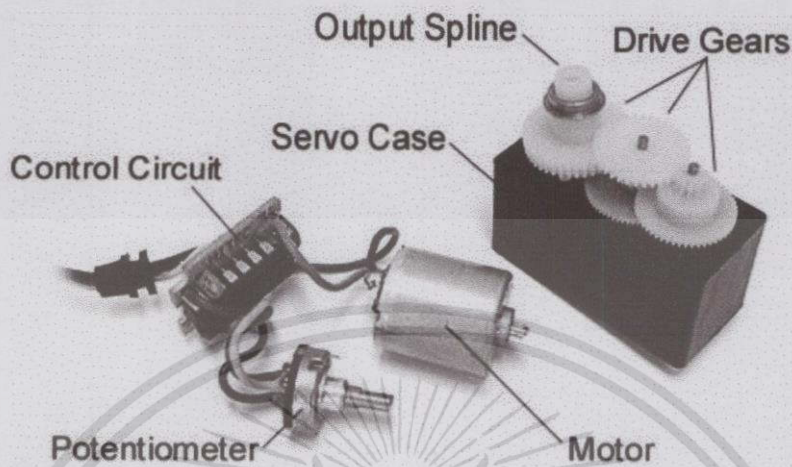
2.3.4 ช่วงเวลารับส่งบิตข้อมูล I²C Bus



รูปที่ 2.8 แสดงช่วงเวลาการรับส่งข้อมูล

- สถานะรับส่งข้อมูลจะกระทำในขณะที่ SCL เป็น 1
- สถานะการเปลี่ยนแปลงข้อมูลจะกระทำในขณะที่ SCL เป็น 0

2.4 องค์ประกอบพื้นฐานของเซอร์โวมอเตอร์



รูปที่ 2.9 แสดงชิ้นส่วนต่างๆ ของเซอร์โวมอเตอร์

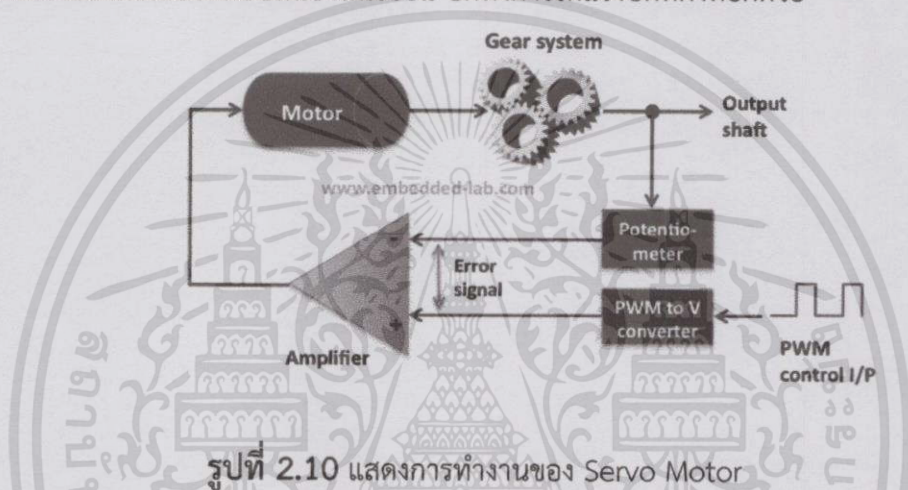
องค์ประกอบหลักของเซอร์โวมอเตอร์โดยทั่วไปแล้วจะมีส่วนประกอบหลักดังนี้คือ

1. Servo Case ซึ่งส่วนใหญ่จะทำมาจากพลาสติก
2. Motor ซึ่งเป็นส่วนให้กำลังในการหมุนของเซอร์โวมอเตอร์
3. Control Circuit มีหน้าที่ในการถอดรหัสสัญญาณควบคุมจากรีซีฟเวอร์ ซึ่งส่งมาเป็นสัญญาณแบบ PWM และส่งการควบคุมไปสั่งการทำงานของมอเตอร์ให้หมุนแขนของเซอร์โวมอเตอร์ไปอยู่ในตำแหน่งที่ได้ถอดรหัสมา
4. Potentiometer คือส่วนที่ตรวจวัดตำแหน่งของเซอร์โวมอเตอร์และส่งสัญญาณกลับไปยัง Control Circuit เพื่อแก้ไขตำแหน่งให้ถูกต้องตามสัญญาณที่ได้ตั้งไว้
5. Drive Gear คือชุดเฟืองที่ถอดจากการหมุนของมอเตอร์เพื่อให้ได้แรงบิดที่สูงขึ้น
6. Output Spline คือส่วนที่ป้องกันการเสียดสีระหว่าง Servo Case และ Output Shaft ซึ่งอาจใช้อุปกรณ์ประเภท Baring เพื่อช่วยลดแรงเสียดทานที่ดี
7. Servo Wire คือสายไฟของเซอร์โวมอเตอร์ซึ่งในกรณีที่เป็นเซอร์โวมอเตอร์ 3 สาย แต่ละสายจะมีหน้าที่และรายละเอียดดังนี้
 - เส้นที่ 1 จ่ายไฟกระแสบวก DC ซึ่งแรงดันปกติจะอยู่ที่ 5-6 โวลท์
 - เส้นที่ 2 เป็นสาย Ground หรือเป็นขั้วลบ DC
 - เส้นที่ 3 เป็นสายสัญญาณ โดยที่รีซีฟเวอร์จะส่งสัญญาณลักษณะ On/Off Pulsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

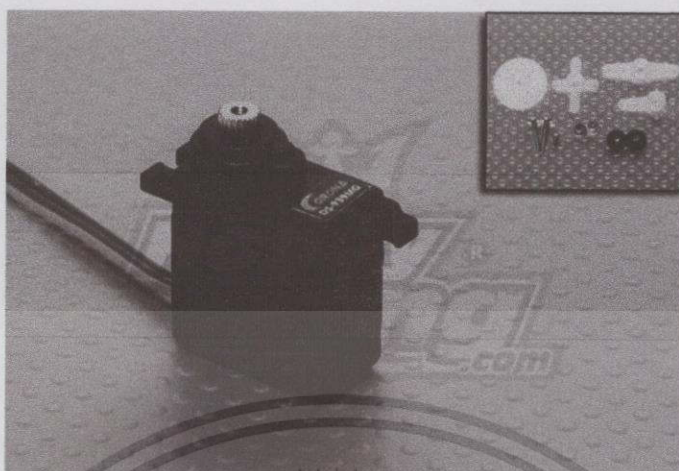
2.4.1 ประเภทของเซอร์โวมอเตอร์ เซอร์โวมอเตอร์นั้นแบ่งได้เป็น 2 ประเภท คือ Analog Servo Operation และ Digital Servo Operation เซอร์โวมอเตอร์ระบบอนาล็อกและระบบดิจิทัลแทบจะเหมือนกันทุกประการแม้กระทั่งสายไฟทั้งสามเส้นที่ต่อจากรีซีฟเวอร์ก็มีลักษณะเหมือนกัน แต่สิ่งที่ต่างกันก็คือการแปลสัญญาณที่ส่งมาจากรีซีฟเวอร์ แล้วส่งต่อไปยังมอเตอร์ของเซอร์โวเท่านั้น

โดยตัวเซอร์โวมอเตอร์ระบบดิจิทัลจะมีไมโครโปรเซสเซอร์อยู่ในตัวเซอร์โวมอเตอร์คอยทำหน้าที่แปลสัญญาณที่ส่งมาจากรีซีฟเวอร์ และส่งเป็นสัญญาณไฟฟ้าที่ความถี่สูงไปยังเซอร์โวมอเตอร์ แทนที่จะเป็น 50 ครั้งต่อวินาทีเหมือนเช่นระบบอนาล็อก แต่ในระบบดิจิทัลนี้มีความถี่ในการอัปเดตสัญญาณสูงถึง 300 ครั้งต่อวินาที ซึ่งแน่นอนว่าคลื่นสัญญาณจะมีความยาวที่สั้นลง แต่สิ่งที่ได้คือความเร็วของมอเตอร์ที่ตอบสนองได้เร็วขึ้น อีกทั้งการให้แรงบิดที่คงที่อีกด้วย



ตามภาพด้านบนซึ่งแสดงระบบการทำงานของเซอร์โวมอเตอร์ โดยที่รีซีฟเวอร์จะส่งสัญญาณการควบคุมตำแหน่งของเซอร์โวมอเตอร์ไปยังส่วน Control Circuit ของเซอร์โวมอเตอร์โดยสัญญาณที่ส่งมาจะเป็นสัญญาณแบบ PWM (Pulse Width Modulation) จากนั้น Control Circuit จะถอดรหัสสัญญาณ PWM ที่ได้ให้เป็นตำแหน่งของเซอร์โวมอเตอร์ที่ต้องการโดยเปรียบเทียบกับค่าตำแหน่งปัจจุบันกับค่าสัญญาณจาก Potentiometer แล้วจึงส่งแรงดันไฟฟ้าไปยังมอเตอร์ให้ไปหมุนไปในทิศทางที่จะทำให้ตำแหน่งของ Potentiometer มีค่าที่ถูกต้องเท่ากับค่าที่ได้ถอดรหัสมา ซึ่งขณะที่มอเตอร์หมุนก็จะมีเฟืองที่ไปต่อกับแกนของ Potentiometer (ปกติจะอยู่ในแกนเดียวกับ Output Shaft) ด้วย ดังนั้นกระบวนการนี้จะเกิดขึ้นซ้ำๆ จนกว่าค่าของ Potentiometer จะมีค่าเท่ากับการถอดรหัสสัญญาณที่ได้รับมาจากรีซีฟเวอร์ การทำงานของมอเตอร์จึงจะหยุด แต่กระบวนการทำงานของ Control Circuit จะยังทำงานอยู่ตลอดเวลาเพียงแต่หากค่าของ Potentiometer มีค่าเท่ากับสัญญาณที่ถอดรหัสจากรีซีฟเวอร์แล้วก็จะไม่มีการส่งแรงดันไฟฟ้าไปยังมอเตอร์ ซึ่งกระบวนการนี้เรียกว่าการอัปเดตสัญญาณโดยมีความเร็วที่ 50 ครั้งต่อหนึ่งวินาที เราจึงเห็นเป็นการเคลื่อนที่ของเซอร์โวมอเตอร์

2.4.2 เซอร์โวมอเตอร์ที่เลือกใช้ สำหรับงานวิจัยนี้คือ Corona DS-939MG



รูปที่ 2.11 เซอร์โวมอเตอร์ในงานวิจัย

คุณสมบัติ

Operating Voltage: 4.8 V / 6.0 V

Operating Current: 200 mA / 240 mA

Operating Speed: 0.14 sec.60° / 0.13 sec.60°

Stall Torque: 2.5 kg.cm / 2.7 kg.cm

Size: 22.5 X 11.5 X 24.6 mm (0.88"x0.45"x0.96")

Weight: 12.5 g

Gear: Metal

Connector Wire: 215 mm

Operating Temperature Range: -20J+60J

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 DC Motor

มอเตอร์กระแสตรงเป็นเครื่องกลไฟฟ้าที่ทำหน้าที่เปลี่ยนแปลงพลังงานไฟฟ้าให้เป็นพลังงานกล (หมุน) ซึ่งเรามักจะใช้มอเตอร์เมื่อต้องการให้วัตถุเกิดการหมุน เช่น ใบพัดล้อรถ เป็นต้น

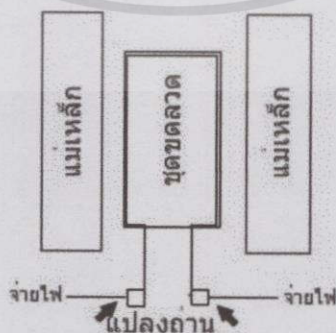


รูปที่ 2.12 มอเตอร์กระแสตรง

รูปที่ 2.13 สัญลักษณ์มอเตอร์กระแสตรง

2.5.1 หลักการทำงานของ DC Motor

มอเตอร์จะทำงานตามหลักการแม่เหล็กไฟฟ้า เมื่อเรานำเอาขดลวดพันแกนเหล็กหลายรอบแล้วป้อนกระแสไฟฟ้าเข้าไปในขดลวดนั้น แกนเหล็กจะกลายเป็นแม่เหล็ก (แต่จะเป็นแบบชั่วคราวเท่านั้น) และเมื่อเรานำไฟฟ้าออก แกนเหล็กจะกลายเป็นแกนเหล็กธรรมดา กล่าวคือ เมื่อแม่เหล็ก 2 อันมีขั้วต่างกันจะเกิดการผลักกัน ดังนั้นในตัวมอเตอร์เองจึงประกอบด้วยเหล็กที่มีขั้วต่างกัน เมื่อเราป้อนพลังงานให้กับมอเตอร์จะเกิดเป็นแม่เหล็กไฟฟ้า และเกิดการผลักกันของแม่เหล็กทั้งสอง โดยการผลักกันนี้คือการหมุนของมอเตอร์นั่นเอง



รูปที่ 2.14 ส่วนประกอบภายในมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

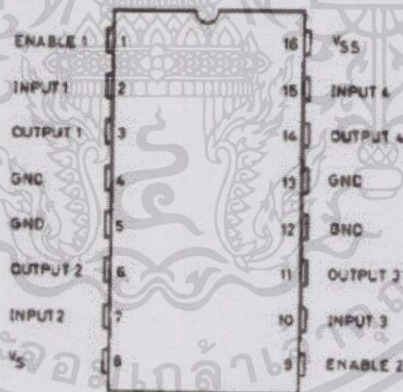
2.5.2 การใช้งานมอเตอร์

การนำมอเตอร์ไปใช้งานจะต้องคำนึงถึงแรงดันไฟฟ้าที่ป้อนให้มอเตอร์ และทิศทางการหมุนของมอเตอร์ การที่เราป้อนแรงดันไฟฟ้าต่ำกว่าแรงดันปกติของมอเตอร์จะทำให้มอเตอร์หมุนช้าลงหรืออาจไม่หมุน และแรงบิดที่ได้จะน้อย และในทำนองเดียวกันหากเราป้อนแรงดันสูงกว่าแรงดันที่กำหนด จะทำให้มอเตอร์หมุนเร็วขึ้น และอาจเกิดความเสียหายแก่ตัวมอเตอร์เอง เช่น ขดลวดไหม้หรือขาดได้

สิ่งที่สำคัญในการนำมอเตอร์ไปใช้งานอย่างหนึ่งคือ การที่มอเตอร์รับภาระในการหมุนมากเกินไป หรือการฝืนให้มอเตอร์หมุนโดยที่มอเตอร์ไม่มีแรงบิดมากพอ จะทำให้มอเตอร์เกิดกระแสลัดวงจรในขดลวด และเกิดการเสียหาย (ซึ่งอาจจะไม่ค่อยเห็นผลในมอเตอร์ขนาดเล็ก แต่สำหรับมอเตอร์ขนาดใหญ่ถือเป็นเรื่องสำคัญมาก)

2.6 IC driver L293

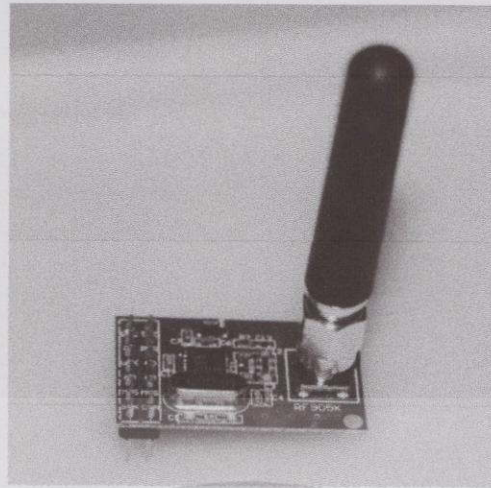
เป็น IC ที่มีจำนวน 16 ขา ถูกออกแบบมาให้ขับเคลื่อนกระแสไปจนถึง 1 แอมแปร์ ใน 2 ทิศทาง ที่แรงดันตั้งแต่ 4.5 – 36 โวลต์ ทั้งนี้เพื่อขับโหลดอินดักทีฟ เช่น รีเลย์ โซลินอยด์ มอเตอร์ไฟฟ้ากระแสตรง และสแต็ปมิ่งมอเตอร์ โดยอินพุตรับสัญญาณมาตรฐาน DTL และ TTL



รูปที่ 2.15 แสดงคุณสมบัติของ IC driver L293

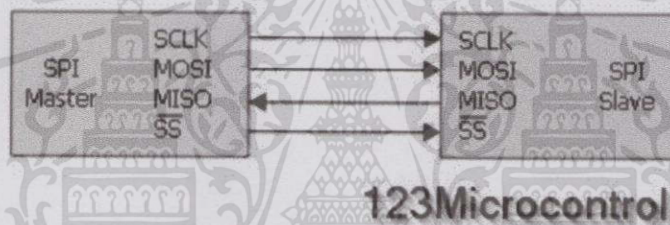
2.7 Wireless Transceiver Module

บอร์ด Wireless Module ชนิดนี้ใช้ชิพที่มีประสิทธิภาพสูงคือชิพ Nordic VLSI NRF905 สามารถส่งข้อมูลได้สูงสุด 50 Kbps with GFSK มีความไวถึง -100 dBm ซึ่งมีความน่าเชื่อถือมาก สามารถนำมาใช้ได้อย่างแพร่หลายงานต่างๆ ที่ระยะห่างของตัวรับ-ส่งไม่ไกลกันมาก



รูปที่ 2.16 Wireless Transceiver Module

2.7.1 หลักการทำงานของ Wireless Transceiver Module



รูปที่ 2.17 การติดต่อสื่อสารด้วย SPI : Serial Peripheral Interface

SPI หรือ Serial Peripheral Interface เป็นวิธีการสื่อสารรูปแบบหนึ่ง ที่ใช้ในการติดต่อสื่อสารกับอุปกรณ์ ตัวอย่างเช่น พริ้นเตอร์ กล้องถ่ายรูป เครื่องสแกนเนอร์ และอื่นๆ อีกมากมาย ถึงแม้ว่าการสื่อสารของ USB ที่มีฟังก์ชันการทำงานที่กว้างกว่า แต่การสื่อสารในรูปแบบ SPI ก็ยังถูกใช้งานกันอยู่ในบาง Application SPI ทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็น MASTER ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็น SLAVE และส่งข้อมูลในโหมด Full-duplex นั้นหมายความว่า สัญญาณสามารถส่งหากันได้ระหว่าง MASTER และ SLAVE ได้อย่างต่อเนื่อง ในการสื่อสารแบบ SPI นี้ ไม่ได้กำหนดมาตรฐานตายตัวว่าข้อมูลที่ส่งหากันจะต้องอยู่ในรูปแบบไหน โดยเป็นการคิด Protocol การสื่อสารกันเอง

2.8 แบตเตอรี่และเครื่องชาร์จแบตเตอรี่

ในงานวิจัยนี้อุปกรณ์ทั้งสองประเภทประกอบไปด้วยแบตเตอรี่ HeliCox Nano 11.1V 3000mAh 30C และเครื่องชาร์จแบตเตอรี่ลิโพ Skyrc E3



รูปที่ 2.18 แบตเตอรี่ลิโพ



รูปที่ 2.19 เครื่องชาร์จแบตเตอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

3.1 วิธีที่ใช้ในการศึกษาค้นคว้า

วิธีที่ใช้ในการศึกษาค้นคว้าและทดลองโดยการใช้ความรู้ที่ได้จากในการเรียนวิชา ไมโครคอนโทรลเลอร์มาประยุกต์ใช้กับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ โดยได้มีการนำเอาตัวอุปกรณ์แต่ละชนิดมาทดลองใช้กับไมโครคอนโทรลเลอร์ว่าให้ค่าอย่างไรบ้าง เป็นอย่างที่ต้องการหรือไม่ เมื่อศึกษาและสามารถที่จะใช้อุปกรณ์แต่ละชนิดเป็นแล้ว จึงนำอุปกรณ์ที่ศึกษาทุกตัวมารวมกันและสร้างระบบควบคุมให้แต่ละอุปกรณ์ทำงานสัมพันธ์กัน เพื่อให้เป็นไปตามจุดประสงค์ของโครงการวิจัย

3.2 ลักษณะข้อมูล สามารถแบ่งออกได้เป็น 2 จำพวก คือ

3.2.1 ข้อมูลเชิงตัวเลข ซึ่งข้อมูลในลักษณะนี้จะถูกประมวลผลผ่านไมโครคอนโทรลเลอร์ออกมาแสดงค่าตัวเลขทางหน้าจอคอมพิวเตอร์

3.2.2 ข้อมูลทางกายภาพ โดยสามารถสังเกตได้จากการมองเห็น อาทิเช่น การหมุนของมอเตอร์ เป็นต้น

3.2.3 การเลือกข้อมูล เราจะนำข้อมูลทั้งสองแบบมาประยุกต์เข้าด้วยกัน โดยผ่านทางไมโครคอนโทรลเลอร์

3.3 เครื่องมือ ที่ใช้ในการทำโครงการประกอบไปด้วยอุปกรณ์สำคัญดังต่อไปนี้

- ไมโครคอนโทรลเลอร์ (Arduino Mega2560 R3)
- จีพีเอสโมดูล (UBLOX NEO-6M)
- เซอร์โวมอเตอร์ (TowerPro 90S รุ่น MG90S)
- Compass Module (HMC5883L)
- Wireless Transceiver Module (nRF905)
- DC Motor ขนาด 9 V
- แผงสวิตช์
- โครงรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนในการสร้าง สามารถแบ่งเป็นหลายขั้นตอน ดังนี้

3.4.1 การคิดหัวข้อและการออกแบบเบื้องต้น ในขั้นตอนนี้เป็นการรวบรวมความคิดในการกำหนดเป้าหมาย และขอบเขตของโครงการวิจัย รวมทั้งแสดงความเห็นในการสร้างลักษณะชิ้นงานให้เป็นรูปธรรม

3.4.2 การศึกษาข้อมูล โดยการศึกษาหาข้อมูลผ่านทางอินเทอร์เน็ต และสอบถามผู้ที่มีความรู้ทางด้านไมโครคอนโทรลเลอร์ ซึ่งในส่วนนี้จะเป็นการศึกษาหาความเป็นไปได้ในการทำโครงการวิจัย และศึกษาวิธีการใช้งานอุปกรณ์ต่าง ๆ รวมถึงการคิดงบประมาณค่าใช้จ่ายในการสร้างชิ้นงาน

3.4.3 การทดลองการทำงานของอุปกรณ์ต่างๆ ในส่วนของขั้นตอนนี้จะเป็นการนำอุปกรณ์ต่างๆ ที่ได้ศึกษามาจากในขั้นตอนที่ 3.4.2 มาทำการทดลองโดยสั่งงานผ่านทางไมโครคอนโทรลเลอร์ ซึ่งเริ่มจากการเขียนโปรแกรมเพื่อนำมาประยุกต์ใช้กับอุปกรณ์ต่างๆ เริ่มแรกสุดจะเป็นการเขียนโปรแกรมเพื่ออ่านค่าจีพีเอสโมดูล เนื่องจากเป็นหัวใจหลักในการทำโครงการนี้ ซึ่งเมื่อสามารถดึงข้อมูลออกมาได้แล้ว ก็นำโปรแกรมมาพัฒนาให้ดียิ่งขึ้นโดยการเขียนฟังก์ชันเพิ่มเติม โดยให้นำเฉพาะข้อมูลที่ต้องการออกมา หลังจากนั้นจึงทดลองกับ Servo Motor และ DC Motor เพื่อให้รถสามารถเคลื่อนตัวได้

3.4.4 การทดลองวิ่งวงที่หนึ่ง ในช่วงที่หนึ่งนี้อุปกรณ์ที่มีบนตัวรถ ได้แก่ จีพีเอสโมดูล ไมโครคอนโทรลเลอร์ แบตเตอรี่ Servo Motor และ DC Motor โดยใช้หลักการการทำงานร่วมกันในบทที่ 2 ซึ่งใช้จีพีเอสโมดูลในการบอกทิศทางรถที่หันหน้าอยู่ กล่าวคือจะให้รถเคลื่อนที่ไปในแนวตรงเสียก่อนเพื่อให้ทราบถึงทิศทางที่รถกำลังหันหน้าอยู่ แล้วใช้ Servo Motor ในการบังคับให้รถพยายามวิ่งไปสู่เป้าหมาย ในขั้นตอนการทดลองนี้เป็นการให้รถวิ่งจากจุดที่หนึ่งไปยังจุดที่สองเท่านั้น ซึ่งสามารถวิ่งได้ในระดับหนึ่ง แต่ก็ยังไม่ดีเพียงพอที่จะนำไปใช้จริงในขั้นต่อไป เนื่องจากจีพีเอสโมดูลไม่เหมาะแก่การนำมาเป็นเซนเซอร์ที่บ่งบอกทิศทางที่รถหันหน้าไป

3.4.5 การพัฒนาโครงการครั้งที่หนึ่ง ขั้นตอนนี้เป็นการศึกษาหาทางแก้ไขปัญหาของการทำหน้าที่ของจีพีเอสโมดูลที่ไม่ดีเพียงพอในการจะใช้เป็นเข็มทิศของรถ ซึ่งได้ข้อสรุปคือทำการซื้อ Compass Module มาติดตั้งเพิ่ม และเปลี่ยนโปรแกรมจากเดิมที่ใช้จีพีเอสโมดูลในการบอกตำแหน่งและทิศทางของรถที่หันไป เป็นใช้จีพีเอสโมดูลในการบอกพิกัดตำแหน่งเพียงอย่างเดียว และนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

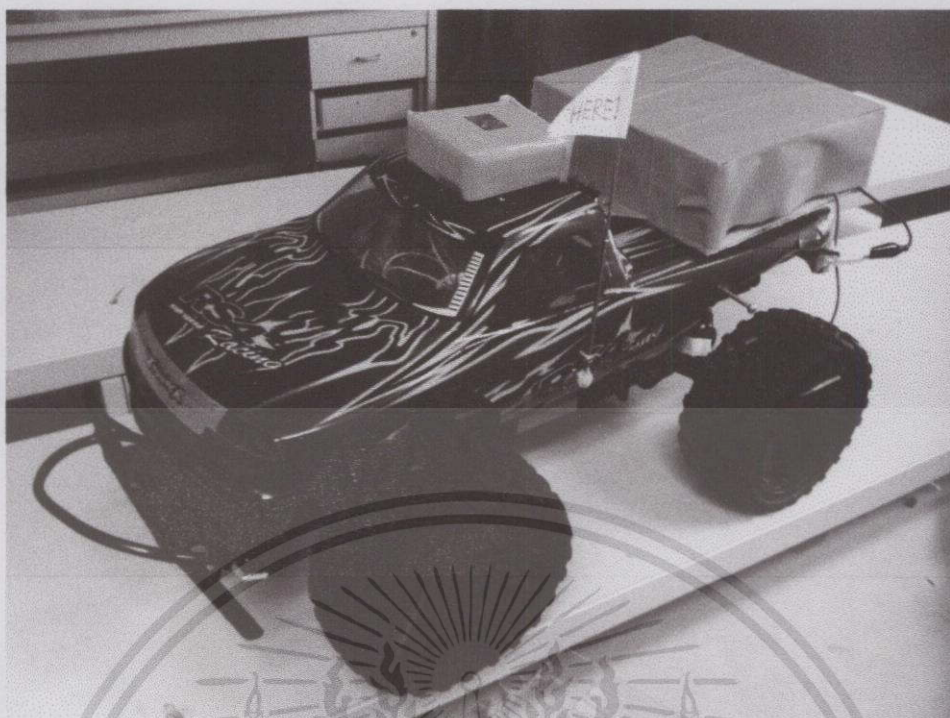
Compass Module มาแทนที่ส่วนในการบ่งบอกทิศทางของรถ เพื่อให้รถสามารถประมวลผลการทำงานได้ไวยิ่งขึ้น

3.4.6 การทดลองวิ่งช่วงที่สอง ในขั้นตอนนี้จะคล้ายคลึงกับขั้นตอน 3.4.4 แต่จะเพิ่มเส้นทางในการวิ่งจากเดิมให้จุดที่หนึ่งไปจุดที่สอง เป็นจุดที่หนึ่งไปจุดที่สอง และเมื่อถึงที่หมายแล้วจะวิ่งไปจุดสามอย่างอัตโนมัติ ซึ่งจากการทดลองพบว่าสามารถวิ่งได้ดี แต่ก็ยังมีปัญหาในเรื่องของความจุของแบตเตอรี่ไม่เพียงพอ ซึ่งในช่วงนี้เกิดปัญหาขึ้นอีกอย่างหนึ่งก็คือบอร์ดไมโครคอนโทรลเลอร์เกิดไหม้ เนื่องจากการจ่ายไฟผิดพลาด จึงจำเป็นต้องเปลี่ยนตัวบอร์ดใหม่ และทำการติดตั้งไดโอดเพิ่มเติมเพื่อป้องกันไม่ให้เกิดปัญหาขึ้นอีก

3.4.7 การปรับแต่งรถ ขั้นตอนนี้เป็นการแก้ไขปัญหาในเรื่องของสายไฟหลวม และปัญหาเมื่อเกิดฝนตกหรือเกิดน้ำท่วมขัง เนื่องจากก่อนหน้านี้ตัวอุปกรณ์จะติดตั้งอยู่บนตัวรถซึ่งเป็นปัญหาเมื่อเกิดเหตุการณ์ดังกล่าว รวมทั้งแก้ไขปัญหาลายไฟหลวม ขั้นตอนนี้จึงมีการทำแผ่นกัสปรีน โดยออกแบบผ่านโปรแกรมเอง และกัสปรีนออกมาแล้วนำมาประกอบใหม่ รวมทั้งทำสายไฟให้แน่นหนาด้วย นอกจากนี้ยังทำการซื้อแบตเตอรี่ใหม่ ซึ่งพบว่าค่าความต่างศักย์ไฟฟ้าของแบตเตอรี่มีค่ามากกว่าที่ระบุไว้ จึงได้ทดลองต่างๆ เพื่อลดค่าต่างศักย์ไฟฟ้านั้น โดยใช้กฎแบ่งแรงดันซึ่งก็พบว่าเป็นไปได้ เนื่องจากโหลดมีค่าความต้านทานน้อยและกินกระแสมาก จึงทำให้ตัวต้านทานที่ทำมาต่อเกิดการไหม้ จึงเปลี่ยนมาใช้การต่อไดโอดเพิ่มจากเดิม 1 ตัว ให้เป็น 4 ตัว ซึ่งก็สามารถทำงานได้ดี

3.4.8 การพัฒนาส่วนของการแสดงผล เนื่องจากก่อนหน้านี้จำเป็นต้องต่อสายเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ เพื่อที่จะสามารถอ่านค่าต่างๆ ของอุปกรณ์ระหว่างที่รถวิ่ง ซึ่งมีความยากลำบาก จึงได้ติดตั้งส่วนของ Wireless Transceiver Module เพิ่มเข้าไป ซึ่งสามารถทำงานได้ดีมีความคล่องตัวมากยิ่งขึ้น

3.4.9 การทดลองวิ่งช่วงที่สาม การทดลองช่วงนี้จะเป็นการให้รถสามารถวิ่งจากจุดใดก็ได้ไปยังจุดที่ต้องการ ซึ่งกำหนดให้ตัวไมโครคอนโทรลเลอร์อ่านค่าพิกัดเริ่มแรก เพื่อให้ทราบถึงจุดที่รถอยู่ และเมื่อกดสวิทช์บนตัวรถเพื่อเป็นการป้อนอินพุตให้ทราบถึงเป้าหมายที่ต้องวิ่งไป ตัวไมโครคอนโทรลเลอร์ก็จะสร้างเส้นทางในการวิ่งซึ่งได้เขียนไว้ภายในโปรแกรม และสั่งการให้อุปกรณ์ต่างๆ ทำงานร่วมกันเพื่อวิ่งไปยังเป้าหมายที่ต้องไป



รูปที่ 3.1 รถที่เสร็จสมบูรณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

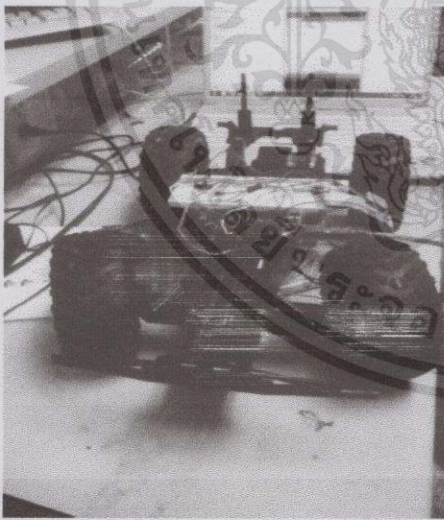
บทที่ 4

การทดลอง

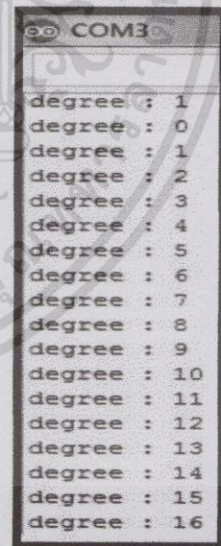
ในบทนี้จะกล่าวถึงการทดลองและผลการทดลอง และการทำงานของรถขับเคลื่อนอัตโนมัติ ด้วยระบบจีพีเอส โดยมีการทดลองดังรายละเอียดดังต่อไปนี้

4.1 การทดลองของ Servo motor

ทำการทดลอง Servo Motor โดยการสั่งให้ Motor เลี้ยวซ้ายสุดและเลี้ยวขวาสุด ซึ่งจากการทดลองนั้น เมื่อล้อรถอยู่ตรงกลางนั้น มุมที่อ่านได้จาก Motor คือ 120 องศา ต่อมาเมื่อรถเลี้ยวซ้ายสุด มุมที่อ่านค่าได้จาก Motor คือ 40 องศา และสุดท้ายเมื่อรถเลี้ยวขวาสุด มุมที่อ่านได้คือ 180 องศา จากการทดลองนั้นสรุปได้ว่า เราสามารถสั่งให้ล้อของรถนั้นเลี้ยวได้มุมที่เราต้องการกำหนดไว้ และ Servo Motor ที่วัสดุเป็นเหล็กนั้นทำงานได้ดีกับบอร์ด Arduino Mega 2560 R3 และทนทานต่อความร้อนและการเสียดสี

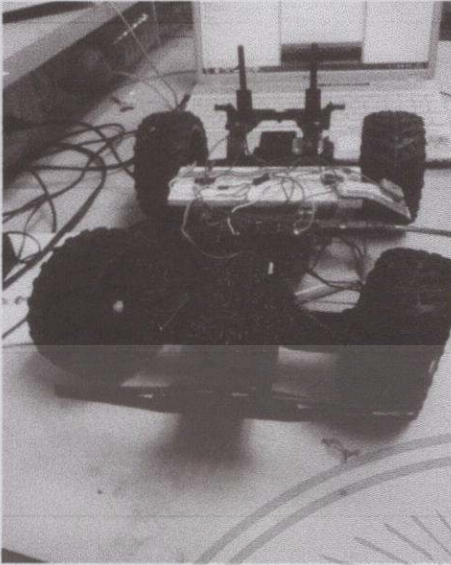


รูปที่ 4.1 แสดงการเลี้ยวซ้าย



angle
degree : 1
degree : 0
degree : 1
degree : 2
degree : 3
degree : 4
degree : 5
degree : 6
degree : 7
degree : 8
degree : 9
degree : 10
degree : 11
degree : 12
degree : 13
degree : 14
degree : 15
degree : 16

รูปที่ 4.2 แสดงมุมจาก servo เมื่อเลี้ยวซ้าย



รูปที่ 4.3 แสดงการเลี้ยวขวา

```

degree : 180
degree : 179
degree : 178
degree : 177
degree : 176
degree : 175
degree : 174
degree : 173
degree : 172
degree : 171
degree : 170
degree : 169
degree : 168
degree : 167
degree : 166
degree : 165

```

รูปที่ 4.4 แสดงมุมจาก servo เมื่อเลี้ยวขวา

4.2 การทดลองสัญญาณเข็มทิศ

Raw:	-506	10	5	Scaled:	-465.52	9.20	4.60	Heading:	3.12 Radians	178.87 Degrees
Raw:	-506	8	6	Scaled:	-465.52	7.36	5.52	Heading:	3.13 Radians	179.09 Degrees
Raw:	-503	9	5	Scaled:	-462.76	8.28	4.60	Heading:	3.12 Radians	178.97 Degrees
Raw:	-503	8	6	Scaled:	-467.36	7.36	5.52	Heading:	3.13 Radians	179.10 Degrees
Raw:	-511	3	5	Scaled:	-470.12	2.76	4.60	Heading:	3.14 Radians	179.86 Degrees
Raw:	-505	3	6	Scaled:	-464.60	2.76	5.52	Heading:	3.14 Radians	179.86 Degrees
Raw:	-512	0	10	Scaled:	-471.04	0.00	9.20	Heading:	3.14 Radians	180.00 Degrees
Raw:	-507	1	2	Scaled:	-466.44	0.92	1.94	Heading:	3.14 Radians	179.89 Degrees
Raw:	-512	-9	4	Scaled:	-471.04	-8.28	3.68	Heading:	3.16 Radians	181.01 Degrees
Raw:	-514	-8	6	Scaled:	-472.88	-7.36	5.52	Heading:	3.16 Radians	180.89 Degrees
Raw:	-515	-12	7	Scaled:	-473.80	-11.04	6.44	Heading:	3.16 Radians	181.33 Degrees
Raw:	-518	-12	10	Scaled:	-476.36	-11.04	9.20	Heading:	3.16 Radians	181.33 Degrees
Raw:	-507	-8	3	Scaled:	-466.44	-7.36	2.76	Heading:	3.16 Radians	180.90 Degrees
Raw:	-513	-14	8	Scaled:	-477.48	-13.80	8.28	Heading:	3.17 Radians	181.56 Degrees
Raw:	-515	-12	8	Scaled:	-473.80	-11.04	7.36	Heading:	3.16 Radians	181.33 Degrees
Raw:	-512	-14	6	Scaled:	-471.04	-12.88	5.52	Heading:	3.17 Radians	181.57 Degrees
Raw:	-515	-12	8	Scaled:	-473.80	-11.04	7.36	Heading:	3.16 Radians	181.33 Degrees
Raw:	-509	-8	3	Scaled:	-468.28	-7.36	2.76	Heading:	3.16 Radians	180.90 Degrees
Raw:	-516	-6	7	Scaled:	-474.72	-5.52	6.44	Heading:	3.15 Radians	180.67 Degrees
Raw:	-511	-3	1	Scaled:	-470.12	-2.76	0.92	Heading:	3.15 Radians	180.34 Degrees

รูปที่ 4.5 แสดงเข็มทิศชี้ไปทางทิศใต้

ภาพนี้เป็นภาพที่ได้จากการทดลองเข็มทิศ โดยที่ภาพได้แสดงข้อมูลที่ 180 Degrees นั้น หมายถึงเข็มทิศได้ชี้ไปทางทิศใต้ แต่ถ้าภาพแสดงแสดงข้อมูลที่ 0 Degrees นั้น หมายถึงเข็มทิศได้ชี้ไปทางทิศเหนือ ซึ่งผลปรากฏว่าได้ให้ค่าที่ใกล้เคียงกับความเป็นจริงจึงสามารถนำไปทำการทดลองต่อไปได้ใน Autonomous car with GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลอง GPS Module

```

$GPGGA,152142.00,1343.74679,N,10046.56056,E,1,07,1.33,1.6,M,-27.0,M,,*79
$GPGSA,A,3,24,25,42,29,21,50,15,,,,,2.93,1.33,2.61*00
$GPGSV,2,1,08,15,29,027,35,21,65,329,35,22,15,309,,24,56,078,49*79
$GPGSV,2,2,08,25,09,173,16,29,27,194,34,42,42,106,43,50,37,103,42*7D
$GPGLL,1343.74679,N,10046.56056,E,152142.00,A,A*65
$GPRMC,152143.00,A,1343.74685,N,10046.56038,E,0.261,,100913,,,A*79
$GPVTG,,T,,M,0.261,N,0.484,K,A*2E

```

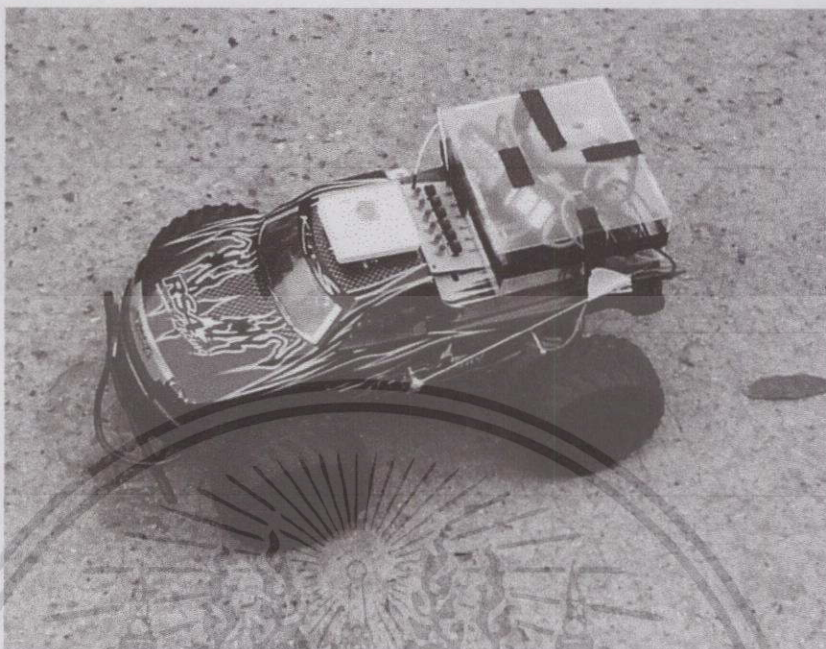
รูปที่ 4.6 ข้อมูลจาก GPS Module

ทำการรับข้อมูลทั้งหมดจาก GPS Module ผ่านพอร์ต Serial ณ บริเวณ หน้าตึก ECC ทำให้ได้ข้อมูลดังรูปที่ 4.6 ซึ่งสามารถนำมาเปรียบเทียบค่าต่างๆ ในส่วนของบรรทัด \$GPRMC ได้ดังนี้

152143	กำหนดการกระทำที่เวลา 15:21:43 UTC
A	สถานะ A = ทำงาน หรือ V = เฉย
1343.74685,N	ละติจูด 13 องศาเหนือ 43.74685 ลิปดา
10046.56038,E	ลองจิจูด 100 องศาตะวันออก 46.56038 ลิปดา
0.261	ความเร็วบนพื้นโลก (knots)
100913	วันที่ 10 เดือน 9 กันยายน ปี ค.ศ. 2013
*79	*Check sum

สามารถนำค่าละติจูด และลองจิจูด มาแปลงหน่วยให้อยู่ในรูปแบบ องศาศนิยมโดยการ นำลิปตามหาร 60 แล้วเอาค่าที่ได้ทั้งหมดบวกกับองศาเดิมทำให้ได้ค่า ละติจูด = 13.729114 องศาเหนือ และลองจิจูด = 100.776006 ซึ่งนำมาเปรียบเทียบกับ <https://maps.google.co.th/> พบว่าพิกัดนั้นถูกต้อง

4.4 การทดลอง DC Motor



รูปที่ 4.7 การทดลอง DC Motor

เมื่อทำการป้อนโปรแกรมทดลองวิ่งเข้าสู่ Microcontroller จะทำให้ DC Motor หรือล้อหลังหลังเคลื่อนที่ไปข้างหน้าในแนวตรง โดยสามารถกำหนดความเร็วของรถได้ผ่านการใช้หลักการของ PWM หรือ Pulse Width Modulation เพื่อเปลี่ยนแปลงแรงดันที่ตกคร่อม DC Motor ซึ่งความเร็วสูงสุดขึ้นอยู่กับแรงบิดของมอเตอร์ และแรงดันไฟฟ้าจากแบตเตอรี่

4.5 การทดลอง Wireless Transceiver Module



รูปที่ 4.8 ลักษณะการใช้งาน Wireless Module

```

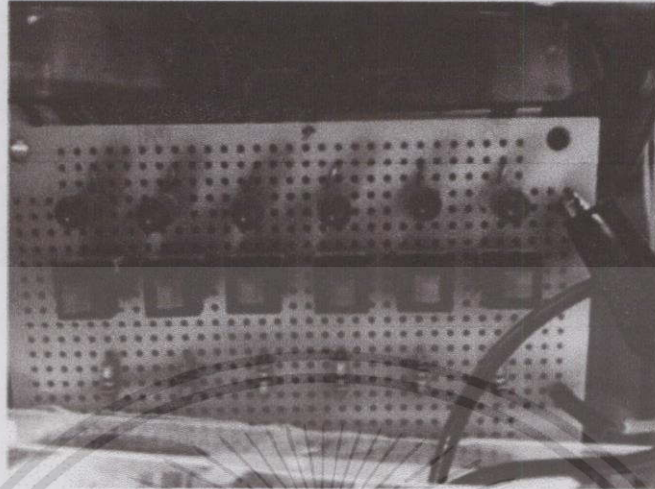
Got data
-----
Lat: 13729250
Lon: 100777740
headingWaiting for data...
Got data
-----
Lat: 13729250
Lon: 100777776
headingWaiting for data...
Got data
-----
Lat: 13729250
Lon: 100777820
headingWaiting for data...
Got data
-----
Lat: 13729250
Lon: 100777878
headingWaiting for data...
Got data

```

รูปที่ 4.9 ข้อมูลที่ได้จากรถผ่าน Wireless Module

เมื่อทดลองต่อ Wireless Module เข้ากับ Microcontroller อีกหนึ่งตัว และต่อเข้ากับคอมพิวเตอร์ เมื่อทำการลงโปรแกรมให้เชื่อมต่อกับ Wireless Module ที่อยู่บนรถแล้ว Wireless Module ที่คอมพิวเตอร์สามารถรับค่าที่ได้จากอุปกรณ์ต่างๆ ที่อยู่ในตัวรถได้ ดังในรูปที่ 4.9 เป็นตัวอย่างของการรับข้อมูลจากอุปกรณ์ในตัวรถ ซึ่งในที่นี้ก็คือ GPS Module นั่นเอง

4.6 การทดลองสวิตช์ในการสั่งการวิ่ง



รูปที่ 4.10 ลักษณะสวิตช์ที่ใช้งาน

ในการทดลองนี้จะทำการทดลองสวิตช์ทั้ง 6 บุ่ม ที่ได้สร้างขึ้น เพื่อเป็นตัวส่งสัญญาณเข้าสู่ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้รถไปสู่จุดหมายที่ต้องการ โดยเรากำหนดตัวแปรต่างๆ ไว้ดังนี้

สวิตช์ตัวที่ 1 (ซ้ายสุด) กำหนดให้จุดหมายคือ หอพักนักศึกษาในสถาบัน

สวิตช์ตัวที่ 2 กำหนดให้จุดหมายคือ โรงอาหาร L

สวิตช์ตัวที่ 3 กำหนดให้จุดหมายคือ ตึกปฏิบัติการ ECC

สวิตช์ตัวที่ 4 กำหนดให้จุดหมายคือ ที่ทำการไปรษณีย์ประจำสถาบัน

สวิตช์ตัวที่ 5 กำหนดให้จุดหมายคือ ที่ทำการธนาคารภายในสถาบัน

สวิตช์ตัวที่ 6 กำหนดหน้าที่คือ รีเซ็ตค่าเพื่อรอการสั่งการ

Lat: 13729284 Lon: 100775941 Status: 1 Route: 1 last goal: 4 going: 6 r1: 0 Waiting for data... Got data	Lat: 13729214 Lon: 100775965 Status: 1 Route: 1 last goal: 2 going: 8 r1: 0 Waiting for data... Got data	Lat: 13729283 Lon: 100776030 Status: 1 Route: 1 last goal: 3 going: 3 r1: 0 Waiting for data... Got data
Lat: 13729284 Lon: 100775941 Status: 1 Route: 1 last goal: 4 going: 6 r1: 0 Waiting for data... Got data	Lat: 13729214 Lon: 100775965 Status: 1 Route: 1 last goal: 2 going: 8 r1: 0 Waiting for data... Got data	Lat: 13729283 Lon: 100776030 Status: 1 Route: 1 last goal: 3 going: 3 r1: 0 Waiting for data... Got data

รูปที่ 4.11 การทดลองสวิตช์ที่ 1 ถึง 3

Lat: 13729284 Lon: 100775941 Status: 1 Route: 1 last goal: 4 going: 6 r1: 0 Waiting for data... Got data	Lat: 13729302 Lon: 100775949 Status: 1 Route: 1 last goal: 5 going: 8 r1: 0 Waiting for data... Got data
Lat: 13729284 Lon: 100775941 Status: 1 Route: 1 last goal: 4 going: 6 r1: 0 Waiting for data... Got data	Lat: 13729302 Lon: 100775949 Status: 1 Route: 1 last goal: 5 going: 8 r1: 0 Waiting for data... Got data

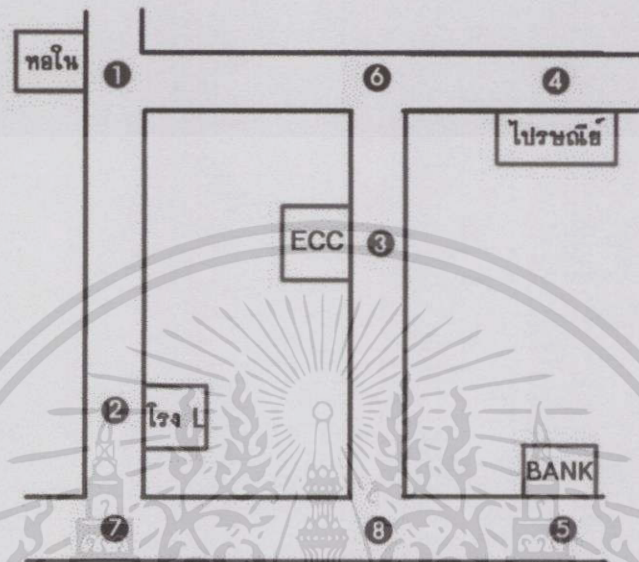
รูปที่ 4.12 การทดลองสวิตช์ที่ 4 และ 5

ค่าที่ได้เป็นการแสดงสถานะต่างๆ เมื่อทำการกดสวิตช์ จากรูปที่ 4.11 และ 4.12 จะเป็นการแสดงผลการกดสวิตช์เบอร์ 1 ถึง 5 ตามลำดับ ซึ่งค่าต่างๆ มีรายละเอียดดังนี้

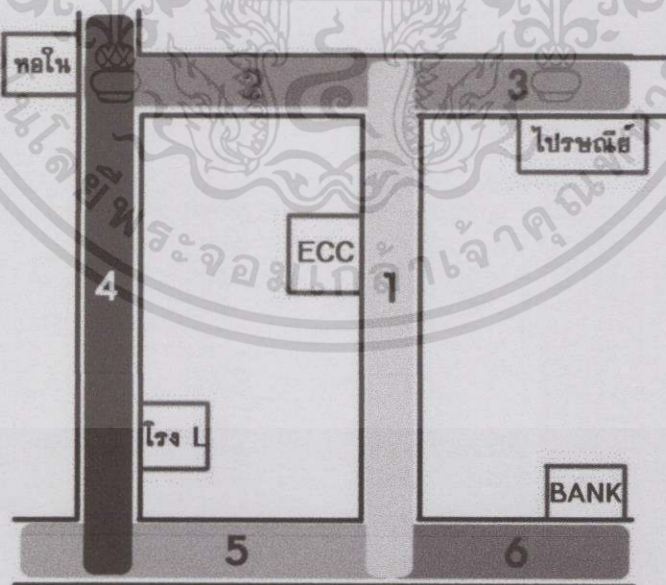
- Lat หมายถึง ค่าละติจูดของตำแหน่งที่รถอยู่
- Lon หมายถึง ค่าลองจิจูดของตำแหน่งที่รถอยู่
- Status เป็นการแสดงสถานะของรถ โดยมี 3 สถานะ คือ
 - 0 คือ กำลังรออินพุตจากสวิตช์
 - 1 คือ กำลังเดินทาง
 - 2 คือ รถถึงที่หมายปลายทาง
- Route คือ เส้นถนนที่รถยืนอยู่ ณ ตอนมี Status เท่ากับ 0 ดูได้จากรูปที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- last goal คือตำแหน่งปลายทางที่รถต้องการไป ดังรูปที่ 4.13
- going คือ ตำแหน่งที่รถกำลังจะเดินทางไป ดังรูปที่ 4.13
- r1 คือ สเต็ปของการวิ่งตอนรถ เริ่มต้นจาก 0



รูปที่ 4.13 ตำแหน่งต่างๆ



รูปที่ 4.14 ถนนภายในสถาบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากการทดลองรถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอส เมื่อนำรถไปตั้งกลางที่แจ้งที่เป็นจุดเริ่มต้นของรถ แล้วเสียบสายเข้ากับแหล่งจ่ายไฟ รถจะรอสัญญาณจีพีเอสสักพัก พอตัวรถรับสัญญาณได้แล้วให้กดสวิตซ์ในที่ๆ ต้องการให้รถไป ผลสรุปออกมาว่ารถวิ่งไปตามเป้าหมายจริงพร้อมกับซูดงขึ้นจริง แต่ก็ยังมีความผิดพลาดของตำแหน่งที่รถจอดเพียงเล็กน้อย เนื่องมาจากสัญญาณจีพีเอสมีค่าที่เปลี่ยนแปลงตลอดเวลาไม่คงที่ ดังนั้นเมื่อรถหยุดที่ตำแหน่งที่เป็นเป้าหมายแล้ว แต่ก็ยังคงมีความคาดเคลื่อนอยู่ดี แต่ถึงแม้ว่าจีพีเอสไม่ดูจะรับสัญญาณได้ค่าใหม่ รถก็จะไม่เคลื่อนที่ต่อเพราะรถได้ถูกตั้งโปรแกรมให้หยุด และตัดวงจรทั้งระบบทันทีที่รถได้หยุดตรงเป้าหมาย

5.2 ปัญหาที่พบและแนวทางการแก้ไข

จากการศึกษาและทำโครงการนี้ในช่วงแรกๆ คือ ปัญหาเกี่ยวกับสัญญาณจีพีเอส ซึ่งตัว GPS Module ไม่สามารถรับค่าจากสัญญาณดาวเทียม ซึ่งทำการแก้ไขโดยการปรับเปลี่ยนโปรแกรมของตัว GPS Module ไปเรื่อยๆ จน GPS Module สามารถรับสัญญาณได้ ปัญหาต่อมาคือเรื่องของเวลาที่ตัวจีพีเอสไม่ดูจะรับสัญญาณช้า พอรับสัญญาณมารถก็จะวิ่งไปสักพักก่อนเพื่อจับทิศทางทิศทางการที่กำลังหันไป ซึ่งพบว่าทำได้ช้าเกินไป จึงได้แก้ปัญหาโดยการใช้ Compass Module เข้ามาช่วยแก้ไข จึงทำให้จับทิศทางได้เร็วยิ่งขึ้น และอีกปัญหาเกี่ยวกับเข็มทิศก็คือ ต้องรู้ว่าควรระวังตัว Compass Module นั้นไว้ตรงไหนของรถ นั่นก็คือวางไว้ให้ห่างจากตัวมอเตอร์เพราะในตัวมอเตอร์นั้นจะมีเรื่องของสนามแม่เหล็กอยู่ หากวางไว้ใกล้กันจะเกิดการแทรกแซงของสนามแม่เหล็กซึ่งกันและกัน และก่อให้เกิดข้อมูลที่ผิดพลาดของ Compass Module ต่อมาคือเรื่องปัญหาของตัวบอร์ดที่ใหม่ อันเนื่องมาจากการต่อไฟผิดขั้ว ซึ่งปัญหานี้สามารถแก้ไขโดยการใส่ไดโอด เพราะไดโอดทำหน้าที่ในการป้องกันกระแสไหลผิดทาง จึงสามารถป้องกันปัญหาการต่อไฟฟ้าผิดขั้วได้ ปัญหาต่อมาคือเรื่องของความจุแบตเตอรี่ไม่เพียงพอ เนื่องมาจากการมีโหลดที่เพิ่มมากขึ้น จึงทำให้แบตเตอรี่จ่ายไฟให้ไม่พอ ซึ่งทำการแก้ไขโดยการซื้อแบตเตอรี่ใหม่ ปัญหาสุดท้ายคือการที่ซื้อแบตเตอรี่ตามที่ต้องการไม่ได้จึงทำให้ต้องซื้อที่มีขนาดโวลต์ที่มากเกินไปขนาดที่ตัวอุปกรณ์จะรับได้ ซึ่งทำการแก้ไขโดยการต่อไดโอด 4 ตัวเพื่อลดจำนวนโวลต์ของตัวแบตเตอรี่ลง แต่ปัญหาที่พบหลักๆ โดยทั่วไปมักจะเป็นเรื่องเกี่ยวกับโปรแกรมซึ่งไม่เป็นไปตามที่เราต้องการ และความคลาดเคลื่อนของสัญญาณจีพีเอส ทั้งนี้สามารถแก้ไขโดยการทดลองการวิ่งซ้ำๆ และปรับปรุงโปรแกรมไปพร้อมกัน

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

ตัว Module ในปัจจุบันนี้มีอยู่หลายรุ่นที่เราสามารถเลือกใช้ได้ ควรที่จะศึกษาเพิ่มเติมว่ามี Module ตัวไหนที่เหมาะสมแก่การใช้งานในโครงการนี้ที่จะให้ผลที่ออกมาได้ดีที่สุด ซึ่งก็ยังคงพบว่ามีบางรุ่นที่มีราคาสูงนั้นจะให้ประสิทธิภาพที่ดีกว่าตัวที่ใช้อยู่ปัจจุบัน ซึ่งในตัวที่ใช้อยู่ปัจจุบันนั้นอย่างเช่น GPS Module นั้นรับสัญญาณได้ช้าและมีความเที่ยงตรงค่อนข้างน้อย ทำให้การเคลื่อนที่ของรถนั้นเป็นไปอย่างไม่ราบรื่น

จากปัญหาข้างต้นที่ได้กล่าวนั้นหากเราได้มีความรู้ความเข้าใจเกี่ยวกับการเลือกตัวอุปกรณ์ และใช้ตัวอุปกรณ์ที่มีคุณภาพมากกว่านี้ ก็จะทำให้รถขับเคลื่อนอัตโนมัติด้วยระบบจีพีเอสเคลื่อนที่ได้อย่างมีประสิทธิภาพมากขึ้น จนกลายเป็นรถต้นแบบในที่สุด



เอกสารอ้างอิง

[1] DC Motor

http://www.tdhobby.com/index.php?option=com_content&view=article&id=89:understand-rc-servo&catid=43:2011-01-30-11-45-16&Itemid=79

[2] Servo Motor

<http://www.smartlearningweb.com/knowledge/motor/motor.htm>

[3] IC L293

<http://www.alldatasheet.com/view.jsp?Searchword=L293%20datasheet>

[4] จีพีเอส

<http://th.wikipedia.org/wiki/%E0%B8%88%E0%B8%B5%E0%B8%9E%E0%B8%B5%E0%B9%80%E0%B8%AD%E0%B8%AA>

[5] NMEA Sentence <http://www.oknation.net/blog/print.php?id=555838>

[6] Microcontroller

<http://th.wikipedia.org/wiki/%E0%B9%84%E0%B8%A1%E0%B9%82%E0%B8%84%E0%B8%A3%E0%B8%84%E0%B8%AD%E0%B8%99%E0%B9%82%E0%B8%97%E0%B8%A3%E0%B8%A5%E0%B9%80%E0%B8%A5%E0%B8%AD%E0%B8%A3%E0%B9%8C>

[7] Arduino Mega2560 R3 <http://arduino.cc/en/Main/arduinoBoardMega2560>

[8] สนามแม่เหล็กโลก

<http://th.wikipedia.org/wiki/%E0%B8%AA%E0%B8%99%E0%B8%B2%E0%B8%A1%E0%B9%81%E0%B8%A1%E0%B9%88%E0%B9%80%E0%B8%AB%E0%B8%A5%E0%B9%87%E0%B8%81%E0%B8%82%E0%B8%AD%E0%B8%87%E0%B9%82%E0%B8%A5%E0%B8%81>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง(ต่อ)

- [9] Compass Module (HMC5883L)
http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf
- [10] การเชื่อมต่อแบบ I2C <http://www.thaimicrotron.com/CCS-628/Reference/I2CBUS.htm>
- [11] GPS Module (Ublox NEO-6M)
[https://www.u-blox.com/images/downloads/Product_Docs/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/images/downloads/Product_Docs/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf)
- [12] คำสั่งเขียนโปรแกรม http://arduino.cc/en/Reference/HomePage#.UwYbx_l_tec
- [13] nRF905 library <http://blog.zakkemble.co.uk/nrf905-avrarduino-librarydriver/>
- [14] nRF905 Datasheet
http://mobiledevices.kom.aau.dk/fileadmin/mobiledevices/opensensor/literature/NRF905_data_sheet.pdf
- [15] HMC5883L Library http://bildr.org/2012/02/hmc5883l_arduino/
- [16] Tiny GPS <http://arduiniiana.org/libraries/tinygps/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมการประมวลผล

โปรแกรม Visual C++ เป็นโปรแกรมที่ใช้ในการตั้งคำสั่งตัวรถผ่านทางตัว Mega Arduino ซึ่งสามารถแบ่งโค้ดโปรแกรมได้ดังต่อไปนี้

ก.1 โค้ดแสดงการรับค่า GPS 2 ค่า

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>

long lat,lon; // create variable for latitude and longitude object
double flat,flon;
TinyGPS gps; // create gps object

void setup(){
  Serial.begin(38400); // connect serial
  Serial2.begin(9600); // connect gps sensor
}

void loop()
{
  while(Serial2.available())
  {
    if(gps.encode(Serial2.read()))
    {
      getCurrentPoint(gps);
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void getCurrentPoint(TinyGPS &gps)
{
  //gps.f_get_position(&flat, &flon);
  gps.get_position(&lat,&lon);
  //gps.get_datetime(&date, &time, &age);
  Serial.print("lat: ");Serial.print(lat);Serial.print(" ");// print latitude
  Serial.print("lon: ");Serial.println(lon); // print longitude
}

```

ก.2 โค้ดแสดงการรับค่าทุกค่า

```

void setup()
{
  Serial.begin(38400);
  Serial2.begin(9600);
}

void loop() {
  if (Serial2.available())
  {
    int inByte = Serial2.read();
    Serial.write(inByte);
  }
}

```

ก.3 โค้ดการขับ DC Motor

```

int IN1 = 4;
int IN2 = 5;
int EN1 = 6;
int EN2 = 7;
int IN3 = 8;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int IN4 = 9;
```

```
int Speed = 255;
```

```
void setup()
```

```
{
```

```
  pinMode(IN1,OUTPUT);
```

```
  pinMode(IN2,OUTPUT);
```

```
  pinMode(IN3,OUTPUT);
```

```
  pinMode(IN4,OUTPUT);
```

```
  pinMode(EN1,OUTPUT);
```

```
  pinMode(EN2,OUTPUT);
```

```
  digitalWrite(IN1,LOW);
```

```
  digitalWrite(IN2,HIGH);
```

```
  digitalWrite(IN3,HIGH);
```

```
  digitalWrite(IN4,LOW);
```

```
  Speed = 255;
```

```
  Serial.begin(19200);
```

```
}
```

```
void loop()
```

```
{
```

```
  analogWrite(EN1,Speed);
```

```
  analogWrite(EN2,Speed);
```

```
}
```

ก.4 โค้ดการขับ Servo Motor

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
                // a maximum of eight servo objects can be created

int pos = 0; // variable to store the servo position

void setup()
{
  Serial.begin(9600);
  myservo.attach(12); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1) // goes from 0 degrees to 180 degrees
  {
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    Serial.print("degree : ");
    Serial.println(pos);
    delay(50); // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos >= 1; pos -= 1) // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    Serial.print("degree : ");
    Serial.println(pos);
    delay(50); // waits 15ms for the servo to reach the position
  }
}
}
```

ก.5 โค้ดแสดงเข็มทิศ

```

#include <Wire.h>
#include <HMC5883L.h>
#include <math.h>

HMC5883L compass;

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  Serial.println("Serial started.");

  compass = HMC5883L();

  Serial.println("Setting scale to +/- 1.3 Ga");
  int error = compass.SetScale(1.3); // Set the scale of the compass.
  if(error != 0) // If there is an error, print it out.
    Serial.println(compass.GetErrorText(error));

  Serial.println("Setting measurement mode to continuous.");
  error = compass.SetMeasurementMode(Measurement_Continuous); // Set the
measurement mode to Continuous
  if(error != 0) // If there is an error, print it out.
    Serial.println(compass.GetErrorText(error));
}

void loop()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    // Retrieve the raw values from the compass (not scaled).
    MagnetometerRaw raw = compass.ReadRawAxis();
    // Retrieved the scaled values from the compass (scaled to the configured scale).
    MagnetometerScaled scaled = compass.ReadScaledAxis();

    // Calculate heading when the magnetometer is level, then correct for signs of axis.
    float heading = atan2(raw.YAxis, raw.XAxis);

    // Correct for when signs are reversed.
    if(heading < 0)
        heading += 2*PI;

    // Convert radians to degrees for readability.
    float headingDegrees = degrees(heading);

    // Output the data via the serial port.
    Output(raw, scaled, heading, headingDegrees);
}

```

```

void Output(MagnetometerRaw raw, MagnetometerScaled scaled, float heading, float
headingDegrees)

```

```

{
    Serial.print("Raw:\t");
    Serial.print(raw.XAxis);
    Serial.print(" ");
    Serial.print(raw.YAxis);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print(" ");
Serial.print(raw.ZAxis);
Serial.print(" \tScaled:\t");

Serial.print(scaled.XAxis);
Serial.print(" ");
Serial.print(scaled.YAxis);
Serial.print(" ");
Serial.print(scaled.ZAxis);

Serial.print(" \tHeading:\t");
Serial.print(heading);
Serial.print(" Radians \t");
Serial.print(headingDegrees);
Serial.print("\n" Degrees \t");
}

```

ก.6 โค้ดแสดงสวิทช์

```

int inPin1 = 24; // the number of the input pin
int inPin2 = 28;
int inPin3 = 32;
int inPin4 = 36;
int inPin5 = 40;
int inPin6 = 44;

int outPin = 13; // the number of the output pin

int reading1,reading2,reading3,reading4,reading5,reading6; // the current
reading from the input pin

void setup()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  pinMode(inPin1, INPUT);
  pinMode(inPin2, INPUT);
  pinMode(inPin3, INPUT);
  pinMode(inPin4, INPUT);
  pinMode(inPin5, INPUT);
  pinMode(inPin6, INPUT);

  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  reading1 = digitalRead(inPin1);
  reading2 = digitalRead(inPin2);
  reading3 = digitalRead(inPin3);
  reading4 = digitalRead(inPin4);
  reading5 = digitalRead(inPin5);
  reading6 = digitalRead(inPin6);

  if (reading1 == HIGH && reading2 == LOW && reading3 == LOW && reading4 ==
LOW && reading5 == LOW && reading6 == LOW)
    Serial.println("1 = HIGH");
  else if (reading1 == LOW && reading2 == HIGH && reading3 == LOW && reading4
== LOW && reading5 == LOW && reading6 == LOW)
    Serial.println("2 = HIGH");
  else if (reading1 == LOW && reading2 == LOW && reading3 == HIGH && reading4
== LOW && reading5 == LOW && reading6 == LOW)
    Serial.println("3 = HIGH");
  else if (reading1 == LOW && reading2 == LOW && reading3 == LOW && reading4
== HIGH && reading5 == LOW && reading6 == LOW)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("4 = HIGH");
else if (reading1 == LOW && reading2 == LOW && reading3 == LOW && reading4
== LOW && reading5 == HIGH && reading6 == LOW)
    Serial.println("5 = HIGH");
else if (reading1 == LOW && reading2 == LOW && reading3 == LOW && reading4
== LOW && reading5 == LOW && reading6 == HIGH)
    Serial.println("6 = HIGH");
else
    Serial.println("ERROR");

delay(500);
}

```

ก.7 โค้ดลอง Wifi ตัวรับ

```

#include <nRF905.h>
#include <SPI.h>

#define RXADDR 0x586F2E10 // Address of this device (4 bytes / long data type)
#define TXADDR 0xFE4CA6E5 // Address of device to send to (4 bytes / long data
type)

void setup()
{
    // Start up
    nRF905_init();

    // Set address of this device
    nRF905_setRXAddress(RXADDR);

    // Lowest transmit level -10db
    nRF905_setTransmitPower(NRF905_PWR_n10);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Reduce receive sensitivity to save a few mA
nRF905_setLowRxPower(NRF905_LOW_RX_ENABLE);

// Put into receive mode
nRF905_receive();

Serial.begin(38400);

Serial.println("Server started");
}

void loop()
{
  Serial.println("Waiting for data...");

  // Make buffer for data
  byte buffer[NRF905_MAX_PAYLOAD];

  // Wait for data
  while(!nRF905_getData(buffer, sizeof(buffer)));

  // Set address of device to send to
  nRF905_setTXAddress(TXADDR);

  // Set payload data (reply with data received)
  nRF905_setData(buffer, sizeof(buffer));

  // Send payload (send fails if other transmissions are going on, keep trying until
  success)
  while(!nRF905_send());
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Put back into receive mode
nRF905_receive();

Serial.println("Got data");
while(!nRF905_getData(buffer, sizeof(buffer)));

    for (byte i=0;i<sizeof(buffer);i++)
    {
        if (i== 0) Serial.print("\n Lat: ");
        if (i<=7)
        {
            Serial.write(buffer[i]);
        }
        if (i== 8) Serial.print("\n Lon: ");
        if (i >= 8 && i<=16)
        {
            Serial.write(buffer[i]);
        }
        if (i == 17)
        {
            Serial.print("\n heading");
            Serial.write(buffer[i]);
        }
    }
}
}

```

ก.8 ใ้ค้ดลอง Wifi ตัวส่ง

```

#include <nRF905.h>
#include <SPI.h>
#include <stdlib.h>

#define RXADDR 0xFE4CA6E5 // Address of this device (4 bytes / long data type)
#define TXADDR 0x586F2E10 // Address of device to send to (4 bytes / long data
type)
#define TIMEOUT 1000 // 1 second ping timeout

// Just some random data to send
byte data[NRF905_MAX_PAYLOAD] = {
0x0A,
0x68,
0x45,
0xFA
};

long spot0_lat = 13729250;
long spot0_lon = 100776010;
char N = 'n';
char myStr[32];
char *copystr,whatt[33];

void setup()
{

    // Start up
    nRF905_init();

```

```

// Set address of this device
nRF905_setRXAddress(RXADDR);

// Lowest transmit level -10db
nRF905_setTransmitPower(NRF905_PWR_n10);

// Reduce receive sensitivity to save a few mA
nRF905_setLowRxPower(NRF905_LOW_RX_ENABLE);

// Put into receive mode
nRF905_receive();

Serial.begin(38400);

Serial.println("Client started");
}

void loop()
{
// Turn on module
nRF905_powerUp();

// Set address of device to send to
nRF905_setTXAddress(TXADDR);

copystr = ltoa(spot0_lat,whatt,10);
myStr[0] = copystr[0];
myStr[1] = copystr[1];
myStr[2] = copystr[2];
myStr[3] = copystr[3];
myStr[4] = copystr[4];
myStr[5] = copystr[5];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

myStr[6] = copyst[6];
myStr[7] = copyst[7];
copyst = ltoa(spot0_lon,whatt,10);
myStr[8] = copyst[0];
myStr[9] = copyst[1];
myStr[10] = copyst[2];
myStr[11] = copyst[3];
myStr[12] = copyst[4];
myStr[13] = copyst[5];
myStr[14] = copyst[6];
myStr[15] = copyst[7];
myStr[16] = copyst[8];
myStr[17] = N;
spot0_lon++;

// Set payload data
nRF905_setData((byte*)myStr, sizeof(myStr));

// Send payload (send fails if other transmissions are going on, keep trying until
success)
while(!nRF905_send());

// Put into receive mode
nRF905_receive();

// Make buffer for reply
byte buffer[NRF905_MAX_PAYLOAD];
bool success;

```

ก.9 โค้ดจริง Wifi ตัวรับ

```

#include <nRF905.h>
#include <SPI.h>

#define RXADDR 0x586F2E10 // Address of this device (4 bytes / long data type)
#define TXADDR 0xFE4CA6E5 // Address of device to send to (4 bytes / long data
type)

void setup()
{
    // Start up
    nRF905_init();

    // Set address of this device
    nRF905_setRXAddress(RXADDR);

    // Lowest transmit level -10db
    nRF905_setTransmitPower(NRF905_PWR_10);

    // Reduce receive sensitivity to save a few mA
    nRF905_setLowRxPower(NRF905_LOW_RX_DISABLE);

    // Put into receive mode
    nRF905_receive();

    Serial.begin(38400);

    Serial.println("Server started");
}

void loop()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  Serial.println("Waiting for data...");

  // Make buffer for data
  byte buffer[NRF905_MAX_PAYLOAD];

  // Wait for data
  while(!nRF905_getData(buffer, sizeof(buffer)));

  // Set address of device to send to
  nRF905_setTXAddress(TXADDR);

  // Set payload data (reply with data received)
  nRF905_setData(buffer, sizeof(buffer));

  // Send payload (send fails if other transmissions are going on, keep trying until
  success)
  while(!nRF905_send());

  // Put back into receive mode
  nRF905_receive();

  Serial.println("Got data");
  while(!nRF905_getData(buffer, sizeof(buffer)));

  for (byte i=0;i<sizeof(buffer);i++)
  {
    if (i== 0) Serial.print("\n Lat: ");
    if (i<=7)
    {
      Serial.write(buffer[i]);
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (i== 8) Serial.print("\n Lon: ");
if (i >= 8 && i<=16)
{
    Serial.write(buffer[i]);
}
if (i == 17)
{
    Serial.print("\n Status: ");
    Serial.write(buffer[i]);
}
if (i == 18)
{
    Serial.print("\n Route: ");
    Serial.write(buffer[i]);
}
if (i == 19)
{
    Serial.print("\n last goal: ");
    Serial.write(buffer[i]);
}
if (i == 20)
{
    Serial.print("\n going: ");
    Serial.write(buffer[i]);
}
if (i == 21)
{
    Serial.print("\n r1: ");
    Serial.write(buffer[i]);
    Serial.print("\n");
    delay(200);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

```

ก.10 โค้ด Run แบบสมบูรณ์

```

#include <math.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <Servo.h>
#include <HMC5883L.h>
#include <Wire.h>
#include <nRF905.h>
#include <SPI.h>
#include <stdlib.h>

#define RXADDR 0xFE4CA6E5 // Address of this device (4 bytes / long data
type)
#define TXADDR 0x586F2E10 // Address of device to send to (4 bytes / long
data type)
#define TIMEOUT 1000 // 1 second ping timeout

// Just some random data to send
byte data[NRF905_MAX_PAYLOAD] = {
0x0A,
0x68,
0x45,
0xFA

```

```
};
```

```
char N = 'n';
```

```
char myStr[32];
```

```
char *copystr,whatt[33];
```

```
HMC5883L compass;
```

```
Servo myservo;
```

```
long lat,lon;
```

```
//float flat,flon;
```

```
float cog;
```

```
TinyGPS gps;
```

```
float deg;
```

```
int inPin1 = 44; // the number of the input switch pin
```

```
int inPin2 = 40;
```

```
int inPin3 = 36;
```

```
int inPin4 = 32;
```

```
int inPin5 = 28;
```

```
int inPin6 = 24;
```

```
int reading1,reading2,reading3,reading4,reading5,reading6;
```

```
int IN1 = 5;
```

```
int IN2 = 4;
```

```
int EN1 = 6;
```

```
int route;
```

```
int flag1 = 45;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int flag2 = 41;
```

```
long spot1_lat = 13729480;
```

```
long spot1_lon = 100775060;
```

```
long spot2_lat = 13728727;
```

```
long spot2_lon = 100775148;
```

```
long spot3_lat = 13729155;
```

```
long spot3_lon = 100776032;
```

```
long spot4_lat = 13729556;
```

```
long spot4_lon = 100776844;
```

```
long spot5_lat = 13728531;
```

```
long spot5_lon = 100776903;
```

```
long spot6_lat = 13729480;
```

```
long spot6_lon = 100776014;
```

```
long spot7_lat = 13728540;
```

```
long spot7_lon = 100775145;
```

```
long spot8_lat = 13728540;
```

```
long spot8_lon = 100776090;
```

```
int i = 0;
```

```
long begin_lat,begin_lon;
```

```
long target_lat,target_lon;
```

```
int c;
```

```
int r1=0;
```

```
char lastgoal[1];
```

```
char going[1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int pos = 0;
int Speed;
int goal = 0;

void setup()
{
  // Start up
  nRF905_init();

  // Set address of this device
  nRF905_setRXAddress(RXADDR);

  // Lowest transmit level -10db
  nRF905_setTransmitPower(NRF905_PWR_6);

  // Reduce receive sensitivity to save a few mA
  nRF905_setLowRxPower(NRF905_LOW_RX_ENABLE);

  // Put into receive mode
  nRF905_receive();

  Wire.begin();
  Serial.begin(38400); // connect serial
  Serial2.begin(9600); // connect gps sensor
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(EN1,OUTPUT);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

myservo.attach(12);
pinMode(inPin1, INPUT);
pinMode(inPin2, INPUT);
pinMode(inPin3, INPUT);
pinMode(inPin4, INPUT);
pinMode(inPin5, INPUT);
pinMode(inPin6, INPUT);
pinMode(flag1,OUTPUT);
pinMode(flag2,OUTPUT);

compass = HMC5883L();

Serial.println("Setting scale to +/- 1.3 Ga");
int error = compass.SetScale(1.3); // Set the scale of the compass.
if(error != 0) // If there is an error, print it out.
    Serial.println(compass.GetErrorText(error));

Serial.println("Setting measurement mode to continuous.");
error = compass.SetMeasurementMode(Measurement_Continuous); // Set
the measurement mode to Continuous
if(error != 0) // If there is an error, print it out.
    Serial.println(compass.GetErrorText(error));
}

void loop()
{
    while(Serial2.available())
    {
        if(gps.encode(Serial2.read()))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  reading1 = digitalRead(inPin1);
  reading2 = digitalRead(inPin2);
  reading3 = digitalRead(inPin3);
  reading4 = digitalRead(inPin4);
  reading5 = digitalRead(inPin5);
  reading6 = digitalRead(inPin6);
  getCurrentPoint(gps);
  Sentdata();

  if (i == 0)
  {
    flagdown();
    analogWrite(EN1, 0);
    myservo.write(95);
    begin_lat = lat;
    begin_lon = lon;
    if (begin_lat >= 13728590 && begin_lat <= 13729544 && begin_lon
    >= 100775950 && begin_lon <= 100776050)
      route = 1;
    else if (begin_lat >= 13729380 && begin_lat <= 13729520 &&
    begin_lon >= 100775060 && begin_lon <= 100776014)
      route = 2;
    else if (begin_lat >= 13729380 && begin_lat <= 13729520 &&
    begin_lon >= 100776015 && begin_lon <= 100776900)
      route = 3;
    else if (begin_lat >= 13728540 && begin_lat <= 13729555 &&
    begin_lon >= 100774945 && begin_lon <= 100775250)
      route = 4;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (begin_lat >= 13728320 && begin_lat <= 13728620 &&
begin_lon >= 100775045 && begin_lon <= 100776090)

```

```

    route = 5;

```

```

else if (begin_lat >= 13728320 && begin_lat <= 13728620 &&
begin_lon >= 100776091 && begin_lon <= 100777000)

```

```

    route = 6;

```

```

if (reading1 == HIGH && reading2 == LOW && reading3 == LOW &&
reading4 == LOW && reading5 == LOW && reading6 == LOW)

```

```

    {goal = 1; i++;}

```

```

else if (reading1 == LOW && reading2 == HIGH && reading3 == LOW
&& reading4 == LOW && reading5 == LOW && reading6 == LOW)

```

```

    {goal = 2; i++;}

```

```

else if (reading1 == LOW && reading2 == LOW && reading3 == HIGH
&& reading4 == LOW && reading5 == LOW && reading6 == LOW)

```

```

    {goal = 3; i++;}

```

```

else if (reading1 == LOW && reading2 == LOW && reading3 == LOW
&& reading4 == HIGH && reading5 == LOW && reading6 == LOW)

```

```

    {goal = 4; i++;}

```

```

else if (reading1 == LOW && reading2 == LOW && reading3 == LOW
&& reading4 == LOW && reading5 == HIGH && reading6 == LOW)

```

```

    {goal = 5; i++;}

```

```

else if (reading1 == LOW && reading2 == LOW && reading3 == LOW
&& reading4 == LOW && reading5 == LOW && reading6 == LOW)

```

```

    i = 0;

```

```

else i = 2;

```

```

r1 = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print("begin lat : ");Serial.print(begin_lat);
Serial.print(" begin lon : ");Serial.println(begin_lon);
}

```

```

else if (i == 2)

```

```

{
  flagup();
  analogWrite(EN1, 0);
  myservo.write(95);
  if((reading1 == LOW && reading2 == LOW && reading3 == LOW &&
reading4 == LOW && reading5 == LOW && reading6 == LOW))
  i = 0;
}

```

```

else if (i == 1)

```

```

{
  digitalWrite(flag1,LOW);
  digitalWrite(flag2,LOW);
  if (goal == 1)
  {
    lastgoal[0] = '1';
    Serial.println("1 = HIGH");
    if (route == 1)
    {
      if(r1 == 0)
      {
        target_lat = spot6_lat;
        target_lon = spot6_lon;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Running();
going[0] = '6';
if (lat >= spot6_lat)
    {r1++;}
}
else if(r1 == 1)
{
    target_lat = spot1_lat;
    target_lon = spot1_lon;
    Running();
    going[0] = '1';
    if (lon <= spot1_lon)
        {i=2;}
    else if (route == 3) {
        target_lat = spot1_lat;
        target_lon = spot1_lon;
        Running();
        going[0] = '1';
        if (lon <= spot1_lon)
            {i=2;}
    }

else if (route == 2)
{
    target_lat = spot1_lat;
    target_lon = spot1_lon;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Running();
going[0] = '1';
if (lon <= spot1_lon)
    {i=2;}
}

```

```

else if (route == 4)
{
    target_lat = spot1_lat;
    target_lon = spot1_lon;
    Running();
    going[0] = '1';
    if (lat >= spot1_lat)
        {i=2;}
else if (route == 5 || 6)
{
    if(r1 == 0)
    {
        target_lat = spot7_lat;
        target_lon = spot7_lon;
        Running();
        going[0] = '7';
        if (lon <= spot7_lon)
            {r1++;}
    }
else if(r1 == 1)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

target_lat = spot1_lat;
target_lon = spot1_lon;
Running();
going[0] = '1';
if (lat >= spot1_lat)
    {i=2;}
}
}
else if (route == 0)
    {i = 2;}

if (reading1 == LOW && reading2 == LOW && reading3 == LOW
&& reading4 == LOW && reading5 == LOW && reading6 == HIGH)
    {i = 2;}
}
else if (goal == 2)
{
    lastgoal[0] = '2';
    Serial.println("2 = HIGH");
    if (route == 1)
    {
        if(r1 == 0)
        {
            target_lat = spot8_lat;
            target_lon = spot8_lon;
            Running();
            going[0] = '8';
            if (lat <= spot8_lat)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {r1++;}
    }
    else if (r1 == 1)
    {
        target_lat = spot7_lat;
        target_lon = spot7_lon;
        Running();
        going[0] = '7';
        if (lon <= spot7_lon)
            {r1=2;}
    }
    else if (r1 == 2)
    {
        target_lat = spot2_lat;
        target_lon = spot2_lon;
        Running();
        going[0] = '2';
        if (lat >= spot2_lat)
            {i=2;}
    }
}

else if (route == 2 || 3)
{
    if(r1 == 0)
    {
        target_lat = spot1_lat;
        target_lon = spot1_lon;
        Running();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

going[0] = '1';
if (lon <= spot1_lon)
    {r1++;}
}
else if (r1 == 1)
{
    target_lat = spot2_lat;
    target_lon = spot2_lon;
    Running();
    going[0] = '2';
    if (lat <= spot2_lat)
        {i=2;}
}
else if (route == 4)
{
    if (begin_lat > spot4_lat)
    {
        target_lat = spot2_lat;
        target_lon = spot2_lon;
        Running();
        going[0] = '2';
        if (lat <= spot2_lat)
            {i=2;}
    }
    else if (begin_lat < spot4_lat)
    {
        target_lat = spot2_lat;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

target_lon = spot2_lon;
Running();
going[0] = '2';
if (lat >= spot2_lat)
    {i=2;}
}
}

```

```

else if(route == 5 || 6)
{
if (r1 == 0)
{
target_lat = spot7_lat;
target_lon = spot7_lon;
Running();
going[0] = '7';
if (lon <= spot7_lon)
    {r1++;}
}
else if (r1 == 1)
{
target_lat = spot2_lat;
target_lon = spot2_lon;
Running();
going[0] = '2';
if (lat >= spot2_lat)
    {i=2;}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if (route == 0)
```

```
{i = 2;}
```

```
if (reading1 == LOW && reading2 == LOW && reading3 == LOW &&
reading4 == LOW && reading5 == LOW && reading6 == HIGH)
```

```
{i = 2;}
```

```
}
```

```
else if (goal == 3)
```

```
{
```

```
lastgoal[0] = '3';
```

```
Serial.println("3 = HIGH");
```

```
if(route == 1)
```

```
{
```

```
if(begin_lat > spot3_lat)
```

```
{
```

```
target_lat = spot3_lat;
```

```
target_lon = spot3_lon;
```

```
Running();
```

```
going[0] = '3';
```

```
if (lat <= spot3_lat)
```

```
{i=2;}
```

```
}
```

```
else if(begin_lat < spot3_lat)
```

```
{
```

```
target_lat = spot3_lat;
```

```
target_lon = spot3_lon;
```

```
Running();
```

```
going[0] = '3';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (lat >= spot3_lat)
        {i=2;}
    }
}

```

```

else if (route == 2)

```

```

{
    if(r1 == 0)
    {
        target_lat = spot6_lat;
        target_lon = spot6_lon;
        Running();
        going[0] = '6';
        if (lon >= spot6_lon)
            {r1++;}
    }
    else if(r1 == 1)
    {
        target_lat = spot3_lat;
        target_lon = spot3_lon;
        Running();
        going[0] = '3';
        if (lat <= spot3_lat)
            {i=2;}
    }
}
}

```

```

else if(route == 3)

```

```

{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(r1 == 0)
{
    target_lat = spot6_lat;
    target_lon = spot6_lon;
    Running();
    going[0] = '6';
    if (lon <= spot6_lon)
        {r1++;}
}
else if(r1 == 1)
{
    target_lat = spot3_lat;
    target_lon = spot3_lon;
    Running();
    going[0] = '3';
    if (lat <= spot3_lat)
        {i=2;}
}
}

else if(route == 4)
{
    if(r1 == 0)
    {
        target_lat = spot1_lat;
        target_lon = spot1_lon;
        Running();
        going[0] = '1';
        if (lat >= spot1_lat)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {r1++;}
    }
    else if(r1 == 1)
    {
        target_lat = spot6_lat;
        target_lon = spot6_lon;
        Running();
        going[0] = '6';
        if (lon >= spot6_lon)
            {r1=2;}
    }
    else if(r1 == 2)
    {
        target_lat = spot3_lat;
        target_lon = spot3_lon;
        Running();
        going[0] = '3';
        if (lat <= spot3_lat)
            {i=2;}
    }
}

else if (route == 5)
{
    if(r1 == 0)
    {
        target_lat = spot8_lat;
        target_lon = spot8_lon;
        Running();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

going[0] = '8';
if (lon >= spot8_lon)
    {r1++;}
}
else if(r1 == 1)
{
    target_lat = spot3_lat;
    target_lon = spot3_lon;
    Running();
    going[0] = '3';
    if (lat >= spot3_lat)
        {i=2;}
}
else if (route == 6)
{
    if(r1 == 0)
    {
        target_lat = spot8_lat;
        target_lon = spot8_lon-150;
        Running();
        going[0] = '8';
        if (lon <= spot8_lon-150)
            {r1++;}
    }
}
else if(r1 == 1)
{
    target_lat = spot3_lat;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

target_lon = spot3_lon;
Running();
going[0] = '3';
if (lat >= spot3_lat)
    {i=2;}
}
}
else if (route == 0)
    {i = 2;}

if (reading1 == LOW && reading2 == LOW && reading3 == LOW &&
reading4 == LOW && reading5 == LOW && reading6 == HIGH)
    {i = 2;}
}
else if (goal == 4)
{
lastgoal[0] = '4';
Serial.println("4 = HIGH");
if (route ==1)
{
if(r1 == 0)
{
target_lat = spot6_lat;
target_lon = spot6_lon;
Running();
going[0] = '6';
if (lat >= spot6_lat)
    {r1++;}
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if(r1 == 1)
{
    target_lat = spot4_lat;
    target_lon = spot4_lon;
    Running();
    going[0] = '4';
    if (lon >= spot4_lon)
        {i=2;}
}
}
else if (route == 2 || 3)
{
    target_lat = spot4_lat;
    target_lon = spot4_lon;
    Running();
    going[0] = '4';
    if (lon >= spot4_lon)
        {i=2;}
}

```

```

else if (route == 4)
{
    if(r1 == 0)
    {
        target_lat = spot1_lat;
        target_lon = spot1_lon;
        Running();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

going[0] = '1';
if (lat >= spot1_lat)
    {r1++;}
}
else if(r1 == 1)
{
    target_lat = spot4_lat;
    target_lon = spot4_lon;
    Running();
    going[0] = '4';
    if (lon >= spot4_lon)
        {i=2;}
}
else if (route == 5)
{
    if(r1 == 0)
    {
        target_lat = spot8_lat;
        target_lon = spot8_lon;
        Running();
        going[0] = '8';
        if (lon >= spot8_lon)
            {r1++;}
    }
}
else if(r1 == 1)
{
    target_lat = spot6_lat;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

target_lon = spot6_lon;
Running();
going[0] = '6';
if (lat >= spot6_lat)
    {r1=2;}
}
else if(r1 == 2)
{
    target_lat = spot4_lat;
    target_lon = spot4_lon;
    Running();
    going[0] = '4';
    if (lon >= spot4_lon)
        {l=2;}
}
else if (route == 6)
{
    if(r1 == 0)
    {
        target_lat = spot8_lat;
        target_lon = spot8_lon-150;
        Running();
        going[0] = '8';
        if (lon <= spot8_lon-150)
            {r1++;}
    }
    else if(r1 == 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    target_lat = spot6_lat;
    target_lon = spot6_lon;
    Running();
    going[0] = '6';
    if (lat >= spot6_lat)
        {r1=2;}
}
else if(r1 == 2)
{
    target_lat = spot4_lat;
    target_lon = spot4_lon;
    Running();
    going[0] = '4';
    if (lon >= spot4_lon)
        {i=2;}
}
}
else if (route == 0)
    {i = 2;}

```

```

if (reading1 == LOW && reading2 == LOW && reading3 == LOW &&
reading4 == LOW && reading5 == LOW && reading6 == HIGH)

```

```

    { i = 2;}

```

```

}

```

```

else if (goal == 5)

```

```

{

```

```

    lastgoal[0] = '5';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("5 = HIGH");
if (route == 1)
{
  if(r1 == 0)
  {
    target_lat = spot8_lat-150;
    target_lon = spot8_lon;
    Running();
    going[0] = '8';
    if (lat <= spot8_lat-150)
      {r1++;}
    }
  else if(r1 == 1)
  {
    target_lat = spot5_lat;
    target_lon = spot5_lon;
    Running();
    going[0] = '5';
    if (lon >= spot5_lon)
      {i=2;}
    }
  }
}

```

```

else if (route == 2)
{
  if(r1 == 0)
  {

```

```

    target_lat = spot6_lat;
    target_lon = spot6_lon;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Running();
going[0] = '6';
if (lon >= spot6_lon)
    {r1++;}
}
else if(r1 == 1)
{
    target_lat = spot8_lat-150;
    target_lon = spot8_lon;
    Running();
    going[0] = '8';
    if (lat <= spot8_lat-150)
        {r1=2;}
}
else if(r1 == 2)
{
    target_lat = spot5_lat;
    target_lon = spot5_lon;
    Running();
    going[0] = '5';
    if (lon >= spot5_lon)
        {i=2;}
}
}

else if (route == 3)
{
    if(r1 == 0)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

target_lat = spot6_lat;
target_lon = spot6_lon;
Running();
going[0] = '6';
if (lon <= spot6_lon)
    {r1++;}
}
else if(r1 == 1)
{
target_lat = spot8_lat-150;
target_lon = spot8_lon;
Running();
going[0] = '8';
if (lat <= spot8_lat-150)
    {r1=2;}
}
else if(r1 == 2)
{
target_lat = spot5_lat;
target_lon = spot5_lon;
Running();
going[0] = '5';
if (lon >= spot5_lon)
    {i=2;}
}
}
}

else if(route == 5 || 6)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

target_lat = spot5_lat;
target_lon = spot5_lon;
Running();
going[0] = '5';
if (lon >= spot5_lon)
    {i=2;}
}

```

```

else if(route == 4)
{
if(r1 == 0)
{
target_lat = spot7_lat;
target_lon = spot7_lon;
Running();
going[0] = '7';
if (lat <= spot7_lat)
    {r1++;}
}
else if(r1 == 1)
{
target_lat = spot5_lat;
target_lon = spot5_lon;
Running();
going[0] = '5';
if (lon >= spot5_lon)
    {i=2;}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (route == 0)
    {i = 2;}

    if (reading1 == LOW && reading2 == LOW && reading3 == LOW
    && reading4 == LOW && reading5 == LOW && reading6 == HIGH)
        i = 2;
    }

else
{
    Serial.println("ERROR");
    i = 2;
}
} // end i = 1

} // end if
} // end while
Sentdata();
}

```

```

void Running()
{
    analogWrite(EN1, 200);
    deg = degrees(atan2((target_lon - lon),(target_lat - lat)));
    if(deg < 0)
        deg += 360;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print("Running degree_after_cal: ");
Serial.print(deg);
Serial.print(" COG: ");
Serial.println(cog);

compass_function();
turnaround();
}

void compass_function()
{
  // Retrieve the raw values from the compass (not scaled).
  MagnetometerRaw raw = compass.ReadRawAxis();
  // Retrieved the scaled values from the compass (scaled to the configured
  scale).
  MagnetometerScaled scaled = compass.ReadScaledAxis();

  // Calculate heading when the magnetometer is level, then correct for
  signs of axis.
  float heading = atan2(raw.YAxis, raw.XAxis);

  // Correct for when signs are reversed.
  if(heading < 0)
    heading += 2*PI;

  // Convert radians to degrees for readability.
  float headingDegrees = degrees(heading);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cog = headingDegrees;
}

void turnaround()
{
    if ((cog > deg) && (cog - deg) > (360-cog+deg))
    {
        if ((cog - deg) - (360 - cog +deg) > 120)
            myservo.write(150);
        else if ( (cog - deg) - (360 - cog +deg) < 25)
            myservo.write(120);
        else myservo.write(130);
    }
    else if ( (cog > deg) && (cog - deg) < (360-cog+deg))
    {
        if ((360 - cog +deg) - (cog - deg) > 120)
            myservo.write(20);
        else if ( (360 - cog +deg) - (cog - deg) < 25)
            myservo.write(75);
        else myservo.write(60);
    }
    else if ((cog < deg) &&(deg - cog) > (360+cog-deg))
    {
        if ((deg - cog) - (360 + cog - deg) > 120)
            myservo.write(20);
        else if ((deg - cog) - (360 + cog -deg) < 25)
            myservo.write(75);
        else myservo.write(60);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if ( (cog < deg) && (deg - cog) < (360+cog-deg))
{
    if ((360 + cog -deg) - (deg - cog) > 120)
        myservo.write(150);
    else if ( (360 + cog -deg) - (deg - cog) < 25)
        myservo.write(120);
    else myservo.write(130);
}
else myservo.write(95);
}

void getCurrentPoint(TinyGPS &gps)
{
    //gps.f_get_position(&flat, &flon);
    gps.get_position(&lat,&lon);
    //gps.get_datetime(&date, &time, &age);

    Serial.print("lat: ");Serial.print(lat);Serial.print(" ");// print latitude
    Serial.print("lon: ");Serial.println(lon);
}

void Sentdata()
{
    // Turn on module
    nRF905_powerUp();

    // Set address of device to send to
    nRF905_setTXAddress(TXADDR);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

copystr = ltoa(lat,whatt,10);
myStr[0] = copystr[0];
myStr[1] = copystr[1];
myStr[2] = copystr[2];
myStr[3] = copystr[3];
myStr[4] = copystr[4];
myStr[5] = copystr[5];
myStr[6] = copystr[6];
myStr[7] = copystr[7];
copystr = ltoa(lon,whatt,10);
myStr[8] = copystr[0];
myStr[9] = copystr[1];
myStr[10] = copystr[2];
myStr[11] = copystr[3];
myStr[12] = copystr[4];
myStr[13] = copystr[5];
myStr[14] = copystr[6];
myStr[15] = copystr[7];
myStr[16] = copystr[8];
copystr = ltoa(i,whatt,10);
myStr[17] = copystr[0];
copystr = ltoa(route,whatt,10);
myStr[18] = copystr[0];
myStr[19] = lastgoal[0];
myStr[20] = going[0];
copystr = ltoa(r1,whatt,10);
myStr[21] = copystr[0];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nRF905_setData((byte*)myStr, sizeof(myStr));
```

```
while(!nRF905_send());
```

```
nRF905_receive();
```

```
byte buffer[NRF905_MAX_PAYLOAD];
```

```
bool success;
```

```
}
```

```
void flagup()
```

```
{
```

```
digitalWrite(flag1,LOW);
```

```
digitalWrite(flag2,HIGH);
```

```
}
```

```
void flagdown()
```

```
{
```

```
digitalWrite(flag1,HIGH);
```

```
digitalWrite(flag2,LOW);
```

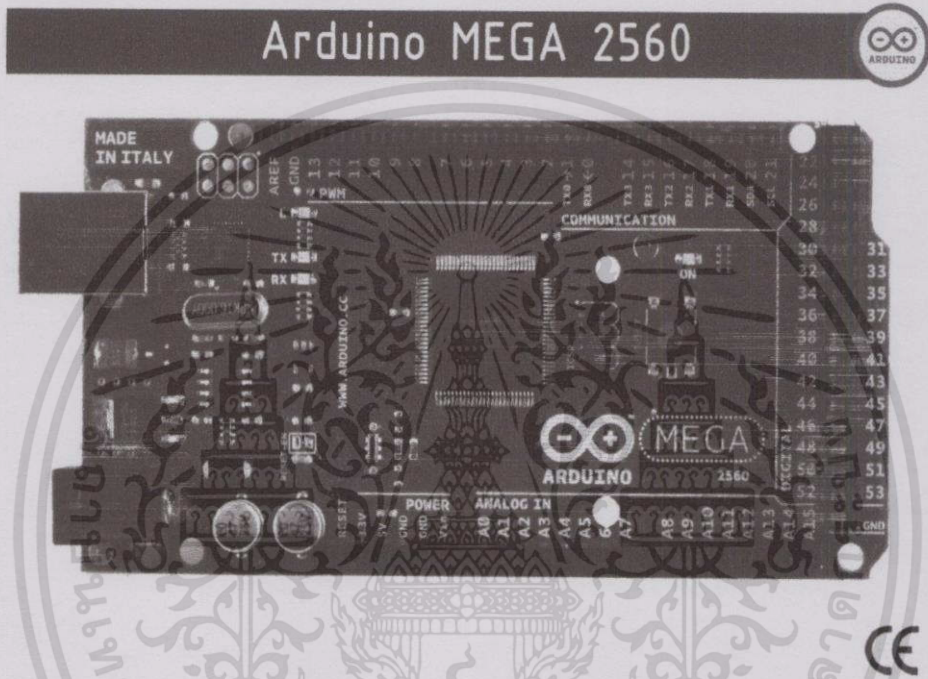
```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 คู่มือ Arduino Mega 2560 R3



The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs),

16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

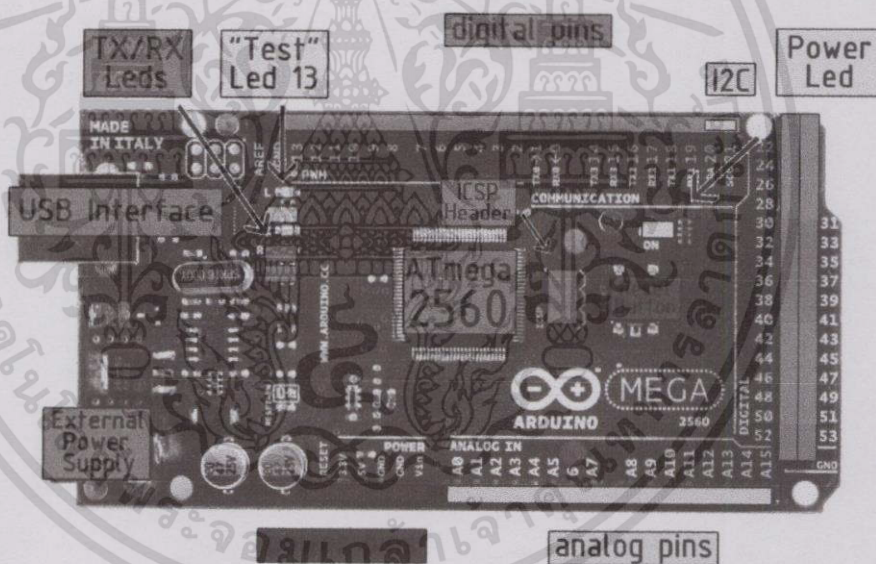
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Technical Specification



Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V Input Voltage
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and

14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.

PWM: 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

I²C: 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the [Wiring website](#)). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AREF. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

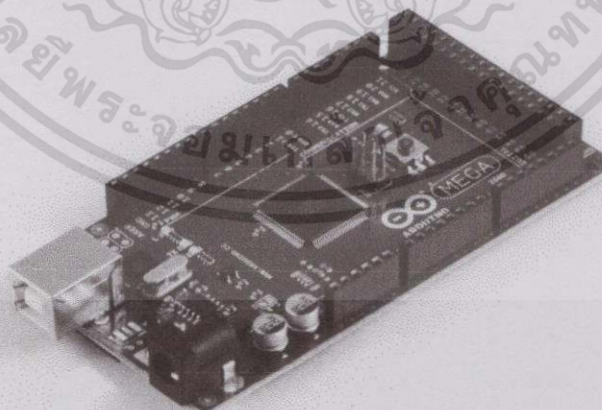
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install Windows Install Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

```

int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop() {
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // set the LED off
  delay(1000);                // wait for a second
}
  
```

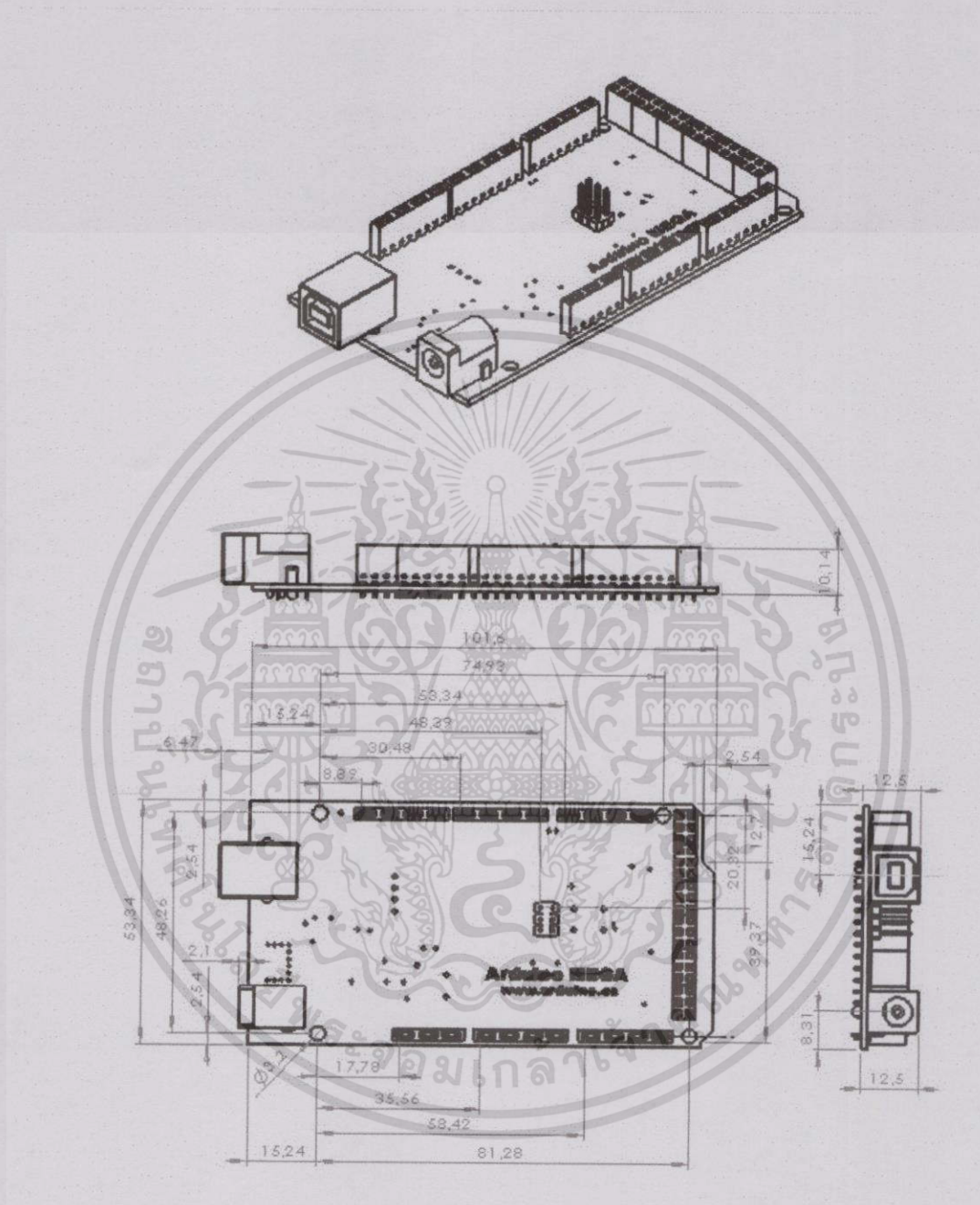
Once you have your sketch you'll see something very close to the screenshot on the right.

In Tools>Board select MEGA Now you have to go to **Tools>SerialPort**

and select the right serial port, the one arduino is attached to.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensioned Drawing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from

Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ข.2 คู่มือ nRF905

Single chip 433/868/915 MHz Transceiver nRF905

FEATURES

- True single chip GFSK transceiver in a small 32-pin package (32L QFN 5x5mm)
- ShockBurst™ mode for low power operation
- Power supply range 1.9 to 3.6 V
- Multi-channel operation – ETSI/FCC
- Data Ready signal when a valid data packet is received or transmitted
- Address Match for detection of incoming packet
- Automatic retransmission of data packet
- Automatic CRC and preamble generation
- Low supply current (TX), typical 9mA @ -10dBm output power
- Low supply current (RX); typical 12.5mA

APPLICATIONS

Wireless data communication
 Alarm and security systems
 Home Automation
 Remote control Surveillance
 Automotive Telemetry
 Industrial sensors Keyless entry Toys

GENERAL DESCRIPTION

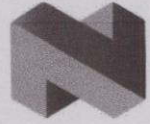
nRF905 is a single-chip radio transceiver for the 433/868/915 MHz ISM band. The transceiver consists of a fully integrated frequency synthesiser, receiver chain with demodulator, a power amplifier, a crystal oscillator and a modulator. The ShockBurst™ feature automatically handles preamble and CRC. Configuration is easily programmable by use of the SPI interface. Current consumption is very low, in transmit only 9mA at an output power of -10dBm, and in receive mode 12.5mA. Built in power down modes makes power saving easily realizable.

QUICK REFERENCE DATA

Parameter	Value	Unit
Supply voltage	1.9 to 3.6	V
Maximum transmit output power	10	dBm
Data rate	50	kbps
Supply current in transmit @ -10dBm output power	9	mA
Supply current in receive mode	12.5	mA
Temperature range	-40 to +85	°C
Typical Sensitivity	-100	dBm
Supply current in power down mode	2.5	µA

Table 1 nRF905 quick reference data.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

ORDERING INFORMATION

Type Number	Description	Version
nRF905 IC	32L QFN 5x5mm	-
nRF905-EVKIT 433	Evaluation kit 433MHz	1.0
nRF905-EVKIT 868/915	Evaluation kit 868/915MHz	1.0

Table 2 nRF905 ordering information.

BLOCK DIAGRAM

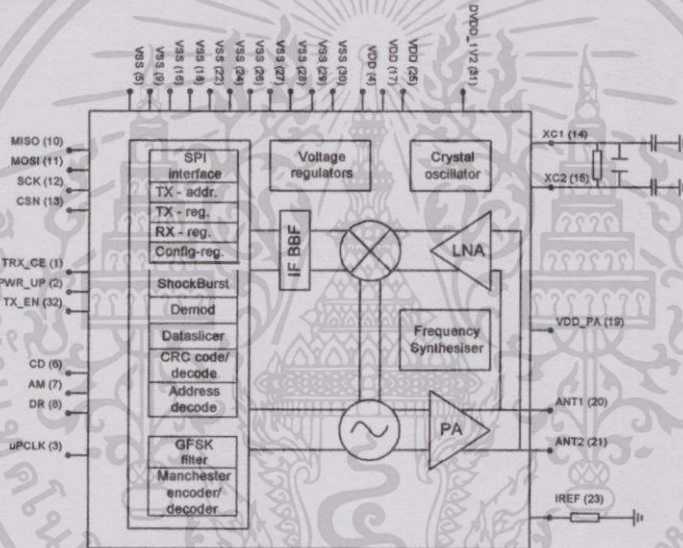
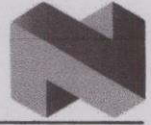


Figure 1 nRF905 with external components.

June 2006

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

TABLE OF CONTENTS

1	Pin Functions	4
2	Pin Assignment	5
3	Electrical Specifications	6
4	Current Consumption	8
5	Modes of Operation	9
5.1	Active Modes	9
5.2	Power Saving Modes	9
5.3	nRF ShockBurst™ Mode	9
5.4	Typical ShockBurst™ TX	10
5.5	Typical ShockBurst™ RX	12
5.6	Power Down Mode	14
5.7	Standby Mode	14
6	Device Configuration	15
6.1	SPI Register Configuration	15
6.2	SPI Instruction Set	16
6.3	SPI Timing	17
6.4	RF – Configuration Register Description	19
6.5	Register Contents	20
7	Important Timing Data	21
7.1	Device Switching Times	21
7.2	ShockBurst™ TX timing	21
7.3	ShockBurst™ RX timing	22
7.4	Preamble	22
7.5	Time On Air	23
8	Peripheral RF Information	24
8.1	Crystal Specification	24
8.2	External Clock Reference	24
8.3	Microprocessor Output Clock	24
8.4	Antenna Output	25
8.5	Output Power Adjustment	25
8.6	Modulation	25
8.7	Output Frequency	26
8.8	PCB Layout and Decoupling Guidelines	27
9	nRF905 features	28
9.1	Carrier Detect	28
9.2	Address Match	28
9.3	Data Ready	28
9.4	Auto Retransmit	29
9.5	RX Reduced Power Mode	29
10	Package Outline	30
10.1	Package marking	31
11	Application Examples	32
11.1	Differential Connection to a Loop Antenna	32
11.2	PCB Layout Example; Differential Connection to a Loop Antenna	33
11.3	Single ended connection to 50Ω antenna	34
11.4	PCB Layout Example; Single Ended Connection to 50Ω Antenna	36
12	Absolute Maximum Ratings	38
13	Glossary of Terms	39
14	Definitions	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตในการค้า

PRODUCT SPECIFICATION



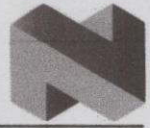
nRF905 Single Chip 433/868/915 MHz Radio Transceiver

1 PIN FUNCTIONS

Pin	Name	Pin function	Description
1	TRX_CE	Digital input	Enables chip for receive and transmit
2	PWR_UP	Digital input	Power up chip
3	uPCLK	Clock output	Output clock, divided crystal oscillator full-swing clock
4	VDD	Power	Power supply (+3V DC)
5	VSS	Power	Ground (0V)
6	CD	Digital output	Carrier Detect
7	AM	Digital output	Address Match
8	DR	Digital output	Receive and transmit Data Ready
9	VSS	Power	Ground (0V)
10	MISO	SPI - interface	SPI output
11	MOSI	SPI - interface	SPI input
12	SCK	SPI - Clock	SPI clock
13	CSN	SPI - enable	SPI enable, active low
14	XC1	Analog Input	Crystal pin 1/ External clock reference pin
15	XC2	Analog Output	Crystal pin 2
16	VSS	Power	Ground (0V)
17	VDD	Power	Power supply (+3V DC)
18	VSS	Power	Ground
19	VDD_PA	Power output	Positive supply (1.8V) to nRF905 power amplifier
20	ANT1	RF	Antenna interface 1
21	ANT2	RF	Antenna interface 2
22	VSS	Power	Ground (0V)
23	IREF	Analog Input	Reference current
24	VSS	Power	Ground (0V)
25	VDD	Power	Power supply (+3V DC)
26	VSS	Power	Ground (0V)
27	VSS	Power	Ground (0V)
28	VSS	Power	Ground (0V)
29	VSS	Power	Ground (0V)
30	VSS	Power	Ground (0V)
31	DVDD 1V2	Power	Low voltage positive digital supply output for de-coupling
32	TX_EN	Digital input	TX_EN="1"TX mode, TX_EN="0"RX mode

Table 3 nRF905 pin function.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

2 PIN ASSIGNMENT

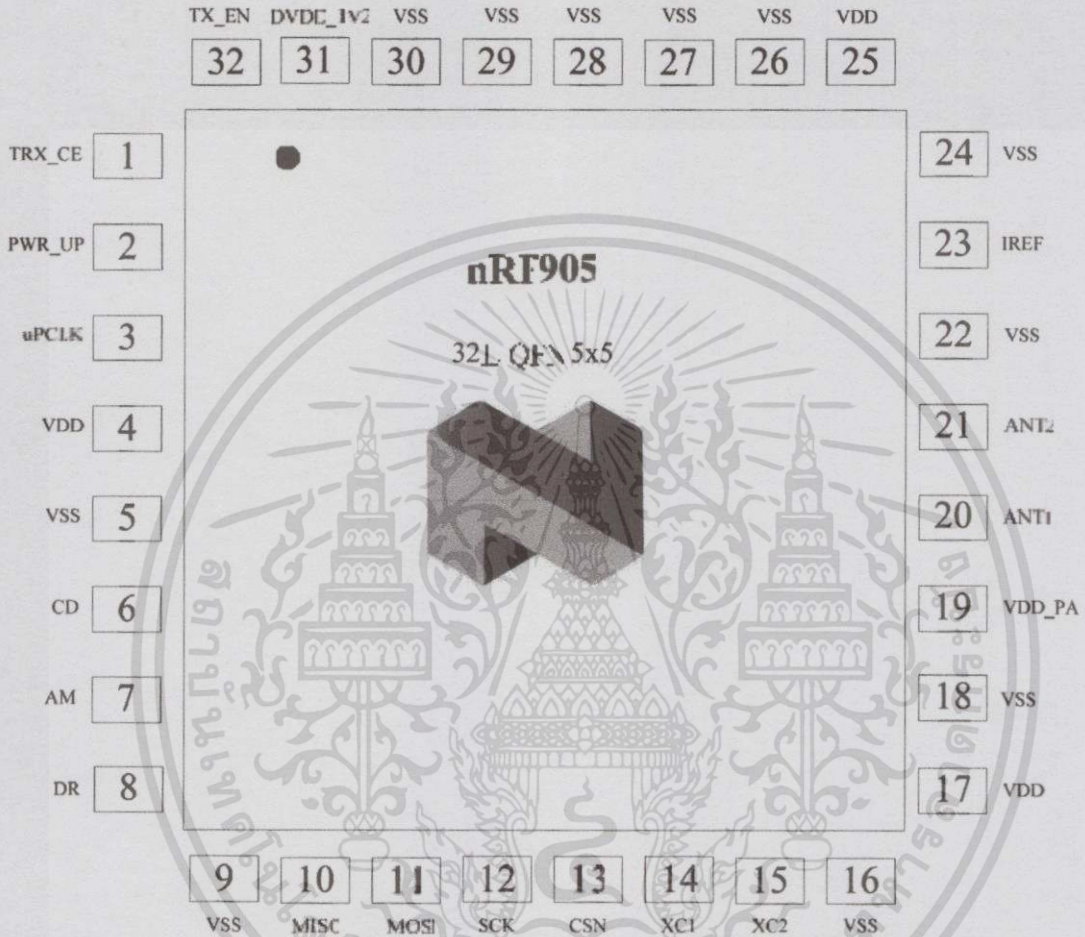


Figure 2 nRF905 pin assignment (top view) for a 32L QFN 5x5 package.

PRODUCT SPECIFICATION

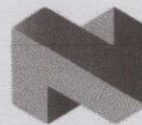


nRF905 Single Chip 433/868/915 MHz Radio Transceiver

- 1) Max value determined by design and characterization testing.
- 2) Output frequency is 4MHz load of external clock pin is 5pF, Crystal is 4MHz.
- 3) Crystal is 4MHz.
- 4) Pin voltages are VSS or VDD.
- 5) Chip in power down, SPI_SCK frequency is 1MHz.
- 6) Operates in the 433, 868 and 915 MHz ISM band.
- 7) The crystal frequency may be chosen from 5 different values (4, 8, 12, 16, and 20MHz)
- 8) Data is Manchester-encoded before GFSK modulation.
- 9) Optimum load impedance, please see peripheral RF information.
- 10) Channel width and channel spacing is 200kHz.
- 11) Channel Level +3dB over sensitivity, interfering signal a standard CW, image lies 2MHz above wanted.



PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

4 CURRENT CONSUMPTION

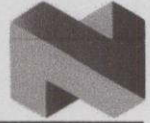
MODE	CRYSTAL FREQ. [MHZ]	OUTPUT CLOCK FREQ. [MHZ]	TYPICAL CURRENT
Power Down	16	OFF	2.5 μ A
Standby	4	OFF	12 μ A
Standby	8	OFF	25 μ A
Standby	12	OFF	27 μ A
Standby	16	OFF	32 μ A
Standby	20	OFF	46 μ A
Standby	4	0.5	110 μ A
Standby	8	0.5	125 μ A
Standby	12	0.5	130 μ A
Standby	16	0.5	135 μ A
Standby	20	0.5	150 μ A
Standby	4	1	130 μ A
Standby	8	1	145 μ A
Standby	12	1	150 μ A
Standby	16	1	155 μ A
Standby	20	1	170 μ A
Standby	4	2	170 μ A
Standby	8	2	185 μ A
Standby	12	2	190 μ A
Standby	16	2	195 μ A
Standby	20	2	210 μ A
Standby	4	4	260 μ A
Standby	8	4	275 μ A
Standby	12	4	280 μ A
Standby	16	4	285 μ A
Standby	20	4	300 μ A
Rx @ 433	16	OFF	12.2 mA
Rx @ 868/915	16	OFF	12.8 mA
Reduced Rx	16	OFF	10.5 mA
Tx @ 10dBm	16	OFF	30 mA
Tx @ 6dBm	16	OFF	20 mA
Tx @ -2dBm	16	OFF	14 mA
Tx @ -10dBm	16	OFF	9 mA

Conditions: VDD = 3.0V, VSS = 0V, T_A = 27°C,
Load capacitance of external clock = 13pF, Crystal load capacitance = 12pF

Table 5 nRF905 current consumption.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

5 MODES OF OPERATION

The nRF905 has two active (RX/TX) modes and two power-saving modes

5.1 Active Modes

- ShockBurst™ RX
- ShockBurst™ TX

5.2 Power Saving Modes

- Power down and SPI - programming
- Standby and SPI - programming

The nRF905 mode is decided by the settings of TRX_CE, TX_EN and PWR_UP.

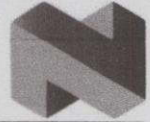
PWR_UP	TRX_CE	TX_EN	Operating Mode
0	X	X	Power down and SPI - programming
1	0	X	Standby and SPI - programming
1	X	0	Read data from RX register
1	1	0	Radio Enabled - ShockBurst™ RX
1	1	1	Radio Enabled - ShockBurst™ TX

Table 6 nRF905 operational modes.

5.3 nRF ShockBurst™ Mode

The nRF905 uses the Nordic Semiconductor ASA ShockBurst™ feature. ShockBurst™ makes it possible to use the high data rate offered by the nRF905 without the need of a costly, high-speed micro controller (MCU) for data processing/clock recovery. By placing all high speed signal processing related to RF protocol on-chip, the nRF905 offers the application micro controller a simple SPI interface, the data rate is decided by the interface-speed the micro controller itself sets up. By allowing the digital part of the application to run at low speed, while maximizing the data rate on the RF link, the nRF905 ShockBurst™ mode reduces the average current consumption in applications. In ShockBurst™ RX, Address Match (AM) and Data Ready (DR) notifies the MCU when a valid address and payload is received respectively. In ShockBurst™ TX, the nRF905 automatically generates preamble and CRC. Data Ready (DR) notifies the MCU that the transmission is completed. All together, this means reduced memory demand in the MCU resulting in a low cost MCU, as well as reduced software development time.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

5.4 Typical ShockBurst™ TX

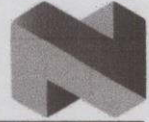
1. When the application MCU has data for a remote node, the address of the receiving node (TX-address) and payload data (TX-payload) are clocked into nRF905 via the SPI interface. The application protocol or MCU sets the speed of the interface.
2. MCU sets TRX_CE and TX_EN high, this activates a nRF905 ShockBurst™ transmission.
3. nRF905 ShockBurst™:
 - Radio is automatically powered up.
 - Data packet is completed (preamble added, CRC calculated).
 - Data packet is transmitted (100kbps, GFSK, Manchester-encoded).
 - Data Ready is set high when transmission is completed.
4. If AUTO_RETRAN is set high, the nRF905 continuously retransmits the packet until TRX_CE is set low.
5. When TRX_CE is set low, the nRF905 finishes transmitting the outgoing packet and then sets itself into standby mode.

If TX_EN is set low while TRX_CE is kept high, the nRF905 finishes transmitting the outgoing packet and then enter RX-mode in the channel already programmed in the RF-CONFIG register.

The ShockBurst™ mode ensures that a transmitted packet that has started always finishes regardless of what TRX_EN and TX_EN is set to during transmission. The new mode is activated when the transmission is completed. Please see subsequent chapters for detailed timing

For test purposes such as antenna tuning and measuring output power it is possible to set the transmitter so that a constant carrier is produced. To do this TRX_CE must be maintained high instead of being pulsed. In addition Auto Retransmit should be switched off. After the burst of data has been sent then the device will continue to send the unmodulated carrier.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

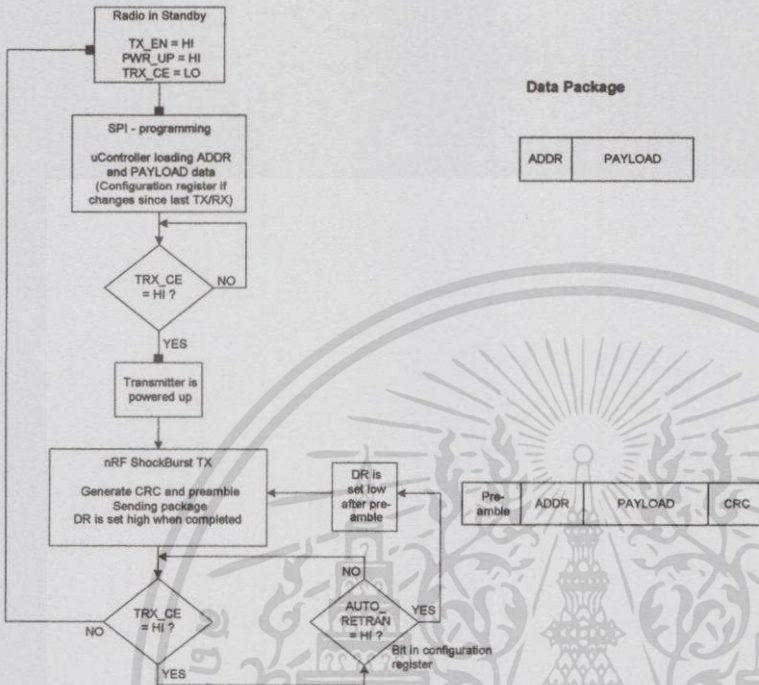
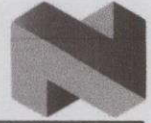


Figure 3 Flowchart ShockBurst™ transmit of nRF905.

NB: DR is set low under the following conditions after it has been set high:

- If TX_EN is set low
- If PWR_UP is set low

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

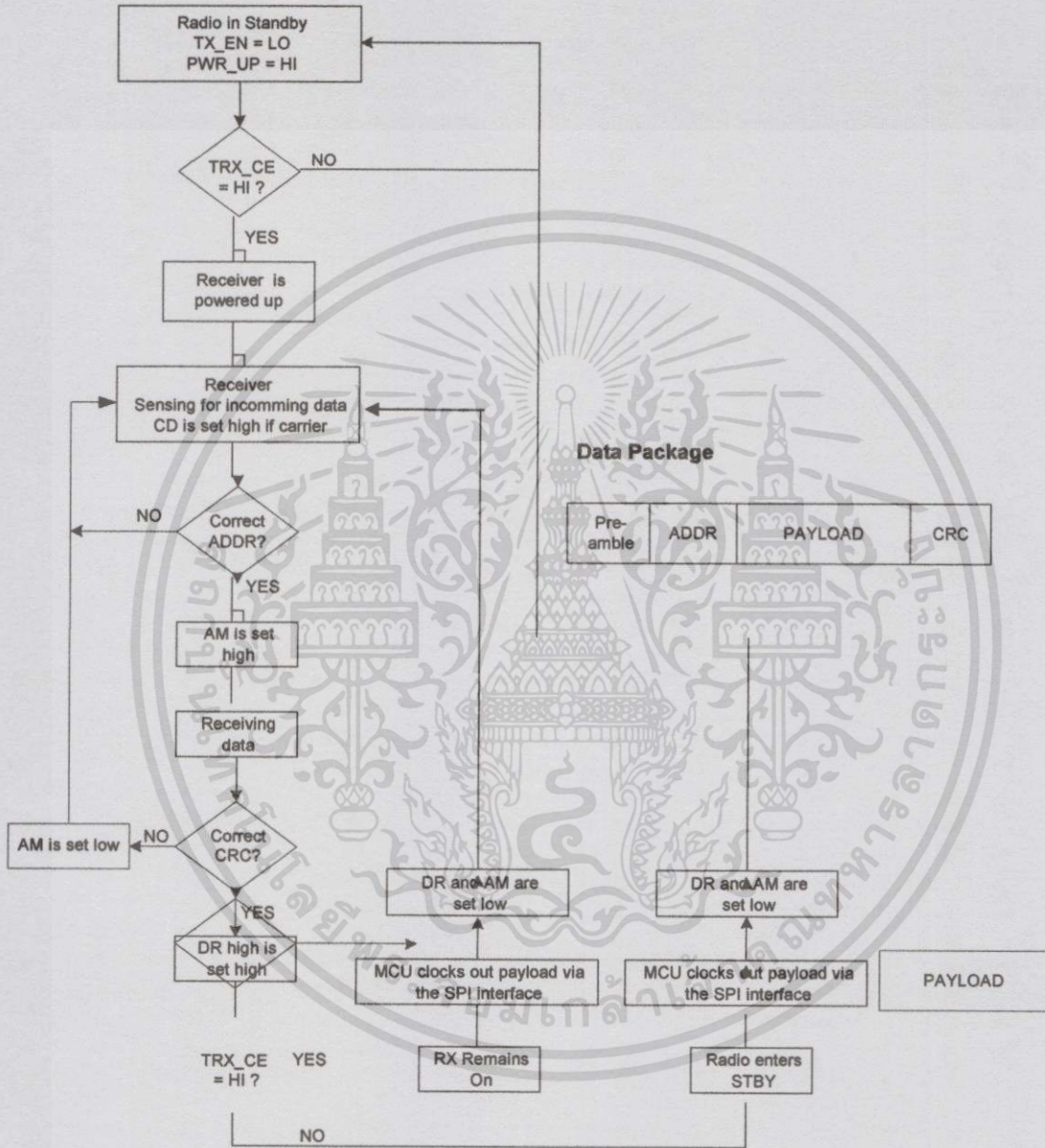


Figure 4 Flowchart ShockBurst™ receive of nRF905.

PRODUCT SPECIFICATION



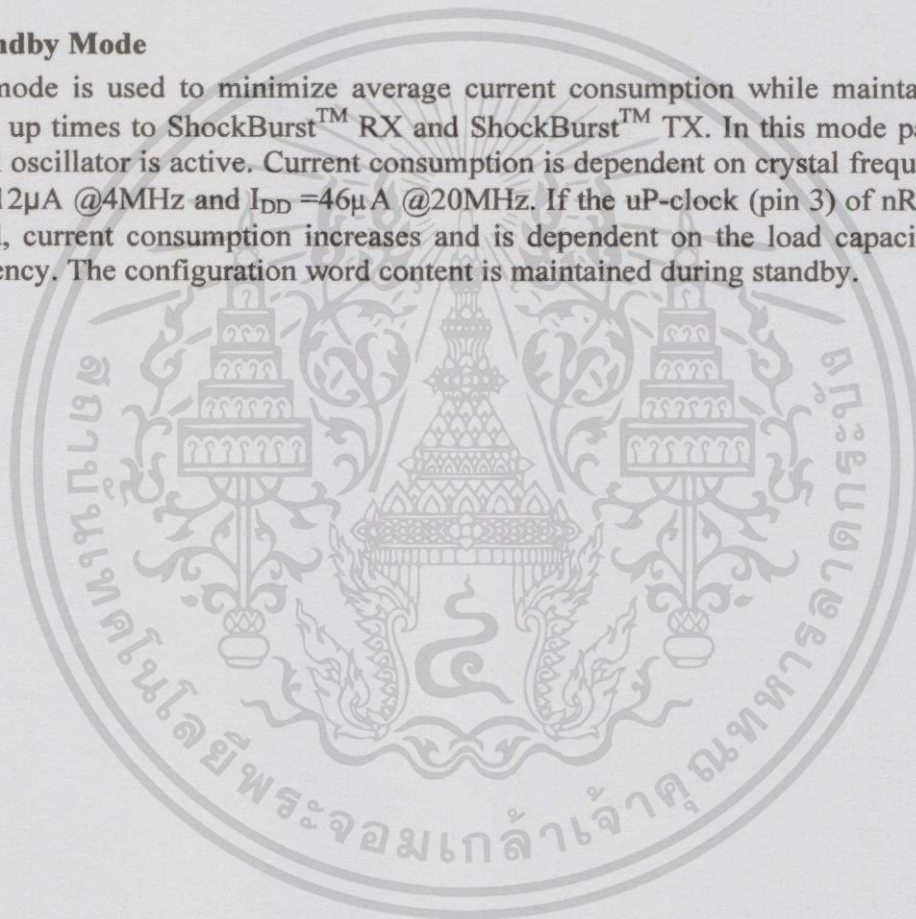
nRF905 Single Chip 433/868/915 MHz Radio Transceiver

5.6 Power Down Mode

In power down the nRF905 is disabled with minimal current consumption, typically less than 2.5 μ A. When entering this mode the device is not active which will minimize average current consumption and maximizing battery lifetime. The configuration word content is maintained during power down.

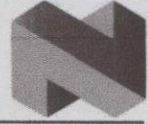
5.7 Standby Mode

Standby mode is used to minimize average current consumption while maintaining short start up times to ShockBurstTM RX and ShockBurstTM TX. In this mode part of the crystal oscillator is active. Current consumption is dependent on crystal frequency, Ex: $I_{DD} = 12\mu A @ 4MHz$ and $I_{DD} = 46\mu A @ 20MHz$. If the uP-clock (pin 3) of nRF905 is enabled, current consumption increases and is dependent on the load capacitance and frequency. The configuration word content is maintained during standby.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

6 DEVICE CONFIGURATION

All configuration of the nRF905 is via the SPI interface. The interface consists of five registers; a SPI instruction set is used to decide which operation shall be performed. The SPI interface can be activated in any mode however Nordic Semiconductor ASA recommends the chip be in standby or power down mode.

6.1 SPI Register Configuration

The SPI interface consists of five internal registers. A register read-back mode is implemented to allow verification of the register contents.

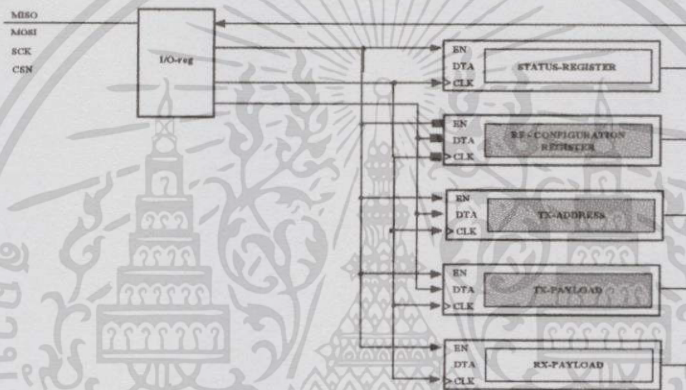


Figure 5 SPI – interface and the five internal registers.

Status – Register

Register contains status of Data Ready (DR) and Address Match (AM).

RF – Configuration Register

Register contains transceiver setup information such as frequency and output power ext.

TX – Address

Register contains address of target device. How many bytes used is set in the configuration register.

TX – Payload

Register containing the payload information to be sent in a ShockBurst™ packet. How many bytes used is set in the configuration register.

RX – Payload

Register containing the payload information derived from a received valid ShockBurst™ packet. How many bytes used is set in the configuration register. Valid data in the RX-Payload register is indicated with a high Date Ready (DR) signal.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

6.2 SPI Instruction Set

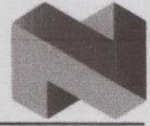
The available commands to be used on the SPI interface is shown below. Whenever CSN is set low the interface expects an instruction. Every new instruction must be started by a high to low transition on CSN.

Instruction set for the nRF905 SPI Serial Interface		
Instruction Name	Instruction Format	Operation
W_CONFIG (WC)	0000 AAAA	Write Configuration-register. AAAA indicates which byte the write operation is to be started from. Number of bytes depends on start address AAAA.
R_CONFIG (RC)	0001 AAAA	Read Configuration-register. AAAA indicates which byte the read operation is to be started from. Number of bytes depends on start address AAAA.
W_TX_PAYLOAD (WTP)	0010 0000	Write TX-payload: 1 – 32 bytes. A write operation will always start at byte 0.
R_TX_PAYLOAD (RTP)	0010 0001	Read TX-payload: 1 – 32 bytes. A read operation will always start at byte 0.
W_TX_ADDRESS (WTA)	0010 0010	Write TX-address: 1 – 4 bytes. A write operation will always start at byte 0.
R_TX_ADDRESS (RTA)	0010 0011	Read TX-address: 1 – 4 bytes. A read operation will always start at byte 0.
R_RX_PAYLOAD (RRP)	0010 0100	Read RX-payload: 1 – 32 bytes. A read operation will always start at byte 0.
CHANNEL_CONFIG (CC)	1000 pphc cccc cccc	Special command for fast setting of CH_NO, HFREQ_PLL and PA_PWR in the CONFIGURATION REGISTER. CH_NO= cccccccc, HFREQ_PLL = h PA_PWR = pp
STATUS REGISTER	N.A.	The content of the status-register (S[7:0]) will always be read to MISO after a high to low transition on CSN as shown in Figure 6 and 7.

Table 7 Instruction set for the nRF905 SPI interface.

A read or a write operation may operate on a single byte or on a set of succeeding bytes from a given start address defined by the instruction. When accessing succeeding bytes one will read or write MSB of the byte with the smallest byte number first.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

6.3 SPI Timing

The interface supports SPI mode 0. SPI operation and timing is given in Figure 6 to Figure 8 and in Table 8. The device must be in one of the power saving modes for the configuration registers to be read or written to.

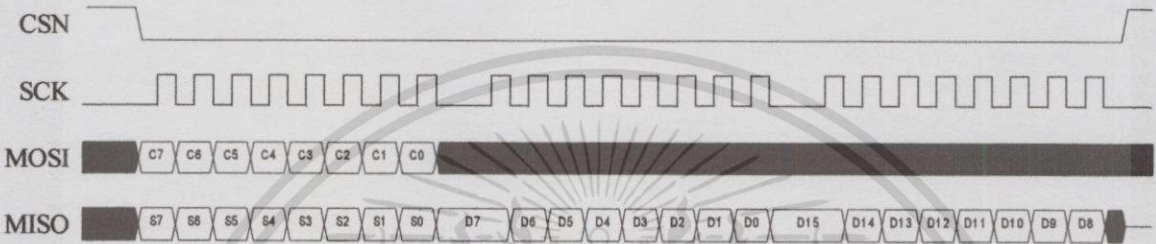


Figure 6 SPI read operation.

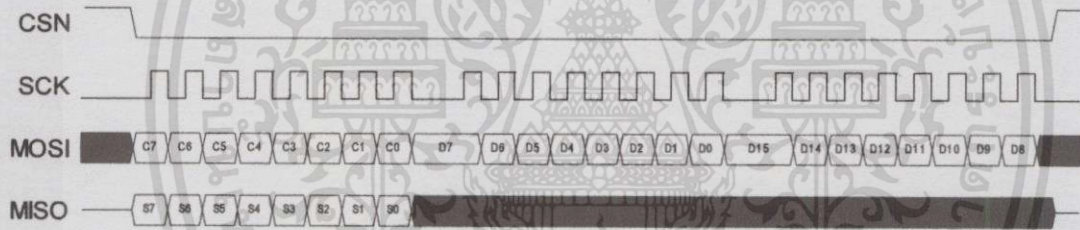


Figure 7 SPI write operation.

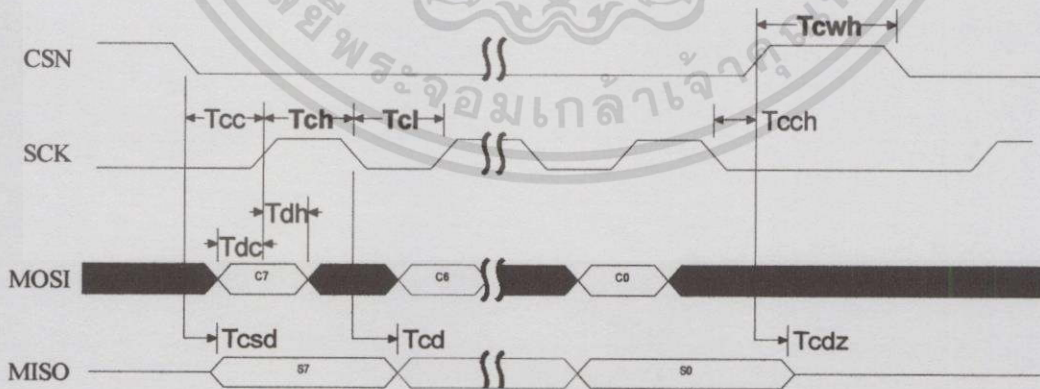


Figure 8 SPI NOP timing diagram.

PRODUCT SPECIFICATION

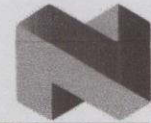


nRF905 Single Chip 433/868/915 MHz Radio Transceiver

PARAMETER	SYMBOL	MIN	MAX	UNITS
Data to SCK Setup	Tdc	5		ns
SCK to Data Hold	Tdh	5		ns
CSN to Data Valid	Tcsd		45	ns
SCK to Data Valid	Tcd		45	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Tsck	DC	10	MHz
SCK Rise and Fall	Tr, Tf		100	ns
CSN to SCK Setup	Tcc	5		ns
SCK to CSN Hold	Tch	5		ns
CSN Inactive time	Tcwh	500		ns
CSN to Output High Z	Tcdz		45	ns

Table 8 SPI timing parameters ($C_{load} = 10\text{pF}$).

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

6.4 RF – Configuration Register Description

Parameter	Bitwidth	Description
CH_NO	9	Sets center freq. together with HFREQ_PLL (default = 001101100 _b = 108 _d). $f_{RF} = (422.4 + CH_NO_d / 10) * (1 + HFREQ_PLL_d)$ MHz
HFREQ_PLL	1	Sets PLL in 433 or 868/915 MHz mode (default = 0). '0' – Chip operating in 433MHz band '1' – Chip operating in 868 or 915 MHz band
PA_PWR	2	Output power (default = 00). '00' -10dBm '01' -2dBm '10' +6dBm '11' +10dBm
RX_RED_PWR	1	Reduces current in RX mode by 1.6mA. Sensitivity is reduced (default = 0). '0' – Normal operation '1' – Reduced power
AUTO_RETRAN	1	Retransmit contents in TX register if TRX_CE and TXEN are high (default = 0). '0' – No retransmission '1' – Retransmission of data packet
RX_AFW	3	RX-address width (default = 100). '001' – 1 byte RX address field width '100' – 4 byte RX address field width
TX_AFW	3	TX-address width (default = 100). '001' – 1 byte TX address field width '100' – 4 byte TX address field width
RX_PW	6	RX-payload width (default = 100000). '000001' – 1 byte RX payload field width '000010' – 2 byte RX payload field width '100000' – 32 byte RX payload field width
TX_PW	6	TX-payload width (default = 100000). '000001' – 1 byte TX payload field width '000010' – 2 byte TX payload field width '100000' – 32 byte TX payload field width
RX_ADDRESS	32	RX address identity. Used bytes depend on RX_AFW (default = E7E7E7E7 _h).
UP_CLK_FREQ	2	Output clock frequency (default = 11). '00' – 4MHz '01' – 2MHz '10' – 1MHz '11' – 500kHz
UP_CLK_EN	1	Output clock enable (default = 1). '0' – No external clock signal available '1' – External clock signal enabled
XOF	3	Crystal oscillator frequency. Must be set according to external crystal resonant-frequency (default = 100). '000' – 4MHz '001' – 8MHz '010' – 12MHz '011' – 16MHz '100' – 20MHz
CRC_EN	1	CRC – check enable (default = 1). '0' – Disable '1' – Enable
CRC_MODE	1	CRC – mode (default = 1). '0' – 8 CRC check bit '1' – 16 CRC check bit

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

6.5 Register Contents

RF-CONFIG REGISTER (R/W)		
Byte #	Content bit[7:0], MSB = bit[7]	Init value
0	CH_NO[7:0]	0110_1100
1	bit[7:6] not used, AUTO_RETRAN, RX_RED_PWR, PA_PWR[1:0], HFREQ_PLL, CH_NO[8]	0000_0000
2	bit[7] not used, TX_AFW[2:0], bit[3] not used, RX_AFW[2:0]	0100_0100
3	bit[7:6] not used, RX_PW[5:0]	0010_0000
4	bit[7:6] not used, TX_PW[5:0]	0010_0000
5	RX_ADDRESS (device identity) byte 0	E7
6	RX_ADDRESS (device identity) byte 1	E7
7	RX_ADDRESS (device identity) byte 2	E7
8	RX_ADDRESS (device identity) byte 3	E7
9	CRC_MODE, CRC_EN, XOF[2:0], UP_CLK_EN, UP_CLK_FREQ[1:0]	1110_0111

TX PAYLOAD (R/W)		
Byte #	Content bit[7:0], MSB = bit[7]	Init value
0	TX_PAYLOAD[7:0]	X
1	TX_PAYLOAD[15:8]	X
-	-	X
-	-	X
30	TX_PAYLOAD[247:240]	X
31	TX_PAYLOAD[255:248]	X

TX ADDRESS (R/W)		
Byte #	Content bit[7:0], MSB = bit[7]	Init value
0	TX_ADDRESS[7:0]	E7
1	TX_ADDRESS[15:8]	E7
2	TX_ADDRESS[23:16]	E7
3	TX_ADDRESS[31:24]	E7

RX PAYLOAD (R)		
Byte #	Content bit[7:0], MSB = bit[7]	Init value
0	RX_PAYLOAD[7:0]	X
1	RX_PAYLOAD[15:8]	X
	-	X
	-	X
30	RX_PAYLOAD[247:240]	X
31	RX_PAYLOAD[255:248]	X

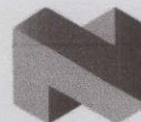
STATUS REGISTER (R)		
Byte #	Content bit[7:0], MSB = bit[7]	Init value
0	AM, bit [6] not used, DR, bit [0:4] not used	X

Table 10 RF register contents.

The length of all registers is fixed. However, the bytes in TX_PAYLOAD, RX_PAYLOAD, TX_ADDRESS and RX_ADDRESS used in ShockBurst™ RX/TX are set in the configuration register.

ไม่ว่ากรณีใดก็ตาม ผู้ออกพิมพ์เอกสารนี้ขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

7 IMPORTANT TIMING DATA

The following timing must be obeyed during nRF905 operation.

7.1 Device Switching Times

nRF905 timing	Max.
PWR_DWN_ST_BY mode	3 ms
STBY_TX ShockBurst™	650 μs
STBY_RX ShockBurst™	650 μs
RX ShockBurst™_TX ShockBurst™	550 μs
TX ShockBurst™_RX ShockBurst™	550 μs

Notes to table:

- 1) RX to TX or TX to RX switching is available without re-programming of the RF configuration register. The same frequency channel is maintained.

Table 11 Switching times for nRF905.

7.2 ShockBurst™ TX timing

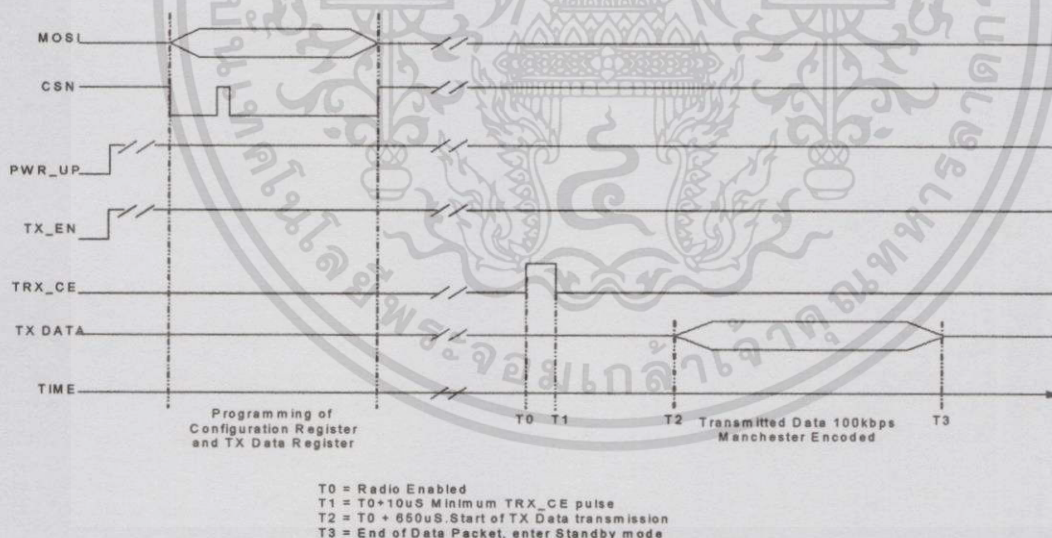
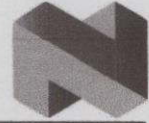


Figure 9 Timing diagram for standby to transmit.

After a data packet has finished transmitting the device will automatically enter Standby mode and wait for the next pulse of TRX_CE. If the Auto Re-Transmit function is enabled the data packet will continue re-sending the same data packet until TRX_CE is set low.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

7.3 ShockBurst™ RX timing

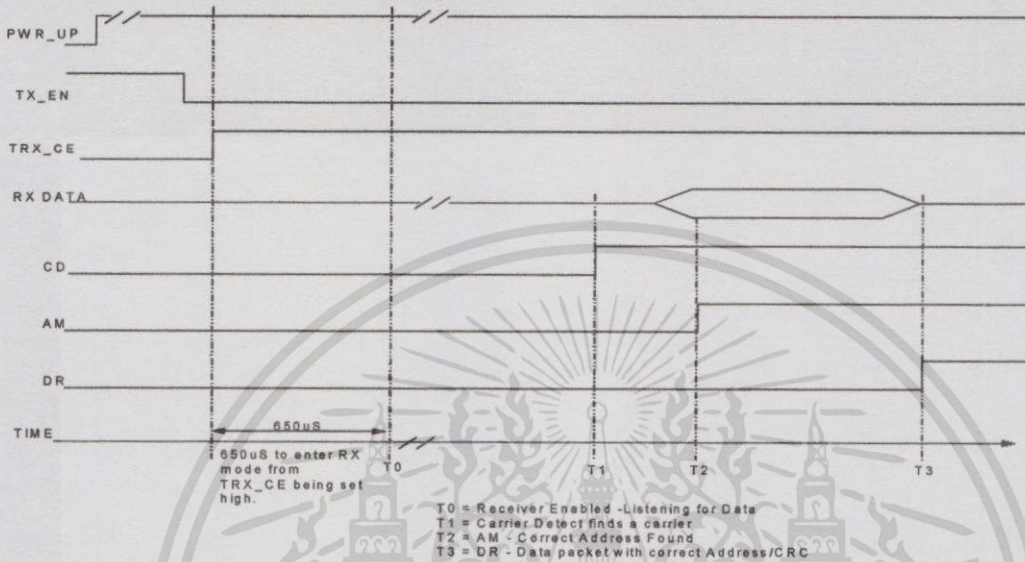


Figure 10 Timing diagram for standby to receiving.

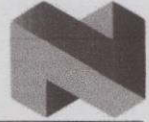
After the Data Ready (DR) has been set high a valid data packet is available in the RX data register. This may be clocked out in RX mode or standby mode. After the data has been clocked out via the SPI interface the Data Ready (DR) and Address Match (AM) pins are reset to low.

The RX register is reset if the PWR_UP pin is taken low or if the device is switched into TX mode i.e. TXEN is taken high. This will also result in the Data Ready (DR) and Address Match (AM) pins being reset to low.

7.4 Preamble

In each data-packet transmitted by the nRF905 a preamble is added automatically. The preamble is a predefined bit-sequence used to adjust the receiver for optimal performance. A ten-bit sequence is used as preamble in nRF905. The length of the preamble, t_{preamble} , is then 200µs.

PRODUCT SPECIFICATION



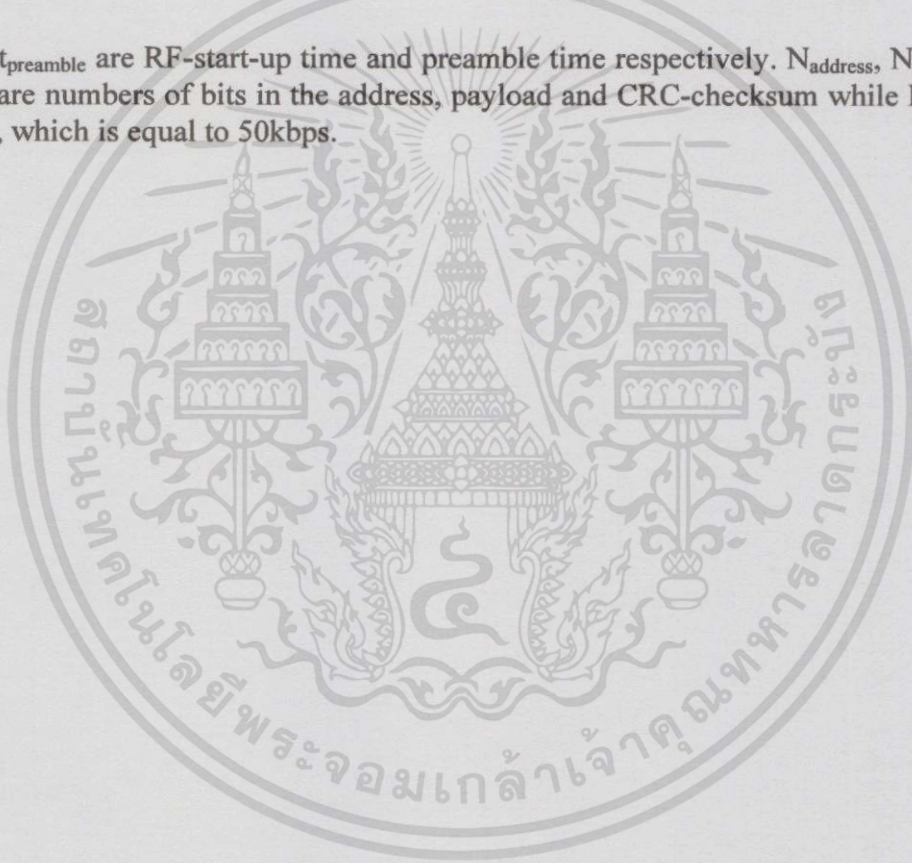
nRF905 Single Chip 433/868/915 MHz Radio Transceiver

7.5 Time On Air

The time-on-air is the sum of the radio start-up time and the data-packet length. The length of the preamble, address field, payload and CRC-checksum give the data-packet length while the radio start-up time is given in Table 11. While preamble length and start-up time are fixed the user sets the other parameters in the RF-configuration register. The below equation shows how to calculate TOA

$$TOA = t_{startup} + t_{preamble} + \frac{N_{address} + N_{payload} + N_{CRC}}{BR}$$

$t_{startup}$ and $t_{preamble}$ are RF-start-up time and preamble time respectively. $N_{address}$, $N_{payload}$ and N_{CRC} are numbers of bits in the address, payload and CRC-checksum while BR is the bitrate, which is equal to 50kbps.



PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

8 PERIPHERAL RF INFORMATION

8.1 Crystal Specification

Tolerance includes initially accuracy and tolerance over temperature and aging.

Frequency	C_L	ESR	C_{0max}	Tolerance @ 868/915 MHz	Tolerance @ 433 MHz
4MHz	8pF – 16pF	150 Ω	7.0pF	± 30 ppm	± 60 ppm
8MHz	8pF – 16pF	100 Ω	7.0pF	± 30 ppm	± 60 ppm
12MHz	8pF – 16pF	100 Ω	7.0pF	± 30 ppm	± 60 ppm
16MHz	8pF – 16pF	100 Ω	7.0pF	± 30 ppm	± 60 ppm
20MHz	8pF – 16pF	100 Ω	7.0pF	± 30 ppm	± 60 ppm

Table 12 Crystal specification of nRF905.

To achieve a crystal oscillator solution with low power consumption and fast start-up time, it is recommended to specify the crystal with a low value of crystal load capacitance. Specifying a lower value of crystal parallel equivalent capacitance, $C_o=1.5$ pF is also good, but this can increase the price of the crystal itself. Typically $C_o=1.5$ pF at a crystal specified for $C_{o_max}=7.0$ pF.

$$C_L = \frac{C_o}{2},$$

The crystal load capacitance, C_L , is given by:

$$C_L = \frac{C_1 \times C_2'}{C_1 + C_2'} \quad \text{where } C_1' = C_1 + C_{PCB1} + C_{I1} \text{ and } C_2' = C_2 + C_{PCB2} + C_{I2}$$

C_1 and C_2 are 0603 SMD capacitors as shown in the application schematics. C_{PCB1} and C_{PCB2} are the layout parasitic on the circuit board. C_{I1} and C_{I2} are the capacitance seen into the XC1 and XC2 pin respectively; the value is typical 1pF.

8.2 External Clock Reference

An external reference clock, such as a MCU clock, may be used instead of a crystal. The clock signal should be applied directly to the XC1 pin, the XC2 pin can be left high impedance. When operating with an external clock instead of a crystal the clock must be applied in standby mode to achieve low current consumption. If the device is set into standby mode with no external clock or crystal then the current consumption will increase up to a maximum of 1mA.

8.3 Microprocessor Output Clock

By default a microprocessor clock output is provided. Providing an output clock will increase the current consumption in standby mode. The current consumption in standby will depend on frequency and load of external crystal, frequency of output clock and capacitive load of the provided output clock. Typical current consumption values are found in Table 5.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

8.4 Antenna Output

The “ANT1 & ANT2” output pins provide a balanced RF output to the antenna. The pins must have a DC path to VDD_PA, either via a RF choke or via the center point in a dipole antenna. The load impedance seen between the ANT1/ANT2 outputs should be in the range 200-700Ω . The optimum differential load impedance at the antenna ports is given as:

900MHz	225Ω +j210
430MHz	300Ω +j100

A low load impedance (for instance 50Ω) can be obtained by fitting a simple matching network or a RF transformer (balun). Further information regarding balun structures and matching networks may be found in the Application Examples chapter.

8.5 Output Power Adjustment

The power amplifier in nRF905 can be programmed to four different output power settings by the configuration register. By reducing output power, the total TX current is reduced.

Power setting	RF output power	DC current consumption
00	-10 dBm	9.0 mA
01	-2 dBm	14.0 mA
10	6 dBm	20.0 mA
11	10 dBm	30.0 mA

Conditions: VDD = 3.0V, VSS = 0V, T_A = 27°C, Load impedance = 400 Ω .

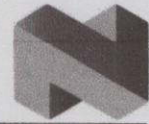
Table 13 RF output power setting for the nRF905.

8.6 Modulation

The modulation of nRF905 is Gaussian Frequency Shift Keying (GFSK) with a data-rate of 100kbps. Deviation is ±50kHz. GFSK modulation results in a more bandwidth effective transmission-link compared with ordinary FSK modulation.

The data is internally Manchester encoded (TX) and Manchester decoded (RX). That is, the effective symbol-rate of the link is 50kbps. By using internally Manchester encoding, no scrambling in the microcontroller is needed.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

8.7 Output Frequency

The operating RF-frequency of nRF905 is set in the configuration register by CH_NO and HFREQ_PLL. The operating frequency is given by:

$$f_{OP} = (422.4 + (CH_NO/10)) (1 + HFREQ_PLL) \text{ MHz}$$

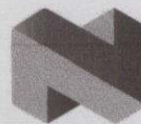
When HFREQ_PLL is '0' the frequency resolution is 100kHz and when it is '1' the resolution is 200kHz.

The application operating frequency has to be chosen to apply with the Short Range Devise regulation in the area of operation.

Operating frequency	HFREQ PLL	CH NO
430.0 MHz	[0]	[001001100]
433.1 MHz	[0]	[001101011]
433.2 MHz	[0]	[001101100]
434.7 MHz	[0]	[001111011]
862.0 MHz	[1]	[001010110]
868.2 MHz	[1]	[001110101]
868.4 MHz	[1]	[001110110]
869.8 MHz	[1]	[001111101]
902.2 MHz	[1]	[100011111]
902.4 MHz	[1]	[100100000]
927.8 MHz	[1]	[110011111]

Table 14 Examples of real operating frequencies.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

8.8 PCB Layout and Decoupling Guidelines

nRF905 is an extremely robust RF device due to internal voltage regulators and requires the minimum of RF layout protocols. However the following design rules should still be incorporated into the layout design.

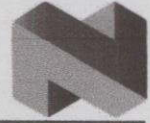
A PCB with a minimum of two layers including a ground plane is recommended for optimum performance. The nRF905 DC supply voltage should be decoupled as close as possible to the VDD pins with high performance RF capacitors. It is preferable to mount a large surface mount capacitor (e.g. 4.7 μ F tantalum) in parallel with the smaller value capacitors. The nRF905 supply voltage should be filtered and routed separately from the supply voltages of any digital circuitry.

Long power supply lines on the PCB should be avoided. All device grounds, VDD connections and VDD bypass capacitors must be connected as close as possible to the nRF905 IC. For a PCB with a topside RF ground plane, the VSS pins should be connected directly to the ground plane. For a PCB with a bottom ground plane, the best technique is to place via holes as close as possible to the VSS pins. A minimum of one via hole should be used for each VSS pin.

Full swing digital data or control signals should not be routed close to the crystal or the power supply lines.

A fully qualified RF-layout for the nRF905 and its surrounding components, including antennas and matching networks, can be downloaded from www.nordicsemi.no.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

9 nRF905 FEATURES

9.1 Carrier Detect.

When the nRF905 is in ShockBurst™ RX, the Carrier Detect (CD) pin is set high if a RF carrier is present at the channel the device is programmed to. This feature is very effective to avoid collision of packets from different transmitters operating at the same frequency. Whenever a device is ready to transmit it could first be set into receive mode and sense whether or not the wanted channel is available for outgoing data. This forms a very simple listen before transmit protocol. Operating Carrier Detect (CD) with Reduced RX Power mode is an extremely power efficient RF system. Typical Carrier Detect level (CD) is typically 5dB lower than sensitivity, i.e. if sensitivity is -100dBm then the Carrier Detect function will sense a carrier wave as low as -105dBm . Below -105dBm the Carrier Detect signal will be low, i.e. 0V. Above -95dBm the Carrier Detect signal will be high, i.e. Vdd. Between approximately -95 to -105 the Carrier Detect Signal will toggle.

9.2 Address Match

When the nRF905 is in ShockBurst™ RX mode, the Address Match (AM) pin is set high as soon as an incoming packet with an address that is identical with the device's own identity is received. With the Address Match pin the controller is alerted that the nRF905 is receiving data actually before the Data Ready (DR) signal is set high. If the Data Ready (DR) pin is not set high i.e. the CRC is incorrect then the Address Match (AM) pin is reset to low at the end of the received data packet. This function can be very useful for an MCU. If Address Match (AM) is high then the MCU can make a decision to wait and see if Data Ready (DR) will be set high indicating a valid data packet has been received or ignore that a possible packet is being received and switch modes.

9.3 Data Ready

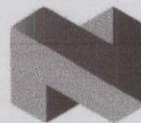
The Data Ready (DR) signal makes it possible to largely reduce the complexity of the MCU software program.

In ShockBurst™ TX, the Data Ready (DR) signal is set high when a complete packet is transmitted, telling the MCU that the nRF905 is ready for new actions. It is reset to low at the start of a new packet transmission or when switched to a different mode i.e. receive mode or standby mode.

In ShockBurst™ TX Auto Retransmit the Data Ready (DR) signal is set high at the beginning of the pre-amble and is set low at the end of the preamble. The Data Ready (DR) signal therefore pulses at the beginning of each transmitted data packet.

In ShockBurst™ RX, the signal is set high when nRF905 has received a valid packet, i.e. a valid address, packet length and correct CRC. The MCU can then retrieve the payload via the SPI interface. The Data Ready (DR) pin is reset to low once the data has been clocked out of the data buffer or the device is switched to transmit mode.

PRODUCT SPECIFICATION



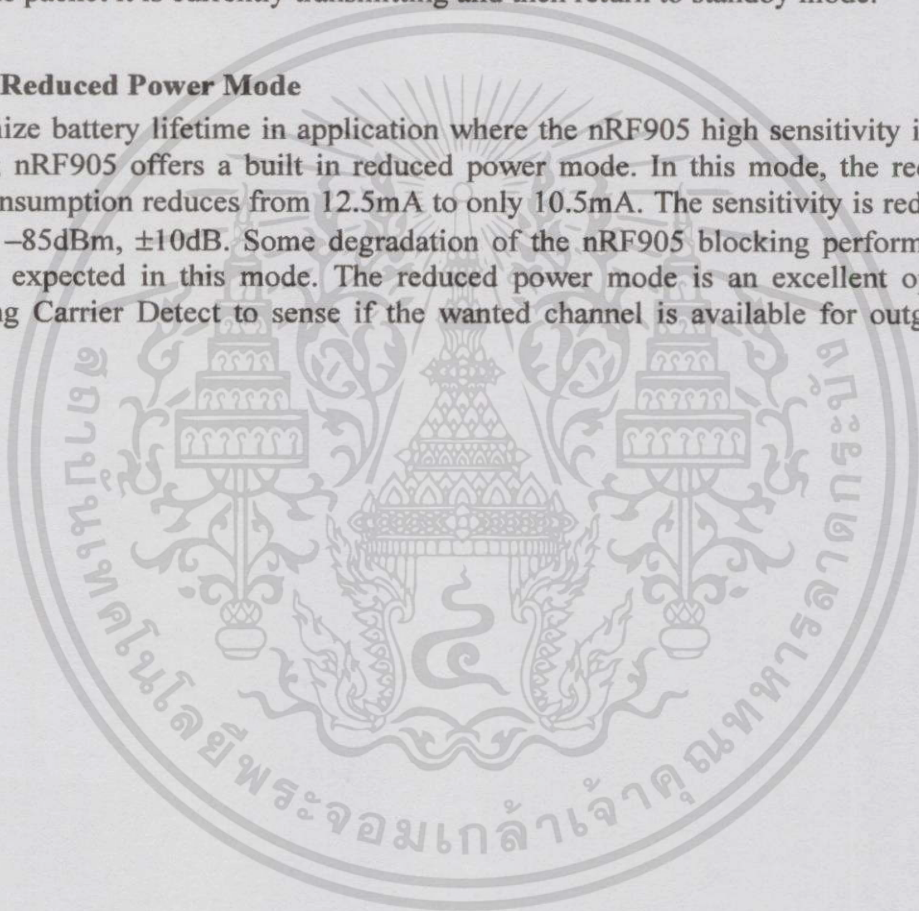
nRF905 Single Chip 433/868/915 MHz Radio Transceiver

9.4 Auto Retransmit

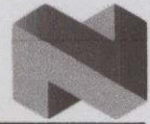
One way to increase system reliability in a noisy environment or in a system without collision control is to transmit a packet several times. This is easily accomplished with the Auto Retransmit feature in nRF905. By setting the AUTO_RETRAN bit to “1” in the configuration register, the circuit keeps sending the same data packet as long as TRX_CE and TX_EN are high. As soon as TRX_CE is set low the device will finish sending the packet it is currently transmitting and then return to standby mode.

9.5 RX Reduced Power Mode

To maximize battery lifetime in application where the nRF905 high sensitivity is not necessary; nRF905 offers a built in reduced power mode. In this mode, the receive current consumption reduces from 12.5mA to only 10.5mA. The sensitivity is reduced to typical -85dBm , $\pm 10\text{dB}$. Some degradation of the nRF905 blocking performance should be expected in this mode. The reduced power mode is an excellent option when using Carrier Detect to sense if the wanted channel is available for outgoing data.



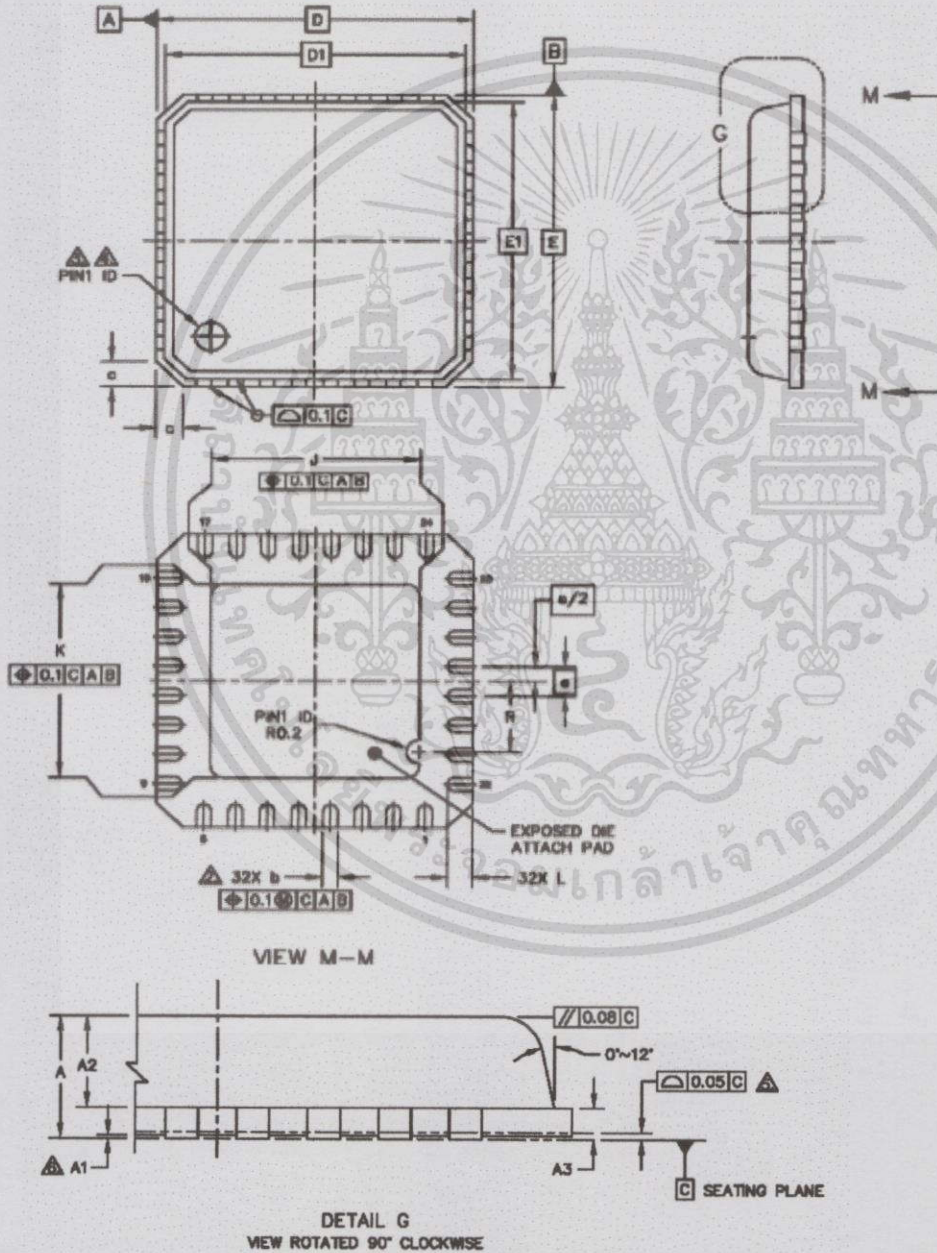
PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

10 PACKAGE OUTLINE

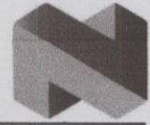
nRF905 uses the QFN 32L 5x5 green package with a mat tin finish. Dimensions are in mm. Recommended soldering reflow profile can be found in application note nAN400-08, QFN soldering reflow guidelines, www.nordicsemi.no.



Package Type		A	A ₁	A ₂	b	D	E	e	J	K	L					
QFN32 (5x5 mm)	Min	0.8	0.0	0.65	0.18	5 BSC	5 BSC	0.5 BSC	3.2	3.2	0.3					
	typ.				0.23											
	Max	0.9	0.05	0.69	0.3										3.3	3.3
											0.5					

Figure 11 nRF905 package outlines.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

10.1 Package marking

n	R	F		B	X
D	D	D	D	D	D
Y	Y	W	W	L	L

Figure 12 nRF905 package marking layout

Abbreviations:

- DDDDDD – Product number, e.g. 905
 B – Build Code, i.e. unique code for silicon revision, production site, package type and test platform
 X – "X" grade, i.e. Engineering Samples (optional)
 YY – 2 digit Year number
 WW – 2 digit Week number
 LL – 2 letter wafer lot number code

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

11.2 PCB Layout Example; Differential Connection to a Loop Antenna

Figure 14 shows a PCB layout example for the application schematic in Figure 13. A double-sided FR-4 board of 1.6mm thickness is used. This PCB has a ground plane on the bottom layer. Additionally, there are ground areas on the component side of the board to ensure sufficient grounding of critical components. A large number of via holes connect the top layer ground areas to the bottom layer ground plane. There is no ground plane beneath the antenna.

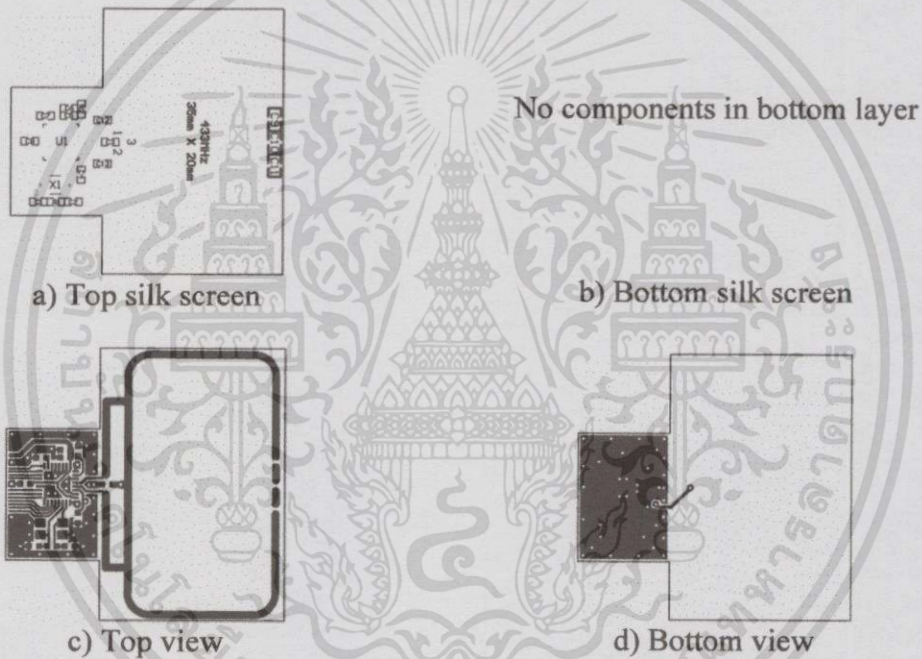
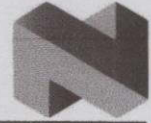


Figure 14 PCB layout example for nRF905, differential connection to a loop antenna.

A fully qualified RF-layout for the nRF905 and its surrounding components, including antennas and matching networks, can be downloaded from www.nordicsemi.no.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

11.3 Single ended connection to 50Ω antenna

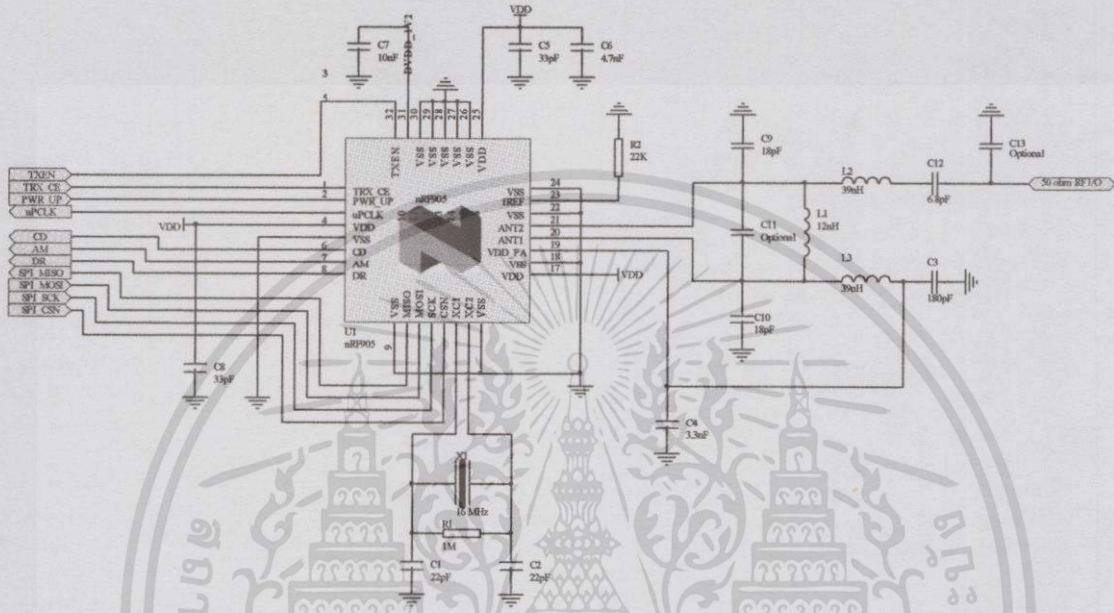


Figure 15 433MHz operating nRF905 Application schematic, single ended connection to 50Ω antenna by using a differential to single ended matching network.

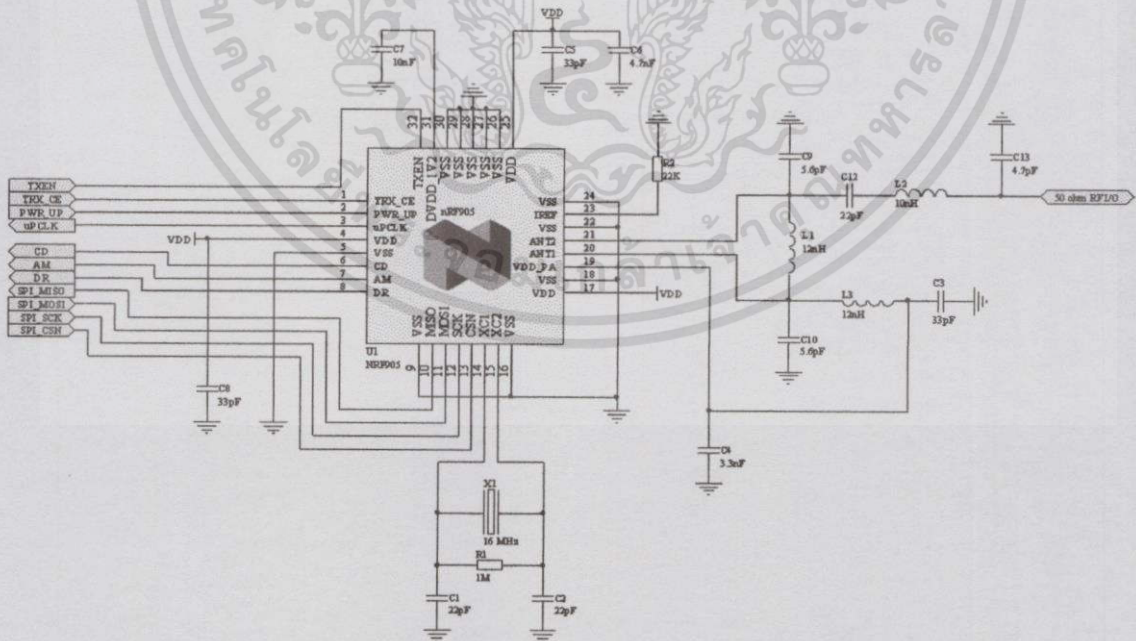


Figure 16 868-915MHz operating nRF905 Application schematic, single ended connection to 50Ω antenna by using a differential to single ended matching network.

It is recommended to add pull up or pull down resistors on signals that can enter a floating state. For the nRF905 it is recommended to have pull up on the CSN signal

ไม่วางกรณเดจทางสน ออกทงทามมเหตดแปลงเนอห่า และตององงดงเจาของเอกสารทุกค้รทมการนำป้ใช้

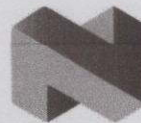
and pull down on the MOSI and SCK signal.

Main office: Nordic Semiconductor ASA - Vestre Rosten 81, N-7075 Tiller, Norway - Phone +4772898900 - Fax +4772898989
Revision: 1.4 Page 34 of 42 June 2006



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

Component	Description	Size	Value	Tol.	Units
C1	NP0 ceramic chip capacitor, (Crystal oscillator)	0603	22	±5%	pF
C2	NP0 ceramic chip capacitor, (Crystal oscillator)	0603	22	±5%	pF
C3	NP0 ceramic chip capacitor, (PA supply decoupling) @ 433MHz @ 868MHz @ 915MHz	0603	180 33 33	±5%	pF
C4	X7R ceramic chip capacitor, (PA supply decoupling)	0603	3.3	±10%	nF
C5	NP0 ceramic chip capacitor, (Supply decoupling)	0603	33	±5%	pF
C6	X7R ceramic chip capacitor, (Supply decoupling)	0603	4.7	±10%	nF
C7	X7R ceramic chip capacitor, (Supply decoupling)	0603	10	±10%	nF
C8	NP0 ceramic chip capacitor, (Supply decoupling)	0603	33	±5%	pF
C9	NP0 ceramic chip capacitor, (Impedance matching) @ 433MHz @ 868MHz @ 915MHz	0603	18 5.6 5.6	±5% <±0.25pF <±0.25pF	pF
C10	NP0 ceramic chip capacitor, (Impedance matching) @ 433MHz @ 868MHz @ 915MHz	0603	18 5.6 5.6	±5% <±0.25pF <±0.25pF	pF
C11	NP0 ceramic chip capacitor, (Impedance matching)	0603	Not fitted		pF
C12	NP0 ceramic chip capacitor, (Impedance matching) @ 433MHz @ 868MHz @ 915MHz	0603	6.8 22 22	±5% ±5% ±5%	pF
C13	NP0 ceramic chip capacitor, (Impedance matching) @ 433MHz @ 868MHz @ 915MHz	0603	Not fitted 4.7 4.7	<±0.25pF <±0.25pF	pF
L1	Chip inductor, (Impedance matching) @ 433MHz: SRF> 433MHz @ 868MHz: SRF> 868MHz @ 915MHz: SRF> 915MHz	0603	12 12 12	±5%	nH
L2	Chip inductor, (Impedance matching) @ 433MHz: SRF> 433MHz @ 868MHz: SRF> 868MHz @ 915MHz: SRF> 915MHz	0603	39 10 10	±5% ±5% ±5%	nH
L3	Chip inductor, (Impedance matching) @ 433MHz: SRF> 433MHz @ 868MHz: SRF> 868MHz @ 915MHz: SRF> 915MHz	0603	39 12 12	±5% ±5% ±5%	nH
R1	0.1W chip resistor, (Crystal oscillator bias)	0603	1	±5%	MΩ
R2	0.1W chip resistor, (Reference bias)	0603	22	±1%	kΩ
U1	nRF905 Transceiver	QFN32L/5x5			
X1	Crystal, C _L = 12pF	LxWxH = 4.0x2.5x0.8	16	±30ppm	MHz

Table 16 Recommended external components, single ended connection to 50Ω antenna.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

11.4 PCB Layout Example; Single Ended Connection to 50Ω Antenna

Figure 17 shows a PCB layout example for the application schematic in Figure 15 and Figure 18 shows a PCB layout example for the application schematic in Figure 16. A double-sided FR-4 board of 1.6mm thickness is used. This PCB has a ground plane on the bottom layer. Additionally, there are ground areas on the component side of the board to ensure sufficient grounding of critical components. A large number of via holes connect the top layer ground areas to the bottom layer ground plane.

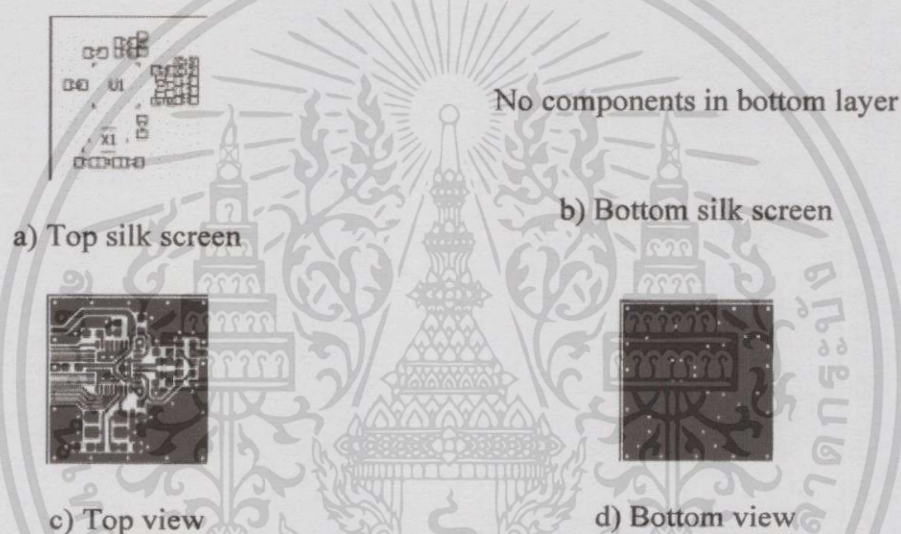
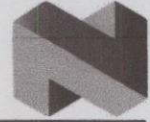
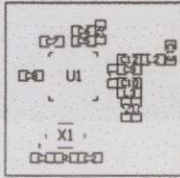


Figure 17 PCB layout example for 433MHz operation nRF905, single ended connection to 50Ω antenna by using a differential to single ended matching network.

PRODUCT SPECIFICATION



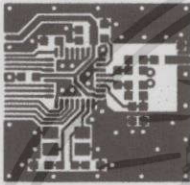
nRF905 Single Chip 433/868/915 MHz Radio Transceiver



No components in bottom layer

a) Top silk screen

b) Bottom silk screen



c) Top view

d) Bottom view

Figure 18 PCB layout example for 868-915MHz operation nRF905, single ended connection to 50Ω antenna by using a differential to single ended matching network.

A fully qualified RF-layout for the nRF905 and its surrounding components, including antennas and matching networks, can be downloaded from www.nordicsemi.no.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

12 ABSOLUTE MAXIMUM RATINGS

Supply Voltages

VDD - 0.3V to + 3.6V

VSS 0V

Input Voltage

V_I - 0.3V to VDD + 0.3V

Output Voltage

V_O - 0.3V to VDD + 0.3V

Total Power Dissipation

P_D (T_A=85°C) 200mW

Temperatures

Operating temperature - 40°C to + 85°C

Storage temperature - 40°C to + 125°C

Note: Stress exceeding one or more of the limiting values may cause permanent damage to the device.

ATTENTION!

Electrostatic sensitive device.

Observe precaution for handling.



PRODUCT SPECIFICATION



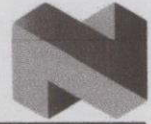
nRF905 Single Chip 433/868/915 MHz Radio Transceiver

13 GLOSSARY OF TERMS

Term	Description
ADC	Analog to Digital Converter
AM	Address Match
CD	Carrier Detect
CLK	Clock
CRC	Cyclic Redundancy Check
DR	Data Ready
GFSK	Gaussian Frequency Shift Keying
ISM	Industrial-Scientific-Medical
kSPS	kilo Samples per Second
MCU	Micro Controller Unit
PWR_DWN	Power Down
PWR_UP	Power Up
RX	Receive
SPI	Serial Programmable Interface
CSN	SPI Chip Select Not
MISO	SPI Master In Slave Out
MOSI	SPI Master Out Slave In
SCK	SPI Serial Clock
SPS	Samples per Second
STBY	Standby
TRX_EN	Transmit/Receive Enable
TX	Transmit
TX_EN	Transmit Enable

Table 17 Glossary of terms.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

14 DEFINITIONS

Product Specification Identification	Product Status Definition
Objective Product Specification	Planned or Under Development. This specification contains the design objectives for product development. nRF: Specifications may change in any manner without notice.
Preliminary Product Specification	Engineering Samples and Pre Production series. This specification contains preliminary data. nRF: Nordic Semiconductor reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product Specification	The product is qualified for production. Changes will be notified according to industry standard criteria for Product/Process Change Notifications.
Obsolete Product Specification	Not In Production. This specification contains specifications on a product that has been discontinued by Nordic Semiconductor. The specification is printed for reference information only.

Table 18 Product status definitions

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor does not assume any liability arising out of the application or use of any product or circuits described herein.

LIFE SUPPORT APPLICATIONS

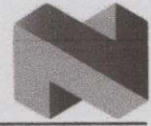
These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

Product specification revision date: 23.06.2006

Datasheet order code: 230606nRF905

All rights reserved ®. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

Nordic Semiconductor ASA – World Wide Distributors

For Your nearest dealer, please see <http://www.nordicsemi.no>



Main Office:

Vestre Rosten 81, N-7075 Tiller, Norway

Phone: +47 72 89 89 00, Fax: +47 72 89 89 89

Visit the Nordic Semiconductor ASA website at <http://www.nordicsemi.no>



Main office: Nordic Semiconductor ASA - Vestre Rosten 81, N-7075 Tiller, Norway - Phone +4772898900 - Fax +4772898989
 Revision: 1.4 Page 42 of 42 June 200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.3 คู่มือ GPS Module

NEO-6

u-blox 6 GPS

Modules Data Sheet

Abstract

Technical data sheet describing the cost-effective, high-performance u-blox 6 based NEO-6 series of GPS modules, that brings the high performance of the u-blox 6 positioning engine to the miniature NEO form factor.

These receivers combine a high level of integration capability with flexible connectivity options in a small package. This makes them perfectly suited for mass-market end products with strict size and cost requirements.



16.0 x 12.2 x 2.4 mm



Document Information

Title	NEO-6
Subtitle	u-blox 6 GPS Modules
Document type	Data Sheet
Document number	GPS.G6-HW-09005-E

Document status

Document status information

Objective Specification	This document contains target values. Revised and supplementary data will be published later.
Advance Information	This document contains data based on early testing. Revised and supplementary data will be published later.
Preliminary	This document contains data from product verification. Revised and supplementary data may be published later.
Released	This document contains the final product specification.

This document applies to the following products:

Name	Type number	ROM/FLASH version	PCN reference
NEO-6G	NEO-6G-0-001	ROM7.03	UBX-TN-11047-1
NEO-6Q	NEO-6Q-0-001	ROM7.03	UBX-TN-11047-1
NEO-6M	NEO-6M-0-001	ROM7.03	UBX-TN-11047-1
NEO-6P	NEO-6P-0-000	ROM6.02	N/A
NEO-6V	NEO-6V-0-000	ROM7.03	N/A
NEO-6T	NEO-6T-0-000	ROM7.03	N/A

This document and the use of any information contained therein, is subject to the acceptance of the u-blox terms and conditions. They can be downloaded from www.u-blox.com.

u-blox makes no warranties based on the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Reproduction, use or disclosure to third parties without express permission is strictly prohibited. Copyright © 2011, u-blox AG.

u-blox® is a registered trademark of u-blox Holding AG in the EU and other countries. ARM® is the registered trademark of ARM Limited in the EU and other countries.



Contents

Contents	3
1 Functional description	5
1.1 Overview	5
1.2 Product features	5
1.3 GPS performance.....	6
1.4 Block diagram.....	7
1.5 Assisted GPS (A-GPS).....	7
1.6 AssistNow Autonomous	7
1.7 Precision Timing.....	8
1.7.1 Time mode	8
1.7.2 Timepulse and frequency reference	8
1.7.3 Time mark	8
1.8 Raw data	8
1.9 Automotive Dead Reckoning	8
1.10 Precise Point Positioning.....	9
1.11 Oscillators	9
1.12 Protocols and interfaces	9
1.12.1 UART.....	9
1.12.2 USB	9
1.12.3 Serial Peripheral Interface (SPI).....	9
1.12.4 Display Data Channel (DDC).....	10
1.13 Antenna.....	10
1.14 Power Management	10
1.14.1 Maximum Performance Mode	10
1.14.2 Eco Mode.....	10
1.14.3 Power Save Mode	11
1.15 Configuration	11
1.15.1 Boot-time configuration	11
1.16 Design-in	11
2 Pin Definition	12
2.1 Pin assignment	12
3 Electrical specifications	14
3.1 Absolute maximum ratings	14
3.2 Operating conditions	15
3.3 Indicative power requirements.....	15
3.4 SPI timing diagrams	16
3.4.1 Timing recommendations.....	16
4 Mechanical specifications	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 เมื่อกฎหมายที่เกี่ยวข้องเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5	Qualification and certification	18
5.1	Reliability tests	18
5.2	Approvals	18
6	Product handling & soldering	19
6.1	Packaging	19
6.1.1	Reels	19
6.1.1	Tapes	20
6.2	Moisture Sensitivity Levels	21
6.3	Reflow soldering	21
6.4	ESD handling precautions	21
7	Default settings	22
8	Labeling and ordering information	23
8.1	Product labeling	23
8.2	Explanation of codes	23
8.3	Ordering information	24
	Related documents	24
	Revision history	24
	Contact	25



1 Functional description

1.1 Overview

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.

The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

1.2 Product features

Model	Type		Supply		Interfaces			Features										
	GPS	PPP	Timing	Raw Data	Dead Reckoning	1.75 V - 2.0 V	2.7 V - 3.6 V	UART	USB	SPI	DDC (I ² C compliant)	Programmable (Flash) RW update	TCXO	RTC crystal	Antenna supply and supervisor	Configuration pins	Timepulse	External interrupt/Wakeup
NEO-6G	•				•	•	•	•	•	•	•	•	•	•	•	3	1	•
NEO-6Q	•				•	•	•	•	•	•	•	•	•	•	•	3	1	•
NEO-6M	•				•	•	•	•	•	•	•	•	•	•	•	3	1	•
NEO-6P	•	•		•	•	•	•	•	•	•	•	•	•	•	•	3	1	•
NEO-6V	•				•	•	•	•	•	•	•	•	•	•	•	3	1	•
NEO-6T	•		•	•	•	•	•	•	•	•	•	•	•	•	•	3	1	•

○ = Requires external components and integration on application processor

Table 1: Features of the NEO-6 Series

All NEO-6 modules are based on GPS chips qualified according to AEC-Q100. See Chapter 5.1 for further information.



1.3 GPS performance

Parameter	Specification			
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix ¹		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Cold Start ²	26 s	27 s	32 s
	Warm Start ²	26 s	27 s	32 s
	Hot Start ²	1 s	1 s	1 s
	Aided Starts ³	1 s	<3 s	<3 s
Sensitivity ⁴		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Tracking & Navigation	-162 dBm -	-161 dBm	-160 dBm
	Reacquisition ⁵	160 dBm -	-160 dBm	-160 dBm
	Cold Start (without aiding)	148 dBm -	-147 dBm	-146 dBm
	Hot Start	157 dBm -	-156 dBm	-155 dBm
Maximum Navigation update rate		NEO-6G/Q/M/T	NEO-6P/V	
		5Hz	1 Hz	
Horizontal position accuracy ⁶	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP ⁷	< 1 m (2D, R50) ⁸		
	SBAS + PPP ⁷	< 2 m (3D, R50) ⁸		
Configurable Timepulse frequency range		NEO-6G/Q/M/P/V	NEO-6T	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated ⁹	15 ns		
Velocity accuracy ⁶		0.1m/s		
Heading accuracy ⁶		0.5 degrees		
Operational Limits	Dynamics	≤ 4 g		
	Altitude ¹⁰	50,000 m		
	Velocity ¹⁰	500 m/s		

Table 2: NEO-6 GPS performance

¹ All satellites at -130 dBm

² Without aiding

³ Dependent on aiding data connection speed and latency

⁴ Demonstrated with a good active antenna

⁵ For an outage duration ≤10s

⁶ CEP, 50%, 24 hours static, -130dBm, SEP: <3.5m

⁷ NEO-6P only

⁸ Demonstrated under following conditions: 24 hours, stationary, first 600 seconds of data discarded. HDOP < 1.5 during measurement period, strong signals. Continuous availability of valid SBAS correction data during full test period.

⁹ Quantization error information can be used with NEO-6T to compensate the granularity related error of the timepulse signal

¹⁰ Assuming Airborne <4g platform

1.4 Block diagram

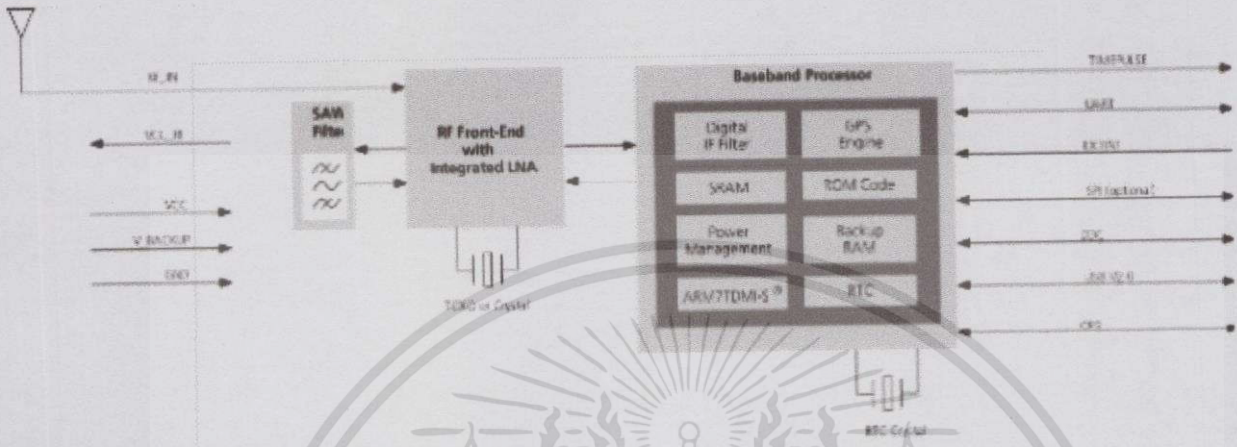


Figure 1: Block diagram (For available options refer to the product features table in section 1.2).

1.5 Assisted GPS (A-GPS)

Supply of aiding information like ephemeris, almanac, rough last position and time and satellite status and an optional time synchronization signal will reduce time to first fix significantly and improve the acquisition sensitivity. All NEO-6 modules support the u-blox AssistNow Online and AssistNow Offline A-GPS services¹¹ and are OMA SUPL compliant.

1.6 AssistNow Autonomous

AssistNow Autonomous provides functionality similar to Assisted-GPS without the need for a host or external network connection. Based on previously broadcast satellite ephemeris data downloaded to and stored by the GPS receiver, AssistNow Autonomous automatically generates accurate satellite orbital data (“AssistNow Autonomous data”) that is usable for future GPS position fixes. AssistNow Autonomous data is reliable for up to 3 days after initial capture.

u-blox’ AssistNow Autonomous benefits are:

- Faster position fix
- No connectivity required
- Complementary with AssistNow Online and Offline services
- No integration effort, calculations are done in the background

For more details see the u-blox 6 Receiver Description including Protocol Specification [2].

¹¹ AssistNow Offline requires external memory.



1.7 Precision Timing

1.7.1 Time mode

NEO-6T provides a special Time Mode to provide higher timing accuracy. The NEO-6T is designed for use with stationary antenna setups. The Time Mode features three different settings described in Table 3: Disabled, Survey-In and Fixed Mode. For optimal performance entering the position of the antenna (when known) is recommended as potential source of errors will be reduced.

Time Mode Settings	Description
Disabled	Standard PVT operation
Survey-In	The GPS receiver computes the average position over an extended time period until a predefined maximum standard deviation has been reached. Afterwards the receiver will be automatically set to Fixed Mode and the timing features will be activated.
Fixed Mode	In this mode, a fixed 3D position and known standard deviation is assumed and the timing features are activated. Fixed Mode can either be activated directly by feeding pre-defined position coordinates (ECEF - Earth Center Earth Fixed format) or by performing a Survey-In. In Fixed mode, the timing errors in the TIMEPULSE signal which otherwise result from positioning errors are eliminated. Single-satellite operation is supported. For details, please refer to the <i>u-blox 6 Receiver Description including Protocol Specification [2]</i> .

Table 3: Time mode settings

1.7.2 Timepulse and frequency reference

NEO-6T comes with a timepulse output which can be configured from 0.25 Hz up to 10 MHz. The timepulse can either be used for time synchronization (i.e. 1 pulse per second) or as a reference frequency in the MHz range. A timepulse in the MHz range provides excellent long-term frequency accuracy and stability.

1.7.3 Time mark

NEO-6T can be used for precise time measurements with sub-microsecond resolution using the external interrupt (EXTINT0). Rising and falling edges of these signals are time-stamped to the GPS or UTC time and counted. The Time Mark functionality can be enabled with the UBX-CFG-TM2 message

For details, please refer to the *u-blox 6 Receiver Description including Protocol Specification [2]*.

1.8 Raw data

Raw data output is supported at an update rate of 5 Hz on the NEO-6T and NEO-6P. The UBX-RXM-RAW message includes carrier phase with half-cycle ambiguity resolved, code phase and Doppler measurements, which can be used in external applications that offer precision positioning, real-time kinematics (RTK) and attitude sensing.

1.9 Automotive Dead Reckoning

Automotive Dead Reckoning (ADR) is u-blox' industry proven off-the-shelf Dead Reckoning solution for tier-one automotive customers. u-blox' ADR solution combines GPS and sensor digital data using a tightly coupled Kalman filter. This improves position accuracy during periods of no or degraded GPS signal.

The NEO-6V provides ADR functionality over its software sensor interface. A variety of sensors (such as wheel ticks and gyroscope) are supported, with the sensor data received via UBX messages from the application processor. This allows for easy integration and a simple hardware interface, lowering costs. By using digital sensor data available on the vehicle bus, hardware costs are minimized since no extra sensors are required for Dead Reckoning functionality. ADR is designed for simple integration and easy configuration of different sensor options (e.g. with or without gyroscope) and vehicle variants, and is completely self-calibrating.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



For more details contact the u-blox support representative nearest you to receive dedicated *u-blox 6 Receiver Description Including Protocol Specification* [3].

1.10 Precise Point Positioning

u-blox' industry proven PPP algorithm provides extremely high levels of position accuracy in static and slow moving applications, and makes the NEO-6P an ideal solution for a variety of high precision applications such as surveying, mapping, marine, agriculture or leisure activities.

Ionospheric corrections such as those received from local SBAS¹² geostationary satellites (WAAS, EGNOS, MSAS) or from GPS enable the highest positioning accuracy with the PPP algorithm. The maximum improvement of positioning accuracy is reached with PPP+SBAS and can only be expected in an environment with unobstructed sky view during a period in the order of minutes.

1.11 Oscillators

NEO-6 GPS modules are available in Crystal and TCXO versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

1.12 Protocols and interfaces

Protocol	Type
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0)
UBX	Input/output, binary, u-blox proprietary
RTCM	Input, 2.3

Table 4: Available protocols

All listed protocols are available on UART, USB, and DDC. For specification of the various protocols see the *u-blox 6 Receiver Description including Protocol Specification* [2].

1.12.1 UART

NEO-6 modules include one configurable UART interface for serial communication (for information about configuration see section 1.15).

1.12.2 USB

NEO-6 modules provide a USB version 2.0 FS (Full Speed, 12Mbit/s) interface as an alternative to the UART. The pull-up resistor on USB_DP is integrated to signal a full-speed device to the host. The VDDUSB pin supplies the USB interface. u-blox provides a Microsoft® certified USB driver for Windows XP, Windows Vista and Windows 7 operating systems.

1.12.3 Serial Peripheral Interface (SPI)

The SPI interface allows for the connection of external devices with a serial interface, e.g. serial flash to save configuration and AssistNow Offline A-GPS data or to interface to a host CPU. The interface can be operated in master or slave mode. In master mode, one chip select signal is available to select external slaves. In slave mode a single chip select signal enables communication with the host.

The maximum bandwidth is 100kbit/s.

¹² Satellite Based Augmentation System

1.12.4 Display Data Channel (DDC)

The I²C compatible DDC interface can be used either to access external devices with a serial interface EEPROM or to interface with a host CPU. It is capable of master and slave operation. The DDC interface is I²C Standard Mode compliant. For timing parameters consult the I²C standard.

The DDC Interface supports serial communication with u-blox wireless modules. See the specification of the applicable wireless module to confirm compatibility.

The maximum bandwidth is 100kbit/s.

1.12.4.1 External serial EEPROM

NEO-6 modules allow an optional external serial EEPROM to be connected to the DDC interface. This can be used to store Configurations permanently.

For more information see the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].

Use caution when implementing since forward compatibility is not guaranteed.

1.13 Antenna

NEO-6 modules are designed for use with passive and active¹³ antennas.

Parameter	Specification	
Antenna Type	Passive and active antenna	
Active Antenna Recommendations	Minimum gain	15 dB (to compensate signal loss in RF cable)
	Maximum gain	50 dB
	Maximum noise figure	1.5 dB

Table 5: Antenna Specifications for all NEO-6 modules

1.14 Power Management

u-blox receivers support different power modes. These modes represent strategies of how to control the acquisition and tracking engines in order to achieve either the best possible performance or good performance with reduced power consumption.

For more information about power management strategies, see the *u-blox 6 Receiver Description including Protocol Specification* [2].

1.14.1 Maximum Performance Mode

During a Cold start, a receiver in Maximum Performance Mode continuously deploys the acquisition engine to search for all satellites. Once the receiver has a position fix (or if pre-positioning information is available), the acquisition engine continues to be used to search for all visible satellites that are not being tracked.

1.14.2 Eco Mode

During a Cold start, a receiver in Eco Mode works exactly as in Maximum Performance Mode. Once a position can be calculated and a sufficient number of satellites are being tracked, the acquisition engine is powered off resulting in significant power savings. The tracking engine continuously tracks acquired satellites and acquires other available or emerging satellites.

Note that even if the acquisition engine is powered off, satellites continue to be acquired.

¹³ For information on using active antennas with NEO-6 modules, see the *LEA-6/NEO-6 Hardware Integration Manual* [1].

1.14.3 Power Save Mode

Power Save Mode (PSM) allows a reduction in system power consumption by selectively switching parts of the receiver on and off.



Power Save mode is not available with NEO-6P, NEO-6T and NEO-6V.

1.15 Configuration

1.15.1 Boot-time configuration

NEO-6 modules provide configuration pins for boot-time configuration. These become effective immediately after start-up. Once the module has started, the configuration settings can be modified with UBX configuration messages. The modified settings remain effective until power-down or reset. If these settings have been stored in battery-backup RAM, then the modified configuration will be retained, as long as the backup battery supply is not interrupted.

NEO-6 modules include both **CFG_COM0** and **CFG_COM1** pins and can be configured as seen in Table 6. Default settings in bold.

CFG_COM1	CFG_COM0	Protocol	Messages	UARTBaud rate	USB power
1	1	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT	9600	BUS Powered
1	0	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT	38400	Self Powered
0	1	NMEA	GSV¹⁴, RMC, GSA, GGA, VTG, TXT	4800	BUS Powered
0	0	UBX	NAV-SOL, NAV-STATUS, NAV-SVINFORM, NAV-CLOCK, INF, MON-EXCEPT, AID-ALPSERV	57600	BUS Powered

Table 6: Supported COM settings

NEO-6 modules include a **CFG_GPS0** pin, which enables the boot-time configuration of the power mode. These settings are described in Table 7. Default settings in bold.

CFG_GPS0	Power Mode
0	Eco Mode
1	Maximum Performance Mode

Table 7: Supported CFG_GPS0 settings



Static activation of the **CFG_COM** and **CFG_GPS** pins is not compatible with use of the SPI interface.

1.16 Design-in

In order to obtain the necessary information to conduct a proper design-in, u-blox strongly recommends consulting the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].

¹⁴ Every 5th fix.



2 Pin Definition

2.1 Pin assignment

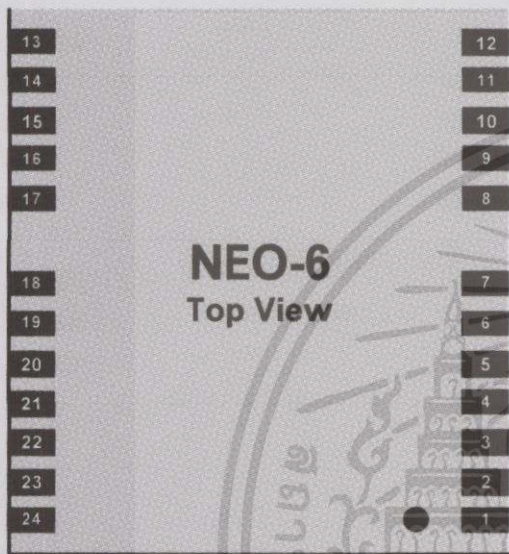



Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SPI Slave Select
3	All	TIMEPULSE	O	Timepulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage RF section Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MOSI/CFG_COM0	O/I	SPI MOSI / Configuration Pin. Leave open if not used.
15	All	MISO/CFG_COM1	I	SPI MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPS0/SCK	I	Power Mode Configuration Pin / SPI Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	DDC Data
19	All	SCL2	I/O	DDC Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1

No	Module	Name	I/O	Description
22	All	V_BCKP	I	Backup voltage supply
23	All	VCC	I	Supply voltage
24	All	GND	I	Ground

Table 8: Pinout

 Pins designated Reserved should not be used. For more information about Pinouts see the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].



3 Electrical specifications

3.1 Absolute maximum ratings

Parameter	Symbol	Module	Min	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	-0.5	2.0	V	
		NEO-6Q, 6M, 6P, 6V, 6T	-0.5	3.6	V	
Backup battery voltage	V_BCKP	All	-0.5	3.6	V	
USB supply voltage	VDDUSB	All	-0.5	3.6	V	
Input pin voltage	Vin	All	-0.5	3.6	V	
	Vin_usb	All	-0.5	VDDU SB	V	
DC current trough any digital I/O pin (except supplies)	Ipin			10	mA	
VCC_RF output current	ICC_RF	All		100	mA	
Input power at RF_IN	Prfin	NEO-6Q, 6M, 6G, 6V, 6T		15	dBm	source impedance = 50Ω, continuous wave
		NEO-6P		-5	dBm	
Storage temperature	Tstg	All	-40	85	°C	

Table 9: Absolute maximum ratings

GPS receivers are Electrostatic Sensitive Devices (ESD) and require special precautions when handling. For more information see chapter 6.4.

Stressing the device beyond the “Absolute Maximum Ratings” may cause permanent damage. These are stress ratings only. The product is not protected against overvoltage or reversed voltages. If necessary, voltage spikes exceeding the power supply voltage specification, given in table above, must be limited to values within the specified boundaries by using appropriate protection diodes. For more information see the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].

3.2 Operating conditions

All specifications are at an ambient temperature of 25°C.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	1.75	1.8	1.95	V	
		NEO-6Q/M	2.7	3.0	3.6	V	
		NEO-6P/V/T					
Supply voltage USB	VDDUSB	All	3.0	3.3	3.6	V	
Backup battery voltage	V_BCKP	All	1.4		3.6	V	
Backup battery current	I_BCKP	All		22		µA	V_BCKP = 1.8 V, VCC = 0V
Input pin voltage range	Vin	All	0		VCC	V	
Digital IO Pin Low level input voltage	Vil	All	0		0.2*VCC	V	
Digital IO Pin High level input voltage	Vih	All	0.7*VCC		VCC	V	
Digital IO Pin Low level output voltage	Vol	All			0.4	V	Iol=4mA
Digital IO Pin High level output voltage	Voh	All	VCC - 0.4			V	Ioh=4mA
USB_DM, USB_DP	VinU	All	Compatible with USB with 22 Ohms series resistance				
VCC_RF voltage	VCC_RF	All		VCC-0.1		V	
VCC_RF output current	ICC_RF	All			50	mA	
Antenna gain	Gant	All			50	dB	
Receiver Chain Noise Figure	NFtot	All		3.0		dB	
Operating temperature	Topr	All	-40		85	°C	

Table 10: Operating conditions

Operation beyond the specified operating conditions can affect device reliability.

3.3 Indicative power requirements

Table 11 lists examples of the total system supply current for a possible application.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Max. supply current ¹⁵	Iccp	All			67	mA	VCC = 3.6 V ¹⁶ / 1.95 V ¹⁷
		Icc Acquisition	All		47 ¹⁸	mA	
		Icc Tracking (Max Performance mode)	NEO-6G/Q/T NEO-6M/P/V		40 ²⁰ 39 ²⁰	mA	
Average supply current ¹⁸	Icc Tracking (Eco mode)	NEO-6G/Q/T		38 ²⁰		mA	VCC = 3.0 V ¹⁶ / 1.8 V ¹⁷
		NEO-6M/P/V		37 ²⁰		mA	
		Icc Tracking (Power Save mode / 1 Hz)	NEO-6G/Q NEO-6M		12 ²⁰ 11 ²⁰	mA	

Table 11: Indicative power requirements

Values in Table 11 are provided for customer information only as an example of typical power requirements. Values are characterized on samples, actual power requirements can vary depending on FW version used, external circuitry, number of SVs tracked, signal strength, type of start as well as time, duration and conditions of test.

¹⁵ Use this figure to dimension maximum current capability of power supply. Measurement of this parameter with 1 Hz bandwidth.

¹⁶ NEO-6Q, NEO-6M, NEO-6P, NEO-6V, NEO-6T

¹⁷ NEO-6G

¹⁸ Use this figure to determine required battery capacity.

¹⁹ >8 SVs in view, CNo >40 dBHz, current average of 30 sec after cold start.

²⁰ With strong signals, all orbits available. For Cold Starts typical 12 min after first fix. For Hot Starts typical 15 s after first fix.

3.4 SPI timing diagrams

In order to avoid a faulty usage of the SPI, the user needs to comply with certain timing conditions. The following signals need to be considered for timing constraints:

Symbol	Description
SS_N	Slave Select signal
SCK	Slave Clock signal

Table 12: Symbol description



Figure 3: SPI timing diagram

3.4.1 Timing recommendations

Parameter	Description	Recommendation
t_{INIT}	Initialization Time	500 μ s
t_{DES}	Deselect Time	1 ms
Bitrate		100 kbit/s

Table 13: SPI timing recommendations

The values in the above table result from the requirement of an error-free transmission. By allowing just a few errors, the byte rate could be increased considerably. These timings – and therefore the byte rate – could also be improved by disabling other interfaces, e.g. the UART.

The maximum bandwidth is 100 kbit/s²¹.

²¹ This is a theoretical maximum, the protocol overhead is not considered.

4 Mechanical specifications

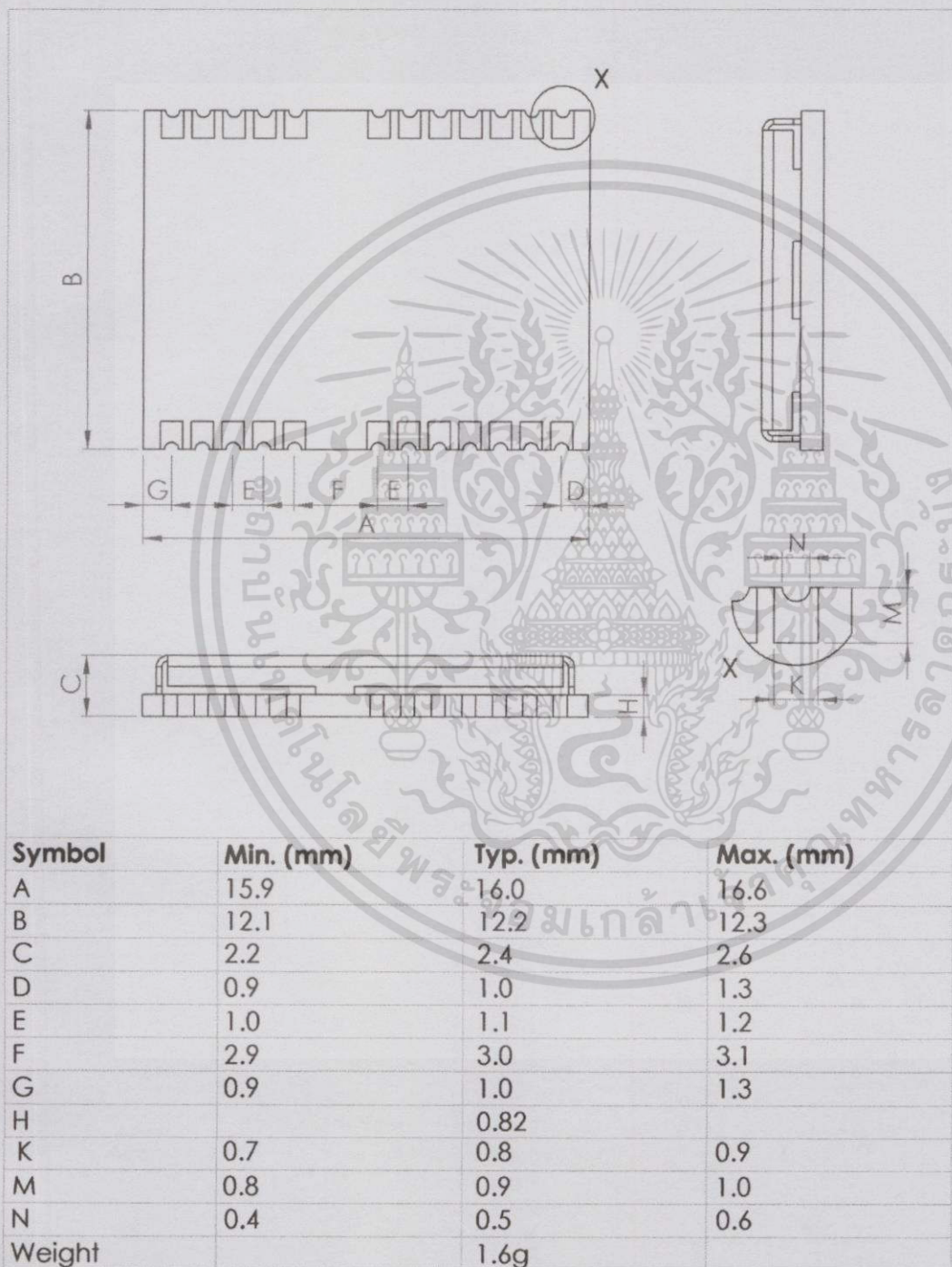



Figure 4: Dimensions

For information regarding the Paste Mask and Footprint see the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].

5 Qualification and certification

5.1 Reliability tests

 All NEO-6 modules are based on AEC-Q100 qualified GPS chips.

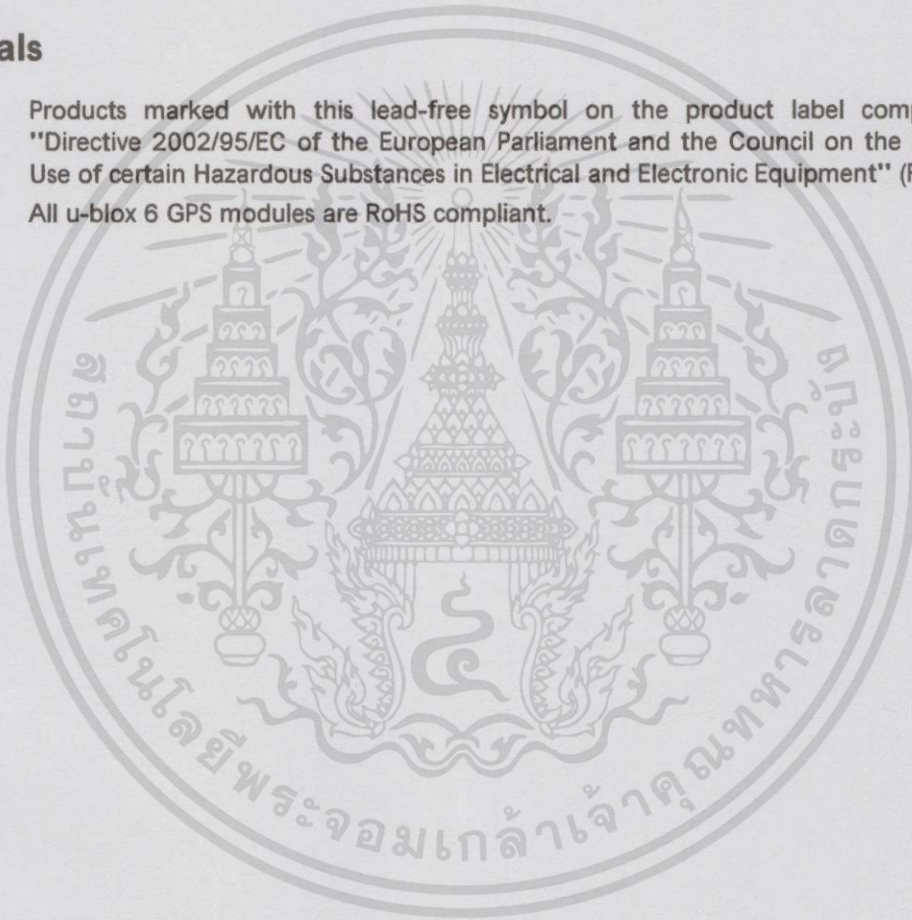
Tests for product family qualifications according to ISO 16750 "Road vehicles - Environmental conditions and testing for electrical and electronic equipment", and appropriate standards.

5.2 Approvals



Products marked with this lead-free symbol on the product label comply with the "Directive 2002/95/EC of the European Parliament and the Council on the Restriction of Use of certain Hazardous Substances in Electrical and Electronic Equipment" (RoHS).

All u-blox 6 GPS modules are RoHS compliant.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

6 Product handling & soldering

6.1 Packaging

NEO-6 modules are delivered as hermetically sealed, reeled tapes in order to enable efficient production, production lot set-up and tear-down. For more information about packaging, see the *u-blox Package Information Guide* [4].

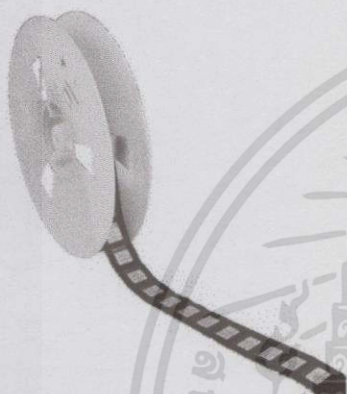


Figure 5: Reeled u-blox 6 modules

6.1.1 Reels

NEO-6 GPS modules are deliverable in quantities of 250pcs on a reel. NEO-6 modules are delivered using reel Type B as described in the *u-blox Package Information Guide* [4].

Parameter	Specification
Reel Type	B
Delivery Quantity	250

Table 14: Reel information for NEO-6 modules

6.1.1 Tapes

Figure 6 shows the position and orientation of NEO-6 modules as they are delivered on tape. The dimensions of the tapes are specified in Figure 7.

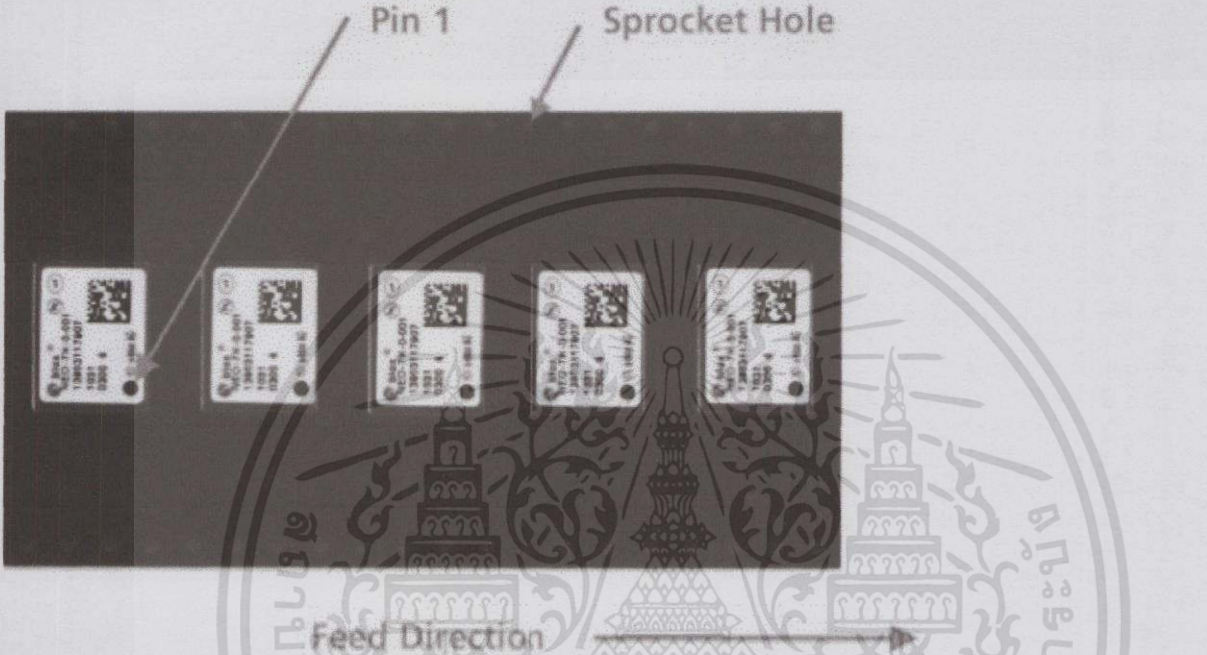


Figure 6: Orientation for NEO-6 modules on tape

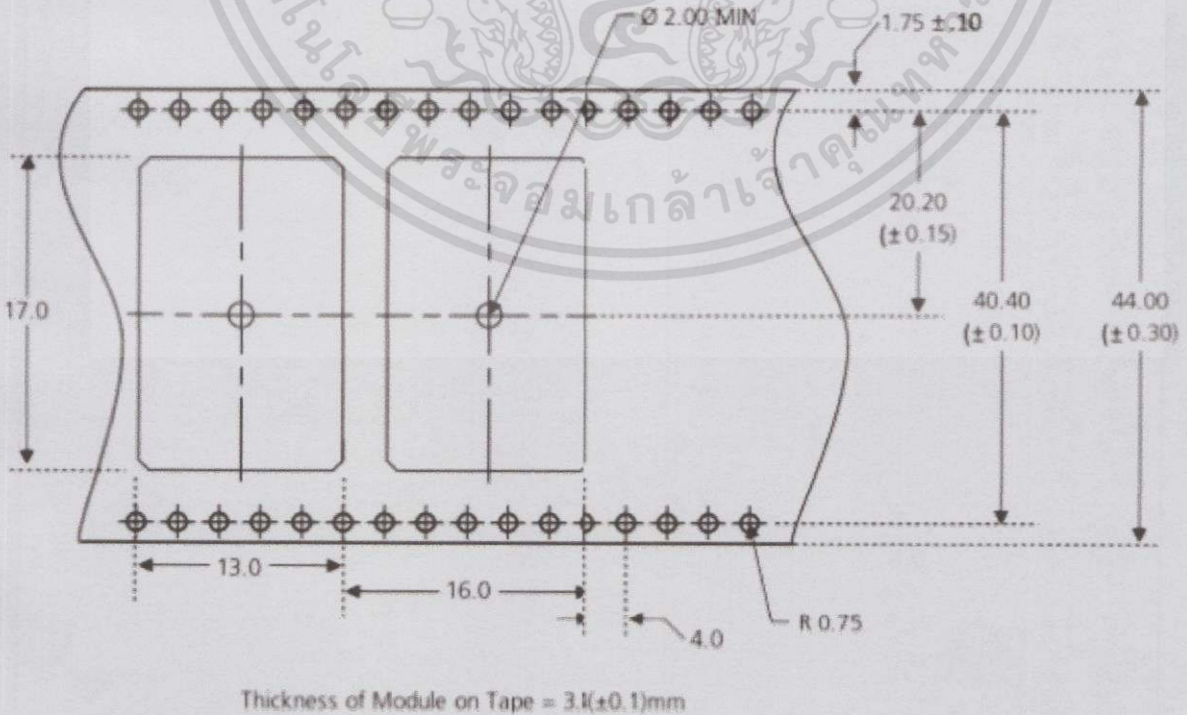


Figure 7: NEO tape dimensions (mm)



6.2 Moisture Sensitivity Levels



NEO-6 modules are **Moisture Sensitive Devices (MSD)** in accordance to the **IPC/JEDEC** specification.

NEO-6 modules are rated at MSL level 4. For more information regarding moisture sensitivity levels, labeling, storage and drying see the *u-blox Package Information Guide* [4].



For MSL standard see IPC/JEDEC J-STD-020, which can be downloaded from www.jedec.org.

6.3 Reflow soldering

Reflow profiles are to be selected according to u-blox recommendations (see *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1]).

6.4 ESD handling precautions

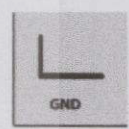
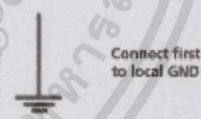


NEO-6 modules contain highly sensitive electronic circuitry and are **Electrostatic Sensitive Devices (ESD)**. Observe precautions for handling! Failure to observe these precautions can result in severe damage to the GPS receiver!

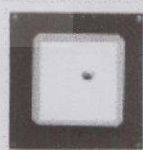
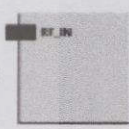


GPS receivers are **Electrostatic Sensitive Devices (ESD)** and require special precautions when handling. Particular care must be exercised when handling patch antennas, due to the risk of electrostatic charges. In addition to standard ESD safety practices, the following measures should be taken into account whenever handling the receiver:

- Unless there is a galvanic coupling between the local GND (i.e. the work table) and the PCB GND, then the first point of contact when handling the PCB must always be between the local GND and PCB GND.
- Before mounting an antenna patch, connect ground of the device
- When handling the RF pin, do not come into contact with any charged capacitors and be careful when contacting materials that can develop charges (e.g. patch antenna ~10pF, coax cable ~50-80pF/m, soldering iron, ...)
- To prevent electrostatic discharge through the RF input, do not touch any exposed antenna area. If there is any risk that such exposed antenna area is touched in non ESD protected work area, implement proper ESD protection measures in the design.
- When soldering RF connectors and patch antennas to the receiver's RF pin, make sure to use an ESD safe soldering iron (tip).



ESD Sensitive



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

7 Default settings

Interface	Settings
Serial Port 1 Output	9600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: GGA, GLL, GSA, GSV, RMC, VTG, TXT (In addition to the 6 standard NMEA messages the NEO-6T includes ZDA).
USB Output	Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: GGA, GLL, GSA, GSV, RMC, VTG, TXT (In addition to the 6 standard NMEA messages the NEO-6T includes ZDA). USB Power Mode: Bus-Powered
Serial Port 1 Input	9600 Baud, 8 bits, no parity bit, 1 stop bit Automatically accepts following protocols without need of explicit configuration: UBX, NMEA The GPS receiver supports interleaved UBX and NMEA messages.
USB Input	Automatically accepts following protocols without need of explicit configuration: UBX, NMEA The GPS receiver supports interleaved UBX and NMEA messages. USB Power Mode: Bus-Powered
TIMEPULSE (1Hz Nav)	1 pulse per second, synchronized at rising edge, pulse length 100ms
Power Mode	Maximum Performance mode
AssistNow Autonomous	Disabled.

Table 15: Default settings

Refer to the *u-blox 6 Receiver Description including Protocol Specification [2]* for information about further settings.

8 Labeling and ordering information

8.1 Product labeling

The labeling of u-blox 6 GPS modules includes important product information. The location of the product type number is shown in Figure 8.

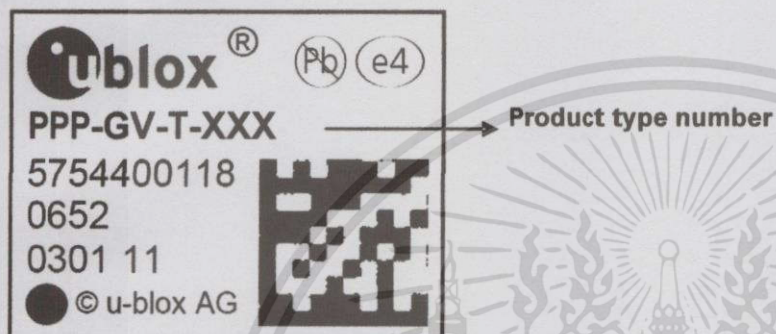


Figure 8: Location of product type number on u-blox 6 module label

8.2 Explanation of codes

3 different product code formats are used. The **Product Name** is used in documentation such as this data sheet and identifies all u-blox 6 products, independent of packaging and quality grade. The **Ordering Code** includes options and quality, while the **Type Number** includes the hardware and firmware versions. Table 16 below details these 3 different formats:

Format	Structure
Product Name	PPP-GV
Ordering Code	PPP-GV-T
Type Number	PPP-GV-T-XXX

Table 16: Product Code Formats

The parts of the product code are explained in Table 17.

Code	Meaning	Example
PPP	Product Family	NEO
G	Product Generation	6 = u-blox6
V	Variant	T = Timing, R = DR, etc.
T	Option / Quality Grade	Describes standardized functional element or quality grade such as Flash size, automotive grade etc.
XXX	Product Detail	Describes product details or options such as hard- and software revision, cable length, etc.

Table 17: part identification code

8.3 Ordering information

Ordering No.	Product
NEO-6G-0	u-blox 6 GPS Module, 1.8V, TCXO, 12x16mm, 250 pcs/reel
NEO-6M-0	u-blox 6 GPS Module, 12x16mm, 250 pcs/reel
NEO-6Q-0	u-blox 6 GPS Module, TCXO, 12x16mm, 250 pcs/reel
NEO-6P-0	u-blox 6 GPS Module, PPP, 12x16mm, 250 pcs/reel
NEO-6V-0	u-blox 6 GPS Module, Dead Reckoning SW sensor, 12x16mm, 250 pcs/reel
NEO-6T-0	u-blox 6 GPS Module, Precision Timing, TCXO, 12x16mm, 250 pcs/reel

Table 18: Product Ordering Codes

Product changes affecting form, fit or function are documented by u-blox. For a list of Product Change Notifications (PCNs) see our website at: <http://www.u-blox.com/en/notifications.html>

Related documents

- [1] LEA-6/NEO-6/MAX-6 Hardware Integration Manual, Docu. GPS.G6-HW-09007
- [2] u-blox 6 Receiver Description Including Protocol Specification (Public version), Docu. No. GPS.G6-SW-10018
- [3] u-blox 6 Receiver Description Including Protocol Specification (Confidential version), Docu. No. GPS.G6-SW-10019
- [4] u-blox Package Information Guide, Docu. No GPS-X-11004

For regular updates to u-blox documentation and to receive product change notifications please register on our homepage.

Revision history

Revision	Date	Name	Status / Comments
	31/08/2009	tgri	Initial Version
1	21/09/2009	tgri	update of section 1.3 GPS performance, section 1.4 block diagram, section 3.2 peak supply current
A	25/02/2010	tgri	Change of status to Advance Information. Addition of NEO-6G. Update of section 1.8.2, removed reference to Vddio – added USB driver certification. Update of section 3.2 table 11: average supply current, Added section 3.3-3.4: SPI & DDC timing, section 5.1: addition of table 12.
B	24/06/2010	dhur	Change of status to Preliminary. Update of section 1.2, 1.8.4, 1.10.4, 3.1, 3.2 and chapter 2 and 4. General clean-up and consistency check.
B1	11/08/2010	dhur	Replaced graphic in figure 2.
C	18/07/2011	dhur	Added chapter 1.6, update to FW7.03.
D	19/10/2011	dhur	Added NEO-6P and NEO-6V. Added chapter 1.7 and 1.8. Revised Chapter 6.
E	05/12/2011	dhur	Added NEO-6T. Added chapter 1.7 and 1.8. Added Accuracy for Timepulse signal in Table 2. Corrected Maximum Input power at RF_IN for NEO-6P in Table 9.



Contact

For complete contact information visit us at www.u-blox.com

Headquarters

u-blox AG

Zuercherstrasse 68
CH-8800 Thalwil
Switzerland

Phone: +41 44 722 74 44
Fax: +41 44 722 74 47
E-mail: info@u-blox.com

Offices

North, Central and South America

u-blox America, Inc.

Phone: +1 (703) 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast:

Phone: +1 (703) 483 3184
E-mail: info_us@u-blox.com

Technical Support:

Phone: +1 (703) 483 3185
E-mail: support_us@u-blox.com

Europe, Middle East, Africa

u-blox AG

Phone: +41 44 722 74 44
E-mail: info@u-blox.com

Technical Support:

Phone: +41 44 722 74 44
E-mail: info@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.

Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office China:

Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office Japan:

Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea:

Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan:

Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า