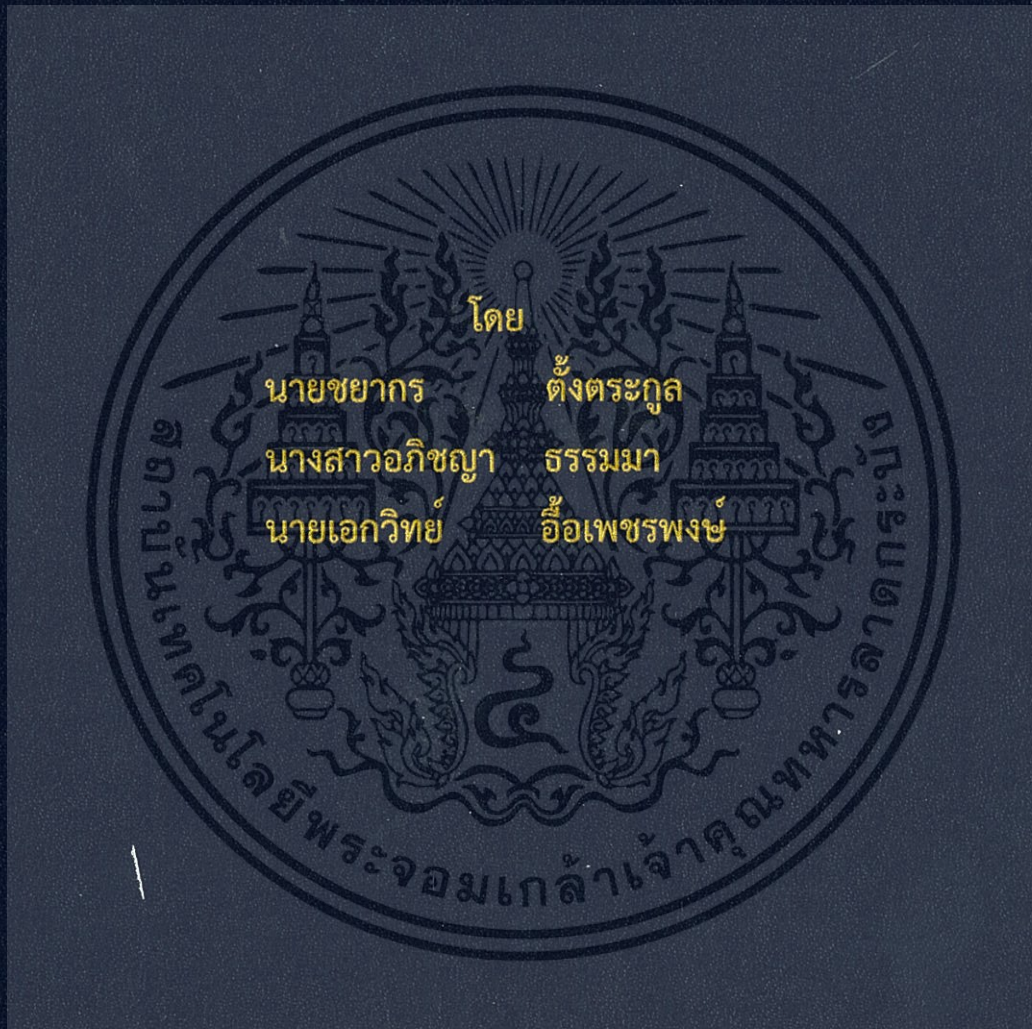


ระบบค้นหาทิศทางสายอากาศดิจิทัลที่ไวอัตโนมัติสำหรับยานพาหนะ
AUTOMATIC DIRECTION TRACKING OF DIGITAL TV ANTENNA FOR VEHICLE



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบค้นหาทิศทางสายอากาศดิจิทัลทีวีอัตโนมัติสำหรับยานพาหนะ
AUTOMATIC DIRECTION TRACKING OF DIGITAL TV ANTENNA FOR VEHICLE



T144382



โดย
นายชยากร ตั้งตระกูล
นางสาวอภิขญา ธรรมมา
นายเอกวิทย์ อ้อเพชรพงษ์

เลขหมู่.....
เลขทะเบียน **144382**
รับเดือน ปี **24** พ.ย. 2559

.b. **12819621**
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบค้นหาทิศทางสายอากาศดิจิทัลที่วัดโนมัติสำหรับยานพาหนะ
Automatic Direction Tracking of Digital TV Antenna for Vehicle

โดย

นายชยากร ตังตระกุล	55010236
นางสาวอภิขญา ธรรมมา	55011408
นายเอกวิทย์ อ้อเพชรพงษ์	55011491

อาจารย์ที่ปรึกษา

รศ.ดร. พิพัฒน์ พรหมมี

ผศ.ดร. มนตรี คำเงิน

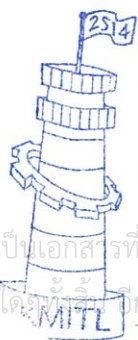
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

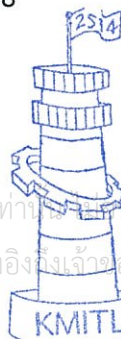


ผ่านการตรวจรูปเล่มแล้ว

(*[Signature]*)
อาจารย์ที่ปรึกษา

18/05/59

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

(*[Signature]*)
กรรมการผู้ตรวจชิ้นงาน

18/5/59

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาตเห็นแก่ประโยชน์ของการค้า

ไม่ว่ากรณีใดๆก็ตาม ผู้จัดทำขอสงวนสิทธิ์ในการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2558

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบค้นหาทิศทางสายอากาศดิจิตอลทีวีอัตโนมัติสำหรับยานพาหนะ

Automatic Direction Tracking of Digital TV Antenna for Vehicle

ผู้จัดทำ

1. นายชยากร ตั้งตระกูล 55010236
2. นางสาวอภิขญา ธรรมมา 55011408
3. นายเอกวิทย์ อ้อเพชรพงษ์ 55011491

..... อาจารย์ที่ปรึกษา
(รศ.ดร.พิพัฒน์ พรหมมี)

..... อาจารย์ที่ปรึกษา
(ผศ.ดร.มนตรี คำเงิน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานประกอบปริญญานิพนธ์เล่มนี้สำเร็จได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์จาก รศ.ดร.พิพัฒน์ พรหมมี และ ผศ.ดร.มนตรี คำเงิน อาจารย์ที่ให้คำปรึกษาโครงการที่ได้ให้คำแนะนำที่ดีตลอดมา และเอื้อเฟื้อสถานที่ในการจัดทำโครงการ ขอขอบคุณรุ่นพี่และเพื่อนๆ ทุกคนที่ให้คำปรึกษาและคอยให้ความช่วยเหลือ ที่สำคัญที่สุดขอบคุณเพื่อนในกลุ่มที่ได้ร่วมกันจัดทำปริญญานิพนธ์นี้ และได้ผ่านความยากลำบากมาด้วยกัน จนโครงการสำเร็จลุล่วงไปด้วยดี ท้ายนี้ผู้จัดทำขอขอบคุณทุกๆ ท่านที่มีส่วนเกี่ยวข้องมา ณ โอกาสนี้



นายชยากร ตั้งตระกูล
นางสาวอภิษฎา ธรรมมา
นายเอกวิทย์ อ้อเพชรพงษ์
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบค้นหาทิศทางสายอากาศดิจิทัลที่ไวอัตโนมัติสำหรับ
ยานพาหนะ

Automatic Direction Tracking of Digital TV Antenna
for Vehicle

โดย นายชยากร ตั้งตระกูล 55010236
นางสาวอภิษฎา ธรรมมา 55011408
นายเอกวิทย์ อ้อเพชรพงษ์ 55011491

อาจารย์ที่ปรึกษา รศ.ดร.พิพัฒน์ พรหมมี
ผศ.ดร.มนตรี คำเงิน

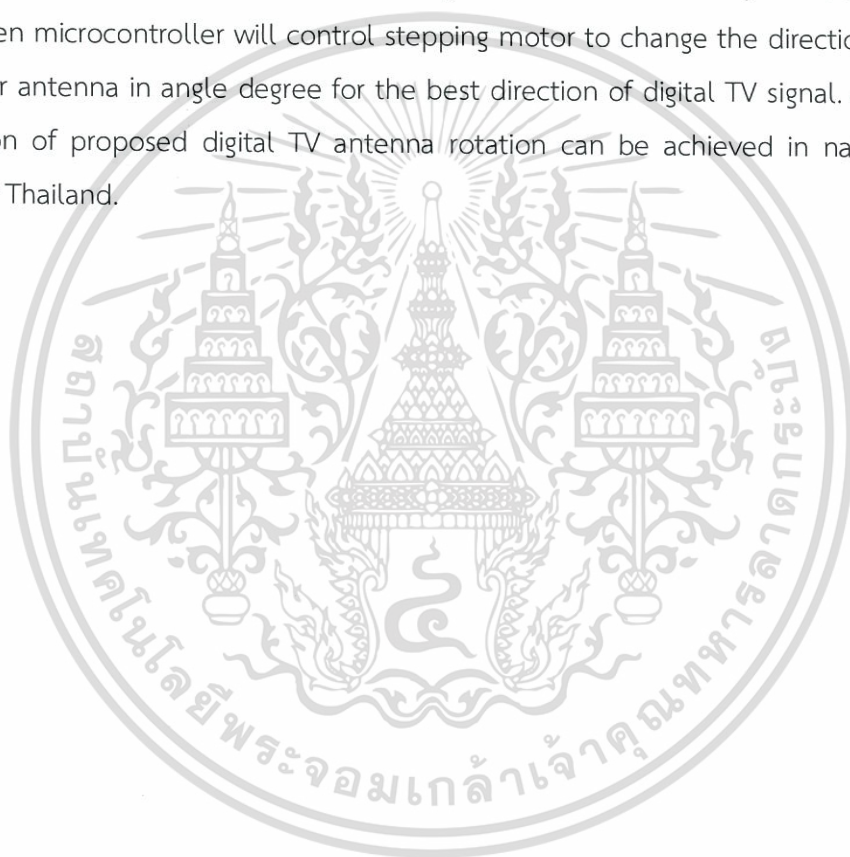
บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้จัดทำขึ้นเพื่อควบคุมเสาอากาศให้จับตำแหน่งเสาส่งสัญญาณทีวีดิจิทัล ในกรณีที่ยานพาหนะเคลื่อนที่ไปยังตำแหน่งต่างๆ การทำงานเริ่มจากโมดูลจีพีเอสส่งข้อมูลพิกัดตำแหน่งของยานพาหนะ แล้วนำไปเปรียบเทียบกับตำแหน่งและข้อมูลการกระจายสัญญาณของเสาส่งทีวีดิจิทัล ประกอบกับข้อมูลทิศทางของยานพาหนะเทียบกับทิศเหนือโดยใช้โมดูลเข็มทิศ เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ในการควบคุมสายอากาศเพื่อจับตำแหน่งเสาส่งสัญญาณที่มีความเข้มดีที่สุดโดยใช้สแต็ปปีงมอเตอร์ ซึ่งมีความแม่นยำในการหมุนหาเสาส่งสัญญาณและสามารถทำงานได้ในทุกพื้นที่ครอบคลุมสัญญาณทีวีดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

This project presents an automatic digital TV antenna based on microcontroller and transmitter location map. The direction of a digital TV antenna which located on top of the vehicle is controlled by GPS module and compass module and digital TV transmitter location data which stored in database. The current position and direction of the receiver antenna is determined by compass and GPS module. The best direction in angle will be calculated by microcontroller and then microcontroller will control stepping motor to change the direction of the receiver antenna in angle degree for the best direction of digital TV signal. Accurate direction of proposed digital TV antenna rotation can be achieved in nationwide area of Thailand.



สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 MICROCONTROLLER (ARDUINO MEGA 2560 R3)	2
2.2 สเต็ปป์มอเตอร์ (STEPPING MOTOR)	5
2.3 SD CARD MODULE	9
2.4 ULN2003	10
2.5 COMPASS MODULE (HMC5883L)	11
2.6 GLOBAL POSITIONING SYSTEM (GPS)	12
2.7 ความรู้เบื้องต้นเกี่ยวกับที่วิดิจิตอล	14
2.8 เสาอากาศ (ANTENNA)	24
2.9 การใช้งานโปรแกรม ARDUINO เบื้องต้น	26
2.10 การใช้โปรแกรม ALTUM DESIGNER เบื้องต้น	30
บทที่ 3	
การออกแบบและการจัดทำปริญญาบัตร	35
3.1 การออกแบบ	35

สารบัญ (ต่อ)

	หน้า
3.1.1 การออกแบบทางฮาร์ดแวร์	35
3.1.2 การออกแบบทางซอฟต์แวร์	38
3.2 เครื่องมือที่ใช้ในการทดลอง	42
3.3 การจัดเก็บผลการทดลอง	43
บทที่ 4 ผลการทดลอง	44
4.1 การทำงานของระบบ	44
บทที่ 5 สรุปลผลและข้อเสนอแนะ	51
5.1 สรุปลผล	51
5.2 ข้อเสนอแนะ	51
บรรณานุกรม	52
ภาคผนวก	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1.1 บอร์ด ARDUINO MEGA 2560 R3	3
2.1.2 โครงสร้างของ ARDUINO MEGA 2560 R3	4
2.2 สเต็ปป์มอเตอร์	5
2.2.2.1 การใช้มอเตอร์วัดค่าความต้านทาน	7
2.2.2.2 การต่อวงจรเพื่อทดสอบโดยการสวิตช์เพื่อหาลำดับ	8
2.4 ULN2003	10
2.5.1 COMPASS MODULE	11
2.5.2 แกนแบบ 3-AXIS ของ COMPASS MODULE	12
2.6 วงโคจรของดาวเทียมจีพีเอส	12
2.6.1 เครื่องรับสัญญาณจีพีเอส (UBLOX NEO-6M)	14
2.7 การส่งสัญญาณแบบอะนาล็อกและดิจิตอล	15
2.7.2.1 ความแตกต่างคุณภาพสัญญาณภาพตามระดับความแรงของสัญญาณที่รับได้	16
2.7.2.2 ความแตกต่างของคุณภาพสัญญาณภาพ	17
2.7.4 สถานีส่งสัญญาณที่วัดดิจิตอล (สถานีหลัก)	23
2.8 องค์กรประกอบของเสาอากาศแบบยาคี (YAGI ANTENNA)	24
2.9.1.1 หน้าต่างเริ่มต้นโปรแกรม ARDUINO	26
2.9.1.2 การตั้งค่า BOARD	27
2.9.1.3 การตั้งค่า PORT การใช้งาน	27
2.9.2.1 โปรแกรมควบคุมการทำงานของระบบ	28
2.9.2.2 การบันทึกโปรแกรมที่ใช้ในการควบคุมรูปแบบ SKETCH.INO	29
2.9.2.3 หลังจากทำการ UPLOAD โปรแกรม	29
2.10.1.1 ตัวอย่างการสร้าง NEW PROJECT	30
2.10.1.2 ตัวอย่างการสร้างพื้นที่ SCHEMATIC	31

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.10.1.3	31
2.10.2.1	32
2.10.2.2	32
2.10.2.3	33
2.10.2.4.1	33
2.10.2.4.2	34
3.1.1	35
3.1.1.1	36
3.1.1.2	37
3.1.1.3	37
3.1.1.4	38
3.1.2.1	38
3.1.2.2	39
3.1.2.3	39
3.1.2.4	40
3.1.2.1.1	41
4.1.1	44
4.1.2	45
4.1.3	45
4.1.4	46
4.1.5	46
4.1.6	47
4.1.7	48
4.1.8	49
4.1.9	50

สารบัญตาราง

ตารางที่	หน้า
2.1 รายละเอียด ARDUINO MEGA 2560 R3	2
2.2.1 การกระตุ้นเฟสแบบฟูลสเต็ป 1 เฟส (SINGLE-PHASE DRIVER)	6
2.2.2 การกระตุ้นเฟสแบบฟูลสเต็ป 2 เฟส (TWO-PHASE DRIVER)	6
2.2.3 การกระตุ้นเฟส แบบฮาล์ฟสเต็ป 2 เฟส (HALF STEP MOTOR)	7
2.3 การต่อขา SD CARD MODULE กับ ARDUINO	9
2.5 การต่อขา COMPASS MODULE กับ ARDUINO	11
2.6.1 คุณสมบัติของเครื่องรับสัญญาณจีพีเอส (UBLOX NEO-6M)	13
2.7.3.1 แสดงประเภทและรายละเอียดของสถานีส่ง	18
2.6.4.1 แสดงค่าพิกัดตำแหน่งของเสาส่งสัญญาณที่วัดจีตอล และพื้นที่การกระจายสัญญาณ	19
2.7.1 แสดงคุณสมบัติของเสาอากาศแต่ละประเภท	25

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันในประเทศไทยได้มีการนำเทคโนโลยีระบบดิจิตอลทีวี (DIGITAL TV) เข้ามาใช้ ซึ่งเป็นระบบส่งสัญญาณแบบใหม่ที่มาแทนที่การส่งแบบ ANALOG TV โดยระบบดิจิตอลมีจุดเด่นกว่าระบบอนาล็อกทั้งในด้านความคมชัดของภาพและเสียง อีกทั้งการส่งข้อมูลแบบดิจิตอลสามารถส่งข้อมูลได้มากกว่าแบบเดิม (MULTICASTING) ซึ่งนอกเหนือจากการรับชมทีวีที่บ้านแล้วก็ยังมีบริการติดตั้งทีวีระบบดิจิตอลในยานพาหนะ ปัญหาที่พบก็คือการได้รับชมภาพที่มีคุณภาพต่ำเนื่องจากในแต่ละพื้นที่นั้นมีความเข้มของสัญญาณไม่เท่ากัน

จากเหตุผลข้างต้นถ้าเราต้องการได้คุณภาพภาพที่ชัดเหมือนติดจานดาวเทียมบนรถ จะต้องมีการควบคุมสายอากาศ ผู้จัดทำจึงศึกษาและนำไมโครคอนโทรลเลอร์ (ARDUINO) มาประยุกต์ใช้งานร่วมกับโมดูลจีพีเอสและโมดูลเข็มทิศ เพื่อเก็บข้อมูลพิกัดตำแหน่งของยานพาหนะแล้วนำไปเปรียบเทียบกับตำแหน่งและข้อมูลการกระจายสัญญาณของเสาส่งสัญญาณทีวีดิจิตอลเพื่อที่จะควบคุมสายอากาศให้จับความเข้มของสัญญาณที่ดีที่สุด

1.2 วัตถุประสงค์

- 1) ศึกษาการทำงานของไมโครคอนโทรลเลอร์ (ARDUINO) โมดูลเข็มทิศและโมดูลจีพีเอส
- 2) เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ในการควบคุมสายอากาศ
- 3) เพื่อสร้างอุปกรณ์ช่วยในการรับชมทีวีระบบดิจิตอลในยานพาหนะ

1.3 ขอบเขตของโครงการ

- 1) สามารถเปรียบเทียบตำแหน่งและข้อมูลการกระจายสัญญาณของสถานีส่งสัญญาณทีวีดิจิตอลกับยานพาหนะ และออกแบบการควบคุมการทำงานของไมโครคอนโทรลเลอร์ในการควบคุมสายอากาศ
- 2) สามารถสร้างสายอากาศดิจิตอลทีวีปรับทิศทางหาสถานีส่งสัญญาณทีวีดิจิตอลได้แบบอัตโนมัติ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 MICROCONTROLLER (ARDUINO MEGA 2560 R3)

ARDUINO เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ OPEN SOURCE คือมีการเปิดเผยข้อมูลทั้งด้าน HARDWARE และ SOFTWARE ตัวบอร์ดถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้น ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้ ความง่ายของบอร์ด ARDUINO ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรมีอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (SHIELD) ประเภทต่างๆ เช่น XBEE SHIELD, MUSIC SHIELD, RELAY SHIELD, WIRELESS SHIELD, GPRS SHIELD เป็นต้น มาต่อเข้ากับบอร์ด ARDUINO แล้วเขียนโปรแกรมพัฒนาต่อได้

ตารางที่ 2.1 รายละเอียด ARDUINO MEGA 2560 R3

MICROCONTROLLER	ATMEGA2560
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMITS)	6-20V
DIGITAL I/O PINS	54 (OF WHICH 14 PROVIDE PWM OUTPUT , 4 UART TTL)
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

2.1.1 จุดเด่นของ ARDUINO

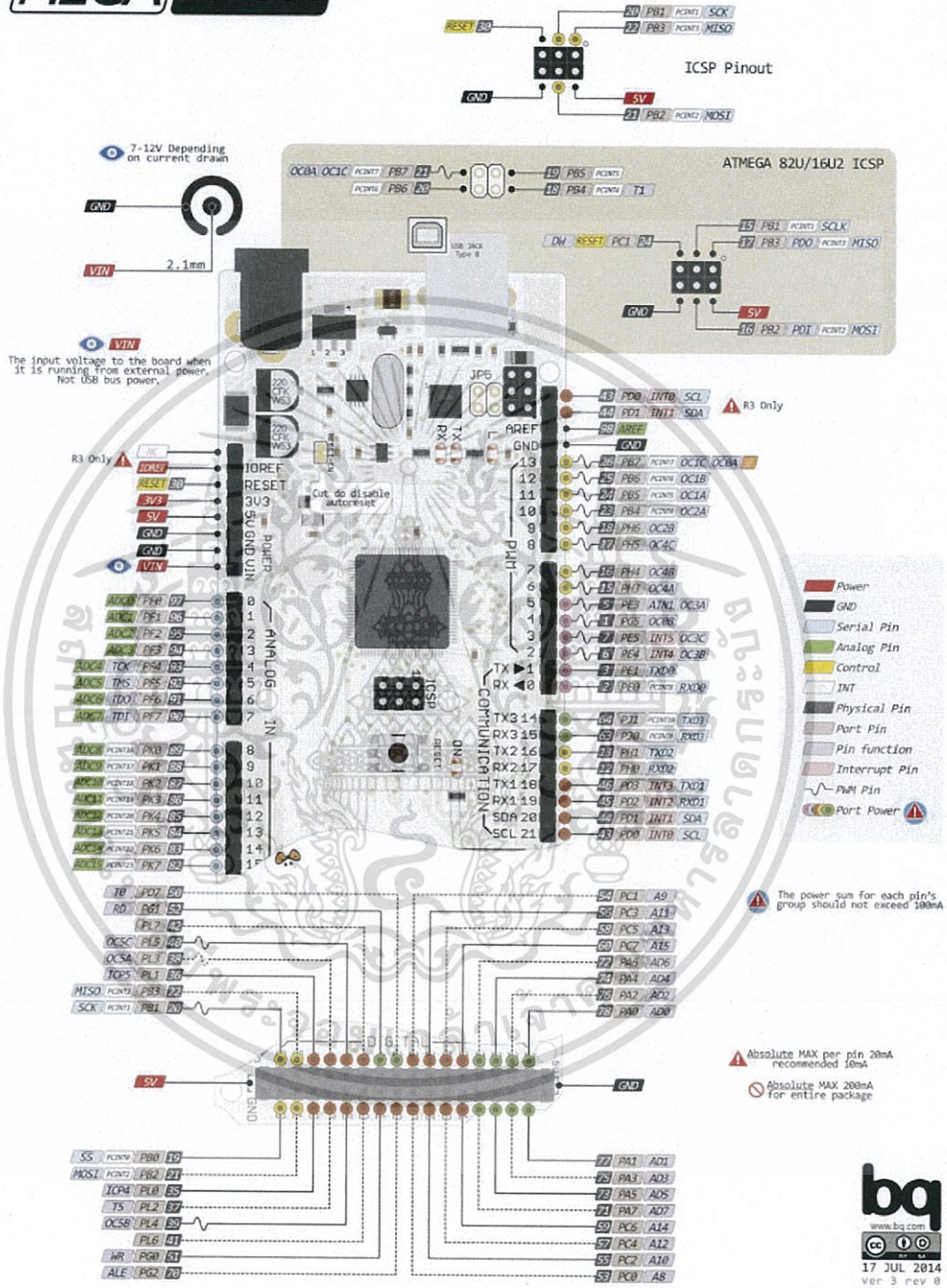
- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี COMMUNITY กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
- OPEN HARDWARE ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- CROSS PLATFORM สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

ARDUINO MEGA 2560 R3 เป็น MICROCONTROLLER BOARD ที่ใช้ ATMEGA2560 เป็น MCU หลัก ซึ่งตัวนี้จะมีขา DIGITAL 54 ขา อินพุต/เอาพุต (สามารถทำเป็น PWM ได้ถึง 14 ขา) และมีขา ANALOG อินพุตได้อีก 16 ขา, รั้นที่ความถี่ 16 MHZ มี USB CONNECTOR และ POWER JACK DC ซึ่ง CONCEPT ของ ARDUINO BOARD นี้ทำมาเพื่อความสะดวก ง่ายในการเชื่อมต่อเข้ากับคอมพิวเตอร์ สามารถต่อ USB เข้ากับช่องคอมพิวเตอร์ ก็สามารถ RUN โปรแกรมที่ BOARD ได้ เหมาะสำหรับผู้ที่กำลังเริ่มต้นเข้าสู่วงการอิเล็กทรอนิกส์อย่างแท้จริง



รูปที่ 2.1.1 บอร์ด ARDUINO MEGA 2560 R3 [1]

MEGA PINOUT



รูปที่ 2.1.2 โครงสร้างของ ARDUINO MEGA 2560 R3 [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 สเต็ปป์มอเตอร์ (STEPPING MOTOR)

สเต็ปป์มอเตอร์เป็นอุปกรณ์เอาต์พุตอย่างหนึ่ง ซึ่งสามารถนำไอซีไมโครคอนโทรลเลอร์ มาทำการควบคุมได้สะดวก และเป็นมอเตอร์ที่เหมาะสมสำหรับใช้ในงานควบคุมการหมุน ที่ต้องการตำแหน่ง และทิศทางที่แน่นอน การทำงานของ สเต็ปป์มอเตอร์จะขับเคลื่อนทีละขั้นๆ ละ (STEP) 0.9, 1.8, 5, 7.5, 15 หรือ 50 องศา ซึ่งขึ้นอยู่กับคุณสมบัติแต่ละชนิดของสเต็ปป์มอเตอร์ตัวนั้นๆ สเต็ปป์มอเตอร์จะแตกต่างจากมอเตอร์กระแสตรงทั่วไป (DC MOTOR) โดยการทำงานของมอเตอร์กระแสตรงจะหมุนไปแบบต่อเนื่อง ไม่สามารถหมุนเป็นแบบสเต็ปๆ ได้ดังนั้นในการนำไปกำหนดตำแหน่งจึงควบคุมได้ยากกว่า



รูปที่ 2.2 สเต็ปป์มอเตอร์ [3]

2.2.1 วิธีการขับสเต็ปป์มอเตอร์ให้หมุนโดยการกระตุ้นเฟส

ในการควบคุมสเต็ปป์มอเตอร์เพื่อที่จะให้ทำการหมุน มีวิธีการควบคุมกระแสไฟที่จ่ายให้กับขดลวดสเตเตอร์ (STATOR) ในแต่ละเฟสของสเต็ปป์มอเตอร์ อย่างเป็นลำดับที่แน่นอน โดยถ้าหากเราต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆ เป็นสถานะลอจิก "1" และในการกระตุ้นเฟสของสเต็ปป์มีอยู่ด้วยกัน 2 แบบคือ

2.2.1.1 การกระตุ้นเฟส แบบฟูลสเต็ปมอเตอร์ (FULL STEP MOTOR) ยังสามารถแบ่งการกระตุ้นเฟสออกได้เป็นอีก 2 วิธีด้วยกันคือ

1) การกระตุ้นเฟสแบบฟูลสเต็ป 1 เฟส (SINGLE-PHASE DRIVER) หรือแบบเวฟ แสดงดังตารางที่ 2.2.1 จะเป็นการป้อนกระแสไฟให้กับขดลวด ของสเต็ปป์มอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไป ดังนั้นกระแส ที่ไหลในขดลวด จะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเต็ปป์มอเตอร์มีน้อย

2) การกระตุ้นเฟสแบบฟูลสเต็ป 2 เฟส (TWO-PHASE DRIVER) แสดงดังตารางที่ 2.2.2 เป็นการป้อนกระแสให้กับขดลวด 2 ขด ของสเต็ปป์มอเตอร์พร้อมๆกันไป และจะกระตุ้นเรียงถัดกันไปเช่นเดียวกับแบบหนึ่งเฟส ดังนั้นการกระตุ้นแบบนี้จึงต้องใช้กำลังไฟมากขึ้น และจะทำให้มีแรงบิดของมอเตอร์มากกว่าการกระตุ้นแบบ 1 เฟส

2.2.1.2 การกระตุ้นเฟส แบบฮาล์ฟสเต็ป (HALF STEP MOTOR) หรือ ONE-TWO PHASE DRIVER คือการกระตุ้นเฟสแบบ ฟูลสเต็ป 1 เฟส และ 2 เฟส เรียงลำดับกันไป แสดงดังตารางที่ 2.2.3 แรงบิดที่ได้จากการกระตุ้นเฟสแบบนี้จะมีเพิ่มมากขึ้น เพราะช่วงของสเต็ปป์มีระยะสั้นลง ในการกระตุ้นแบบนี้ เราจะต้องมีการกระตุ้นที่เฟสถึง 2 ครั้ง จึงจะได้ระยะของ สเต็ปป์เท่ากับการกระตุ้นเพียงครั้งเดียว ของแบบฟูลสเต็ป 2 แบบแรก ความละเอียดของการหมุนตำแหน่งองศาต่อสเต็ป ก็เป็นสองเท่าของแบบแรก ความถูกต้องของตำแหน่งที่กำหนดจึงมีมากขึ้น

ตารางที่ 2.2.1 การกระตุ้นเฟสแบบฟูลสเต็ป 1 เฟส (SINGLE-PHASE DRIVER)

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

ตารางที่ 2.2.2 การกระตุ้นเฟสแบบฟูลสเต็ป 2 เฟส (TWO-PHASE DRIVER)

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

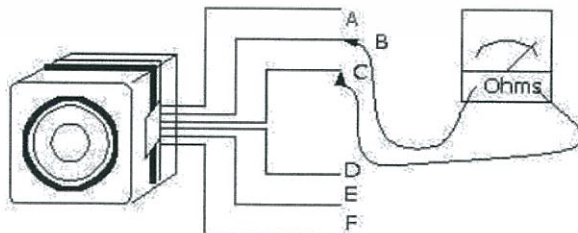
ตารางที่ 2.2.3 การกระตุ้นเฟส แบบฮาล์ฟสเต็ป 2 เฟส (HALF STEP MOTOR)

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

2.2.2 วิธีการตรวจสอบหาเฟสของขดลวดสเต็ปปั๊มมอเตอร์

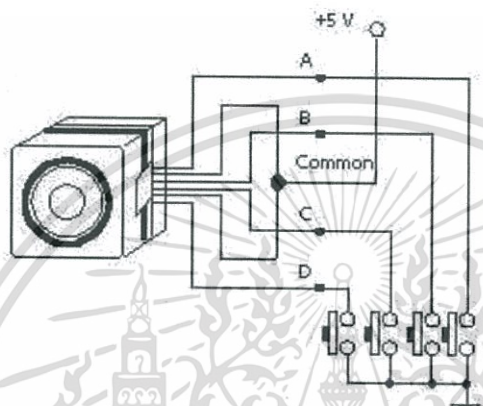
ในขั้นตอนที่ 1 ให้สังเกตว่าสเต็ปปั๊มมอเตอร์ที่นำมาทดลองที่เป็นแบบยูนิโพลาร์ (Uni-polar stepper motor) จะมีจำนวนสาย 5 เส้นหรือ 6 เส้น (ถอดจากดิสก์ไดร์ฟเก่าขนาด 5 นิ้วนำมาใช้งานได้)

ในขั้นตอนที่ 2 ใช้มิเตอร์วัดค่าความต้านทานของเส้นลวดในแต่ละขดดังรูป 6.17 ขั้นตอนการวัด ให้หาสายที่ต่อเป็นจุดร่วมเสียก่อน (common) โดยให้ใช้ มัลติมิเตอร์ตั้งค่าไว้สำหรับการวัดค่าความต้านทาน แต่ละเส้น สังเกตที่ค่าความต้านทาน ถ้าหากเราไม่ได้วัดระหว่าง จุดต่อร่วม (common) กับสายแต่ละเส้น ค่าความต้านทานจะมีค่าเป็น 2 เท่าของการวัดระหว่างจุดต่อร่วมกับสายที่ใช้งาน ตัวอย่างเช่น ถ้าให้จุด B เป็นจุดร่วม หากวัดระหว่างที่จุด A กับจุด B จะมีค่าเท่ากับ 60 Ohm แต่ถ้าวัดระหว่างที่จุด A และจุด C ซึ่งไม่ใช่จุดร่วมก็จะได้ค่าเท่ากับ 120 Ohm หากเป็นแบบที่มีสาย 6 เส้นก็จะมีจุดร่วมสองจุด เพราะมีขดลวดคนละชุดกัน และสายที่เป็นจุดร่วมส่วนใหญ่จะมีสีเหมือนกัน ทำนองเดียวกันหากเป็นแบบที่มีสาย 5 เส้นก็จะมีจุดร่วมเพียงจุดเดียวเท่านั้น



รูปที่ 2.2.2.1 การใช้มิเตอร์วัดค่าความต้านทาน [4]

ในขั้นตอนที่ 3 หากเป็นแบบที่มีสาย 6 เส้นก็ให้ทำการต่อจุดร่วมเข้าด้วยกันจะได้เป็น 5 เส้น แล้วต่อวงจรตาม รูปหลังจากนั้นให้ทดลองกดสวิตช์ ที่ต่อเข้ากับแต่ละจุดโดยเริ่มที่ จุด A จุด B จุด C และจุด D แล้วให้สังเกตการหมุนของสเต็ปปีงมอเตอร์ว่าหมุนได้ต่อเนื่องหรือไม่ หากมีการกระโดดข้ามสเต็ปก็ให้ทดลองโดยเรียงลำดับการกดสวิตช์ใหม่ จนหาลำดับของสายได้ถูกต้องคือ มอเตอร์เดินตามที่สเต็ป อย่างเป็นลำดับ



รูปที่ 2.2.2.2 การต่อวงจรเพื่อทดสอบโดยการสวิตช์เพื่อหาลำดับ [4]

2.3 SD CARD MODULE

โมดูลอ่าน / เขียน SD CARD ใช้สำหรับการเก็บข้อมูลปริมาณมาก ที่ EEPROM ไม่สามารถเก็บไว้ได้หมด การนำไปใช้งานเช่น การเก็บค่าอุณหภูมิแบบต่อเนื่อง เป็นต้น โมดูล SD CARD เป็นโมดูลที่ช่วยในการนำ ARDUINO ไปเชื่อมต่อกับ SD CARD ได้ง่ายมากยิ่งขึ้น ซึ่ง SD CARD ใช้การเชื่อมต่อ รับ-ส่งข้อมูลโดยใช้โปรโตคอล SPI ทำงานที่แรงดัน 3.3V โมดูลนี้ได้รวมวงจรเรกกูเลเตอร์ แปลงแรงดันไฟตรงจาก 5V เป็น 3.3V และมีวงจร LOGIC LEVEL CONVERTER บนบอร์ด เพื่อให้สามารถนำไปใช้งานกับบอร์ด ARDUINO ที่ทำงานที่แรงดัน 5V ได้ทันที นอกจากนี้ โมดูลยังสามารถใช้งานกับบอร์ดไมโครคอนโทรลเลอร์อื่นๆได้ อย่างเช่น นำไปใช้งานกับโมดูล ESP8266 เพื่อเก็บค่าอุณหภูมิย้อนหลังได้เป็นเดือนๆได้อีกด้วย

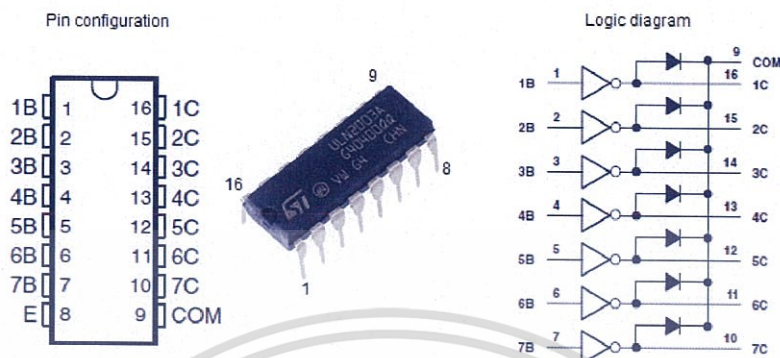
ตารางที่ 2.3 การต่อขา SD CARD MODULE กับ ARDUINO

บอร์ด ARDUINO	โมดูลอ่าน/เขียน SD CARD
D13	CLK
D12	MISO
D11	MOSI
D4	CS
5V	5V
GND	GND

โดยที่ CLK	:	ขาสัญญาณอนุกรม SERIAL CLOCK (CLK)
MOSI	:	ขาสัญญาณ MOSI (DIN)
MISO	:	ขาสัญญาณ MISO (DOUT)
CS	:	ขาสัญญาณ CHIP SELECT สำหรับเปิดปิดการรับส่งข้อมูลของ SD CARD
VCC	:	ขาไฟเลี้ยง LCD ช่วงแรงดันที่ใช้ 3.3V-5.5V
GND	:	ขากราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ULN2003



รูปที่ 2.4 ULN2003 [5]

IC เบอร์ ULN2003 เป็นไอซี Driver สามารถขับอุปกรณ์เอาต์พุตขนาดไม่เกิน 500 mA เช่นมอเตอร์ รีเลย์ หรือ stepmother โดยเราสามารถจะนำมาให้ทำหน้าที่เป็นบัฟเฟอร์ (Buffer) ข้อมูลขนาด 8 บิต ในกรณีที่ต้องการเชื่อมต่อพอร์ตกับอุปกรณ์ภายนอก ซึ่งสามารถขับได้ 7ช่อง โดยที่

Output pins : 1B, 2B, 3B, 4B, 5B, 6B, 7B

Input pins : 1C, 2C, 3C, 4C, 5C, 6C, 7C

Vcc (5V) : COM (ขา 9)

GND : E (ขา 8)

การทำงานของ ULN2003 คือเมื่อ input high จะทำให้ output ออกมาเป็นค่า GND และเมื่อ input low จะทำให้ output ออกมาเป็นค่า high impedance

2.5 COMPASS MODULE (HMC5883L)

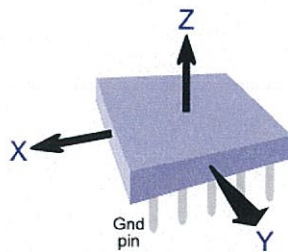
DIGITAL COMPASS เป็นเครื่องมือช่วยบอกทิศทางแบบอิเล็กทรอนิกส์ แสดงค่ามุมเมื่อเทียบกับทิศเหนือเป็นตัวเลข ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ ซึ่ง DIGITAL COMPASS จะให้ค่าทิศทางเมื่อเทียบกับทิศเหนือของโลก หรือสามารถที่จะตั้งค่าของมุมหรือทิศทางอ้างอิงได้ จึงสามารถนำไปใช้ในการพัฒนาเครื่องมือสำหรับบอกหรือกำหนดทิศทางได้

ตารางที่ 2.5 การต่อขา COMPASS MODULE กับ ARDUINO

บอร์ด ARDUINO	COMPASS MODULE
ANALOG PIN 4	SDA (DATA)
ANALOG PIN 5	SCL (CLOCK)
3.3V	VCC
GND	GND

รูปที่ 2.5.1 COMPASS MODULE [6]

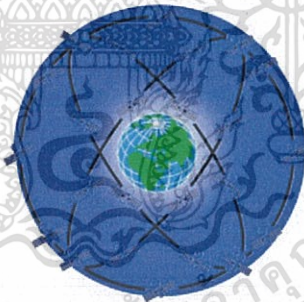
การทำงานของ HMC5883L ไมโครชิพ 3 แกน วัดสนามแม่เหล็กในสามทิศทางหรือแกนเรียกว่า X,Y และ Z หรือเข้าใจอย่างง่ายว่าไมโครชิพสามารถใช้งานเป็นเข็มทิศธรรมดาได้ เพื่อหาทิศแม่เหล็กทิศเหนือของโลก สามารถรับรู้ถึงความแรงของสนามแม่เหล็กที่มาจากแหล่งกำเนิดข้างเคียง และในหน้าที่ของการเป็นเซนเซอร์สามารถจับสนามแม่เหล็กในสามแกน และยังสามารถหาระยะสัมพัทธ์และทิศทางเมื่อเทียบกับแหล่งกำเนิดด้วย



รูปที่ 2.5.2 แกนแบบ 3-AXIS ของ COMPASS MODULE [7]

2.6 GLOBAL POSITIONING SYSTEM (GPS)

Global Positioning System: GPS เรียกกันย่อ ๆ ว่า ระบบจีพีเอส คือระบบบอกตำแหน่งบนพื้นผิวโลกด้วยดาวเทียม โดยอาศัยการคำนวณจากข้อมูลที่ส่งมาจากดาวเทียมที่โคจรอยู่รอบโลกจำนวน 24 ดวง ซึ่งทราบตำแหน่งที่แน่นอนของตัวเองตามรูปที่ 2.6.1 ระบบนี้ในปัจจุบันไม่ได้มีแต่เฉพาะสหรัฐอเมริกาเท่านั้นที่เป็นผู้พัฒนา แต่ยังมีเครือข่ายดาวเทียมอีกหลายค่าย คือ ระบบ GLONASS ของรัสเซีย ระบบ Galileo โดยสหภาพยุโรป ร่วมกับจีน อิสราเอล อินเดีย โมร็อกโก ซาอุดีอาระเบีย เกาหลีใต้ และยูเครน ระบบ Beidou โดยสาธารณรัฐประชาชนจีน และระบบ QZSS โดยญี่ปุ่น



รูปที่ 2.6 วงโคจรของดาวเทียมจีพีเอส [8]

ระบบจีพีเอสของค่ายอเมริกา มีใช้กันมาตั้งแต่แรกเริ่มและนิยมใช้กันมากที่สุด ชื่อ NAVSTAR (Navigation Satellite Timing and Ranging GPS) มีดาวเทียม 28 ดวง ใช้งานจริง 24 ดวง อีก 4 ดวงเป็นดาวเทียมสำรอง มีรัศมีวงโคจรจากพื้นโลก 20,162.81 กม.หรือ 12,600 ไมล์ ดาวเทียมแต่ละดวงใช้ เวลาในการโคจรรอบโลก 12 ชั่วโมง จีพีเอสทำงานโดยการรับสัญญาณจากดาวเทียมแต่ละดวง ซึ่งสัญญาณดาวเทียมนี้อาจประกอบด้วย ข้อมูลที่ระบุตำแหน่ง และเวลาขณะส่งสัญญาณ และที่ตัวเครื่องรับสัญญาณจีพีเอส จะต้องประมวลผลความแตกต่างของเวลาในการรับ

สัญญาณเทียบกับเวลาจริง ณ ปัจจุบันเพื่อแปรเป็นระยะทางระหว่างเครื่องรับสัญญาณกับดาวเทียมแต่ละดวง และประมวลผลหาตำแหน่งของดาวเทียมที่ส่งมากับสัญญาณดังกล่าวข้างต้น

ดังนั้นหากเครื่องรับสัญญาณ สามารถรับสัญญาณจากดาวเทียมจีพีเอส มากดวงเท่าใด ก็จะสามารถระบุตำแหน่งได้แม่นยำยิ่งขึ้นเท่านั้น ในทางปฏิบัติจะต้องมีดาวเทียมอย่างน้อย 4 ดวง เพื่อหาตัวแปร 4 ตัวคือ ตำแหน่งบนผิวโลก X, Y,Z และความคลาดเคลื่อนของเวลา dt

ตารางที่ 2.6.1 คุณสมบัติของเครื่องรับสัญญาณจีพีเอส (Ublox NEO-6M)

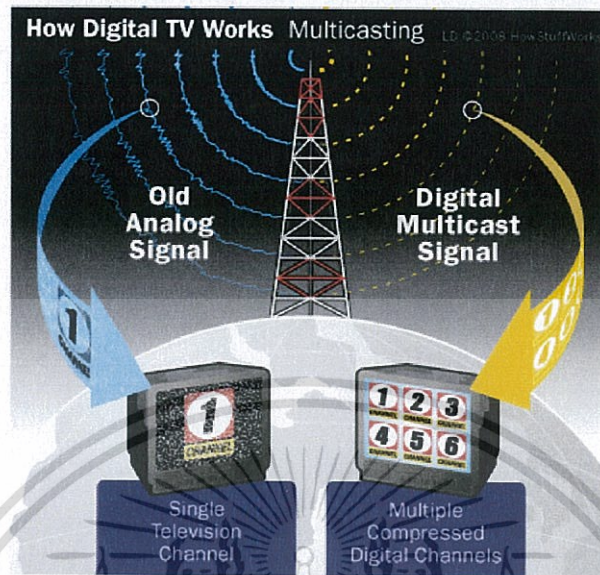
RECEIVER TYPE	50 CHANNELS GPS L1 FREQUENCY, C/A CODESBAS: WAAS, EGNOS, MSAS
TIME-TO-FIRST-FIX	COLD START : 27 S
	WARM START : 27 S
	HOT START : 1 S
	AIDED STARTS < 3 S
SENSITIVITY	TRACKING & NAVIGATION : 161 DBM
	REACQUISITION : 160 DBM
	COLD START (WITHOUT AIDING) : 147 DBM
	HOT START : 156 DBM
MAXIMUM NAVIGATION UPDATE RATE	5HZ
HORIZONTAL POSITION ACCURACY	GPS : 2.5 M
	SBAS : 2.0 M
	SBAS + PPP : < 1 M (2D, R50)
	SBAS + PPP : < 2 M (3D, R50)
TIMEPULSE FREQUENCY RANGE	16
ACCURACY FOR TIMEPULSE SIGNAL	RMS : 30 NS
	99% : <60 NS
	GRANULARITY : 21 NS

	COMPENSATED : 15 NS
VELOCITY ACCURACY	0.1M/S
HEADING ACCURACY	0.5 DEGREES
OPERATIONAL LIMITS	DYNAMICS 4 G
	ALTITUDE : 50,000 M
	VELOCITY : 500 M/S

รูปที่ 2.6.1 เครื่องรับสัญญาณจีพีเอส (UBLOX NEO-6M) [9]

2.7 ความรู้เบื้องต้นเกี่ยวกับทีวีดิจิตอล

ทีวีดิจิตอลหรือโทรทัศน์ระบบดิจิตอล (DIGITAL TELEVISION) เป็นระบบการส่งผ่านสัญญาณภาพวิดีโอและเสียงโดยระบบดิจิตอล ซึ่งแต่เดิมระบบการรับสัญญาณโทรทัศน์ในบ้านเราใช้เป็นระบบอนาล็อก โดยระบบดิจิตอลมีจุดเด่นกว่าระบบอนาล็อกทั้งในด้านความคมชัดของภาพและเสียง และการส่งข้อมูลแบบดิจิตอลสามารถส่งข้อมูลได้มากกว่าแบบเดิม (MULTICASTING) ซึ่งในปัจจุบันได้มีการพัฒนาระบบการรับส่งสัญญาณดิจิตอลไปอีกระดับ เช่น โทรทัศน์จอกว้าง (WIDE SCREEN) โทรทัศน์ความคมชัดสูง (HDTV) ในขณะที่ระบบอนาล็อกไม่สามารถพัฒนาประสิทธิภาพได้ ซึ่งระบบสัญญาณดิจิตอลเกิดขึ้นมาจากการนำระบบคอมพิวเตอร์นำมาพัฒนาใช้ในการช่วยโทรทัศน์ แล้วจึงได้มีการปรับปรุงโทรทัศน์ให้ใช้ระบบดิจิตอลด้วย



รูปที่ 2.7 การส่งสัญญาณแบบอะนาล็อกและดิจิทัล [10]

2.7.1 จุดเด่นของระบบการส่งสัญญาณแบบดิจิทัล

- 1) ระบบ ดิจิตอลมีระบบการบีบอัดสัญญาณ (DIGITAL COMPRESSION) ทำให้สามารถส่งรายการต่อช่องได้มากขึ้น จากเดิม 1 ช่องส่งได้ 1 รายการ แต่ระบบดิจิทัล 1 ช่องจะสามารถส่งได้ถึง 4-6 รายการทางภาคพื้นดิน และ 8-10 รายการทางดาวเทียม
- 2) สามารถให้บริการเสริมอื่นๆ ได้ (ในกรณีที่ถูกกฎหมาย)
- 3) สามารถรับชมขณะอยู่ในพาหนะเคลื่อนที่ได้ เช่น รับโทรทัศน์บนรถยนต์
- 4) สามารถให้บริการฟรี (FREE TO AIR) หรือบริการเก็บค่าสมาชิกได้
- 5) เนื่องจากเครื่องส่ง 1 เครื่อง สามารถส่งได้หลายรายการ ทำให้ค่าใช้จ่ายต่อรายการลดลง(จากเดิม 1 ช่องส่งได้ 1 รายการ)
- 6) ระบบ ดิจิตอล สามารถพัฒนาให้มีประสิทธิภาพเพิ่มขึ้นได้ เพื่อรองรับเทคโนโลยีในการส่งและรับโทรทัศน์ในอนาคตได้ เช่น โทรทัศน์จอกว้าง (WIDE SCREEN) โทรทัศน์ความคมชัดสูง (HDTV) ในขณะที่ระบบอะนาล็อกไม่สามารถพัฒนาประสิทธิภาพได้
- 7) เนื่องจากระบบดิจิทัลนั้น เครื่องส่งใช้กำลังออกอากาศลดลง ทำให้ประหยัดพลังงาน
- 8) ข้อที่สำคัญที่สุดก็คือ คุณภาพในการรับชมดีขึ้น ไม่มีเงา การรบกวนน้อย เพราะถ้าจะรับได้ชัดก็ชัดเลยแต่ถ้าอยู่ในที่รับไม่ชัดก็จะรับไม่ได้

2.7.2 ความแตกต่างของ DIGITAL TV กับ ANALOG TV

1) คุณภาพของสัญญาณภาพ สัญญาณภาพของระบบ DIGITAL TV จะมีคุณภาพดี คมชัด อยู่เสมอ ต่างกับระบบ ANALOG TV ที่สัญญาณจะเปลี่ยนไปตามระดับความแรงของสัญญาณที่ได้รับ



รูปที่ 2.7.2.1 ความแตกต่างคุณภาพสัญญาณภาพตามระดับความแรงของสัญญาณที่ได้รับ [11]

2) คุณภาพของภาพ คุณภาพภาพในระบบ Digital TV จะสูงมากกว่าระบบ Analog TV โดยที่

- ระบบ Digital TV ความละเอียดของภาพสูงสุดจะอยู่ในระดับ Full High Definition (Full HD) 1080p

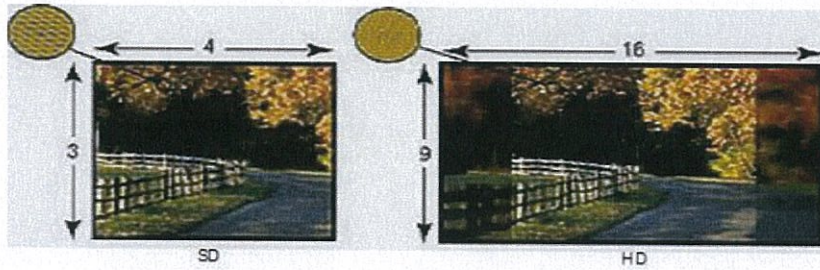
- Picture is 1920 x 1080 pixels, sent at 60 frame/sec

- ส่วนในระบบ Analog TV ความละเอียดของภาพสูงสุดจะอยู่ในระดับเพียง Standard definition (SD) 480p

- Picture is 704 x 480 pixels, sent at 60 complete

frame/sec เท่านั้น

- ในส่วนของอัตราส่วนของภาพ ระบบ Digital TV จะเป็น จอ กว้าง 16:9 แต่ระบบ Analog TV จะเป็นเพียง 4:3 เท่านั้น



รูปที่ 2.7.2.2 ความแตกต่างของคุณภาพสัญญาณภาพ [11]

3) คุณภาพเสียง Digital TV จะส่งสัญญาณเสียงมาในระบบ Surround 5.1 channels ขณะที่ Analog TV จะส่งสัญญาณเสียงมาในระบบ Stereo 2 channels

4) ประสิทธิภาพการใช้คลื่นความถี่ ใน 1 ช่องความถี่ ขนาด 8 MHz ถ้าถ่ายทอดสัญญาณด้วยระบบ Analog TV จะส่งช่องรายการได้เพียงช่องรายการเดียว แต่ในขณะที่ระบบ Digital TV (DVB-T) ส่งสัญญาณในแบบ SDTV ได้ 8-10 รายการ หรือ HDTV 2 รายการ + SDTV 2 รายการ จึงเป็นการใช้งานคลื่นความถี่อย่างมีประสิทธิภาพมากกว่า

5) รองรับรูปแบบบริการใหม่ๆ การให้บริการ Features ใหม่ๆ ที่จะเกิดขึ้น หลังจากที่มีการปรับเปลี่ยนการแพร่สัญญาณโทรทัศน์ภาคพื้นดินไป สู่ระบบ Digital ซึ่งล้วนแล้วแต่เป็นบริการที่ยังไม่เคยมีให้บริการมาก่อนในประเทศไทย เช่น Data broadcasting, Interactivity, Multimedia, Emergency warnings ฯลฯ ยกตัวอย่างให้เห็นชัดเจน เช่น การแทรกข้อมูลพยากรณ์อากาศ การเตือนภัยพิบัติฉุกเฉินแบบตามเวลาจริงให้ปรากฏบนจอ การสื่อสารระหว่างผู้ชมกับผู้จัดรายการในทันที การแสดงความคิดเห็น การโหวตรายการ เป็นต้น

2.7.3 จำนวนเขตบริการและจำนวนสถานี

กำหนดให้มีเขตบริการจำนวน 39 เขตบริการ ซึ่งการแบ่งเขตบริการ คำนึงถึงข้อจำกัดทางเทคนิคเป็นหลักรวมทั้งคำนึงถึงระยะห่างจากสถานีหลักตลอดจนลักษณะทางสังคมและวัฒนธรรมด้วย โดยแต่ละเขตบริการประกอบไปด้วย 1 สถานีหลัก และสถานีเสริมอีกจำนวนหนึ่ง รวมทั้งสิ้นจาก 39 เขตบริการ ซึ่งจะมี 39 สถานีหลัก และ 132 สถานีเสริม ทั้งนี้ อาจกำหนดสถานีเสริมเป็นการเพิ่มเติมในภายหลัง เพื่อแก้ไขปัญหาการรับสัญญาณ หรือเพื่อปรับปรุงการรับสัญญาณแบบพกพาภายในอาคาร (PORTABLE INDOOR RECEPTION)

ในการจัดทำแผนความถี่วิทยุสำหรับกิจการโทรทัศน์ แบ่งประเภทของสถานีเป็นดังตารางที่ 2.7.3.1

ตารางที่ 2.7.3.1 แสดงประเภทและรายละเอียดของสถานีส่ง [12]

อักษรย่อ	ประเภทสถานี	จำนวนสถานี	หมายเหตุ
M	สถานีหลัก	39	สถานีหลักตามแผนการขยายโครงข่ายภาคพื้นดินในระบบดิจิทัลในปีที่ 1 และ 2 โดยเป็นสถานีซึ่งมีสถานีที่ตั้งอยู่เดิม และสถานีที่ตั้งนี้ได้รับความเห็นชอบร่วมกันระหว่างผู้ได้รับอนุญาตประกอบกิจการโทรทัศน์สำหรับการให้บริการโครงข่ายโทรทัศน์ในระบบดิจิทัลทุกราย
A1	สถานีเสริมกลุ่ม A1	45	เป็นสถานีซึ่งมีสถานีที่ตั้งอยู่เดิม และสถานีที่ตั้งนี้ได้รับความเห็นชอบร่วมกันระหว่างผู้ได้รับอนุญาตประกอบกิจการโทรทัศน์สำหรับการให้บริการโครงข่ายโทรทัศน์ในระบบดิจิทัลทุกราย โดยสถานีเสริมกลุ่ม A1 เป็นการติดตั้งสถานีเสริมเพื่อให้ได้พื้นที่ครอบคลุมร้อยละ 90 ของจำนวนครัวเรือนทั่วประเทศ
A2	สถานีเสริมกลุ่ม A2	38	เป็นสถานีซึ่งมีสถานีที่ตั้งอยู่เดิม และสถานีที่ตั้งนี้ได้รับความเห็นชอบร่วมกันระหว่างผู้ได้รับอนุญาตประกอบกิจการโทรทัศน์สำหรับการให้บริการโครงข่ายโทรทัศน์ในระบบดิจิทัลทุกราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.4 พื้นที่ครอบคลุมของเขตบริการ

พื้นที่ครอบคลุมของเขตบริการภายหลังการติดตั้ง 39 สถานีหลัก และ 83 สถานีเสริม จากผลการวิเคราะห์คำนวณโดยอาศัยซอฟต์แวร์สร้างแบบจำลองการแพร่กระจายคลื่น (WAVE PROPAGATION MODEL) โดยอาศัยคุณลักษณะทางเทคนิคของสถานีวิทยุคมนาคมสำหรับกิจการโทรทัศน์ภาคพื้นดิน

ตารางที่ 2.7.4.1 แสดงค่าพิกัดตำแหน่งของเสาส่งสัญญาณทีวีดิจิตอล และพื้นที่การกระจายสัญญาณ [12]

ลำดับ	ชื่อสถานี	ประเภท	ลองจิจูด	ละติจูด	ความสูงสายอากาศ (เมตร)	รัศมีการกระจาย สัญญาณ (กิโลเมตร)
1.00	กรุงเทพ	M	100.540270	13.754300	328	74.658
1.01	เขาดลาง	A1	100.949558	13.190653	40	26.071
1.02	พืทยา	A1	100.866450	12.921333	60	31.931
1.03	จอมบึง	A2	99.613515	13.627185	112	43.626
1.04	บ่อทอง	A2	101.439334	13.278336	72	34.978
1.05	สมุทรสงคราม	A2	100.003068	13.409810	73	35.220
2.00	กาญจนบุรี	M	99.420556	14.074444	124	45.904
2.01	ไทรโยค	A2	99.144000	14.118230	70	34.489
2.02	ทองผาภูมิ	A2	98.623956	14.752401	70	34.489
3.00	สิงห์บุรี	M	100.377004	14.836115	126	46.272
3.01	สุพรรณบุรี	A1	100.180238	14.463391	100	41.222
3.02	วังคัน (ด่านช้าง 2)	A1	99.667900	14.930610	60	31.931
3.03	มวกเหล็ก	A1	101.153715	14.652015	20	18.435
3.04	สระบุรี(ศูนย์การทหารม้า)	A1	100.926111	14.508056	80	36.871
3.05	ด่านช้าง	A2	99.695923	14.842848	107	42.641
4.00	ระยอง	M	101.412925	12.675987	52	29.726
4.01	โป่งน้ำร้อน	A2	102.274523	12.889700	92	39.539
5.00	สระแก้ว	M	102.104231	13.804003	156	51.487
5.01	ปราจีนบุรี (ค่ายพรหมโยธี)	A1	101.373976	14.134715	80	36.871
5.02	ตาพระยา	A2	102.804796	14.002854	97	40.599
6.00	ตราด	M	102.298440	12.195510	64	32.978
7.00	ประจวบคีรีขันธ์	M	99.801300	11.905940	60	31.931
7.01	หัวหิน	A1	99.935176	12.565142	75	35.700
7.02	เพชรบุรี	A1	99.929012	13.104522	80	36.871
7.03	ทับสะแก	A1	99.588250	99.588250	84	37.781
8.00	นครราชสีมา	M	101.995052	14.947688	156	51.487

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.01	ชุมพวง	A1	102.794060	15.273160	120	45.157
8.02	เขายายเที่ยง	A1	101.546679	14.786090	80	36.871
8.03	เขาสลัดได รังน้ำเขียว	A2	101.900449	14.415619	40	26.071
9.00	ชัยภูมิ	M	102.026740	15.729570	120	45.157
9.01	หนองบัวแดง	A1	101.895300	15.999858	120	45.157
9.02	หนองบัวแดง	A2	101.804348	16.082553	70	34.489
10.00	สุรินทร์	M	103.507680	14.919520	126	46.272
10.01	บุรีรัมย์	A1	103.098123	14.934247	80	36.871
10.02	นางรอง	A2	102.794802	14.630352	72	34.978
11.00	ศรีสะเกษ	M	104.345228	15.042331	120	45.157
11.01	กันทรลักษ์	A2	104.651717	14.641479	72	34.978
12.00	อุบลราชธานี	M	104.923611	15.381667	156	51.487
12.01	อำนาจเจริญ	A1	104.618922	15.898967	100	41.222
12.02	บุณฑริก	A2	105.413027	14.756517	102	41.632
12.03	น้ำยืน	A2	105.002684	14.490170	107	42.641
12.04	ห้วยหินครก	A2	104.985411	15.942518	100	41.222
13.00	มุกดาหาร	M	104.718181	16.613679	124	45.904
13.01	ภูหินขัน (มุกดาหาร)	A1	104.629602	16.468387	80	36.871
14.00	ร้อยเอ็ด	M	103.624210	15.978350	126	46.272
14.01	ยโสธร	A1	104.142504	15.803563	100	41.222
15.00	ขอนแก่น	M	102.946222	16.463686	156	51.487
15.01	ชุมแพ	A1	102.061324	16.562602	124	45.904
15.02	ภูสิงห์ ภูพานินทร์	A1	103.518611	16.722778	112	43.626
15.03	พล	A1	102.600232	15.809009	60	31.931
15.04	แก้งคร้อ (ภูผาแดง)	A2	102.295658	16.098189	25	20.611
16.00	เลย	M	101.444519	17.470655	35	24.388
16.01	หนองบัวลำภู	A1	102.429346	17.228005	100	41.222
17.00	อุดรธานี	M	102.794088	17.664041	156	51.487
18.00	บึงกาฬ	M	103.553190	18.354690	100	41.223
19.00	สกลนคร	M	103.987520	17.138050	109	43.038
19.01	นครพนม	A1	104.770131	17.374367	120	45.157
20.00	เชียงใหม่	M	98.915020	18.808140	78	36.407
20.01	ไชยปราการ	A1	99.144830	19.628480	19	17.968
20.02	เชียงดาว(เขาห้วยหินฝน)	A1	98.951143	19.250543	32	23.319
20.03	พร้าว	A2	99.197111	19.369972	30	22.578
20.04	ลี้	A2	98.947841	17.801315	85	38.005
20.05	ดอยเต่า	A2	98.681750	17.953028	58	31.394

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20.06	เวียงแหง	A2	98.725861	19.521639	30	22.578
21.00	แม่ฮ่องสอน(ตอยกองมู)	M	97.957950	19.297550	64	32.978
21.01	แม่ฮ่องสอน (หัวนางปู้)	A1	98.034777	19.106500	50	29.148
21.02	แม่สะเรียง	A1	97.944614	18.169806	82	37.328
21.03	ปาย	A1	98.421248	19.385555	100	41.222
22.00	ลำปาง	M	99.560597	18.234928	80	36.871
22.01	เถิน	A1	99.104840	17.664020	80	36.871
22.02	วังเหนือ	A2	99.619450	19.144704	75	35.700
23.00	เชียงใหม่	M	99.867041	19.813755	70	34.489
23.01	พะเยา(เมือง)	A1	99.880542	19.194063	100	41.222
23.02	พะเยา(บ้านไร่คนเดียว)	A1	99.911417	18.988389	120	45.157
23.03	เชียงใหม่	A2	100.411051	20.246216	80	36.871
23.04	เวียงป่าเป้า	A2	99.508308	19.344348	95	40.179
23.05	แม่สรวย (ตอยมอนป่าก่อ)	A2	99.466883	19.663794	97	40.599
24.00	น่าน	M	100.741150	18.742640	70	34.489
24.01	เชียงใหม่(ตอยหินแก้ว)/ปัว	A1	100.822040	19.219755	22	19.335
25.00	แพร่	M	100.001030	17.932690	64	32.978
25.01	วังชิ้น	A2	99.601694	17.909139	58	31.394
26.00	อุตรดิตถ์	M	100.125433	17.619686	120	45.157
27.00	สุโขทัย	M	100.010563	16.984763	156	51.487
27.01	กำแพงเพชร	A1	99.524040	16.517987	120	45.157
27.02	คลองลาน	A2	99.319228	16.204277	90	39.107
27.03	นครไทย (เขาหัวหินลาด)	A2	100.856456	16.873538	72	34.978
28.00	ตาก	M	98.927687	16.778691	80	36.871
28.01	แม่สอด	A1	98.566522	16.732602	100	41.222
28.02	พบพระ	A2	98.691930	16.392920	70	34.489
29.00	นครสวรรค์	M	100.133580	15.716170	86	38.228
29.01	พิจิตร	A1	100.320572	16.448243	33	23.680
29.02	ลานสัก	A2	99.549740	15.460930	70	34.489
30.00	เพชรบูรณ์	M	101.071373	16.251014	154	51.156
30.01	วิเชียรบุรี	A1	101.103522	15.656473	80	36.871
30.02	หล่มเก่า	A2	101.232273	16.884594	70	34.489
31.00	ชุมพร	M	99.191480	10.530810	154	51.156
31.01	หลังสวน	A1	99.064853	9.959558	120	45.157
31.02	ท่าแซะ	A1	99.215555	10.835377	32	23.319
32.00	ระนอง	M	98.669486	10.028664	123	45.718
33.00	สุราษฎร์ธานี	M	99.348530	9.092330	84	37.781

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

33.01	เกาะสมุย	A1	99.988110	9.490660	60	31.931
33.02	พนม (ทับคริสต์)	A2	98.800205	8.835879	45	27.653
34.00	ภูเก็ต	M	98.395300	7.898710	64	32.978
34.01	ตะกั่วป่า	A1	98.350806	8.839694	60	31.931
34.02	กระบี่	A1	98.906656	8.063000	100	41.222
34.03	พังงา	A1	98.506540	8.434645	100	41.222
34.04	ป่าตอง	A2	98.316287	7.880979	35	24.388
34.05	ถลาง	A2	98.329306	8.033861	35	24.388
34.06	ปลายพระยา	A2	98.863561	8.529853	87	38.450
35.00	นครศรีธรรมราช	M	99.977210	8.366790	124	45.904
35.01	ทุ่งสง (เขาดาว)	A1	99.489998	8.210180	63	32.719
35.02	เขาคาดฟ้า	A2	99.825820	9.118560	70	34.489
36.00	ตรัง	M	99.486511	7.656956	126	46.272
37.00	สงขลา	M	100.519690	7.015170	66	33.490
37.01	เทพา	A2	100.938000	6.739820	50	29.148
37.02	นาทวี	A2	100.691846	6.741278	47	28.260
38.00	สตูล	M	100.025600	6.635370	64	32.978
39.00	ยะลา	M	101.387220	6.336110	84	37.781
39.01	เบตง	A1	101.051735	5.771216	100	41.222
39.02	สุโหงปาตี	A1	101.283149	6.477219	80	36.871
39.03	นราธิวาส	A1	101.801944	6.411111	100	41.222

จาก LINE OF SIGHT EQUATIONS : $D = 3.57\sqrt{KH}$

โดยที่ D : รัศมีการกระจายสัญญาณของเสาส่งสัญญาณที่วัดจัตอล (กิโลเมตร)

H: ความสูงสายอากาศ (เมตร)

K : ADJUSTMENT FACTOR TO ACCOUNT FOR REFRACTION,
RULE OF THUMB (K= 4/3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7.4 สถานที่ส่งสัญญาณทีวีดิจิทัล (สถานีหลัก) [13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 เสาอากาศ (Antenna)

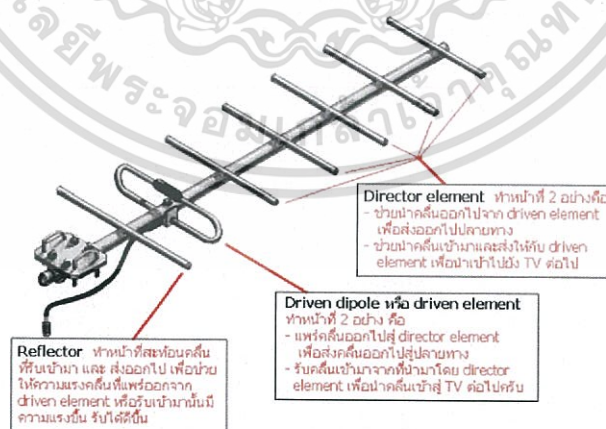
หน้าที่หลักของเสาอากาศก็คือ การแปลงสัญญาณวิทยุไปเป็นคลื่นแม่เหล็กไฟฟ้า เพื่อส่งออกอากาศไปยังภาคส่งคลื่นวิทยุ และทำหน้าที่ในการแปลงคลื่นแม่เหล็กไฟฟ้าที่อยู่ในอากาศไปเป็น สัญญาณวิทยุเพื่อส่งให้ภาครับทำการถอดรหัสข้อมูลออกจากสัญญาณวิทยุต่อไป

สายอากาศมีหลายขนาดและรูปแบบ ขึ้นอยู่กับการใช้งาน เช่น สายอากาศสำหรับเครื่องรับโทรทัศน์ในบ้าน ส่วนใหญ่เป็นสายอากาศชนิด ยากิ-อูตะ มักติดตั้งไว้บนหลังคา ทำด้วยอะลูมิเนียม เพราะน้ำหนักเบาและทนต่อสภาพอากาศได้ดีกว่าโลหะทั่วไป สายอากาศของไม้ค้อยเป็นเพียงสายไฟสั้นๆ หรือสายอากาศของโทรศัพท์มือถือ เป็นเพียงจุดเชื่อมต่อเล็กๆ เท่านั้น

เสาอากาศสามารถแบ่งได้เป็น 2 รูปแบบ จากรูปแบบการกระจายของคลื่นคือ

1) เสาอากาศแบบรอบตัว (Omni Directional Antenna) เสาอากาศประเภทนี้มีทิศทางการแพร่กระจายคลื่นรอบทิศทาง 360 องศา เหมาะสำหรับใช้ติดต่อกับเครื่องลูกข่ายที่เคลื่อนไหวยู่ในตำแหน่งและทิศทางที่ไม่แน่นอน

2) เสาอากาศแบบทิศทาง (Directional Antenna) เป็นเสาอากาศที่มีทิศทางการแพร่กระจายของคลื่นที่มีทิศทางชัดเจน เหมาะสำหรับการติดต่อระหว่างจุดต่อจุด สามารถเพิ่มระยะทางการใช้งานได้ไกลกว่าเสาอากาศแบบรอบตัว แต่มีข้อเสียคือ ถ้าไม่อยู่ในทิศทางการแพร่กระจายของคลื่นจะไม่สามารถรับสัญญาณได้เลย ชื่อของเสาอากาศแบบนี้ที่นิยมใช้กันได้แก่ เสาอากาศแบบเซ็กเตอร์ (Sector Antenna) เสาอากาศแบบยากิ (Yagi Antenna) เสาอากาศแบบกริด (Grid Antenna) เสาอากาศแบบจาน (Solid Dish Antenna) อัตราขยายของเสาอากาศแบบนี้อยู่ใน Gain สูง



รูปที่ 2.8 องค์ประกอบของเสาอากาศแบบยากิ (Yagi Antenna) [14]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับระบบ Digital TV ในประเทศไทยนั้น ใช้ความถี่ย่าน UHF ในการออกอากาศ ดังนั้นจึงควรเลือกใช้เสาอากาศที่รองรับความถี่ย่าน UHF ในการรับสัญญาณ ซึ่งเสาอากาศที่รองรับความถี่ย่าน UHF นั้นมีทั้งเสาอากาศชนิดที่ใช้ภายในบ้าน (เช่นเสาหนวดกุ้ง) และเสาอากาศชนิดที่ใช้ภายนอกบ้าน (เช่นเสาข้างปลา) โดยเสาอากาศทั้งสองชนิดนี้ยังสามารถแบ่งได้เป็นประเภท Active และ Passive ดังตารางที่ 2.8.1

ตารางที่ 2.8.1 แสดงคุณสมบัติของเสาอากาศแต่ละประเภท [14]

ประเภทของเสาอากาศ (Antenna)	
เสาอากาศชนิดที่ใช้ภายในบ้าน (Indoor Antenna)	Active
	Passive (เช่นเสาหนวดกุ้ง)
เสาอากาศชนิดที่ใช้ภายนอกบ้าน (Outdoor Antenna)	Active
	Passive (เช่น เสาข้างปลา)

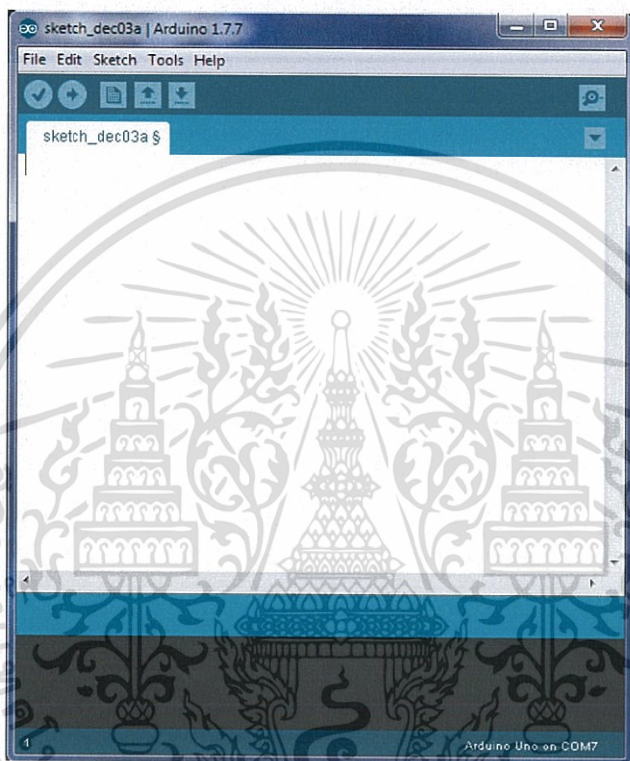
เสาอากาศประเภท Active คือ เสาอากาศที่ต้องใช้ไฟเลี้ยงในการทำงานโดยจะมีวงจรอิเล็กทรอนิกส์อยู่ภายใน เพื่อทำการเพิ่มความแรงสัญญาณที่ได้รับให้สูงขึ้น

เสาอากาศประเภท Passive คือ เสาอากาศที่ไม่ต้องใช้ไฟเลี้ยงในการทำงาน

2.9 การใช้งานโปรแกรม Arduino เบื้องต้น

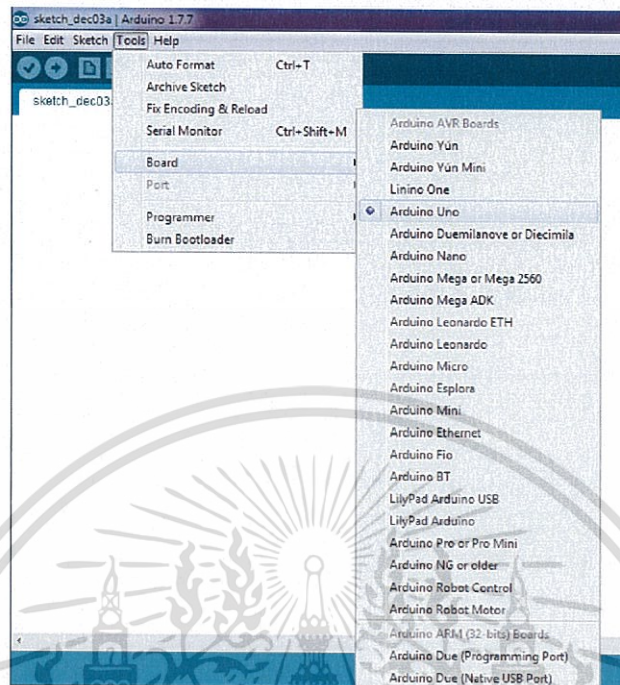
2.9.1 การเริ่มใช้งานเบื้องต้น

2.9.1.1 เชื่อมต่อบอร์ด Arduino เข้ากับคอมพิวเตอร์ผ่านทางพอร์ต USB จากนั้นเปิดโปรแกรม Arduino จะปรากฏหน้าต่างดังรูปที่ 2.9.1.1



รูปที่ 2.9.1.1 หน้าต่างเริ่มต้นโปรแกรม Arduino

2.9.1.2 การตั้งค่า Board คลิกเมนู Tools >> Board >> เลือก Arduino UNO กรณีใช้รุ่นอื่นก็สามารถเลือกให้ตรงกับรุ่นที่ใช้งานได้ ดังรูปที่ 2.9.1.2



รูปที่ 2.9.1.2 การตั้งค่า Board

2.9.1.3 การตั้งค่า Port ใช้งาน ให้เราไปที่ Tools >> Serial Port >> เลือก COM 9 ดังรูปที่ 2.9.1.3



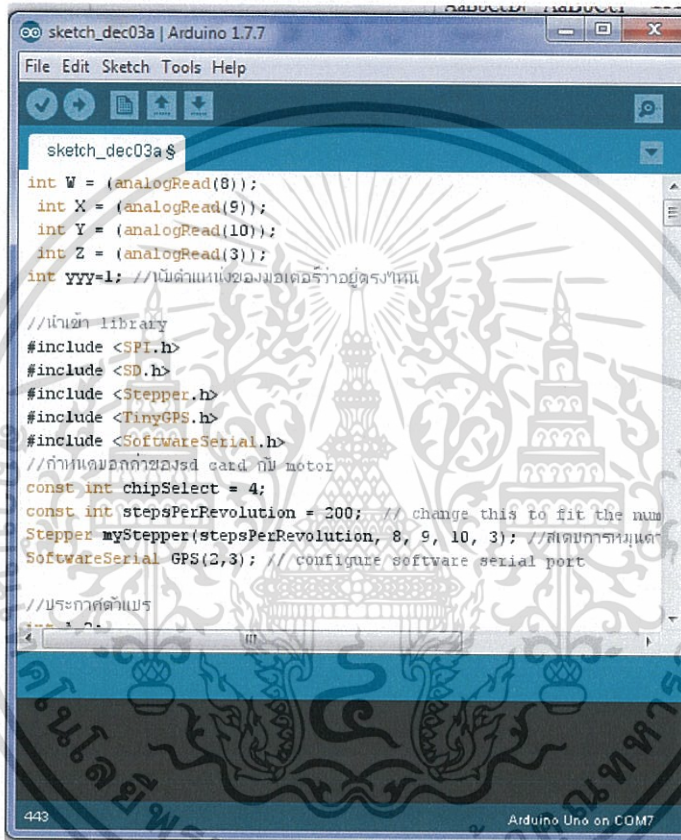
รูปที่ 2.9.1.3 การตั้งค่า Port การใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 การ Upload โปรแกรม

เมื่อทำการเชื่อมต่อบอร์ด ARDUINO เข้ากับคอมพิวเตอร์ผ่านทางพอร์ท USB ทำการตั้งค่าเริ่มต้นต่างๆแล้ว โดยโปรแกรมจะสร้างไฟล์ SKETCH เพื่อใช้ในการโปรแกรมลงบอร์ด ARDUINO การ UPLOAD โปรแกรมสามารถทำได้ดังนี้

2.9.2.1 ทำการเขียนโปรแกรมที่ใช้ในการควบคุม ดังรูปที่ 2.9.2.1



```

sketch_dec03a $
int W = (analogRead(8));
int X = (analogRead(9));
int Y = (analogRead(10));
int Z = (analogRead(3));
int yyy=1; //เป็นตำแหน่งของมอเตอร์ว่าอยู่ตรงไหน

//นำเข้า library
#include <SPI.h>
#include <SD.h>
#include <Stepper.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>

//กำหนดค่าของ card กับ motor
const int chipSelect = 4;
const int stepsPerRevolution = 200; // change this to fit the num
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 3); //ใส่การหมุนเด
SoftwareSerial GPS(2,3); // configure software serial port

//ประกาศตัวแปร
  
```

รูปที่ 2.9.2.1 โปรแกรมควบคุมการทำงานของระบบ

2.9.2.2 ทำการบันทึกโปรแกรมที่ใช้ในการควบคุมรูปแบบ Sketch.ino โดยคลิกที่

เม นู



เมื่อทำการบันทึกเรียบร้อยแล้วจะปรากฏข้อความ done saving ดังรูปที่ 2.9.2.2

```


sketch_dec03a
int V = (analogRead(8));
int X = (analogRead(9));
int Y = (analogRead(10));
int Z = (analogRead(3));
int yyy=1; //เป็นตำแหน่งของมอเตอร์ว่าอยู่ตรงไหน

//นำเข้า library
#include <SPI.h>
#include <SD.h>
#include <Stepper.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>
//กำหนดขั้วต่อของsd card กับ motor
const int chipSelect = 4;
const int stepsPerRevolution = 200; // change this to fit the num
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 3); //กำหนดการหมุนของ
SoftwareSerial GPS(2,3); // configure software serial port

//ประกาศตัวแปร
Done Saving

```

รูปที่ 2.9.2.2 การบันทึกโปรแกรมที่ใช้ในการควบคุมรูปแบบ Sketch.ino

จากรูปที่ 2.9.2.2 เมื่อทำการ Upload โปรแกรมเข้าบอร์ด Arduino UNO โดยไปที่เมนู File->Upload หรือคลิกเมนู  เมื่อทำการ Upload โปรแกรม เรียบร้อยแล้วจะปรากฏข้อความ done uploading ดังรูปที่ 2.9.2.3

```

sketch_dec03a
//นำเข้า library
#include <SPI.h>
#include <SD.h>
#include <Stepper.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>
//กำหนดขั้วต่อของsd card กับ motor
const int chipSelect = 4;
const int stepsPerRevolution = 200; // change this to fit the num
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 3); //กำหนดการหมุนของ
SoftwareSerial GPS(2,3); // configure software serial port

//ประกาศตัวแปร
int A;
int N;
int P,Q,R,S;
File myFile;
int i=0;
int pin1, pin2, pin3, pin4;
int motorPin1 = 8;
Done uploading

```

รูปที่ 2.9.2.3 หลังจากทำการ Upload โปรแกรม

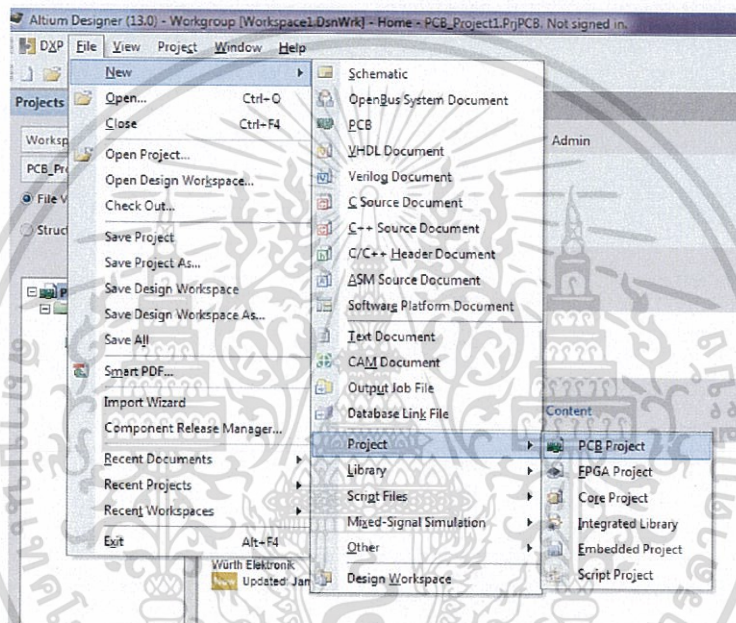
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 การใช้โปรแกรม Altium Designer เบื้องต้น

2.10.1 การออกแบบวงจรใน Schematic

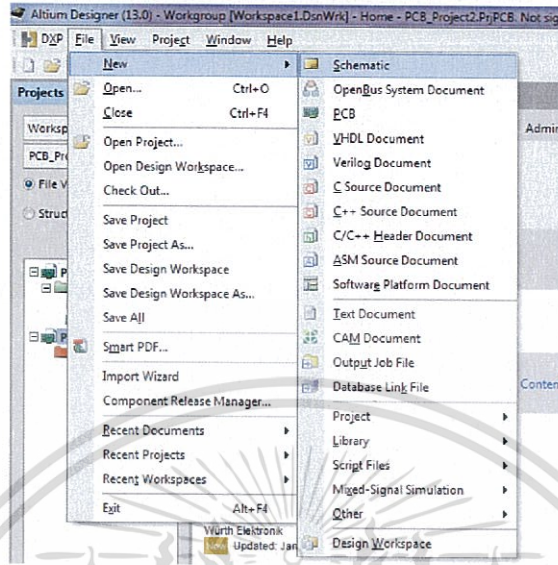
โดย schematic จะมีหน้าที่รวบรวมสัญลักษณ์ต่างๆ และการเชื่อมต่อของสัญลักษณ์ต่างๆ ก่อนที่จะนำไปออกแบบ PCB มีขั้นตอนดังต่อไปนี้

2.10.1.1 การสร้าง New Project ขึ้นมาโดยเลือกเมนู File --> New --> Project --> PCB Project



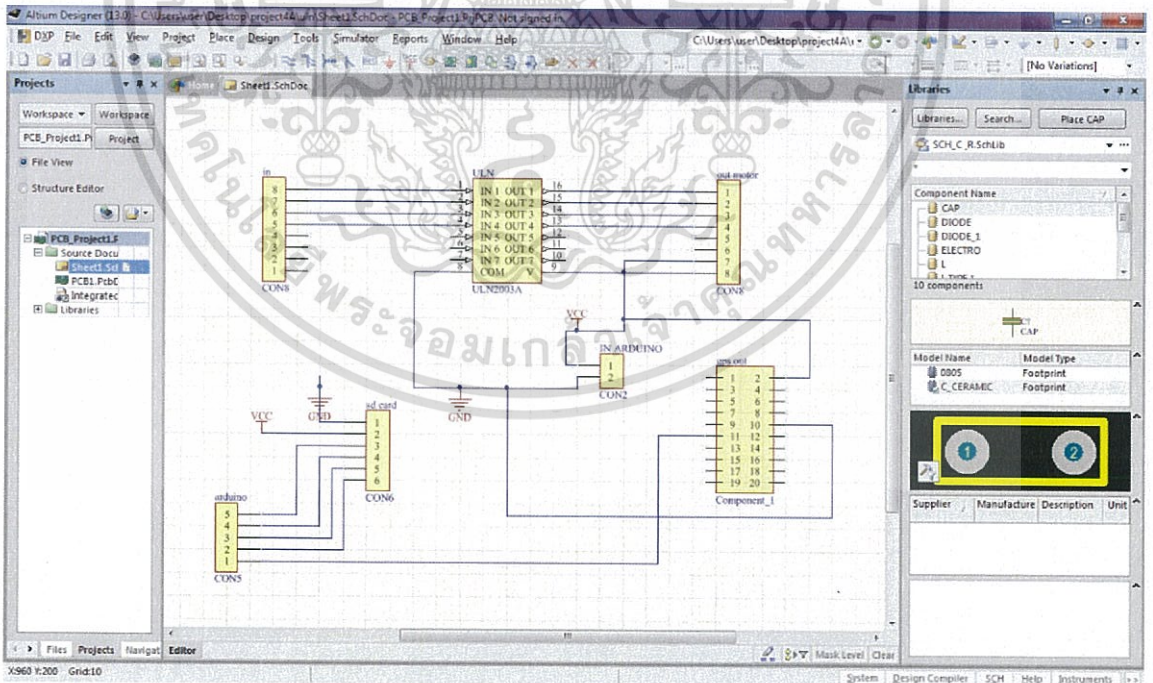
รูปที่ 2.10.1.1 ตัวอย่างการสร้าง New Project

2.10.1.2 สร้างพื้นที่ Schematic ขึ้นมาโดย File --> New --> Schematic



รูปที่ 2.10.1.2 ตัวอย่างการสร้างพื้นที่ Schematic

2.10.1.2 นำอุปกรณ์ต่างๆวางลงบน schematic โดยเลือกอุปกรณ์ในแถบ Libraries จากนั้นทำการจัดวางและลากสายสัญญาณให้เชื่อมต่อกันตามต้องการ

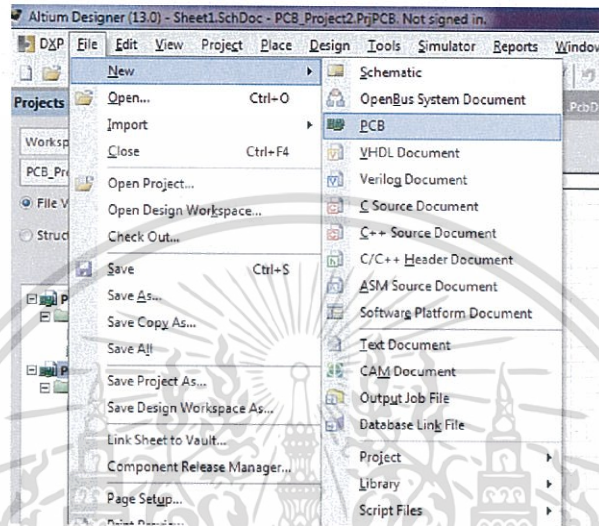


รูปที่ 2.10.1.3 การนำอุปกรณ์ต่างๆวางลงบน schematic และทำการลากสายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

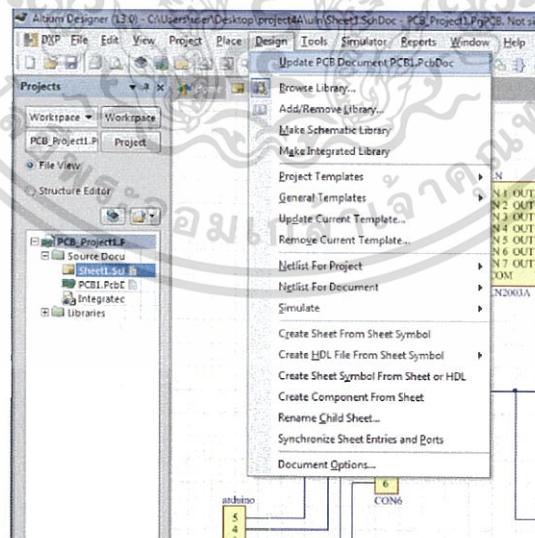
2.10.2 การออกแบบ PCB

2.10.2.1 การสร้างไฟล์ PCB โดยใช้คำสั่ง File --> New --> PCB ไฟล์ PCB จะถูกสร้างขึ้นที่ไฟล์ Project ที่สร้างไว้



รูปที่ 2.10.2.1 ตัวอย่างการสร้างไฟล์ PCB

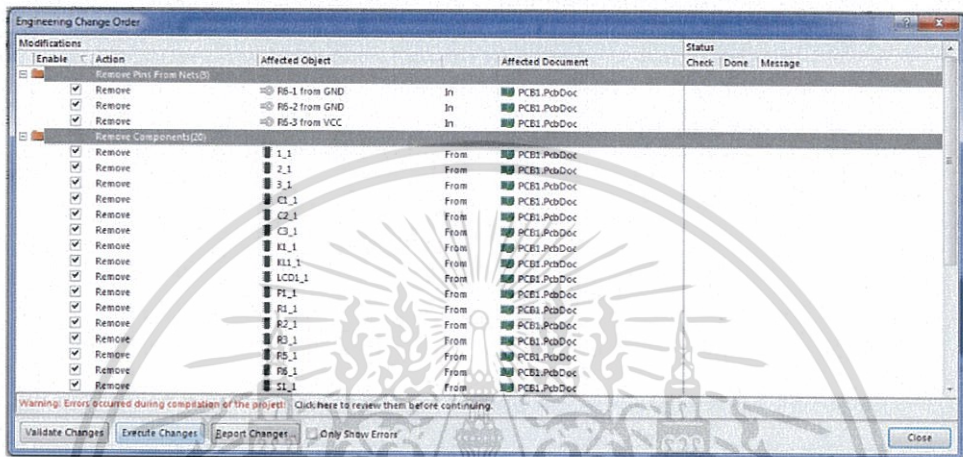
2.10.2.2 การสร้างไฟล์ PCB จากไฟล์ Schematic ที่ได้สร้างไว้ก่อนหน้านี้ โดยใช้คำสั่ง Design --> Update PCB



รูปที่ 2.10.2.2 การสร้างไฟล์ PCB จากไฟล์ Schematic

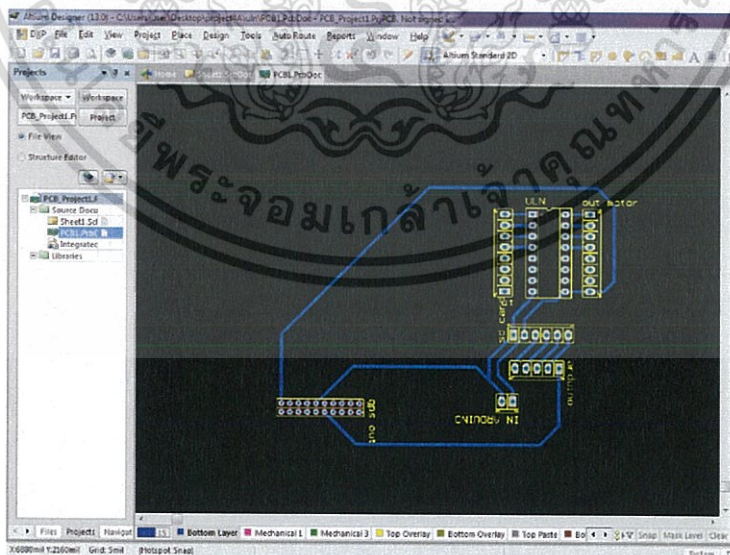
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.2.3 การตรวจข้อผิดพลาดต่างๆ โปรแกรม จะแสดงหน้าต่าง Engineering Change Order ให้คลิก Execute Changes โปรแกรมจะทำการตรวจข้อผิดพลาดต่างๆ ถ้ามีข้อผิดพลาดจะมีสัญลักษณ์กากบาทสีแดงขึ้น ให้กลับไปตรวจสอบที่ Schematic ถ้าไม่มีข้อผิดพลาด Footprint จะไปปรากฏที่ไฟล์ PCB



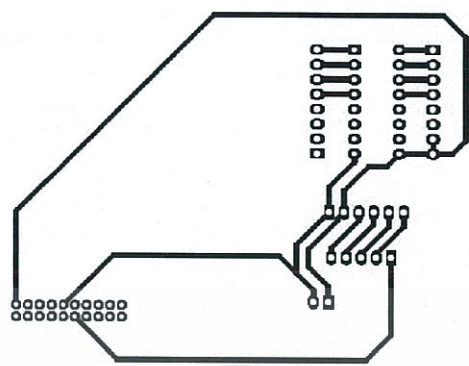
รูปที่ 2.10.2.3 หน้าต่าง Engineering Change Order ในขั้นตอนการตรวจข้อผิดพลาดต่างๆ

2.10.2.4 ทำการจัดวาง Footprint ตามที่ต้องการ แล้วทำการเดินเส้น Track เชื่อมอุปกรณ์ต่างๆเข้าด้วยกัน



รูปที่ 2.10.2.4.1 การจัดวาง Footprint และเดินเส้น Track

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10.2.4.2 ภาพรวมของสายพีซีซี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

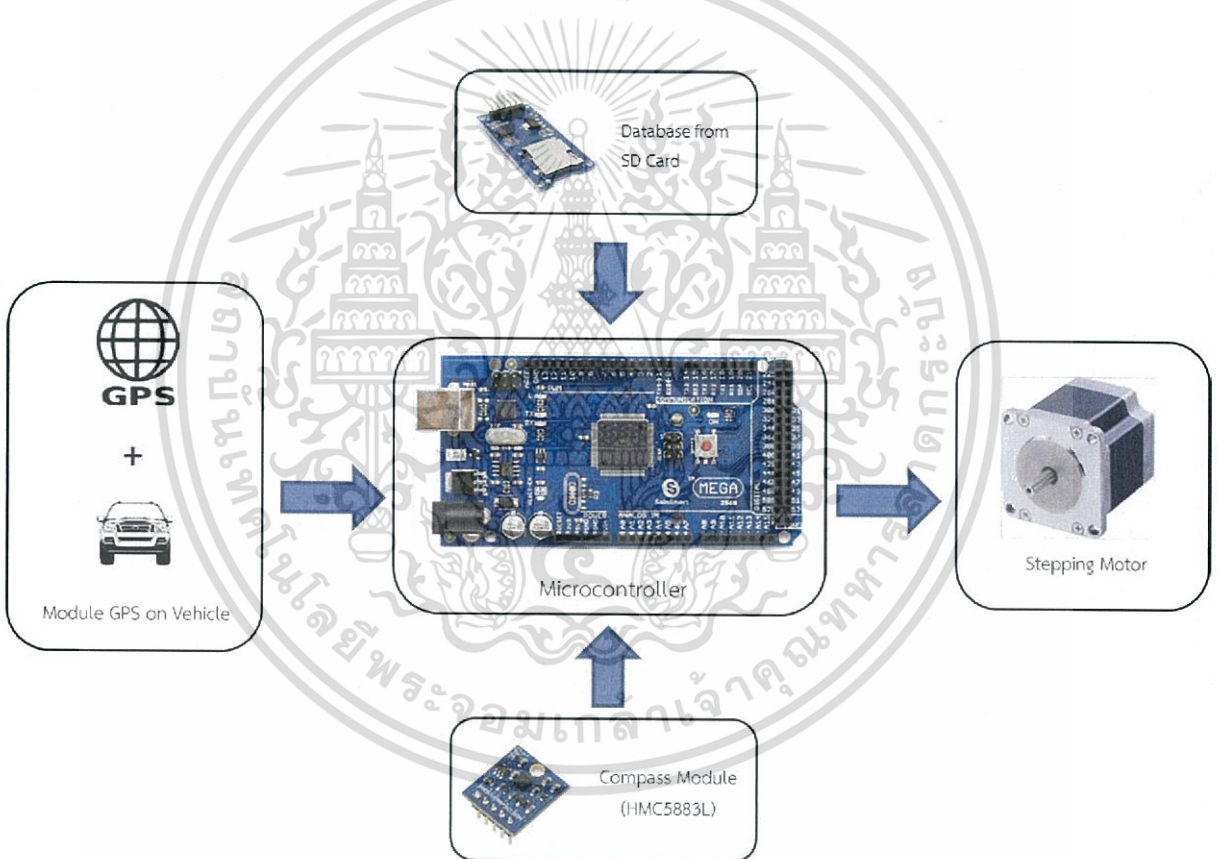
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบทางฮาร์ดแวร์

การออกแบบฮาร์ดแวร์ของระบบโดยใช้ ARDUINO ในการควบคุมการทำงานแต่ละส่วนของระบบ แสดงผังแผนผังการทำงานได้ดังรูปที่ 3.1.1



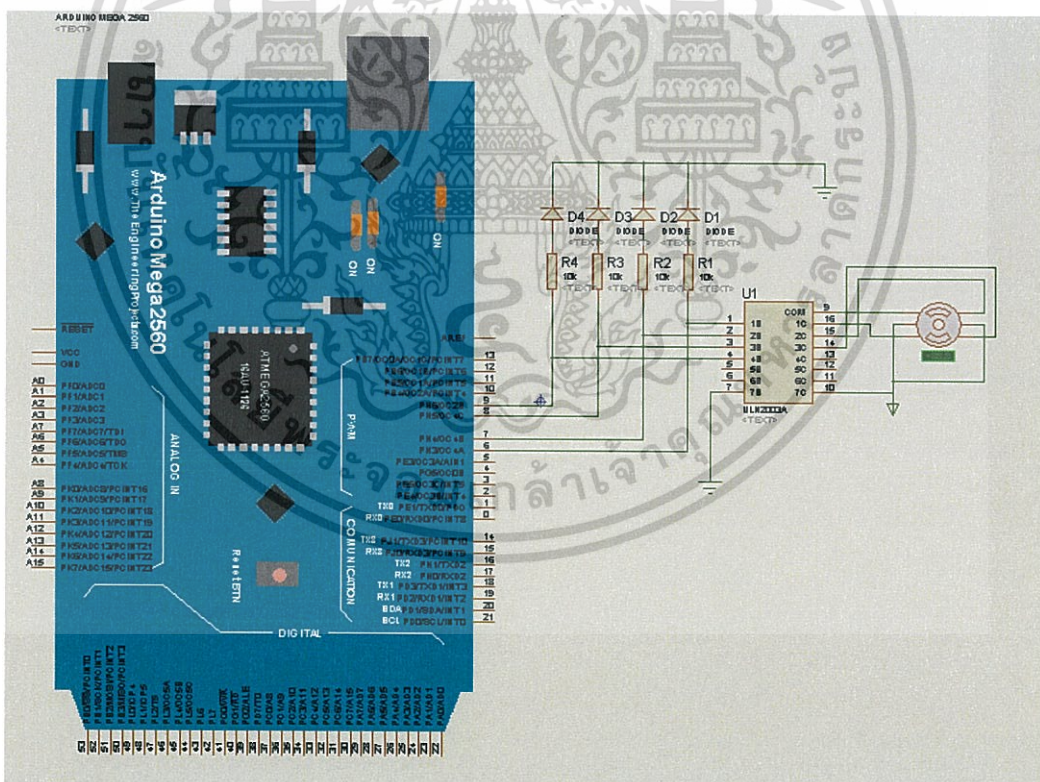
รูปที่ 3.1.1 BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นการทำงาน โมดูลเซ็นเซอร์ส่งข้อมูลทิศทางของยานพาหนะ เพื่อปรับทิศทางเหนือ โมดูลจีพีเอสส่งข้อมูลพิกัดตำแหน่งของยานพาหนะ เพื่อนำไปเปรียบเทียบกับตำแหน่งและข้อมูล การกระจายสัญญาณของเสาส่งสัญญาณที่วิดิจิตอล โดยใช้ไมโครคอนโทรลเลอร์ในการควบคุมและ ปรับทิศทางสายอากาศเพื่อจับ ความเข้มของสัญญาณที่ดีที่สุด

3.1.1.1 วงจรขับสเต็ปมอเตอร์

ในระบบการทำงานใช้สเต็ปมอเตอร์ในการปรับทิศทางของสายอากาศ เพื่อรับความเข้มของสัญญาณที่ดีที่สุด ซึ่งการควบคุมอุปกรณ์ขับเคลื่อนสเต็ปมอเตอร์ จะใช้ กระแสขับสูงเพื่อสร้างแรงบิด จึงไม่สามารถต่อกับ MICROCONTROLLER ได้โดยตรง ดังนั้นต้องมี อุปกรณ์ขับกระแสเชื่อมต่อระหว่าง MICROCONTROLLER กับ สเต็ปมอเตอร์ คือ IC ขับ กระแสเบอร์ ULN2003 ซึ่งจ่ายกระแสสูงสุด 500 MA ใช้สำหรับควบคุมสเต็ปมอเตอร์แบบ UNIPOLAR เท่านั้น แสดงดังรูปที่ 3.1.1.1

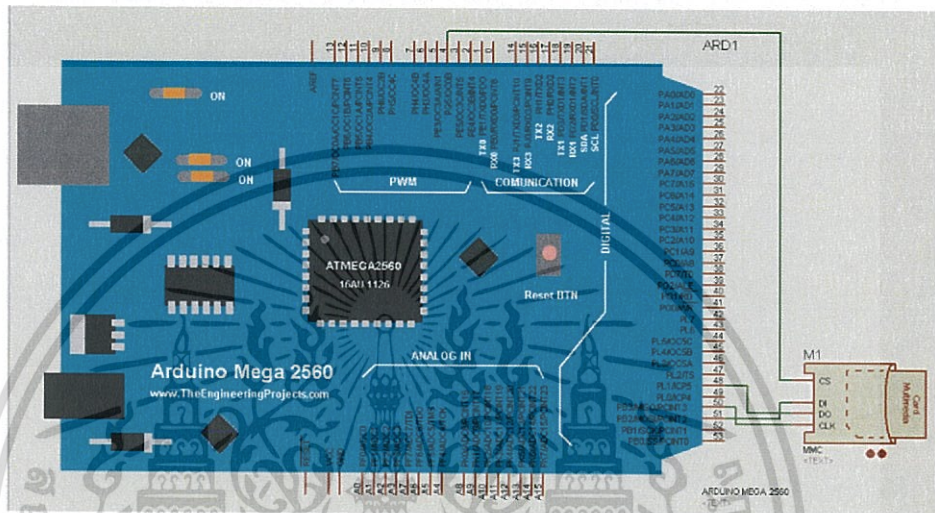


รูปที่ 3.1.1.1 วงจรขับสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2 วงจร SD CARD

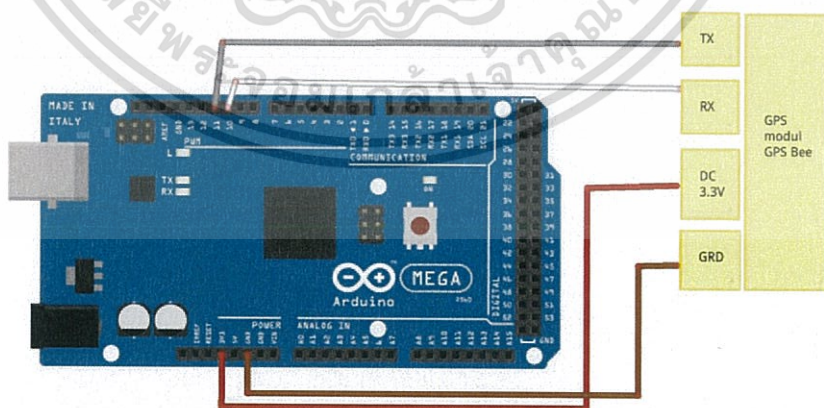
ในส่วนของการเก็บข้อมูลพิกัดตำแหน่งและข้อมูลการกระจายสัญญาณของเสาส่งสัญญาณทีวีดิจิตอล นั้นจะจัดเก็บในรูปแบบไฟล์ .TEXT ใน SD DARD ซึ่งมีการต่อเข้ากับ ARDUINO แสดงดังรูปที่ 3.1.1.2



รูปที่ 3.1.1.2 วงจร SD Card

3.1.1.3 วงจร MODULE GPS

ในส่วนของการหาข้อมูลพิกัดตำแหน่งปัจจุบันของยานพาหนะ โดยใช้ MODULE GPS ซึ่งมีการต่อเข้ากับ ARDUINO แสดงดังรูปที่ 3.1.1.3

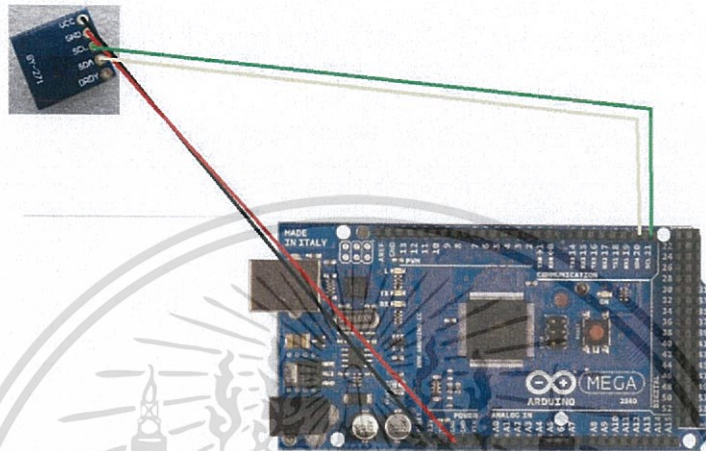


รูปที่ 3.1.1.3 วงจร MODULE GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.4 วงจร COMPASS MODULE

ในส่วนของการหาทิศทางของยานพาหนะ โดยใช้ COMPASS MODULE ซึ่งมีการต่อเข้ากับ ARDUINO แสดงดังรูปที่ 3.1.1.4



รูปที่ 3.1.1.4 วงจร COMPASS MODULE

3.1.2 การออกแบบทางซอฟต์แวร์

การออกแบบทางซอฟต์แวร์โดยใช้การเขียนโปรแกรมด้วยภาษา C โดยใช้โปรแกรม ARDUINO ในการเขียนคำสั่งการทำงานทั้งหมดของระบบโดยมีขั้นตอนการทำงานดังนี้

```

sketch_apr07a | Arduino 1.7.7
File Edit Sketch Tools Help
sketch_apr07a

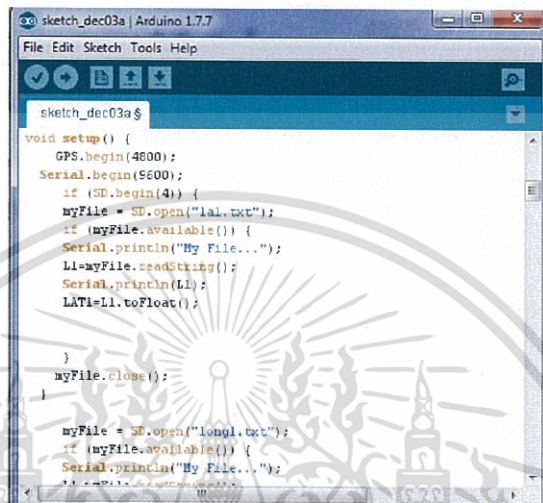
void setup(){
  Serial.begin(9600); // connection serial
  gpsSerial.begin(9600); // gps burd rate
  Wire.begin();
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, LOW);
  digitalWrite(motorPin3, LOW);
  digitalWrite(motorPin4, LOW);
  rdsd();
  setupHMC5883L(); //setup the HMC5883L
  compass = HMC5883L(); //new instance of HMC5883L library
  Gen();
}

```

รูปที่ 3.1.2.1 โปรแกรมดึงค่าจากโมดูลเข็มทิศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

จากรูปที่ 3.1.2.1 แสดงโปรแกรมดึงค่าจากโมดูลเข็มทิศเพื่อควบคุมให้สแต็ป มอเตอร์หมุนสายอากาศชี้ทิศทางไปยังทิศเหนือ โดยโปรแกรมในการดึงค่าพิกัดเสาส่งสัญญาณ ทีวีดิจิตอลจาก SD CARD แสดงในรูปที่ 3.1.2.2



```

sketch_dec03a | Arduino 1.7.7
File Edit Sketch Tools Help

sketch_dec03a $

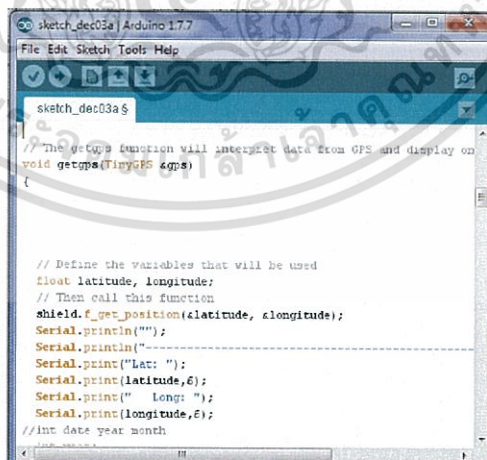
void setup() {
  GPS.begin(4800);
  Serial.begin(5600);
  if (SD.begin(4)) {
    myFile = SD.open("lat.txt");
    if (myFile.available()) {
      Serial.println("My File...");
      LI=myFile.readString();
      Serial.println(LI);
      LATI=LI.toFloat();
    }
    myFile.close();
  }

  myFile = SD.open("long1.txt");
  if (myFile.available()) {
    Serial.println("My File...");
  }
}

```

รูปที่ 3.1.2.2 โปรแกรมดึงค่าพิกัดเสาส่งสัญญาณทีวีดิจิตอล จาก SD Card

จากรูปที่ 3.1.2.2 แสดงโปรแกรมในการดึงค่าพิกัดเสาส่งสัญญาณทีวี ดิจิตอลจาก SD CARD เพื่อนำไปเปรียบเทียบกับค่าพิกัดตำแหน่งปัจจุบันของยานพาหนะ ตั้งโปรแกรมในรูปที่ 3.1.2.3



```

sketch_dec03a | Arduino 1.7.7
File Edit Sketch Tools Help

sketch_dec03a $

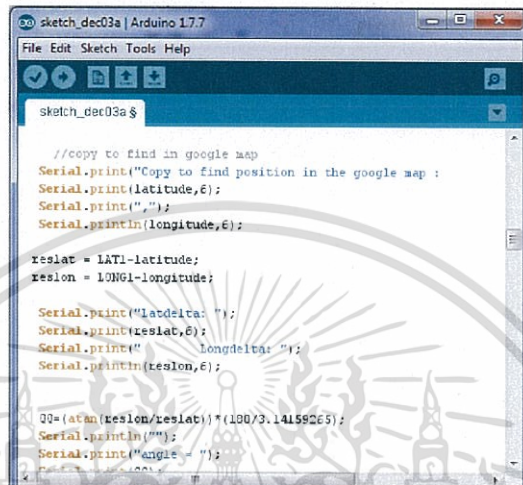
// The getgps function will interpret data from GPS and display on
void getgps(TinyGPS &gps)
{
  // Define the variables that will be used
  float latitude, longitude;
  // Then call this function
  shield.f_get_position(&latitude, &longitude);
  Serial.println("");
  Serial.println("-----");
  Serial.print("LAT: ");
  Serial.print(latitude,6);
  Serial.print(" long: ");
  Serial.print(longitude,6);
  //int date year month
}

```

รูปที่ 3.1.2.3 โปรแกรมควบคุมการทำงานของ GPS MODULE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1.2.3 แสดงโปรแกรมควบคุมการทำงานของ GPS MODULE ในการหาพิกัดตำแหน่งปัจจุบันของยานพาหนะเพื่อนำค่า LATITUDE และ LONGITUDE ที่ได้ไปคำนวณมุมในการหมุนตั้งโปรแกรมในรูปที่ 3.1.2.4



```

sketch_dec03a | Arduino 1.7.7
File Edit Sketch Tools Help

sketch_dec03a $

//copy to find in google map
Serial.print("Copy to find position in the google map :
Serial.print(latitude,6);
Serial.print(",");
Serial.println(longitude,6);

reslat = LAT1-latitude;
reslon = LONG1-longitude;

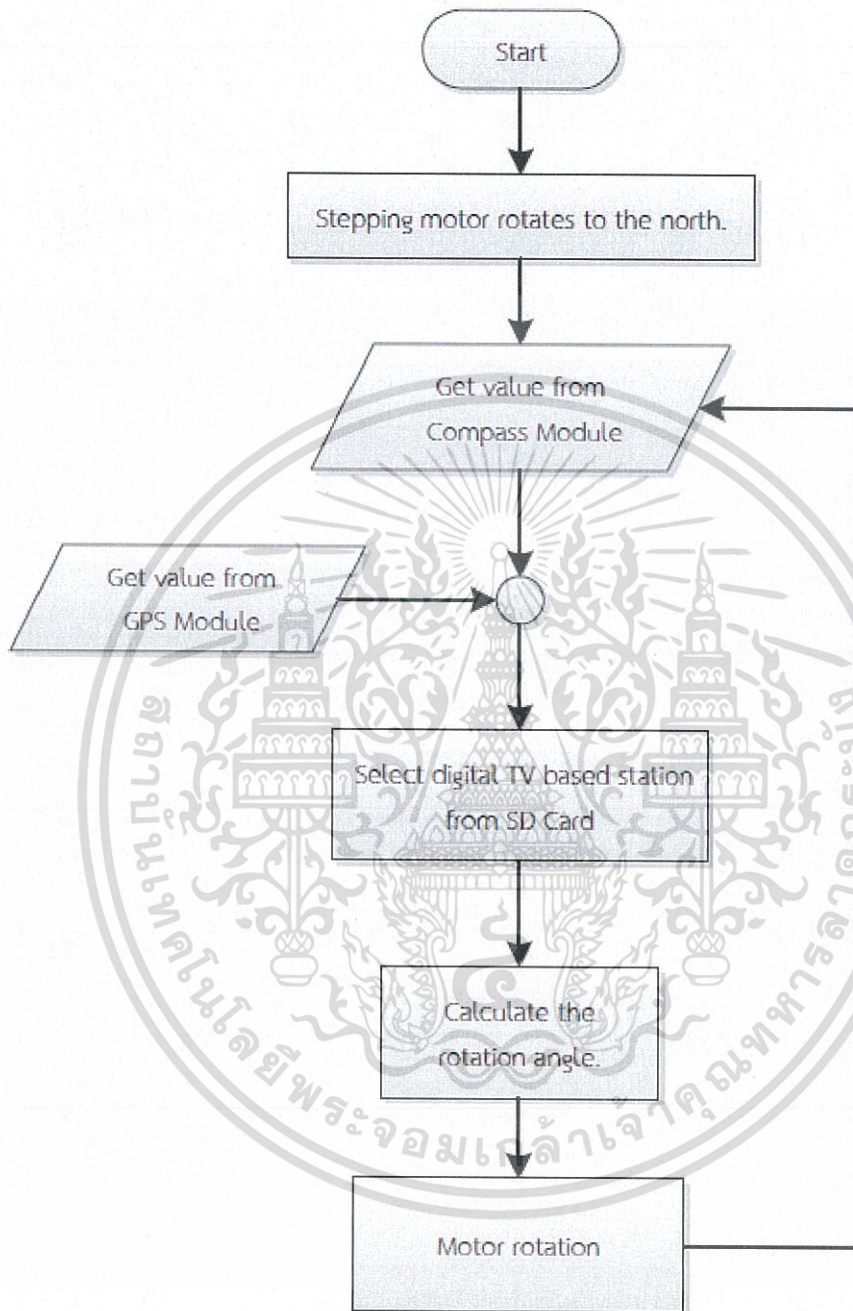
Serial.print("latdelta: ");
Serial.print(reslat,6);
Serial.print(" Longdelta: ");
Serial.println(reslon,6);

Q0=(atan(reslon/reslat))*(180/3.14159265);
Serial.println(" ");
Serial.print("angle = ");
Serial.println(Q0);
  
```

รูปที่ 3.1.2.4 โปรแกรมคำนวณมุมในการหมุนของสเต็ปมอเตอร์

3.1.2.1 FLOW CHART

การทำงานโมดูลจีพีเอสส่งข้อมูลพิกัดตำแหน่งของยานพาหนะ เพื่อนำไปเปรียบเทียบกับตำแหน่งและข้อมูลการกระจายสัญญาณของเสาส่งสัญญาณทีวีดิจิตอล โดยใช้ ARDUINO ในการควบคุมและปรับทิศทางสายอากาศเพื่อจับ ความเข้มของสัญญาณที่ดีที่สุด แสดงดังรูปที่ 3.1.2.1.1



รูปที่ 3.1.2.1.1 FLOW CHART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1 การทำงานของระบบ

การทำงานของระบบในการคำนวณให้สแต็ปมอเตอร์ปรับทิศทางสายอากาศหาสถานีส่งสัญญาณทีวีดิจิตอล ยกตัวอย่างเช่น

พิกัดสถานีส่งสัญญาณทีวีดิจิตอล LAT : 13.754302 ,LONG : 100.540275 (อาคารโบหยก 2)

พิกัดตำแหน่งปัจจุบัน LAT : 13.727514 ,LONG : 100.775924

คำนวณค่า LATDELTA = LAT(สถานีส่ง) - LAT(ตำแหน่งปัจจุบัน)

$$= 13.754302 - 13.727514$$

$$= 0.026788$$

คำนวณค่า LONGDELTA = LONG(สถานีส่ง) - LONG(ตำแหน่งปัจจุบัน)

$$= 100.540275 - 100.775924$$

$$= -0.235649$$

คำนวณมุมในการหมุน $\tan^{-1}\left(\frac{LongDelta}{LatDelta}\right) = \tan^{-1}\left(\frac{-0.235649}{0.026788}\right)$
 $= -83.51$

เข้ากรณีในการหมุน LATDELTA > 0 && LONGDELTA < 0

จะได้มุมที่ต้องหมุนคือ $360 + \tan^{-1}\left(\frac{LongDelta}{LatDelta}\right) = 360 - 83.51 = 277$ องศา

3.2 เครื่องมือที่ใช้ในการทดลอง

- | | |
|----------------------------------|--------|
| 1) บอร์ดทดลอง | 1 แผ่น |
| 2) Arduino UNO R3 | 1 ตัว |
| 3) driver stepping motor ULN2003 | 1 ตัว |
| 4) HOLUX GR-83 GPS Module | 1 ตัว |
| 5) GPS Antenna | 1 ตัว |
| 6) SD Card Module | 1 ตัว |
| 7) Stepping Motor | 1 ตัว |
| 8) เสาอากาศรับสัญญาณทีวีดิจิตอล | 1 ตัว |
| 9) Compass Module | 1 ตัว |
| 10) สายสำหรับต่ออุปกรณ์ต่างๆ | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดเก็บผลการทดลอง

ระบบค้นหาทิศทางสายอากาศดิจิทัลที่วีไอทีโนมัติสาหรับยานพาหนะ ประกอบด้วยการทำงานร่วมกันของ GPS Module ,SD Card Module และ Arduino โดยจะเขียนโปรแกรมให้ไมโครจีสทีเอสส่งข้อมูลพิกัดตำแหน่งของยานพาหนะ ประกอบกับการส่งข้อมูลทิศทางของยานพาหนะจากโมดูลเข็มทิศ แล้วนำไปเปรียบเทียบกับตำแหน่งเสาส่งสัญญาณที่วีดิจิตอล โดยแสดงผลผ่านการหมุนของสตีปั้งมอเตอร์ ซึ่งได้กล่าวไว้ในบทที่ 4



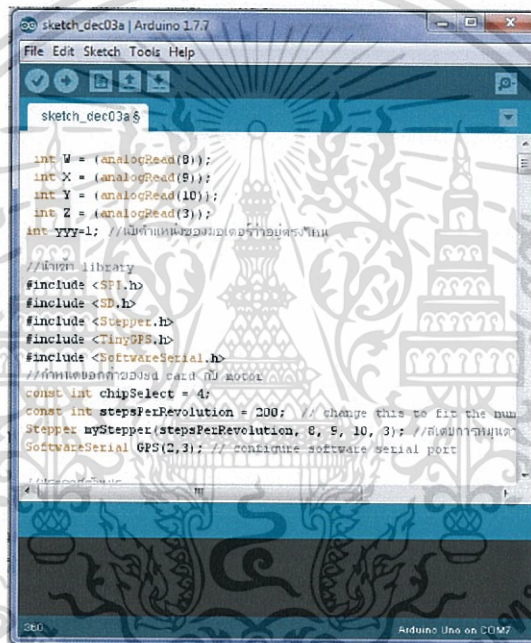
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 การทำงานของระบบ

การทำงานของไมโครคอนโทรลเลอร์ในการควบคุมการทำงานของสเต็ปมอเตอร์ เพื่อควบคุมการหมุนของสายอากาศเพื่อหาเสาส่งสัญญาณทีวีดิจิตอล ดังรูปที่ 4.1.1

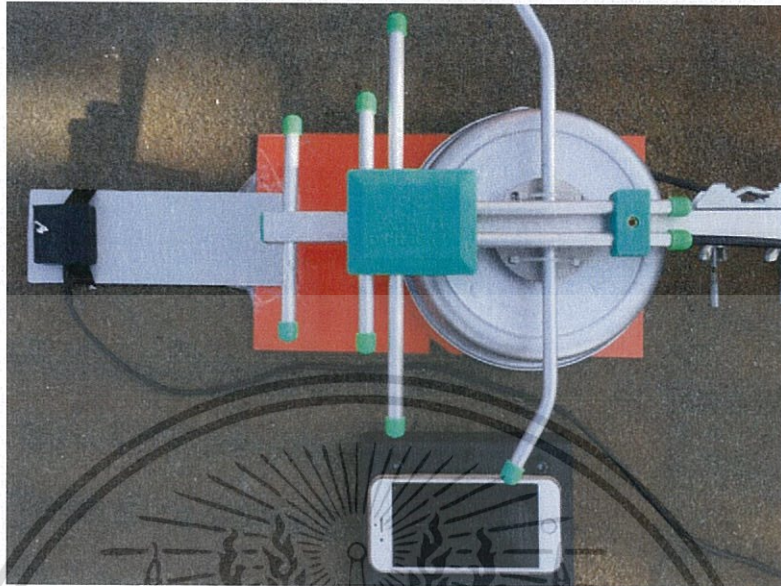


```

sketch_dec03a | Arduino 1.7.7
File Edit Sketch Tools Help
sketch_dec03a.s
int W = (analogRead(8));
int X = (analogRead(9));
int Y = (analogRead(10));
int Z = (analogRead(3));
int yyy=1; //เป็นค่าหนึ่งของมอเตอร์ที่ขยับตรงนั้น
//เพิ่ม library
#include <SPI.h>
#include <SD.h>
#include <Stepper.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>
//กำหนดค่าของ SD card กับ motor
const int chipSelect = 4;
const int stepsPerRevolution = 200; // change this to fit the mm
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 3); //ตั้งค่ากับ motor
SoftwareSerial GPS(2,3); // continue software serial port
  
```

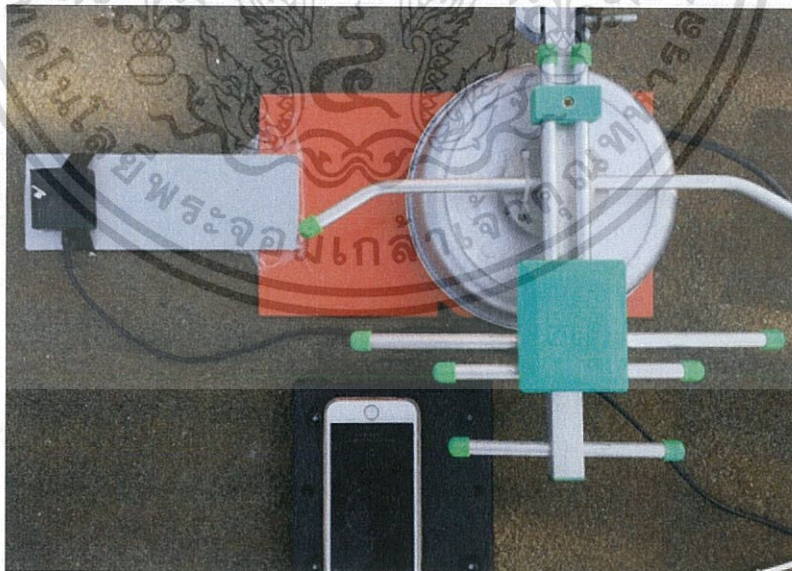
รูปที่ 4.1.1 โปรแกรมควบคุมการทำงาน โดยใช้โปรแกรม Arduino

จากรูปที่ 4.1.1 เมื่อทำการเตรียมโปรแกรมสำหรับควบคุมแล้วทำการต่ออุปกรณ์ Arduino, สเต็ปมอเตอร์, Compass Module และ SD Card Module เข้าด้วยกันเพื่อเตรียมการทดลอง Upload code ดังรูปที่ 4.1.2



รูปที่ 4.1.2 การต่ออุปกรณ์เพื่อเตรียมการทดลอง

จากรูปที่ 4.1.2 ก่อนการทำงานของระบบ กำหนดให้สายอากาศหันไปทางเดียวกับ Compass Module คือทำมุม 90 องศากับทิศเหนือ เมื่อทำการ Upload code สแต็ปปีงมอเตอร์จะหมุนสายอากาศชี้ทิศทางไปยังทิศเหนือ ดังรูปที่ 4.1.3



รูปที่ 4.1.3 สายอากาศหมุนชี้ทิศทางไปยังทิศเหนือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1.3 สเต็ปป์มอเตอร์เมื่อทำการ Upload code สเต็ปป์มอเตอร์จะหมุนชี้ทิศทางไปยังทิศเหนือ คือ 0 องศา โดยเปรียบเทียบกับค่าที่รับมาจาก Compass Module ในตอนแรกคือ 90 องศา ในการคำนวณมุมที่ต้องหมุนหาเสาส่งสัญญาณทีวีดิจิตอลที่ใกล้ที่สุด แสดงใน Serial Monitor ดังรูปที่ 4.1.4

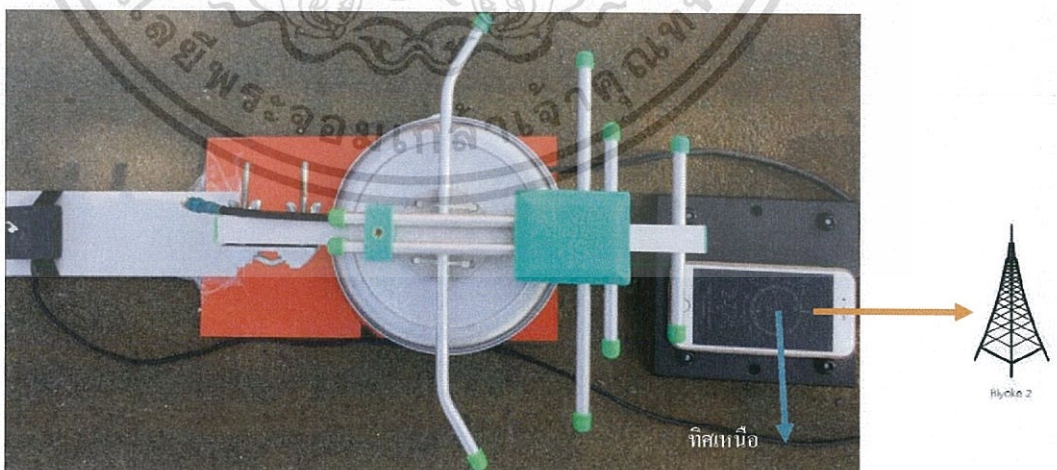
```

13.754302
100.540275
13.754302
100.540275
14.074441
99.420562
279
lat: 13.727514
lon: 100.775924
SWING SELECT BASE ANGLE
277
Stop
SWING LOCK ANGLE
0
Stop

```

รูปที่ 4.1.4 การคำนวณมุมที่ต้องหมุนหาเสาส่งสัญญาณทีวีดิจิตอลที่ใกล้ที่สุด

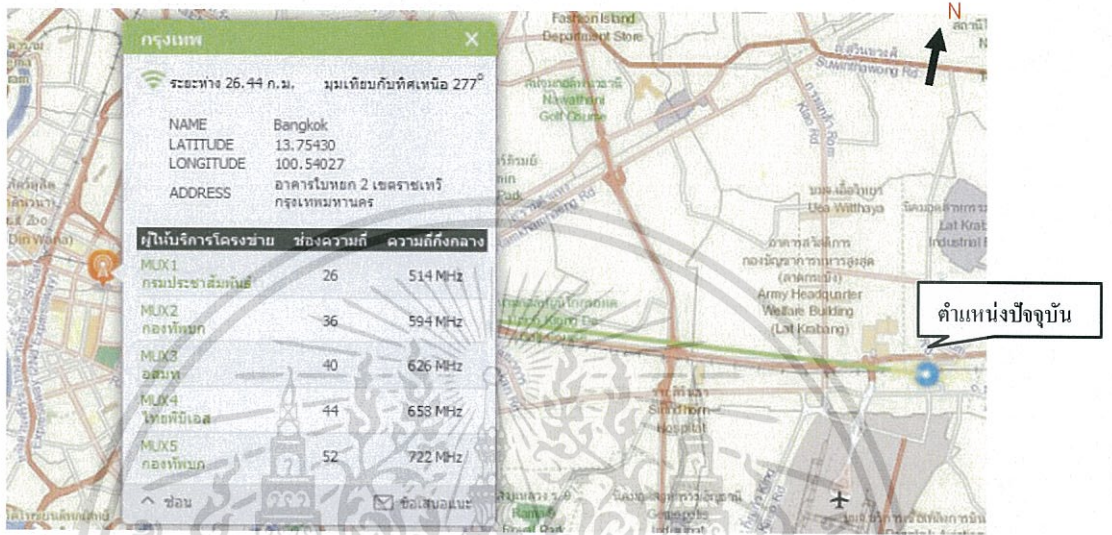
จากรูปที่ 4.1.4 แสดงค่าพิกัดตำแหน่งเสาส่งสัญญาณทีวีดิจิตอลที่ใกล้ที่สุดที่เก็บไว้ใน SD Card คือ 13.754302 ,100.540275 (อาคารโบสถ์ 2) เมื่อนำมาคำนวณกับพิกัดตำแหน่งปัจจุบันคือ 13.727514 ,100.775924 โดยใช้หลักการพิธากอรัส จะได้ค่ามุมที่ต้องหมุนหาเสาส่งสัญญาณทีวีดิจิตอลที่ใกล้ที่สุดเป็น 277 องศา แล้วทำการจำกัดทิศทางของเสาส่งสัญญาณทีวีดิจิตอลนั้นไว้ตลอด ดังรูปที่ 4.1.5



รูปที่ 4.1.5 สายอากาศหมุนหาเสาส่งสัญญาณทีวีดิจิตอลที่ใกล้ที่สุด

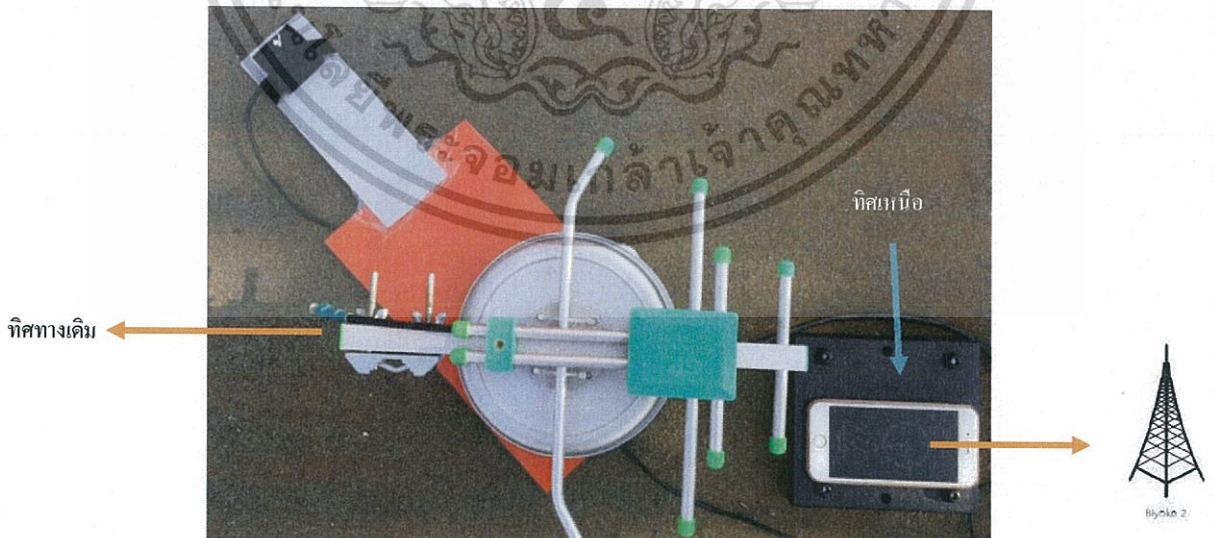
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1.5 สายอากาศหุนหาเสาส่งสัญญาณทีวีดิจิตอลที่ใกล้ที่สุด เป็นมุม 277 องศา ซึ่งตรงกับค่าเมื่อทดสอบมุมในการหุนในระบบตรวจสอบพื้นที่ครอบคลุมสัญญาณทีวีดิจิตอลของ กสทช แสดงดังรูปที่ 4.1.6



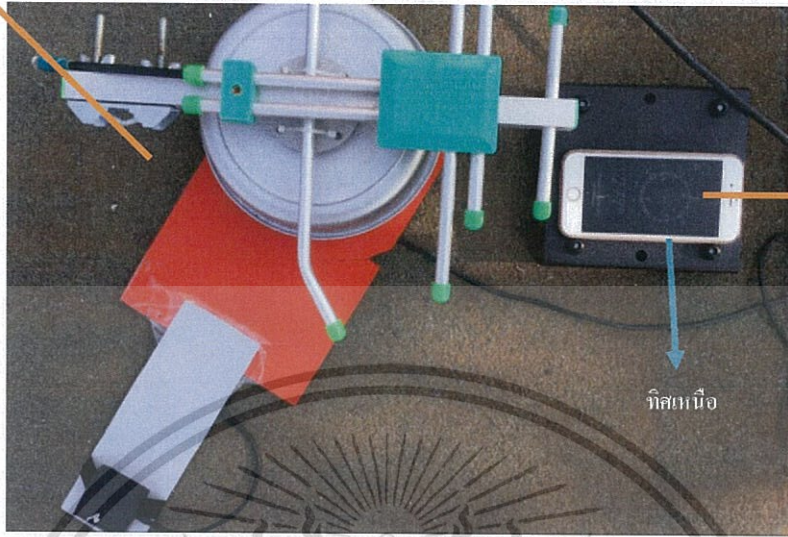
รูปที่ 4.1.6 เมื่อทดสอบมุมในการหุนในระบบตรวจสอบพื้นที่ครอบคลุมสัญญาณทีวีดิจิตอล [15]

จากรูปที่ 4.1.6 เมื่อทำการหุนสายอากาศเพื่อเปลี่ยนไปยังทิศทางอื่น สเต็ปปิ้งมอเตอร์ก็จะหุนสายอากาศชี้ทิศทางไปยังเสาส่งสัญญาณทีวีดิจิตอลนั้นเสมอดังรูปที่ 4.1.7

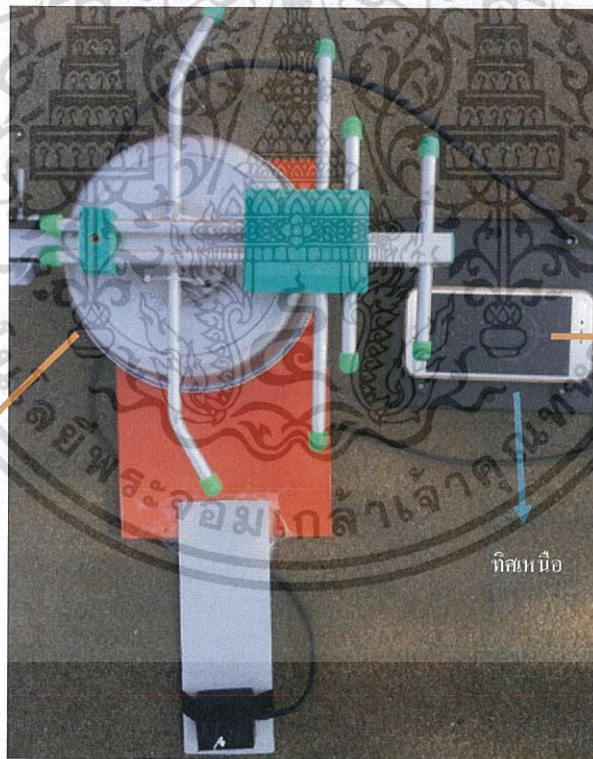


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทิศทางเดิม



ทิศทางเดิม



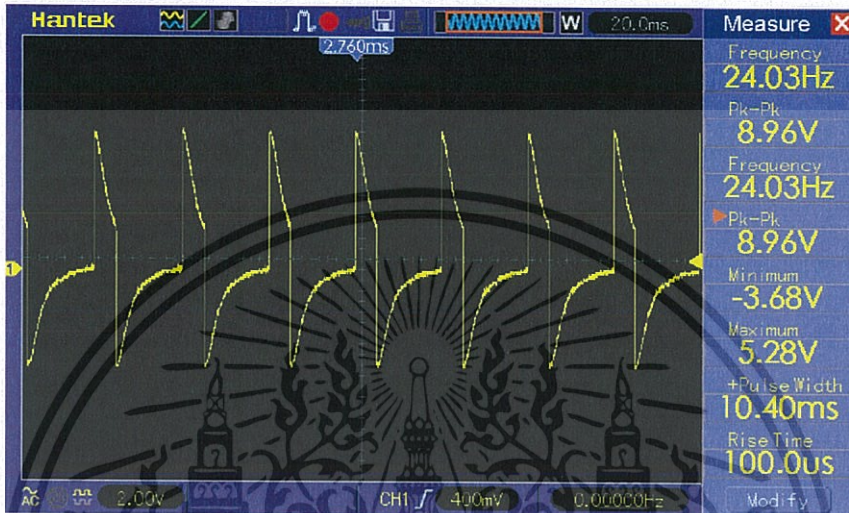
ทิศทางเดิม

ทิศทางเดิม

รูปที่ 4.1.7 เมื่อทำการหมุนสายอากาศเพื่อเปลี่ยนไปยังทิศทางอื่น

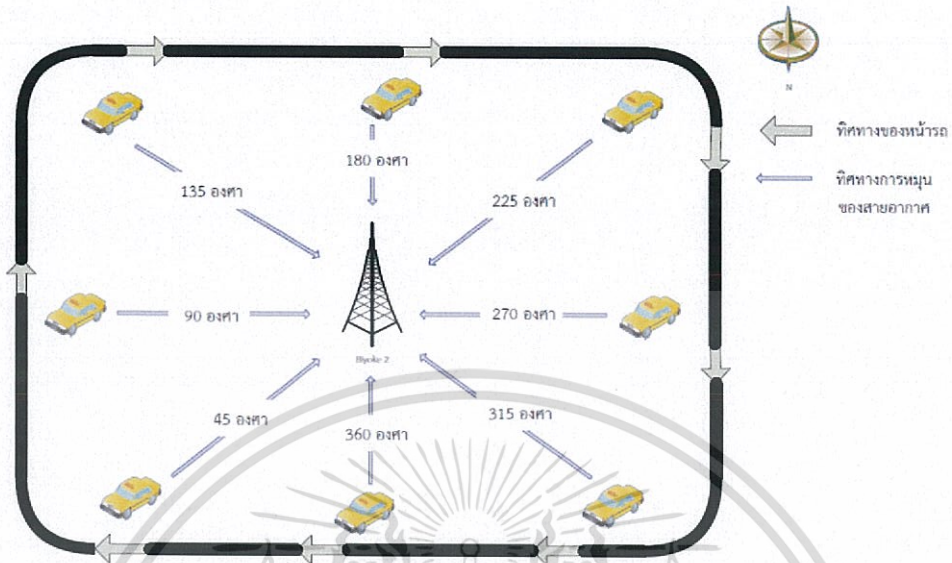
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1.7 เมื่อทำการวัดสัญญาณพัลส์ที่ขาเอาต์พุตของ Arduino ที่ส่งไปยัง สเต็ปป์มอเตอร์ เพื่อส่งให้สเต็ปป์มอเตอร์หมุนสายอากาศซีทีคทางไปยังเสาส่งสัญญาณทีวีดิจิตอล ได้ผลดังรูปที่ 4.1.8



รูปที่ 4.1.8 สัญญาณพัลส์ที่ขาเอาต์พุตของ Arduino

จากรูปที่ 4.1.8 เมื่อยานพาหนะมีการเคลื่อนที่ไปยังตำแหน่งต่างๆ ทำให้ค่าพิกัดตำแหน่งเปลี่ยนแปลงไป สเต็ปป์มอเตอร์ก็ยังคงหมุนสายอากาศซีทีคทางไปยังเสาส่งสัญญาณทีวีดิจิตอลนั้นเสมอ จนกว่าจะมีการเลือกเสาส่งสัญญาณดิจิตอลเสาใหม่ ตั้งแบบจำลองการทำงานของระบบในรูปที่ 4.1.9



รูปที่ 4.1.9 แบบจำลองการทำงานของระบบ

หมายเหตุ การคำนวณมุมในการหมุนจากสูตร $\tan^{-1}\left(\frac{LongDelta}{LatDelta}\right)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญาานิพนธ์ฉบับนี้ได้จัดทำขึ้นเพื่อควบคุมเสอากาศให้จับตำแหน่งเสาส่งสัญญาณทีวีดิจิตอล ในกรณีที่ยานพาหนะเคลื่อนที่ไปยังตำแหน่งต่างๆ โดยการใช้โมดูลจีพีเอสในการส่งข้อมูลพิกัดตำแหน่งของยานพาหนะ และโมดูลเข็มในการส่งข้อมูลทิศทางของยานพาหนะ เมื่อนำฟังก์ชันการทำงานของทั้งสองอุปกรณ์มารวมกับไมโครคอนโทรลเลอร์สามารถใช้ในการควบคุมเสอากาศดิจิตอลทีวีให้สามารถปรับทิศทางเสาเสาส่งสัญญาณทีวีดิจิตอลได้แบบอัตโนมัติ เพื่อแก้ปัญหาการได้รับชมภาพที่มีคุณภาพต่ำเนื่องจากในแต่ละพื้นที่นั้นมีความเข้มของสัญญาณไม่เท่ากัน โดยคำนวณที่คำนวณในการปรับทิศทางแต่ละครั้งมีความคลาดเคลื่อนประมาณ 5 องศา

5.2 ข้อเสนอแนะ

- 1) ในการทดลองเมื่อนำโมดูลจีพีเอสเข้ามาในตัวอาคาร ทำให้การส่งค่าพิกัดตำแหน่งปัจจุบันผิดเพี้ยนไป ส่งผลให้การคำนวณมุมที่ใช้ในการหมุนมีความคลาดเคลื่อน
- 2) โมดูลเข็มทิศไม่มีความเสถียรเมื่อเข้าใกล้วัสดุที่ทำจากเหล็ก ส่งผลให้การอ่านค่ามุมด้านหน้าของยานพาหนะมีความคลาดเคลื่อน

บรรณานุกรม

- [1] Arduino Mega [ออนไลน์].
เข้าถึงได้จาก: <https://www.arduino.cc/en/Main/arduinoBoardMega>
- [2] Arduino Boards-Pin mapping [ออนไลน์].
เข้าถึงได้จาก: <http://icircuit.net/arduino-boards-pin-mapping/141>
- [3] Electricity With Stepper Motors [ออนไลน์].
เข้าถึงได้จาก: <http://www.reuk.co.uk/Electricity-with-Stepper-Motors.htm>
- [4] การตรวจสอบหาเฟสของขดลวดสแต็ปมอเตอร์ [ออนไลน์].
เข้าถึงได้จาก: <http://www.adisak51.com/page22.html>
- [5] ULN2003A driver IC [ออนไลน์].
เข้าถึงได้จาก: <http://www.circuitstoday.com/interfacing-dot-matrix-led-display-to-8051>
- [6] Arduino and HMC5883L [ออนไลน์].
เข้าถึงได้จาก: <http://hobbylogs.me.pn/?p=17>
- [7] Arduino + Compass Module 3-Axis HMC5883L [ออนไลน์].
เข้าถึงได้จาก: <http://fusion94.org/blog/2012/11/29/arduino-plus-compass-module-3-axis-hmc5883l/>
- [8] การใช้งานจีพีเอส (GPS) กับไมโครคอนโทรลเลอร์[ออนไลน์].
เข้าถึงได้จาก: <http://aimagin.com/blog/gps-กับไมโครคอนโทรลเลอร์/?lang=th>
- [9] GPS Ublox NEO-6M Module [ออนไลน์].
เข้าถึงได้จาก: <http://www.arduinoberry.com/product/12/gps-ublox-neo-6m-module>
- [10] Digital TV คืออะไร [ออนไลน์].
เข้าถึงได้จาก: http://www.bict.moe.go.th/2516/doc/4_2.pdf
- [11] ความแตกต่างของ DIGITAL TV กับ ANALOG TV [ออนไลน์].
เข้าถึงได้จาก: http://www.tv5.co.th/technics/tv_digital_in_diff_analog.html
- [12] แผนความถี่วิทยุสำหรับกิจการโทรทัศน์ภาคพื้นดินในระบบดิจิตอล [ออนไลน์].
เข้าถึงได้จาก: <http://www.ratchakitcha.soc.go.th/DATA/PDF/2558/E/180/17.PDF>

[13] Smart Mapping [ออนไลน์].

เข้าถึงได้จาก: <http://www.esri.com/software/arcgis/smart-mapping>

[14] สายอากาศ [ออนไลน์].

เข้าถึงได้จาก: <https://th.wikipedia.org/wiki/สายอากาศ>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <SOFTWARESERIAL.H>
#include <TINYGPS.H>
FLOAT LATITUDE, LONGITUDE;
SOFTWARESERIAL GPSSERIAL(10, 11); // CREATE GPS PIN CONNECTION
TINYGPS SHIELD;

INT RADIUS = 6373;
FLOAT A[3];
FLOAT C[3];
FLOAT DISTANCE[3];
FLOAT DELXARRAY[3]; //K/2
FLOAT DELYARRAY[3];
INT NUM;
INT G1=0;
INT G2=0;
INT G3=0;
INT G;
INT P=0;
INT SW;
INT GENK=0;
INT C1;

#include <HMC5883L.H>
#include <WIRE.H>
HMC5883L COMPASS;
INT ERROR=0;
INT CHECK=0;
INT C2;
INT C;
INT SELECT=0;

#include <SD.H>
#include <SPI.H>
FILE MYFILE;
FLOAT L[28];
INT I=1;
INT K=6;

#include <STEPPER.H>
CONST INT CHIPSELECT = 4;
CONST INT STEPPERREVOLUTION = 200;
STEPPER MYSTEPPER(STEPPERREVOLUTION, 6, 7, 8, 9);
INT MOTORPIN1 = 6;
INT MOTORPIN2 = 7;
INT MOTORPIN3 = 8;
INT MOTORPIN4 = 9;
INT DELAYTIME = 10;

VOID SETUP() {
SERIAL.BEGIN(9600); // CONNECTION SERIAL
GPSSERIAL.BEGIN(9600); // GPS BURD RATE
WIRE.BEGIN();
DIGITALWRITE(MOTORPIN1, LOW);
DIGITALWRITE(MOTORPIN2, LOW);
DIGITALWRITE(MOTORPIN3, LOW);
DIGITALWRITE(MOTORPIN4, LOW);
RDSD();
SETUPHMC5883L(); //SETUP THE HMC5883L
COMPASS = HMC5883L(); //NEW INSTANCE OF HMC5883L LIBRARY
GEN();
}

VOID BLUE() {
DIGITALWRITE(MOTORPIN1, HIGH);
DIGITALWRITE(MOTORPIN2, LOW);
DIGITALWRITE(MOTORPIN3, LOW);
DIGITALWRITE(MOTORPIN4, LOW);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    DELAY (DELAYTIME);
}
VOID YELLOW () {
    DIGITALWRITE (MOTORPIN1, LOW);
    DIGITALWRITE (MOTORPIN2, HIGH);
    DIGITALWRITE (MOTORPIN3, LOW);
    DIGITALWRITE (MOTORPIN4, LOW);
    DELAY (DELAYTIME);
}
VOID ORANGE () {
    DIGITALWRITE (MOTORPIN1, LOW);
    DIGITALWRITE (MOTORPIN2, LOW);
    DIGITALWRITE (MOTORPIN3, HIGH);
    DIGITALWRITE (MOTORPIN4, LOW);
    DELAY (DELAYTIME);
}
VOID RED () {
    DIGITALWRITE (MOTORPIN1, LOW);
    DIGITALWRITE (MOTORPIN2, LOW);
    DIGITALWRITE (MOTORPIN3, LOW);
    DIGITALWRITE (MOTORPIN4, HIGH);
    DELAY (DELAYTIME);
}

VOID GEN ()
{
    IF (GENK==0) {
        INT HEADING = GETHEADING ();
        INT GEN = HEADING;
        SERIAL.PRINTLN (GEN);
        IF (GEN!=0&&GEN>180) {
            GEN = (360-GEN)/1.8;
            FOR (GEN; GEN>0; ) {
                RED ();
                GEN=GEN-1;
                ORANGE ();
                GEN=GEN-1;
                YELLOW ();
                GEN=GEN-1;
                BLUE ();
                GEN=GEN-1;
            }
        }
        IF (GEN<=180) {
            IF (GEN<0) {
                GEN=0-GEN;
            }

            GEN = GEN/1.8;
            FOR (GEN; GEN>0; ) {
                BLUE ();
                GEN=GEN-1;
                YELLOW ();
                GEN=GEN-1;
                ORANGE ();
                GEN=GEN-1;
                RED ();
                GEN=GEN-1;
            }
        }
        GENK++;
    }
}

VOID RDS ()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    IF (SD.BEGIN(4)){
        MYFILE = SD.OPEN("GPS.TXT");
        IF (MYFILE.AVAILABLE()) {
            FOR(I;I<=K;){
                L[I]=MYFILE.PARSEFLOAT();
                SERIAL.PRINTLN(L[I],6);
                I++;
            }
        }
    }
    MYFILE.CLOSE();
    DELAY(500);
}

VOID RGPS()
{
    SERIAL.PRINT("LAT: ");SERIAL.PRINTLN(LATITUDE,6);
    SERIAL.PRINT("LON: ");SERIAL.PRINTLN(LONGITUDE,6);
}

VOID COMP(){
    INT HEADING = GETHEADING();
    IF(HEADING==0){
        HEADING=360;
    }
    IF(P==0){
        INT HEADING = GETHEADING();
        C2 = HEADING;
        C1 = C2;
        DELAY(100);
        C = C2-C1;
        P++;
    }
    C2=HEADING;
    C = C2-C1;
}

FLOAT GETMIN(FLOAT* ARRAY, INT SIZE)
{
    FLOAT MINIMUM = ARRAY[0];
    FOR (INT I = 0; I < SIZE; I++)
    {
        IF (ARRAY[I] < MINIMUM) MINIMUM = ARRAY[I];
    }
    RETURN MINIMUM;
}

VOID COMPUTE(){
    // SERIAL.PRINTLN("CHECK COMPUTE");

    INT JO=1;
    FOR(JO;JO<=K/2;){
        DELXARRAY[JO] = RADIANS(L[(2*JO)-1]-LATITUDE);
        DELYARRAY[JO] = RADIANS(L[2*JO]-LONGITUDE);
        A[JO] = SQUARE(SIN(DELXARRAY[JO]/2)) + COS(RADIANS(LATITUDE)) * COS(RADIANS
(LATITUDE)) * SQUARE(SIN(DELYARRAY[JO]/2));
        C[JO] = 2 * ASIN(SQRT(A[JO]));
        DISTANCE[JO-1] = RADIUS * C[JO];
        JO++;
    }

    FLOAT X = GETMIN(DISTANCE,3);

    IF(X==DISTANCE[0])
    {
        SELECT1();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        IF(X==DISTANCE[1])
        {
            SELECT2();
        }

        IF(X==DISTANCE[2])
        {
            SELECT3();
        }

        // SERIAL.PRINTLN("IT,S OK.");
        // SERIAL.PRINTLN(" ");

    }

VOID SELECT1(){
    NUM = (ATAN(DELYARRAY[1]/DELXARRAY[1]))*(180/3.14159265);
    COMPG2();
    NUMB();
    IF(SELECT==0){
        SWINGSELECT();
    }
    NUMB();
    LOCK();
}

VOID SELECT2(){
    NUM = (ATAN(DELYARRAY[2]/DELXARRAY[2]))*(180/3.14159265);
    COMPG2();
    NUMB();
    IF(SELECT==0){
        SWINGSELECT();
    }
    NUMB();
    LOCK();
}

VOID SELECT3(){
    NUM = (ATAN(DELYARRAY[3]/DELXARRAY[3]))*(180/3.14159265);

    COMPG2();
    NUMB();
    IF(SELECT==0){
        SWINGSELECT();
    }
    NUMB();
    LOCK();
}

VOID COMPG2(){
    IF(DELXARRAY[1]<=0&&DELYARRAY[1]<=0){
        G2 = 180+NUM;
    }
    IF(DELXARRAY[1]>=0&&DELYARRAY[1]<=0){
        G2 = 360+NUM;
    }
    IF(DELXARRAY[1]>=0&&DELYARRAY[1]>=0){
        G2 = NUM;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
IF (DELXARRAY [1] <=0&&DELYARRAY [1] >=0) {
  G2 = 180+NUM;
}
}

```

```

VOID NUMB () {
  IF (CHECK==0) {
    G=G3=G2-G1;
    CHECK++;
  }
  ELSE IF (CHECK!=0) {
    G=G2-G3;
  }
}

```

```

VOID SWINGSELECT () {

```

```

  SW=G-C;

```

```

  SERIAL.PRINTLN ("SWING SELECT BASE ANGLE");

```

```

  SERIAL.PRINTLN (SW);

```

```

    IF (SW>5) {
      SW=SW/1.8;
      FOR (SW;SW>=5;) {
        BLUE ();
        SW=SW-1;
        YELLOW ();
        SW=SW-1;
        ORANGE ();
        SW=SW-1;
        RED ();
        SW=SW-1;
      }
      IF (SW<-5) {
        SW=SW/1.8;
        FOR (SW;SW<=-5;) {
          RED ();
          SW=SW+1;
          ORANGE ();
          SW=SW+1;
          YELLOW ();
          SW=SW+1;
          BLUE ();
          SW=SW+1;
        }
      }
      IF (-5<SW<5) {
        SERIAL.PRINTLN ("STOP");
      }
    }

```

```

  G3=G2;

```

```

  C1=C2;

```

```

  SELECT++;

```

```

}

```

```

VOID LOCK () {

```

```

  SW=G-C;

```

```

  SERIAL.PRINTLN ("SWING LOCK ANGLE");

```

```

  SERIAL.PRINTLN (SW);

```

```

  IF (SW>5) {
    SW=SW/1.8;
    FOR (SW;SW>=5;) {
      RED ();
    }
  }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SW=SW-1;
ORANGE();
SW=SW-1;
YELLOW();
SW=SW-1;
BLUE();
SW=SW-1;
}
}
IF(SW<-5){
    SW=SW/1.8;
    FOR(SW;SW<=-5;){
        BLUE();
        SW=SW+1;
        YELLOW();
        SW=SW+1;
        ORANGE();
        SW=SW+1;
        RED();
        SW=SW+1;
    }
}

IF(-5<SW<5){
    SERIAL.PRINTLN("STOP");
}

G3=G2;
C1=C2;
}

VOID LOOP(){
    WHILE(GPSSERIAL.AVAILABLE()){ // CHECK FOR GPS DATA
        IF(SHIELD.ENCODE(GPSSERIAL.READ())){ // ENCODE GPS DATA
            SHIELD.F_GET_POSITION(&LATITUDE, &LONGITUDE);
            RGPS();
            COMP();
            COMPUTE();
            DELAY(1000);
        }
    }
}

VOID SETUPHMC5883L(){
    //SETUP THE HMC5883L, AND CHECK FOR ERRORS
    ERROR = COMPASS.SETSCALE(1.3); //SET THE SCALE OF THE COMPASS.
    IF(ERROR != 0) SERIAL.PRINTLN(COMPASS.GETERRORTTEXT(ERROR)); //CHECK IF THERE IS AN
    ERROR, AND PRINT IF SO
    ERROR = COMPASS.SETMEASUREMENTMODE(MEASUREMENT_CONTINUOUS); // SET THE MEASUREMENT
    MODE TO CONTINUOUS
    IF(ERROR != 0) SERIAL.PRINTLN(COMPASS.GETERRORTTEXT(ERROR)); //CHECK IF THERE IS AN
    ERROR, AND PRINT IF SO
}

FLOAT GETHEADING(){
    //GET THE READING FROM THE HMC5883L AND CALCULATE THE HEADING
    MAGNETOMETERSCALED SCALED = COMPASS.READSCALEDAXIS(); //SCALED VALUES FROM COMPASS.
    MAGNETOMETERRAW RAW = COMPASS.READRAWAXIS();
    INT MILLIGAUSS_ONTHE XAXIS = SCALED.XAXIS;// (OR YAXIS, OR ZAXIS)
    FLOAT HEADING = ATAN2(SCALED.YAXIS, SCALED.XAXIS);
    // CORRECT FOR WHEN SIGNS ARE REVERSED.
    IF(HEADING < 0) HEADING += 2*PI;
    IF(HEADING > 2*PI) HEADING -= 2*PI;
    RETURN HEADING * RAD_TO_DEG; //RADIANS TO DEGREES
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้