

รถหุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ

Hand gesture control robot car

อภิสิทธิ์ ไชยปลัด

Apisit Chaipalad

อภิสิทธิ์ นาใจดี

Apisit Najaidee

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2561

รถหุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ

Hand gesture control robot car

โดย

อภิสิทธิ์ ไชยปลัด

อภิสิทธิ์ นาใจดี

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.ยุทธนา คิดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2561

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง รถหุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ

Hand gesture control robot car

ผู้จัดทำ นายอภิสิทธิ์ ไชยปลัด รหัสประจำตัว 58011417

นายอภิสิทธิ์ นาใจดี รหัสประจำตัว 58011418

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(ผู้ช่วยศาสตราจารย์ ดร.ยุธนา คิดใจเดียว)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาานิพนธ์	รถหุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ
นักศึกษา	นายอภิสิทธิ์ ไชยปลัด รหัสประจำตัว 58011417 นายอภิสิทธิ์ นาใจดี รหัสประจำตัว 58011418
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2561
อาจารย์ที่ปรึกษาปริญญาานิพนธ์	ผู้ช่วยศาสตราจารย์ ดร.ยุทธนา คิดใจเดียว

บทคัดย่อ

ผู้พัฒนาได้ประยุกต์ใช้สัญญาณ Electromyography (EMG) หรือสัญญาณไฟฟ้ากล้ามเนื้อจากร่างกายของมนุษย์โดยใช้อุปกรณ์ที่มีชื่อว่า MYO Gesture Control Armband เป็นเซนเซอร์ตรวจจับสัญญาณไฟฟ้ากล้ามเนื้อขณะทำท่าทางของมือในลักษณะต่างๆ เช่น กำมือ กางมือ โบกมือ เข้าหาตัว โบกมือออกจากตัว แล้วส่งสัญญาณไฟฟ้ากล้ามเนื้อผ่านทางบลูทูธไปประมวลผลที่บอร์ดไมโครคอนโทรลเลอร์เพื่อนำไปควบคุมการหมุนของมอเตอร์ทำให้รถหุ่นยนต์สามารถเคลื่อนที่ในทิศทางต่างๆได้ เช่น การเดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ซึ่งโครงงานนี้สามารถเป็นต้นแบบในการพัฒนาต่อยอดโดยการใช้ท่าทางมือควบคุมหุ่นยนต์รูปแบบอื่นๆได้

Thesis title	Hand gesture control robot car
Student	Mr. Apisit Chaipalad Student ID 58011417 Mr. Apisit Najaidee Student ID 58011418
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2018
Thesis Advisor	Assistant Professor Dr. Yuttana Kitjaidure

ABSTRACT

Developers have applied Electromyography signal (EMG) by using a device called Myo Gesture Control Armband. It is a sensor that detects electric signals from muscles while making a hand gesture such as fist, finger spread, wave in and wave out. Then it will send EMG signal via Bluetooth and process in microcontroller board to control the rotation of the motor and makes the robot car move in different directions. Such as forward, backward, turn left and turn right. This project can be a model for further development by using hand gestures to control other form of robot.

กิตติกรรมประกาศ

โครงการ “รถหุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ ” สำเร็จลุล่วงได้ด้วยดี ด้วยความกรุณาจากผู้ช่วยศาสตราจารย์ ดร.ยุทธนา คิดใจเดียว อาจารย์ที่ปรึกษา ที่ให้ความรู้ต่างๆเกี่ยวกับความรู้พื้นฐานในการออกแบบการทำงานของชิ้นงานและคอยให้คำแนะนำตลอดการดำเนินงานรวมถึงเอื้อเฟื้อสถานที่และอุปกรณ์ในการทดสอบชิ้นงาน

ขอขอบพระคุณผู้ปกครองของคณะผู้จัดทำที่เป็นผู้สนับสนุนเงินทุนในการทำโครงการนี้และคอยเป็นผู้ให้กำลังใจในการทำงาน รวมไปถึงนักศึกษารุ่นพี่ที่ให้คำแนะนำ รวมทั้งช่วยเหลือกันแก้ปัญหาที่เกิดขึ้นในระหว่างการทำโครงการนี้ทั้งหมดทำให้โครงการสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้คณะผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์สำหรับผู้ที่สนใจไม่มากก็น้อย

อภิสิทธิ์ ไชยปลัด

อภิสิทธิ์ นาใจดี



สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูปภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ขอบเขตของงานวิจัย.....	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	1
บทที่ 2 หลักการและทฤษฎี.....	2
2.1 คลื่นไฟฟ้ากล้ามเนื้อ(Electromyography).....	2
2.1.1 ความหมายของคลื่นไฟฟ้ากล้ามเนื้อ.....	2
2.1.2 เครื่องวัดคลื่นไฟฟ้ากล้ามเนื้อ.....	2
2.2 ความรู้เบื้องต้นเกี่ยวกับกล้ามเนื้อส่วนหัวไหล่และแขน.....	3
2.2.1 กล้ามเนื้อส่วนหัวไหล่.....	3
2.2.2 กล้ามเนื้อแขนส่วนต้น.....	4
2.2.3 กล้ามเนื้อส่วนปลายแขน.....	4
2.2.4 กล้ามเนื้อส่วนมือและนิ้ว.....	4
2.3 Arduino Microcontroller.....	5
2.3.1 Arduino.....	5
2.3.2 Arduino Board.....	5
2.3.3 Arduino IDE.....	5
2.3.4 Layout & Pin out Arduino Board (Model: Arduino UNO R3).....	6
2.4 Layout & Pin out Motor Shield Driver (Model: L293D).....	7
2.5 HM-11 Bluetooth Module.....	8
2.6 มอเตอร์ไฟฟ้ากระแสตรง.....	8

สารบัญ(ต่อ)

หน้า

2.6.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง.....	8
2.6.2 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง.....	9
2.6.3 การควบคุมความเร็วของมอเตอร์ขั้นพื้นฐาน.....	10
2.7 MYO Gesture Control Armband.....	12
บทที่ 3 วิธีดำเนินการวิจัย.....	13
3.1 Block Diagram.....	13
3.2 การเชื่อมต่อ MYO Armband กับ Bluetooth HM-11.....	14
3.3 การรับข้อมูลท่าทางจาก MYO Gesture Control Armband.....	16
3.4 โครงร่างของตัวรถหุ่นยนต์.....	18
3.5 Schematic and PCB	20
3.6 รถหุ่นยนต์.....	21
บทที่ 4 การทดลองและผลการทดลอง.....	22
4.1 ผลการทดสอบการตรวจจับท่าทางของมือด้วย MYO Armband.....	22
4.2 ระยะเวลาเชื่อมต่อระหว่าง MYO Armband กับรถหุ่นยนต์	23
4.3 ผลการทดสอบการหมุนของมอเตอร์ตามท่าทางการเคลื่อนไหวของมือ.....	23
4.4 ผลการทดสอบการเคลื่อนไหวของรถหุ่นยนต์ตามท่าทางการเคลื่อนไหวของมือ.....	24
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	25
5.1 สรุปผลการวิจัย.....	25
5.2 ปัญหาและอุปสรรค.....	25
บรรณานุกรม.....	26
ภาคผนวก.....	27
ภาคผนวก ก.....	27
ภาคผนวก ข.....	31

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ชนิดศักย์ไฟฟ้าและความถี่ที่ตอบสนองของสัญญาณไฟฟ้าสมอง หัวใจและกล้ามเนื้อ.....	3
ตารางที่ 4.1 ผลการทดสอบการหมุนของมอเตอร์ตามท่าทางการเคลื่อนไหวของมือ.....	23
ตารางที่ 4.2 ผลการทดสอบการเคลื่อนไหวของรถหุ่นยนต์ตามท่าทางการเคลื่อนไหวของมือ.....	24



สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 (ก) Needle electrode.....	3
รูปที่ 2.1 (ข) Surface of the upper limb.....	3
รูปที่ 2.2 โปรแกรม Arduino IDE.....	5
รูปที่ 2.3 บอร์ด Arduino UNO R3.....	6
รูปที่ 2.4 บอร์ด Motor Shield Drive L293D.....	7
รูปที่ 2.5 HM-11 Bluetooth Module.....	8
รูปที่ 2.6 DC Motor.....	8
รูปที่ 2.7 วงจรควบคุมความเร็วของมอเตอร์แบบใช้ตัวต้านทานอนุกรม.....	10
รูปที่ 2.8 การควบคุมความเร็วมอเตอร์โดยเปลี่ยนค่าแรงดัน.....	10
รูปที่ 2.9 แสดงสัญญาณ PWM ซึ่งแสดงค่า Duty cycle ที่ต่าง ๆ กัน.....	11
รูปที่ 2.10 MYO Gesture Control Armband.....	12
รูปที่ 3.1 Block Diagram กระบวนการรับสัญญาณ EMG นำไปควบคุมมอเตอร์.....	13
รูปที่ 3.2 MYO Armband และ Bluetooth Dongle 4.0	14
รูปที่ 3.3 MYO Armband และ Bluetooth Bluetooth Module HM-11.....	14
รูปที่ 3.4 การเชื่อมต่อวงจรสำหรับการ Flash HM-11.....	15
รูปที่ 3.5 การสั่งเปิดโปรแกรม CCLoader.exe ใน Command Prompt.....	15
รูปที่ 3.6 การ import firmware ในโปรแกรม CCLoader.exe ผ่าน Command Prompt.....	15
รูปที่ 3.7 โปรแกรม CCLoader flash Firmware CC2541 ลงใน Bluetooth HM-11.....	16
รูปที่ 3.8 MYO Bridge.....	17
รูปที่ 3.9 การเปรียบเทียบท่าทางที่ตรวจจับจาก MYO กับ MYO Bridge Library.....	17
รูปที่ 3.10 ตัวอย่างโค้ดอาดูโนที่ใช้ในเปรียบเทียบสัญญาณ.....	17
รูปที่ 3.11 โครงร่างส่วนบนและล่างที่ออกแบบในโปรแกรม Illustrator.....	18
รูปที่ 3.12 โครงร่างส่วนบริเวณตัวรถที่ออกแบบในโปรแกรม Illustrator.....	19
รูปที่ 3.13 PCB.....	20
รูปที่ 3.14 Schematic.....	20
รูปที่ 3.15 รถหุ่นยนต์.....	21
รูปที่ 4.1 ผลการทดสอบการตรวจจับท่าทางของมือด้วย MYO Armband	22

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีหุ่นยนต์ประสานกับความรู้ทางการแพทย์ ก่อให้เกิดการพัฒนาสิ่งประดิษฐ์ใหม่ๆ มากมาย ซึ่งหนึ่งในความรู้ทางการแพทย์ที่สามารถนำมาประยุกต์ใช้งานทางด้านอิเล็กทรอนิกส์ได้ คือ สัญญาณไฟฟ้ากล้ามเนื้อ เนื่องจากกล้ามเนื้อเป็นเนื้อเยื่อที่ไวต่อสิ่งเร้า สามารถสร้างสัญญาณไฟฟ้าและส่งผ่านไปยังเส้นประสาทตามจุดต่างๆ ของร่างกาย ศักย์ไฟฟ้าที่เกิดการเปลี่ยนแปลงสามารถนำไปใช้จำแนกประเภทการเคลื่อนไหว ณ ขณะนั้นได้ โดยโครงการนี้มีจุดประสงค์เพื่อนำไปควบคุมการทำงานของหุ่นยนต์ให้ทำงานตามคำสั่งที่ได้รับจากการประมวลผลของสัญญาณไฟฟ้ากล้ามเนื้อด้วยโปรแกรมคอมพิวเตอร์และไมโครคอนโทรลเลอร์

1.2 วัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อนำความรู้ทางการแพทย์มาประยุกต์ใช้ร่วมกับการประดิษฐ์หุ่นยนต์
- 1.2.2 เพื่อศึกษาและวิเคราะห์สัญญาณไฟฟ้าที่ถูกสร้างขึ้นจากการเกร็งตัวของกล้ามเนื้อ
- 1.2.3 เพื่อออกแบบระบบหุ่นยนต์ที่สามารถใช้สัญญาณไฟฟ้ากล้ามเนื้อไปควบคุมการทำงานได้

1.3. สมมติฐานของการศึกษา

รถหุ่นยนต์บังคับด้วยการเคลื่อนไหวของมือสามารถเคลื่อนไหวได้ตามท่าทางของมือที่ถูกกำหนดโดยใช้สัญญาณกล้ามเนื้อไฟฟ้าเป็นสัญญาณในการควบคุมและสามารถนำไปประยุกต์ใช้งานได้จริงในการควบคุมหุ่นยนต์รูปแบบอื่นๆ

1.4 ขอบเขตการวิจัย

รถหุ่นยนต์บังคับด้วยการเคลื่อนไหวของมือใช้การตรวจจับสัญญาณไฟฟ้ากล้ามเนื้อด้วย EMG sensor แล้วนำสัญญาณที่ได้ไปประมวลผลผ่านซอฟต์แวร์เพื่อทำการคิดแยกท่าทางและสามารถนำผลลัพธ์ที่ได้ไปใช้ในการควบคุมรถหุ่นยนต์บังคับด้วยการเคลื่อนไหวของมือให้เคลื่อนที่ไปในทิศทางตามที่ต้องการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 เข้าใจถึงกลไกการสร้างสัญญาณไฟฟ้าที่มาจากกล้ามเนื้อของร่างกายมนุษย์
- 1.5.2 สามารถวิเคราะห์สัญญาณไฟฟ้ากล้ามเนื้อและนำผลลัพธ์ที่ได้ไปใช้ควบคุมระบบต่างๆ
- 1.5.3 สามารถนำโครงการนี้ไปต่อยอดในการควบคุมหุ่นยนต์รูปแบบอื่นๆ ได้

บทที่ 2

หลักการและทฤษฎี

2.1 คลื่นไฟฟ้ากล้ามเนื้อ(Electromyography)

2.1.1 ความหมายของคลื่นไฟฟ้ากล้ามเนื้อ

คลื่นไฟฟ้ากล้ามเนื้อ หมายถึง ระดับความตึงตัวของกล้ามเนื้อที่เกิดจากการกระตุ้นอย่างรุนแรง กล้ามเนื้อประกอบไปด้วยเส้นใยจำนวนมากหลายล้านเส้นเมื่อเส้นใยกล้ามเนื้อเกิดการเกร็งนั้นเกิดจากการกระทำของเซลล์ประสาททางกลไก (Motorneuron) ที่ส่งกระแสประสาทไปยังเส้นใยกล้ามเนื้อ นั้น ๆ ไม่ว่าจะเวลาใดก็ตามที่เส้นใยกล้ามเนื้อ 2-3 เส้นที่อยู่ในขณะพักตัวซึ่งดูเหมือนจะมีการเกร็งอยู่ ภายใต้สภาวะปกติ การกระตุ้นที่เกิดขึ้นที่กล้ามเนื้อจะถูกกระตุ้นจนกระทั่งเกิดการตอบสนองทั้งหมด หรืออาจจะเพียงบางส่วนของกล้ามเนื้อรวมเข้าด้วยกันเพื่อทำให้เกิดการตอบสนองของกล้ามเนื้อทั้ง การเปลี่ยนแปลงอย่างเห็นได้ชัดในการส่งกระแสไฟฟ้ามาที่พื้นผิวด้านนอกของผิวหนังไม่ใช่จะถูกวัดได้ในทุกสภาวะอารมณ์ และความรู้สึกที่เกิดขึ้นจากจิตใจ กิจกรรมของกล้ามเนื้อระดับพื้นผิวนั้นสามารถวัดได้ด้วยเครื่องวัดคลื่นไฟฟ้ากล้ามเนื้อที่เรียกว่า Electromyography (EMG) ซึ่งเป็นเครื่องมือวัดที่ไวต่อการเปลี่ยนแปลงของกิจกรรมทางไฟฟ้าของกล้ามเนื้อในขณะที่กล้ามเนื้อเกิดการเกร็ง

2.1.2 เครื่องวัดคลื่นไฟฟ้ากล้ามเนื้อ

เครื่องวัดคลื่นไฟฟ้ากล้ามเนื้อ (Electromyography หรือ Electromyogram) คือ เครื่องที่ใช้สำหรับบันทึกสัญญาณไฟฟ้าที่เกิดจากการทำงานของ Motor unit ซึ่ง Motor unit นั้นก็คือ การหดตัวของกล้ามเนื้อ ถูกควบคุมโดยเซลล์ประสาทสั่งการ (Motor neuron) ซึ่งจะอยู่ที่ไขสันหลังสำหรับกล้ามเนื้อแขน ขา ลำตัว และจะอยู่ที่ก้านสมอง สำหรับกล้ามเนื้อศีรษะและคอ 1 Motor unit คือ เซลล์ประสาท 1 ตัวกับ Muscle fiber จำนวนหนึ่งที่เซลล์ประสาทตัวนั้นไปเลี้ยงกล้ามเนื้อที่ทำงาน ละเอียดจะมี Motor unit ขนาดเล็ก เช่น กล้ามเนื้อมือ การทำลาย Motor neuron ทำให้เกิดการ อัมพาต และผลที่ได้จากการบันทึก เรียกว่า Electromyography หรือที่เรียกกันโดยย่อว่า EMG นั้นเองการตรวจวัดสัญญาณไฟฟ้าอีเอ็มจี (Electromyography) เป็นที่ทราบกันดีแล้วว่า สมอง เส้นประสาท และกล้ามเนื้อเป็นเนื้อเยื่อที่ไวต่อสิ่งเร้า สามารถสร้างสัญญาณไฟฟ้าและส่งผ่านสัญญาณ เมื่อถูกกระตุ้นด้วยสิ่งเร้าดังกล่าวไปตามเส้นประสาทใยกล้ามเนื้อ การตรวจคลื่นไฟฟ้ากล้ามเนื้อหรือ อีเอ็มจี เป็นเทคนิคที่ใช้ตรวจวัดสัญญาณไฟฟ้าที่สร้างจากเส้นประสาทและกล้ามเนื้อโดยตรงซึ่งคล้ายกับการตรวจคลื่นไฟฟ้าหัวใจ และคลื่นไฟฟ้าสมอง แตกต่างเฉพาะขนาดของความถี่และศักย์ไฟฟ้า ดังตารางที่ 2.1

ตารางที่ 2.1 ชนิดศักย์ไฟฟ้าและความถี่ที่ตอบสนองของสัญญาณไฟฟ้าสมอง หัวใจและกล้ามเนื้อ

ชนิดของสัญญาณไฟฟ้า	ศักย์ไฟฟ้า (mV)	ความถี่ตอบสนอง (Hz)
สมอง (EEG)	0.001-0.10	0.02-100
หัวใจ (ECG,EKG)	0.02-3.0	0.1-30
กล้ามเนื้อ (EMG)	0.003-5.0	2-10,000

เครื่องมือและอุปกรณ์สำหรับตรวจวัดคลื่นไฟฟ้ากล้ามเนื้อ ประกอบด้วยตัวเครื่องและอิเล็กโทรดบันทึกสัญญาณเข้า ตัวเครื่องมักมีอุปกรณ์/วงจรไฟฟ้าที่ทำหน้าที่กรองขยายสัญญาณและส่วนแสดงผลอิเล็กทรอนิกส์ หรือขั้วไฟฟ้าที่ใช้สำหรับตรวจวัดคลื่นไฟฟ้ากล้ามเนื้ออีเอ็มจี นั้นแบ่งเป็นสองชนิด คือ ขั้วไฟฟ้าสำหรับกระตุ้น และขั้วไฟฟ้าสำหรับบันทึกสัญญาณไฟฟ้า ขั้วไฟฟ้าสำหรับกระตุ้นมักเป็นตัวกระตุ้นขนาดเล็ก และเป็นชนิดสองขั้ว (Bipolar electrode) ยึดติดกับส่วนอิเล็กทรอนิกส์สำหรับบันทึกสัญญาณไฟฟ้านั้น แบ่งออกเป็น 2 ชนิด คือ อิเล็กโทรดแบบเข็ม (Needle electrode) มักใช้แทงเข้าไปในกล้ามเนื้อเพื่อบันทึกสัญญาณ/ ศักย์ไฟฟ้าที่ใยกล้ามเนื้อโดยตรง และอิเล็กโทรดแบบวางที่ผิวหนัง (Surface electrode) เป็นแผ่นขั้วไฟฟ้าที่วางบนผิวหนังบริเวณที่วัด



รูปที่ 2.1 (ก) Needle electrode และ (ข) Surface electrode

2.2 ความรู้เบื้องต้นเกี่ยวกับกล้ามเนื้อส่วนหัวไหล่และแขน

2.2.1 กล้ามเนื้อส่วนหัวไหล่

2.2.1.1 Deltoid เป็นกล้ามเนื้อคลายขนนกหลาย ๆ อันมารวมกันเป็นมัดใหญ่หนารูปสามเหลี่ยมจุดเกาะอยู่ที่ไหปลาร้า และกระดูกสะบัก แล้วไปเกาะที่ตอนกลางของกระดูกต้นแขน ทำหน้าที่ยกไหล่และยกต้นแขน เป็นส่วนที่บ่งบอกลักษณะเพศชายได้อย่างชัดเจน

2.2.1.2 Supraspinatus เริ่มเกาะจากกระดูกสะบักไปยังกระดูกต้นแขน ทำหน้าที่ช่วยกล้ามเนื้อ Deltoid ในการยก หรือกางแขน

2.2.1.3 Infraspinatus เริ่มเกาะจากกระดูกสะบักไปยังกระดูกต้นแขน ทำหน้าที่หมุนต้นแขนออกด้านนอก และดึงแขนไปด้านหลัง

2.2.1.4 Teres minor และ Teres major เกาะที่กระดูกสะบัก แล้วมาเกาะที่กระดูกต้นแขนโดย ทำหน้าที่หมุนแขนออกด้านนอกและหมุนแขนเข้าด้านใน

2.2.1.5 Subscapularis มีจุดเกาะที่กระดูกสะบักและต้นแขน ทำหน้าที่หมุนต้นแขนเข้าด้านใน

2.2.2 กล้ามเนื้อแขนส่วนต้น

2.2.2.1 *Biceps brachii* เป็นกล้ามเนื้อด้านหน้าของต้นแขน มีที่เกาะส่วนบนแยก 2 ทาง คือ เกาะจาก Coracoid process และ Supraglenoid tubercle ไปยัง Tuberosity ของกระดูกปลายแขนท่อนนอก (Radius) ทำหน้าที่งอต้นแขนและปลายแขน หมุนแขนเข้าและดึงออก

2.2.2.2 *Brachialis* เป็นกล้ามเนื้อต้นแขนที่อยู่ตรงกลางค่อนมาด้านล่าง เกาะจากกระดูกต้นแขนไปยัง Tuberosity ของกระดูกปลายแขนท่อนใน (Ulna) ทำหน้าที่งอข้อศอก

2.2.2.3 *Coracobrachialis* เกาะจาก Coracoid process ของกระดูกสะบักไปยังกึ่งกลางของกระดูกต้นแขน ทำหน้าที่งอต้นแขน

2.2.2.4 *Triceps brachii* เป็นกล้ามเนื้อด้านหลังของต้นแขน ปลายบนแยก 3 ทางเกาะที่กระดูกสะบักหนึ่งที และอีก 2 ทางเกาะที่กระดูกต้นแขน และมีจุดเกาะปลายที่กระดูกปลายแขนท่อนใน (Ulna) กล้ามเนื้อมัดนี้จะทำหน้าที่ตรงกันข้ามกับกล้ามเนื้อ *Biceps brachii* คือ ทำหน้าที่เหยียดปลายแขน

2.2.3 กล้ามเนื้อส่วนปลายแขน

2.2.3.1 *Brachioradialis* เป็นกล้ามเนื้อด้านนอกของปลายแขน มีจุดเกาะต้นที่ตอนล่างของกระดูกแขน ไปเกาะที่ด้านนอกของกระดูกปลายแขนท่อนนอก (Radius) ทำหน้าที่งอปลายแขน

2.2.3.2 *Flexor carpi radialis* เป็นกล้ามเนื้อที่อยู่ด้านหน้าของปลายแขน มีจุดเกาะที่กระดูกต้นแขนแล้วมาเกาะที่กระดูกฝ่ามือชิ้นที่ 2 และ 3 ทำหน้าที่งอข้อมือและกางมือ

2.2.3.3 *Palmaris longus* เป็นกล้ามเนื้อที่อยู่ทางด้านหน้าของแขน จุดเกาะต้นเริ่มจากกระดูกต้นแขนไปยังกระดูกปลายแขน แล้วกลายเป็นเอ็น (Tendon) ไปเกาะที่ฝ่ามือทำหน้าที่งอข้อมือ

2.2.3.4 *Flexor carpi ulnaris* เป็นกล้ามเนื้อที่อยู่ทางด้านหลังของกระดูกปลายแขนท่อนใน (Ulna) ผ่านมาที่ข้อมือ ทำหน้าที่งอข้อมือ

2.2.3.5 *Extensor carpi radialis longus* เป็นกล้ามเนื้อที่มีจุดเกาะต้นจากกระดูกต้นแขนแล้วไปเกาะที่กระดูกฝ่ามือทางด้านหลัง ทำหน้าที่กางและเหยียดข้อมือ

2.2.3.6 *Extensor digitorum* เป็นกล้ามเนื้อที่มีจุดเกาะต้นจากกระดูกต้นแขน และมีปลายเป็นเอ็น 4 อัน ไปเกาะยังกระดูกนิ้วมือทั้ง 4 นิ้ว ทำหน้าที่เหยียดนิ้วมือและข้อมือ

2.2.4 กล้ามเนื้อส่วนมือและนิ้ว

กล้ามเนื้อส่วนมือและนิ้วมือ เป็นกล้ามเนื้อขนาดเล็กและสั้น ส่วนมากจะเป็นเอ็นของกล้ามเนื้อซึ่งติดต่อมาจากแขนท่อนล่าง ทำหน้าที่ช่วยในการงอและเหยียดมือและข้อมือรวมทั้งช่วยให้นิ้วหัวแม่มือสามารถเคลื่อนไปแตะนิ้วอื่น ๆ ได้จึงเรียกว่า Opposition กล้ามเนื้อในกลุ่มนี้ที่สำคัญได้แก่

2.2.4.1 *Thenar eminence* เป็นกล้ามเนื้อหัวแม่มือเกาะที่ฝ่ามือ โดยเฉพาะที่ได้ฐานหัวแม่มือจะเห็นเป็นเนินชัดเจน ทำหน้าที่งอนิ้วหัวแม่มือ

2.2.4.2 Hypothenar eminence เป็นกล้ามเนื้อที่อยู่ใต้นิ้วก้อย ทำหน้าที่งอนิ้วก้อย

2.2.4.3 Dorsal interosseus เป็นกล้ามเนื้อที่กระดูกฝ่ามือชั้นที่ 1 และ 2 ผ่านมาเกาะที่นิ้วชี้ ทำหน้าที่กางนิ้วชี้และหมุนหัวแม่มือ

2.2.4.4 Abductor pollicis เกาะอยู่ที่ฐานของนิ้วหัวแม่มือ ทำหน้าที่งอนิ้วหัวแม่มือ

2.3 Arduino Microcontroller

2.3.1 Arduino

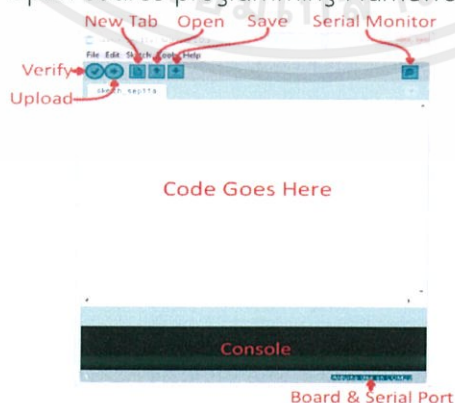
Arduino คือ แพลตฟอร์ม open-source ซึ่งมีส่วนที่เป็นทั้ง Software และ Hardware สำหรับการสร้าง Prototype หรืออุปกรณ์ต้นแบบของโปรเจกต์ที่เกี่ยวข้องกับอิเล็กทรอนิกส์และ Embedded System โดย Arduino นั้นถูกออกแบบมาเพื่อให้ง่ายต่อการใช้งาน เหมาะกับผู้เริ่มต้น และสามารถนำไปประยุกต์ใช้พัฒนาต่อยอดได้หลากหลาย

2.3.2 Arduino Board

บอร์ด PCB (Printed Circuit Board) ที่ถูกออกแบบมาเป็นพิเศษ โดยมีชิป Microcontroller ที่สามารถโปรแกรมสำหรับใช้ในการควบคุมหรือรับค่าจากอุปกรณ์ต่างๆผ่านพอร์ต Input และ Output การรับค่า Input เช่น การอ่านค่าจากเซ็นเซอร์ ส่วนการส่งงาน Output นั้น Arduino ก็ทำได้ เช่น การส่ง Pulse ไปควบคุมมอเตอร์ นอกจากนี้บอร์ด Arduino ยังสามารถเชื่อมต่อกับวงจรรีเลย์อิเล็กทรอนิกส์ต่างๆจากภายนอกได้ หรือเพื่อให้สะดวกต่อการใช้งาน ได้มีนักพัฒนาทำบอร์ดเสริมหรือ Arduino Shield ออกมา ซึ่งเป็นบอร์ดวงจรรีเลย์อิเล็กทรอนิกส์สำหรับทำงานเฉพาะอย่าง เช่น Ethernet Shield เป็น Shield ที่มีพอร์ต Ethernet ไว้สำหรับต่อ LAN เป็นต้น

2.3.3 Arduino IDE (Arduino Integrated Development Environment)

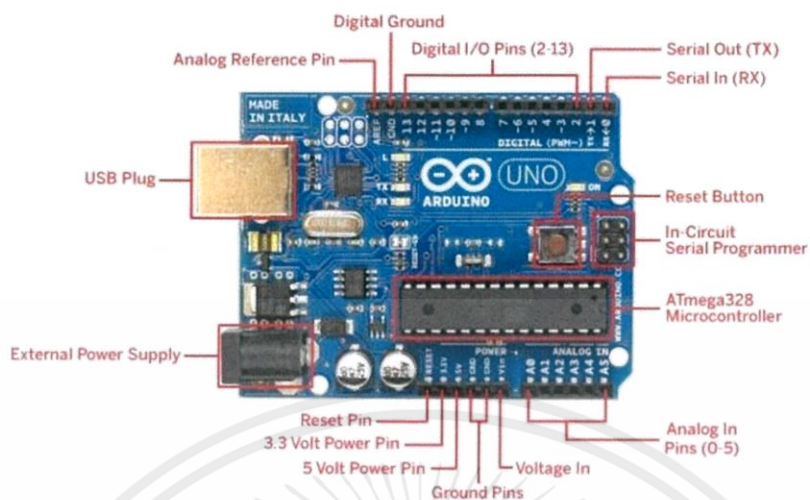
คือ Software ที่ใช้สำหรับเขียนโค้ด คอมไพล์และอัปโหลดโปรแกรมลงบอร์ด Arduino ซึ่งเป็นการบอกกับ Microcontroller ว่าต้องการจะให้ทำงานอะไร โดยในการเขียนโปรแกรมเพื่อควบคุม Arduino นั้นจะใช้ภาษา Arduino Programming Language ซึ่งมีรากฐานจากภาษา Wiring ซึ่งเป็น Open-source programming Framework สำหรับ Microcontroller



รูปที่ 2.2 โปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



รูปที่ 2.3 บอร์ด Arduino Uno R3

2.3.4.1 USB Plug สำหรับต่อกับคอมพิวเตอร์เพื่ออัปโหลดโปรแกรมและจ่ายไฟให้กับบอร์ด

2.3.4.2 Reset Button เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

2.3.4.3 ICSP Port ของ Atmega16 U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

2.3.4.4 I/O Port Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา TX,RX Serial, Pin 3,5,6,9,10 และ 11 เป็นขา PWM

2.3.4.5 ICSP Port:Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

2.3.4.6 MCU Atemega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

2.3.4.7 I/O Port Analog I/O ตั้งแต่ขา A0 ถึง A5

2.3.4.8 Power Port ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วย ขาไฟเลี้ยง +3.3V, +5V, GND, Vin

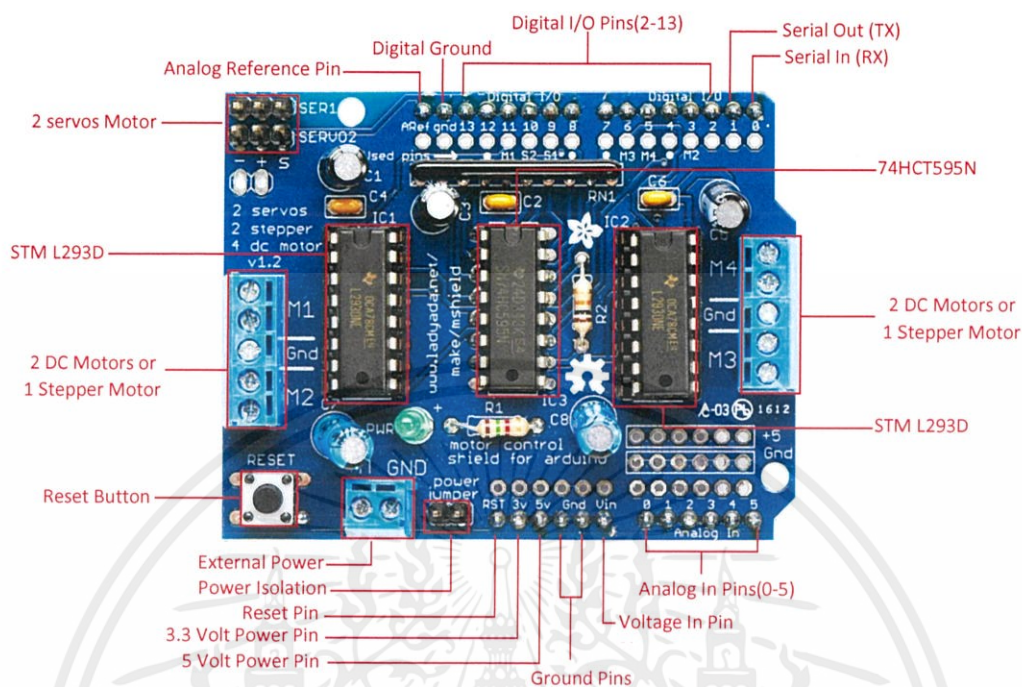
2.3.4.9 Power Jack รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

2.3.4.10 MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

2.3.4.11 Digital Ground จุดอ้างอิงสำหรับวัดระดับสัญญาณดิจิทัล

2.3.4.12 Ground Pin จุดอ้างอิงสำหรับวัดระดับสัญญาณอนาล็อก

2.4 Layout & Pin out Motor Shield Driver (Model: L293D)



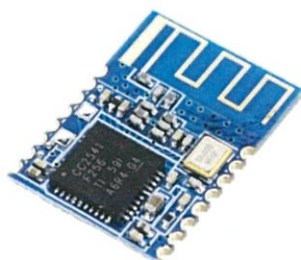
รูปที่ 2.4 บอร์ด Motor Shield Driver L293D

Motor Shield Driver (Model:L293D) เป็นบอร์ดที่ใช้สำหรับการขับมอเตอร์ โดยสามารถขับกระแสต่อเนื่องได้ไม่เกิน 600 mA รับแรงดันที่จ่ายให้กับมอเตอร์ได้ตั้งแต่ 4.5 ไปถึง 36 V สามารถเสียบลงบนบอร์ด Arduino Uno R3 เพื่อต่อใช้งานได้เลย โดยการทำงานจะใช้ IC L293D เป็นหลัก ซึ่งมีลักษณะเป็น H-Bridge สามารถขับมอเตอร์ประเภท DC Motor , Stepper Motor และ Servo Motor ได้โดยกำหนดทิศทางและความเร็วโดยการปรับค่า Pulse Width Modulation ส่วนหลักๆ ของบอร์ดจะประกอบไปด้วย

- 2.4.1 Reset Button ใช้เมื่อต้องการให้บอร์ดที่เชื่อมต่อกับ Motor driver ทำการ Reset
- 2.4.2 IC 74HCT595N ใช้ในการติดต่อรับส่งข้อมูลกับบอร์ดอาร์ดูโน้
- 2.4.3 IC L293D ใช้ในการขับมอเตอร์โดยสามารถกำหนดทิศทางและความเร็วของมอเตอร์ได้
- 2.4.4 I/O Port Digital I/O ตั้งแต่ขา D0 ถึง D13
- 2.4.5 I/O Port Analog I/O ตั้งแต่ขา A0 ถึง A5
- 2.4.6 External Power Terminal ใช้ต่อแหล่งจ่ายจากภายนอกโดยตรงเมื่อไม่ใช่ไฟเลี้ยงจากบอร์ด Arduino รับแรงดันได้ตั้งแต่ 4.5 V ถึง 36 V
- 2.4.7 Power Isolation จะมี Jumper ไว้สำหรับเชื่อมไฟเลี้ยงที่มาจากบอร์ด Arduino (หากต้องการใช้ไฟเลี้ยงจากภายนอกให้เอา Jumper ออก)
- 2.4.8 Digital Ground Pin จุดอ้างอิงสำหรับวัดระดับสัญญาณดิจิทัล
- 2.4.9 Analog Ground Pin จุดอ้างอิงสำหรับวัดระดับสัญญาณอนาล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 HM-11 Bluetooth Module



รูปที่ 2.5 HM-11 Bluetooth Module

HM-11 Bluetooth Module เป็นเทคโนโลยี Bluetooth 4.0 สามารถควบคุมผ่านคำสั่ง AT ซึ่งถูกส่งผ่านการเชื่อมต่อ UART แบบอนุกรม ใช้ชิป TL CC2541 ซึ่งใช้พลังงานต่ำทำให้ประหยัดไฟ และมีความสามารถใหม่ ๆ ที่ดีกว่า Bluetooth รุ่นก่อนหน้า สามารถวัดความแรงของสัญญาณออกมาคำนวณเป็นระยะห่างจากอุปกรณ์ได้ รองรับการเชื่อมต่อสำหรับ Android 4.3 ขึ้นไป สำหรับ IOS รองรับตั้งแต่ IOS 7 ขึ้นไป และยังสามารถนำมาใช้ร่วมกับอุปกรณ์ไมโครคอนโทรลเลอร์อย่างบอร์ดอาดูโนได้

คุณสมบัติเด่นของ HM-11 Bluetooth Module มีดังต่อไปนี้

2.5.1 เทคโนโลยี Bluetooth Low Energy 4.0

2.5.2 มีระยะในการเชื่อมต่อในสภาพแวดล้อมเปิด(ไม่มีสิ่งกีดขวาง) อยู่ที่ 30 เมตร

2.5.3 สามารถรับส่งข้อมูลได้โดยไม่จำกัดจำนวนไบต์

2.5.4 รองรับการใช้คำสั่ง AT command

2.5.5 สามารถทำงานเป็นได้ทั้ง Master (ใช้ในการควบคุมบลูทูธโมดูลอีกตัวได้) และ Slave

2.6 มอเตอร์ไฟฟ้ากระแสตรง

2.6.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

การทำงานของมอเตอร์ไฟฟ้ากระแสตรงเมื่อมีแรงดันและไฟฟ้ากระแสตรงเข้าไปในมอเตอร์ ไฟฟ้ากระแสตรงจะเข้าที่แปรงถ่านผ่านคอมมิวเตเตอร์ และเข้าไปในขดลวดอาร์เมเจอร์เพื่อสร้างสนามแม่เหล็กขึ้น และไฟฟ้ากระแสตรงอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) เพื่อสร้างสนามแม่เหล็กขั้วเหนือ ขั้วใต้ขึ้นจากสนามแม่เหล็กทั้งสองส่วนที่สร้างขึ้นนี้จะมีทิศทางเดียวกัน เกิดการผลักกันทำให้เกิดแรงบิดในขั้วอาร์เมเจอร์หรือโรเตอร์



รูปที่ 2.6 DC Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงที่ผลิตมาใช้งาน มีโครงสร้างและส่วนประกอบคล้ายกับเครื่องกำเนิดไฟฟ้ากระแสตรง มีส่วนประกอบที่สำคัญเหมือนกัน มีรูปร่างลักษณะภายนอกคล้ายกัน แตกต่างกันตรงการนำไปใช้งาน ส่วนมอเตอร์ไฟฟ้ากระแสตรงเมื่อจ่ายแรงดันไฟตรงให้มอเตอร์ ทำให้มอเตอร์หมุนเกิดพลังกลขึ้นมา ส่วนประกอบหลักๆของมอเตอร์ไฟฟ้ากระแสตรง ประกอบด้วยส่วนต่างๆ ดังนี้

2.6.2.1 ขดลวดสนามแม่เหล็ก (Field Coil) คือขดลวดที่ถูกพันอยู่กับขั้วแม่เหล็กที่ยึดติดกับโครงมอเตอร์ ทำหน้าที่กำเนิดขั้วแม่เหล็กขั้วเหนือ (N) และขั้วใต้ (S) แทนแม่เหล็กถาวรขดลวดที่ใช้เป็นขดลวดอาบนํ้ายาคนวน สนามแม่เหล็กจะเกิดขึ้นเมื่อจ่ายแรงดันไฟตรงให้มอเตอร์

2.6.2.2 ขั้วแม่เหล็ก (Pole Pieces) คือแกนสำหรับรองรับขดลวดสนามแม่เหล็กถูกยึดติดกับโครงมอเตอร์ด้านใน ขั้วแม่เหล็กทำมาจากแผ่นเหล็กอ่อนบางๆ อัดซ้อนกัน (Lamination Sheet Steel) เพื่อลดการเกิดกระแสไหลวน (Edy Current) ที่จะทำให้ความเข้าของสนามแม่เหล็กลดลง ขั้วแม่เหล็กทำหน้าที่ให้กำเนิดขั้วสนามแม่เหล็กมีความเข้มสูงสุด แทนขั้วสนามแม่เหล็กถาวร ผิวด้านหน้าของขั้วแม่เหล็กทำให้โค้งรับกับอาร์เมเจอร์พอดี

2.6.2.3 โครงมอเตอร์ (Motor Frame) คือส่วนเปลือกหุ้มภายนอกของมอเตอร์ และยึดส่วนอยู่กับที่ (Stator) ของมอเตอร์ไว้ภายในร่วมกับฝาปิดหัวท้ายของมอเตอร์ โครงมอเตอร์ทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กระหว่างขั้วแม่เหล็กให้เกิดสนามแม่เหล็กครบวงจร

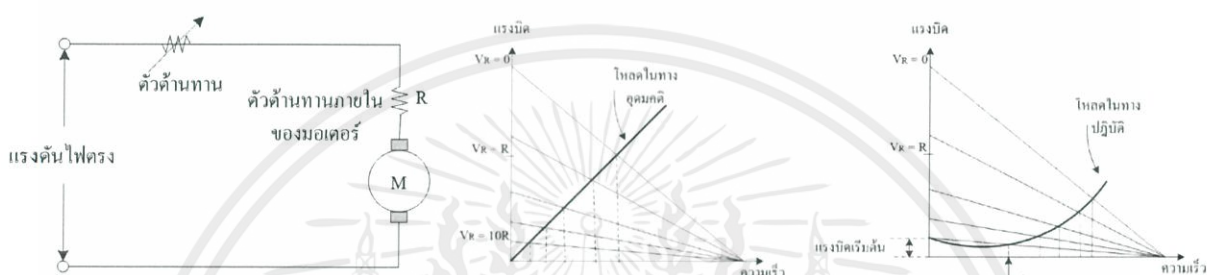
2.6.2.4 อาร์เมเจอร์ (Armature) คือส่วนเคลื่อนที่ (Rotor) ถูกยึดติดกับเพลา (Shaft) และรองรับการหมุนด้วยที่รองรับการหมุน (Bearing) ตัวอาร์เมเจอร์ทำจากเหล็กแผ่นบางๆ อัดซ้อนกัน ถูกเจาะร่องออกเป็นส่วนๆ เพื่อไว้พันขดลวดอาร์เมเจอร์ (Armature Winding) ขดลวดอาร์เมเจอร์เป็นขดลวดอาบนํ้ายาคนวน ร่องขดลวดอาร์เมเจอร์จะมีขดลวดพันอยู่และมีลิ้มไฟเบอร์อัดแน่นยึดขดลวดอาร์เมเจอร์ไว้ ปลายขดลวดอาร์เมเจอร์ต่อกับคอมมิวเตเตอร์ อาร์เมเจอร์ผลัดกันของสนามแม่เหล็กทั้งสอง ทำให้อาร์เมเจอร์หมุนเคลื่อนที่

2.6.2.5 คอมมิวเตเตอร์ (Commutator) คือส่วนเคลื่อนที่อีกส่วนหนึ่ง ถูกยึดติดเข้ากับอาร์เมเจอร์และเพลาร่วมกัน คอมมิวเตเตอร์ทำจากแท่งทองแดงแข็งประกอบเข้าด้วยกันเป็นรูปทรงกระบอก แต่ละแท่งทองแดงของคอมมิวเตเตอร์ถูกแยกออกจากกันด้วยฉนวนไมก้า (Mica) อาร์เมเจอร์ คอมมิวเตเตอร์ทำหน้าที่เป็นขั้วรับแรงดันไฟตรงที่จ่ายมาจากแปรงถ่าน เพื่อส่งไปให้ขดลวดอาร์เมเจอร์

2.6.2.6 แปรงถ่าน (Brush) คือตัวสัมผัสกับคอมมิวเตเตอร์ ทำเป็นแท่งสี่เหลี่ยมผลิตมาจากคาร์บอนหรือแกรไฟต์ผสมผงทองแดง เพื่อให้แข็งและนำไฟฟ้าได้ดี มีสายตัวนำต่อร่วมกับแปรงถ่านเพื่อไปรับแรงดันไฟตรงที่จ่ายเข้ามา แปรงถ่านทำหน้าที่รับแรงดันไฟตรงจกแหล่งจ่าย จ่ายผ่านไปให้คอมมิวเตเตอร์

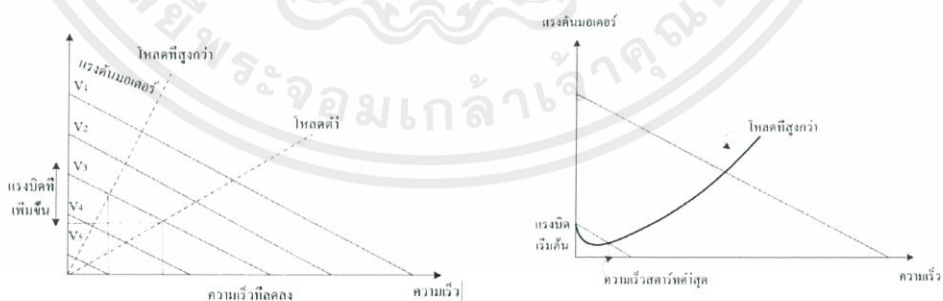
2.6.3 การควบคุมความเร็วของมอเตอร์ขั้นพื้นฐาน

2.6.3.1 การควบคุมด้วยตัวต้านทานที่ปรับค่าได้ เป็นรูปแบบพื้นฐานที่สุดของการควบคุมมอเตอร์คือ ใช้ตัวต้านทานปรับค่าได้อนุกรมกับมอเตอร์ โดยตัวต้านทานที่ปรับค่าได้จะเป็นตัวกำหนดความเร็วในการหมุนของมอเตอร์การบังคับแบบนี้ไม่มีประสิทธิภาพเพราะกำลังไฟสูญเสียไปในตัวความต้านทาน มักนิยมใช้กับมอเตอร์ตัวเล็กๆ การบังคับแบบนี้ให้คุณสมบัติการสตาร์ทดี (ให้แรงบิดสูงที่ความเร็วต่ำ) แต่จะให้ความเร็วสูงมากเมื่อมอเตอร์อยู่ในภาวะที่มีโหลดน้อยๆ การบังคับแบบนี้มีประโยชน์เฉพาะภาวะที่แรงต้านคงที่ เช่น การบังคับความเร็วของเครื่องจักรเย็บผ้า เป็นต้น



รูปที่ 2.7 วงจรควบคุมความเร็วของมอเตอร์แบบใช้ตัวต้านทานอนุกรม

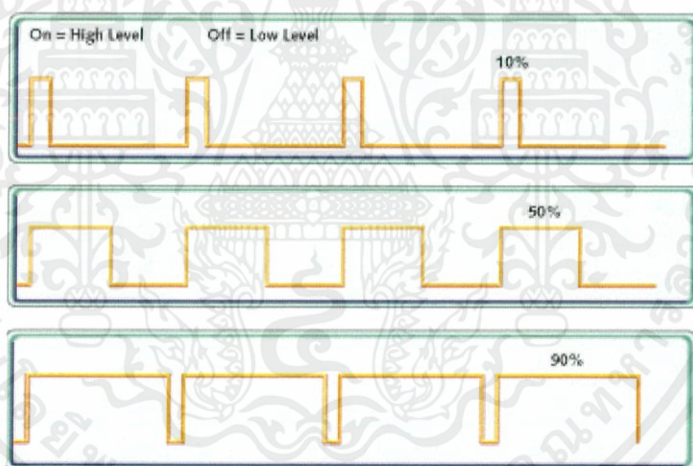
2.6.3.2 การควบคุมด้วยวิธีเปลี่ยนค่าแรงดัน วิธีการนี้ดีกว่าวิธีการแรกแต่จะซับซ้อนกว่าต้องใช้อุปกรณ์อิเล็กทรอนิกส์ที่มีอัตราขยายกำลังสูง และมอเตอร์จะถูกป้อนด้วยแรงดันที่เปลี่ยนแปลงค่าได้จากแหล่งจ่ายที่มีอิมพีแดนซ์ต่ำ ข้อดีของการควบคุมวิธีนี้คือ ถ้าความเร็วลดลงจากผลของแรงบิด แรงดันที่ป้อนให้กับมอเตอร์จะเพิ่มขึ้นเพื่อรักษาระดับความเร็ว ส่วนข้อเสียจากการควบคุมวิธีนี้คือ เมื่อมอเตอร์มีความเร็วต่ำแรงดันที่ป้อนให้กับมอเตอร์จะมีค่าต่ำเช่นกัน



รูปที่ 2.8 การควบคุมความเร็วมอเตอร์โดยเปลี่ยนค่าแรงดัน

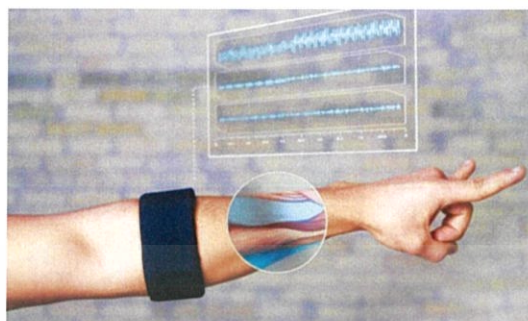
2.6.3.3 การควบคุมแบบ PWM (Pulse Width Modulation) คือ เทคนิคสำหรับควบคุมวงจรถานด้านฮาร์ดแวร์โดยใช้สัญญาณเอาต์พุตแบบดิจิทัลของไมโครโปรเซสเซอร์ควบคุม PWM คือเทคนิคการส่งสัญญาณแบบสวิตช์ หรือ ส่งค่าดิจิทัล 0-1 โดยให้สัญญาณความถี่คงที่ การควบคุมระยะเวลาสัญญาณสูงและสัญญาณต่ำที่ต่างกัน ก็จะทำให้ค่าแรงดันเฉลี่ยของสัญญาณสวิตช์ต่างกันได้ สำหรับโมดูล PWM ของ Arduino มีความละเอียด 8 bit หรือ ปรับได้ 255 ระดับ ดังนั้นค่าสัญญาณ 0 โวลต์ถึง 5 โวลต์ จะถูกแสดงเป็นสัญญาณแบบดิจิทัลได้ 0 ถึง 255 ซึ่งเราสามารถเทียบสัดส่วนคำนวณจากเลขจริงเป็นเลขทางดิจิทัลได้

- โดย 10a แสดงสัญญาณ PWM ที่ 10% duty cycle คือ มีช่วงสัญญาณในการ on จะเป็น 10% ของคาบสัญญาณ และ จะมีช่วงสัญญาณ off เป็น 90% ของคาบสัญญาณ
- โดย 10b แสดงสัญญาณ PWM ที่ 50% duty cycle คือ มีช่วงสัญญาณในการ on จะเป็น 50% ของคาบสัญญาณ และ จะมีช่วงสัญญาณ off เป็น 50% ของคาบสัญญาณ
- โดย 10c แสดงสัญญาณ PWM ที่ 90% duty cycle คือ มีช่วงสัญญาณในการ on จะเป็น 90% ของคาบสัญญาณ และ จะมีช่วงสัญญาณ off เป็น 10% ของคาบสัญญาณ



รูปที่ 2.9 แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycle ที่ต่างกันได้

2.7 MYO Gesture Control Armband



รูปที่ 2.10 MYO Gesture Control Armband

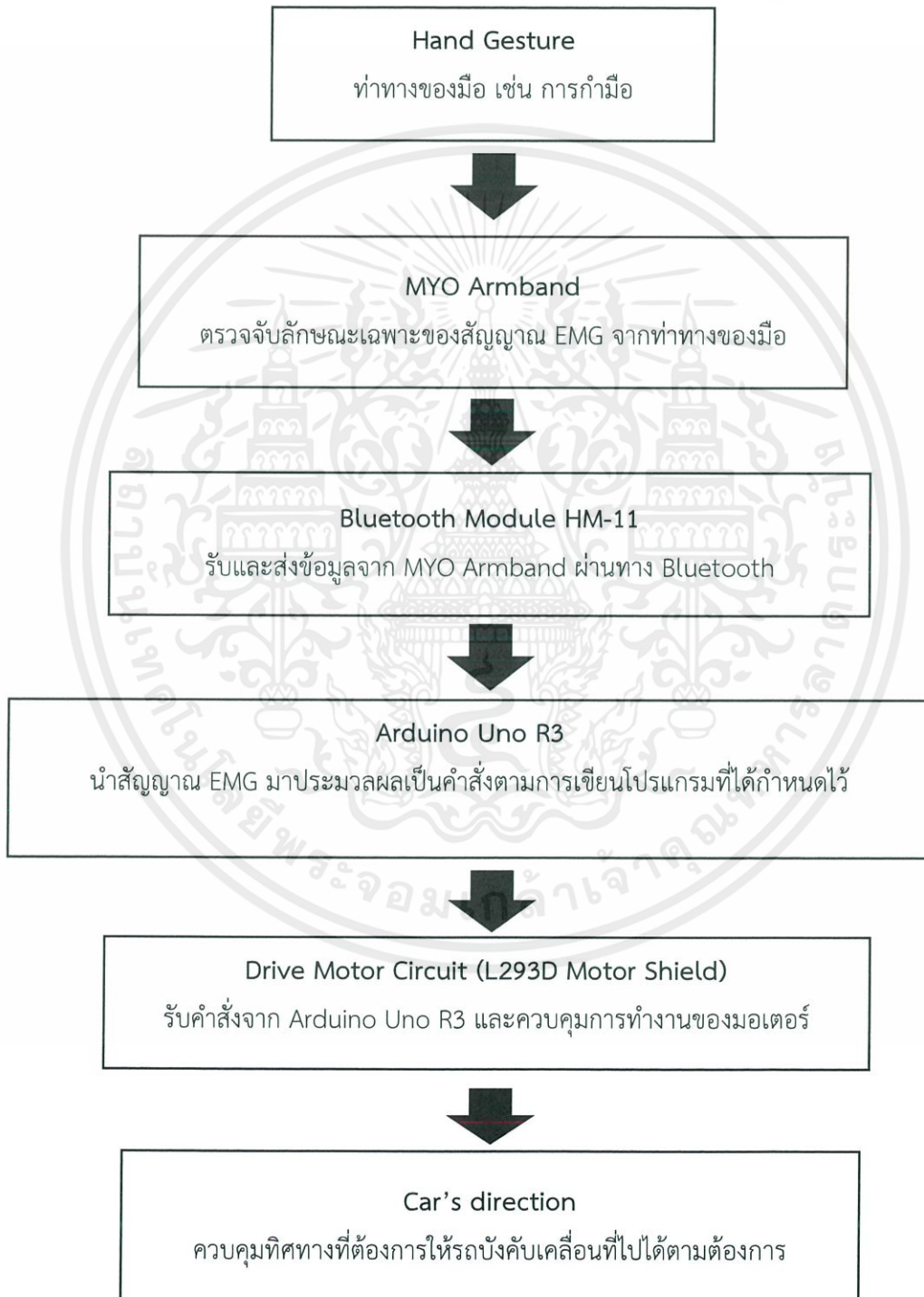
MYO Gesture Control Armband เป็นอุปกรณ์ EMG Sensor ที่พัฒนาขึ้นโดยบริษัท Thalmic Labs ตัวปลอกแขนนี้ประกอบไปด้วย EMG Sensor จำนวนทั้งหมด 8 Channel ในแต่ละ Channel มีอิเล็กโทรดชนิด Surface เพื่อใช้ในการตรวจจับการเปลี่ยนแปลงของสัญญาณไฟฟ้ากล้ามเนื้อได้ ซึ่งใช้หลักการที่เรียกว่า Electromyography เพราะว่าเวลาที่กล้ามเนื้อมีการขยับหรือเคลื่อนไหว ก็จะเกิดจากการที่สมองสั่งงานผ่านเส้นประสาท เพื่อให้กล้ามเนื้อขยับตามต้องการ ซึ่งตรงนี้ก็ถือเป็นสัญญาณไฟฟ้านั่นเอง จึงทำให้สามารถตรวจจับคลื่นสัญญาณดังกล่าวได้ เพื่อวิเคราะห์การทำงานของกล้ามเนื้อ ซึ่งเดิมที่เป็นวิธีที่นิยมใช้ในทางการแพทย์เพื่อวิเคราะห์โรคภัยไข้เจ็บ แต่ในปัจจุบันเรื่องของ Wearable Device ก็ได้เริ่มเข้ามามีบทบาทมากขึ้นทำให้การตรวจวัดสัญญาณเป็นไปโดยสะดวกและรวดเร็วมากขึ้น

MYO Gesture Control Armband MYO สามารถเชื่อมต่อกับอุปกรณ์ต่าง ๆ ได้ผ่านทางสัญญาณไร้สาย Bluetooth หลักๆถูกออกแบบมาสำหรับใช้งานกับคอมพิวเตอร์ได้ทั้งบนระบบปฏิบัติการ Windows และ Mac เพื่อใช้งานในการสั่งงานต่างๆ ไม่ว่าจะเป็นเล่นอินเทอร์เน็ต ฟังเพลง นำเสนองาน หรือเล่นเกม และอื่นๆ อีกมากมาย

บทที่ 3

วิธีดำเนินการวิจัย

3.1 Block Diagram



รูปที่ 3.1 Block Diagram กระบวนการรับสัญญาณ EMG นำไปควบคุมมอเตอร์

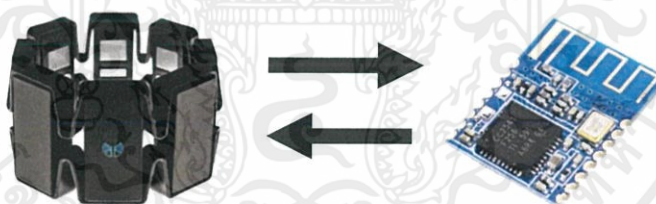
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเชื่อมต่อ MYO Armband กับ Bluetooth HM-11



รูปที่ 3.2 MYO Armband และ Bluetooth Dongle 4.0

MYO Armband มีวิธีการเชื่อมต่อกับระบบปฏิบัติการอาทิเช่น Windows และ Mac ผ่านทางการเชื่อมต่อไร้สายบลูทูธซึ่งในการเชื่อมต่อจำเป็นต้องใช้ Bluetooth Dongle 4.0 ที่ทางผู้พัฒนา Thalmics Lab ได้ผลิตออกมาโดยเฉพาะสำหรับ MYO Armband เสียบเข้ากับเครื่องคอมพิวเตอร์เพื่อเริ่มต้นใช้งาน MYO Armband แต่เนื่องจากอุปกรณ์ที่ทางผู้จัดทำได้ใช้ในโครงงานนี้ คือ Bluetooth Module HM-11 และ Arduino Uno R3 ซึ่งในการใช้งานจริงไม่สามารถนำมาประยุกต์ใช้กับ Bluetooth Dongle 4.0 ได้ ผู้จัดทำจึงต้องทำการเชื่อมต่อ MYO Armband เข้ากับ Bluetooth Module HM-11 โดยตรง คือ เมื่อทำการเปิดสวิตซ์ทั้ง Myo armband และ Bluetooth Module HM-11 ทั้งสองอุปกรณ์ก็จะเชื่อมต่อกันผ่านทางบลูทูธโดยอัตโนมัติ

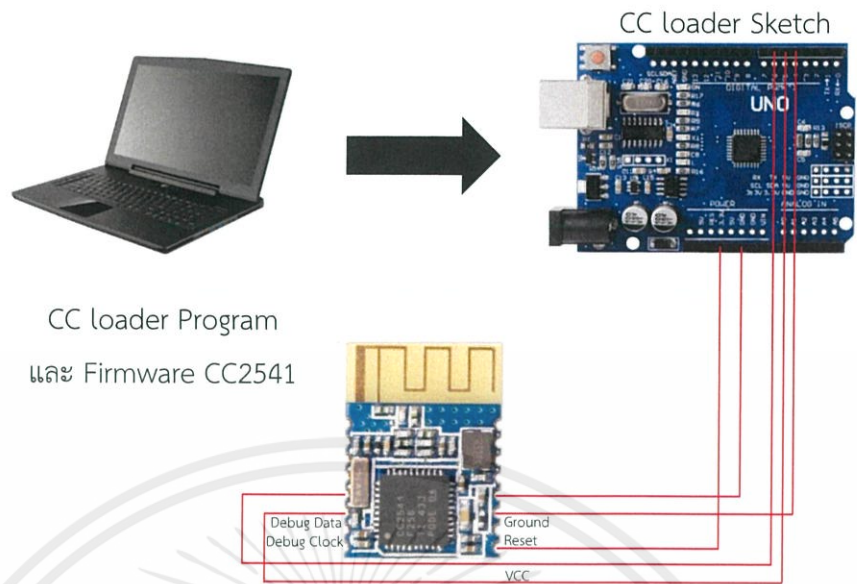


รูปที่ 3.3 MYO ARMBAND และ Bluetooth Module HM-11

ในการเชื่อมต่อ MYO Armband กับ Bluetooth Module HM-11 จำเป็นต้องอัปเดตเฟิร์มแวร์ของ Bluetooth Module HM-11 ใหม่เสียก่อน เนื่องจาก MYO Armband ไม่สามารถเชื่อมต่อได้เหมือนกับอุปกรณ์ทั่วไป ผู้จัดทำได้นำ Firmware ที่ถูกพัฒนาขึ้นโดยผู้พัฒนา Valentin Roland สำหรับใช้ในการ Flash Bluetooth Module HM-11 เพื่อให้เชื่อมต่อและรับส่งข้อมูลกับ MYO Armband ได้โดยตรงอุปกรณ์ที่ใช้ในการ Flash Bluetooth Module HM-11 มีดังต่อไปนี้

- Bluetooth Module HM-11
- Arduino Uno R3
- Firmware CC2541
- CC Loader Program

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การเชื่อมต่อวงจรสำหรับการ Flash HM-11

ขั้นตอนการ Flash HM-11 ด้วยโปรแกรม CCloder

1. ต่อยังวงจรดังรูปที่ 3.3 และอัปโหลดโค้ด CCloder.ino ลงในบอร์ด Arduino Uno R3
2. เปิดหน้าต่าง Command Prompt ขึ้นมา และพิมพ์ที่อยู่ของไฟล์โปรแกรม CCloder ลงไป

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\HIFI>cd /d E:\Final Project 4C\CCloader-master\CCloader-master\Windows
E:\Final Project 4C\CCloader-master\CCloader-master\Windows>CCloader.exe
  
```

รูปที่ 3.5 การสั่งเปิดโปรแกรม CCloder.exe ใน Command Prompt

3. ทำการเลือกพอร์ตของ Arduino Uno R3 ที่เชื่อมต่ออยู่กับคอมพิวเตอร์ , ที่อยู่ของไฟล์ Firmware CC2541 และ ประเภทของบอร์ด Arduino (ในที่นี้ใช้ Arduino Uno R3 จึงเลือกเป็นประเภทที่ 0)

```

Command Prompt
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\HIFI>cd /d E:\Final Project 4C\CCloader-master\CCloader-master\Windows
E:\Final Project 4C\CCloader-master\CCloader-master\Windows>CCloader.exe
Copyright (c) 2013 RedBearLab.com
CCloader.exe version 0.5
Invalid parameters.
Usage: CCloder.exe <com number> <bin file> <device>
Example: CCloder.exe 2 abc.bin 0
<device>: 0 — Default (e.g. UNO)
          1 — Leonardo
  
```

รูปที่ 3.6 การ import Firmware ในโปรแกรม CCloder.exe ผ่าน Command Prompt

4. โปรแกรมจะเริ่มทำงานและเชื่อมต่อกับ HM-11 Bluetooth Module ผ่าน Arduino Uno R3 และทำการ Flash Firmware CC2541 ลงไป

```

File open!
**WARNING: File's size isn't the integer multiples of 512 bytes, and
            the last block will be filled in up to 512 bytes with 0xFF!
Block total: 274

Enable transmission...
Request sent already!
/*****
* If there is no respond last for 3s, please press "Ctrl+C" to exit!
* And pay attention to :
*   1. The connection between computer and Arduino;
*   2. The connection between Arduino and CC2540;
*   3. Whether the device you using is Leonardo or not;
*   4. Other unexpected errors.
*****/

Waiting for respond from Arduino...

Uploading firmware...

1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
3  24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
3  44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
3  64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
3  84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102
  103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
  119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134
  135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
  151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
  167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182
  183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
  199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214
  215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230
  231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246
  247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262
  263 264 265 266 267 268 269 270 271 272 273 274 Upload successfull
y!
File closed!
Comport closed!

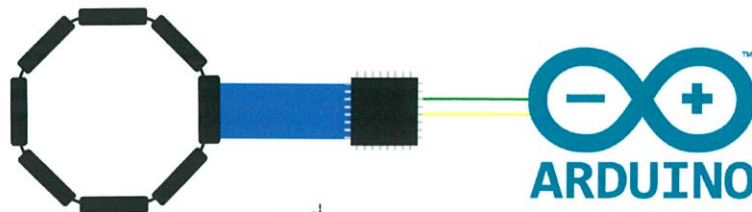
```

รูปที่ 3.7 โปรแกรม CCloder flash Firmware CC2541 ลงใน Bluetooth HM-11

3.3 การรับข้อมูลท่าทางจาก MYO Gesture Control Armband

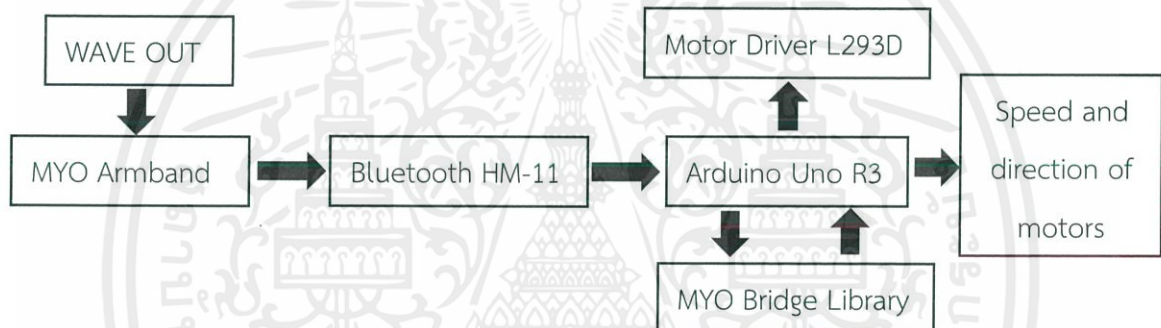
คุณสมบัติต่างๆที่ MYO Gesture Control Armband สามารถตรวจจับได้จากการสวมใส่ของผู้ใช้งานประกอบไปด้วย สัญญาณ EMG(Electromyography signal) , Motion Sensor ทั้งหมด 9 แกน ซึ่งประกอบไปด้วย Accelerometer (3 แกน) Gyroscope (3 แกน) , Magnetometer (3 แกน) และการตรวจจับท่าทางของมือ โดยเดิมทีแล้วทางผู้พัฒนา Thalmic labs ได้มีการใส่ข้อมูลการตรวจจับท่าทางของมือไว้จำนวน 6 ท่าทางด้วยกัน แต่จะสามารถนำมาใช้ได้ก็ต่อเมื่อทำการเชื่อมต่อกับคอมพิวเตอร์ผ่านซอฟต์แวร์ MYO connect ที่ทางผู้พัฒนา Thalmiclabs ได้เขียนขึ้น การนำท่าทางของมือทั้ง 6 ท่าทางมาประยุกต์ใช้กับโครงงานนี้ ผู้จัดทำได้เลือกใช้ Library MYO bridge ของผู้พัฒนา Valentin Roland มาทำการปรับปรุงและแก้ไขจากเดิม โดยการเขียนคำสั่งโปรแกรมลงใน Arduino IDE ซึ่งคำสั่งหลักๆที่ใช้งานมีดังต่อไปนี้

MyoBridge bridge(bridgeSerial); คำสั่งเปิดการเชื่อมต่อระหว่าง MYO กับ Bluetooth HM-11
 bridge.enablePoseData(); คำสั่งอนุญาตให้ส่งข้อมูลท่าทางจาก MYO Bluetooth ไปที่ HM-11
 bridge.setPoseEventCallBack(handlePoseData); คำสั่ง hold ข้อมูลนั้นๆไว้ขณะที่ทำท่าทาง



รูปที่ 3.8 MYO Bridge

หลักการในการเขียนโปรแกรมใน Arduino IDE ให้สามารถรับข้อมูลท่าทางจาก MYO ไปทำการควบคุมมอเตอร์ให้เคลื่อนไหวได้ คือ ผู้จัดทำเขียนโปรแกรมให้ Arduino Uno R3 รับค่าข้อมูลท่าทางจาก MYO ไปเก็บไว้ในตัวแปรตัวหนึ่งแล้วนำตัวแปรนั้นไปเปรียบเทียบกับข้อมูลของผู้พัฒนา Thalmic labs ที่ได้มีการใส่ข้อมูลท่าทางของมือไว้จำนวน 6 ท่าทาง ได้แก่ WAVE IN , WAVE OUT , DOUBLE TAP , FINGER SPREAD , FIST และ REST ถ้าหากข้อมูลที่ได้รับมาตรงกับท่าทางของมือทั้ง 6 ท่าทางอันใดอันหนึ่งทีกล่าวมา ก็จะเริ่มทำคำสั่งที่เขียนไว้สำหรับควบคุมมอเตอร์แต่ละตัว ดังรูปที่ 3.8



รูปที่ 3.9 การเปรียบเทียบท่าทางที่ตรวจจับจาก MYO กับ MYO Bridge Library

```

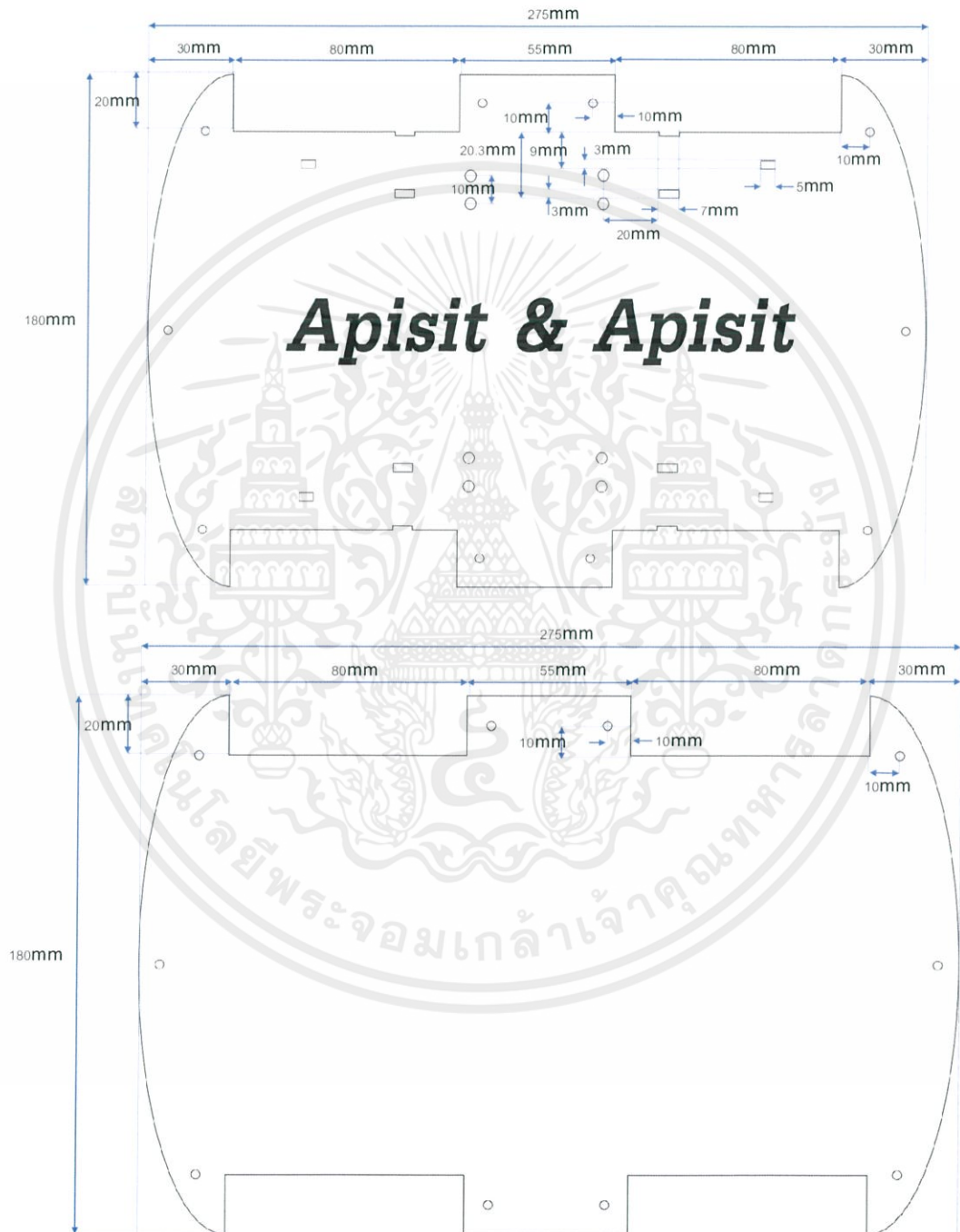
MyoPose pose;
pose = (MyoPose)data.pose;
if (bridge.poseToString(pose) == "MYO_POSE_DOUBLE_TAP")
{
  i++;
}
if (i%2 != 0)
{
  if (bridge.poseToString(pose) == "MYO_POSE_REST")
  {
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
    motor4.run(RELEASE);
  }
  else if (bridge.poseToString(pose) == "MYO_POSE_WAVE_IN")
  {
    motor3.run(RELEASE);
    motor4.run(RELEASE);
    motor2.run(FORWARD);
    motor1.run(FORWARD);
    motor1.setSpeed(250);
    motor2.setSpeed(250);
  }
}
  
```

รูปที่ 3.10 ตัวอย่างโค้ดอาณูโนที่ใช้ในการเปรียบเทียบสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

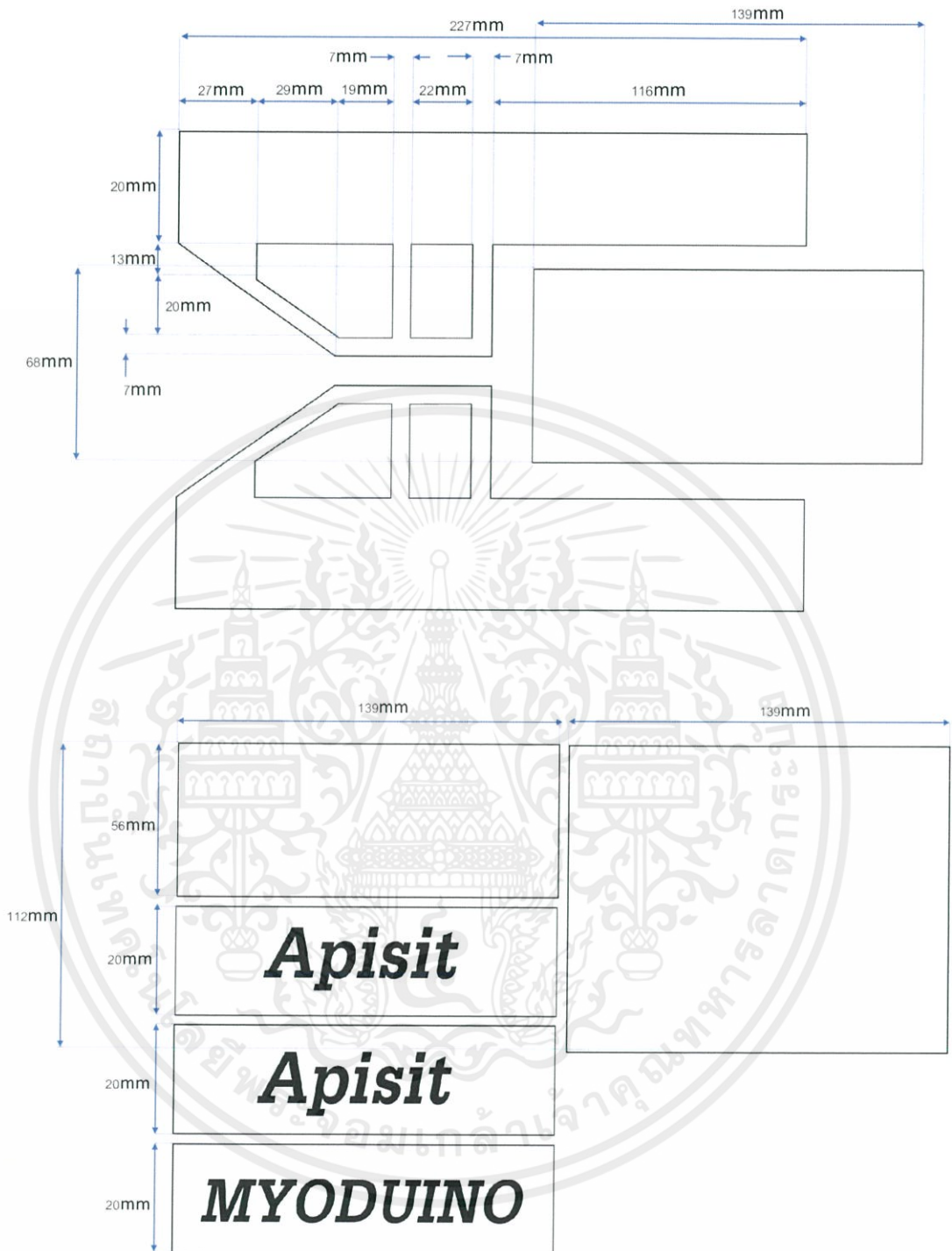
3.4 โครงร่างของตัวรถหุ่นยนต์

การออกแบบโครงร่างของตัวรถหุ่นยนต์ใช้โปรแกรม Illustrator ในการออกแบบ และวัสดุที่นำมาใช้ทำโครงรถเป็นอะคริลิกเพื่อไม่ให้ตัวรถมีน้ำหนักมากจนเกินไป โดยมีการออกแบบให้เป็นสองชั้นเพื่อให้สามารถใส่อุปกรณ์ต่างๆ เช่น แบตเตอรี่ บอร์ด PCB



รูปที่ 3.11 โครงรถส่วนบนและล่างที่ออกแบบในโปรแกรม Illustrator

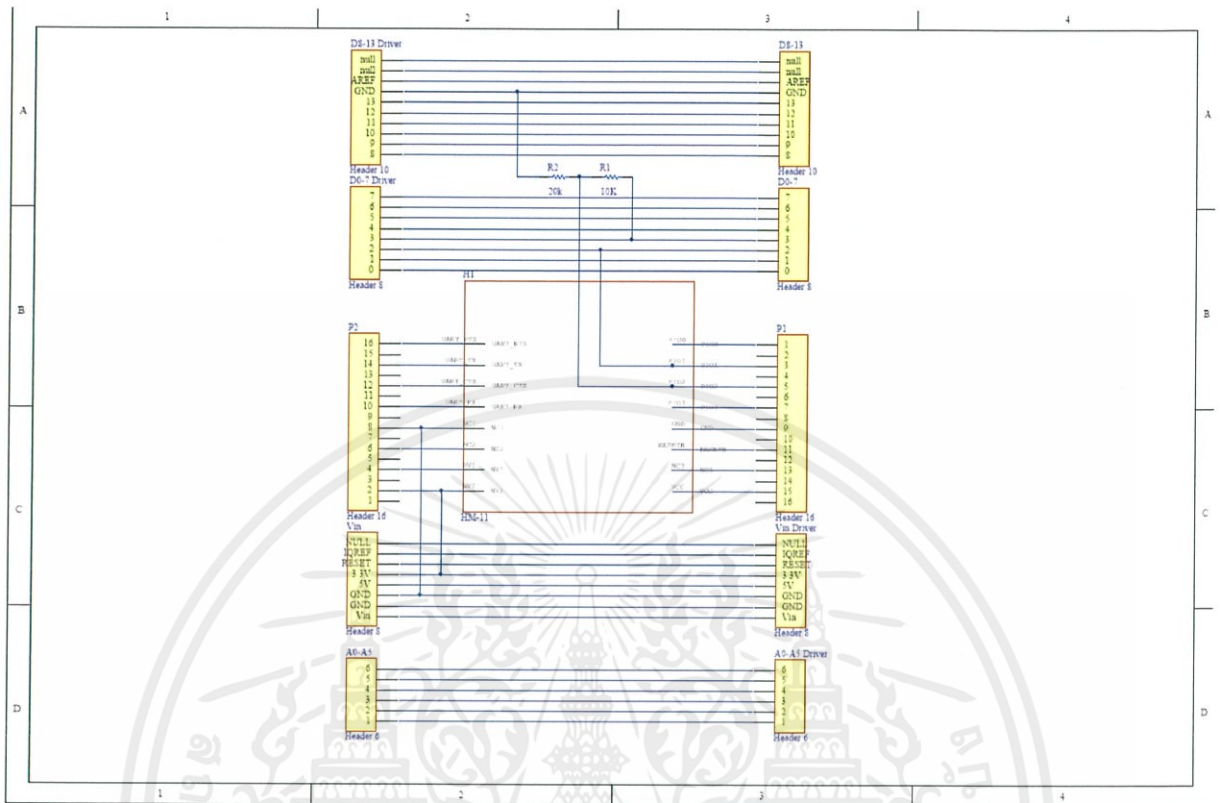
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



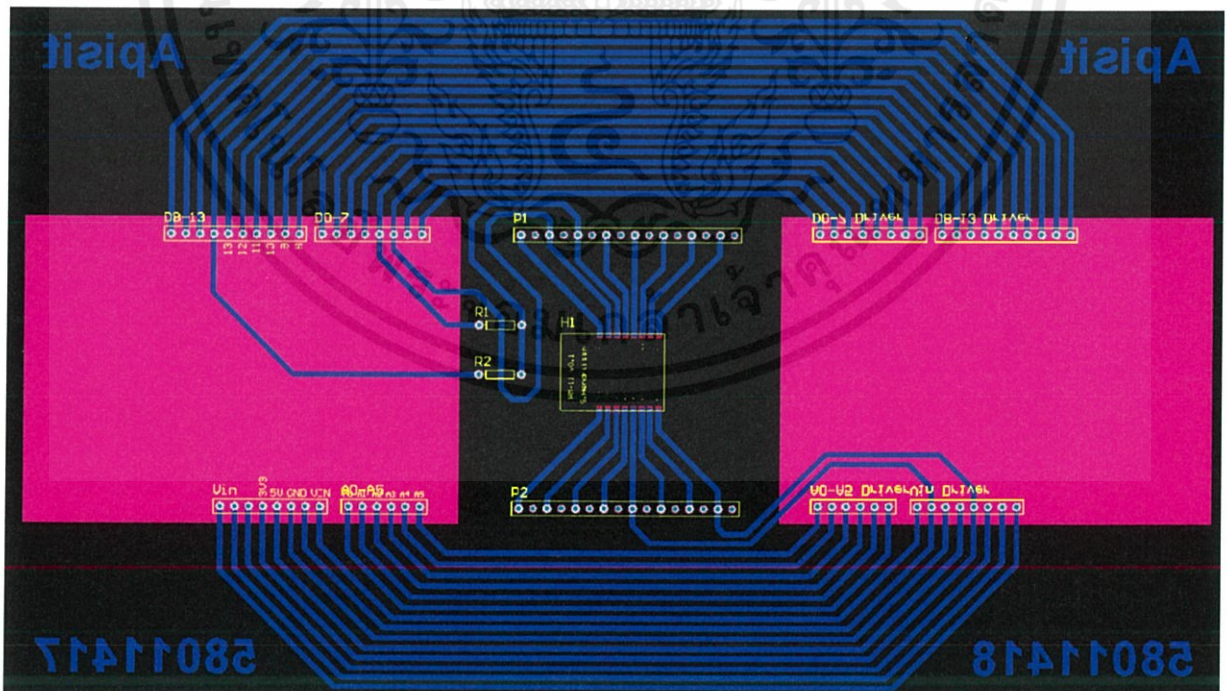
รูปที่ 3.12 โครงร่างส่วนบริเวณตัวรถที่ออกแบบในโปรแกรม Illustrator

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Schematic and PCB



รูปที่ 3.13 PCB



รูปที่ 3.14 Schematic

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 รถหุ่นยนต์



รูปที่ 3.15 รถหุ่นยนต์และวงจรภายใน

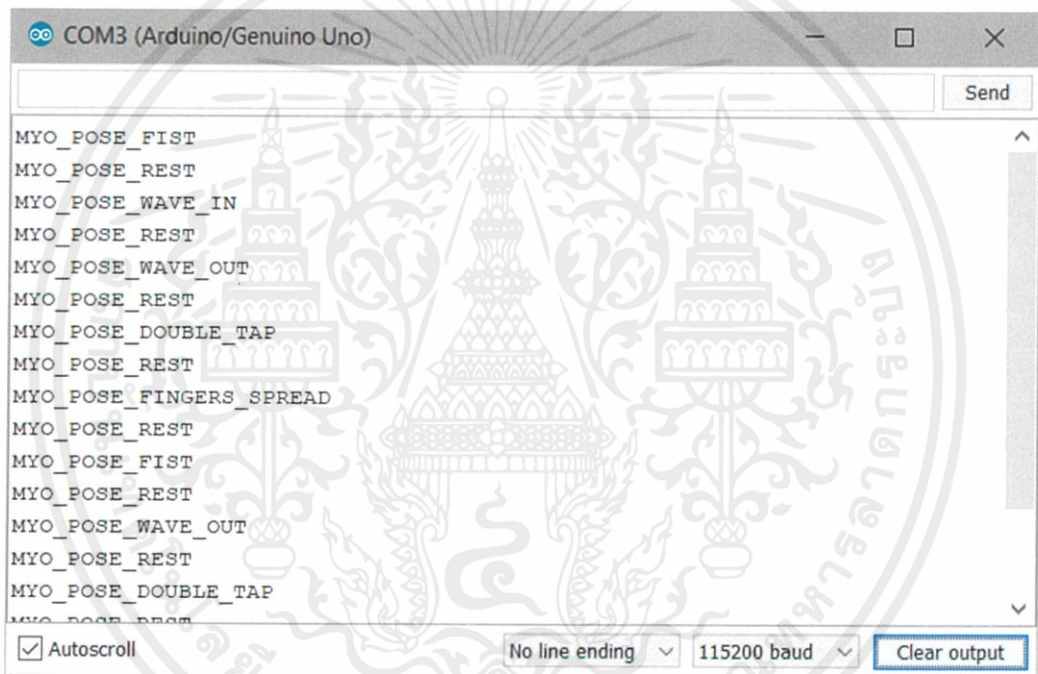
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดสอบการตรวจจับท่าทางของมือด้วย MYO Armband

ในการทดสอบการรับส่งข้อมูลระหว่าง MYO armband ซึ่งมี EMG SENSOR ทั้งหมด 8 ตัวกับ Bluetooth module HM-11 รับข้อมูลมาประมวลผลผ่าน Arduino UNO R3 โดยการเขียนโค้ดในโปรแกรม Arduino IDE เพื่อดึงข้อมูลของสัญญาณส่วนที่ต้องการนำไปใช้ออกมาจาก MYO Armband และตรวจสอบผลที่ได้ผ่านทาง Serial Monitor ในโปรแกรม Arduino IDE



รูปที่ 4.1 Serial monitor แสดงชื่อท่าทางที่ MYO Gesture สามารถตรวจจับได้

ท่าทางพื้นฐานที่ MYO Gesture Armband สามารถตรวจจับได้มีอยู่ทั้งหมดจำนวน 6 ท่าทาง ได้แก่ WAVE IN (โบกมือเข้าหาตัว) , WAVE OUT (โบกมือออกจากตัว) , DOUBLE TAP (นิ้วโป้งและนิ้วชี้แตะกันสองครั้ง) , FINGER SPREAD (กางมือ) , FIST (กำมือ) และ REST (ไม่เกร็งมือหรือไม่แสดงท่าทางใดๆ) ซึ่งพบว่าการเปลี่ยนจากท่าทางหนึ่งไปยังอีกท่าทางหนึ่งในทันทีจะทำให้การตรวจจับท่าทางที่ได้มีความคลาดเคลื่อน เช่น การเปลี่ยนจากท่า FIST เป็นท่าทาง FINGER SPREAD ท่าทางที่ตรวจจับได้กลับเป็น WAVE OUT แล้วจึงค่อยกลายเป็น FINGER SPREAD ส่งผลให้คำสั่งที่นำไปควบคุมมอเตอร์เคลื่อนไหวผิดเพี้ยนจากที่ต้องการ







เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ระยะเวลาเชื่อมต่อระหว่าง MYO Armband กับรถหุ่นยนต์

ระยะที่ยังสามารถควบคุมรถหุ่นยนต์ด้วย MYO Armband ได้มีระยะทางไม่เกิน 15 เมตร หากเกินระยะนี้รถหุ่นยนต์จะหยุดการทำงานและรองจนกว่าจะมีการเชื่อมต่อสัญญาณบลูทูธจาก MYO Armband อีกครั้ง จึงจะเริ่มเคลื่อนที่ได้

4.3 ผลการทดสอบการหมุนของมอเตอร์ตามท่าทางการเคลื่อนไหวของมือ

ตารางที่ 4.1 ผลการทดสอบการหมุนของมอเตอร์ตามท่าทางการเคลื่อนไหวของมือ







ท่าทางของมือ	ทิศทางการหมุนของมอเตอร์แต่ละตัว			
	มอเตอร์ตัวที่ 1	มอเตอร์ตัวที่ 2	มอเตอร์ตัวที่ 3	มอเตอร์ตัวที่ 4
 REST	ไม่หมุน	ไม่หมุน	ไม่หมุน	ไม่หมุน
 WAVE IN	หมุนตามเข็มนาฬิกา	หมุนตามเข็มนาฬิกา	ไม่หมุน	ไม่หมุน
 WAVE OUT	ไม่หมุน	ไม่หมุน	หมุนทวนเข็มนาฬิกา	หมุนทวนเข็มนาฬิกา
 FINGER SPREAD	หมุนทวนเข็มนาฬิกา	หมุนทวนเข็มนาฬิกา	หมุนตามเข็มนาฬิกา	หมุนตามเข็มนาฬิกา
 FIST	หมุนตามเข็มนาฬิกา	หมุนตามเข็มนาฬิกา	หมุนทวนเข็มนาฬิกา	หมุนทวนเข็มนาฬิกา
 *DOUBLE TAP	-	-	-	-

* DOUBLE TAP เป็นท่าที่ใช้ในการเริ่มต้นหรือหยุดรับส่งข้อมูลท่าทางของมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบการเคลื่อนไหวของรถหุ่นยนต์ตามท่าทางการเคลื่อนไหวของมือ

ตารางที่ 4.2 ผลการทดสอบการเคลื่อนไหวของรถหุ่นยนต์ตามท่าทางการเคลื่อนไหวของมือ

ท่าทางของมือ	การเคลื่อนไหวของรถหุ่นยนต์
 REST	หยุดนิ่ง
 WAVE IN	เลี้ยวซ้าย
 WAVE OUT	เลี้ยวขวา
 FINGER SPREAD	ถอยหลัง
 FIST	เดินหน้า
 *DOUBLE TAP	-

* DOUBLE TAP เป็นท่าที่ใช้ในการเริ่มต้นหรือหยุดรับส่งข้อมูลท่าทางของมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การควบคุมรถหุ่นยนต์ด้วยการใช้ท่าทางของมือ โดยใช้การตรวจจับสัญญาณกล้ามเนื้อไฟฟ้า บริเวณแขนด้วย Myo Gesture Control Armband ซึ่งเป็นอุปกรณ์ EMG SENSOR ด้วยการเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ผ่านทาง การสื่อสารไร้สายบลูทูธเพื่อนำข้อมูลสัญญาณไปประมวลผลสั่งการให้มอเตอร์ขับเคลื่อน จากการทดสอบพบว่าท่าทางของมือที่สามารถตัดแยกได้มีจำนวน 6 ท่า ได้แก่ FIST, WAVE IN, WAVE OUT, FINGER SPREAD, REST, DOUBLE TAP

เมื่อผู้ใช้แสดงท่าทางต่างๆพบว่า การควบคุมรถหุ่นยนต์ให้เดินหน้า ถอยหลัง เลี้ยวซ้าย และเลี้ยวขวาเป็นไปตามคำสั่งที่ได้เขียนไว้ในไมโครคอนโทรลเลอร์ แต่พบปัญหาในการเปลี่ยนท่าทางจากท่าทางหนึ่งไปยังอีกท่าหนึ่งในทันทีที่มีความคลาดเคลื่อนในการตรวจจับส่งผลให้คำสั่งที่นำไปใช้ในการควบคุมมอเตอร์เกิดความผิดพลาด ทำให้การเคลื่อนไหวยของรถหุ่นยนต์ไม่เป็นไปตามที่ผู้ใช้งานต้องการ

ระยะในการเชื่อมต่อที่ยังสามารถควบคุมรถหุ่นยนต์ได้จะมีระยะไม่เกิน 15 เมตร ซึ่งขึ้นอยู่กับสภาพแวดล้อมและสิ่งกีดขวางในขณะนั้นด้วย หากการเชื่อมต่อขัดข้องจะต้องรองจนกว่าจะมีการเชื่อมต่อสัญญาณบลูทูธใหม่อัตโนมัติจึงจะสามารถควบคุมรถหุ่นยนต์ได้

5.2 ปัญหาและอุปสรรค

- MYO Armband ต้องใช้ Bluetooth Dongle ในการปิดทำงานเท่านั้น เนื่องจากยังไม่มีซอฟต์แวร์รองรับ คำสั่งที่ใช้ในการปิด ซึ่ง ณ ตอนนี้ทำได้เพียงทำให้ MYO อยู่ในโหมด Sleep ถึงแม้จะใช้พลังงานน้อยแต่ถ้าปล่อยทิ้งไว้เป็นระยะเวลานานก็ทำให้แบตเตอรี่หมดได้
- MYO Armband มีขนาดวงรอบที่ไม่เหมาะกับผู้ใช้ที่มีรอบวงข้อมือเล็ก ทำให้เมื่อสวมใส่มีขนาดที่ไม่พอดีและจะไม่สามารถตรวจจับสัญญาณ EMG ได้เนื่องจากแผ่นอิเล็กโทรดไม่ได้สัมผัสโดนกับผิวหนังของผู้สวมใส่
- การเชื่อมต่อระหว่าง MYO Armband กับ รถหุ่นยนต์ จะเกิดการขัดข้องได้ง่ายหากบริเวณรอบๆมีสัญญาณบลูทูธจากอุปกรณ์อื่นๆ ส่งผลให้การเชื่อมต่อไม่เสถียรภาพ และไม่สามารถควบคุมรถหุ่นยนต์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Electromyography [ออนไลน์] เข้าถึงได้จาก <http://haamor.com/th/> สืบค้น 4 กุมภาพันธ์ 2562
- [2] Myo Bridge [ออนไลน์] เข้าถึงได้จาก <https://github.com/vroland/MyoBridge> สืบค้น 5 กุมภาพันธ์ 2562
- [3] Myo Support [ออนไลน์] เข้าถึงได้จาก <https://support.getmyo.com/hc/en-us> สืบค้น 5 กุมภาพันธ์ 2562
- [4] Motor Driver L293D [ออนไลน์] เข้าถึงได้จาก <https://lastminuteengineers.com/L293d-motor-driver-shield-arduino-tutorial/> สืบค้น 6 กุมภาพันธ์ 2562
- [5] Bluetooth HM-11 [ออนไลน์] เข้าถึงได้จาก <https://www.elecrow.com/bluetooth-low-energy-module-hm11-cc2540-p-859.html> สืบค้น 6 กุมภาพันธ์ 2562
- [6] การควบคุมมอเตอร์กระแสตรง [ออนไลน์] เข้าถึงได้จาก <http://aimagin.com/blog/motor/?lang=th> สืบค้น 7 กุมภาพันธ์ 2562
- [7] Hand posture and gesture recognition [ออนไลน์] เข้าถึงได้จาก https://www.researchgate.net/publication/304298597_Hand_posture_and_gesture_recognition_using_MYO_arduino_and_spectral_collaborative_representation_based_classification สืบค้น 8 กุมภาพันธ์ 2562
- [8] Signal Processing [ออนไลน์] เข้าถึงได้จาก <https://signalprocessingsociety.org/our-story/signal-processing-101> สืบค้น 12 กุมภาพันธ์ 2562



ภาคผนวก ก
โค้ดคำสั่งที่เขียนลงในบอร์ดไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code ที่ใช้ในการนำท่าทางของมือมาสั่งการควบคุมมอเตอร์

```

#include <AFMotor.h>
#include <MyoBridge.h>
#include <SoftwareSerial.h>

AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

//SoftwareSerial connection to MyoBridge
SoftwareSerial bridgeSerial(2,3);

//initialize MyoBridge object with software serial connection
MyoBridge bridge(bridgeSerial);

int i=0;

//declare a function to handle pose data
void handlePoseData(MyoPoseData& data)
{
    //convert pose data to MyoPose
    MyoPose pose;
    pose = (MyoPose)data.pose;
    if (bridge.poseToString(pose) == "MYO_POSE_DOUBLE_TAP")
    {
        i++;
    }
    if (i%2 != 0)
    {
        if (bridge.poseToString(pose) == "MYO_POSE_REST")
        {
            motor1.run(RELEASE);
            motor2.run(RELEASE);
            motor3.run(RELEASE);
            motor4.run(RELEASE);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (bridge.poseToString(pose) == "MYO_POSE_WAVE_IN")
{
    motor3.run(RELEASE);
    motor4.run(RELEASE);
    motor2.run(FORWARD);
    motor1.run(FORWARD);
    motor1.setSpeed(250);
    motor2.setSpeed(250);
}
else if (bridge.poseToString(pose) == "MYO_POSE_WAVE_OUT")
{
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    motor3.run(FORWARD);
    motor4.run(FORWARD);
    motor3.setSpeed(255);
    motor4.setSpeed(255);
}
else if (bridge.poseToString(pose) == "MYO_POSE_FIST")
{
    motor1.run(FORWARD);
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    motor4.run(FORWARD);
    motor1.setSpeed(150);
    motor2.setSpeed(150);
    motor3.setSpeed(150);
    motor4.setSpeed(150);
}
else if (bridge.poseToString(pose) == "MYO_POSE_FINGERS_SPREAD")
{
    motor1.run(BACKWARD);
    motor2.run(BACKWARD);
    motor3.run(BACKWARD);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    motor4.run(BACKWARD);
    motor1.setSpeed(150);
    motor2.setSpeed(150);
    motor3.setSpeed(150);
    motor4.setSpeed(150);
}
//print the pose
Serial.println(bridge.poseToString(pose));
//Serial.println(poseString);
}
}
void setup() {
    //initialize both serial connections
    Serial.begin(115200);
    bridgeSerial.begin(115200);
    //wait until MyoBridge has found Myo and is connected. Make sure Myo is not connected to
    anything else and not in standby!
    Serial.println("Searching for Myo...");
    bridge.begin();
    Serial.println("connected!");

    //set the function that handles pose events
    bridge.setPoseEventCallBack(handlePoseData);
    //tell the Myo we want Pose data
    bridge.enablePoseData();
    //make sure Myo is unlocked
    bridge.unlockMyo();
}

void loop() {
    //update the connection to MyoBridge
    bridge.update();
}

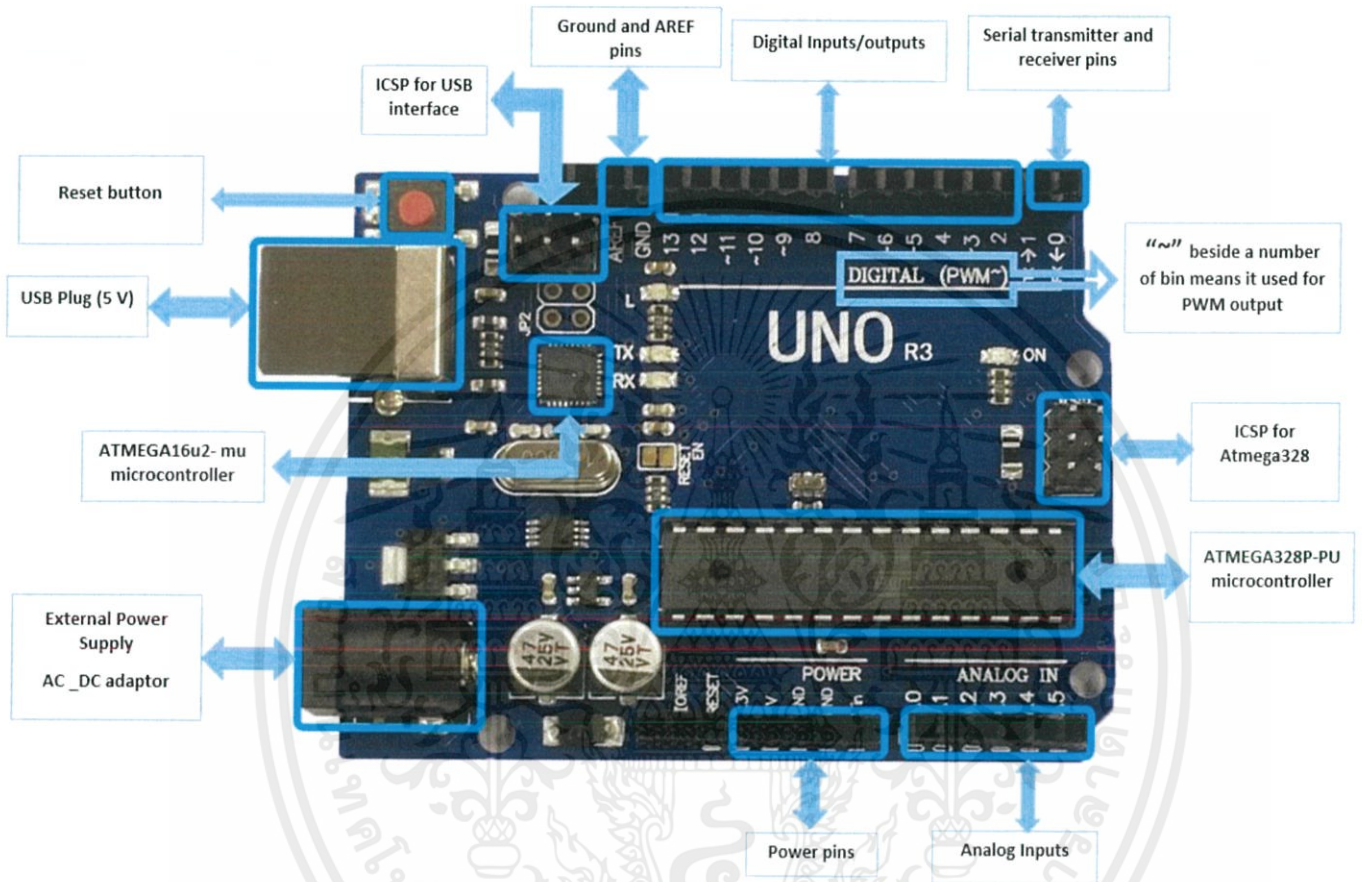
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Uno R3



INTRODUCTION

Arduino is used for building different types of electronic circuits easily using of both a physical programmable circuit board usually microcontroller and piece of code running on computer with USB connection between the computer and Arduino.

Programming language used in Arduino is just a simplified version of C++ that can easily replace thousands of wires with words.

ARDUINO UNO-R3 PHYSICAL COMPONENTS

ATMEGA328P-PU microcontroller

The most important element in Arduino Uno R3 is ATMEGA328P-PU is an 8-bit Microcontroller with flash memory reach to 32k bytes. It's features as follow:

- **High Performance, Low Power AVR**
- **Advanced RISC Architecture**
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- **High Endurance Non-volatile Memory Segments**
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
 - 256/512/512/1K Bytes EEPROM
 - 512/1K/1K/2K Bytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- **Peripheral Features**
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART

- Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface (Philips I2 C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

• Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

• I/O and Packages

- 23 Programmable I/O Lines
- 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

• Operating Voltage:

- 1.8 - 5.5V

• Temperature Range:

- -40°C to 85°C

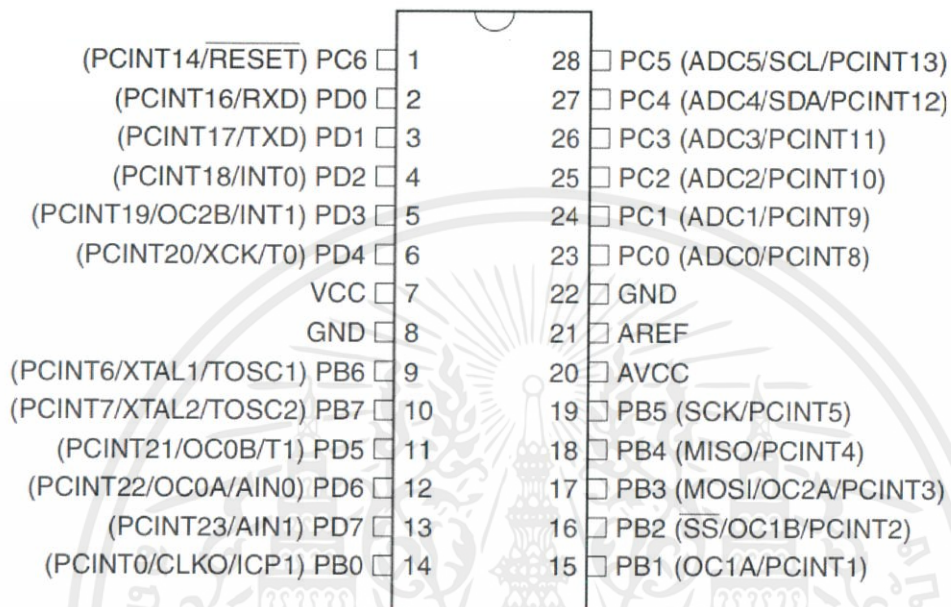
• Speed Grade:

- 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5.V, 0 - 20 MHz @ 4.5 - 5.5V

• Power Consumption at 1 MHz, 1.8V, 25°C

- Active Mode: 0.2 mA
- Power-down Mode: 0.1 μ A
- Power-save Mode: 0.75 μ A (Including 32 kHz RTC)

- Pin configuration



ATMEGA16u2- mu microcontroller

Is a 8-bit microcontroller used as USB driver in Arduino uno R3 it's features as follow:

- High Performance, Low Power AVR
- Advanced RISC Architecture
 - 125 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
- Non-volatile Program and Data Memories
 - 8K/16K/32K Bytes of In-System Self-Programmable Flash
 - 512/512/1024 EEPROM
 - 512/512/1024 Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C

- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by on-chip Boot Program hardware-activated after reset
- Programming Lock for Software Security
- **USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion**
 - Complies fully with Universal Serial Bus Specification REV 2.0
 - 48 MHz PLL for Full-speed Bus Operation: data transfer rates at 12 Mbit/s
 - Fully independent 176 bytes USB DPRAM for endpoint memory allocation
 - Endpoint 0 for Control Transfers: from 8 up to 64-bytes
 - 4 Programmable Endpoints:
 - IN or Out Directions
 - Bulk, Interrupt and Isochronous Transfers
 - Programmable maximum packet size from 8 to 64 bytes
 - Programmable single or double buffer
 - Suspend/Resume Interrupts
 - Microcontroller reset on USB Bus Reset without detach
 - USB Bus Disconnection on Microcontroller Request
- **Peripheral Features**
 - One 8-bit Timer/Counters with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
 - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Mode (three 8-bit PWM channels)
 - USART with SPI master only mode and hardware flow control (RTS/CTS)
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- **On Chip Debug Interface (debug WIRE)**
- **Special Microcontroller Features**
 - Power-On Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby
- **I/O and Packages**
 - 22 Programmable I/O Lines
 - QFN32 (5x5mm) / TQFP32 packages

- Operating Voltages

- 2.7 - 5.5V

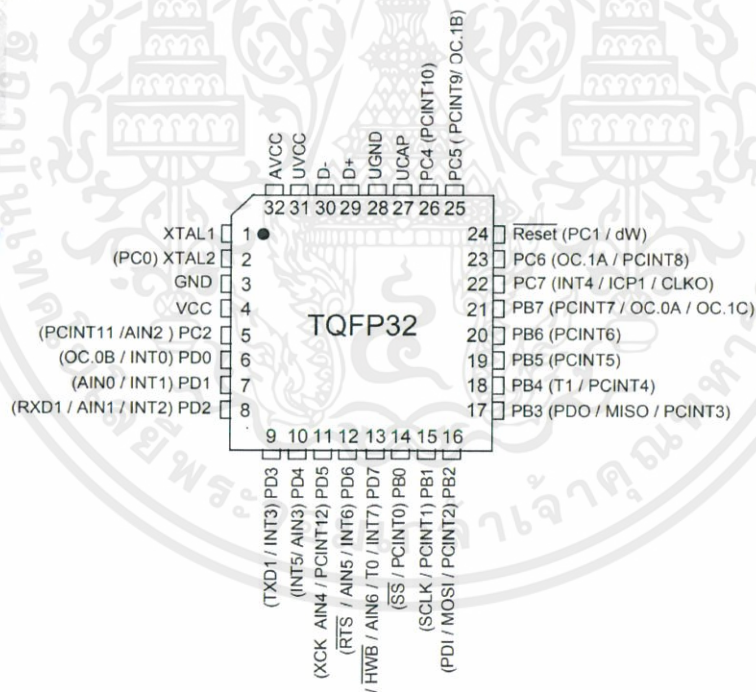
- Operating temperature

- Industrial (-40°C to +85°C)

- Maximum Frequency

- 8 MHz at 2.7V - Industrial range
- 16 MHz at 4.5V - Industrial range

- Pin configuration



OTHER ARDUINO UNO R3 PARTS

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

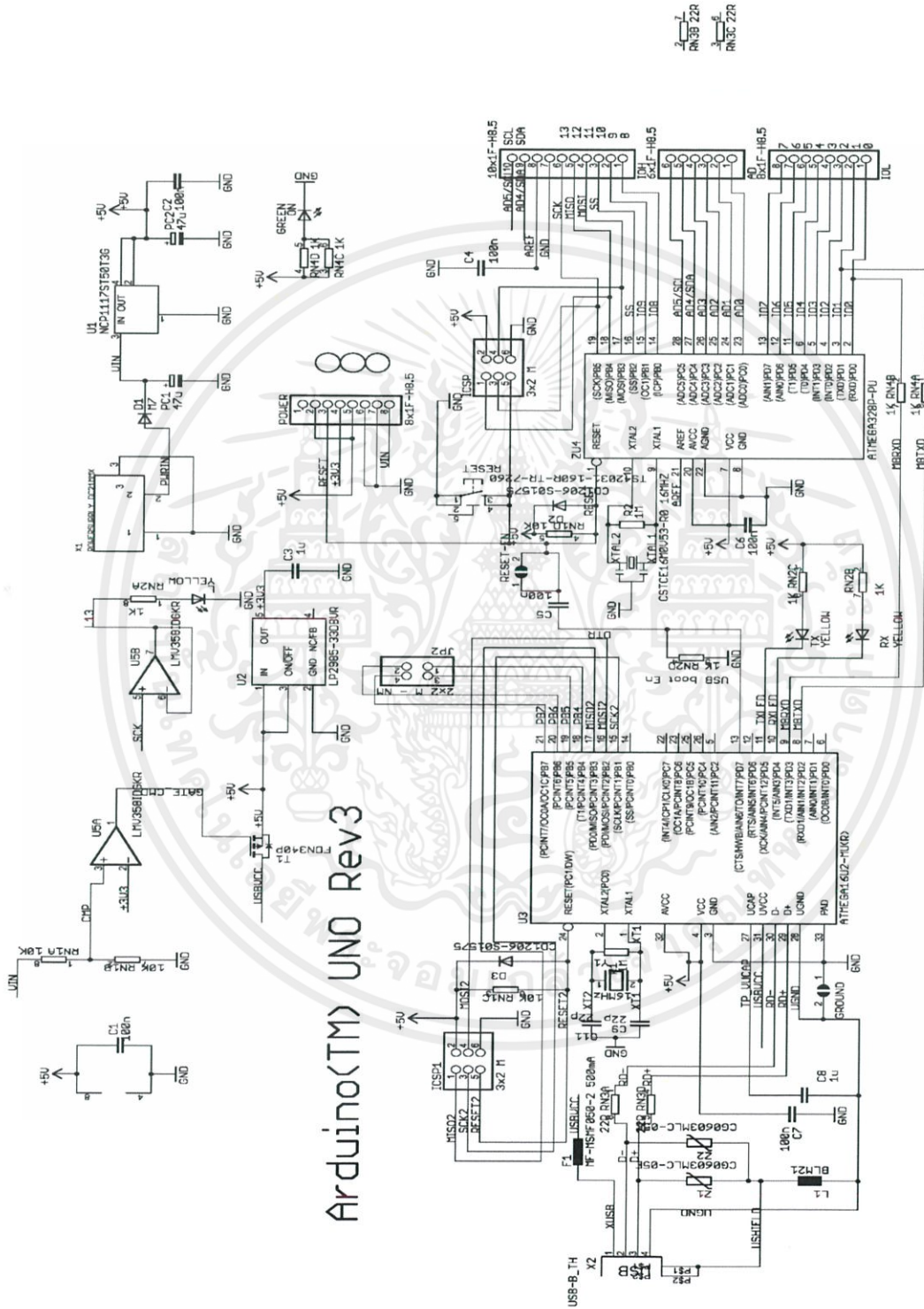
The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- AREF: Reference voltage for the analog inputs. Used with `analogReference()`.
- Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

ARDUINO UNO R3 SCHEMATIC DIAGRAM

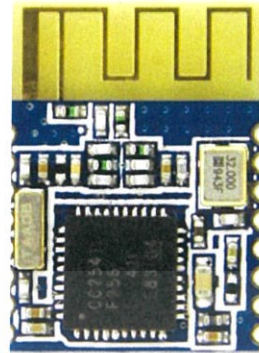


Arduino(TM) UNO Rev3

Bluetooth HM-11

Product Description

This is a SMD BLE module used in our BLE Bee and Xadow BLE. It is based on TI cc2541 chip, enables robust network nodes to be built with low total bill-of-material costs and highly suited for ultralow power consumption systems. The module is small and easy to use, with the preprogrammed firmware of manufacturer, you could quickly build BLE communications via its AT command. Supporting BLE communications with iphone, ipad and Android 4.3.



Features:

- Bluetooth protocol: Bluetooth Specification V4.0 BLE
- Working frequency: 2.4 GHz ISM band
- Interface way: a serial port
- Open environment within 30 meters can realize communication between modules
- To send and receive no byte limit between modules
- Modulation method: GFSK (Gaussian Frequency Shift Keying)
- Transmission power: - DBM, 23-6 DBM, 0 DBM, 6 DBM, can be modified by the AT command
- use TI CC2540 chip, configuration space of 256 KB, support the AT command, the user can according to need to change the role (master, slave mode) and the serial port baud rate, name of equipment, matching parameters such as passwords, use agile.
- power supply: + 3.3 VDC 50 mA
- working temperature: - 5 ~ + 65 Centigrade

Specification:

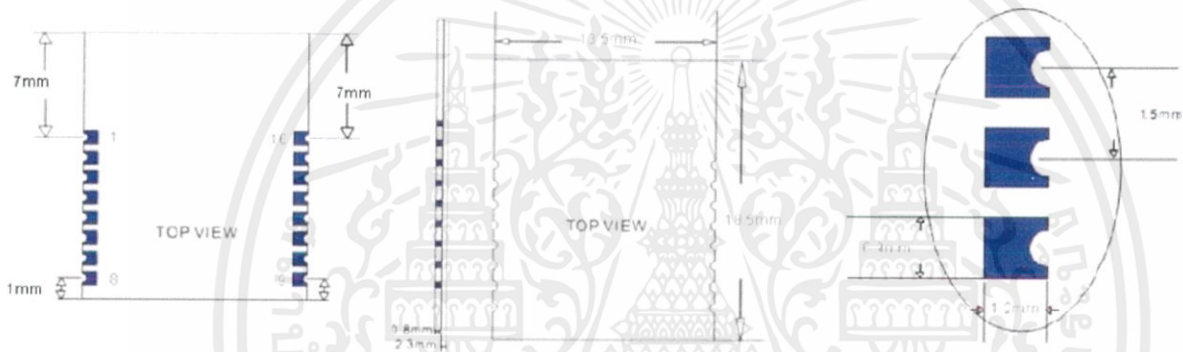
Specification	Value
Microprocessor	CC2541
Resources	Support the AT command, the user can according to need to change the role (master, slave mode) and the serial port baud rate, name of equipment, Matching parameters such as password, the use of flexible.
Outline Dimension	13.5mm x 18.5mm x 2.3mm
Power supply	3.3V
Communication Protocol	Uart(3.3V LVTTTL)
IO counts	2
Key input IO	1
LED Indicators IO	1

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics:

Specification	Min	Typ	Max	Unit
Max Input Voltage	-0.3		3.6	V
Working Input Voltage	2.0	3.3	3.6	V
Transmit Current		15		mA
Receive Current		8.5		mA
Deep Sleep Current		600		uA
Operating Temperature	-40		125	°C

Pin Definition:



Pin	Name	Description
1	UART_RTS	UART
2	UART_TX	UART
3	UART_CTS	UART
4	UART_RX	UART
5	NC	
6	NC	
7	NV	
8	NV	

Pin	Name	Description
9	VCC	Power supply 3.3V
10	NC	
11	RESETB	Reset, active low at least in 5ms
12	GND	GND
13	PIO3	IO port, used for connect to DHT11/DS18B20
14	PIO2	Digital input, output
15	PIO1	LED indicator
16	PIO0	Button pin

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

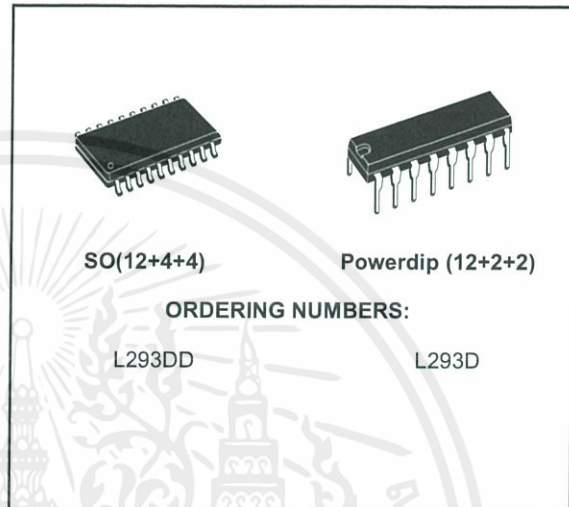
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

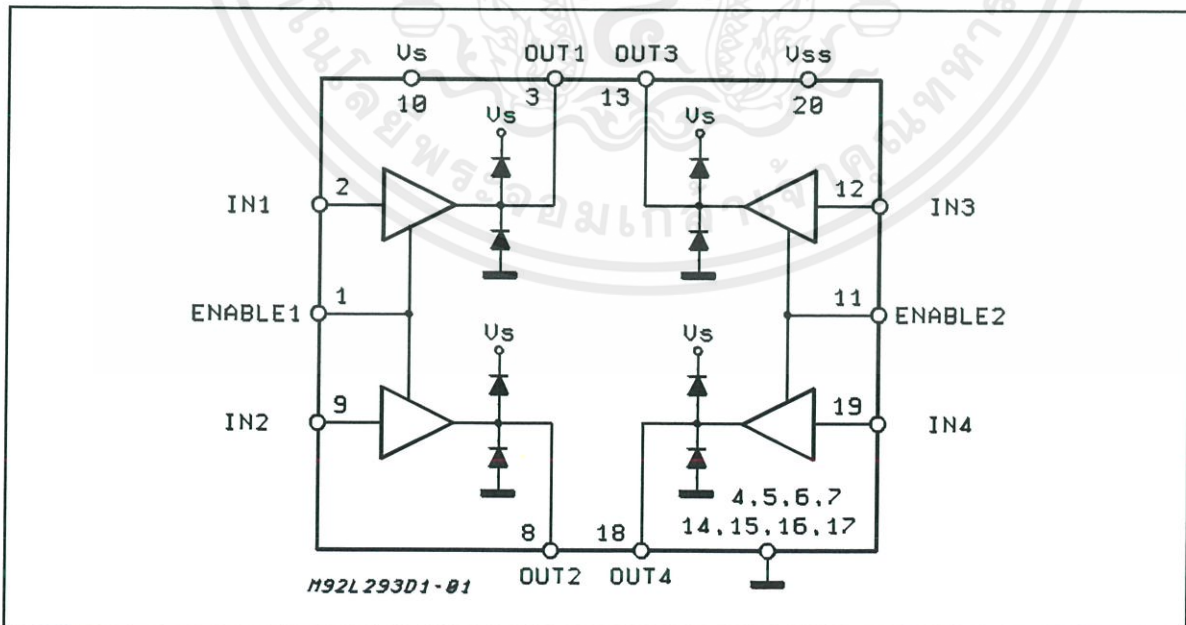
This device is suitable for use in switching applications at frequencies up to 5 kHz.



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

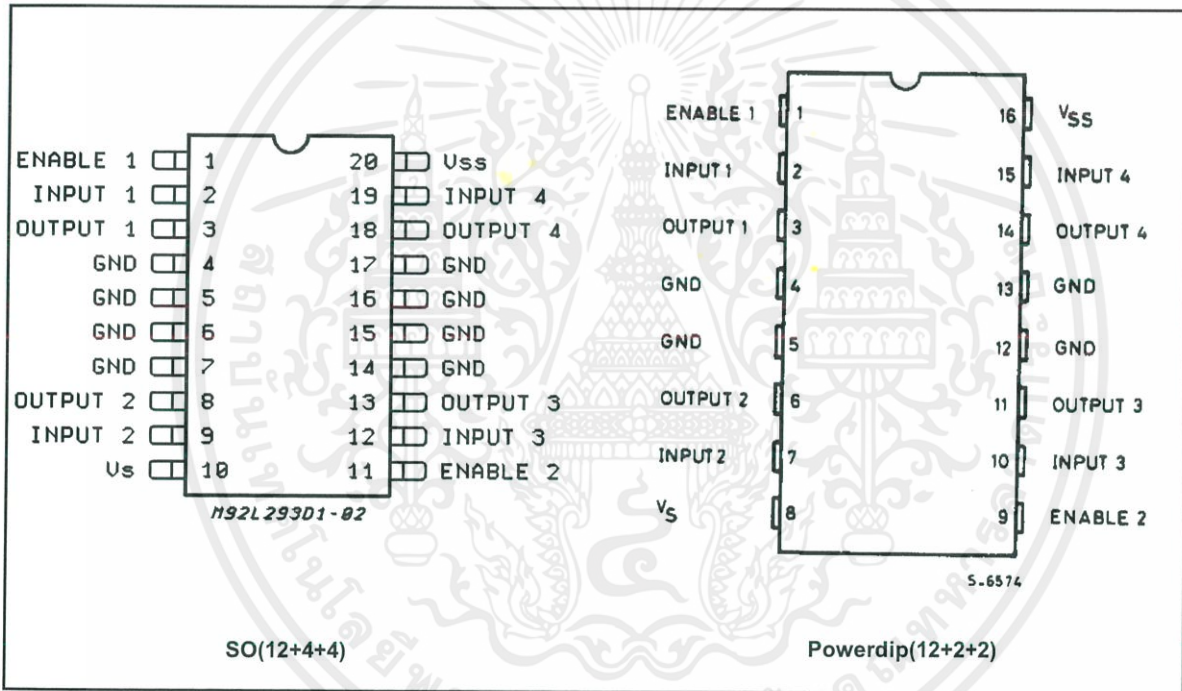
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _S	Supply Voltage	36	V
V _{SS}	Logic Supply Voltage	36	V
V _i	Input Voltage	7	V
V _{en}	Enable Voltage	7	V
I _o	Peak Output Current (100 μs non repetitive)	1.2	A
P _{tot}	Total Power Dissipation at T _{pins} = 90 °C	4	W
T _{stg} , T _j	Storage and Junction Temperature	- 40 to 150	°C

PIN CONNECTIONS (Top view)



THERMAL DATA

Symbol	Description	DIP	SO	Unit
R _{th j-pins}	Thermal Resistance Junction-pins	max.	14	°C/W
R _{th j-amb}	Thermal Resistance junction-ambient	max.	50 (*)	°C/W
R _{th j-case}	Thermal Resistance Junction-case	max.	-	

(*) With 6sq. cm on board heatsink.

ELECTRICAL CHARACTERISTICS (for each channel, $V_S = 24\text{ V}$, $V_{SS} = 5\text{ V}$, $T_{amb} = 25\text{ }^\circ\text{C}$, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 10)		V_{SS}		36	V
V_{SS}	Logic Supply Voltage (pin 20)		4.5		36	V
I_S	Total Quiescent Supply Current (pin 10)	$V_i = L$; $I_O = 0$; $V_{en} = H$		2	6	mA
		$V_i = H$; $I_O = 0$; $V_{en} = H$		16	24	mA
		$V_{en} = L$			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	$V_i = L$; $I_O = 0$; $V_{en} = H$		44	60	mA
		$V_i = H$; $I_O = 0$; $V_{en} = H$		16	22	mA
		$V_{en} = L$		16	24	mA
V_{IL}	Input Low Voltage (pin 2, 9, 12, 19)		-0.3		1.5	V
V_{IH}	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{IL}	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_{IL} = 1.5\text{ V}$			-10	μA
I_{IH}	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3\text{ V} \leq V_{IH} \leq V_{SS} - 0.6\text{ V}$		30	100	μA
V_{enL}	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
V_{enH}	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{enL}	Low Voltage Enable Current (pin 1, 11)	$V_{enL} = 1.5\text{ V}$		-30	-100	μA
I_{enH}	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{enH} \leq V_{SS} - 0.6\text{ V}$			± 10	μA
$V_{CE(sat)H}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = -0.6\text{ A}$		1.4	1.8	V
$V_{CE(sat)L}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = +0.6\text{ A}$		1.2	1.8	V
V_F	Clamp Diode Forward Voltage	$I_O = 600\text{ nA}$		1.3		V
t_r	Rise Time (*)	0.1 to 0.9 V_O		250		ns
t_f	Fall Time (*)	0.9 to 0.1 V_O		250		ns
t_{on}	Turn-on Delay (*)	0.5 V_i to 0.5 V_O		750		ns
t_{off}	Turn-off Delay (*)	0.5 V_i to 0.5 V_O		200		ns

(*) See fig. 1.

TRUTH TABLE (one channel)

Input	Enable (*)	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

Z = High output impedance
 (*) Relative to the considered channel

Figure 1: Switching Times

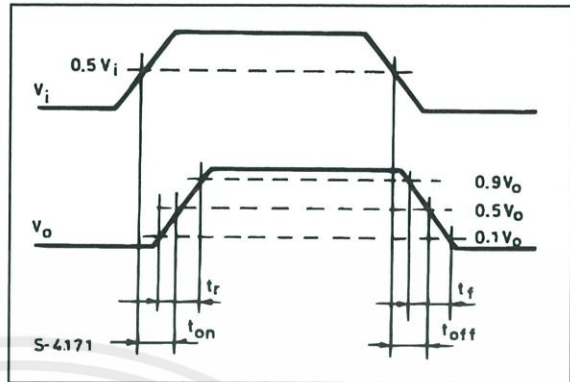
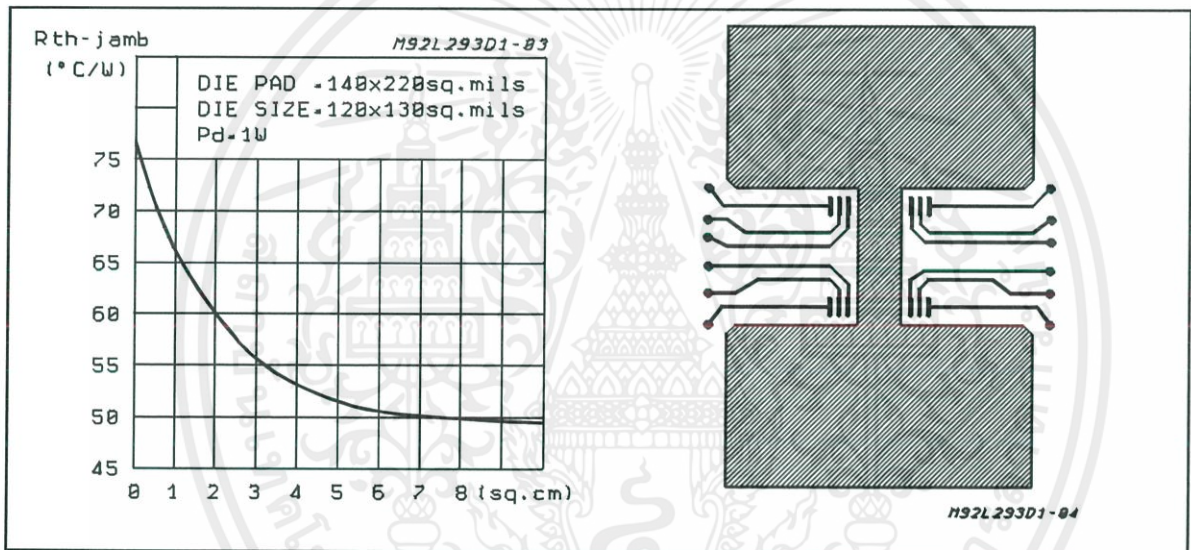
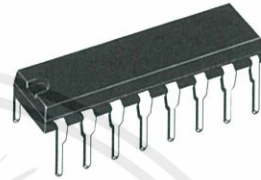


Figure 2: Junction to ambient thermal resistance vs. area on board heatsink (SO12+4+4 package)

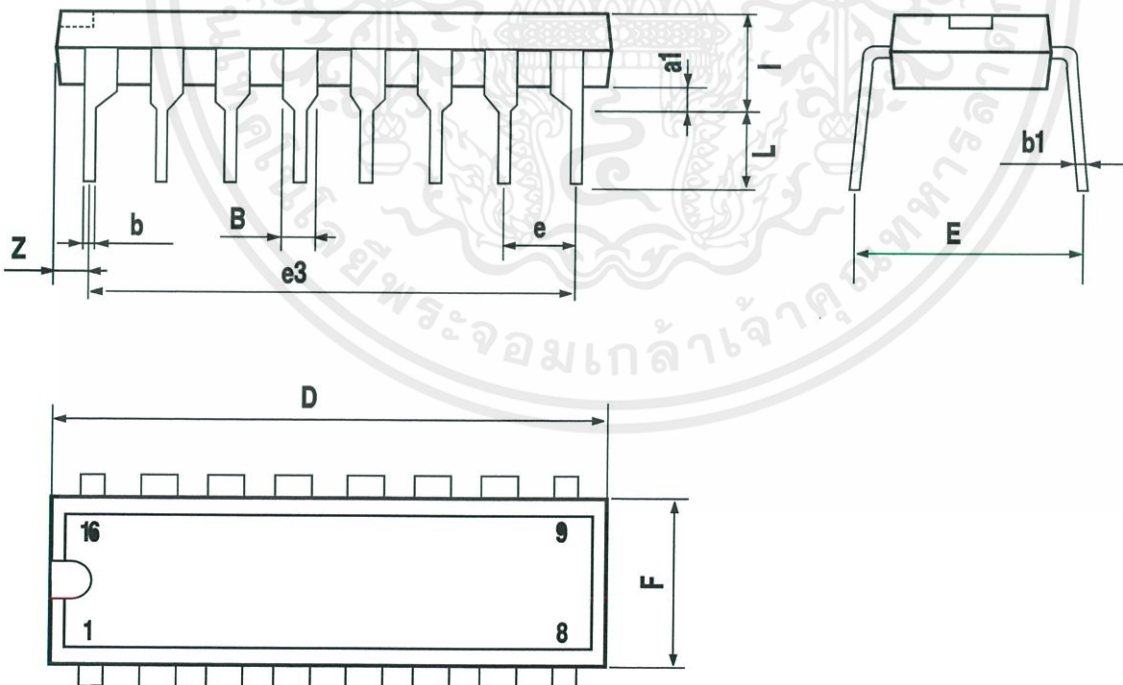


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.85		1.40	0.033		0.055
b		0.50			0.020	
b1	0.38		0.50	0.015		0.020
D			20.0			0.787
E		8.80			0.346	
e		2.54			0.100	
e3		17.78			0.700	
F			7.10			0.280
I			5.10			0.201
L		3.30			0.130	
Z			1.27			0.050

OUTLINE AND MECHANICAL DATA



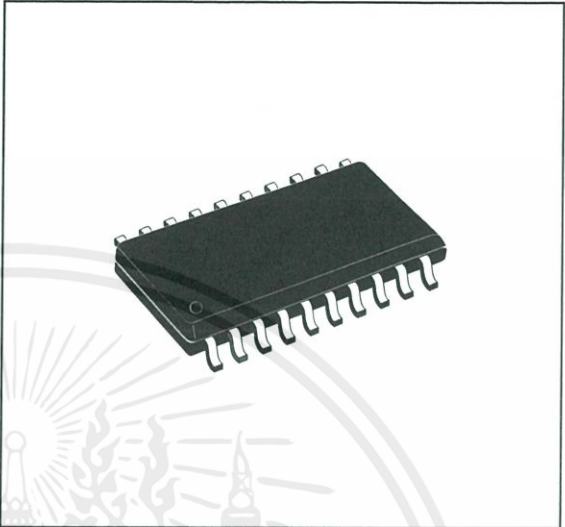
Powerdip 16



L293D - L293DD

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	2.35		2.65	0.093		0.104
A1	0.1		0.3	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.6		13	0.496		0.512
E	7.4		7.6	0.291		0.299
e		1.27			0.050	
H	10		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
L	0.4		1.27	0.016		0.050
K	0° (min.)8° (max.)					

OUTLINE AND MECHANICAL DATA



SO20

