



## โครงการ

รถขนส่งสินค้าอัตโนมัตินำทางด้วยเส้นแฉบแม่เหล็ก



นายนาวิน                      ฤกษ์เจริญ                      B5927128

นายฐิติมา                      หมั่นแก้ว                      B5926992

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ สำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2561

หัวข้อ : รถขนส่งสินค้าอัตโนมัติทางด้วยเส้นแถบแม่เหล็ก  
ชื่อนักศึกษา : นายนาวิน ฤกษ์เจริญ  
รหัสนักศึกษา : B5927128  
ชื่อนักศึกษา : นางสาวฐิติมา หมั่นแก้ว  
รหัสนักศึกษา : B5926992  
หลักสูตร : วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเมคคาทรอนิกส์  
ปีการศึกษา : 2561  
อาจารย์ที่ปรึกษา : อาจารย์วิชัย ศรีสุรักษ์

#### บทคัดย่อ

จากสถานที่ทำงานการเดินทางไปรับสินค้าที่สโตร์ใช้เวลาในการเดินทาง 10 – 15 นาที ทำให้การซ่อมเครื่องจักรเกิดความล่าช้า จึงได้ออกแบบและสร้างโครงงาน รถขนส่งชิ้นส่วนอัตโนมัติทางด้วยเส้นแถบแม่เหล็กนี้ขึ้นมาเพื่อความสะดวกและรวดเร็วในการขนส่ง รถจะถูกออกแบบและสร้างให้ขับเคลื่อนด้วยระบบไฟฟ้า ระยะทำการไปกลับประมาณ 600 เมตร ความเร็วในการวิ่งไม่เกิน 20 กิโลเมตรต่อชั่วโมง สามารถรับน้ำหนักบรรทุกไม่เกิน 10 กิโลกรัม ขนาดของรถ 30x40x15 เซนติเมตร รถจะทำงานด้วยการกดปุ่มและนำทางด้วยเส้นแถบแม่เหล็ก โดยใช้ Arduino ในการควบคุม ผลที่ได้จากโครงงาน คือ ลดระยะเวลาในการขนส่งชิ้นส่วนและอุปกรณ์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลงได้ด้วยดีด้วยการให้คำปรึกษา ความรู้และคำแนะนำที่ดีจากอาจารย์ที่ปรึกษา อาจารย์ วิชัย ศรีสุรรัมย์ จนรายงานฉบับนี้สำเร็จไปได้ด้วยดี

สุดท้ายนี้ โครงการฉบับนี้ถูกล่วงไปด้วยดีคุณค่าและประโยชน์อันพึงมาจากโครงการฉบับนี้ ข้าพเจ้าขอขอบแต่ผู้มีพระคุณทุกท่าน และสำหรับผู้ที่นำไปพัฒนาระบบต่อไปในภายภาคหน้าแก่ผู้ที่สนใจศึกษา



นาวิน ฤกษ์เจริญ  
ฐิติมา หมั่นแกว  
มิถุนายน 2562

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## สารบัญ

|  |   |
|--|---|
| บทคัดย่อภาษาไทย .....  |   |
| กิตติกรรมประกาศ.....   | ก |
| สารบัญ .....   | ข |
| สารบัญตาราง .....  | จ |
| สารบัญรูปภาพ .....   | ฉ |
| บทที่ 1 บทนำ .....   | 1 |
| 1.1 ความเป็นมาและความสำคัญของปัญหา.....                          | 1 |
| 1.2 ปัญหาและแรงจูงใจ .....                                       | 2 |
| 1.3 วัตถุประสงค์ของรายงาน.....                                   | 2 |
| 1.4 ขอบเขตของรายงาน .....  | 2 |
| 1.5 ประโยชน์ของรายงาน .....                                      | 2 |
| 1.6 โครงสร้างของรายงาน.....                                      | 2 |
| บทที่ 2 พื้นฐานและทฤษฎีที่เกี่ยวข้อง.....                        | 3 |
| 2.1 หลักการของเอจีวีเบื้องต้น .....                              | 3 |
| 2.1.1 ตัวรถกลไก, ระบบขับเคลื่อน .....                            | 3 |
| 2.1.2 ชุดควบคุมการทำงานของรถ .....                               | 3 |
| 2.1.3 ระบบนำร่องทางเดินรถ .....                                  | 3 |
| 2.1.4 ส่วนติดต่อสื่อสารข้อมูลหรือเชื่อมต่อกับคอมพิวเตอร์ .....   | 4 |
| 2.2 ชนิดของเอจีวี .....  | 4 |
| 2.2.1 เอจีวีแบบลากจูง (AGV Towing Vehicle).....                  | 4 |
| 2.2.2 เอจีวีแบบบรรทุกทุกขนาดเบา(AGV Light Load Transporter)..... | 4 |
| 2.2.3 เอจีวีแบบบรรทุกทุกพาเลท(AGV Pallet Truck) .....            | 5 |
| 2.3 การนำร่องของเอจีวี.....                                      | 5 |
| 2.3.1 การนำร่องโดยใช้ทางเดินนำร่อง.....                          | 5 |
| 2.3.1.1 การใช้แถบโลหะ (Metal Tape).....                          | 5 |

## สารบัญ(ต่อ)

|  |    |
|--|----|
| 2.3.1.2 การใช้แถบแม่เหล็ก (Magnetic Tape).....                                       | 5  |
| 2.3.1.3 การใช้กล้องทีวีหรือซีซีดี (CCD Camera) .....                                 | 6  |
| 2.3.2 การนำร่องแบบไร้สาย (Wireless Guidance).....                                    | 6  |
| 2.3.2.1 การนำทางด้วยรีโมทคอนโทรล (Remote Control Navigation).....                    | 6  |
| 2.3.2.2 การใช้อัลตราโซนิกวัดระยะทาง (Ultrasonic Distance Measurement).....           | 6  |
| 2.4 บอร์ด Arduino Uno R3 .....   | 7  |
| 2.4.1 คุณสมบัติทางเทคนิคของบอร์ด Arduino Uno R3 .....                                | 7  |
| 2.5 ไดร์ขับเคลื่อนมอเตอร์ (L298 Motor Driver Module 2A) .....                        | 9  |
| 2.5.1 รายละเอียดของบอร์ด .....   | 10 |
| 2.5.2 สเปกทางเทคนิค.....   | 10 |
| 2.5.3 การต่อเข้าใช้งานกับ Arduino .....  | 11 |
| 2.6 รีกิวเลเตอร์ (Regulator) .....   | 12 |
| บทที่ 3 การออกแบบระบบ.....   | 13 |
| 3.1 โครงสร้างและการออกแบบระบบ .....  | 13 |
| 3.1.1 ส่วนซอฟต์แวร์ (Software).....  | 13 |
| 3.1.2 ส่วนฮาร์ดแวร์ (Hardware) .....   | 13 |
| 3.2 รายละเอียดโครงสร้างขององค์ประกอบ .....   | 14 |
| 3.2.1 ส่วนเชื่อมต่อผู้ใช้งาน (User) .....  | 14 |
| 3.2.2 ส่วนควบคุมวงจรระบบและประมวลผลของเครื่อง (Programming).....                     | 14 |
| 3.2.3 ส่วนควบคุมการ ON /OFF เครื่อง .....  | 14 |
| 3.3 การออกแบบการเชื่อมต่อสั่งงานกันระหว่าง บอร์ด Arduino กับ เซนเซอร์และมอเตอร์..... | 14 |
| 3.3.1 ส่วนที่เป็นอุปกรณ์ บอร์ด Arduino .....   | 14 |
| 3.3.2 เซนเซอร์และมอเตอร์.....  | 14 |
| 3.4 การออกแบบ Hardware ในส่วนของวงจร .....   | 15 |
| บทที่ 4 การทดลองและผลการทดลอง.....   | 18 |

## สารบัญ(ต่อ)

|  |    |
|--|----|
| 4.1การทดลอง.....   | 18 |
| 4.1.1 ทดลองใช้บอร์ด Arduino ในการสั่งงานมอเตอร์ทั้ง 2 ตัว .....  | 18 |
| 4.1.2 ทดลองใช้บอร์ด Arduino ในการสั่งงานเซ็นเซอร์ทั้ง 3 ตัว..... | 18 |
| บทที่ 5 สรุปผลการทดลอง .....                                     | 23 |
| 5.1 ประโยชน์จากการทำรายงาน .....                                 | 23 |
| 5.2 ปัญหาและอุปสรรคของรถขนส่งสินค้าอัตโนมัติ.....                | 23 |
| 5.3 แนวทางการนำไปพัฒนาต่อ.....                                   | 23 |
| เอกสารอ้างอิง .....  | 24 |
| ภาคผนวก .....  | 25 |
| ภาคผนวก ก.....   | 26 |
| ภาคผนวก ข.....   | 36 |
| ประวัติผู้เขียน .....  | 50 |

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## สารบัญตาราง

| ตารางที่   | หน้า |
|--|------|
| 4.1 แสดงผลการทดสอบการหมุนของมอเตอร์.....                       | 19   |
| 4.2 แสดงผลการทดสอบสถานะและแสดงสัญญาณของเซ็นเซอร์ตัวที่ 1 ..... | 21   |
| 4.3 แสดงผลการทดสอบสถานะและแสดงสัญญาณของเซ็นเซอร์ตัวที่ 2 ..... | 21   |
| 4.4 แสดงผลการทดสอบสถานะและแสดงสัญญาณของเซ็นเซอร์ตัวที่ 3 ..... | 22   |



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## สารบัญรูปภาพ

| รูปที่                                      | หน้า |
|---|------|
| 2.1 ส่วนประกอบของบอร์ด Arduino Uno R3 ..... | 8    |
| 2.2 การจัดเรียงขา Arduino Uno R3 .....      | 8    |
| 2.3 แสดงรายละเอียดของบอร์ด .....            | 9    |
| 2.4 แสดงการต่อใช้งานอุปกรณ์ .....           | 11   |
| 2.5 รีจิวเลเตอร์ (Regulator).....           | 12   |
| 3.1 องค์ประกอบของระบบโดยรวม .....           | 13   |
| 3.2 การออกแบบวงจรควบคุมมอเตอร์โดยรวม.....   | 15   |
| 3.3 ชุดอุปกรณ์และวงจรไฟฟ้าโดยรวม .....      | 16   |
| 4.1 แสดงรถขนส่งสินค้าอัตโนมัติ.....         | 18   |
| 4.2 แสดงการทดสอบการหมุนของมอเตอร์.....      | 19   |
| 4.3 แสดงสถานะของเซ็นเซอร์.....              | 20   |
| 4.4 แสดงสัญญาณของเซ็นเซอร์.....             | 20   |

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันระบบอัตโนมัติมีความจำเป็นในโรงงานอุตสาหกรรมอย่างมาก ซึ่งมีผลมาจากการแข่งขันกันด้านการตลาดอย่างจริงจัง และมีแนวโน้มที่จะเพิ่มขึ้นเรื่อย ๆ ปัจจัยในด้านการตลาดที่ขาดไม่ได้ก็คือ การกำหนดเวลาส่งมอบผลิตภัณฑ์และปริมาณที่ต้องการ ซึ่งจะต้องมีการผลิตแข่งกับเวลา ประโยชน์ของระบบอัตโนมัติทำให้ช่วยลดเวลาในการทำงานจากปกติทั้งในรูปแบบการเปลี่ยนเครื่องมือ การขนย้ายผลิตภัณฑ์ และกระบวนการต่าง ๆ ที่คนไม่สามารถเข้าไปปฏิบัติงานได้ การขนถ่ายวัสดุเป็นกิจกรรมที่มีความสำคัญมากต่อระบบการผลิต ดังนั้นการขนย้ายวัสดุหรืออุปกรณ์จำเป็นต้องมีระบบการเคลื่อนย้ายอัตโนมัติมาช่วยในการทำงาน เพื่อลดต้นทุน เพิ่มความรวดเร็ว ความถูกต้องและไม่ทำให้วัสดุเกิดความเสียหาย สำหรับอุปกรณ์ที่เหมาะสมต่อการทำงานในระบบอัตโนมัติก็คือระบบสายพานลำเลียง และเอจวีวี (Automatic Guided Vehicle : AGV) แต่ระบบสายพานลำเลียงจะเกี่ยวข้องกับวิธีการผลิตแบบเป็นจำนวนมากและมีทิศทางไหลของวัสดุคงที่ สำหรับเอจวีวีนั้นจะมีความคล่องตัวมากกว่าในการขนถ่ายวัสดุอัตโนมัติเอจวีวีจึงเหมาะที่จะใช้งานขนถ่ายวัสดุชนิดแตกต่างกันแล้วต้องถูกส่งไปยังตำแหน่งเป้าหมายที่แตกต่าง

แต่ในโรงงานอุตสาหกรรมส่วนใหญ่มักจะใช้ระบบนำร่องแบบกำหนดเส้นทางเดินแน่นอนซึ่งมีการติดตั้งที่ลำบากเพราะต้องฝังสายลงในพื้น โรงงานเวลาที่มีการปรับเปลี่ยนสายการผลิตจึงไม่สามารถทำได้ง่ายนักหรือใช้สีสะท้อนแสงทาลงบนพื้นแล้วเอจวีวีจะใช้เซนเซอร์ในการตรวจจับ เส้นทางเดินซึ่งจะมีปัญหาเวลาสีเกิดหลุดหายไปหรือพื้น โรงงานมีความสกปรกทำให้เอจวีวี เคลื่อนที่ผิดพลาด อาจเกิดความเสียหายขึ้นได้ซึ่งเป็นข้อจำกัดในการทำงานแบบยืดหยุ่น เอจวีวีก็มีลำดับการทำงานที่แน่นอน การที่จะเปลี่ยนลำดับการทำงานก็ทำได้ยากเนื่องจากต้องทำการเขียน โปรแกรมเข้าไปใหม่

## 1.2 ปัญหาและแรงจูงใจ

โครงการนี้ได้ให้ความสำคัญของปัญหาที่เกิดขึ้นในข้างต้น จึงได้เกิดแนวความคิดใน การสร้างรถขนส่งสินค้าอัตโนมัติทางด้วยเส้นแถบแม่เหล็ก(Automatic Guided Vehicle : AGV)เพื่อเพิ่มความเร็วในการขนส่งและลดปัญหาการตรวจจับเส้นทางเดินของเอจิวี โดยนำบอร์ด Arduino ซึ่งเป็นที่นิยมและราคาข้อมเยาว์ มาเป็นอุปกรณ์ควบคุมการทำงาน

## 1.3 วัตถุประสงค์

เพื่อศึกษา ออกแบบและสร้างรถขนส่งชิ้นส่วนอัตโนมัติทางด้วยเส้นแถบแม่เหล็ก

## 1.4 ขอบเขตของโครงการ

1.4.1 สร้างและออกแบบให้สามารถรับน้ำหนักบรรทุกได้ 10 กิโลกรัม

1.4.2 ใช้การนำทางโดยเส้นแถบแม่เหล็ก

1.4.3 ขับเคลื่อนด้วยระบบไฟฟ้า ระยะทำการไปกลับไม่เกิน 600 เมตร ที่ความเร็วไม่เกิน 20 กิโลเมตร/ชั่วโมง

## 1.6 โครงสร้างของรายงาน

สำหรับเนื้อหาของรายงานฉบับนี้จะถูกแบ่งออกได้ดังนี้

บทที่ 1 บทนำกล่าวถึงปัญหาที่เกิดขึ้นในสถานการณ์ใช้รถขนส่งสินค้าอัตโนมัติ

บทที่ 2 ทฤษฎีและหลักการทํางาน เป็นการศึกษาการทํางานของอุปกรณ์ต่าง ๆ ที่จะนำมาใช้เป็นข้อมูลเพื่อใช้ในการออกแบบระบบ

บทที่ 3 การออกแบบระบบและแสดงขั้นตอนการทํางานของระบบในการ ออกแบบระบบจะมีขั้นตอนการทํางานแต่ละขั้นตอนต่าง ๆ เพื่อให้การทํางานเป็นไป อย่างมีประสิทธิภาพ

บทที่ 4 การทดลองและผลการทดลองจะนำเอาระบบที่ออกแบบไว้มาทดลองใช้เพื่อนำผลที่ทดลองได้มาประเมินประสิทธิภาพของการทำงาน

บทที่ 5 สรุปผลและข้อเสนอแนะ นำเอาผลที่ได้จากการทดลองการใช้งาน เพื่อหาข้อผิดพลาดเพื่อเป็นแนวทางในการปรับปรุงแก้ไขให้ดีขึ้นต่อไปต่อไป

## บทที่ 2

### พื้นฐานและทฤษฎีที่เกี่ยวข้อง

#### 2.1 หลักการของเอจิวีเบื้องต้น

เอจิวี(Automatic Guide Vehicle : AGV) เป็นรถขนาดเล็กที่เคลื่อนที่ได้เองโดยไม่ต้องมี คนขับ เอจิวีถูกนำมาใช้งานจริงในอุตสาหกรรมครั้งแรกเมื่อปี ค.ศ. 1953 ในประเทศ สหรัฐอเมริกาโดยตอน นั้นถูกนำมาใช้ขนถ่ายสินค้าใน โกดังเก็บสินค้าทำให้สามารถประหยัดใน เรื่องของแรงงานคนและเวลา ได้เป็นอย่างดี ซึ่งเป็นที่รู้จักกันดีว่าเป็นอุปกรณ์ลำเลียงที่มีการ บังคับด้วยคนน้อยที่สุดนั้นเป็นการ รวมกันของแต่ละส่วนของระบบการผลิตรอัตโนมัติกับการ เจริญเติบโตของระบบการผลิตที่มีปริมาณ และความหลากหลายในระดับกลางซึ่งเป็นระบบการ ผลิตแบบยืดหยุ่น(Flexible Manufacturing System, FMS) ที่มีความต้องการไม่เฉพาะ เครื่องจักรที่มีการผลิตแบบยืดหยุ่นแต่ยังต้องการรวมถึงการ ขนย้ายการจัดเก็บและการนำออก อย่างยืดหยุ่น ซึ่งนั่นเป็นพื้นฐานในการศึกษาเรื่องเอจิวีและระบบการ จัดเก็บและนำออกแบบ อัตโนมัติ(Automatic Storage and Retrieval System : AS/RS) โดยเชื่อมโยงกัน กับระบบการ ผลิตแบบยืดหยุ่น[1]

ระบบเอจิวีเป็นระบบขนถ่ายวัสดุที่ใช้รถทำงานได้โดยอิสระต่อกัน ขับเคลื่อนด้วยตัวเอง ระบบ เอจิวีนี จะประกอบด้วย

2.1.1 ตัวรถกลไก, ระบบขับเคลื่อน โดยทั่วไปส่วนของตัวรถและกลไกของรถนั้นก็ถูก ออกแบบมาให้เหมาะกับการประยุกต์ใน การใช้งานในอุตสาหกรรมแต่ละประเภทแล้วตัวขับเคลื่อนให้ รถเคลื่อนที่นั่นก็จะเป็นมอเตอร์ไฟฟ้าที่

2.1.2 ชุดควบคุมการทำงานของรถ เป็นการควบคุมด้วยวงจรรีเลย์อิเล็กทรอนิกส์ซึ่งพลังงานในการ หมุนมอเตอร์จะได้มาจากแบตเตอรี่ซึ่งจะมีแรงดันอยู่ระหว่าง 12 ถึง 48 โวลต์ ส่วนชุดควบคุมการ ทำงานของรถจะมีอยู่ ด้วยกันหลายลักษณะเช่น พีแอลซี(Programmable Logic Control:PLC) ไมโคร โปรเซสเซอร์บอร์ด ซิงเกิลบอร์ดคอมพิวเตอร์ซึ่งขึ้นอยู่กับว่าผู้ผลิตรายใดจะออกแบบระบบไว้ อย่างไร

2.1.3 ระบบนำร่องทางเดินรถ ระบบนำร่องทางเดินของเอจิวีจะแบ่งออกเป็นสองแบบ คือ แบบ กำหนดเส้นทางเดินแน่นอน เช่น แบบใช้สายและแบบใช้แถบสีบนพื้นกำหนดทางเดิน และแบบ

เส้นทางเดินอิสระ ซึ่งจะมีการใช้เซนเซอร์หลายชนิดหลายแบบมารวมกัน เช่น ระบบจีพีเอส (Global Positioning System : GPS) สัญญาณวิทยุและเลเซอร์

2.1.4 ส่วนติดต่อสื่อสารข้อมูลหรือเชื่อมต่อกับคอมพิวเตอร์ จะมีการเชื่อมต่อกันทั้งแบบระบบเครือข่ายไร้สาย(Local Area Network : LAN)และระบบคลื่นวิทยุซึ่งการเชื่อมต่อกับระบบคอมพิวเตอร์ดังกล่าวมีไว้เพื่อ การจัดการกับระบบทั้งหมด

## 2.2 ชนิดของเอจีวี

2.2.1 เอจีวีแบบลากจูง (AGV Towing Vehicle) เป็นแบบชนิดของเอจีวีที่มีความนิยมนามากชนิดหนึ่ง เอจีวีนี้จะเป็นอุปกรณ์นำทางอัตโนมัติ แบบลากจูง มีความกว้างให้เลือกหลายชนิด ชนิดของอุปกรณ์บรรทุกถูกจัดสำหรับบรรทุกและ ปลอดภัยของสัมภาระ รถพ่วง(Trailer) ยังรวมถึงเอจีวีที่ใช้จูงรถไฟ เคน อุปกรณ์ขนส่งอัตโนมัติ คนทำงานกันเอง การส่งผ่าน โดยรถไฟและอุปกรณ์ที่โปรแกรมการบรรทุกและปลดแบบ อัตโนมัติ[2]

โดยมากการประยุกต์การลากจูง จะเป็นในลักษณะของการเคลื่อนที่ของหีบผลิตภัณฑ์ให้ ไปสู่ภายนอกโกดัง ซึ่งการลากจูงจะใช้กับการขนส่งที่ปริมาณมาก ๆ โดยมีระยะทางขนส่งที่ไกล มากกว่า 1000 เมตร

2.2.2 เอจีวีแบบบรรทุกขนาดเล็ก (AGV Light Load Transporter) เป็นเอจีวีที่มีความสามารถในการบรรทุกได้น้อยกว่า 500 ปอนด์ ใช้ในการขนส่งที่มีขนาดเล็กและเบา มีความยาวพอประมาณใช้ในการขนถ่ายระหว่างที่จัดเก็บ โดยมีความเร็วปกติอยู่ที่ 100 ฟุต/นาที รัศมี การเลี้ยว 2 ฟุต มันถูกออกแบบมาเพื่อใช้ในการทำงานในพื้นที่จำกัดเช่นสายการประกอบ การ ใช้ในส่วนองไปรษณีย์[2]

2.2.3 เอจีวีแบบบรรทุกพาเลท(AGV Pallet Truck)เอจีวีแบบบรรทุกพาเลทถูกออกแบบมาเพื่อใช้ในการหลบหลีกและการขนส่งวัสดุที่อยู่บน แท่น พวกมันยังถูกใช้อย่างกว้างขวางในหลายหน้าที่ โดยที่อุปกรณ์นี้ถูกนำไปใช้ในการขนของ ขึ้นและลงในระดับชั้นต่าง ๆ โดยไม่เจาะจงสิ่งที่จะขนมาและไม่มีอุปกรณ์พิเศษใด ๆ ที่ต้อง นำมาใช้นอกเสียจากสิ่งของที่ขนนั้นต้องวางอยู่บนพาเลท[2]

## 2.3 การนำร่องของเอจีวี

### 2.3.1 การนำร่องโดยใช้ทางเดินนำร่อง

#### 2.3.1.1 การใช้แถบโลหะ (Metal Tape)

การใช้แถบโลหะดีเป็นแถบแบบเดียวกับแถบสี วิธีการนี้จะต้องเปลี่ยนตัวตรวจจับสีไปเป็นพรอกซิมีตี้(Proximity Sensor)แทน พรอกซิมีตี้เซนเซอร์จะให้เอาต์พุตออกมาในกรณีที่ตรวจพบ แถบโลหะเท่านั้น เอาต์พุตมีให้เลือกทั้งแบบที่เป็นอนาล็อกและดิจิทัล ข้อดีของวิธีการนี้คือ ความแม่นยำสูงคือ แม้มันฝุ่นละอองมาเกาะติดก็ยังทำงานได้ดี ข้อเสียของวิธีนี้คือ ราคาแพง การซ่อมบำรุงทำได้ยาก

#### 2.3.1.2 การใช้แถบแม่เหล็ก (Magnetic Tape)

ฝังลงในพื้นมีลักษณะเป็นตารางทั่ว ๆ ไปบนพื้นเอจีวี จะทำการตรวจจับแถบแม่เหล็ก ด้วยแมกเนติกเซนเซอร์(Magnetic Sensor) โดยเอจีวีจะเคลื่อนที่ไปตามแนวของเส้นแถบแม่เหล็ก จะเป็นตัวจับตำแหน่งในการเคลื่อนที่ ลักษณะของแมกเนติก เซนเซอร์นั้นประกอบด้วย ขดลวดกระตุ้น 1 ชุด (Exciting Coil) ชุดตรวจจับ 2 ชุด (Detecting Coil) การทำงานเริ่มจากขดลวดกระตุ้น ผลิตสนามแม่เหล็กไฟฟ้าโดยมีชุดตรวจจับคอยตรวจจับ

### 2.3.1.3 การใช้กล้องทีวีหรือซีซีดี (CCD Camera) เป็นตัวนำร่อง

โดยการนำสัญญาณรูปที่ได้รับซึ่งอาจเป็นดิจิทัลมาทำการประมวลผลเพื่อหาทางเดินของแถบที่ติดอยู่บนพื้นการนำร่องที่วิธีมีข้อดีคือสามารถหาตำแหน่งที่ถูกต้องของแถบสีได้แม้ว่าแถบสีนั้นจะมีการเลอะเลือนหรือชำรุดเสียหายไปบ้างก็ตามด้วยข้อดีที่กล่าวมาทำให้สามารถนำไปใช้กับการขนส่งภายนอกอาคารได้โดยเฉพาะการขนถ่ายวัสดุจากอาคารหนึ่งไปสู่อีกอาคารหนึ่ง แต่ข้อเสียก็คือความเร็วในการประมวลผลช้ามาก โดยเฉพาะการทำงานกับคอมพิวเตอร์ที่มีความเร็วในการทำงานต่ำแต่ปัญหาข้อนี้จะถูกแก้ไขด้วยการปรับปรุงหน่วยประมวลผลกลางให้มีประสิทธิภาพสูงขึ้น

### 2.3.2 การนำร่องแบบไร้สาย (Wireless Guidance)

#### 2.3.2.1 การนำทางด้วยรีโมทคอนโทรล (Remote Control Navigation)

วิธีการนี้จะทำการส่งข้อมูลทางดิจิทัลไปกับความถี่แสงย่านอินฟราเรด มีการนำเทคนิคเข้ารหัสต่าง ๆ มาใช้กันอย่างแพร่หลาย ข้อดีของวิธีการนี้คือราคาถูก ข้อเสียคือระยะทางในการควบคุมได้ไม่ไกล

#### 2.3.2.2 การใช้อัลตราโซนิกวัดระยะทาง (Ultrasonic Distance Measurement)

วิธีการนี้เอจิวี่จะเก็บแผนที่ของเส้นทางทุกทางที่เป็นไปได้ในหน่วยความจำวิธีการเคลื่อนที่จะเริ่มจากผู้กำหนดเป้าหมายปลายทางที่จะให้เอจิวี่ได้รับคำสั่งจะทำการคำนวณหาเส้นทางที่เหมาะสมที่สุดแล้วจะเคลื่อนที่ไปได้โดยใช้อัลตราโซนิก ทำหน้าที่คอยตรวจจับวัตถุที่ขวางเพื่อหาแนวทางในการเคลื่อนที่ ข้อดีของเทคนิคนี้คือความถูกต้องในระดับมิลลิเมตร ข้อเสียคือจำเป็นต้องมีสถานที่ที่มีลักษณะเป็นกำแพง ทำให้สิ้นเปลืองจำนวนของตัวตรวจจับและราคาแพงของตัวตรวจจับ ข้อเสียอีกอย่างหนึ่งคือตัวตรวจจับประเภทอัลตราโซนิกจะมีความถูกต้องมากที่สุด ในขณะที่ทำงานในอุณหภูมิต่ำหากมีการถ่ายเทอากาศไม่ดี หรือการถูกรบกวนของแหล่งกำเนิดที่มีความถี่สูงอาทิเช่นมอเตอร์หรือเครื่องจักรที่กำลังทำงาน[3]

## 2.4 บอร์ด Arduino Uno R3

คำว่า Uno เป็นภาษาอิตาลี ซึ่งแปลว่าหนึ่ง เป็นบอร์ด Arduino รุ่นแรกที่ยังคงมีขนาดประมาณ 68.6x53.4mm เป็นบอร์ดมาตรฐานที่นิยมใช้งานมากที่สุด เนื่องจากเป็นขนาดที่เหมาะสมสำหรับการเริ่มต้นเรียนรู้ Arduino และมี Shields ให้เลือกใช้งานได้มากกว่าบอร์ด Arduino รุ่นอื่น ๆ ที่ออกแบบมาเฉพาะมากกว่า โดยบอร์ด Arduino Uno ได้มีการพัฒนาเรื่อยมา ตั้งแต่ R2 R3 และรุ่นย่อยที่เปลี่ยนชิปไอซีเป็นแบบ SMD[4]

### 2.4.1 คุณสมบัติทางเทคนิคของบอร์ด Arduino Uno R3

- 1).ชิปไอซีไมโครคอนโทรลเลอร์ ATmega328
- 2).ใช้แรงดันไฟฟ้า 5V
- 3).รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ) 7 – 12V
- 4).รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด) 6 – 20V
- 5).พอร์ต Digital I/O 14 พอร์ต (มี 6 พอร์ต PWM output)
- 6).พอร์ต Analog Input 6 พอร์ต
- 7).กระแสไฟฟ้าที่จ่ายได้ในแต่ละพอร์ต 40mA
- 8).กระแสไฟฟ้าที่จ่ายได้ในพอร์ต 3.3V 50mA
- 9).พื้นที่โปรแกรมภายใน 32KB พื้นที่โปรแกรม, 500B ใช้โดย Bootloader
- 10).พื้นที่แรม 2KB
- 11).พื้นที่หน่วยความจำถาวร (EEPROM) 1KB
- 12).ความถี่คริสตัล 16MHz
- 13).ขนาด 68.6x53.4 mm
- 14).น้ำหนัก 25 กรัม



## 2.5 ไดรฟ์ขับเคลื่อนมอเตอร์ (L298N Motor Driver Module 2A)

วงจร H-Bridge ของ L298N จะขับเคลื่อนมอเตอร์ตามพัลส์ที่กำหนดด้วยลอจิกเพื่อควบคุมทิศทาง ส่วนความเร็วของมอเตอร์นั้นจะถูกควบคุมด้วย สัญญาณ (PWM Pulse Width Modulation)

โค้ดการทำงานของ Arduino มีการพัฒนาขึ้นเรื่อย ๆ แต่ยังไม่มีการสร้าง library ที่เกี่ยวกับตัว L298N Dual H-Bridge เพื่อควบคุมมอเตอร์ ดังนั้นผู้ใช้งานจึงต้องประกาศ pin เพื่อใช้งานขึ้นมาเองสามารถใช้โค้ด `int dir(number)Pin(letter)` ต่อเข้ากับพินดิจิทัลที่เลือกใช้ แค่นี้ก็สามารถทำงานได้อย่างถูกต้องและช่วยให้ตัว L298N Dual H-Bridge ควบคุมมอเตอร์ได้อย่างอเนกประสงค์ถ้าบอร์ด Arduino ที่เลือกใช้งานมีพินหลายตัว และถ้าต้องการปรับความเร็วของมอเตอร์ด้วย PWM สามารถใช้คำสั่ง `int speedPin(letter)` แล้วต่อเข้ากับพินที่เลือกใช้ หากต้องการวิธีลัดเพื่อใช้งาน PWM อย่างรวดเร็วสามารถเลือกพินใช้งานได้ตามรายการด้านล่าง AT MEGA –PWM 2-13 และ 44-46 ตั้งค่าเอาต์พุตของ PWM ให้เป็น 8 bit ด้วยฟังก์ชัน `analogWrite()` UNO-PWM 3, 5, 6, 9, 10 และ 11 ตั้งค่าเอาต์พุตของ PWM ให้เป็น 8 bit ด้วยฟังก์ชัน `analogWrite()`[5]



รูปที่ 2.3 แสดงรายละเอียดของบอร์ด

### 2.5.1 รายละเอียดของบอร์ด

Out 1: ช่องต่อขั้วไฟของมอเตอร์ A

Out 2: ช่องต่อขั้วไฟของมอเตอร์ A

Out 3: ช่องต่อขั้วไฟของมอเตอร์ B

Out 4: ช่องต่อขั้วไฟของมอเตอร์ B

12V: ช่องจ่ายไฟเลี้ยงมอเตอร์ 12V (ต่อได้ตั้งแต่ 5V ถึง 35V)

GND: ช่องต่อไฟลบ (Ground)

5V: ช่องจ่ายไฟเลี้ยงมอเตอร์ 5V (หากมีการต่อไฟเลี้ยงที่ช่อง 12V แล้ว

ช่องนี้จะทำหน้าที่จ่ายไฟออก เป็น 5V Output

สามารถต่อไฟจากช่องนี้ไปเลี้ยงบอร์ด Arduino ได้

ENA: ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ A

IN1: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A

IN2: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A

IN3: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B

IN4: ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B

ENB: ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ B

### 2.5.2 คุณสมบัติทางเทคนิค

Dual H bridge Drive Chip : L298N

แรงดันสัญญาณลอจิก : 5V Drive voltage: 5V-35V

กระแสของสัญญาณลอจิก : 0-36mA

กระแสขั้วมอเตอร์ : สูงสุดที่ 2A (เมื่อใช้มอเตอร์เดี่ยว)

กำลังไฟฟ้าสูงสุด : 25W

ขนาด : 43 x 43 x 26 มิลลิเมตร

น้ำหนัก : 26 กรัม

\*มี Power Supply 5V ในตัว สามารถจ่ายไฟออกจากช่อง 5V (เพื่อจ่ายให้บอร์ด Arduino) ได้ เมื่อต่อไฟเลี้ยงเข้าที่ช่อง 12V

### 2.5.3 การต่อเข้าใช้งานกับ Arduino

ในการต่อกับ Arduino นั้น ขา IN1,IN2,IN3 และ IN4 นั้น สามารถต่อกับพอร์ต Digital ใด ๆ ก็ได้ เนื่องจาก 4 ขานี้ จะใช้ในการควบคุมสัญญาณลอจิกบอกลทิศทางให้กับมอเตอร์ ส่วน ENA และ ENB นั้น จำเป็นที่จะต้องต่อกับพอร์ต Digital ที่รองรับ PWM เนื่องจากจะต้องใช้สัญญาณ PWM ในการควบคุมความเร็วของมอเตอร์[5]



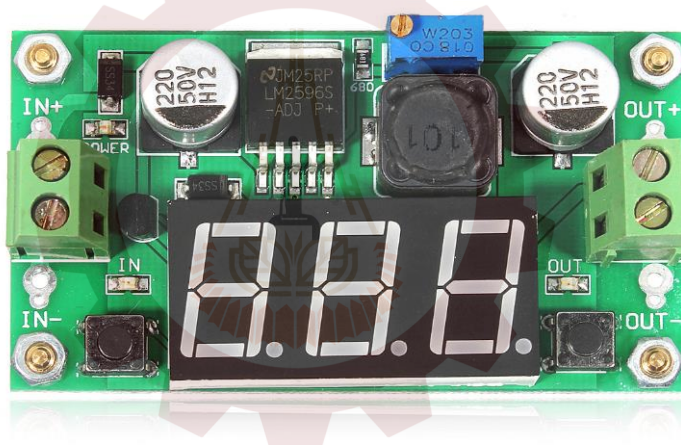
รูปที่ 2.4 แสดงการต่อใช้งานอุปกรณ์

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## 2.6 รีเกิลเตอร์ (Regulator)

รีเกิลเตอร์ (Regulator) ทำหน้าที่ในการจ่ายไฟตรงให้กับ โหลดและสามารถรักษาระดับแรงดันให้คงที่ได้โดยจะต่ออยู่ระหว่างแหล่งจ่ายไฟกระแสตรง กับ โหลด หรือ ยกตัวอย่างง่ายๆก็คือ สมมติว่าเรามีแหล่งจ่ายอยู่ 12 โวลต์ แต่ว่าวงจรของเราต้องการไฟเพียง 10 โวลต์ เราก็สามารถนำเจ้าตัวรีเกิลเตอร์ นี้มาใช้ต่อเพื่อจ่ายให้กับวงจรของเราได้นั่นเอง[6]



รูปที่ 2.5 รีเกิลเตอร์ (Regulator)

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

### บทที่ 3

#### การออกแบบระบบ

ในบทนี้จะนำเอาแนวทางการแก้ปัญหาที่เกิดขึ้นมาออกแบบระบบและวิเคราะห์เพื่อใช้การแก้ปัญหาจากบทที่ 1 โดยรายงานฉบับนี้จะเป็นการพัฒนาซอฟต์แวร์ โปรแกรมและฮาร์ดแวร์ขึ้นมาเพื่อจะนำไปแก้ไขปัญหาและความต้องการดังกล่าว

#### 3.1 โครงสร้างและการออกแบบระบบ

ระบบสามารถแบ่งออกเป็น 2 ส่วนใหญ่ ๆ ดังนี้

- 3.1.1 ส่วนซอฟต์แวร์ (Software) ส่วนนี้จะประกอบด้วยการทำ โปรแกรมหลัก และโปรแกรมย่อยเพื่อควบคุมอุปกรณ์ส่วนต่าง ๆ
- 3.1.2 ส่วนฮาร์ดแวร์ (Hardware) เป็นการประกอบอุปกรณ์ ให้เป็น ชิ้นงาน



รูปที่ 3.1 องค์ประกอบของระบบโดยรวม

จากรูปที่ 3.1 แสดงให้เห็น การทำงานของระบบจะติดต่อกับผู้ใช้ผ่านการกดปุ่ม เพื่อส่งคำสั่งให้กับบอร์ด Arduino สั่งการเซ็นเซอร์และมอเตอร์ ทำงาน ตามที่กำหนดไว้

### 3.2 รายละเอียดโครงสร้างขององค์ประกอบ

รายละเอียดของโครงสร้างและองค์ประกอบของแต่ละส่วน ได้มีการอธิบายรายละเอียดในหัวข้อต่าง ๆ ทั้ง 3 หัวข้อ ดังต่อไปนี้

3.2.1 ส่วนเชื่อมต่อผู้ใช้งาน (User) ส่วนนี้จะเป็นการควบคุมคอนโทรลการทำงาน ผ่านช่องทางการกดปุ่ม จากนั้นก็จะเข้าสู่หน้าเมนูของระบบในการเข้าไปสั่งการระบบ เช่น การสั่งให้รถวิ่งไปยังจุดที่กำหนด

3.2.2 ส่วนควบคุมวงจรระบบและประมวลผลของเครื่อง (Programing) ในส่วนนี้จะเป็นการรับเอาคำสั่ง จากส่วนผู้ใช้งาน โดยจะรับเอาคำสั่งจากปุ่มกดเป็นผู้จัดการ ติดต่อ โปรแกรมที่สร้างด้วยภาษา C++ ในตัวบอร์ด Arduino ทำงาน

3.2.3 ส่วนควบคุมการ ON /OFF เครื่องในส่วนนี้จะมีหน้าที่หลักควบคุมวงจร ทำให้ระบบไฟฟ้าครบวงจร ส่งผล ให้มอเตอร์และเซนเซอร์ทำงานตามโปรแกรมที่ควบคุม

### 3.3 การออกแบบการเชื่อมต่อใช้งานกันระหว่าง บอร์ด Arduino กับ เซนเซอร์และมอเตอร์

หลักการทำงานอยู่ 2 ส่วนที่สำคัญ มีดังนี้.

#### 3.3.1 อุปกรณ์ บอร์ด Arduino

ส่วนนี้จะเป็นการใช้คำสั่งในการโปรแกรมสั่งงานออกไปเพื่อให้เซนเซอร์และมอเตอร์ทำงาน โดยการเขียนโปรแกรมสั่งงานด้วยภาษา C, C++ ควบคุม Pin Number ของ บอร์ด Arduino ให้ทำงาน ในรูปแบบเป็นสัญญาณ Digital คือ 1 กับ 0 นั้นเอง

#### 3.3.2 เซนเซอร์และมอเตอร์

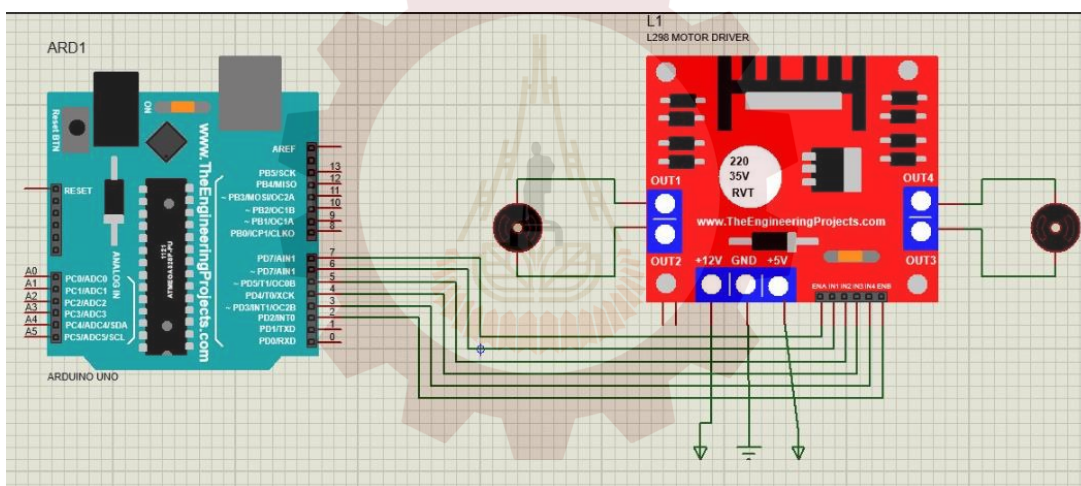
ส่วนนี้เป็นวงจรอิเล็กทรอนิกส์ โดยจะมีหลักการทำงานเชื่อมต่อสัญญาณ กับทาง Pin Number ของ บอร์ด Arduino โดยจะมีอุปกรณ์ดังนี้

- มอเตอร์ชนิดทวดเฟือง 12 VDC ความเร็วรอบ 500 รอบต่อนาที 2 ตัว
- ชุดขับเคลื่อนมอเตอร์ (Driver Motor) 1 ชุด
- ชุดปรับแรงดันไฟฟ้า (Regulator) 6 VDC 1 ชุด
- บอร์ด Arduino 1 ชุด
- Battery 12 VDC 1 ตัว

- เส้นแถบแม่เหล็ก ยาว 10 เมตร 1 เส้น
- Magnetic sensor 3 ตัว

3.4 การออกแบบ Hardware ในส่วนของวงจร

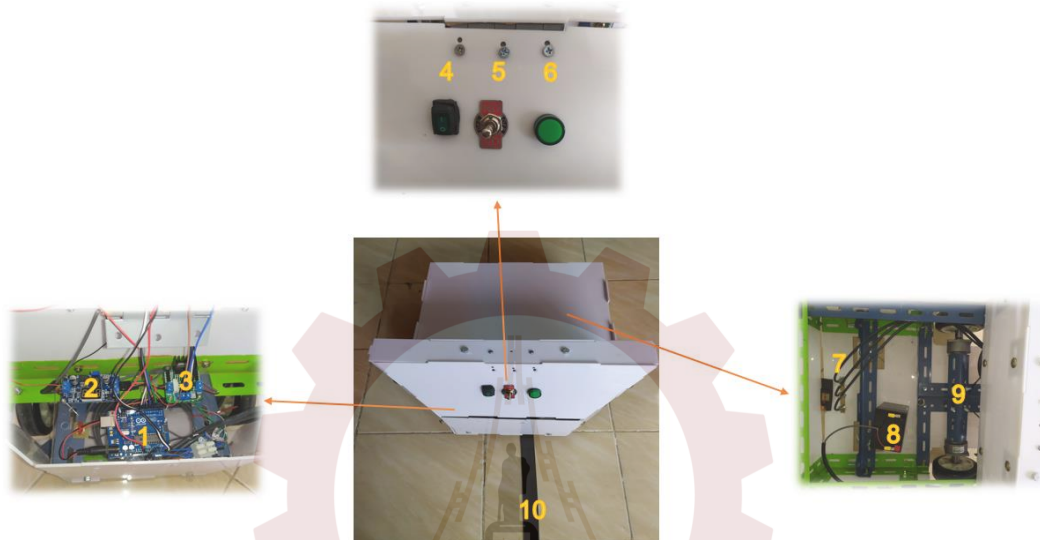
ส่วนนี้เป็นส่วนของการออกแบบวงจรและการประกอบวงจร เพื่อนำไปควบคุมระบบการจ่ายไฟฟ้าให้กับมอเตอร์ไฟฟ้ากระแสตรง ขนาด 12 โวลต์ และเซนเซอร์ทั้ง 3 ตัว



รูปที่ 3.2 การออกแบบวงจรควบคุมมอเตอร์โดยรวม

จากรูปที่ 3.2 แสดงวงจรควบคุมมอเตอร์โดยรวม โดยมีหลักการทำงานดังนี้ บอร์ด Arduino

สั่ง Motor Driver โดยการส่งสัญญาณ PWM ไปที่ขา ENA และ ENB เพื่อควบคุมความเร็วของและส่งสัญญาณดิจิทัล เข้าที่ขา IN1 IN2 IN3 IN4 ในการควบคุมทิศทางหมุนของมอเตอร์



รูปที่ 3.3 ชุดอุปกรณ์และวงจรไฟฟ้าโดยรวม

จากรูปที่ 3.2 แสดงชุดอุปกรณ์ และการจางจรไฟฟ้า ที่ใช้จริง โดยสามารถ แจกแจงรายละเอียดได้ดังนี้

หมายเลข 1 แสดงตำแหน่งของบอร์ด Arduino

หมายเลข 2 แสดงตำแหน่งเร็กกูเลเตอร์ (Regulator) ทำหน้าที่ปรับความเสถียรของไฟฟ้ากระแสตรง

หมายเลข 3 แสดงตำแหน่งของตัวขับมอเตอร์ (Motor driver) ทำหน้าที่ควบคุมการทำงานของมอเตอร์

หมายเลข 4 แสดงตำแหน่งของสวิตช์ ปิด-เปิด ทำหน้าที่ในการปิด-เปิดวงจร

หมายเลข 5 แสดงตำแหน่งของสวิตช์ 3 ทาง ทำหน้าที่สั่งให้รถวิ่งไปในจุดที่กำหนด

หมายเลข 6 แสดงตำแหน่งของไฟแสดงสถานะ ทำหน้าที่แสดงสถานะเมื่อทำงาน

หมายเลข 7 แสดงตำแหน่งของเซ็นเซอร์แม่เหล็ก ทำหน้าที่ตรวจจับเส้นแถบแม่เหล็ก

หมายเลข 8 แสดงตำแหน่งของแบตเตอรี่ 12 V ทำหน้าที่จ่ายไฟให้วงจร

หมายเลข 9 แสดงตำแหน่งของมอเตอร์ ทำหน้าที่ในการขับเคลื่อน

หมายเลข 10 แสดงตำแหน่งของเส้นแถบแม่เหล็ก ทำหน้าที่นำทางรถให้วิ่งไปยังจุดที่ต้องการ



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## บทที่ 4

### การทดลองและผลการทดลอง

จากการที่ได้คิดและออกแบบไว้จากบทที่ผ่านมานั้นทำให้มีการสร้างวงจรและ การการควบคุมเพื่อใช้ในการแก้ปัญหาตามแนวความคิดซึ่งแสดงให้เห็นถึงกระบวนการ ทำงานของเชื่อมต่อบอร์ด Arduino กับมอเตอร์และเซ็นเซอร์

#### 4.1 การทดลองแบ่งออกเป็น 2 ตอน ดังนี้

4.1.1 ทดลองใช้บอร์ด Arduino ในการสั่งงานมอเตอร์ทั้ง 2 ตัว โดยผ่านการควบคุมจาก Motor Driver จะแบ่งเป็นทั้งหมด 3 ตัวเลือก ได้แก่สั่งมอเตอร์เดินหน้า 2 วินาที ถอยหลัง 2 วินาที และหมุนซ้าย-ขวา 2 วินาทีแล้วหยุดการทดสอบนี้จะเป็นการสั่งงานจากบอร์ดอาดูโน่โดยตรง โดยเขียน Code ให้เป็นตัวสั่งให้บอร์ดทำงาน เพื่อดูผลการทดลอง

4.1.2 ทดลองใช้บอร์ด Arduino ในการสั่งงานเซ็นเซอร์ทั้ง 3 ตัว โดยดูสถานะ Input จากเซ็นเซอร์ผ่านหน้าจอมอนิเตอร์เมื่อเซ็นเซอร์ตรวจจับได้จะแสดงเป็นสัญญาณ Signal คือ 1 และ 0 การทดสอบนี้จะเป็นการสั่งงานจากบอร์ดอาดูโน่โดยตรง โดยเขียน Code ให้เป็นตัวสั่งให้บอร์ดทำงาน เพื่อดูผลการทดลอง



รูปที่ 4.1 แสดงรถขนส่งสินค้าอัตโนมัติ

#### 4.1.1 การทดสอบการหมุนมอเตอร์



รูปที่ 4.2 แสดงการทดสอบการหมุนของมอเตอร์

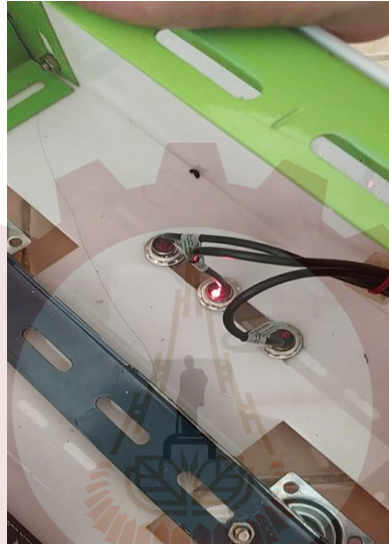
จากรูปที่ 4.2 แสดงการหมุนของมอเตอร์ โดยสั่งการทำงานจากหน้า Main Program จนกระทั่งหยุดหมุน

ตารางที่ 4.1 แสดงผลการทดสอบการหมุนของมอเตอร์

| ลำดับการหมุนของมอเตอร์ | เวลาที่ใช้(วินาที) | ผลการทดสอบ |
|------------------------|--------------------|------------|
| เดินหน้า               | 2                  | ผ่าน       |
| ถอยหลัง                | 2                  | ผ่าน       |
| หมุนซ้าย-ขวา           | 2                  | ผ่าน       |

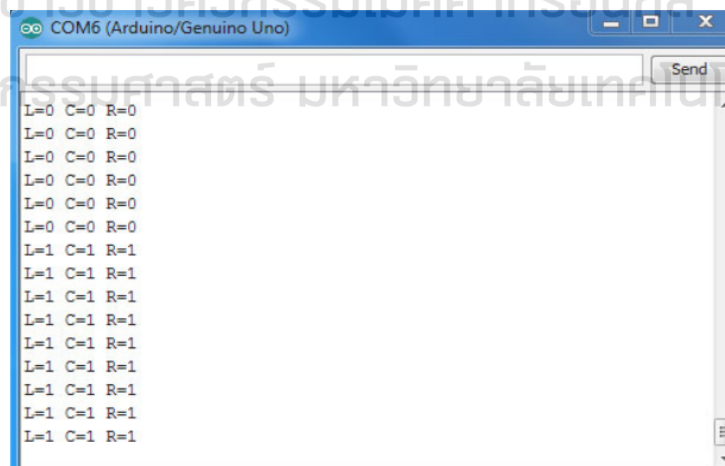
จากตารางที่ 4.1 แสดงผลลัพธ์ที่ได้จากการทดสอบ ว่าสามารถทำงานได้ตรงคำสั่งที่โปรแกรมไว้

#### 4.1.2 การทดสอบสถานะของเซ็นเซอร์



รูปที่ 4.3 แสดงสถานะของเซ็นเซอร์

จากรูปที่ 4.3 เป็นการตรวจสอบสถานะของเซ็นเซอร์เมื่อสามารถตรวจจับวัตถุที่เป็นแม่เหล็ก เมื่อตรวจพบแสดงสถานะไฟจะติด และคัมเมื่อไม่สามารถตรวจพบ



รูปที่ 4.4 แสดงสัญญาณของเซ็นเซอร์

จากรูปที่ 4.4 เป็นการตรวจสอบสัญญาณของเซ็นเซอร์เมื่อสามารถตรวจจับวัตถุที่เป็นแม่เหล็กเมื่อตรวจพบจะแสดงสัญญาณเป็น 1 และ 0 ไม่สามารถตรวจพบ

ตารางที่ 4.2 แสดงผลการทดสอบสถานะและแสดงสัญญาณของเซ็นเซอร์ตัวที่ 1

| ครั้งที่ | การตรวจจับวัตถุที่เป็นแม่เหล็ก | ไฟสถานะ | สัญญาณ | ผลการทดสอบ |
|----------|--------------------------------|---------|--------|------------|
| 1        | ตรวจพบ                         | ติด     | 0-1    | ไม่ผ่าน    |
| 2        | ตรวจพบ                         | ติด     | 1      | ผ่าน       |
| 3        | ตรวจไม่พบ                      | ไม่ติด  | 0      | ผ่าน       |

จากตารางที่ 4.2 แสดงผลลัพธ์ที่ได้จากการทดสอบเซ็นเซอร์ตัวที่ 1 โดยในครั้งที่ 1 สัญญาณที่แสดงออกมานั้นแสดง 0 และ 1 สลับกันแก้ไขโดยต่อความต้านทานและทดสอบอีกครั้ง

ตารางที่ 4.3 แสดงผลการทดสอบสถานะและแสดงสัญญาณของเซ็นเซอร์ตัวที่ 2

| ครั้งที่ | การตรวจจับวัตถุที่เป็นแม่เหล็ก | ไฟสถานะ | สัญญาณ | ผลการทดสอบ |
|----------|--------------------------------|---------|--------|------------|
| 1        | ตรวจพบ                         | ติด     | 1      | ผ่าน       |
| 2        | ตรวจไม่พบ                      | ไม่ติด  | 0      | ผ่าน       |
| 3        | ตรวจพบ                         | ติด     | 1      | ผ่าน       |

จากตารางที่ 4.3 แสดงผลลัพธ์ที่ได้จากการทดสอบเซ็นเซอร์ตัวที่ 2

ตารางที่ 4.4 แสดงผลการทดสอบสถานะและแสดงสัญญาณของเซ็นเซอร์ตัวที่ 3

| ครั้งที่ | การตรวจจับวัตถุที่เป็นแม่เหล็ก | ไฟสถานะ | สัญญาณ | ผลการทดสอบ |
|----------|--------------------------------|---------|--------|------------|
| 1        | ตรวจไม่พบ                      | ไม่ติด  | 0      | ผ่าน       |
| 2        | ตรวจไม่พบ                      | ไม่ติด  | 0      | ผ่าน       |
| 3        | ตรวจพบ                         | ติด     | 1      | ผ่าน       |

จากตารางที่ 4.4 แสดงผลลัพธ์ที่ได้จากการทดสอบเซ็นเซอร์ตัวที่ 3



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## บทที่ 5

### สรุปผลการทดลอง

การนำเทคโนโลยีมาประยุกต์ใช้กับชีวิตประจำวันนั้น ส่งผลให้ชีวิตปัจจุบัน มีความคล่องตัวมากขึ้น ตอบสนองภารกิจที่จะต้องทำได้รวดเร็วเสมือนทำด้วยตนเอง ดังจะเห็นได้จาก รายงานชุดนี้ที่นำเอา Micro controller อย่าง บอร์ด Arduino มาควบคุมการทำงานของ Magnetic sensor เพื่อส่งมอเตอร์วิ่งไปยังจุดที่กำหนดได้อย่างถูกต้อง โดยการสั่งงานจากการกดปุ่ม

#### 5.1 ประโยชน์จากการทำโครงการ

5.1.1 ลดระยะเวลาในการขนส่งชิ้นส่วนและอุปกรณ์เพื่อลดเวลาในการซ่อมเครื่องจักรที่เกิดจากการรอชิ้นส่วนหรืออุปกรณ์ได้

5.1.2 เพื่อศึกษาการนำบอร์ด Arduino มาใช้ในการควบคุมมอเตอร์และเซนเซอร์ มาสร้างรถขนส่งสินค้าอัตโนมัติทางด้วยเส้นแถบแม่เหล็ก

#### 5.2 ปัญหาและอุปสรรคของรถขนส่งสินค้าอัตโนมัติ

ระยะการตรวจนับระหว่าง Magnetic sensor กับเส้นแถบแม่เหล็กน้อยทำให้การตั้งระยะมีความยุ่งยากและภาษาที่ใช้ในการเขียน โปรแกรมลงบนบอร์ด Arduino เป็นภาษาคอมพิวเตอร์ ทำให้ยากต่อผู้ที่ไม่คุ้นเคย

#### 5.3 แนวทางการนำไปพัฒนาต่อ

สำหรับรถขนส่งสินค้าอัตโนมัตินั้นเราสามารถนำไปพัฒนาต่อในเรื่องการควบคุมโดยผ่านแอปพลิเคชันบนมือถือ การเพิ่มจุดที่ใช้ในการรับ-ส่งสินค้า

สำหรับตัวบอร์ด Arduino นั้นข้อดีของอุปกรณ์ คือ มีความเล็กกะทัดรัด, ประหยัดพลังงานใช้ซอฟต์แวร์แบบฟรีลิขสิทธิ์ทั้งหมด, จึงเป็นอุปกรณ์ที่เป็นที่นิยมในการนำไปสร้างสรรค์ผลงานได้หลากหลาย

### เอกสารอ้างอิง

- [1] พิสิทธิ์ วิสุทธิเมธีกร. “AGV รถขนถ่ายวัสดุอัตโนมัติในโรงงานอุตสาหกรรม.” วารสาร Mechanical Technology. ปีที่ 3(กุมภาพันธ์ 2547) : 43-47.
- [2] จิรภาศ แซ่หลี (2560) “ระบบขนถ่ายวัสดุอัตโนมัติ” [ออนไลน์]. ได้จาก : [http://jirapas053.blogspot.com/2017/06/blog-post\\_24.html](http://jirapas053.blogspot.com/2017/06/blog-post_24.html)
- [3] การใช้ไมโครคอนโทรลเลอร์และเซนเซอร์อัตโนมัติ [ออนไลน์]. ได้จาก : <http://aimagin.com/blog/%E0%B8%AD%E0%B8%B1%E0%B8%A5%E0%B8%95%E0%B8%A3%E0%B8%B2%E0%B9%82%E0%B8%8B%E0%B8%99%E0%B8%B4%E0%B8%81-hc-sr04/?lang=th>
- [4] มานพ ปักยี. (2562) Arduino ขั้นพื้นฐานสำหรับผู้เริ่มต้น 1 สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย
- [5] การใช้งาน L298N motor drive [ออนไลน์]. ได้จาก : <http://gamerbloggerport.blogspot.com/2017/09/l298n-motor-drive.html>
- [6] วงจรรักษาระดับแรงดัน (Voltage Regulator) คืออะไร [ออนไลน์] ได้จาก : <https://phukphan.blogspot.com/2017/02/voltage-regulator.html>
- [7] Tanabodin Kamol (2018) “มารู้จัก PWM กันเถอะ” [ออนไลน์]. ได้จาก : <https://medium.com/creative-systems/to-know-pwm-373efa1322ce>

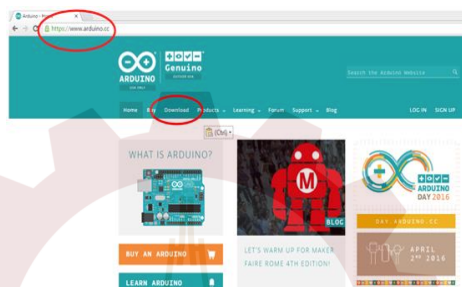


1. ขั้นตอนการติดตั้ง Arduino IDE และการใช้งานบอร์ด Arduino Uno R3
2. Pulse Width Modulation (PWM)

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## 1. ขั้นตอนการติดตั้ง Arduino IDE และการใช้งานบอร์ด Arduino Uno R3

1).ดาวน์โหลด Arduino IDE จาก <https://www.arduino.cc> โดยคลิกที่เมนู Download



รูปที่ ก.1 เว็บไซต์ที่ใช้ดาวน์โหลด Arduino IDE

2).เลือกระบบปฏิบัติการของเครื่องคอมพิวเตอร์ที่ท่านจะใช้ในการเขียนโปรแกรม Arduino



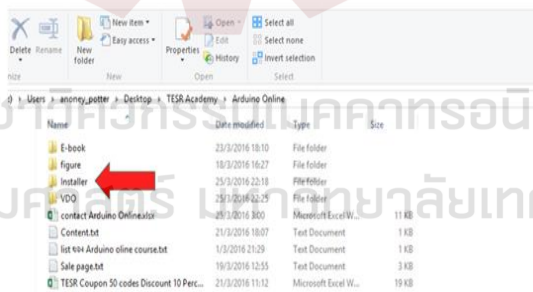
รูปที่ ก.2 เลือกระบบปฏิบัติการของเครื่องคอมพิวเตอร์

3). กด JUST DOWNLOAD (หากต้องการร่วมบริจาคช่วยการพัฒนา Arduino Software สามารถกด CONTRIBUTE & DOWNLOAD)



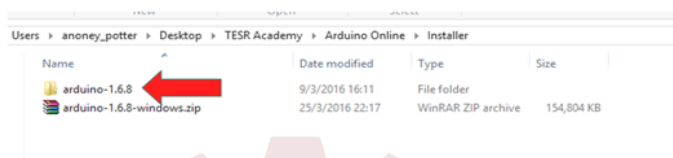
รูปที่ ก.3 คาวน์โหลด

4). บันทึกไฟล์ติดตั้งโปรแกรมไปยังโฟลเดอร์ที่ต้องการ



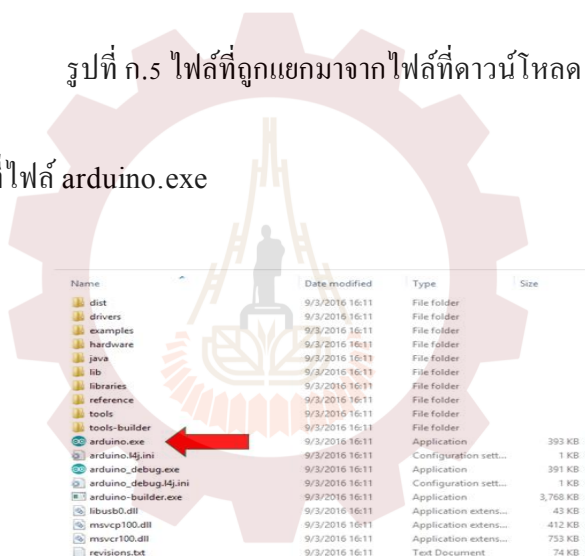
รูปที่ ก.4 ไฟล์ติดตั้งโปรแกรม

## 5).Unzip ไฟล์ติดตั้ง



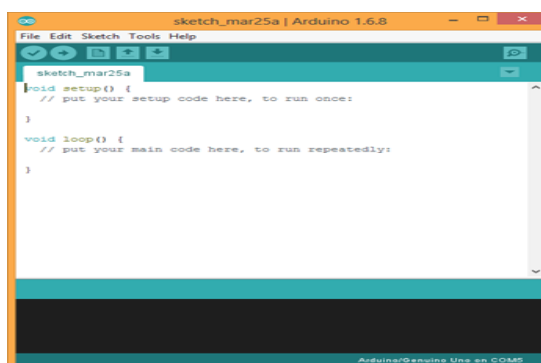
รูปที่ ก.5 ไฟล์ที่ถูกแยกมาจากไฟล์ที่ดาวน์โหลด

## 6).ดับเบิลคลิกที่ไฟล์ arduino.exe



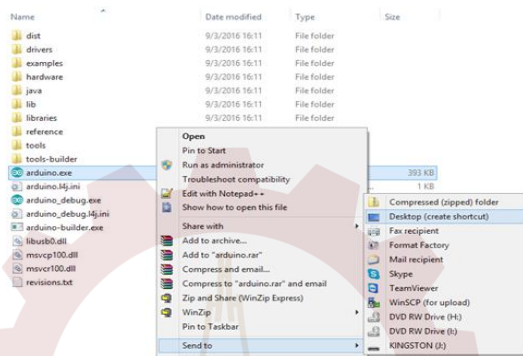
รูปที่ ก.6 ไฟล์เพื่อเข้าหน้าต่างโปรแกรม Arduino

## 7).หน้าต่างโปรแกรม Arduino จะปรากฏขึ้น ดังรูป



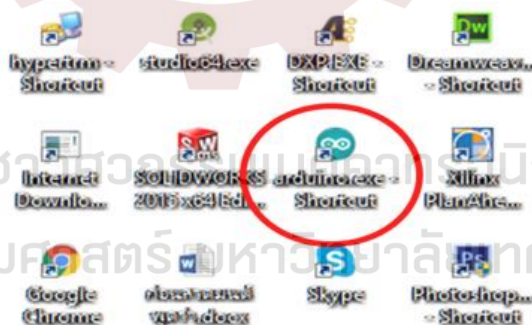
รูปที่ ก.7 หน้าต่างโปรแกรม

8).สร้าง Shortcut บน Desktop เพื่อความสะดวกในการเปิดโปรแกรมในครั้งต่อไป



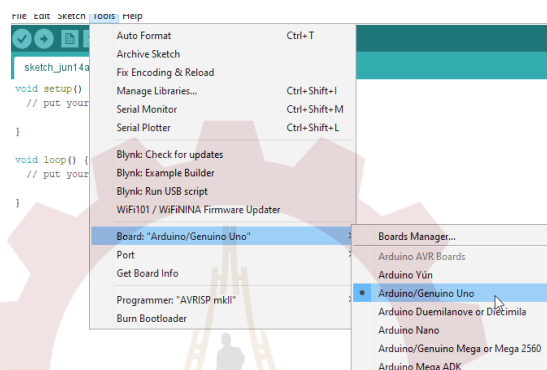
รูปที่ ก.8 สร้าง Shortcut บน Desktop

9).จะปรากฏไอคอนของโปรแกรม Arduino บน Desktop ดังแสดงในรูป



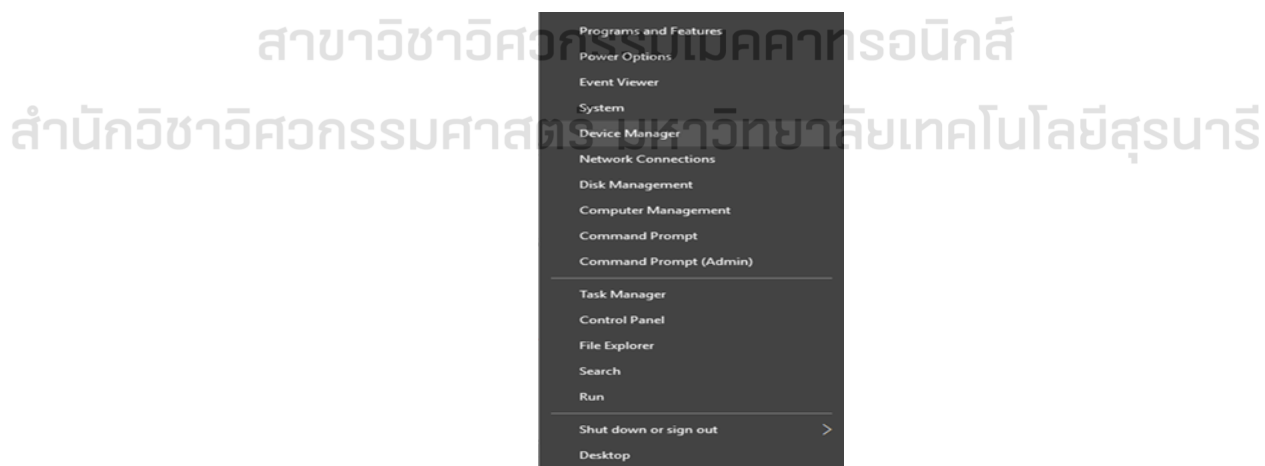
รูปที่ ก.9 ไอคอนของโปรแกรม Arduino บน Desktop

10). ในหน้าต่างโปรแกรม Arduino ให้คลิกไปที่เมนู Tools -> Board -> Arduino/Genuino Uno



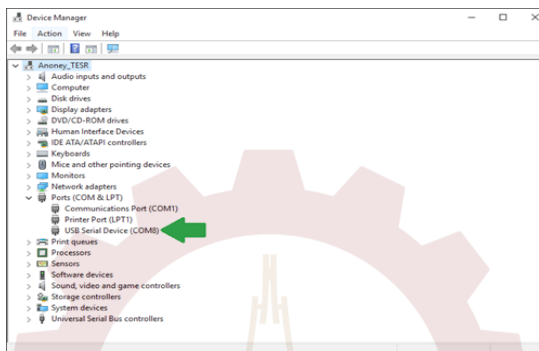
รูปที่ ก.10 ตั้งค่าบอร์ด Arduino

11). เสียบบอร์ด Arduino Uno R3 เข้ากับคอมพิวเตอร์ จากนั้นไปที่ Device Manager (หากเป็น Window ให้คลิกขวาที่ไอคอนบน Toolbars) เพื่อดูว่าบอร์ด Arduino Uno R3 นั้นต่ออยู่กับ COM Port หมายเลขใด



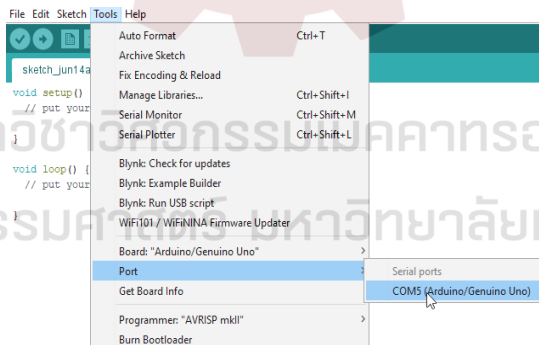
รูปที่ ก.11 เลือก Device Manager

12). ใน Device Manager คลิกที่ Ports เพื่อดูหมายเลข COM Port ดังภาพ



รูปที่ ก.12 หมายเลข COM Port

13). การเปิด Arduino IDE ขึ้นมาอีกครั้ง ไปที่เมนู Tools และทำการตั้งค่าบอร์ดและหมายเลขพอร์ตให้ตรงกับที่ได้จากขั้นตอนที่ 12 ซึ่งในตัวอย่างคือ COM8 ดังนี้



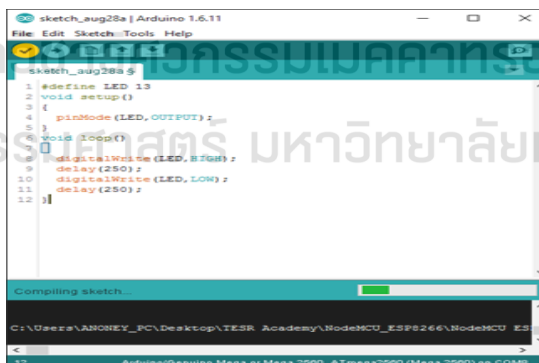
รูปที่ ก.13 การตั้งค่าหมายเลขพอร์ต

14). ในขั้นตอนนี้ เราจะทดลองเขียน โปรแกรมให้หลอดไฟ LED บนบอร์ดกระพริบเพื่อทดสอบการทำงานของบอร์ดเบื้องต้นดังนี้ โดยท่านสามารถคัดลอกโค้ดทั้งหมดข้างล่างไปวางทับลงในหน้าโปรแกรม

```
#define LED 13
void setup()
{
  pinMode(LED,OUTPUT);
}
void loop()
{
  digitalWrite(LED,HIGH);
  delay(250);
  digitalWrite(LED,LOW);
  delay(250);
}
```

รูปที่ ก.14 ทดสอบการทำงานของบอร์ดเบื้องต้น

15).คอมไพล์ (Compile) โดยคลิกที่ปุ่ม  เพื่อตรวจสอบว่าโค้ดที่เขียนไม่มีข้อผิดพลาด



รูปที่ ก.14 เพื่อตรวจสอบโค้ดที่เขียน

16). หากไม่มีข้อผิดพลาด จะปรากฏข้อความว่า “Done compiling” ดังภาพ



```

sketch_aug28a $
1 #define LED 13
2 void setup()
3 {
4   pinMode(LED, OUTPUT);
5 }
6 void loop()
7 {
8   digitalWrite(LED, HIGH);
9   delay(250);
10  digitalWrite(LED, LOW);
11  delay(250);
12 }

```

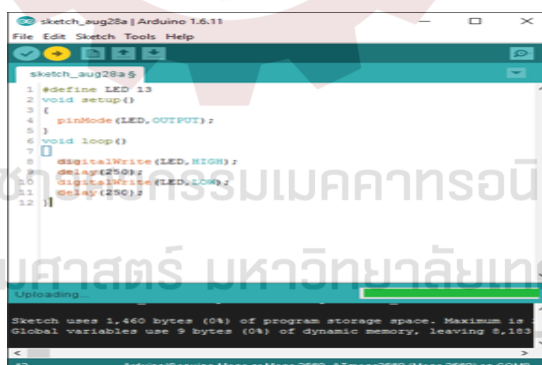
Done compiling

Sketch uses 1,460 bytes (0%) of program storage space. Maximum is Global variables use 9 bytes (0%) of dynamic memory, leaving 0,183

Arduino/Genuine Mega or Mega 2560, ATmega2560 (Mega 2560) on COM8

รูปที่ ก.16 โค้ดที่เขียน ไม่มีข้อผิดพลาด

17). จากนั้นให้ทำการคลิกที่ปุ่ม  เพื่อทำการอัปโหลดโค้ดเข้าสู่บอร์ด Arduino Uno R3



```

sketch_aug28a $
1 #define LED 13
2 void setup()
3 {
4   pinMode(LED, OUTPUT);
5 }
6 void loop()
7 {
8   digitalWrite(LED, HIGH);
9   delay(250);
10  digitalWrite(LED, LOW);
11  delay(250);
12 }

```

Uploading

Sketch uses 1,460 bytes (0%) of program storage space. Maximum is Global variables use 9 bytes (0%) of dynamic memory, leaving 0,183

Arduino/Genuine Mega or Mega 2560, ATmega2560 (Mega 2560) on COM8

รูปที่ ก.17 อัปโหลดโค้ดเข้าสู่บอร์ด Arduino Uno R3

18).หากไม่มีข้อผิดพลาด จะขึ้นคำว่า “Done uploading” ดังภาพ และไฟบนบอร์ดจะกระพริบ ทุก 250 ms (2 ครั้งต่อวินาที)

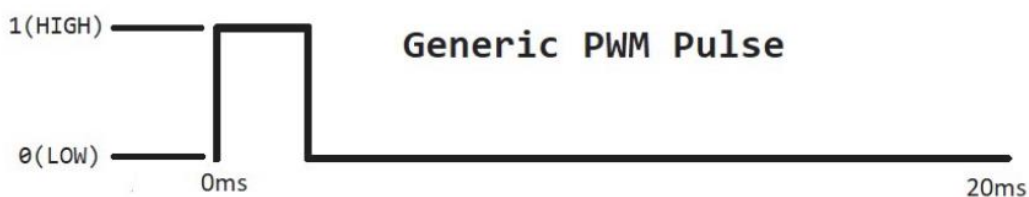


รูปที่ ก.18 อัฟโหลดสำเร็จ

## 2. Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) เป็นเทคนิคที่ทำให้เราสามารถอ่าน/เขียนข้อมูลแบบ analog ด้วยสัญญาณ digital ได้ โดยตัวควบคุมการสร้างสัญญาณดิจิทัล (Digital control) จะสร้างสัญญาณคลื่นสี่เหลี่ยมออกมาดังรูปที่ 1 โดยสัญญาณที่สร้างออกมามีระดับสลับกันระหว่าง เปิด(HIGH) กับ ปิด(LOW) รูปแบบสัญญาณเปิด-ปิดนี้สามารถจำลองเป็นแรงดันไฟฟ้าระหว่าง เปิด (5 Volts) กับ ปิด (0 Volts)

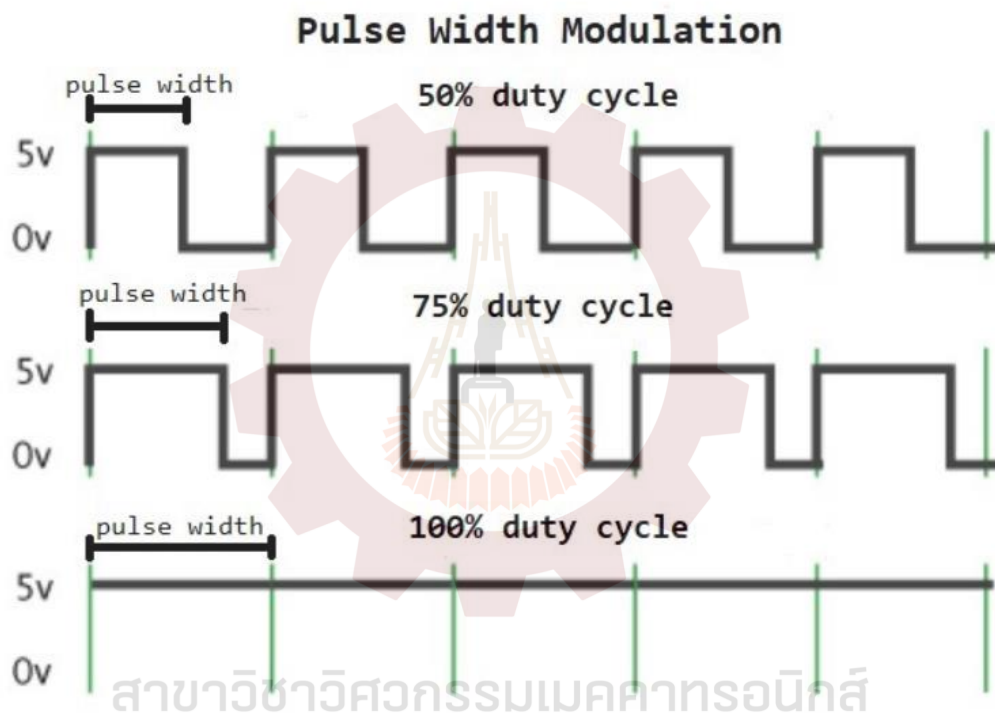
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ ก.19 แสดงสัญญาณเปิด(HIGH) และปิด(LOW)

จากรูปที่ ก.19 เราเรียกช่วงที่เป็นปิดหรือเปิดตามการทำงาน เพราะช่วงที่เป็นปิดแรงดันไฟฟ้าจะเป็น 0 ทำให้ไม่มีการทำงาน ส่วนช่วงที่เป็นเปิดคือช่วงที่มีแรงดันไฟฟ้า ทำให้อุปกรณ์ทำงานได้ ซึ่งช่วงของเวลาที่สัญญาณเป็นเปิด เมื่อเทียบกับช่วงเวลาที่สัญญาณออกมาทั้งหมด เราจะเรียกช่วงเวลาดังนี้ว่า

ความกว้างของสัญญาณ (Pulse width) ดังภาพที่ ก.20 ดังนั้นเพื่อให้ได้ข้อมูลแบบ analog ที่ต่างกัน เราจะใช้ความกว้างของสัญญาณที่แตกต่างกันในการบ่งบอกค่าของข้อมูล analog



รูปที่ ก.20 ตัวอย่างสัญญาณ PWM

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

จากรูปที่ ก.20 ความกว้างของเส้นสี่เหลี่ยมจะแสดงถึงช่วงเวลาปกติ ระยะเวลาหรือช่วงเวลานี้เรียกว่า period เป็นค่าผกผันของความถี่ของสัญญาณ หากความถี่ของสัญญาณมีค่าประมาณ 50 Hz (ความถี่ขึ้นอยู่กับอุปกรณ์) ความกว้างของเส้นสี่เหลี่ยมจะเป็นช่วงเวลา 20 ms (หาได้จาก  $T = 1/f$ ) [7]



**1. Code Program สำหรับทดสอบการหมุนของมอเตอร์**

**2. Code Program สำหรับทดสอบเซ็นเซอร์**

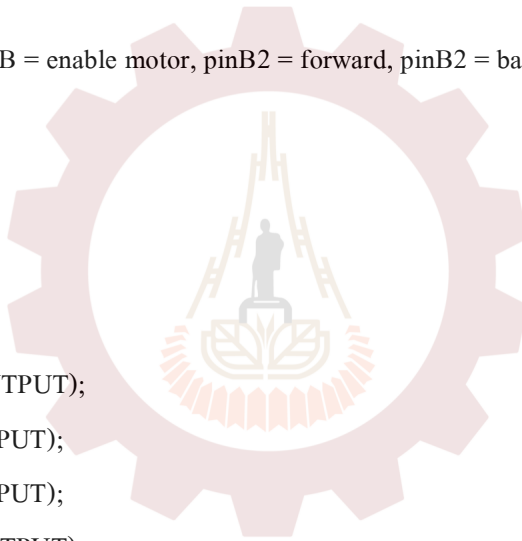
**3. Code Program ที่ใช้งานจริง**

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

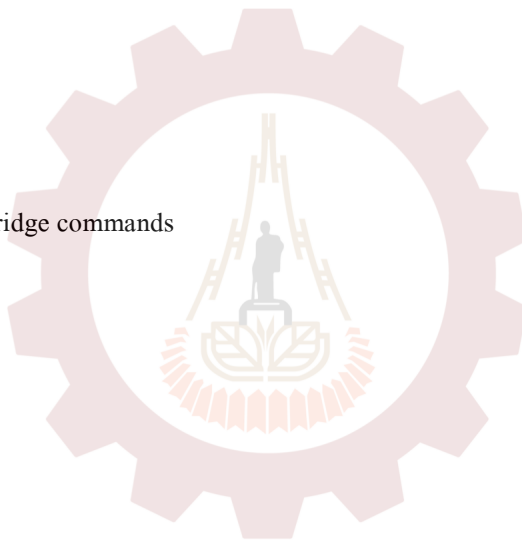
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

### 1. Code Program สำหรับทดสอบการหมุนของมอเตอร์

```
// Motor A pins (enableA = enable motor, pinA1 = forward, pinA2 = backward)
int enableA = 3;
int pinA1 = 6;
int pinA2 = 7;
//Motor B pins (enableB = enable motor, pinB1 = forward, pinB2 = backward)
int enableB = 5;
int pinB1 = 8;
int pinB2 = 9;
boolean run = true;
void setup() {
  pinMode(enableA, OUTPUT);
  pinMode(pinA1, OUTPUT);
  pinMode(pinA2, OUTPUT);
  pinMode(enableB, OUTPUT);
  pinMode(pinB1, OUTPUT);
  pinMode(pinB2, OUTPUT);
}
void loop() {
  if (run) {
    delay(2000);
    enableMotors();
    //Go forward
    forward(2000);
    //Go backward
    backward(2000);
    //Turn left
    turnLeft(400);
  }
}
```



```
coast(2000);
//Turn right
turnRight(400);
coast(2000);
//This stops the loop
run = false;
}
}
//Define high-level H-bridge commands
void enableMotors()
{
  motorAOn();
  motorBOn();
}
void disableMotors()
{
  motorAOff();
  motorBOff();
}
void forward(int time)
{
  motorAForward();
  motorBForward();
  delay(time);
}
void backward(int time)
{
  motorABackward();
```

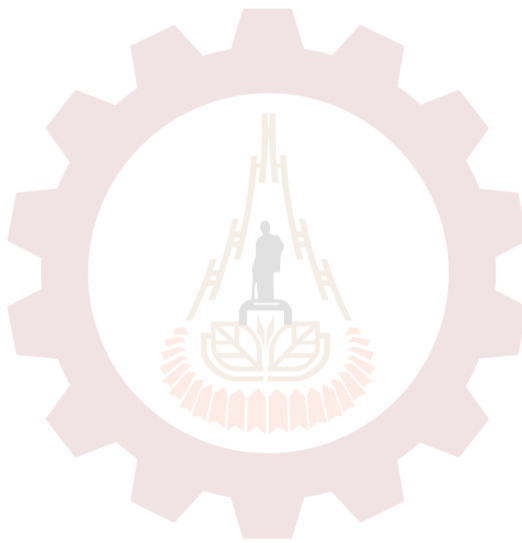


สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

```

motorBBackward();
delay(time);
}
void turnLeft(int time)
{
motorABackward();
motorBForward();
delay(time);
}
void turnRight(int time)
{
motorAForward();
motorBBackward();
delay(time);
}
void coast(int time)
{
motorAcoast();
motorBcoast();
delay(time);
}
void brake(int time)
{
motorABrake();
motorBBrake();
delay(time);
}
//Define low-level H-bridge commands

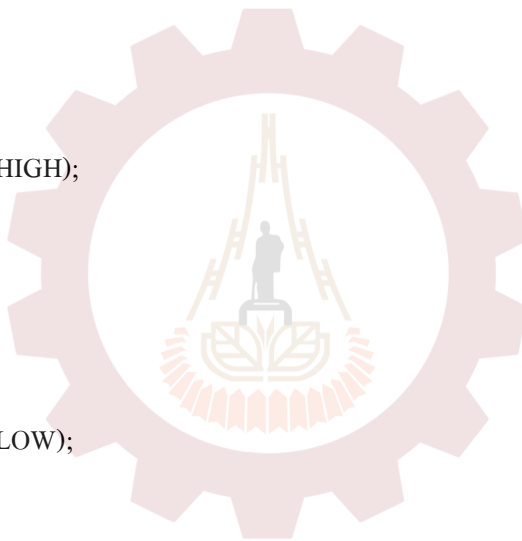
```



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
 สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

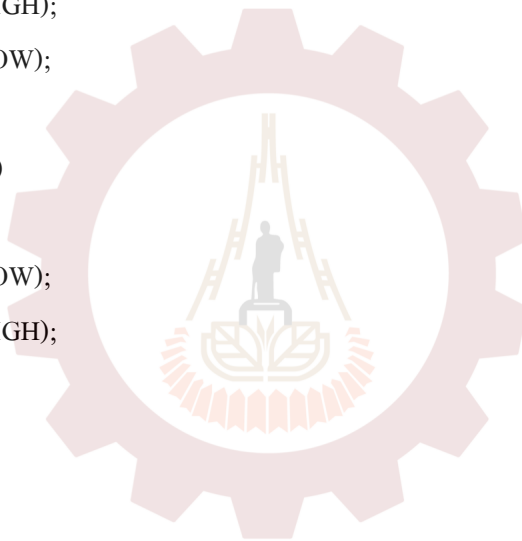
```
//enable motors
void motorAOn()
{
    digitalWrite(enableA, HIGH);
}
void motorBOn()
{
    digitalWrite(enableB, HIGH);
}
//disable motors
void motorAOff()
{
    digitalWrite(enableA, LOW);
}
void motorBOff()
{
    digitalWrite(enableB, LOW);
}

//motor A controls
void motorAForward()
{
    digitalWrite(pinA1, HIGH);
    digitalWrite(pinA2, LOW);
}
void motorABackward()
{
    digitalWrite(pinA1, LOW);
    digitalWrite(pinA2, HIGH);
}
```



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

```
}  
  
//motor B controls  
void motorBForward()  
{  
    digitalWrite(pinB1, HIGH);  
    digitalWrite(pinB2, LOW);  
}  
  
void motorBBackward()  
{  
    digitalWrite(pinB1, LOW);  
    digitalWrite(pinB2, HIGH);  
}  
  
//coasting and braking  
void motorACoast()  
{  
    digitalWrite(pinA1, LOW);  
    digitalWrite(pinA2, LOW);  
}  
  
void motorABrake()  
{  
    digitalWrite(pinA1, HIGH);  
    digitalWrite(pinA2, HIGH);  
}  
  
void motorBCoast()  
{  
    digitalWrite(pinB1, LOW);  
    digitalWrite(pinB2, LOW);  
}
```



สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

```
void motorBBrake()
{
  digitalWrite(pinB1, HIGH);
  digitalWrite(pinB2, HIGH);
}
```

## 2.Code Program สำหรับทดสอบเซ็นเซอร์

```
/*
  read analog data from A1-A5 and send to PC via Serial port
*/
int sensor_L , sensor_C , sensor_R , Switch_1 , Switch_2 ; //optical sensor values
String tmp ;
int ADC_stabilize = 5 ;
void setup () {
  // initialize the serial communication:
  Serial . begin (115200 ) ;
}
void loop () {
  //take a snapshot
  sensor_L = digitalRead ( A2 ) ;
  delay ( ADC_stabilize ) ; //stabilize
  sensor_L = digitalRead ( A2 ) ;
  delay ( ADC_stabilize ) ;
  sensor_C = digitalRead ( A3 ) ;
  delay ( ADC_stabilize ) ;
  sensor_C = digitalRead ( A3 ) ;
  delay ( ADC_stabilize ) ;
```

```

sensor_R = digitalRead ( A1 ) ;
delay ( ADC_stabilize ) ;
sensor_R = digitalRead ( A1 ) ;
delay ( ADC_stabilize ) ;
Switch_1 = digitalRead ( A4 ) ;
delay ( ADC_stabilize ) ;
Switch_1 = digitalRead ( A4 ) ;
delay ( ADC_stabilize ) ;
Switch_2 = digitalRead ( A5 ) ;
delay ( ADC_stabilize ) ;
Switch_2 = digitalRead ( A5 ) ;
delay ( ADC_stabilize ) ;
tmp = "L=" + String ( sensor_L ) + " C=" + String ( sensor_C ) + " R=" + String ( sensor_R ) + "
l=" + String ( Switch_1 ) + " 2=" + String ( Switch_2 ) ;
// send the value of analog inputs:
Serial . println ( tmp ) ;
// wait a bit for next reading
delay ( 1000 ) ; //1000=1 sec
}

```

### 3.Code Program ที่ใช้งานจริง

```

// Motor A pins (enableA = enable motor, pinA1 = forward, pinA2 = backward)
int enableA = 3;
int pinA1 = 6;
int pinA2 = 7;
//Motor B pins (enabledB = enable motor, pinB 1 = forward, pinB2 = backward)
int enableB = 5;
int pinB1 = 8;
int pinB2 = 9;

```

```

int putbutton[2] = {0, 0};
int sensor[3] = {0, 0, 0};
void setup()
{
  pinMode(enableA, OUTPUT);
  pinMode(pinA1, OUTPUT);
  pinMode(pinA2, OUTPUT);
  pinMode(enableB, OUTPUT);
  pinMode(pinB1, OUTPUT);
  pinMode(pinB2, OUTPUT);
}
void loop()
{
  enableMotors();
  analogWrite(enableA, 225); // ปรับค่าความเร็วของหุ่นยนต์
  analogWrite(enableB, 225); // ปรับค่าความเร็วของหุ่นยนต์
  sensor[0] = digitalRead(A2);
  sensor[1] = digitalRead(A3);
  sensor[2] = digitalRead(A1);
  putbutton[0] = digitalRead(A4);
  putbutton[1] = digitalRead(A5);
  if((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 0) && (putbutton[0] == 1)) // Point
  A
    forward(10);
  else if((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (putbutton[0] == 1)) //
  Point A
    turnRight(10);

```

```
else if ((sensor[0] == 1) && (sensor[1] == 0) && (sensor[2] == 0) && (putbutton[0] == 1)) //
```

Point A

```
    turnLeft(10);
```

```
else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 0) && (putbutton[1] == 1)) //
```

Point B

```
    forward(10);
```

```
else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (putbutton[1] == 1)) //
```

Point B

```
    turnRight(10);
```

```
else if ((sensor[0] == 1) && (sensor[1] == 0) && (sensor[2] == 0) && (putbutton[1] == 1)) //
```

Point B

```
    turnLeft(10);
```

```
else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (putbutton[0] == 1)) //
```

Point A

```
    brake(40);
```

```
else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (putbutton[1] == 1)) //
```

Point A

```
    forward(10);
```

```
else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (putbutton[1] == 1)) //
```

Point B

```
    brake(40);
```

```
else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (putbutton[0] == 1)) //
```

Point B

```
    forward(10);
```

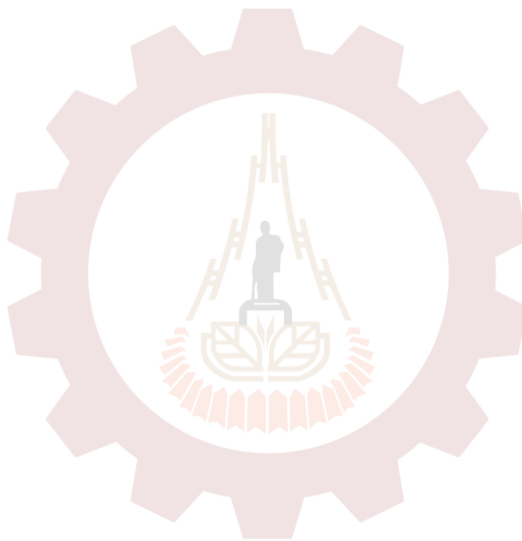
```
else if ((putbutton[0] == 1) || (putbutton[1] == 1))
```

```
    forward(10);
```

```
else
```

```
    brake(40);
```

```
}  
  
//Define high-level H-bridge commands  
void enableMotors()  
{  
    motorAOn();  
    motorBOn();  
}  
void disableMotors()  
{  
    motorAOff();  
    motorBOff();  
}  
void forward(int time)  
{  
    motorAForward();  
    motorBForward();  
    delay(time);  
}  
void backward(int time)  
{  
    motorABackward();  
    motorBBackward();  
    delay(time);  
}  
void turnLeft(int time)  
{  
    motorABackward();  
    motorBForward();
```



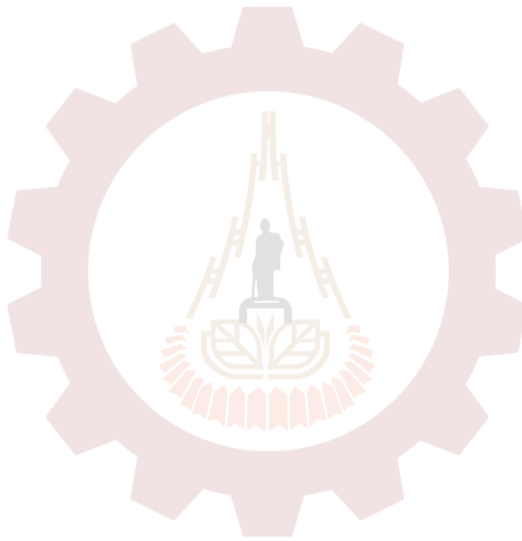
สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

```

    delay(time);
}
void turnRight(int time)
{
    motorAForward();
    motorBBackward();
    delay(time);
}
void coast(int time)
{
    motorACoast();
    motorBCoast();
    delay(time);
}
void brake(int time)
{
    motorABrake();
    motorBBrake();
    delay(time);
}

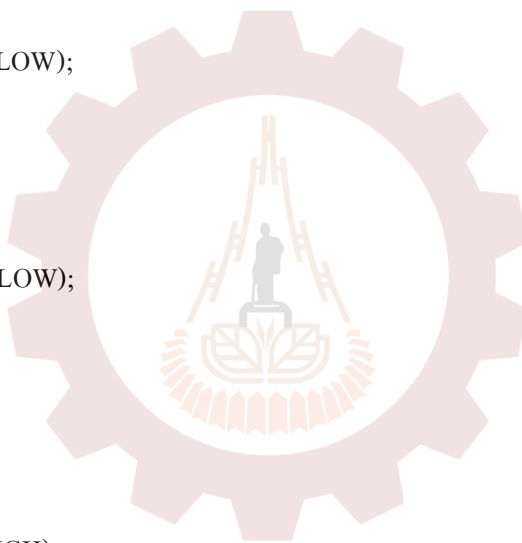
//Define low-level H-bridge commands
//enable motors
void motorAOn()
{
    digitalWrite(enableA, HIGH);
}
void motorBOn()
{

```



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
 สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

```
digitalWrite(enableB, HIGH);
}
//disable motors
void motorAOff()
{
digitalWrite(enableB, LOW);
}
void motorBOff()
{
digitalWrite(enableA, LOW);
}
//motor A controls
void motorAForward()
{
digitalWrite(pinA1, HIGH);
digitalWrite(pinA2, LOW);
}
void motorABackward()
{
digitalWrite(pinA1, LOW);
digitalWrite(pinA2, HIGH);
}
//motor B controls
void motorBForward()
{
digitalWrite(pinB1, HIGH);
digitalWrite(pinB2, LOW);
}
```



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
สำนักวิทยบริการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีสุรนารี

```
void motorBBackward()
{
  digitalWrite(pinB1, LOW);
  digitalWrite(pinB2, HIGH);
}
```

```
//coasting and braking
```

```
void motorACoast()
{
  digitalWrite(pinA1, LOW);
  digitalWrite(pinA2, LOW);
}
```

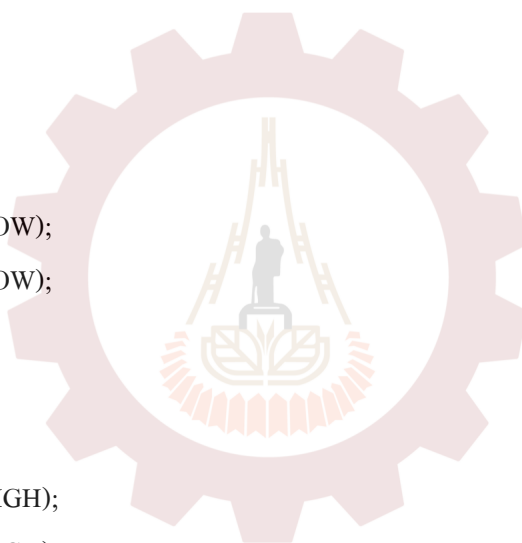
```
void motorABrake()
{
  digitalWrite(pinA1, HIGH);
  digitalWrite(pinA2, HIGH);
}
```

```
void motorBCoast()
```

```
{
  digitalWrite(pinB1, LOW);
  digitalWrite(pinB2, LOW);
}
```

```
void motorBBrake()
```

```
{
  digitalWrite(pinB1, HIGH);
  digitalWrite(pinB2, HIGH);
}
```



สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## ประวัติผู้เขียน

ชื่อ : นายนาวิน ฤกษ์เจริญ

### ประวัติส่วนตัว

เกิดวันที่ 27 กันยายน พ.ศ. 2536 ปัจจุบันอาศัยอยู่บ้านเลขที่ 31/11 หมู่ 14 บ.บุใหญ่ ต.สูงเนิน อ.สูงเนิน จ.นครราชสีมา รหัสไปรษณีย์ 30170

### ประวัติการศึกษา

ปี 2552 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น โรงเรียนโตนดพิทยาคม

ปี 2555 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนโตนดพิทยาคม

ปี 2557 สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพชั้นสูง วิทยาลัยการอาชีพพิมาย

### ประวัติการทำงาน

ปี 2557 เริ่มต้นทำงานอยู่ที่บริษัทซีเกท เทคโนโลยี ประเทศไทย สาขานครราชสีมา ในตำแหน่งช่าง

เทคนิค

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

## ประวัติผู้เขียน

ชื่อ : นางสาวจิตติมา หมั่นแก้ว

### ประวัติส่วนตัว

เกิดวันที่ 11 กรกฎาคม พ.ศ. 2532 ปัจจุบันอาศัยอยู่บ้านเลขที่ 40 หมู่ 15 คุ้มกรเพรส ต.สูงเนิน อ.สูงเนิน จ.นครราชสีมา รหัสไปรษณีย์ 30170

### ประวัติการศึกษา

ปี 2548 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น โรงเรียนบ้านไผ่ศึกษา

ปี 2551 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนบ้านไผ่

ปี 2557 สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพชั้นสูง วิทยาลัยเทคนิคนครราชสีมา

### ประวัติการทำงาน

ปี 2551 เริ่มต้นทำงานอยู่ที่บริษัทซีเกท เทคโนโลยี ประเทศไทย สาขานครราชสีมา ในตำแหน่ง พนักงานฝ่ายผลิต

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี