

หุ่นยนต์ตรวจตราภาคพื้นดิน  
Unman Ground Surveillance System Vehicle



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2559

# หุ่นยนต์ตรวจตราภาคพื้นดิน

Unman Ground Surveillance System Vehicle

โดย

ฉัตริน หุุ่มเรืองวงศ์

ภาณุวัชร อติสรณกุล

อาจารย์ที่ปรึกษา

ผศ.พลศาสตร์ เลิศประเสริฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2559

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์  
คณะ วิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เรื่อง หุ่นยนต์ตรวจตราภาคพื้นดิน  
Unman Ground Surveillance System Vehicle  
ผู้จัดทำ นายฉัตริน หุ่มเรืองวงษ์ รหัสประจำตัว 56010232  
นายภาณุวัชร อติสรณกุล รหัสประจำตัว 56010936

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ผศ.พลศาสตร์ เลิศประเสริฐ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาานิพนธ์ หุ่นยนต์ตรวจตราภาคพื้นดิน  
นักศึกษา นายฉัตริน หุ่มรุ่งวงศ์ รหัสประจำตัว 56010232  
นายภาณุวัชร อติสรณกุล รหัสประจำตัว 56010936  
ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์  
ปีการศึกษา 2559  
อาจารย์ที่ปรึกษา ผศ.พลศาสตร์ เลิศประเสริฐ

### บทคัดย่อ

โครงการฉบับนี้อธิบายเกี่ยวกับการออกแบบและการสร้างหุ่นยนต์ต้นแบบสำหรับตรวจตราภาคพื้นดินเพื่อใช้สำรวจในพื้นที่ระยะไกลไร้คนขับและสามารถส่งภาพพื้นที่มาสู่ผู้ควบคุมได้อีกทั้งยังสามารถระบุตำแหน่งของหุ่นยนต์ได้ทำให้ง่ายต่อการควบคุมโดยใช้ไมโครคอนโทรลเลอร์ Arduino รับสัญญาณ Analog จาก Joystick ในการควบคุมมอเตอร์กระแสตรงให้เคลื่อนที่ตามทิศทางที่ต้องการโดยใช้วงจร H-Bridge ในการขับเคลื่อนทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแสตรงและสามารถปรับความเร็วด้วยPWM (Pulse Width Modulation) ซึ่งรับส่งข้อมูลระหว่างฝั่งรับและฝั่งส่งผ่านทางโมดูลสื่อสารไร้สาย โดยจากการทดลองพบว่าตัวหุ่นยนต์ต้นแบบสามารถบังคับได้ระยะทางประมาณ 400 เมตร และสามารถรับภาระได้หนักประมาณ 30 กิโลกรัมโดยที่ความเร็วเปลี่ยนแปลงไปเล็กน้อย

Thesis Title	Unman Ground Surveillance System Vehicle	
Student	Mr.Chattarin Roomreungwong	Student ID 56010232
	Mr. Panuwat Adisaranakul	Student ID 56010936
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2016	
Thesis Advisor	Associate Professor Polsart Lertprasert	

## ABSTRACT

This project describes a design and construction of the prototype of Unman Ground Surveillance System Vehicle to reaching long distance area without human in it. And transmit a photo to the controller. Other from this it be able to get the direction of it to easy for control. Controlled by a microcontroller Arduino that receive analog signal from joystick to controls the direction of the DC motor. By using H-Bridge circuit to direct the motion of the DC motor. And speed of the vehicle with PWM (Pulse Width Modulation). Which receiving and transmitting the data by wiress module. The results showed that the prototype robot can only operate within a distance of about 400 meters and It can carry a load of about 30 kilograms with a slight change in speed.

## กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงสมบูรณ์ได้ เนื่องจากการสนับสนุนจากหลายฝ่าย ทั้ง อาจารย์ที่ปรึกษา ผศ.พลศาสตร์ เลิศประเสริฐ ที่ให้คำแนะนำ ให้การสนับสนุนด้านอุปกรณ์และสถานที่ในการทำงาน นอกจากนี้ยังได้รับการสนับสนุนจากอาจารย์ทุกท่านในภาควิชาอิเล็กทรอนิกส์ เพื่อนักศึกษาชั้นปีสี่และพี่ปริญญาโท รวมไปถึงคุณพ่อ คุณแม่ ที่ให้การสนับสนุนเงินทุนในการทำโครงการขึ้นนี้ จึงใคร่ขอขอบพระคุณผู้มีอุปการคุณทุกท่านมา ณ ที่นี้



ฉัตริน หุ่มเรืองวงษ์  
ภาณูวัชร อติสรณกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

เรื่อง	หน้า
บทคัดย่อ.....	I
Abstract.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ขอบเขตของการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 มอเตอร์ไฟฟ้า.....	3
2.2 มอเตอร์ไฟฟ้ากระแสตรง.....	3
2.2.1 โครงสร้างมอเตอร์.....	3
2.3 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง.....	5
2.4 การควบคุมมอเตอร์ไฟฟ้ากระแสตรง.....	6
2.5 ไมโครคอนโทรลเลอร์ arduino.....	7
2.5.1 การปรับความเร็วโดยใช้ PWM.....	9
2.6 การสื่อสาร SPI (Serial Peripheral Interface).....	10
2.7 โมดูลสื่อสารไร้สาย NRF24L01 ความถี่ 2.4GHz.....	12

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบ	13
3.1 ส่วนส่งข้อมูล (Remote).....	14
3.2 ส่วนรับข้อมูล (ตัวรถ).....	15
3.3 ส่วนขับเคลื่อน.....	15
3.3.1 ไฟเลี้ยงหรือแบตเตอรี่.....	15
3.3.2 วงจรขับเคลื่อน.....	16
3.4 ตัวถังและระบบขับเคลื่อน.....	17
3.5 ส่วนของชุดรับและส่งสัญญาณภาพ.....	17
3.6 โครงสร้างและการประกอบ.....	19
บทที่ 4 การทดลองและผลการทดลอง	20
4.1 ช่วงที่ใช้ควบคุม arduino จาก joystick.....	21
4.2 ค่ากระแสที่ใช้ในการขับเคลื่อน.....	21
4.3 กราฟแสดงการควบคุม input และ output.....	22
4.4 การทดลองความเร็ว.....	28
4.5 การทดลองภาระที่หุ่นรับได้ขณะขับเคลื่อน.....	29
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	30
5.1 สรุปผลการวิจัย.....	30
5.2 ข้อเสนอแนะ.....	30
เอกสารอ้างอิง.....	31
ภาคผนวก.....	32

## สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงค่าช่วงที่ใช้ควบคุม arduino จาก joystick.....	20
4.2 แสดงค่ากระแสที่ใช้ในการขับเคลื่อนมอเตอร์.....	21
4.3 แสดงค่าความเร็วที่วัดได้.....	28
4.4 แสดงค่าระยะเวลาการบังคับ.....	28
4.5 แสดงภาวะที่ตัวหุ่นได้รับ.....	29



## สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบภายในมอเตอร์ไฟฟ้ากระแสตรง.....	4
2.2 การทำงานของมอเตอร์ไฟฟ้ากระแสตรง.....	5
2.3 การทำงานของวงจร H-Bridge สำหรับควบคุมการหมุนของ DC Motor.....	6
2.4 ไมโครคอนโทรลเลอร์ Arduino.....	7
2.5 Layout & Pin out Arduino Board (Model: Arduino UNO R3).....	8
2.6 ไมโครคอนโทรลเลอร์ Arduino MEGA.....	9
2.7 แสดงการเปลี่ยนแปลง duty cycle ของ PWM.....	10
2.8 การเชื่อมต่อการสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master – Slave.....	11
2.9 โมดูลสื่อสารไร้สาย NRF24L01.....	12
3.1 diagram แสดงหลักการทำงาน.....	13
3.2 joystick ที่ใช้ในการควบคุม.....	14
3.3 Arduino UNO R3 และ Module nrf24l01.....	14
3.4 Arduino MEGA 2560 และ Module nrf24l01.....	15
3.5 มอเตอร์ที่ใช้.....	15
3.6 แบตเตอรี่ที่ใช้.....	16
3.7 บอร์ดขับมอเตอร์ดีซี SE-HB-100 แบบ H-Bridge 80A.....	16
3.8 โครงหุ่นเมื่อประกอบกับมอเตอร์.....	17
3.9 จอภาพ ZULEX TM707 และชุดรับสัญญาณภาพ.....	17
3.10 กล้อง CAMTEC AT818 และชุดส่งสัญญาณภาพ.....	18
3.11 ชุดรับและส่งสัญญาณภาพเมื่อประกอบเรียบร้อยแล้ว.....	18
3.12 ตัวถังและระบบขับเคลื่อนของหุ่นยนต์เมื่อประกอบเสร็จแล้ว.....	19
4.1 สัญญาณ input ขณะเดินหน้า.....	22
4.2 สัญญาณ input ขณะถอยหลัง.....	23
4.3 สัญญาณ input ขณะเลี้ยวซ้าย.....	23
4.4 สัญญาณ input ขณะเลี้ยวขวา.....	24
4.5 สัญญาณ output ขณะไม่ได้ทำการควบคุม.....	25

4.6 สัญญาณ output ขณะเดินหน้า.....	26
4.7 สัญญาณ output ขณะถอยหลัง.....	26
4.8 สัญญาณ output ขณะเลี้ยวซ้าย.....	27
4.9 สัญญาณ output ขณะเลี้ยวขวา.....	27



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันมีการพัฒนาเทคโนโลยีและประดิษฐ์สิ่งอำนวยความสะดวกให้สามารถขับเคลื่อนโดยปราศจากคนขับ นำมาใช้งานในส่วนต่างๆหลายแขนง เช่น หุ่นยนต์กู้ภัย รถสำรวจทางทหาร เป็นต้น ส่วนมากมีจุดประสงค์ในการนำมาใช้งานเพื่อลดความเสี่ยงในการใช้มนุษย์ลงพื้นที่ที่เสี่ยงอันตราย ซึ่งจะสามารถลดการบาดเจ็บและสูญเสียชีวิต โดยสิ่งประดิษฐ์นั้นสามารถควบคุมด้วยมนุษย์ผ่านเครื่องมือต่างๆ อาทิเช่น รีโมทคอนโทรล คอมพิวเตอร์ ฯลฯ

### 1.2 ความมุ่งหมายและวัตถุประสงค์

1. เพื่อสร้างหุ่นยนต์ที่สามารถขับเคลื่อนได้ในพื้นที่ที่ยากต่อการเข้าถึง
2. ศึกษากระบวนการรับ-ส่งข้อมูลผ่านโมดูลการสื่อสารไร้สาย
3. ศึกษาการประยุกต์ใช้งานบอร์ด Arduino กับระบบขับเคลื่อนมอเตอร์ โดยใช้ Joystick เป็นตัวควบคุม

### 1.3 สมมติฐานของการศึกษา

1. หุ่นยนต์สามารถเคลื่อนที่ได้ตามทิศทางที่ต้องการอย่างอิสระ
2. หุ่นยนต์สามารถรับน้ำหนักได้ประมาณ 30 kg
3. สามารถส่งผ่านข้อมูลไร้สายได้อย่างรวดเร็วและแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 ขอบเขตของการศึกษา

1. ประดิษฐ์หุ่นยนต์ที่สามารถเคลื่อนที่ได้ตามทิศทางที่ต้องการอย่างอิสระ
2. ประดิษฐ์ชุดควบคุมการรับ-ส่งข้อมูลแบบไร้สาย

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ผู้จัดทำหวังว่าในการทำโครงงานชิ้นนี้ทางคณะผู้จัดทำจะได้ความรู้และประสบการณ์ต่างๆทั้งที่เกี่ยวกับภาควิชาอิเล็กทรอนิกส์ตรงๆและนอกเหนือจากภาควิชา ซึ่งสามารถนำไปประยุกต์ใช้งานต่างๆได้อีกมากมาย และเพื่อให้ผู้ที่สนใจนำไปพัฒนาต่อยอดให้มีประสิทธิภาพสูงยิ่งขึ้นไปได้



## บทที่ 2

# หลักการและทฤษฎี

### 2.1 มอเตอร์ไฟฟ้า

มอเตอร์คือเครื่องกลไฟฟ้า (Electromechanical Energy) ที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้า (Electric Energy) ให้เป็นพลังงานกล (Mechanical Energy) เป็นอุปกรณ์ที่นิยมใช้กันอย่างแพร่หลายในโรงงานต่างๆ เป็นอุปกรณ์ที่ใช้ควบคุมเครื่องจักรกลต่างๆในงานอุตสาหกรรม มอเตอร์มีหลายแบบให้เลือกใช้ให้เหมาะสมกับงาน ดังนั้นต้องทราบถึงชนิดของมอเตอร์ไฟฟ้า ตลอดจนคุณสมบัติในการใช้งานของแต่ละชนิดเพื่อให้เกิดประสิทธิภาพสูงสุดในการใช้งานมอเตอร์นั้นๆ

### 2.2 มอเตอร์ไฟฟ้ากระแสตรง (Direct current motor)

#### 2.2.1 โครงสร้างมอเตอร์

##### 1.) โรเตอร์ (Rotor)

ในมอเตอร์ไฟฟ้า ส่วนที่เคลื่อนที่คือโรเตอร์ ซึ่งจะหมุนเพลาเพื่อจ่ายพลังงานกล โรเตอร์มักจะมีขดลวดตัวนำพันอยู่โดยรอบ ซึ่งเมื่อมีกระแสไหลผ่าน จะเกิดอำนาจแม่เหล็กที่จะไปทำปฏิกิริยากับสนามแม่เหล็กถาวรของสเตเตอร์ ขับเพลลาให้หมุนได้ อย่างไรก็ตามโรเตอร์บางตัวจะเป็นแม่เหล็กถาวรและสเตเตอร์จะมีขดลวดตัวนำสลับที่กัน

##### 2.) สเตเตอร์ (Stator)

ส่วนที่ไม่เคลื่อนที่คือ สเตเตอร์ มักจะมีขดลวด หรือ แม่เหล็กถาวร

##### 3.) ช่องว่างอากาศ (Air gap)

ระหว่างโรเตอร์และสเตเตอร์จะเป็นช่องว่างอากาศ ซึ่งจะต้องมีขนาดเล็กที่สุดเท่าที่จะเป็นไปได้ ช่องว่างขนาดใหญ่จะมีผลกระทบทางลบอย่างมากต่อประสิทธิภาพการทำงานของมอเตอร์ไฟฟ้า

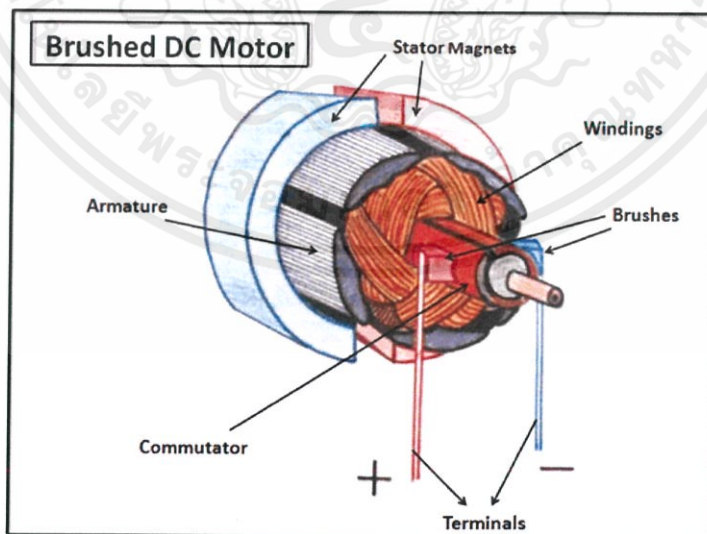
##### 4.) ขดลวด (Winding)

ขดลวดจะพันโดยรอบเป็นคอยล์ ปกติจะพันรอบแกนแม่เหล็กอ่อนที่เคลือบฉนวน เพื่อให้เป็นขั้วแม่เหล็กเมื่อมีกระแสไฟฟ้าไหลผ่านมอเตอร์ไฟฟ้ามีขั้วสนามแม่เหล็กในสองรูปแบบ ได้แก่แบบขั้วที่เห็นได้ชัดเจนและแบบขั้วที่เห็นได้ไม่ชัดเจน ในขั้วที่ชัดเจน สนามแม่เหล็กของขั้วจะถูกผลิตโดยขดลวดพันรอบ

แกนด้านล่าง ในข้อที่ไม่ชัดเจน หรือเรียกว่าแบบสนามแม่เหล็กกระจาย หรือแบบรอบๆโรเตอร์ ขดลวดจะกระจายอยู่ในช่องบนแกนรอบโรเตอร์ มอเตอร์แบบขั้วแปงมีขดลวดรอบส่วนหนึ่งของขั้วเพื่อหน่วงเฟสของสนามแม่เหล็กของขั้วนั้นให้ช้าลง มอเตอร์บางตัวขดลวดเป็นโลหะหนากว่า เช่นแท่งหรือแผ่นโลหะที่มักจะเป็นทองแดง บางทีก็เป็น อะลูมิเนียม มอเตอร์เหล่านี้โดยปกติจะถูกขับเคลื่อนโดยการเหนี่ยวนำของแม่เหล็กไฟฟ้า

### 5.) ตัวสับเปลี่ยน (commutator)

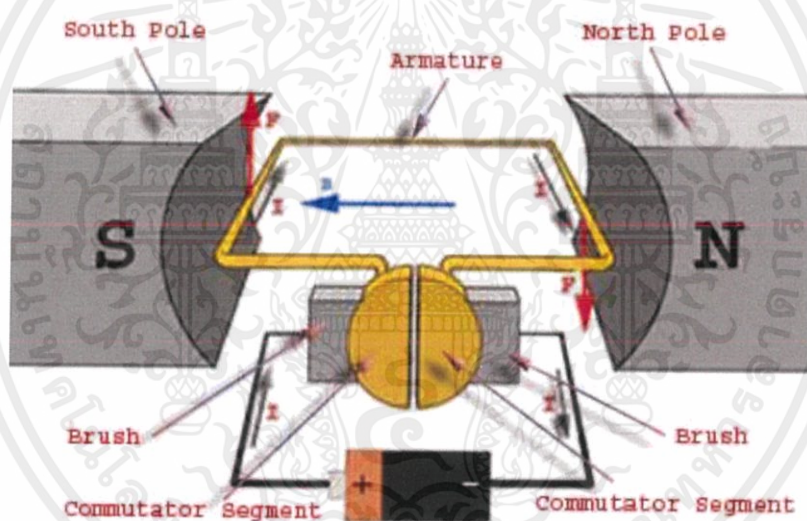
ตัวสับเปลี่ยนเป็นกลไกที่ใช้ในการสลับอินพุทของมอเตอร์ AC และ DC เพื่อให้กระแสที่ไหลในขดลวดในโรเตอร์ไหลทางเดียวตลอดเวลาในระหว่างการหมุน ประกอบด้วยวงแหวนลื่น (slip ring) ชิ้นเล็กๆ แยกจากกันด้วยฉนวน วงแหวนนี้ยังแยกจากเฟลาของมอเตอร์ด้วยฉนวนอีกด้วย วงแหวนแต่ละคู่ที่อยู่ตรงข้ามกันจะเป็นขดลวดหนึ่งชุด กระแสที่จ่ายให้มัดขั้วต้ม หรือที่เรียกว่า armature ของมอเตอร์จะถูกส่งผ่านแปรงถ่าน (brush) สองตัวที่แตะอยู่กับตัวสับเปลี่ยนแต่ละด้านที่กำลังหมุนอยู่ ซึ่งจะทำให้กระแสจากแหล่งจ่ายไฟ AC ที่ไหลกลับทาง ไหลในขดลวดทิศทางเดียวในขณะที่โรเตอร์หมุนจากขั้วหนึ่งไปอีกขั้วหนึ่ง ในกรณีที่ไม่มีกระแสแหล่งจ่ายไม่กลับทางมอเตอร์จะ เบรกหยุดอยู่กับที่ ในแง่ของความก้าวหน้าที่สำคัญในช่วงไม่กี่ทศวรรษที่ผ่านมา อันเนื่องมาจากเทคโนโลยีที่ดีขึ้นในการควบคุมอิเล็กทรอนิกส์ มอเตอร์เหนี่ยวนำที่ควบคุมโดยไมโครโพรเซสเซอร์ และมอเตอร์ที่มีสนามแม่เหล็กถาวร มอเตอร์ที่มีตัวสับเปลี่ยนแบบกลไกไฟฟ้า กำลังถูกแทนที่เพิ่มขึ้นด้วยมอเตอร์เหนี่ยวนำที่ใช้ตัวสับเปลี่ยนภายนอกและมอเตอร์แบบแม่เหล็กถาวร



รูปที่ 2.1 ส่วนประกอบภายในมอเตอร์ไฟฟ้ากระแสตรง

## 2.3 หลักการของมอเตอร์ไฟฟ้ากระแสตรง

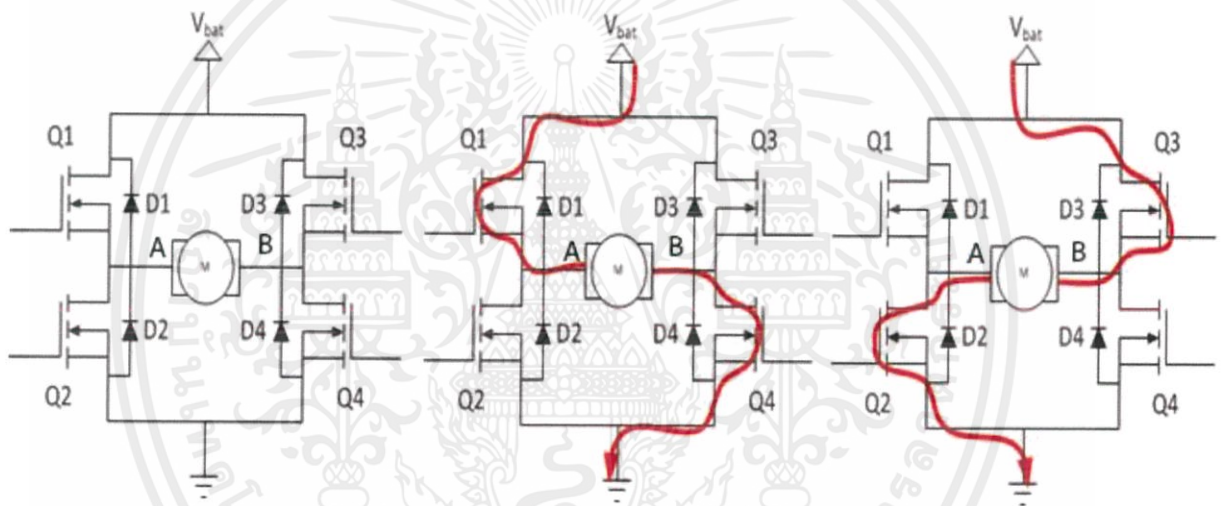
หลักการของมอเตอร์ไฟฟ้ากระแสตรง เมื่อแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่ง จะแปรปรวนผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field Coil) สร้างขั้วเหนือ-ใต้ ขึ้น จะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดกัน ทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวกันจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวของอาร์มาเจอร์ ซึ่งวางแกนเพลลา และแกนเพลลานี้สวมอยู่กับตลับลูกปืนของมอเตอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ ขณะที่ตัวของอาร์มาเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่าตัวหมุน การที่อำนาจสนามแม่เหล็กทั้งสองมีปฏิสัมพันธ์กัน ทำให้ขดลวดอาร์มาเจอร์หรือโรเตอร์หมุนไปนั้น เป็นไปตามกฎมือซ้ายของเฟลมมิ่ง (Fleming's Left Hand Rule)



รูปที่ 2.2 การทำงานของมอเตอร์ไฟฟ้ากระแสตรง

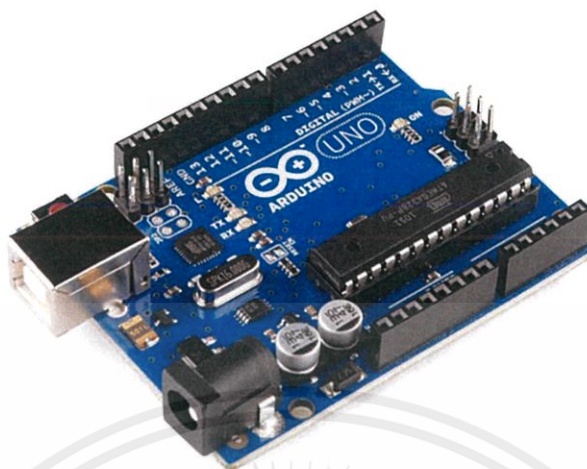
## 2.4 การควบคุมมอเตอร์ไฟฟ้ากระแสตรง

ชุดขับเคลื่อนสี่มอเตอร์ส่วนใหญ่จะเป็นแบบ H-Bridge ซึ่งวงจรประกอบด้วยทรานซิสเตอร์หรือมอเตอร์เฟส โดยทำหน้าที่เป็นสวิตช์เปิดปิด จำนวน 4 ชุด (Q1-Q4) โดยต่อกับ DC Motor ซึ่งสามารถควบคุมการทิศทางการไหลของกระแสได้ เมื่อส่งสัญญาณควบคุมให้ทรานซิสเตอร์ Q1 และ Q4 ทำงาน และปิดการทำงานของทรานซิสเตอร์ Q2 และ Q3 กระแสจะไหลจากจุด A ไปจุด B จึงทำให้มอเตอร์เริ่มหมุน เมื่อส่งสัญญาณควบคุมให้ทรานซิสเตอร์ Q3 และ Q2 ทำงาน และปิดการทำงานของทรานซิสเตอร์ Q1 และ Q4 กระแสจะไหลจากจุด B ไปจุด A ดังรูป (c) เป็นผลให้มอเตอร์หมุนกลับทิศ



รูปที่ 2.3 การทำงานของวงจร H-Bridge สำหรับควบคุมการหมุนของ DC Motor

## 2.5 ไมโครคอนโทรลเลอร์ Arduino

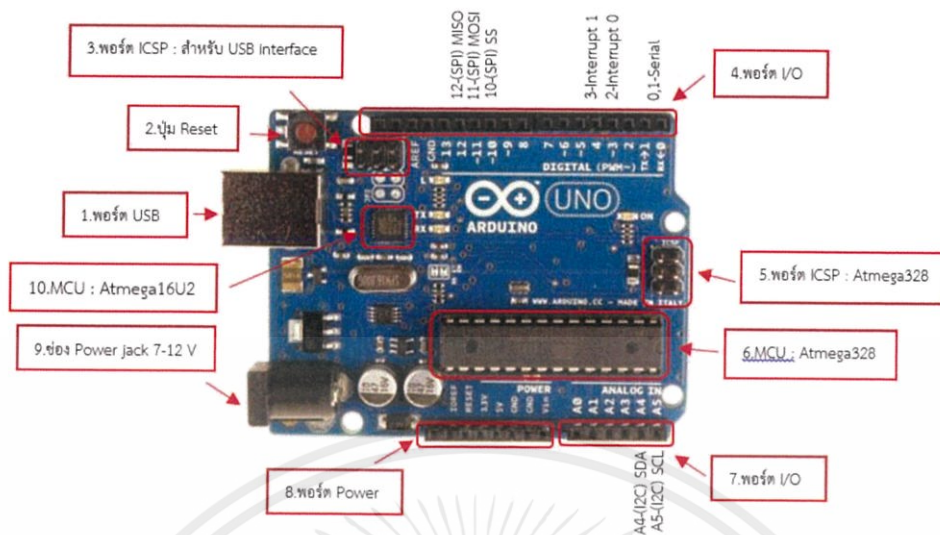


รูปที่ 2.4 ไมโครคอนโทรลเลอร์ Arduino

Arduino อ่านว่า (อา-ดู-อี-โน้ หรือ อาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

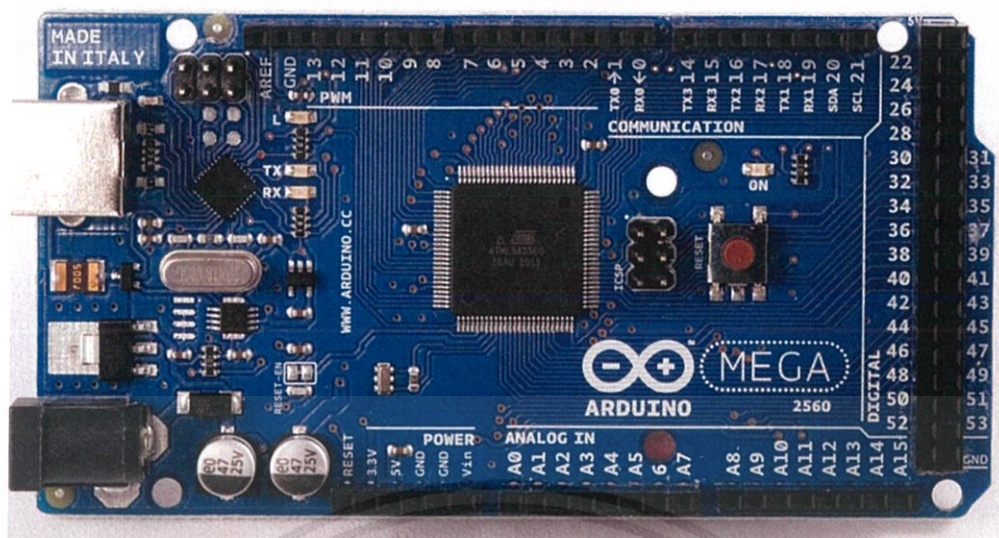
## Layout & Pin out Arduino Board (Model: Arduino UNO R3)



รูปที่ 2.5 Layout & Pin out Arduino Board (Model: Arduino UNO R3)

- 1.USBPort: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
- 2.Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
- 3.ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
4. I/OPort:Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
- 5.ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
7. I/OPort: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
- 8.Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขา ไฟเลี้ยง +3.3 V, +5V, GND, Vin
9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่านAtmega16U2

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



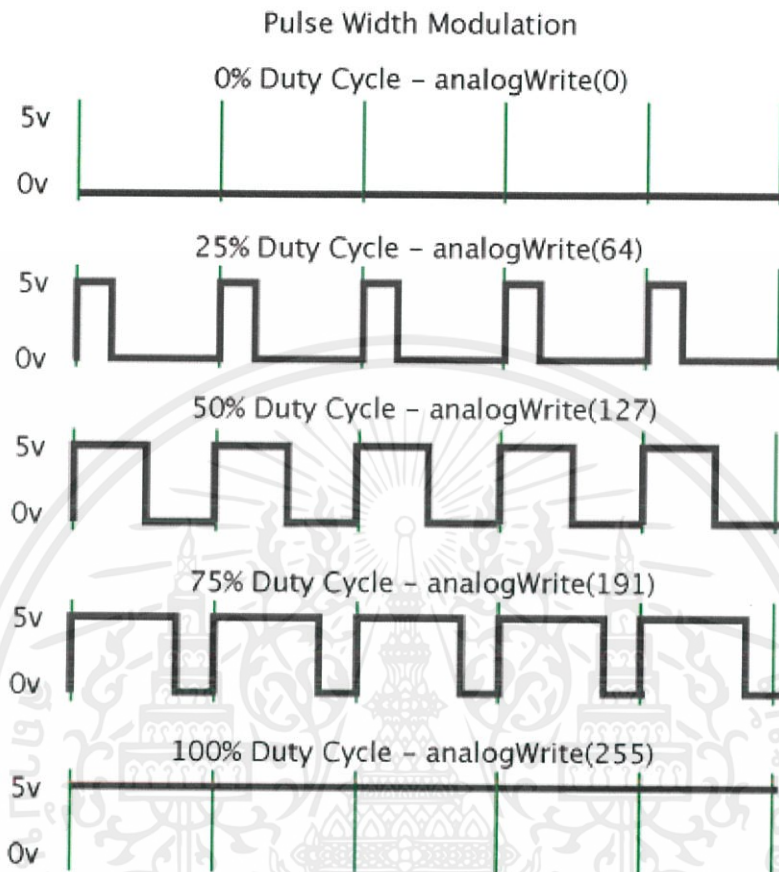
รูปที่ 2.6 ไมโครคอนโทรลเลอร์ Arduino MEGA

Arduino Mega 2560 R3 เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน

### 2.5.1 การปรับความเร็วโดยใช้ PWM

PWM หมายถึง Pulse Width Modulation เป็นเทคนิคที่ Arduino ใช้ในการควบคุมวงจรและ เขียนค่าแบบอนาล็อก (Analog) ด้วยพอร์ตดิจิทัล (Digital) คือ โดยปกติแล้ว พอร์ตดิจิทัล จะสามารถมีได้แค่ 2 สถานะ คือ HIGH (5 โวลท์) กับ LOW (0 โวลท์)เท่านั้น จึงทำให้สร้างค่าสัญญาณลอจิกได้เพียง เปิดหรือปิด (1 หรือ 0 , มีไฟหรือไม่มีไฟ) แค่นั้น ซึ่งการใช้เทคนิค PWM นั้น จะเป็นการทำให้พอร์ตดิจิทัล สามารถเขียนค่าได้มากกว่า HIGH หรือ LOW โดย ทำให้สามารถเขียนค่าเป็นแบบอนาล็อกได้ (อาจเป็น 0-255 หรือ 0-1023) โดยวิธีการนั้นจะการใช้การปรับสถานะของสัญญาณลอจิก HIGH / LOW สลับกันไปมาด้วยคาบเวลาหนึ่งๆ โดยค่าที่ได้นั้นจะขึ้นอยู่กับ สัดส่วนเวลาของสัญญาณในช่วงเวลาที่มีสถานะเป็น HIGH กับช่วงเวลาที่ เป็น LOW โดย ช่วงเวลาทั้งหมดที่สัญญาณมีสถานะเป็น HIGH นั้นเราจะเรียกว่าเป็น ความกว้าง Pulse (Pulse Width)"

โดยสัญญาณพัลส์ เมื่อเทียบ % ของช่วงเวลาที่เป็น HIGH (หรือก็คือ % ของ Pulse Width) กับ % ของคาบเวลา (Period) ของพัลส์สัญญาณนั้นๆ เราจะเรียกว่า Duty Cycle



รูปที่ 2.7 แสดงการเปลี่ยนแปลง duty cycle ของ PWM

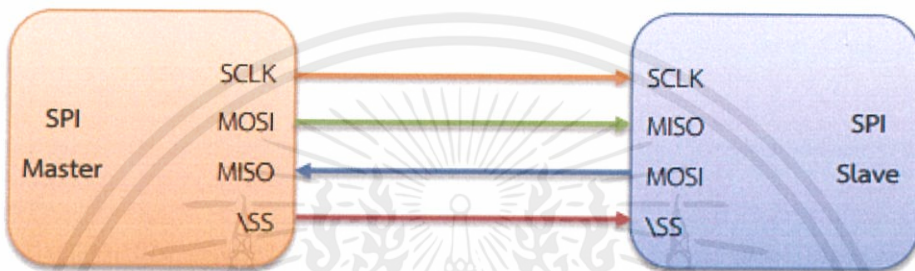
## 2.6 การสื่อสาร SPI (Serial Peripheral Interface)

SPI หรือ Serial Peripheral Interface เป็นวิธีการสื่อสารอนุกรมแบบ Synchronous อีกรูปแบบหนึ่ง ซึ่งทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็น Master ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็น Slave และสามารถส่งข้อมูลในโหมด Full-duplex นั้นหมายความว่า สัญญาณสามารถส่งหากันได้ระหว่าง Master และ Slave ได้อย่างต่อเนื่อง รูปแบบข้อมูลการสื่อสารหรือ Protocol ของแบบ SPI นี้ ไม่ได้มาตรฐานกำหนดตายตัว ว่าข้อมูลที่ส่งหากันต้องอยู่ในรูปแบบหรือ Format แบบไหน เป็นการคิด Protocol การสื่อสารกันเอาเอง หรือดูจาก Datasheet ของอุปกรณ์

ยกตัวอย่างอุปกรณ์ที่ใช้การสื่อสารแบบ SPI ได้แก่

- โมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัล และโมดูลแปลงสัญญาณดิจิทัลเป็นอนาล็อก
- การติดต่อกับหน่วยความจำ EEPROM และ FLASH
- โมดูลนาฬิกาดิจิทัล หรือ Real Time Clock : RTC
- เซ็นเซอร์วัดอุณหภูมิ และความดัน

อุปกรณ์อื่น ๆ เช่น Signal mixer , Potentiometer , LCD controller , USART , CAN controller , USB controller , Amplifier



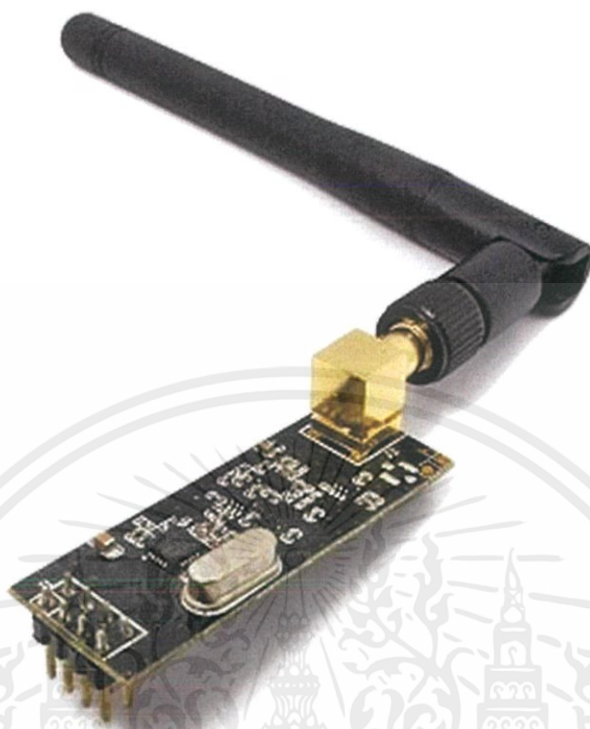
รูปที่ 2.8 การเชื่อมต่อการสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master – Slave

- SCLK (Serial Clock) ใช้ส่งสัญญาณนาฬิกาจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave เพื่อกำหนดจังหวะการรับส่งข้อมูล
- MOSI (Master Out Slave In) ใช้ส่งข้อมูลจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave
- MISO (Master In Slave Out) ใช้รับข้อมูลจากอุปกรณ์ Slave
- SS (Slave Select) หรือ ขา CS (Chip Select) ใช้ส่งสัญญาณ Low ไปยังอุปกรณ์ Slave ที่ต้องการรับส่งข้อมูล

อุปกรณ์ Master ทำหน้าที่เป็นตัวควบคุมการสื่อสารทั้งหมด โดยควบคุมการสื่อสารตามสัญญาณนาฬิกา และสายสัญญาณ SS ตัวมาสเตอร์จะเป็นตัวที่ตัดสินใจเลือก รับ หรือ ส่งข้อมูลไปยังอุปกรณ์ Slave เมื่อเปรียบเทียบระหว่างการสื่อสาร SPI กับ I2C เป็นการสื่อสารอนุกรมแบบ Synchronous เหมาะสำหรับการสื่อสารข้อมูลความเร็วต่ำ เช่น หน่วยความจำ EEPROM หรือ โมดูลนาฬิกาดิจิทัล ข้อดีของการสื่อสารแบบ SPI คือ สามารถสื่อสารแบบ Full Duplex กล่าวคือสามารถรับและส่งข้อมูลได้พร้อมๆ กัน เพราะมีสายสัญญาณรับและส่งข้อมูลโดยเฉพาะ รูปแบบการสื่อสารของ SPI ไม่ต้องกำหนด Address เพื่อระบุอุปกรณ์ที่ต้องการสื่อสารเหมือน I2C เนื่องจากใช้สายสัญญาณ SS เป็นตัวควบคุม จึงมีอัตราการรับส่งข้อมูลสูงกว่า I2C และเหมาะสำหรับการรับส่งข้อมูลแบบต่อเนื่อง หรือ Streaming อย่างไรก็ตาม หากมีอุปกรณ์ Slave หลายตัวดังรูป การสื่อสารแบบ SPI ต้องใช้สายสัญญาณมากกว่า I2C

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 โมดูลสื่อสารไร้สาย NRF24L01 ความถี่ 2.4GHz



รูปที่ 2.9 โมดูลสื่อสารไร้สาย NRF24L01

โมดูล NRF24L01 เป็นโมดูลสื่อสารไร้สาย ที่สามารถเขียนโปรแกรมให้เป็นได้ ทั้งตัวรับและตัวส่ง สามารถใช้กับ Arduino ได้หลาย ๆ ตัวพร้อมกัน มีความเร็ว 2.4GHz จึงสื่อสารได้รวดเร็วและไม่ต้องการเสาอากาศที่ยาว มีขนาดเล็กสะดวกในการต่อใช้งาน สามารถประยุกต์ใช้งานได้หลายอย่างเช่น ใช้เป็นอุปกรณ์ส่งข้อมูลของเซนเซอร์อัตโนมัติสำหรับควบคุม อุณหภูมิ ความชื้น การแจ้งเตือนต่าง ๆ ควบคุมและติดตามหุ่นยนต์ Robot Control and Monitoring ได้ในระยะ 15-500 เมตร โมดูลนี้ใช้ชิพ nRF24L01 ทำงานด้วยความเร็วสูง High-speed SPI interface ใช้พลังงานต่ำรองรับการทำงานร่วมกับ Arduino และมีเสาอากาศมาให้ในตัว

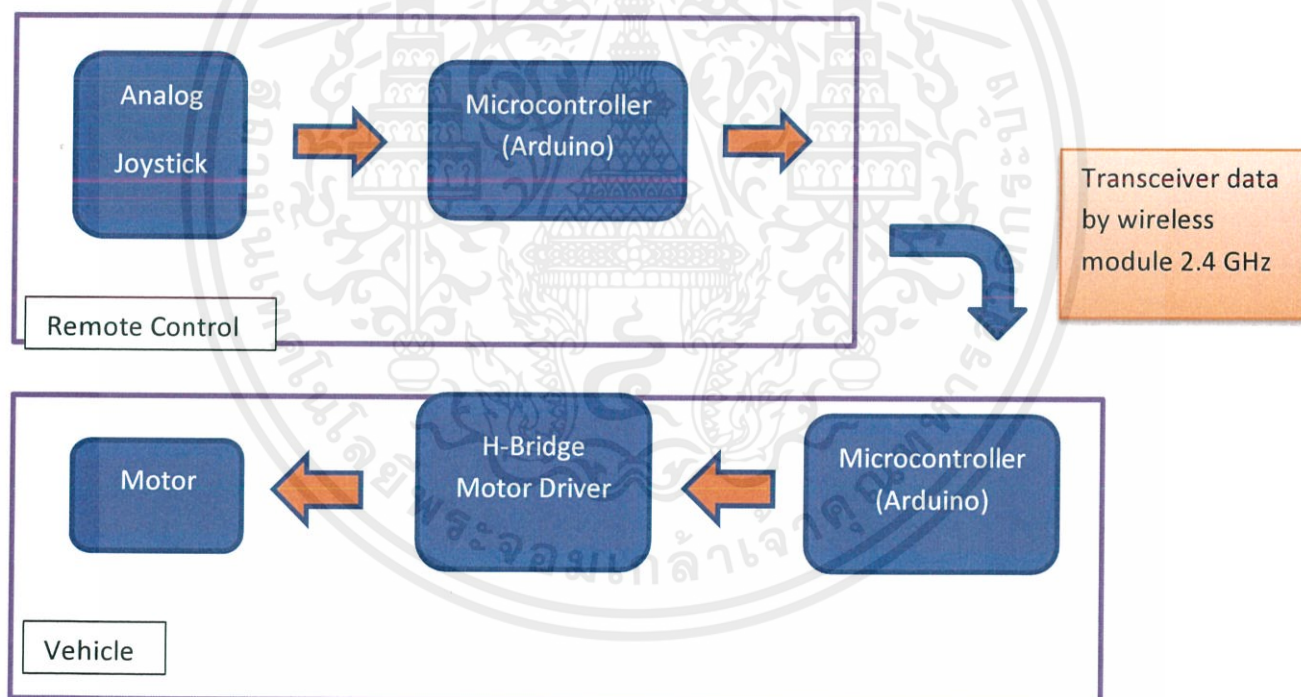
## บทที่ 3

### การออกแบบ

แบ่งเป็นส่วนใหญ่ๆ 3 ส่วนดังนี้

1. ส่วนส่งข้อมูล (รีโมทคอนโทรล)
2. ส่วนรับข้อมูล (ตัวรถ)
3. ส่วนขับเคลื่อน

ซึ่งมีหลักการทำงานคร่าวๆดังรูป 3.1



รูปที่ 3.1 diagram แสดงหลักการทำงาน

จากรูปที่ 3.1 จะเห็นได้ว่าการขับเคลื่อนจะใช้การบังคับโดย joystick ซึ่งจะมีการส่งข้อมูลไปเพื่อกำหนดทิศทางและการหมุนของมอเตอร์ให้ตัวรถเคลื่อนที่ไปตามที่ต้องการ

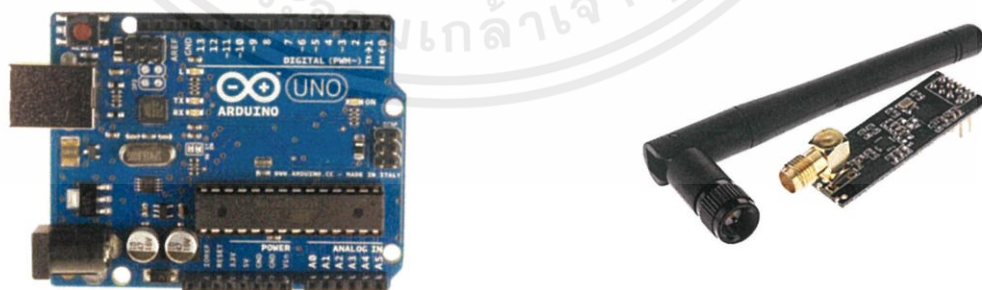
### 3.1 ส่วนส่งข้อมูล (Remote)

ก้านควบคุม (joystick) เป็นอุปกรณ์นำข้อมูลเข้ารูปแบบหนึ่งของคอมพิวเตอร์ มีลักษณะเป็นคันโยกบนฐาน ใช้ควบคุมการเคลื่อนที่ของตัวชี้ หรือ pointer บนจอภาพ



รูปที่ 3.2 joystick ที่ใช้ในการควบคุม

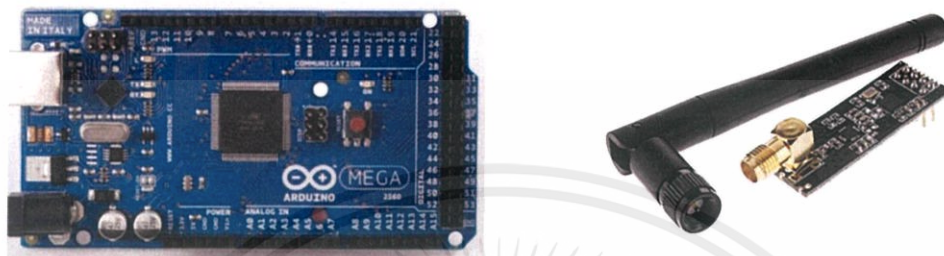
ซึ่งภายในจะเป็นตัวต้านทานปรับค่าได้ เมื่อขยับตัวคันโยก ค่าความต้านทานก็จะเปลี่ยนไป แรงดันที่ตกคร่อมก็จะเปลี่ยนไป ซึ่งแรงดันที่ตกคร่อมตัว joystick นี้ จะส่งไปเป็น อินพุตให้กับไมโครคอนโทรลเลอร์ ซึ่งตัวที่เลือกใช้คือ Arduino UNO R3 จะรับแรงดันที่ตกคร่อมตัว Joystick เป็นอินพุตเพื่อให้ส่งข้อมูลออกไปตามเงื่อนไขที่ได้กำหนดไว้ โดยส่งข้อมูลไร้สายผ่านโมดูล nrf24l01



รูปที่ 3.3 Arduino UNO R3 และ Module nrf24l01

## 3.2 ส่วนรับข้อมูล (ตัวรถ)

ในการรับข้อมูลจากรีโมทคอนโทรล จะต้องใช้ตัวรับที่ใช้คลื่นความถี่เดียวกันกับตัวส่ง ฉะนั้นในส่วนนี้จึงใช้โมดูลตัวเดียวกันกับตัวส่งซึ่งก็คือ nrf24l01 ในการรับข้อมูลแล้วส่งไปที่ไมโครคอนโทรลเลอร์ตัวหลัก ซึ่งได้เลือกใช้ Arduino MEGA 2560 เป็นส่วนควบคุมหลักของตัวรถ



รูปที่ 3.4 Arduino MEGA 2560 และ Module nrf24l01

## 3.3 ส่วนขับเคลื่อน

### 3.3.1 ไฟเลี้ยงหรือแบตเตอรี่

-ในส่วนของการขับเคลื่อนนั้นเลือกใช้มอเตอร์ DC 24 V จำนวน 6 ตัวในการขับเคลื่อนตัวรถ



รูปที่ 3.5 มอเตอร์ที่ใช้

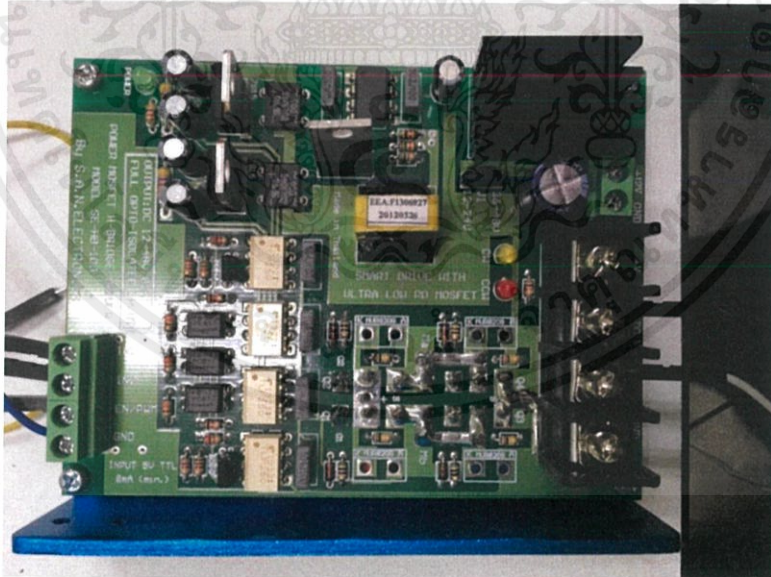
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ใช้แบตเตอรี่ 12V ต่ออนุกรมกัน 2 ลูก เพื่อจ่ายไฟ 24V



### 3.3.2 วงจรขับมอเตอร์

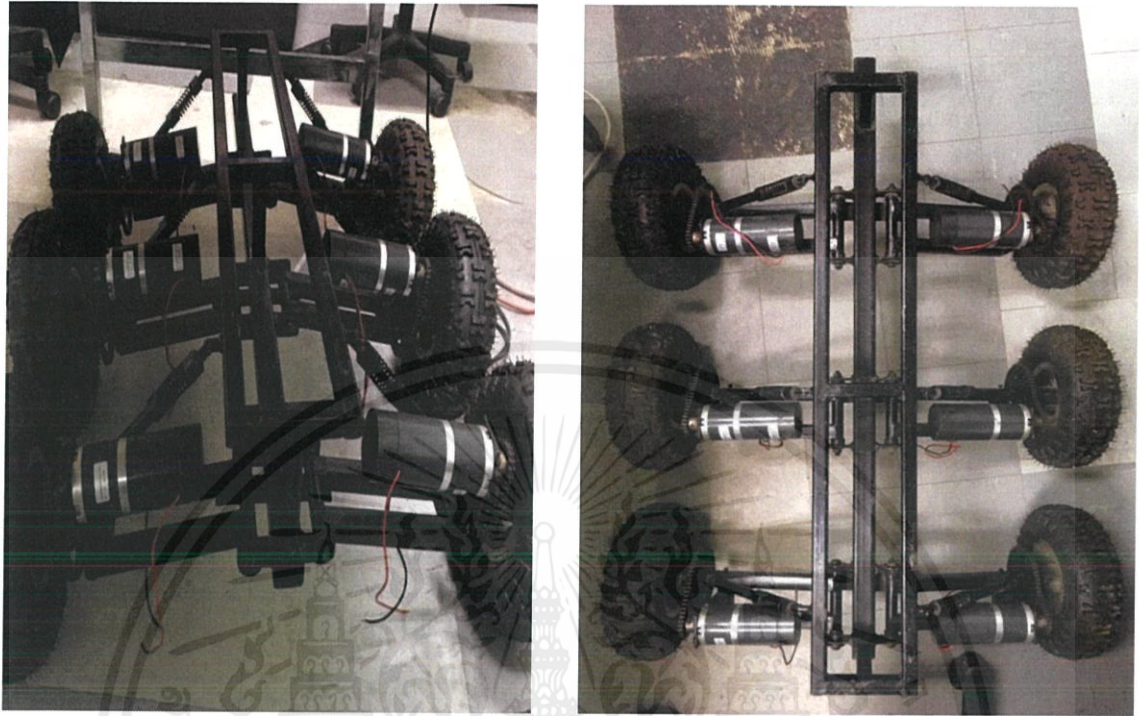
-ใช้บอร์ดสำเร็จรูป SE-HB-100 บอร์ดขับมอเตอร์ดีซี แบบ H-Bridge 80A



รูปที่ 3.7 บอร์ดขับมอเตอร์ดีซี SE-HB-100 แบบ H-Bridge 80A

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ตัวถังและระบบขับเคลื่อนของหุ่นยนต์



รูปที่ 3.8 โครงหุ่นเมื่อประกอบกับมอเตอร์

### 3.5 ส่วนของชุดรับและส่งสัญญาณภาพ



รูปที่ 3.9 จอภาพ ZULEX TM707 และชุดรับสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



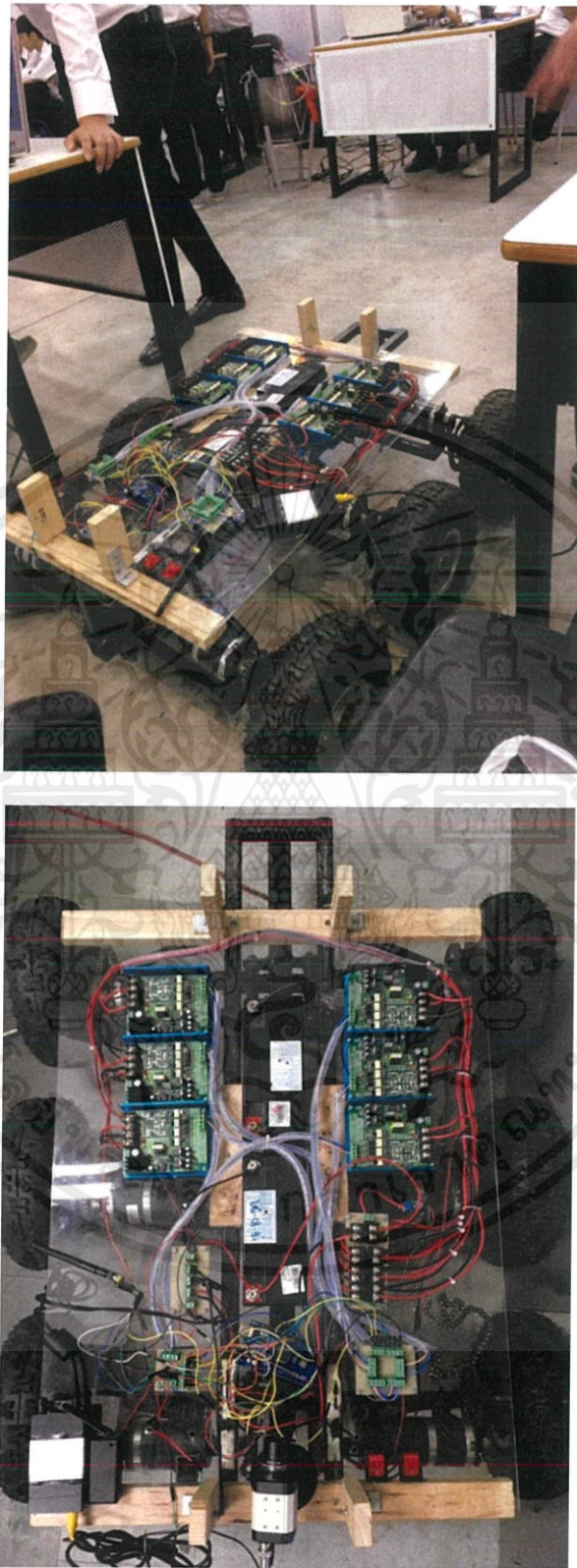
รูปที่ 3.10 กล้อง CAMTEC AT818 และชุดส่งสัญญาณภาพ



รูปที่ 3.11 ชุดรับและส่งสัญญาณภาพเมื่อประกอบเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 โครงสร้างและการประกอบ



รูปที่ 3.12 ตัวถังและระบบขับเคลื่อนของหุ่นยนต์เมื่อประกอบเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 ช่วงที่ใช้ควบคุม arduino จาก joystick

ตารางที่ 4.1 แสดงค่าช่วงที่ใช้ควบคุม arduino จาก joystick

Input	Direction	Volt(V)	ADC จากการคำนวณ	ADC จากการทดลอง
A <sub>0</sub> (แกนX)	ไม่มีการควบคุม	2.60	132.6	133
	เดินหน้า(max)	0.64	32.64	33
	ถอยหลัง(max)	4.66	237.66	241
A <sub>1</sub> (แกนY)	ไม่มีการควบคุม	2.02	103.02	112
	เลี้ยวซ้าย(max)	0.50	25.50	24
	เลี้ยวขวา(max)	4.02	205.02	209

ค่าADCจากการคำนวณ

$$\text{คำนวณได้จาก ADC} = \frac{V \times 8\text{bit}}{V_{\text{max}}}$$

ADC คือ ค่า analog to digital converter ขนาด 8 bit

V คือ ค่าแรงดันที่ส่งมาจาก joystick

8bit คือ ค่า digital(max) ขนาด 8 bit มีค่า 255

V<sub>max</sub> คือ ค่าแรงดันที่ analog pin ของ arduino รับได้สูงสุด

จากตารางจะเห็นว่า

ที่  $A_0$ (แกน X)

เมื่อไม่มีการควบคุม ค่า  $V$  มีค่า 2.60 โวลต์ และเมื่อเริ่มบังคับเดินหน้าค่า  $V$  จะลดลงเรื่อยๆจนถึง 0.64 โวลต์ ซึ่งเป็นค่าสูงสุดที่ใช้ในการสั่งการ arduino ในทางตรงกันข้าม เมื่อเราเริ่มบังคับถอยหลังค่า  $V$  จะเพิ่มขึ้นเรื่อยๆจนถึง 4.66 โวลต์

ที่  $A_1$ (แกน Y)

เมื่อไม่มีการควบคุม ค่า  $V$  มีค่า 2.02 โวลต์ และเมื่อเริ่มบังคับเลี้ยวซ้ายค่า  $V$  จะลดลงเรื่อยๆจนถึง 0.50 โวลต์ ซึ่งเป็นค่าสูงสุดที่ใช้ในการสั่งการ arduino ในทางตรงกันข้าม เมื่อเราเริ่มบังคับเลี้ยวขวาค่า  $V$  จะเพิ่มขึ้นเรื่อยๆจนถึง 4.02 โวลต์

## 4.2 ค่ากระแสที่ใช้ในการขับมอเตอร์

ตารางที่ 4.2 แสดงค่ากระแสที่ใช้ในการขับมอเตอร์

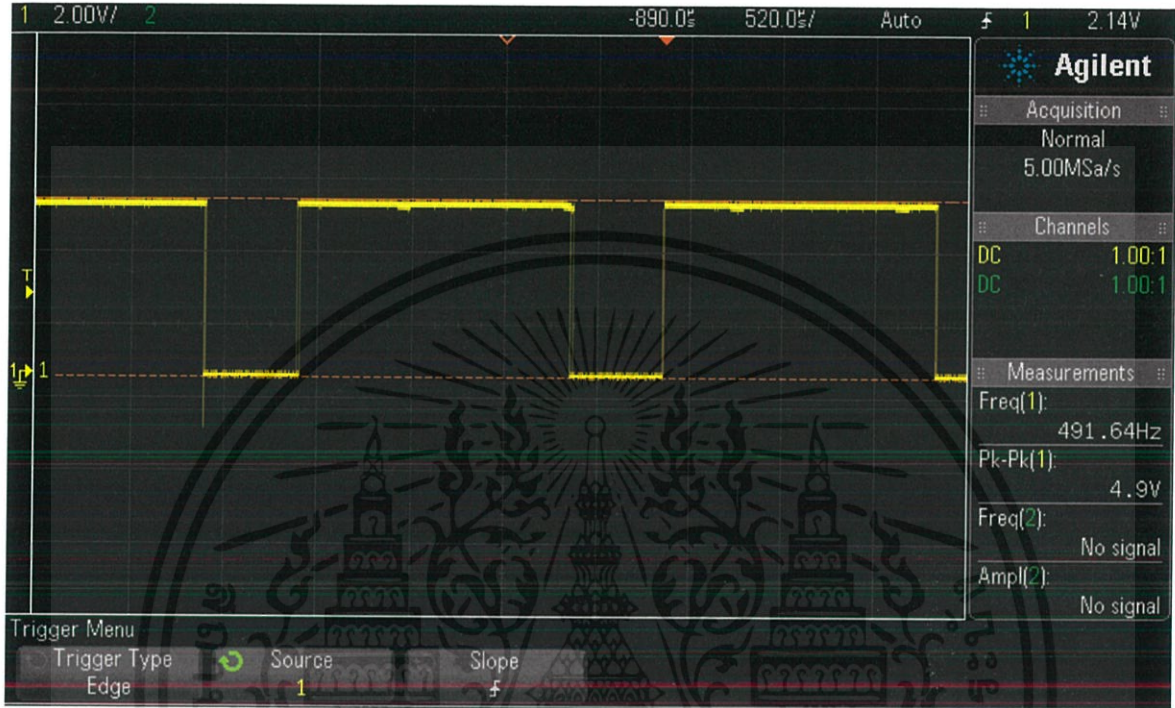
รูปแบบการใช้งาน	ตัวหุ่นเริ่มเคลื่อนที่	ตัวหุ่นมีความเร็วสูงสุด
No load	0.7 A	1.1 A
มีภาระแค่ตัวหุ่นเอง	1 A	3 A

จากตารางจะเห็นว่า ในขณะที่ขับมอเตอร์แบบ no load เมื่อเริ่มเคลื่อนที่ใช้กระแส 0.7 A และเมื่อเร่งความเร็วจนถึงสูงสุดใช้กระแสเพิ่มเป็น 1.1 A (ต่อมอเตอร์ 1 ตัว)

ในขณะที่ขับมอเตอร์แบบใช้งานจริง(มีแค่น้ำหนักตัว) เมื่อเริ่มเคลื่อนที่ใช้กระแส 1 A และเมื่อเร่งความเร็วจนถึงสูงสุดใช้กระแสเพิ่มเป็น 3 A (ต่อมอเตอร์ 1 ตัว)

### 4.3 กราฟแสดงการควบคุม input และ output

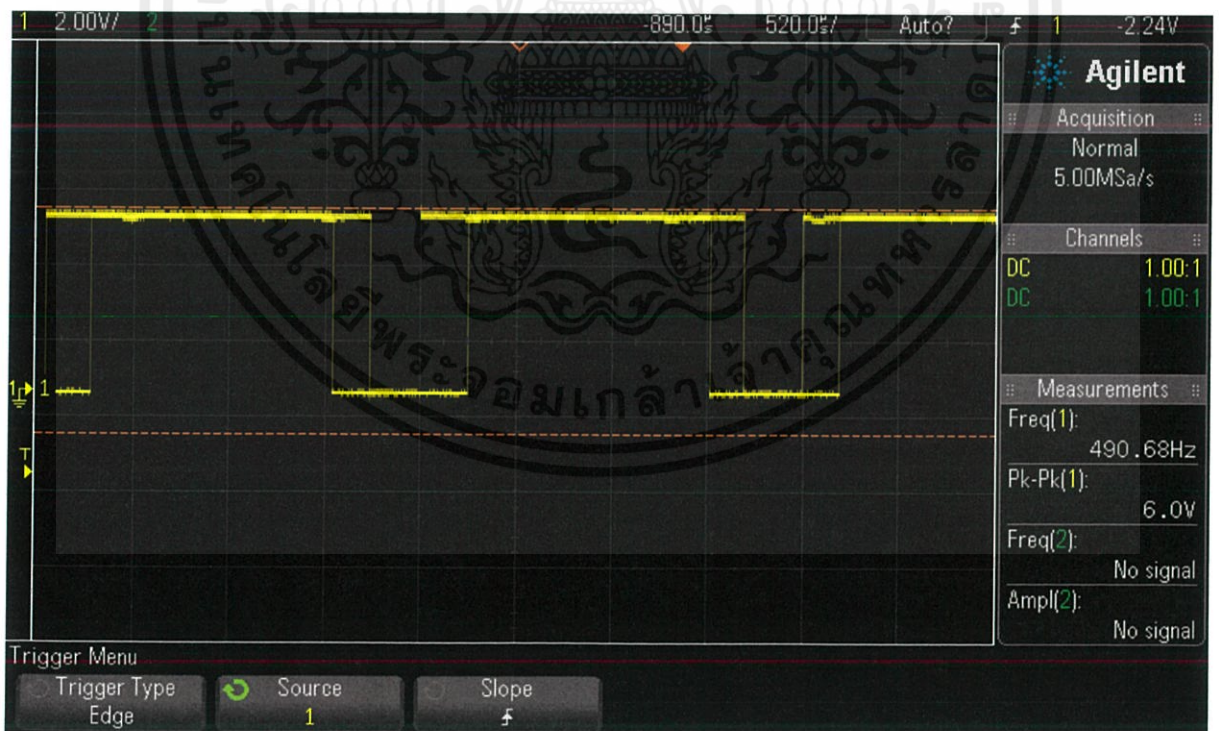
กราฟแสดงผลการทดลอง input



รูปที่ 4.1 สัญญาณ input ขณะเดินหน้า



รูปที่ 4.2 สัญญาณ input ขณะถอยหลัง



รูปที่ 4.3 สัญญาณ input ขณะเลี้ยวซ้าย

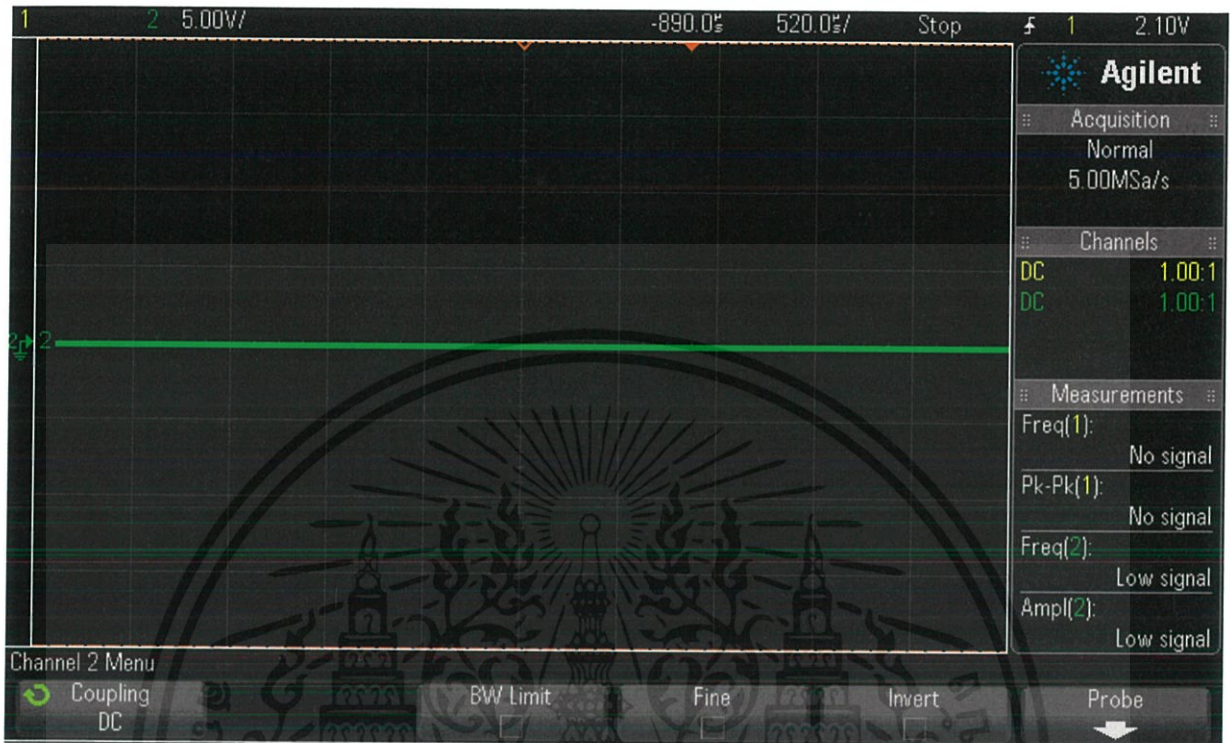
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 สัญญาณ input ขณะเลี้ยวขวา

จากรูปจะเห็นว่าสัญญาณที่ออกจาก arduino มีลักษณะเป็น PWM โดยมีแรงดันประมาณ 5 v และความถี่ประมาณ 490 Hz

กราฟแสดงผลการทดลอง output

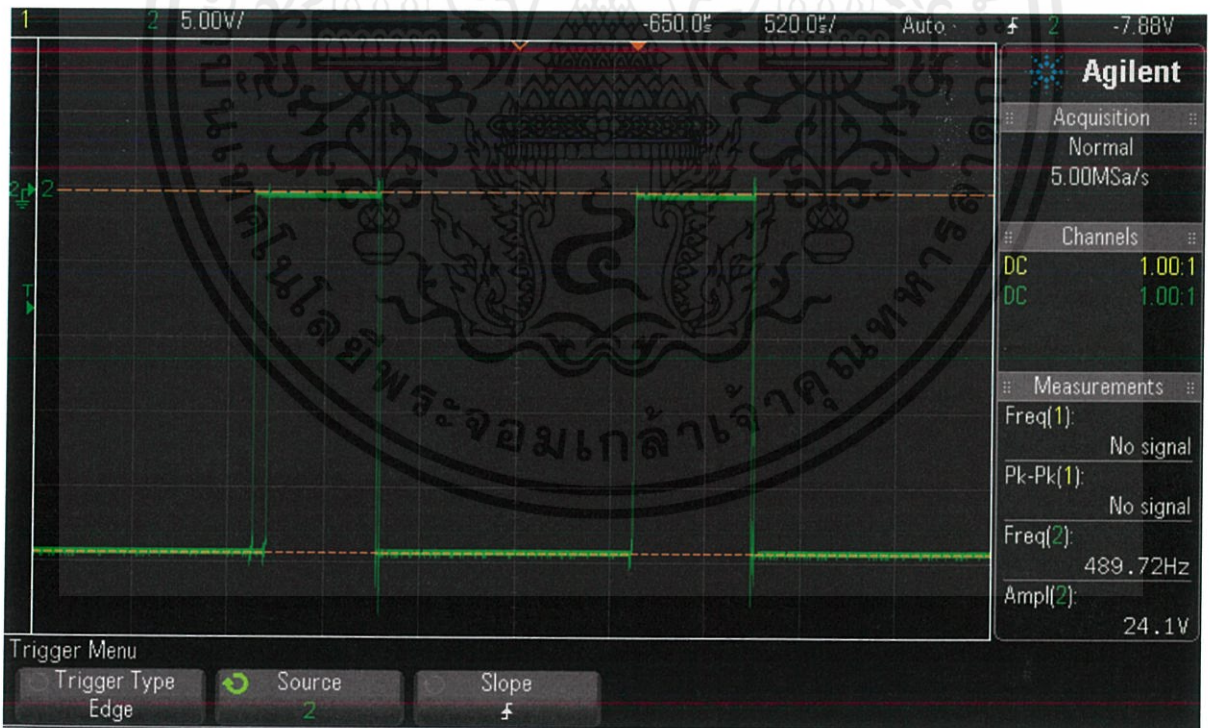


รูปที่ 4.5 สัญญาณ output ขณะไม่ได้ทำการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 สัญญาณ output ขณะเดินหน้า



รูปที่ 4.7 สัญญาณ output ขณะถอยหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 สัญญาณ output ขณะเลียฯ่าย



รูปที่ 4.9 สัญญาณ output ขณะเลียฯวฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### กราฟแสดงผลการทดลอง input

จากรูปจะเห็นว่าสัญญาณที่ออกจาก arduino มีลักษณะเป็น PWM โดยมีแรงดันประมาณ 5 v และความถี่ประมาณ 490 Hz

### กราฟแสดงผลการทดลอง output

จากรูปจะเห็นว่าสัญญาณมีลักษณะเป็น PWM โดยมีแรงดันประมาณ 24 v ซึ่งเท่ากับไฟเลี้ยงที่จ่าย และความถี่ประมาณ 490 Hz

## 4.4 การทดลองความเร็วและระยะการบังคับ

ตารางที่ 4.3 แสดงค่าความเร็วที่วัดได้

ระยะทาง(เมตร)	เวลา(วินาที)	ความเร็ว(เมตร/วินาที)
50	63.29	0.79
100	123.45	0.81
150	182.92	0.82
200	246.91	0.81
300	535.71	0.56

ตารางที่ 4.4 แสดงค่าระยะการบังคับ

ระยะทางที่บังคับ(เมตร)	การรับ-ส่ง สัญญาณ		ค่าErrorการรับส่งสัญญาณ(%)
	ส่งสัญญาณ(ค่า)	รับสัญญาณได้(ค่า)	
50	100	100	0
100	100	94	6
150	100	85	15
200	100	74	26
300	100	41	59
400	100	23	77

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.3 จะเห็นได้ว่า ที่ระยะทาง 50 เมตร ถึง 200 เมตร จะมีความเร็วเฉลี่ยประมาณ 0.81 เมตร/วินาที แต่ที่ระยะทาง 300 เมตร จะมีความเร็วที่ต่ำลงหรือเพิ่มขึ้นไป เป็นผลมาจากการส่ง-รับสัญญาณที่ผิดพลาดไป ดังที่แสดงในตารางที่ 4.4 คือ ในช่วงระยะทาง 50 ถึง 200 เมตร มี Error ที่ยังสามารถบังคับตัวหุ่นให้มีความเร็วไม่ผิดพลาดมาก แต่พอถึงช่วงระยะทาง 300 ถึง 400 เมตร ค่า Error จะมีค่ามาก ทำให้การควบคุมตัวหุ่นไม่สม่ำเสมอและผิดพลาดเพิ่มขึ้น เป็นผลให้ความเร็วช้าลงและบังคับให้ได้ตามต้องการยากขึ้น

#### 4.5 การทดลองภาระที่หุ่นรับได้ขณะขับเคลื่อน

ตารางที่ 4.5 แสดงภาระที่ตัวหุ่นรับได้

น้ำหนักภาระที่ใส่ให้ตัวหุ่น(kg)	สถานะตัวหุ่น
5	วิ่งได้ปกติความเร็วไม่ลด
10	วิ่งได้ปกติความเร็วไม่ลด
15	วิ่งได้ความเร็วเริ่มลดลง
20	วิ่งได้ความเร็วเริ่มลดลง
25	วิ่งได้ความเร็วเริ่มลดลง
30	วิ่งได้ความเร็วเริ่มลดลง
40	วิ่งได้ความเร็วเริ่มลดลงแบบเห็นได้ชัด

จากตารางที่ 4.5 จะเห็นได้ว่า หุ่นยนต์สามารถขับเคลื่อนขณะที่รับภาระโดยที่ความเร็วยังไม่เปลี่ยนแปลงหรือเปลี่ยนแปลงไปเล็กน้อยได้สูงสุด 30 kg โดยเป็นไปตามสมมติฐานที่ตั้งไว้ และความเร็วจะลดลงอย่างเห็นได้ชัดเมื่อเพิ่มภาระขึ้นไปเป็น 40 kg

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

### 5.1 สรุปผลการวิจัย

จากการศึกษา และทำการสร้างหุ่นยนต์ตรวจตราภาคพื้นดินต้นแบบนี้ พบว่าหุ่นยนต์สามารถเคลื่อนที่ได้ตามทิศทางที่ต้องการ คือ เดินหน้า ถอยหลัง เลี้ยวซ้าย และเลี้ยวขวา โดยการหมุน joystick ไปในทิศทางที่ต้องการโดยควบคุมผ่านโมดูลสื่อสารไร้สายและมีระยะการควบคุมค่อนข้างไกล คือ ประมาณ 300-400 เมตร สามารถเคลื่อนที่ขึ้นทางลาดชันที่เอียงทำมุมประมาณ 45 องศาได้ และยังสามารถบรรทุกน้ำหนักได้ประมาณ 30 kg อีกทั้งสามารถรับ-ส่งสัญญาณภาพจากตัวหุ่นยนต์ มาสู่ผู้ควบคุมเพื่อระบุตำแหน่งของตัวหุ่นได้ซึ่งเป็นไปตามวัตถุประสงค์

### 5.2 ข้อเสนอแนะ

เนื่องจากโครงการนี้มีชื่อว่าหุ่นยนต์ตรวจตราภาคพื้นดิน ดังนั้นตัวหุ่นควรมีระบบป้องกันสภาพแวดล้อมมากกว่านี้ เช่น น้ำ ฝุ่น โคลน หรือ มีโครงสร้างที่แข็งแรง ทนทาน และปกปิดมิดชิดมากกว่านี้ เป็นต้น เพื่อให้สามารถเข้าถึงได้ในพื้นที่หลากหลายรูปแบบและกล้องควรมีฟังก์ชันการใช้งานที่หลากหลาย เช่น หมุนได้ บันทึกภาพได้ เป็นต้น และควรมีเซนเซอร์สำหรับตรวจจับสิ่งต่างๆ เช่น ความร้อน ควัน สี หรือ สามารถวัดระยะห่างระหว่างตัวหุ่นกับสิ่งกีดขวางได้ เพื่อป้องกันการชน ซึ่งอาจทำให้ตัวหุ่นเสียหายได้ จึงจะสามารถปฏิบัติการกิจอันตรายหรือยากที่มนุษย์จะทำได้เองได้อย่างหลากหลาย ถูกต้องและแม่นยำ

## เอกสารอ้างอิง

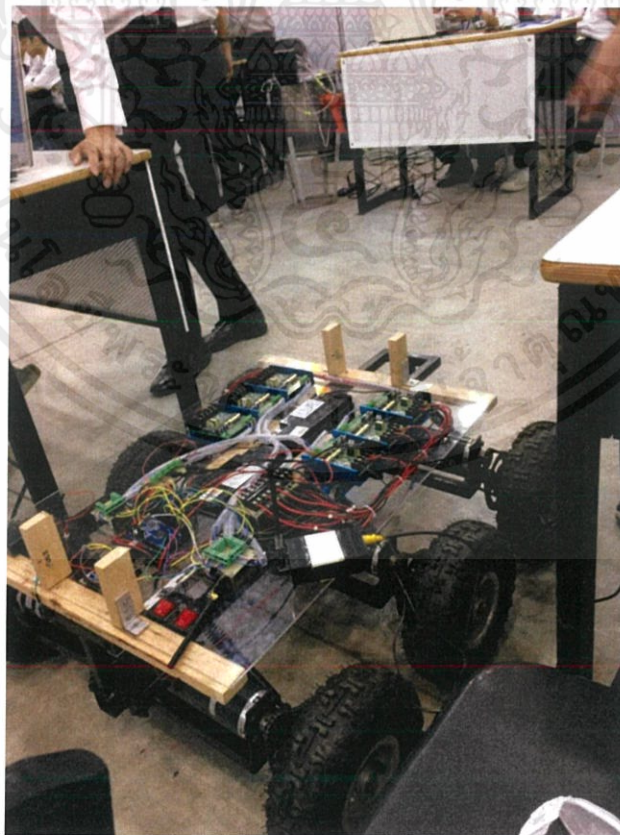
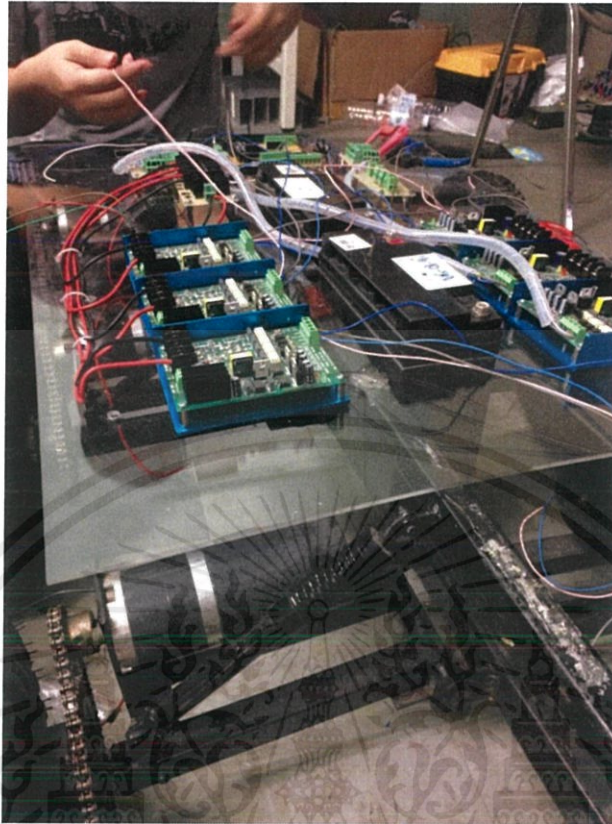
- [1] เอกชัย มະการ. (2552). *เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino*. กรุงเทพมหานคร: บริษัท อีทีที จำกัด.
- [2] Jeremy Blum (2013) : “Exploring Arduino : Tools and Techniques for Engineering Wizardy” , John Wiley & Sons, Indianapolis, Indiana.
- [3] ไกรสิทธิ์ สุวรรณศิริ, ชัยวัฒน์ ชัยชนะสงคราม , ชาญวิทย์ เอกจินดา (พ.ศ.2553): ปรินญา  
นิพนธ์ “*ยานตรวจการไร้คนขับ*” , คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร  
ลาดกระบัง



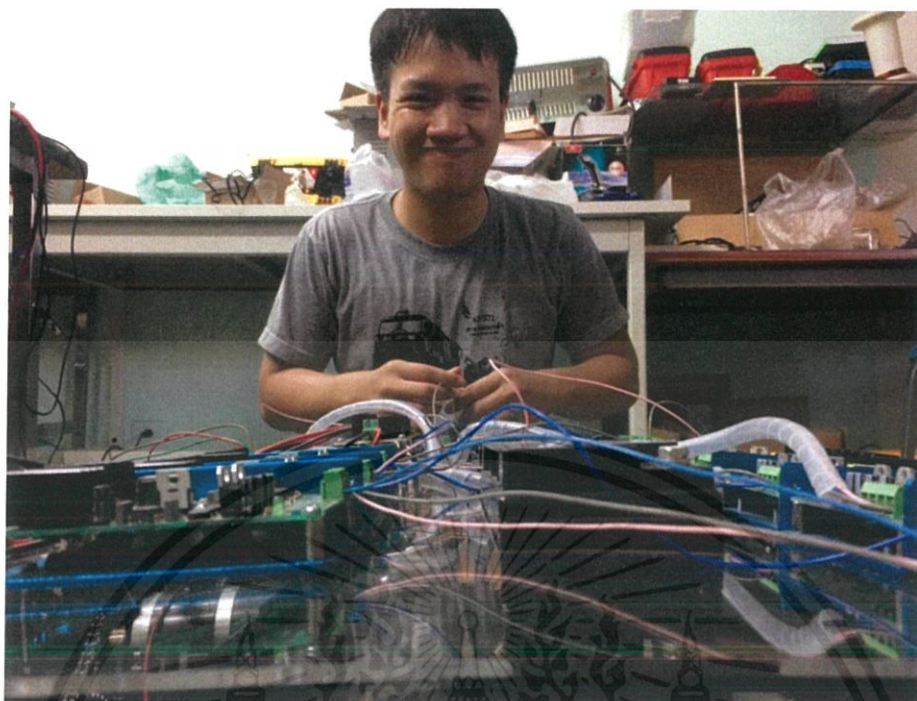


ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Code ที่ใช้ในการควบคุม

## Transmitter(remote)

```
#include <SPI.h>

#include <nRF24L01p.h>

#define Xval A0

#define Yval A1

nRF24L01p transmitter(7,8);//CSN,CE

unsigned int Val[2] ,Xdir,Ydir ;

void setup()

{

  delay(150);

  Serial.begin(115200);

  SPI.begin();

  SPI.setBitOrder(MSBFIRST);

  transmitter.channel(90);

  transmitter.TXaddress("ALL");

  transmitter.init();

}

void loop()

{

  Val[0] = analogRead(Xval);
```

```
Val[1] = analogRead(Yval);
```

```
Xdir = (Val[0])/4;
```

```
Ydir = (Val[1])/4;
```

```
transmitter.txPL(Xdir);
```

```
transmitter.txPL(Ydir);
```

```
Serial.print("XVal = ");
```

```
Serial.print(Xdir);
```

```
Serial.print(" YVal2 = ");
```

```
Serial.println(Ydir);
```

```
transmitter.send(FAST);
```

## Receiver(vehicle)

```
#include <SPI.h>
```

```
#include <nRF24L01p.h>
```

```
int pin2 = 2; //motor1
```

```
int pin3 = 3;
```

```
int pin22 = 22;
```

```
int pin4 = 4; //motor2
```

```
int pin5 = 5;
```

```
int pin24 = 24;
```

```

int pin6 = 6; //motor3
int pin9 = 9;
int pin26 = 26;
int pin10 = 10; //motor4
int pin11 = 11;
int pin28 = 28;
unsigned int Xdir,Ydir;
nRF24L01p receiver(7,8);//CSN,CE
void setup()
{
  Serial.begin(115200);
  SPI.begin();
  SPI.setBitOrder(MSBFIRST);
  receiver.channel(90);
  receiver.RXaddress("ALL");
  receiver.init();
  pinMode(pin2,OUTPUT);
  pinMode(pin3,OUTPUT);
  pinMode(pin22,OUTPUT);
  pinMode(pin4,OUTPUT);
  pinMode(pin5,OUTPUT);
  pinMode(pin24,OUTPUT);
  pinMode(pin6,OUTPUT);
  pinMode(pin9,OUTPUT);
  pinMode(pin26,OUTPUT);
  pinMode(pin10,OUTPUT);
  pinMode(pin11,OUTPUT);
  pinMode(pin28,OUTPUT);
}
void loop()
{
  if(receiver.available())
  {
    receiver.read();
    receiver.rxPL(Xdir);
    receiver.rxPL(Ydir);
    Serial.print("XVal = ");
    Serial.print(Xdir);
    Serial.print(" YVal = ");
    Serial.println(Ydir);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (Xdir<125 && Ydir<100) // forward+turn
left
{
analogWrite (pin2,0);// motor1
analogWrite (pin3,0);
analogWrite (pin4,200-Xdir);// motor2
analogWrite (pin5,0);
analogWrite (pin6,0);// motor3
analogWrite (pin9,0);
analogWrite (pin10,200-Xdir);// motor4
analogWrite (pin11,0);
digitalWrite (pin22,HIGH);
digitalWrite (pin24,HIGH);
digitalWrite (pin26,HIGH);
digitalWrite (pin28,HIGH);
}
else if (Xdir<125 && Ydir>140) // forward+turn
right
{
analogWrite (pin2,200-Xdir);// motor1
analogWrite (pin3,0);
analogWrite (pin4,0);// motor2
analogWrite (pin5,0);
analogWrite (pin6,200-Xdir);// motor3
analogWrite (pin9,0);
analogWrite (pin10,0);// motor4
}
else if (Xdir>140 && Ydir<100) //
backward+turn left
{
analogWrite (pin2,0);// motor1
analogWrite (pin3,0);
analogWrite (pin4,0);// motor2
analogWrite (pin5,0);
analogWrite (pin6,200-Xdir);// motor3
analogWrite (pin9,0);
analogWrite (pin10,0);// motor4
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

analogWrite (pin11,0);
digitalWrite (pin22,HIGH);
digitalWrite (pin24,HIGH);
digitalWrite (pin26,HIGH);
digitalWrite (pin28,HIGH);
}
else if (Xdir>140 && Ydir>140) //
backward+turn right
{
analogWrite (pin2,0);// motor1
analogWrite (pin3,Xdir-70);
analogWrite (pin4,0);// motor2
analogWrite (pin5,0);
analogWrite (pin6,0);// motor3
analogWrite (pin9,Xdir-70);
analogWrite (pin10,0);// motor4
analogWrite (pin11,0);
digitalWrite (pin22,HIGH);
digitalWrite (pin24,HIGH);
digitalWrite (pin26,HIGH);
digitalWrite (pin28,HIGH);
}
else if (Xdir<125 &&
100<=Ydir<=140)//forward
{
analogWrite (pin2,280-Xdir);// motor1
analogWrite (pin3,0);
analogWrite (pin4,280-Xdir);// motor2
analogWrite (pin5,0);
analogWrite (pin6,280-Xdir);// motor3
analogWrite (pin9,0);
analogWrite (pin10,280-Xdir);// motor4
analogWrite (pin11,0);
}
digitalWrite (pin22,HIGH);
digitalWrite (pin24,HIGH);
digitalWrite (pin26,HIGH);
digitalWrite (pin28,HIGH);
else if (Xdir>140 && 100<=Ydir<=140)
//backward
{
analogWrite (pin2,0);// motor1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

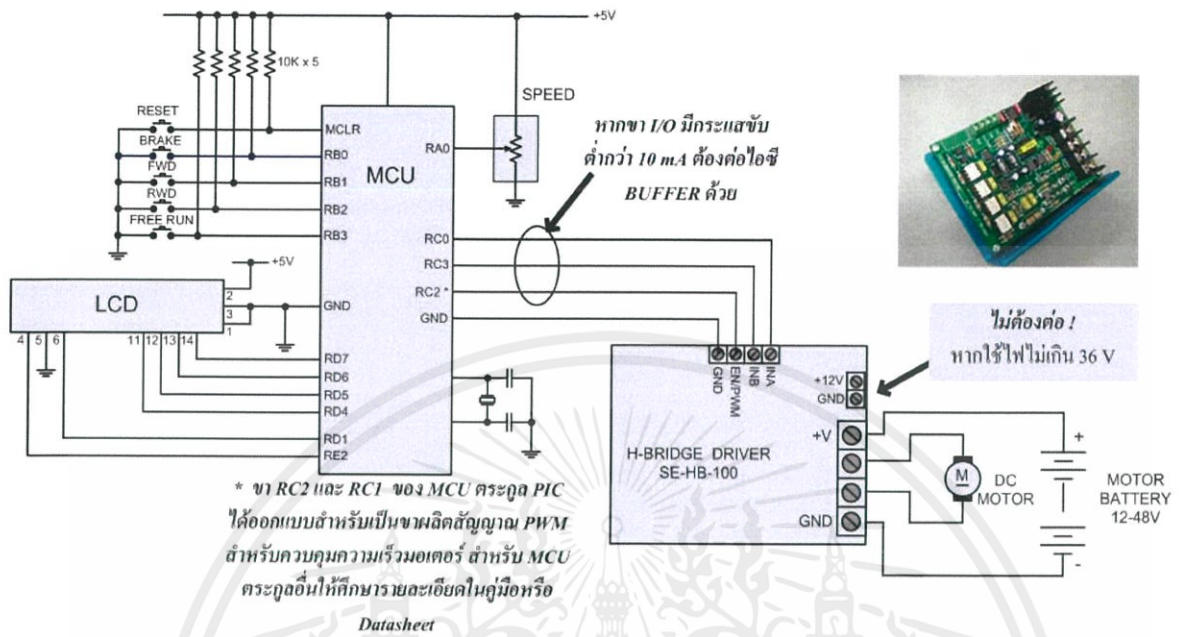
```

analogWrite (pin3,Xdir-70);
analogWrite (pin4,0);// motor2
analogWrite (pin5,Xdir-70);
analogWrite (pin6,0);// motor3
analogWrite (pin9,Xdir-70);
analogWrite (pin10,0);// motor4
analogWrite (pin11,Xdir-70);
digitalWrite (pin22,HIGH);
digitalWrite (pin24,HIGH);
digitalWrite (pin26,HIGH);
digitalWrite (pin28,HIGH);
}
else if (125<Xdir<140 && 100<=Ydir<=140)
{
analogWrite (pin2,20);// motor1
analogWrite (pin3,20);
analogWrite (pin4,20);// motor2
analogWrite (pin5,20);
analogWrite (pin6,20);// motor3
analogWrite (pin9,20);
analogWrite (pin10,20);// motor4
analogWrite (pin11,20);
digitalWrite (pin22,LOW);
digitalWrite (pin24,LOW);
digitalWrite (pin26,LOW);
digitalWrite (pin28,LOW);
}
else // stop
{
analogWrite (pin2,0);// motor1
analogWrite (pin3,0);
analogWrite (pin4,0);// motor2
analogWrite (pin5,0);
analogWrite (pin6,0);// motor3
analogWrite (pin9,0);
analogWrite (pin10,0);// motor4
digitalWrite (pin22,HIGH);
digitalWrite (pin24,HIGH);
digitalWrite (pin26,HIGH);
digitalWrite (pin28,HIGH);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# รายละเอียดการใช้งาน บอร์ดขับมอเตอร์ดีซี แบบ H-Bridge 80A



- หมายเหตุ**
1. ตัว JUMPER ในบอร์ด หากต่อไฟใช้งานไม่เกิน 36 V ให้เสียบต่อไว้ที่ตำแหน่งเดิม คือ 12-36 V
  2. บอร์ดขับมอเตอร์แบบ H-Bridge รุ่นนี้ได้ออกแบบให้สัญญาณอินพุตกับเอาต์พุตแยกกันทางแสง ดังนั้นจึงสามารถแยกขา GND แหล่งจ่ายไฟขับเคลื่อนมอเตอร์ ออกจากไฟเลี้ยง MCU ได้
  3. ที่ขาอินพุต INA , INB, EN/PWM ต้องการกระแสไม่ต่ำกว่า 10 mA ที่ 5V

## รายละเอียดทางเทคนิค

1. Output : Single motor driver
  - Motor DC Supply 12-48 V 80A (Max. กระแสพัลส์)
  - Full-Complementary Power MOSFET Driver
  - With ultra-fast reverse recovery protection diodes
2. Input :
  - Full Opto-isolated input interface signals
  - 5V 8 mA TTL – Level

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Drive Mode : independently with :

- ON – OFF Control
- Direction Control
- Speed Control (PWM Drives)

4. PWM Frequency : 400 Hz - 1000 Hz ( 400-800 Hz Recommend )

### ตาราง ควบคุมการทำงาน

EN/PWM	INB	INA	การทำงานของมอเตอร์
0V	X	X	Free Run Stop (หยุดเมื่อหมดแรงเฉื่อย)
5V	0V	5V	หมุนเดินหน้า
5V	5V	0V	หมุนกลับทาง
5V	5V	5V	Fast Stop หรือ Brake
5V	0V	0V	Fast Stop หรือ Brake



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# nRF24L01+

## Single Chip 2.4GHz Transceiver

### Product Specification v1.0

#### Key Features

- Worldwide 2.4GHz ISM band operation
- 250kbps, 1Mbps and 2Mbps on air data rates
- Ultra low power operation
- 11.3mA TX at 0dBm output power
- 13.5mA RX at 2Mbps air data rate
- 900nA in power down
- 26µA in standby-I
- On chip voltage regulator
- 1.9 to 3.6V supply range
- Enhanced ShockBurst™
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™
- Drop-in compatibility with nRF24L01
- On-air compatible in 250kbps and 1Mbps with nRF2401A, nRF2402, nRF24E1 and nRF24E2
- Low cost BOM
- ±60ppm 16MHz crystal
- 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

#### Applications

- Wireless PC Peripherals
- Mouse, keyboards and remotes
- 3-in-1 desktop bundles
- Advanced Media center remote controls
- VoIP headsets
- Game controllers
- Sports watches and sensors
- RF remote controls for consumer electronics
- Home and commercial automation
- Ultra low power sensor networks
- Active RFID
- Asset tracking systems
- Toys

All rights reserved.

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

September 2008

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.1 Features

Features of the nRF24L01+ include:

- Radio
  - ▶ Worldwide 2.4GHz ISM band operation
  - ▶ 126 RF channels
  - ▶ Common RX and TX interface
  - ▶ GFSK modulation
  - ▶ 250kbps, 1 and 2Mbps air data rate
  - ▶ 1MHz non-overlapping channel spacing at 1Mbps
  - ▶ 2MHz non-overlapping channel spacing at 2Mbps
- Transmitter
  - ▶ Programmable output power: 0, -6, -12 or -18dBm
  - ▶ 11.3mA at 0dBm output power
- Receiver
  - ▶ Fast AGC for improved dynamic range
  - ▶ Integrated channel filters
  - ▶ 13.5mA at 2Mbps
  - ▶ -82dBm sensitivity at 2Mbps
  - ▶ -85dBm sensitivity at 1Mbps
  - ▶ -94dBm sensitivity at 250kbps
- RF Synthesizer
  - ▶ Fully integrated synthesizer
  - ▶ No external loop filter, VCO varactor diode or resonator
  - ▶ Accepts low cost  $\pm 60$ ppm 16MHz crystal
- Enhanced ShockBurst™
  - ▶ 1 to 32 bytes dynamic payload length
  - ▶ Automatic packet handling
  - ▶ Auto packet transaction handling
  - ▶ 6 data pipe MultiCeiver™ for 1:6 star networks
- Power Management
  - ▶ Integrated voltage regulator
  - ▶ 1.9 to 3.6V supply range
  - ▶ Idle modes with fast start-up times for advanced power management
  - ▶ 26 $\mu$ A Standby-I mode, 900nA power down mode
  - ▶ Max 1.5ms start-up from power down mode
  - ▶ Max 130us start-up from standby-I mode
- Host Interface
  - ▶ 4-pin hardware SPI
  - ▶ Max 10Mbps
  - ▶ 3 separate 32 bytes TX and RX FIFOs
  - ▶ 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

### 3 Absolute maximum ratings

Note: Exceeding one or more of the limiting values may cause permanent damage to nRF24L01+.

Operating conditions	Minimum	Maximum	Units
<b>Supply voltages</b>			
VDD	-0.3	3.6	V
VSS		0	V
<b>Input voltage</b>			
V <sub>I</sub>	-0.3	5.25	V
<b>Output voltage</b>			
V <sub>O</sub>	VSS to VDD	VSS to VDD	
<b>Total Power Dissipation</b>			
P <sub>D</sub> (T <sub>A</sub> =85°C)		60	mW
<b>Temperatures</b>			
Operating Temperature	-40	+85	°C
Storage Temperature	-40	+125	°C

Table 2. Absolute maximum ratings



## 4 Operating conditions

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
VDD	Supply voltage		1.9	3.0	3.6	V
VDD	Supply voltage if input signals >3.6V		2.7	3.0	3.3	V
TEMP	Operating Temperature		-40	+27	+85	°C

Table 3. Operating conditions



## 5.6 DC characteristics

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
$V_{IH}$	HIGH level input voltage		$0.7V_{DD}$		$5.25^a$	V
$V_{IL}$	LOW level input voltage		$V_{SS}$		$0.3V_{DD}$	V

a. If the input signal  $>3.6V$ , the  $V_{DD}$  of the nRF24L01+ must be between 2.7V and 3.3V ( $3.0V \pm 10\%$ )

Table 12. Digital input pin

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
$V_{OH}$	HIGH level output voltage ( $I_{OH}=-0.25mA$ )		$V_{DD} - 0.3$		$V_{DD}$	V
$V_{OL}$	LOW level output voltage ( $I_{OL}=0.25mA$ )				0.3	V

Table 13. Digital output pin

## 5.7 Power on reset

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
$T_{PUP}$	Power ramp up time	a			100	ms
$T_{POR}$	Power on reset	b	1		100	ms

a. From 0V to 1.9V.

b. Measured from when the  $V_{DD}$  reaches 1.9V to when the reset finishes.

Table 14. Power on reset