

ระบบนำทางยานขนส่งอัจฉริยะ  
INTELLIGENT TRANSPORT NAVIGATION SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# INTELLIGENT TRANSPORT NAVIGATION SYSTEM



A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2021

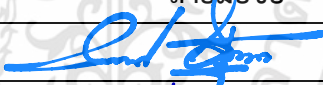
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2564  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ ระบบนำทางยานขนส่งอัจฉริยะ  
INTELLIGENT TRANSPORT NAVIGATION SYSTEM

นักศึกษาผู้จัดทำ นายชานน เจริญธรรม รหัสนักศึกษา 62015031  
นายชินุพัฒน์ ประทุมแดง รหัสนักศึกษา 62015034  
นายธนกร เหล็กม่วง รหัสนักศึกษา 62015046  
นายขมาวีร์ มั่งมี รหัสนักศึกษา 62015112

ปริญญญา วิศวกรรมศาสตรบัณฑิต  
สาขา วิศวกรรมการวัดคุม(โครงการพันธุ์ใหม่)  
ปีการศึกษา 2564

|                                               |                                                                                      |
|-----------------------------------------------|--------------------------------------------------------------------------------------|
| อาจารย์ผู้ควบคุมปริญญานิพนธ์                  | ลายมือชื่อ                                                                           |
| รองศาสตราจารย์ ดร.สุพรรณ กุลพาณิชย์           |  |
| ผู้ช่วยศาสตราจารย์ ดร.นรินทร์ ธรรมารักษ์วัฒนะ |  |

|                    |                                                                                      |           |                       |
|--------------------|--------------------------------------------------------------------------------------|-----------|-----------------------|
| หัวข้อปริญญานิพนธ์ | ระบบนำทางยานขนส่งอัจฉริยะ<br>INTELLIGENT TRANSPORT NAVIGATION SYSTEM                 |           |                       |
| นักศึกษาผู้จัดทำ   | นายชานน                                                                              | เจริญธรรม | รหัสนักศึกษา 62015031 |
|                    | นายชินุพัฒน์                                                                         | ประทุมแดง | รหัสนักศึกษา 62015034 |
|                    | นายธนกร                                                                              | เหล็กม่วง | รหัสนักศึกษา 62015046 |
|                    | นายขมาวีร์                                                                           | มั่งมี    | รหัสนักศึกษา 62015112 |
| อาจารย์ที่ปรึกษา   | รองศาสตราจารย์ ดร.สุพรรณ กุลพาณิชย์<br>ผู้ช่วยศาสตราจารย์ ดร.นรินทร์ ธรรมารักษ์วัฒนะ |           |                       |
| ปีการศึกษา         | 2564                                                                                 |           |                       |

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษา การใช้ระบบนำทางแบบไร้คนขับในการขนส่งวัตถุสิ่งของไปยังเป้าหมายปลายทางที่กำหนดซึ่งแต่เดิมเป็นระบบ Manual ที่ต้องอาศัยผู้ควบคุมคอยควบคุมบังคับทิศทางรถ สำหรับงานวิจัยนี้ขอนำเสนอการควบคุมด้วยระบบนำทางไร้คนขับ โดยอาศัยกล้อง Pixy ประมวลผลภาพและตรวจจับเส้นนำทางด้วยแถบสี จากนั้นควบคุมชุดขับเคลื่อนที่มีมอเตอร์และเอ็นโค้ดเดอร์ติดตั้งเข้าด้วยกันจำนวน 4 ชุดที่แต่ละชุดเป็นอิสระต่อกันนอกจากนี้ยังมี Ultrasonic Sensor ช่วยในการตรวจจับเพื่อหยุดเมื่อพบสิ่งกีดขวางบนเส้นทางเดิน การควบคุมและประมวลผลทั้งหมดนี้จะอาศัยชุดไมโครคอนโทรลเลอร์ Arduino Mega 2560 เป็นหน่วยประมวลผลหลักของโครงการ

**Thesis Title** INTLELLIGENT TRANSPORT NAVIGATION SYSTEM

**Authors** Mr.Chanon Charoentam  
Mr.Chinupat Prathumdang  
Mr.Thanakorn Lekmoung  
Mr.Samawee Mungmee

**Thesis Advisor** Assoc.Prof.Dr.Suphan Gulpanich  
Asst.Prof.Dr.Narin Tammarugwattana

**Year** 2021

### ABSTRACT

This thesis is a study. The use of unmanned navigation systems to transport objects to a specified destination, which was originally a manual system that required the operator to control the direction of the car. For this research, we would like to present the control with unmanned navigation system. It uses the Pixy camera to process images and detect guide lines with colored stripes. Then control the mecanum wheel drive with four motors and encoders mounted together, each independent of one another. There is also an Ultrasonic Sensor for detecting to stop when something is found. obstruction on the walking path All this control and processing relies on the Arduino Mega 2560 microcontroller as the main processor of the project.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดีเพราะคำแนะนำจาก รศ.ดร.สุพรรณ กุลพาณิชย์ ผศ.ดร.นรินทร์ ธรรมารักษ์วัฒน์ และคณาจารย์ทุกท่านของภาควิชาวิศวกรรมการวัดคุมที่ช่วยแนะนำ จัดหา และสอนสั่งในเนื้อหาวิชาการรวมถึงสิ่งสำคัญอันเป็นประโยชน์ต่อการจัดทำปริญญานิพนธ์ในครั้งนี้ อีกทั้งยังเอื้อเพื่อต่ออุปกรณ์ที่จำเป็นต่อโครงงานมากมาย ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อน ๆ นักศึกษาวิศวกรรมการวัดคุมที่คอยช่วยเหลือและต่างก็ให้กำลังใจในระหว่างการทำโครงงานที่อาจพบเรื่องผิดพลาดและไม่สามารถแก้ไขได้ด้วยตนเองด้วยคณะผู้จัดทำเองด้วยคณะผู้จัดทำเอง อีกทั้งยังไม่ลืมอุปกรณ์ที่ช่วยในการทำโครงการผ่านพ้นไปด้วยดี

และที่ลืมไม่ได้เสีย ทางคณะผู้จัดทำขอกราบขอบพระคุณครอบครัวอันประกอบไปด้วย พ่อ แม่ พี่น้อง รวมถึงปู่ย่า ตายาย ทุกคนที่ช่วยสนับสนุนและเป็นแรงบันดาลใจในการทำปริญญานิพนธ์จนประสบผลสำเร็จเช่นนี้ได้

คุณค่าและคุณประโยชน์จากการทำปริญญานิพนธ์ในครั้งนี้ ผู้จัดทำขอบแต่ผู้มีพระคุณทุกท่านที่มีส่วนช่วยในโครงงานชิ้นนี้ประสบผลสำเร็จ

คณะผู้จัดทำ

# สารบัญ

|                                                                    | หน้า |
|--------------------------------------------------------------------|------|
| บทคัดย่อภาษาไทย.....                                               | I    |
| บทคัดย่อภาษาอังกฤษ.....                                            | II   |
| กิตติกรรมประกาศ .....                                              | III  |
| สารบัญ .....                                                       | IV   |
| สารบัญตาราง .....                                                  | VIII |
| สารบัญภาพ .....                                                    | IX   |
| <br>                                                               |      |
| บทที่ 1 บทนำ .....                                                 | 1    |
| 1.1 ความสำคัญของปริญญานิพนธ์ .....                                 | 1    |
| 1.2 วัตถุประสงค์ของปริญญานิพนธ์ .....                              | 1    |
| 1.3 ขอบเขตของปริญญานิพนธ์ .....                                    | 1    |
| 1.4 ขั้นตอนการศึกษา.....                                           | 2    |
| <br>                                                               |      |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้อง .....                                   | 3    |
| 2.1 เอ็นโค้ดเดอร์ .....                                            | 3    |
| 2.1.1 หลักการทำงาน .....                                           | 4    |
| 2.1.1.1 เพลา.....                                                  | 4    |
| 2.1.1.2 แผ่นดิสก์.....                                             | 4    |
| 2.1.1.3 แหล่งแสง .....                                             | 4    |
| 2.1.1.4 ตัวรับแสง .....                                            | 5    |
| 2.1.2 การนำไปใช้งาน .....                                          | 5    |
| 2.1.3 การต่อสายใช้งาน .....                                        | 5    |
| 2.2 Ultrasonic Sensor .....                                        | 7    |
| 2.2.1 การลดสัญญาณรบกวนและสภาวะการทำงานของเซนเซอร์อัลตราโซนิก ..... | 7    |
| 2.2.2 การต่อใช้งาน Distance measuring Sensor.....                  | 8    |
| 2.3 Mecanum wheel.....                                             | 8    |
| 2.4 แบตเตอรี่ .....                                                | 9    |

## สารบัญ(ต่อ)

|                                                                                | หน้า |
|--------------------------------------------------------------------------------|------|
| 2.4.1 หลักการทำงาน .....                                                       | 9    |
| 2.4.2 ชนิดของแบตเตอรี่.....                                                    | 10   |
| 2.4.2.1 แบตเตอรี่.....                                                         | 10   |
| 2.4.2.2 เจล.....                                                               | 10   |
| 2.4.2.3 ตะกั่ว.....                                                            | 10   |
| 2.4.2.4 ลิเทียม .....                                                          | 11   |
| 2.5 Arduino .....                                                              | 11   |
| 2.5.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม .....                           | 12   |
| 2.5.2 รูปแบบการเขียนโปรแกรมบน Arduino .....                                    | 12   |
| 2.5.3 Layout & Pin out Arduino Bord.....                                       | 15   |
| 2.6 PIXY2 .....                                                                | 16   |
| 2.6.1 ส่วนประกอบของกล่อง PIXY2 .....                                           | 16   |
| 2.7 Pixy Mon.....                                                              | 16   |
| 2.8 L298N motor drive .....                                                    | 17   |
| 2.8.1 หลักการทำงาน .....                                                       | 17   |
| 2.8.2 พิจารณาโครงสร้างโค้ดของ Arduino .....                                    | 18   |
| 2.9 Wireless Joystick.....                                                     | 19   |
| 2.9.1 คุณสมบัติทางเทคนิค.....                                                  | 19   |
| 2.9.2 สัญญาณที่ใช้ในการเชื่อมต่อ PS2 กับไมโครคอนโทรลเลอร์.....                 | 20   |
| 2.9.3 ข้อมูลที่เกิดขึ้นในการติดต่อ .....                                       | 21   |
| 2.10 Lux Meter.....                                                            | 21   |
| 2.10.1 หลักการทำงานของเครื่องมือวัดแสง.....                                    | 22   |
| 2.10.2 ประเภทของเครื่องมือวัดแสง.....                                          | 22   |
| 2.10.3 ระดับความเข้มของแสงสว่างทั่วไปโดยประมาณ.....                            | 22   |
| 2.11 ตัวแปรและฟังก์ชันที่ใช้ในการควบคุมในการเขียนโค้ดรับค่าจากกล่อง PIXY ..... | 23   |
| 2.11.1 ฟังก์ชัน.....                                                           | 23   |
| 2.12.2 ตัวแปร .....                                                            | 24   |

## สารบัญ(ต่อ)

|                                                                                | หน้า      |
|--------------------------------------------------------------------------------|-----------|
| 2.12 Pulse Width Modulation (PWM) .....                                        | 24        |
| 2.11.1 Algorithm การสร้าง PWM โดยใช้ Timer Interrupt.....                      | 25        |
| 2.12.2 ข้อควรระวัง .....                                                       | 26        |
| <b>บทที่ 3 การออกแบบและการทำงาน .....</b>                                      | <b>27</b> |
| 3.1 การออกแบบระบบ .....                                                        | 27        |
| 3.1.1 ระบบทั้งหมดประกอบไปด้วย.....                                             | 27        |
| 3.1.2 ขั้นตอนการทำงานของระบบ .....                                             | 28        |
| 3.2 วงจรของระบบ.....                                                           | 29        |
| 3.3 โครงสร้างของรถ .....                                                       | 29        |
| 3.4 ออกแบบการเคลื่อนที่ของล้อแมกคาน้ำ .....                                    | 31        |
| 3.5 ออกแบบการทำงานจอยสติ๊กกับการเคลื่อนที่ของล้อแมกคาน้ำ .....                 | 32        |
| 3.6 กำหนดช่วงแกน X ของกล้อง PIXY.....                                          | 33        |
| 3.7 กำหนดตัวแปรควบคุมล้อแมกคาน้ำผ่านโปรแกรม PIXY MON .....                     | 34        |
| 3.7.1 ขั้นตอนการตั้งค่าการตรวจจับสี.....                                       | 34        |
| 3.7.2 กำหนดค่าตัวแปรควบคุมให้รถเคลื่อนที่เข้าหาแกน X.....                      | 36        |
| 3.8 โค้ดควบคุมล้อแมกคาน้ำ .....                                                | 38        |
| 3.8 โค้ดกล้อง PIXY2 รับค่าจากแกน X ส่งไปควบคุมยังบอร์ดไมโครคอนโทรลเลอร์.....   | 44        |
| <b>บทที่ 4 ผลการทดลอง.....</b>                                                 | <b>46</b> |
| 4.1 การทดลองระยะการตรวจจับสีของกล้องกับแสงสว่างแนวตั้งฉากกับพื้น .....         | 46        |
| 4.1.1 การทดลองวัดปริมาณแสงกับวัตถุขนาด 26x19 เซนติเมตร แนวตั้งฉากกับพื้น ..... | 46        |
| 4.1.2 การทดลองวัดปริมาณแสงกับวัตถุขนาด 13x19 เซนติเมตร แนวตั้งฉากกับพื้น ..... | 47        |
| 4.1.3 การทดลองวัดปริมาณแสงกับวัตถุขนาด 13x10 เซนติเมตร แนวตั้งฉากกับพื้น ..... | 48        |
| 4.2 การทดลองระยะการตรวจจับสีของกล้องกับแสงสว่างแนวราบกับพื้น .....             | 49        |
| 4.2.1 การทดลองวัดปริมาณแสงกับวัตถุขนาด 26x19 เซนติเมตร แนวราบกับพื้น.....      | 49        |
| 4.2.2 การทดลองวัดปริมาณแสงกับวัตถุขนาด 13x19 เซนติเมตร แนวราบกับพื้น.....      | 50        |

## สารบัญ(ต่อ)

|                                                                                | หน้า      |
|--------------------------------------------------------------------------------|-----------|
| 4.2.3 การทดลองวัดปริมาณแสงกับวัตถุขนาด 13x10 เซนติเมตร แนวราบกับพื้น.....      | 51        |
| 4.3 การทดสอบแรงดันไฟฟ้าที่ทำการจ่ายให้ล้อแมกคานัมโดยการควบคุม PWM มอเตอร์..... | 52        |
| 4.4 การทดลองความเร็วในการเคลื่อนที่ตามสีแนวแกน X ของกล้อง PIXY .....           | 53        |
| 4.5 การทดลองความเร็วในการเคลื่อนที่ตามสีแนวแกน X ของกล้อง PIXY .....           | 54        |
| 4.5.1 การทดลองวิ่งแนวเส้นตรง .....                                             | 56        |
| 4.5.2 การทดลองวิ่งแนวโค้งไปทางซ้าย.....                                        | 57        |
| 4.5.3 การทดลองวิ่งแนวโค้งไปทางขวา.....                                         | 58        |
| 4.5.4 การทดลองวิ่งแนวเส้นตรงแนวโค้งไปทางซ้ายและแนวโค้งไปทางขวา.....            | 59        |
| <b>บทที่ 5 สรุปผลการทดลอง.....</b>                                             | <b>60</b> |
| 5.1 สรุปผลการทดลอง .....                                                       | 60        |
| 5.2 ปัญหาที่พบขณะทำการทดลอง .....                                              | 60        |
| 5.3 วิธีการแก้ไขปัญหา .....                                                    | 60        |
| 5.4 ข้อจำกัดในการใช้งาน.....                                                   | 61        |
| 5.5 ข้อเสนอแนะเพิ่มเติม .....                                                  | 61        |
| <b>บรรณานุกรม.....</b>                                                         | <b>62</b> |
| <b>ภาคผนวก.....</b>                                                            | <b>63</b> |

## สารบัญตาราง

| ตารางที่                                                                   | หน้า |
|----------------------------------------------------------------------------|------|
| 3.1 การเคลื่อนที่และทิศทางของล้อแม็กคานัม.....                             | 31   |
| 4.1.1 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 26x19 เซนติเมตร ..... | 46   |
| 4.1.2 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x19 เซนติเมตร ..... | 47   |
| 4.1.3 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x10 เซนติเมตร ..... | 48   |
| 4.2.1 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 26x19 เซนติเมตร ..... | 49   |
| 4.2.2 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x19 เซนติเมตร ..... | 50   |
| 4.2.3 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x10 เซนติเมตร ..... | 51   |
| 4.3.1 การทดสอบวัดแรงดันไฟฟ้าที่ความเร็วในการควบคุมมอเตอร์ต่างกัน.....      | 52   |
| 4.4.1 การทดสอบความเร็วในการเคลื่อนที่ของรถ .....                           | 53   |
| 4.5.1 การทดลองวิ่งแนวเส้นตรง .....                                         | 56   |
| 4.5.2 การทดลองวิ่งแนวโค้งไปทางซ้าย .....                                   | 57   |
| 4.5.3 การทดลองวิ่งแนวโค้งไปทางขวา .....                                    | 58   |
| 4.5.4 การทดลองวิ่งทางตรงแนวทางโค้งไปทางซ้ายและแนวทางโค้งไปทางขวา .....     | 59   |

## สารบัญรูป

| รูปที่                                                                                     | หน้า |
|--------------------------------------------------------------------------------------------|------|
| 2.1 ลักษณะมอเตอร์เอ็นโค้ดเดอร์ .....                                                       | 3    |
| 2.2 ส่วนประกอบเบื้องต้นของเอ็นโค้ดเดอร์.....                                               | 4    |
| 2.3 ลักษณะสายมอเตอร์ Encoder .....                                                         | 5    |
| 2.4 ลักษณะเอาต์พุตของมอเตอร์เอ็นโค้ดเดอร์ .....                                            | 6    |
| 2.5 บอกลักษณะของสายไฟ.....                                                                 | 6    |
| 2.6 แสดงการสะท้อนกลับของ Ultrasonic sensor.....                                            | 7    |
| 2.7 ลักษณะของผลที่เกิดจากสัญญาณรบกวน .....                                                 | 8    |
| 2.8 ลักษณะของขา Ultrasonic sensor.....                                                     | 8    |
| 2.9 ลักษณะของล้อ Mecanum wheel .....                                                       | 9    |
| 2.10 บอร์ดต่อเข้ากับ LED .....                                                             | 11   |
| 2.11 บอร์ดต่อเข้ากับ XBee Shield .....                                                     | 12   |
| 2.12 การเชื่อมต่อ Arduino เข้ากับคอมพิวเตอร์.....                                          | 12   |
| 2.13 เลือกบอร์ด Arduino ที่ต้องการ upload .....                                            | 13   |
| 2.14 เลือกหมายเลข Comport ของบอร์ด .....                                                   | 13   |
| 2.15 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง และ Compile โปรแกรม .....                       | 14   |
| 2.16 Upload โค้ดโปรแกรม .....                                                              | 14   |
| 2.17 Model : Arduino Uno R3.....                                                           | 15   |
| 2.18 แสดง Port การเชื่อมต่อของกล้อง PiXY 2.....                                            | 16   |
| 2.19 แสดงโปรแกรม Pixy Mon .....                                                            | 17   |
| 2.20 แสดงรายละเอียดของบอร์ด L298N Motor Drive .....                                        | 18   |
| 2.21 ไดอะแกรมเวลาของการติดต่อระหว่างไมโครคอนโทรลเลอร์และตัวควบคุมหรือจอยสติ๊ก PS2.....     | 20   |
| 2.22 แสดงตำแหน่งปุ่มกดของจอยสติ๊ก PS2 ไร้สายเพื่อใช้อ้างอิงกับข้อมูลสถานะที่ได้กลับมา..... | 20   |
| จากจอยสติ๊ก PS2                                                                            |      |
| 2.23 ข้อมูลที่เกิดขึ้นเมื่อตัวควบคุมหรือจอยสติ๊ก PS2 ทำงานในโหมดนอนาล็อก .....             | 21   |
| 2.24 เครื่องมือวัดแสง.....                                                                 | 23   |
| 2.25 Pulse width .....                                                                     | 25   |
| 2.26 การสร้าง PWM โดยใช้ Timer Interrupt.....                                              | 25   |

## สารบัญรูป(ต่อ)

| รูปที่                                                                                     | หน้า |
|--------------------------------------------------------------------------------------------|------|
| 3.1 วงจรการทำงานของระบบ .....                                                              | 29   |
| 3.2 โครงสร้างด้านหน้าของรถ .....                                                           | 29   |
| 3.3 โครงสร้างด้านข้างของรถ .....                                                           | 30   |
| 3.4 โครงสร้างด้านหลังของรถ .....                                                           | 30   |
| 3.5 บอกตำแหน่งของการควบคุมล้อแม็กคานัม .....                                               | 32   |
| 3.6 ช่วงแกน X ที่รถจะทำการเคลื่อนที่ไปข้างหน้า .....                                       | 33   |
| 3.7 ช่วงแกน X ที่รถจะทำการเคลื่อนที่ทางทิศตะวันออกเฉียงใต้และทางทิศตะวันตกเฉียงเหนือ ..... | 33   |
| 3.8 เซ็ตค่าของกล่องเป็นค่าเริ่มต้น .....                                                   | 34   |
| 3.9 เลือก Set signature 1 .....                                                            | 34   |
| 3.10 เลือกวัตถุที่จะทำการตรวจจับสี .....                                                   | 35   |
| 3.11 การตรวจจับวัตถุ .....                                                                 | 35   |
| 3.12 แสดงค่าการกำหนดตัวแปรให้รถเคลื่อนที่ไปข้างหน้า .....                                  | 36   |
| 3.13 แสดงค่าการกำหนดตัวแปรให้รถเคลื่อนที่ไปยังทิศตะวันออกเฉียงใต้ .....                    | 36   |
| 3.14 แสดงค่าการกำหนดตัวแปรให้รถเคลื่อนที่ไปยังทิศตะวันตกเฉียงเหนือ .....                   | 37   |
| 3.15 แสดงค่าการกำหนดตัวแปรให้รถหยุดการเคลื่อนที่ .....                                     | 37   |
| 4.1 วัดปริมาณแสงของแสงกับวัตถุแนวตั้งฉากกับพื้นขนาด 26x19 เซนติเมตร .....                  | 46   |
| 4.2 วัดปริมาณแสงของแสงกับวัตถุแนวตั้งฉากกับพื้นขนาด 13x19 เซนติเมตร .....                  | 47   |
| 4.3 วัดปริมาณแสงของแสงกับวัตถุแนวตั้งฉากกับพื้นขนาด 13x10 เซนติเมตร .....                  | 48   |
| 4.4 วัดปริมาณแสงของแสงกับวัตถุแนวราบกับพื้นขนาด 26x19 เซนติเมตร .....                      | 49   |
| 4.5 วัดปริมาณแสงของแสงกับวัตถุแนวราบกับพื้นขนาด 13x19 เซนติเมตร .....                      | 50   |
| 4.6 วัดปริมาณแสงของแสงกับวัตถุแนวราบกับพื้นขนาด 13x10 เซนติเมตร .....                      | 51   |
| 4.7 วัดแรงดันไฟฟ้าทั้ง 3 ความเร็วของมอเตอร์ .....                                          | 52   |
| 4.8 การทดสอบความเร็วในการเคลื่อนที่ของรถ .....                                             | 53   |
| 4.9 เส้นทางตรงที่ใช้ในการทดลอง .....                                                       | 54   |
| 4.10 เส้นทางโค้งไปทางซ้ายที่ใช้ในการทดลอง .....                                            | 54   |
| 4.11 เส้นทางโค้งไปทางขวาที่ใช้ในการทดลอง .....                                             | 55   |
| 4.12 ความเร็วในการควบคุมมอเตอร์ของรถระดับที่ 1 .....                                       | 55   |

## สารบัญรูป(ต่อ)

| รูปที่                                              | หน้า |
|-----------------------------------------------------|------|
| 4.13 ความเร็วในการควบคุมมอเตอร์ของรระดับที่ 2 ..... | 55   |
| 4.14 ความเร็วในการควบคุมมอเตอร์ของรระดับที่ 3 ..... | 55   |



## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันเทคโนโลยีการใช้นานพาหนะไฟฟ้า (Electric Vehicle) ในภาคการขนส่งเริ่มได้ให้ความสนใจมากขึ้นโดยมีนัยสำคัญจากประเทศต่าง ๆ ทั่วโลก เนื่องจากจากเทคโนโลยีที่กล่าวอาศัยพลังงานไฟฟ้าที่เก็บสะสมมาจากแบตเตอรี่มาเป็นแหล่งพลังงานหลักในการขับเคลื่อนยานพาหนะที่ใช้ในการขนส่งดังกล่าว โดยเมื่อเทียบกับยานพาหนะที่ใช้เครื่องยนต์สันดาปภายใน (Internal Combustion Engine : ICE) ซึ่งใช้น้ำมันหรือก๊าซธรรมชาติในการขับเคลื่อนทำให้ก๊าซหรือพลังงานที่เกิดจากการเผาไหม้ส่งผลเสียต่อสภาพแวดล้อมทางอากาศทำให้อาจเกิดปัญหามลภาวะโลกร้อนได้ จะพบเห็นได้ว่ายานพาหนะไฟฟ้านั้นจะไม่ปล่อยมลพิษที่มีไอเสียที่ทำให้สภาพแวดล้อมเสียหาย เนื่องจากด้วยเทคโนโลยียานยนต์ไฟฟ้าได้เติบโตขึ้นอย่างมาก จนทำให้มีผู้ต้องการที่จะจัดหาจัดซื้อรถยนต์ไฟฟ้าที่ทันสมัยสมเหตุสมผลเพื่อทำการขนส่ง โดยระบบหลักในอนาคตที่จะเติบโตขึ้นของรถยนต์ไฟฟ้านั้น จะเป็นระบบนำทางซึ่งไร้คนขับซึ่งจะใช้มาทำงานแทนการขนส่งที่ต้องใช้มนุษย์ในการบังคับหรือขับเคลื่อนยานพาหนะที่มีสิ่งของที่ต้องการส่งให้ถึงมือผู้รับ โดยโรงงานอุตสาหกรรมนั้นต้องการระบบนำทางในการขนส่งที่ไร้คนขับเพื่อที่จะช่วยในการลดต้นทุนในการจ้างงานเพื่อการขนส่ง ร่วมกับการใช้ยานยนต์ไฟฟ้า

ดังนั้นเพื่อแก้ไขปัญหาที่ได้กล่าวมาข้างต้นจึงได้มีการจัดทำโครงการนี้ขึ้นมาโดยการนำระบบของกล้องมาใช้ในการนำทางของการขนส่งสิ่งของภายในโรงงานอุตสาหกรรม โดยระบบนำทางที่ถูกทำขึ้นนั้นสามารถใช้ในการควบคุมซึ่งไร้คนขับได้จะใช้กล้องในการจับพิกัดตำแหน่งที่กำหนดไว้ในการขนส่งให้ไปถึงจุดหมายปลายทางที่กำหนดในโรงงานอุตสาหกรรม

#### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อออกแบบระบบนำทางรถขนส่งสิ่งของของรถไฟฟ้า โดยการใช้กล้องควบคุมมอเตอร์ไฟฟ้าในการขับเคลื่อน
2. เพื่อเป็นระบบขนส่งซึ่งไร้คนขับ ในการขนส่งให้ถึงจุดหมายปลายทางที่สมบูรณ์
3. ระบบนำทางสามารถหลบหลีกสิ่งกีดขวางได้ ขณะที่มือของหรือคนยืนขวางทาง

### 1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถหยุดเมื่อพบสิ่งขัดขวางบนเส้นทางเดินได้
2. สามารถใช้กล้องนำทางไปถึงจุดหมายได้
3. เขียนโปรแกรมให้ บอร์ดอิเล็กทรอนิกส์ เพื่อสั่งการทำงานในการควบคุมของผู้ใช้งาน

### 1.4 ขั้นตอนการศึกษา

1. ศึกษาการเขียนโปรแกรมของบอร์ด Arduino
2. ศึกษาเงื่อนไขการทำงานของโค้ดโปรแกรม
3. ศึกษาข้อมูลตัวกล้องที่จะนำมาใช้งานในการควบคุม
4. ทำการเชื่อมต่อระหว่างอุปกรณ์และนำไปทดสอบการทำงานของระบบนำทาง



## บทที่ 2

# ทฤษฎีและหลักการที่เกี่ยวข้อง

### 2.1 เอ็นโค้ดเดอร์ (Encoder)

เอ็นโค้ดเดอร์ Encoder มีอีกชื่อหนึ่งที่นิยมเรียกกันว่า เอ็นโค้ดเดอร์แบบแกนหมุน คือ เซ็นเซอร์ชนิดหนึ่งที่ทำหน้าที่ในการเข้ารหัส จากระยะทางจากการหมุนรอบตัวเอง และแปลงออกมาเป็นรหัสในรูปแบบของสัญญาณไฟฟ้าโดยเราสามารถนำเอารหัสเหล่านี้มาแปลงกลับ เพื่อหาค่าต่าง ๆ ที่เราต้องการได้ ไม่ว่าจะเป็นระยะทางการหมุน องศาการเคลื่อนที่ หรือ ความเร็วรอบก็ได้ แล้วนำมาแสดงผลให้เราได้ทราบค่าผ่านหน้าจอแสดงผล เช่น ถ้าต้องการวัดระยะทาง เราจะต้องต่อเข้ากับตัวนับจำนวน เพื่อแสดงผลเป็นระยะทาง หรือ ถ้าต้องการวัดความเร็วรอบ เราจะต้องต่อเข้ากับตัววัดพัลส์ โดยการประยุกต์ใช้เอ็นโค้ดเดอร์นั้น สามารถใช้ทำงานได้อย่างหลากหลาย เช่น กระบวนการประกอบชิ้นส่วน อิเล็กทรอนิกส์ อุตสาหกรรมเคมีคอนดักเตอร์ เครื่องมือวัดต่าง ๆ เช่น ในการวัดความยาว หรือ อุตสาหกรรมเครื่องมือแพทย์ เป็นต้น

ส่วนการแสดงผลเป็นความเร็วรอบของ RPM, RPS โดยอาศัยสัญญาณที่ผ่านการเข้ารหัสแล้วออกมาเป็นสัญญาณทางไฟฟ้านั้น สามารถแบ่งรูปแบบของการเข้ารหัสได้อีกหลากหลายรูปแบบ เช่น สัญญาณดิจิทัล ศูนย์กับหนึ่ง ธรรมดา หรือ เป็นแบบ Binary Code, BCD Code, Gray Code

ซึ่งในปัจจุบันก็มีสัญญาณที่เป็นการสื่อสารแบบอนุกรมอื่นๆ อีกมากมายที่ถูกนำมาใช้กับ เอ็นโค้ดเดอร์ เช่น SSI, CAN, PROFIBUS, ETHERCAT โดยการสื่อสารแบบต่างๆเหล่านี้ ถูกออกแบบมาเพื่อเพิ่มระยะในการสื่อสารให้ไกลมากขึ้น อีกทั้งยังเป็นการลดปัญหาเรื่องการส่งสัญญาณ ที่ผิดพลาด และสามารถตอบสนองต่อยุคการผลิตแบบ Industry 4.0 หรือ Smart Factory ได้เป็นอย่างดี



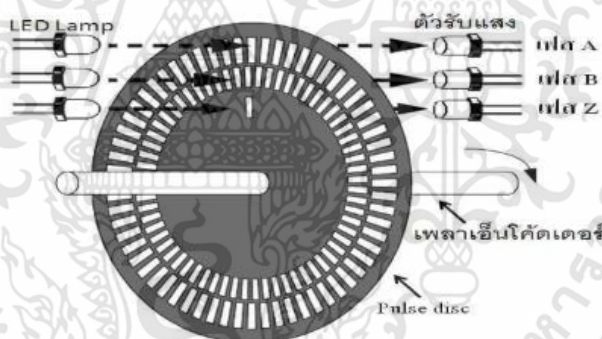
รูปที่ 2.1 ลักษณะมอเตอร์ Encoder

เราสามารถแบ่งประเภทของเอ็นโค้ดเดอร์ได้หลายแบบ โดยขึ้นอยู่กับเราจะยึดเกณฑ์ใดเพื่อใช้ในการแบ่ง เช่น เราอาจแบ่งได้จากโครงสร้างการใช้งาน หรือแบ่งตามโครงสร้างของเอ็นโค้ดเดอร์ก็ได้ และนอกจากนี้เรายังสามารถใช้ประเภทเอาต์พุตของเอ็นโค้ดเดอร์มาใช้ในการแบ่งได้อีกเช่นกัน โดยความสำคัญของทั้งหมดที่กล่าวมานั้น จะเป็นส่วนจำเป็นต่อการเลือกใช้เอ็นโค้ดเดอร์ให้เหมาะสมกับการใช้งานและการติดตั้งเอ็นโค้ดเดอร์ตามที่ต้องการได้ แต่ถ้าหากพบเจอกับข้อจำกัดต่าง ๆ ที่ก่อให้เกิดความลำบากในการทำงานหรือการติดตั้ง สิ่งหนึ่งที่จะสามารถช่วยลดหรือจัดข้อจำกัดต่าง ๆ ไปได้นั้นก็คือ อุปกรณ์เสริมของเอ็นโค้ดเดอร์ ซึ่งอุปกรณ์เหล่านี้จะถูกออกแบบมาเพื่อช่วยงานที่มีข้อจำกัดต่าง ๆ โดยเฉพาะ

ซึ่งจะเห็นได้แล้วว่าการใช้เอ็นโค้ดเดอร์ที่ดีนั้น ไม่ใช่เพียงแค่เราารู้วิธีการใช้ แต่เราจำเป็นต้องรู้จักกับเอ็นโค้ดเดอร์ให้ดีเสียก่อน จึงจะทำให้เราสามารถดึงประสิทธิภาพการทำงานเหล่านั้นออกมาใช้ได้อย่างเต็มที่

### 2.1.1 หลักการทำงาน

ส่วนประกอบเบื้องต้นของเอ็นโค้ดเดอร์



รูปที่ 2.2 ส่วนประกอบเบื้องต้นของเอ็นโค้ดเดอร์

#### 2.1.1.1. เพลลา (Shaft)

ใช้สำหรับต่อเข้ากับวัตถุที่หมุนเช่น มอเตอร์

#### 2.1.1.2. แผ่นดิสก์ (Code หรือ PulseDisc)

จะเป็นแผ่นที่มีแตร็กหรือร่องเล็ก ๆ มีทั้งส่วนที่โปร่งแสงและทึบแสง เพื่อให้แสงอินฟราเรดลอดผ่านได้

#### 2.1.1.3. แหล่งแสง (Light Source)

เป็นไฟ LED คุณภาพสูง

#### 2.1.1.4. ตัวรับแสง (Photodetector หรือ Photodiode)

ใช้รับแสงจาก LED เพื่อแปลงไปเป็นรหัสข้อมูล

#### 2.1.2 การนำไปใช้งาน

จากส่วนประกอบที่ได้กล่าวไป จะเห็นว่าตัวส่งแสงและรับแสงจะมี 3 ชุด เรียกว่าเฟส A, B และ Z เอาต์พุตของ Encoder ที่ออกมาจึงมี 3 ชุดด้วยเช่นกัน เพื่อนำไปใช้งานหลากหลาย โดยจะมีข้อแตกต่างคือ เฟส A และ B จะเลื่อนกันอยู่ 90 องศา (ทำให้นำไปใช้งานเพื่อดูทิศทางการหมุนได้ ว่าหมุนซ้ายหรือขวา คือเจอ A ก่อน B หรือ B ก่อน A) ส่วนเฟส Z เมื่อครบ 1 รอบจึงจะปล่อยแสงออกมา 1 ครั้ง (นำไปใช้อ้างอิงตำแหน่งเมื่อครบรอบได้)

นอกจากนี้ จะยังมีสัญญาณพิเศษเสริมให้อีก 1 ชุด ของทั้งเฟส A B และ Z (สัญญาณคือ มีขีดอยู่บนหัวตัวอักษร) คือจะปล่อยสัญญาณที่กลับหัว (Inverted) ของ A, B และ Z ออกมาให้ ก็สามารถนำไปประยุกต์ใช้งานได้อีกมากมายเช่นกัน

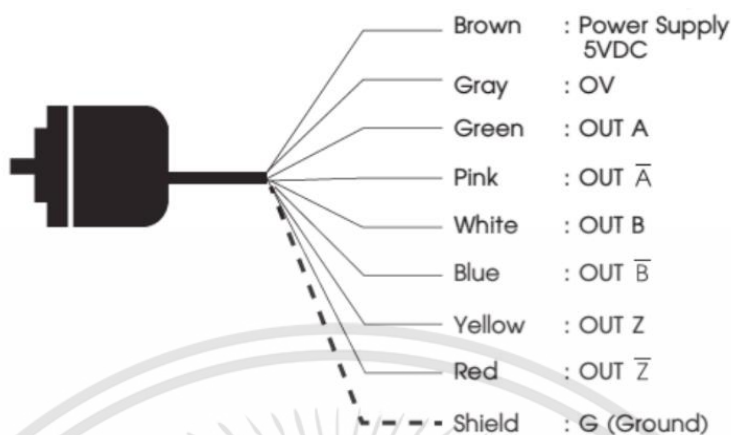
ส่วนใหญ่ Encoder มักถูกนำไปใช้ในงานต่อกับคัปปลิง หรือต่อเข้ากับลูกล้อ ไม่ว่าจะเป็นมอเตอร์หรือสายพานลำเลียง เพื่อวัดรอบ, วัดระยะทาง, วัดทิศทางการหมุน ไปจนถึงการตรวจจับตำแหน่ง จึงเป็นที่นิยมใช้อย่างแพร่หลายในงานหลากหลายรูปแบบ

#### 2.1.3 การต่อสายใช้งาน

ขอยกตัวอย่างการต่อสายใช้งานของ Encoder ยี่ห้อ Kubler รุ่น 3700 Series จากประเทศเยอรมนี ให้เห็นภาพว่าเป็นอย่างไร เบื้องต้นต้องเข้าใจก่อนว่า ด้วย Encoder 1 ตัว มีเอาต์พุตให้เลือกใช้หลากหลาย จำนวนสายจึงจะเยอะมาก แต่หากเราเข้าใจหลักการทำงานของมันก็ไม่ได้ยาก



รูปที่ 2.3 ลักษณะสายของมอเตอร์ Encoder



รูปที่ 2.4 ลักษณะเอาต์พุตของมอเตอร์ Encoder

| Output circuit | Type of connection | Cable (isolate unused wires individually before initial start-up) |     |    |    |           |    |           |    |           |
|----------------|--------------------|-------------------------------------------------------------------|-----|----|----|-----------|----|-----------|----|-----------|
| 1, 3, 4        | 1 ... 8            | Signal:                                                           | 0 V | +V | A  | $\bar{A}$ | B  | $\bar{B}$ | 0  | $\bar{0}$ |
|                |                    | Cable colour:                                                     | WH  | BN | GN | YE        | GY | PK        | BU | RD        |

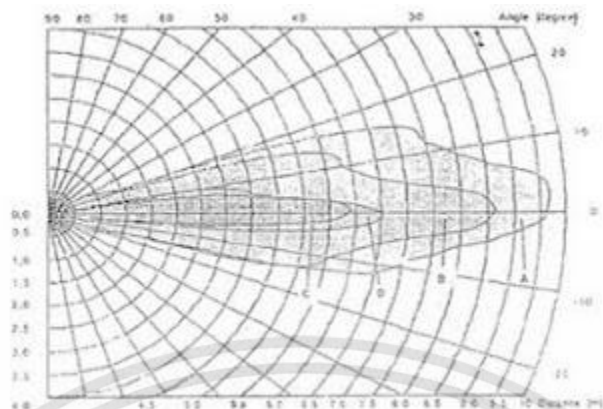
+V: Encoder power supply +V DC  
 0 V: Encoder power supply ground GND (0 V)  
 A,  $\bar{A}$ : Incremental output channel A  
 B,  $\bar{B}$ : Incremental output channel B  
 0,  $\bar{0}$ : Reference signal

รูปที่ 2.5 บอกลักษณะของสายไฟ

## 2.2 Ultrasonic sensor

Ultrasonic sensor เป็นเซ็นเซอร์ที่ทำงานโดยอาศัยคลื่นเสียงที่มีความถี่สูงกว่า 20 กิโลเฮิร์ต (kHz) ซึ่งเป็นคลื่นในย่านที่มนุษย์ไม่สามารถได้ยินเสียง เซ็นเซอร์ชนิดอัลตราโซนิกทำงานโดยอาศัยการกระจาย หรือการเคลื่อนที่ของคลื่นเสียงไปกระทบกับพื้นผิวของตัวกลาง ซึ่งอาจเป็นของแข็งหรือของเหลว บางส่วนของคลื่นเสียงจะแทรกผ่านเข้าไปในตัวกลางนั้น และส่วนใหญ่ของคลื่นความถี่สูงนี้จะสะท้อนกลับเรียกว่า "Echo" โดยช่วงเวลาของการสะท้อนกลับของคลื่นเสียงเป็นสัดส่วนโดยตรงกับระยะห่างระหว่างวัตถุกับเซ็นเซอร์





รูปที่ 2.7 ลักษณะของผลที่เกิดจากสัญญาณรบกวน

## 2.2.2 การต่อขาใช้งาน Distance measuring Sensor



รูปที่ 2.8 ลักษณะของขา Ultrasonic sensor

## 2.3 Mecanum wheel

Mecanum wheel คิดค้นโดย Bengt Erland Ilon วิศวกรจากบริษัทสวีเดน Mecanum AB หลักการทำงานของ Mecanum wheel ใช้ชุด roller ที่ติดอยู่รอบ Mecanum wheel สามารถทำให้ Mecanum wheel มีความพิเศษที่ขาดไม่ถึง เมื่อผสานรวมกับการหมุนของล้อทั้ง 4 ข้าง ทำให้เกิดการเคลื่อนที่ได้ออกทิศทางอย่างมหัสจรรย์ Mecanum wheel เป็นการออกแบบให้เป็นล้อทรงกลมที่ติดตั้ง roller ทำมุม  $45^\circ$  เรียงรายกันอยู่บน Mecanum wheel เพื่อรองรับการหมุนด้านข้าง ยานยนต์ที่ติดตั้ง Mecanum wheel จะต้องมีการสั่งการด้วยไฟฟ้าแบบแยก 4 ล้อ เพราะการทำงานของ Mecanum wheel จะใช้ความสัมพันธ์ในการหมุนของล้อทั้ง 4 Mecanum wheel การทำงานที่สัมพันธ์ทั้ง 4 ล้อ หมุนไปในด้านเดียวกันจะทำให้ ยานยนต์ ที่ติดตั้ง Mecanum wheel เกิดการเคลื่อนที่เดินหน้า และถอยหลัง เหมือนล้อ และยางปกติที่ใช้กันหาก Mecanum wheel ทำงานสัมพันธ์กันแบบทะแยงมุม คือ การหมุนที่เหมือนกัน 2 ชุด ล้อซ้ายบนหมุนไปทางเดียวกับขวาล่าง และขวามบนหมุนไปทาง

เดียวกับซ้ายล่าง จะทำให้ยานยนต์ที่ใช้ Mecanum wheel เกิดการเคลื่อนที่ด้านข้างโดยไม่ต้องปรับเปลี่ยนหน้ารถ ด้วยการเคลื่อนที่ลักษณะนี้ roller ที่อยู่บน Mecanum wheel จะทำหน้าที่ในการหมุนเคลื่อนที่ได้โดยไม่เสียสมดุล Mecanum wheel ยังมีการเคลื่อนที่แบบพิเศษนั้นคือการหมุนรอบตัวเองโดยใช้การหมุนที่สัมพันธ์กันของล้อ Mecanum wheel ด้านซ้ายทั้งบน และล่างจะหมุนไปในทางเดียวกัน และจะด้านด้วยล้อด้านขวา ทั้งบน และล่างที่หมุนไปในทางตรงข้ามของ ล้อด้านซ้ายทั้งบน และล่าง ตามรูปด้านบนทำให้เกิดการหันของหน้ายานยนต์นั้น ๆ ได้อย่างสมดุล และแนบเนียน



รูปที่ 2.9 ลักษณะของล้อ Mecanum wheel

## 2.4 แบตเตอรี่

### 2.4.1 หลักการทำงาน

แบตเตอรี่จะทำการเก็บไฟฟ้าไว้สำหรับการใช้งานในอนาคต โดยจะสร้างแรงดันไฟฟ้าจากปฏิกิริยาเคมีที่เกิดขึ้นเมื่อวัสดุสองอย่างนั้นมีความแตกต่างกัน เช่น เฟลทที่เป็นบวกและลบที่ถูกจุ่มลงในอิเล็กโทรไลต์ หรือสารละลายของกรดซัลฟิวริกและน้ำ ในแบตเตอรี่ตะกั่ว-กรดโดยทั่วไป แรงดันไฟฟ้าจะอยู่ที่ประมาณ 2 โวลต์ต่อหนึ่งเซลล์รวมทั้งหมด 12 โวลต์ กระแสไฟฟ้าจะไหลจากแบตเตอรี่ทันทีที่มีการต่อวงจรระหว่างขั้วบวกและขั้วลบ สิ่งนี้จะเกิดขึ้นเมื่อโหลดต่าง ๆ จำเป็นต้องใช้ไฟฟ้า ซึ่งได้มีการเชื่อมต่อไว้กับแบตเตอรี่ เช่น วิทยุ

คนส่วนใหญ่ไม่รับรู้ว่าแบตเตอรี่ตะกั่ว-กรดนั้นดำเนินการในกระบวนการชาร์จและการคายประจุอย่างต่อเนื่อง เมื่อแบตเตอรี่ถูกเชื่อมต่อเข้ากับโหลดที่จำเป็นต้องใช้ไฟฟ้า เช่น สตาร์ทเตอร์ในรถยนต์ของคุณ กระแสไฟฟ้าจะไหลออกมาจากแบตเตอรี่ ทำให้แบตเตอรี่เริ่มมีการคายประจุในขั้นตอนกลับกัน แบตเตอรี่จะถูกชาร์จเมื่อกระแสไฟฟ้าไหลกลับเข้าไปยังแหล่งกำเนิด ซึ่งทำให้ความแตกต่างของสารเคมีระหว่างเฟลทกลับเป็นเหมือนเดิม กระบวนการนี้จะเกิดขึ้นเมื่อคุณขับรถที่ไม่มีอุปกรณ์เสริมใด ๆ และตัวกำเนิดไฟฟ้าสลับจะทำให้กระแสไฟฟ้ากลับเข้าไปในแบตเตอรี่ ในขณะที่แบตเตอรี่คายประจุ เฟลทตะกั่วจะกลายเป็นสารเคมีมากขึ้น กรดจะเกิดการ อ่อนตัว และแรงดันไฟฟ้าจะลดลง ในที่สุดแบตเตอรี่จะคายประจุออกมาจนหมดกระทั่งไม่สามารถส่งกระแสไฟฟ้าด้วยแรงดันไฟฟ้าที่มีประสิทธิภาพได้อีกต่อไป

คุณสามารถชาร์จแบตเตอรี่ที่คายประจุได้โดยการปล่อยกระแสไฟฟ้ากลับคืนเข้าไปในแบตเตอรี่ การชาร์จที่เต็มประสิทธิภาพจะทำให้เกิดความแตกต่างทางเคมีระหว่างแผ่นเพลท ทำให้แบตเตอรี่พร้อมที่จะส่งมอบพลังงานได้อย่างเต็มที่ที่กระบวนการที่มีเอกลักษณ์เฉพาะในการคายประจุและการชาร์จไฟในแบตเตอรี่ตะกั่ว-กรดนั้นหมายถึงพลังงานสามารถถูกปล่อยออกไปและกลับเข้ามาใหม่ครั้งแล้วครั้งเล่า นี่คือสิ่งที่เรียกว่าความสามารถในการหมุนรอบของแบตเตอรี่

## 2.4.2 ชนิดของแบตเตอรี่

แบตเตอรี่มี 4 ชนิดได้แก่ 1.แบตเตอรี่น้ำ (Flood) 2.เจล (Gel) 3.ตะกั่ว (Lead) 4.ลิเทียม (Lithium) ในที่นี้จะพูดถึงแบตเตอรี่ที่ใช้ในงานโซล่าเซลล์นะครับ จะไม่ได้ลงลึกถึงโครงสร้างของแบตเตอรี่ เพื่อให้สามารถเลือกใช้แบตเตอรี่ตามความต้องการได้

### 2.4.2.1 แบตเตอรี่น้ำ (Flood)

คือ แบตเตอรี่ที่ใช้กันทั่วไป ที่คุ้น ๆ กันคือแบตเตอรี่รถยนต์ ที่ต้องเติมน้ำกลั่น ราคาถูก เมื่อก่อนสมัยที่โซล่าเซลล์ พังเข้ามาใหม่ ๆ ได้ใช้แบตเตอรี่ สามารถทำให้ระบบทำงานได้ แต่ก็ไม่ประสบความสำเร็จมากนัก เพราะเนื่องจากแบตเตอรี่ต้องเติมน้ำกลั่นทุกสัปดาห์ ทำให้เวลาที่ติดตั้งแล้วการที่จะเติมน้ำกลั่นที่เสาไฟถนนโซล่าเซลล์เป็นเรื่องที่ยาก ยิ่งจำนวนเสาไฟถนนโซล่าเซลล์เยอะ ยิ่งมีความลำบากไปอีก

### 2.4.2.2 เจล (Gel)

คือ แบตเตอรี่ที่เป็นระบบปิด ไม่มีการถ่ายเทอากาศ และไม่ต้องการการดูแลหรือการเติมน้ำกลั่น ทำให้ช่วงหนึ่งมีการใช้กันมากขึ้น ตามทฤษฎี เพื่อลดปัญหาการรั่วไหลของน้ำกลั่นออกจากแบตเตอรี่ ทำให้มีข้อเสียในเรื่องของการใช้งานที่ไม่อาจเทียบได้กับแบตเตอรี่ชนิดตะกั่ว (Lead) แต่จากการทดสอบประสิทธิภาพแล้วก็พอกับแบตเตอรี่แบบตะกั่ว และราคายังถูกกว่าอีกด้วย

### 2.4.2.3 ตะกั่ว (Lead)

คือ แบตเตอรี่ที่เป็นระบบปิดเหมือนกับแบตเตอรี่เจล (Gel) รู้จักกันในชื่อแบตเตอรี่กรดตะกั่ว, ตะกั่วกรด (Lead acid) ที่ตอนนี้ใช้กันอย่างแพร่หลาย มีความทนทานสูง การดูแลรักษาทำได้ง่าย เนื่องจากแบตเตอรี่ไม่ต้องเติมน้ำกลั่นแล้ว หากซื้อมาแล้วไม่ได้ใช้งานต้องอัดประจุแบตเตอรี่ซ้ำทุก 3 เดือนทำให้สามารถเก็บแบตเตอรี่ไว้ได้นาน

#### 2.4.2.4 ลิเทียม (Lithium)

คือ แบตเตอรี่ที่มีประสิทธิภาพสูงสุด แบ่งเป็น 2 แบบ (จริงๆ แล้วแบตเตอรี่ลิเทียมมีหลายแบบ แต่ที่เราใช้มี 2 แบบ)

1. แบตเตอรี่ลิเทียมไอออน (Lithium-Ion Battery)

2. แบตเตอรี่ลิเทียมไอออนฟอสเฟต (Lithium Iron Phosphate)(LiFePO4)

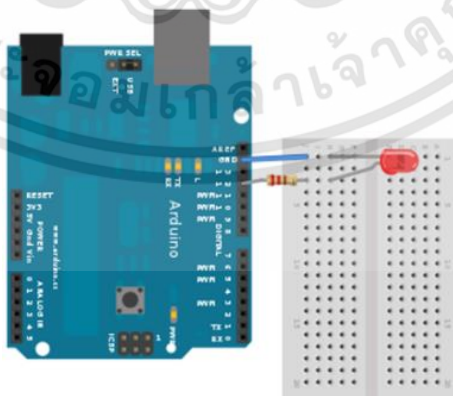
- แบตเตอรี่ลิเทียมไอออน (Lithium-Ion Battery) เป็นแบตเตอรี่ที่ค่าการจ่ายไฟที่แรง และคงที่ มีระยะเวลาการชาร์จประจุไฟจนเต็มได้ไวกว่า ราคาที่แพงกว่าแบตเตอรี่ชนิดอื่นเช่นกัน

- แบตเตอรี่ลิเทียมไอออนฟอสเฟต (Lithium Iron Phosphate)(LiFePO4) เป็นแบตเตอรี่ที่พัฒนามาจากแบตเตอรี่ลิเทียมไอออน (Lithium-Ion Battery) ทนต่อความร้อนที่เกิดจากปฏิกิริยาเคมี ซึ่งสามารถให้พลังงานที่สูงกว่า ไม่เป็นพิษ มีอายุการใช้งานที่มากกว่าแบตเตอรี่รุ่นเก่า ราคาจึงสูง

### 2.5 Arduino

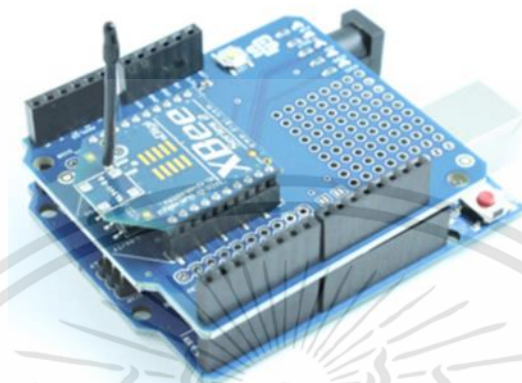
Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด (ดูตัวอย่างรูปที่ 2.10) หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ (ดูตัวอย่างรูปที่ 2.11)



รูปที่ 2.10 บอร์ดต่อเข้ากับ LED

เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

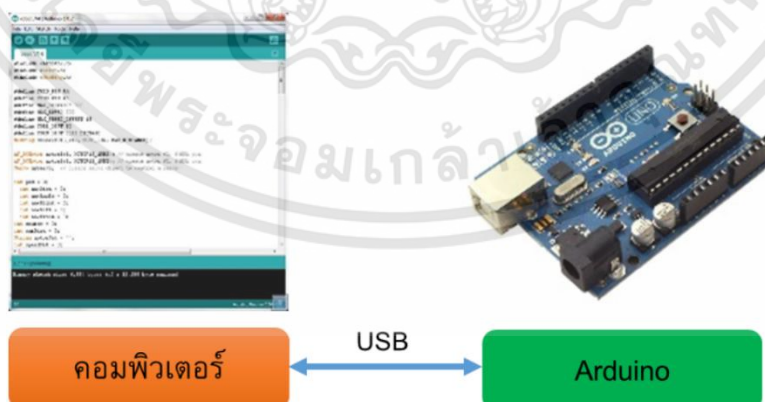


รูปที่ 2.11 บอร์ดต่อเข้ากับ XBee Shield

### 2.5.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

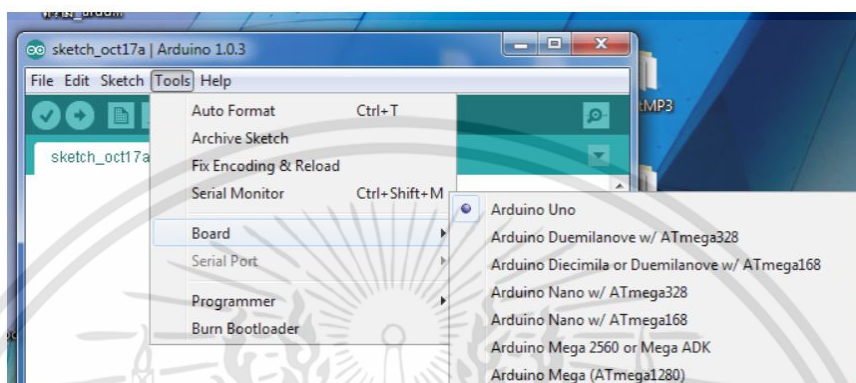
- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแกร่ง
- Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- ราคาไม่แพง
- Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

### 2.5.2 รูปแบบการเขียนโปรแกรมบน Arduino

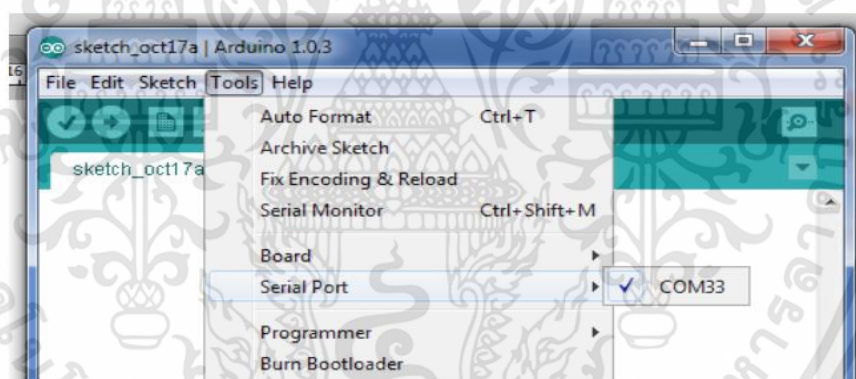


รูปที่ 2.12 การเชื่อมต่อ Arduino เข้ากับคอมพิวเตอร์

1. เขียนโปรแกรมบนคอมพิวเตอร์ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก [Arduino.cc/en/main/software](http://Arduino.cc/en/main/software)
2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้วให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com port

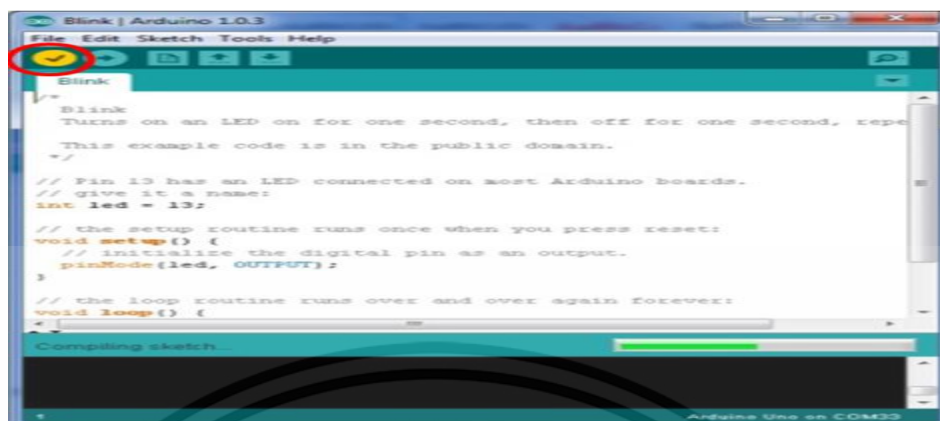


รูปที่ 2.13 เลือกุ่นบอร์ด Arduino ที่ต้องการ upload

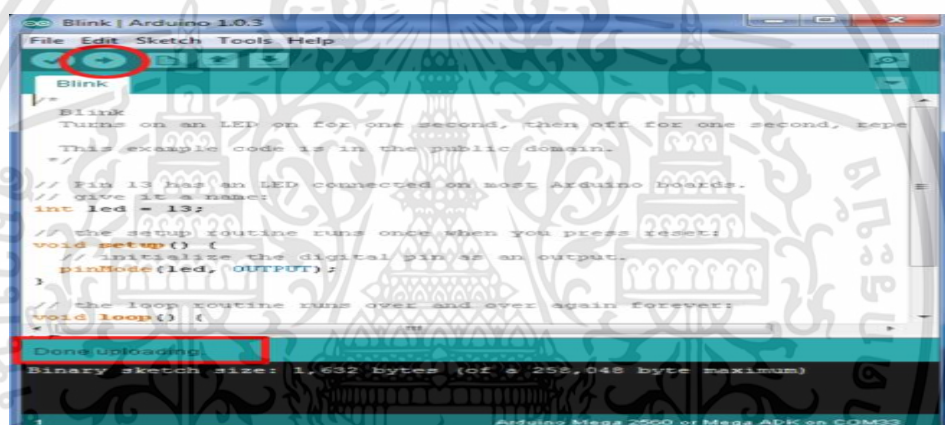


รูปที่ 2.14 เลือกหมายเลข Comport ของบอร์ด

3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที

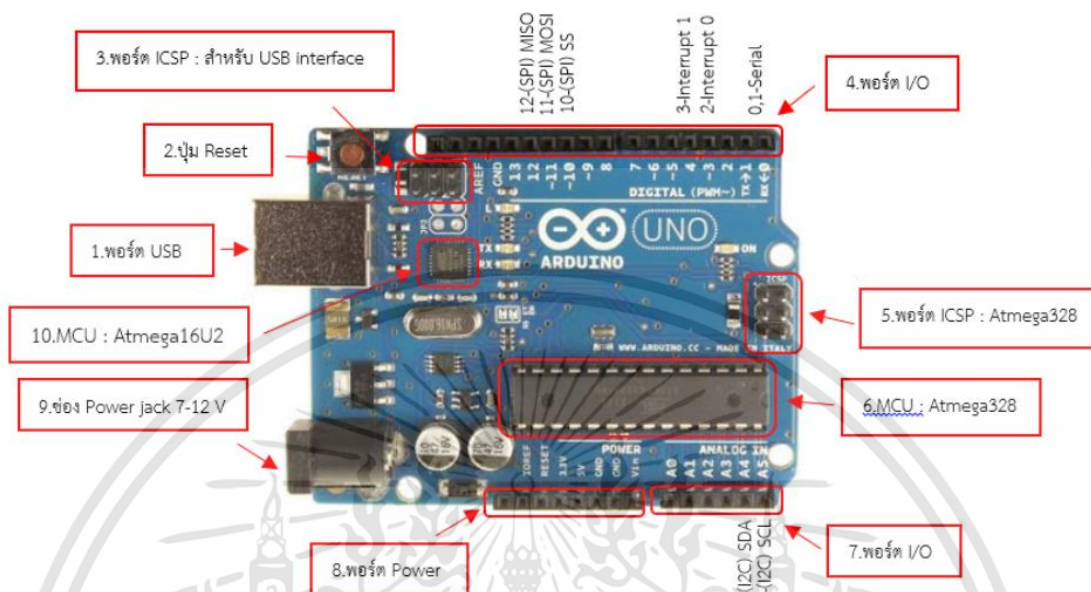


รูปที่ 2.15 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง และ Compile โค้ดโปรแกรม



รูปที่ 2.16 Upload โค้ดโปรแกรม

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



รูปที่ 2.17 Model : Arduino UNO R3

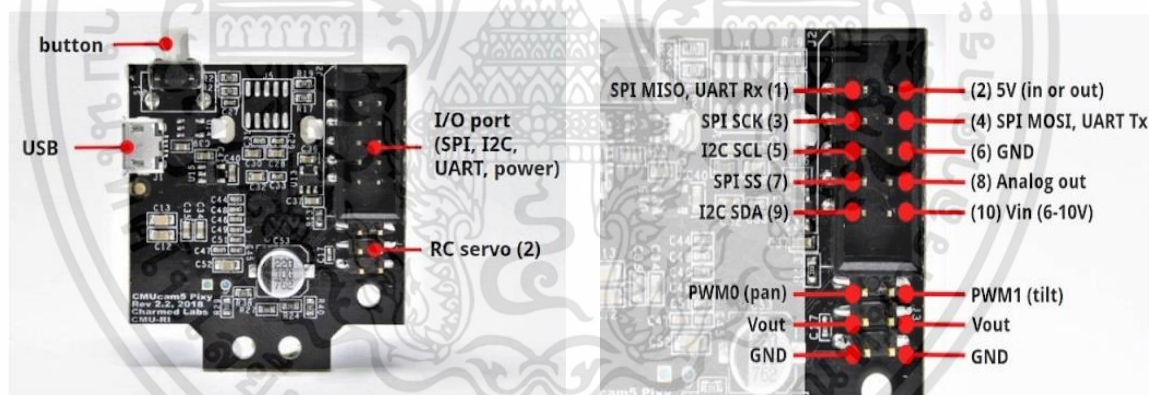
1. USB Port: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port
4. I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่น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

## 2.6 กล้อง PIXY 2

PIXY2 โมดูลกล้องตรวจจับสีและภาพระยะไกลได้รับการปรับปรุงจาก PIXY1 หลายด้าน อาทิ ความเร็วในการตรวจจับภาพเพิ่มเป็น 60 เฟรมต่อวินาที, มีขนาดโดยรวมเล็กลง, มีโหมดสำหรับการตรวจจับเส้นเพื่อนำไปใช้สร้างหุ่นยนต์เคลื่อนที่ตามเส้นแบบฉลาด เพราะมันสามารถตรวจจับทางแยกและเครื่องหมายจราจรได้ด้วยเชื่อมต่อกับบอร์ดไมโครฯ สมัยใหม่ได้ทุกบอร์ดผ่านทาง UART หรือ I<sup>2</sup>C หรือ SPI

### 2.6.1 ส่วนประกอบของกล้อง PIXY 2

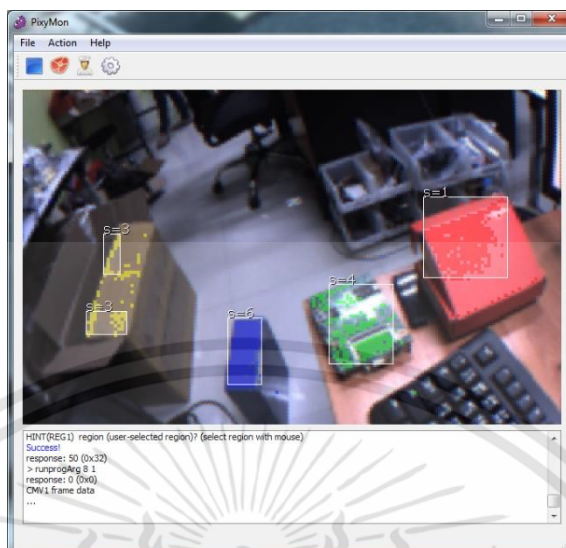
1. Button คือ ปุ่มที่ใช้เลือกวัตถุที่ต้องการตรวจจับ ตามรูปแบบที่บันทึกไว้ 1-7 และเป็นปุ่มที่ใช้เริ่มและหยุดการตรวจจับอีกด้วย
2. USB คือ พอร์ตที่ใช้ในการเชื่อมต่อไปยังคอมพิวเตอร์หรืออุปกรณ์อื่นๆ เพื่อกำหนดค่าต่างๆ
3. I/O Port (UART, SPI, I2C etc.) คือ พอร์ตที่ใช้ในการเชื่อมต่อกับไมโครคอนโทรเลอร์ในรูปแบบต่างๆ สามารถดูรายละเอียดจากภาพด้านล่าง
4. RC Servo(2) คือ พอร์ตที่ใช้ในการเชื่อมต่อกับ Servo สามารถต่อได้สูงสุดถึง 2 Servo



รูปที่ 2.18 แสดง Port การเชื่อมต่อของกล้อง PIXY 2

## 2.7 Pixy Mon

PixyMon เป็นแอปพลิเคชันที่ช่วยให้สามารถกำหนดค่า Pixy2 และดูผ่านโปรแกรมให้เห็นว่าทราบถึงค่า “Signature” ที่เราทำการกำหนดค่าไปใช้งาน รวมทั้งทำงานบนแพลตฟอร์มที่แตกต่างกันหลายรวมถึง Windows, MacOS และ Linux รวมถึงระบบฝังตัวขนาดเล็กอื่น ๆ เช่น Raspberry Pi BeagleBone Black และ Arduino



รูปที่ 2.19 แสดงโปรแกรม Pixy Mon

## 2.8 L298N motor drive

เมื่อพูดถึงการใช้งานมอเตอร์กระแสตรงแล้ว หลีกเลียงไม่ได้ที่จะนึกถึง หุ่นยนต์ หรือแม้กระทั่งรถบังคับ ส่วนใหญ่จะใช้มอเตอร์เป็นตัวขับเคลื่อนการทำงาน เช่น การเคลื่อนที่ของรถบังคับ หรือ แม้กระทั่งใช้เป็นกลไกในการทำงานของของหุ่นยนต์ เมื่อพูดถึงมอเตอร์ การที่จะต้องใช้งานมอเตอร์ทำงานอย่างมีประสิทธิภาพทั้ง การควบคุมความเร็วรอบของมอเตอร์และการควบคุมการหมุนทวนเข็มและตามเข็ม สิ่งเหล่านี้ทำได้ แต่ต้องมีโมดูลเสริมหรือตัวกำหนดตัวแปรที่จะทำให้สามารถควบคุมการทำงานของมอเตอร์ได้ ซึ่งโมดูลที่ใช้ได้มีด้วยกันหลายรุ่น แต่ในที่นี้จะขอยกตัวอย่างเอารุ่น L298N motor drive มาเป็นกรณีศึกษา

L298N เป็นชุดขับมอเตอร์ชนิด H-Bridge ซึ่งส่วนใหญ่จะถูกนำไปใช้ในการควบคุมทิศทางและความเร็วของมอเตอร์ ซึ่งสามารถควบคุมมอเตอร์ได้ทั้งหมด 2 Channel

### 2.8.1 หลักการทำงาน

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

PWM หมายถึง การควบคุมช่วงจังหวะการทำงานของอิเล็กทรอนิกส์ ลองจินตนาการถึงแปรงขดลวดในมอเตอร์เป็นระหัดวิดน้ำและอิเล็กทรอนิกส์เป็นน้ำที่ตกลงมาจากระหัดวิดน้ำค่าแรงดันไฟฟ้าก็คล้ายกับกระแสน้ำที่ไหลผ่านระหัดวิดน้ำด้วยความเร็วคงที่ ยิ่งกระแสน้ำไหลเร็วเท่าไรก็จะหมายความว่า

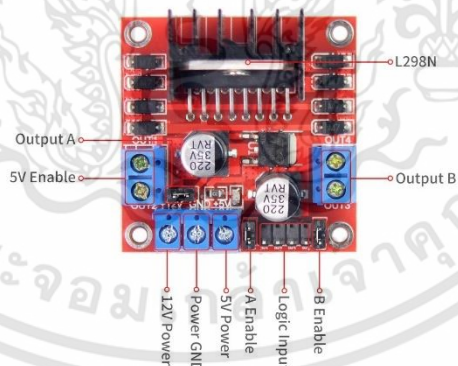
แรงดันไฟฟ้ายิ่งสูงขึ้น แต่มอเตอร์มีอัตราความเร็วคงที่และสามารถเสียหายได้หากมีแรงดันไฟฟ้าสูงไหลผ่านหรือหยุดทันทีเพื่อที่จะหยุดมอเตอร์ ดังนั้น PWM คล้ายกับการควบคุมระดับน้ำให้ตกน้ำในจังหวะคงที่ที่กระแสน้ำคงที่ ยิ่งระดับน้ำหมุนเร็วเท่าไรช่วงของ pulse ก็จะยาวขึ้น ในทางกลับกันถ้าระดับน้ำหมุนช้าช่วงของ pulse จะสั้นลง ดังนั้นเพื่อยืดอายุการใช้งานของมอเตอร์จึงควรที่จะควบคุมมอเตอร์ด้วย PWM

## 2.8.2 พิจารณาโครงสร้างโค้ดของ Arduino

โค้ดการทำงานของ Arduino มีการพัฒนาขึ้นเรื่อยๆ แต่ยังไม่มีการสร้าง library ที่เกี่ยวกับตัว L298N Dual H-Bridge เพื่อควบคุมมอเตอร์ ดังนั้นผู้ใช้งานจึงต้องประกาศ pin เพื่อใช้งานขึ้นมาเองสามารถใช้โค้ด `int dir(number)Pin(letter)` ต่อเข้ากับพินดิจิทัลที่เลือกใช้ แค่นี้ก็สามารถทำงานได้อย่างถูกต้องและช่วยให้ตัว L298N Dual H-Bridge ควบคุมมอเตอร์ได้อย่างมีประสิทธิภาพกับบอร์ด 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()`



รูปที่ 2.20 แสดงรายละเอียดของบอร์ด L298N Motor Drive

## 2.9 Wireless Joystick

### 2.9.1. คุณสมบัติทางเทคนิค

- ใช้สำหรับติดต่อกับแผงวงจรไมโครคอนโทรลเลอร์เพื่อรับส่งข้อมูลจากจอยสติ๊กไร้สาย
- ความถี่ในการสื่อสารข้อมูล 2.4GHz ระยะทำการ 8 เมตร
- ใช้งานจอยสติ๊กไร้สายได้พร้อมกันสูงสุด 20 คู่ (จอยสติ๊กไร้สาย 1 ตัวและกล่องรับส่งคลื่นวิทยุ 1 ตัว)
- ตัวจอยสติ๊กไร้สายทำงานเข้ากันได้กับตัวควบคุมหรือจอยสติ๊กสำหรับเครื่องเล่นเกม PlayStation2 ของ Sony โดยทำงานในโหมดคานาล็อก ใช้งานปุ่มควบคุมได้ครบทุกปุ่ม มีไฟแสดงสถานะการเชื่อมต่อ ที่ตัวจอยสติ๊กด้วย ใช้แบตเตอรี่ AAA ชนิดดิกิตัล 2 ก้อนเป็นแหล่งจ่ายไฟเลี้ยง
- กล่องรับส่งคลื่นวิทยุใช้ไฟเลี้ยง +3.3V หรือ +5V ก็ได้ โดยต่อกับแผงวงจร ZX-PSX2 กินกระแสไฟฟ้า 200mA มีไฟแสดงสถานะการเชื่อมต่อ
- แผงวงจรเชื่อมต่อ ZX-PSX2 รองรับการทำงานทั้งกับจอยสติ๊กไร้สายและแบบมีสาย มีจุดต่อเพื่อ เชื่อมต่อกับไมโครคอนโทรลเลอร์ 2 แบบคือ JST 2 มม. 3 ขา ตัวผู้ 4 ตัว และแบบ IDC 2.54 มม. 6 ขา ทั้งตัวผู้และตัวเมีย
- ใช้งานกับบอร์ดไมโครคอนโทรลเลอร์ได้ทุกแบบ

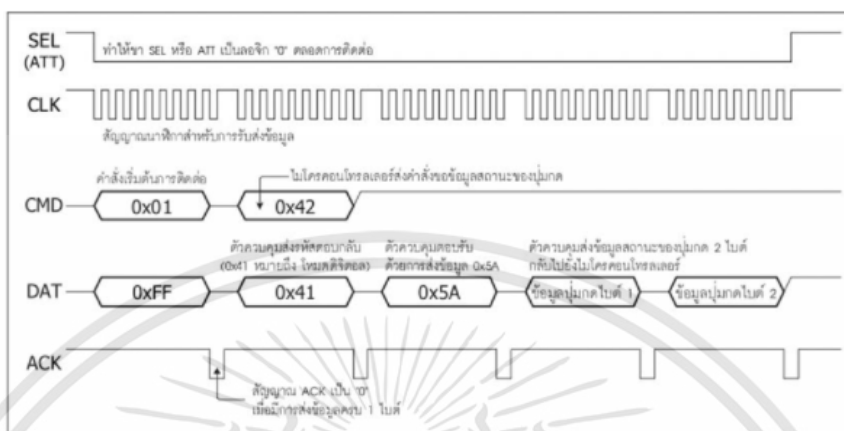
### 2.9.2. สัญญาณที่ใช้ในการเชื่อมต่อจอยสติ๊ก PS2 กับไมโครคอนโทรลเลอร์

ในการนำตัวควบคุมหรือจอยสติ๊ก PS2 มาใช้งานกับไมโครคอนโทรลเลอร์ ก็ทำได้ไม่ยาก สัญญาณที่ใช้งานมีเพียง 4 เส้นคือ DATA, COMMAND, SELECT และ CLOCK

- DATA (DAT) เป็นสัญญาณข้อมูลปุ่มของตัวควบคุมหรือจอยสติ๊กที่ถูกกด จะถูก ส่งมาจากตัวควบคุมหรือจอยสติ๊ก โดยข้อมูลที่ส่งมามีขนาด 8 บิต โดยส่งบิต LSB คมมาก่อน
- COMMAND (CMD) เป็นสัญญาณคำสั่งสำหรับตัวควบคุมหรือจอยสติ๊ก สัญญาณ นี้ ไมโครคอนโทรลเลอร์จะเป็นผู้ส่ง
- ATT SELECT (SEL) เป็นสัญญาณเลือกการทำงานหรือเปิดการติดต่อ ซึ่งทางไมโครคอนโทรลเลอร์ต้องส่งสัญญาณ “0” มาที่ขานี้และต้องทำให้ขานี้เป็นลอจิก “0” ตลอดเวลาที่ติดต่อกัน
- CLOCK (CLK) เป็นสัญญาณนาฬิกาสำหรับกำหนดจังหวะการทำงาน ฝั่งของไมโครคอนโทรลเลอร์จะเป็นฝ่ายสร้างสัญญาณส่งมายังตัวควบคุมหรือจอยสติ๊ก การตอบสนองต่อสัญญาณของอุปกรณ์จะเกิดขึ้นที่ขอบขาของสัญญาณนาฬิกา เมื่อมีการรับ รู้คำสั่ง (COMMAND) จอยสติ๊กจะทำให้ขา ACK กลายเป็นลอจิก “0” (ในที่นี้ไม่ได้ใช้งาน)

เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากตัวควบคุมหรือจอยสติ๊กจะต้องทำให้ขา ATT หรือ SEL เป็นลอจิก “0” แล้วส่งคำสั่งเริ่มต้น 0x01 ไปยังตัวควบคุมหรือจอยสติ๊ก เมื่อตัวควบคุมหรือ จอยสติ๊ก

ได้รับก็จะตอบกลับด้วยรหัสหรือ ID ของตัวเอง เพื่อให้รู้ว่า มันทำงานอยู่ในโหมดใด ซึ่งมีด้วยกัน 4 รหัสหลักคือ



รูปที่ 2.21 ไตอะแกรมเวลาของการติดต่อระหว่างไมโครคอนโทรลเลอร์และตัวควบคุมหรือจอยสติ๊ก PS2

0x41 หมายถึง จอยสติ๊กทำงานในโหมดดิจิทัล (Digital) 0x23 หมายถึง จอยสติ๊กทำงานในโหมด NegCon (ในที่นี้ไม่ใช้งาน)

0x73 หมายถึง จอยสติ๊กทำงานในโหมดอะนาล็อก LED สีแดง (Analog Red LED) - จอยสติ๊กในชุด Wireless Joystick ได้รับการกำหนดให้ทำงานในโหมดนี้เพียงโหมดเดียว

0x53 หมายถึง จอยสติ๊กทำงานในโหมดอะนาล็อก LED สีเขียว (Analog Green LED)

ในช่วงเวลาเดียวกันนั้น ไมโครคอนโทรลเลอร์ต้องส่งคำสั่ง 0x42 เพื่อร้องขอข้อมูลสถานะของ ปุ่มกดจากตัวควบคุมหรือจอยสติ๊ก จากนั้นขา CMD จะเข้าสู่ภาวะสงบ (idle) เพื่อรอให้ตัวควบคุม หรือจอยสติ๊กส่งข้อมูลกลับมายังขา DAT โดยตัวควบคุมหรือจอยสติ๊กจะเริ่มต้นด้วยการส่งข้อมูล 0x5A เพื่อบอกให้ไมโครคอนโทรลเลอร์ทราบว่า ข้อมูล 2 ถึง 6 ไบต์ถัดไปจากนี้คือ ข้อมูลสถานะ ของปุ่มกดทั้งหมด ในรูปที่ 2.21 แสดงไตอะแกรมเวลาที่อธิบายถึงกระบวนการในการติดต่อกันระหว่างไมโครคอนโทรลเลอร์และตัวควบคุมหรือจอยสติ๊ก PS2 เมื่อทำงานในโหมดดิจิทัล



## 2.10 Lux Meter

Lux meter (หรือ Light Meter, เครื่องวัดแสง, เครื่องวัดแสงสว่าง) คือ เครื่องมือวัดชนิดหนึ่งที่ใช้วัดปริมาณความเข้มของแสงสว่างในสภาพแวดล้อมที่ถูกกำหนดไว้ ซึ่งวัดค่าออกมาเป็นหน่วย Lux (ลักซ์) หรือ FC (Foot candle) โดย Lux meter (เครื่องวัดแสง) ปัจจุบันถูกออกแบบมาให้สามารถใช้งานได้หลากหลาย มีความรวดเร็วในการวัด สามารถพกพาได้สะดวก เล็กน้ำหนักเบาและมีความแม่นยำในการวัดสูง ซึ่งเหมาะในการวัดในโรงงานอุตสาหกรรม, อาคารสำนักงาน, โรงพยาบาล, ห้องปฏิบัติการ เป็นต้น

### 2.10.1 หลักการทำงานของเครื่องมือวัดแสง

หลักการทำงานของ Lux meter (เครื่องวัดแสง) ทำงานโดยใช้ Photo Electric cell หรือตัว Sensor ซึ่งถูกติดตั้ง ในโคมพลาสติกสีขาวขุ่น เมื่อจับแสงที่กระทบได้แล้ว จะส่งข้อมูลประมวลผล แปลงคลื่นแสงเป็นประจุไฟฟ้า คำนวณค่าเป็นหน่วย Lux แสดงผลออกมาที่หน้าจอ Digital เพื่อสะดวกในการอ่านค่าที่วัดได้ให้ผู้ใช้งานทราบ

### 2.10.2 ประเภทของเครื่องมือวัดแสง

เครื่องมือวัดแสงมี 2 ประเภท คือ

1. เครื่องมือวัดแสงประเภท Analog หน้าจอในการวัดเป็นเข็ม
2. เครื่องมือวัดแสงประเภท Digital แสดงผลเป็นตัวเลขหน้าจอเป็น LED ซึ่งแบ่งเป็นช่วงการวัดเฉลี่ย 5 point ของ Range ที่ใช้งาน โดยทั่วไปแล้วนิยมใช้เครื่องมือวัดแบบ Digital โดยที่ Calibration Laboratory รับสอบเทียบ Lux meter ได้ถึง 25000 Lux

### 2.10.3 ระดับความเข้มของแสงสว่างทั่วไปโดยประมาณ

- แสงดวงอาทิตย์ 100,000 Lux
- ภายนอกอาคาร 10,000 Lux
- ในโรงงานอุตสาหกรรม 750 – 1,500 Lux
- ใน Supermarket 500 – 1,000 Lux
- Office สำนักงาน 350 – 750 Lux
- บ้านที่อยู่อาศัย 150 – 250 Lux
- โดย  $1 \text{ Lux} = 1 \text{ Lumen/M}^2$  หรือ  $= 0.0929 \text{ FC}$
- $1 \text{ FC} = 1 \text{ Lumen / Ft}^2$  หรือ  $= 10.752 \text{ Lux}$

ในภาคอุตสาหกรรมนั้นเครื่องวัดแสง มีบทบาทสำคัญในการวัดและควบคุมคุณภาพแสง ในสถานที่ปฏิบัติงานให้เป็นไปตามกฎหมายกำหนดไว้ ทำให้ทุกโรงงานอุตสาหกรรมส่วนใหญ่ต้องมีเครื่องมือวัดนี้ไว้ใช้งาน และจะต้องมีการวัดปริมาณของแสงเป็นประจำ อย่างน้อยปีละ 1 ครั้ง หรือเมื่อมี

การเปลี่ยนแปลงสภาพแวดล้อมของสถานที่ทำงาน ยกตัวอย่างเช่น ทาสีผนังใหม่ การเปลี่ยนหลอดไฟ หรือต่อเติมผนัง ฝ้า / เพดาน เป็นต้น

ทั้งหมดนี้เป็นเหตุผลที่ เครื่องวัดแสงมีความจำเป็นอย่างยิ่งที่โรงงาน อุตสาหกรรมต่างต้อง มีใช้ ดังนั้นการเลือกซื้อเครื่องมือวัดแสง Lux meter ก็มีความสำคัญเพื่อให้เหมาะกับการใช้งาน ต้องมีการรับรองมาตรฐานอุตสาหกรรม หรือ มอก.รวมทั้ง สิ่งที่เขาไม่ได้คือ การสอบเทียบเครื่องมือวัด ให้ได้ มาตรฐานทุกปี โดย Calibration Laboratory สามารถสอบเทียบเครื่องมือวัดแสง Lux meter ได้และ ได้รับการรับรองมาตรฐาน ANAB Range 1-20000 Lux และ 1 – 20000 cd/m<sup>2</sup>



รูปที่ 2.24 เครื่องมือวัดแสง

## 2.11 ตัวแปรและฟังก์ชันที่ใช้ในการควบคุมในการเขียนโค้ดรับค่าจากกล้อง PIXY

### 2.11.1 ฟังก์ชัน

```
int8_t getBlocks(bool wait [optional], uint8_t sigmap [optional], uint8_t maxBlocks [optional])
```

getBlocks() ได้รับบล็อกที่ตรวจพบทั้งหมดในเฟรมล่าสุด จากนั้นข้อมูลใหม่จะพร้อมใช้งานในตัวแปรสมาชิกบล็อก บล็อกที่ส่งกลับจะถูกเรียงลำดับตามพื้นที่โดยมีบล็อกที่ใหญ่ที่สุดปรากฏเป็นอันดับแรกในอาร์เรย์บล็อก โดยจะส่งกลับค่าความผิดพลาด (=0) ถ้าสำเร็จการตั้งค่าการรอเป็นเท็จทำให้ getBlocks() กลับมาทันทีหากไม่มีข้อมูลใหม่ (polling mode) การตั้งค่าการรอเป็น true (ค่าเริ่มต้น) ทำให้ getBlocks() บล็อก (รอ) จนกว่าเฟรมถัดไปของข้อมูลบล็อกจะพร้อมใช้งาน หมายเหตุ อาจไม่มีข้อมูลบล็อกถ้าไม่มีการตรวจพบวัตถุ sigmap เป็นบิตแมปของลายเซ็นทั้ง 7 ลายเซ็นที่ต้องการรับข้อมูลบล็อก ตัวอย่างเช่น หากสนใจเฉพาะการบล็อกข้อมูลจากลายเซ็น 1 จะผ่านค่า 1 หากสนใจที่จะบล็อกข้อมูลจากทั้งลายเซ็น 1 และ 2 จะผ่านค่า 3 หากสนใจที่จะบล็อกข้อมูลจากลายเซ็น 1, 2 และ 3 จะผ่านค่า 7 และอื่น ๆ บิตที่สำคัญที่สุด (128 หรือ 0x80) ใช้สำหรับรหัสสี ค่า 255 (ค่าเริ่มต้น) บ่งชี้ว่าสนใจข้อมูลบล็อกทั้งหมดmaxblocks ระบุจำนวนบล็อกสูงสุดที่คุณต้องการรับ ตัวอย่างเช่น การส่งผ่านค่า 1 จะส่งกลับที่ 1 บล็อกมากที่สุด ค่า 255 (ค่าเริ่มต้น) บ่งชี้ว่าสนใจบล็อกทั้งหมด

### 2.11.2 ตัวแปร

- uint8\_t numBlocks = จำนวนบล็อกที่มีอยู่ในตัวแปรสมาชิกบล็อก
- Block Blocks[] = อาร์เรย์นี้มีข้อมูลบล็อกทั้งหมดอันเป็นผลมาจาก getBlocks() บล็อกในอาร์เรย์นี้จัดเรียงตามพื้นที่ โดยบล็อกที่ใหญ่ที่สุดจะปรากฏเป็นอันดับแรกในอาร์เรย์บล็อก
- void print() = การเรียกสิ่งนี้จะพิมพ์ข้อมูลบล็อกไปยังคอนโซล
- uint16\_t m\_signature = ตัวแปรนี้มีหมายเลขลายเซ็นหรือหมายเลขรหัส
- uint16\_t m\_x = ตัวแปรนี้มีตำแหน่ง x ของศูนย์กลางของบล็อก ค่าอยู่ระหว่าง 0 ถึง frameWidth (315)
- uint16\_t m\_y = ตัวแปรนี้มีตำแหน่ง y ของศูนย์กลางของบล็อก ค่าอยู่ระหว่าง 0 ถึง frameHeight (207)
- uint16\_t m\_width = ตัวแปรนี้มีความกว้างของบล็อก ช่วงค่าอยู่ระหว่าง 0 ถึง frameWidth (316)
- uint16\_t m\_height = ตัวแปรนี้มีความสูงของบล็อก ค่าอยู่ระหว่าง 0 ถึง frameHeight (208)
- int16\_t m\_angle = ตัวแปรนี้มีมุมของรหัสเป็นองศา ค่าอยู่ระหว่าง -180 ถึง 180 หากบล็อกเป็นลายเซ็นปกติ (ไม่ใช่รหัส) ค่ามุมจะเป็น 0
- uint8\_t m\_index = ตัวแปรนี้มีดัชนีการติดตามของบล็อก เมื่อ Pixy2 ตรวจพบบล็อกใหม่ มันจะเพิ่มลงในตารางของบล็อกที่กำลังติดตามและกำหนดดัชนีการติดตามให้กับบล็อกนั้น จากนั้นจะพยายามค้นหาบล็อก (และทุกบล็อกในตาราง) ในเฟรมถัดไปโดยค้นหารายการที่ตรงกันที่สุด ดัชนีบล็อกแต่ละรายการจะถูกเก็บไว้สำหรับบล็อกนั้นจนกว่าจะออกจากระยะการมองเห็นของ Pixy2 หรือ Pixy2 ไม่พบวัตถุในเฟรมต่อไป (เนื่องจากการบดเบี้ยว ไม่มีแสง ฯลฯ)
- uint8\_t m\_age = ตัวแปรนี้มีตัวเลข ของเฟรมที่มีการติดตามบล็อกที่กำหนด เมื่ออายุถึง 255 ก็ยังคงอยู่ที่ 255

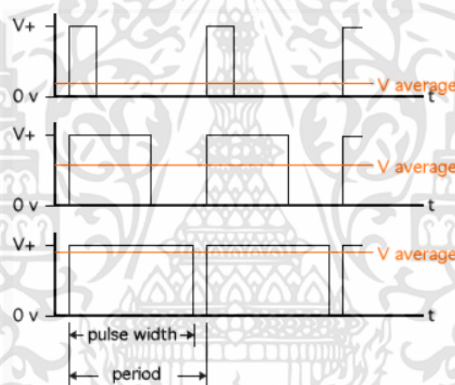
### 2.12 Pulse Width Modulation (PWM)

การควบคุมพลังงานที่ส่งออกไปยังอุปกรณ์ที่ต้องการควบคุม เช่น ความแรงมอเตอร์ ความสว่างของไฟ LED โดยปกติหมายถึงการลดแรงดันที่ส่งออกไปยังมอเตอร์ แต่การลดแรงดันนั้นเป็นแนวทางที่ต้องใช้วงจรที่ซับซ้อนมีความยุ่งยากค่อนข้างมาก ดังนั้นโดยทั่วไปจึงนิยมใช้เทคนิคที่เรียกว่า Pulse Width Modulation (PWM) ซึ่งไม่ได้ลดแรงดัน หากแต่ใช้หลักการเปิด/ปิดมอเตอร์ด้วยความเร็วสูงๆ จนผล

ค่าเฉลี่ยของแรงดันที่ได้ออกมาเทียบเท่ากับการเปลี่ยนแรงดันโดยตรง เทคนิคนี้ทำให้ไม่ต้องใช้วงจรซับซ้อน แต่การเขียนโปรแกรมจะยุ่งยากขึ้นบ้าง

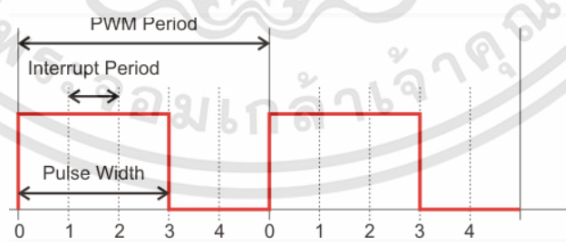
PWM นั้นโดยทั่วไปจะมีการสร้างลูกคลื่นสี่เหลี่ยม (Square Wave) ออกมาโดยกำหนดคาบของสัญญาณ (Period) ให้สั้นๆ ซึ่งปกติคาบจะมีค่าไม่เกิน 33 ms (30 Hz) สำหรับการทดลองทั่วไป และอาจมีค่าน้อยถึง 0.01 ms (100KHz) หรือน้อยกว่าในงานอุตสาหกรรมบางชนิด หลักการสำคัญของ PWM คือการปรับเปลี่ยนความกว้างของลูกคลื่นในแต่ละคาบ โดยถ้าลูกคลื่นสั้นก็จะทำให้แรงดันเฉลี่ยที่ออกมามีค่าน้อย และถ้าลูกคลื่นยาวแรงดันเฉลี่ยก็จะมีค่ามากขึ้น จากรูปด้านล่าง V เฉลี่ย (เส้นสีส้ม) จะสูงหรือต่ำขึ้นขึ้นอยู่กับความกว้างของลูกคลื่น ซึ่งความกว้างของลูกคลื่นนี้เรียกว่า pulse width หรือ Duty Cycle

Pulse width จะต้องน้อยกว่าค่าความยาวคาบเสมอ Duty Cycle จะมีหน่วยเป็น % ของความยาวคาบ เช่น ถ้าคาบ = 10ms และ Duty Cycle = 40% นั้นหมายความว่า Pulse width =  $10\text{ms} \times 0.4 = 4\text{ms}$  เป็นต้น



รูปที่ 2.25 Pulse width

### 2.12.1 Algorithm การสร้าง PWM โดยใช้ Timer Interrupt



รูปที่ 2.26 การสร้าง PWM โดยใช้ Timer Interrupt

### 2.12.1.1 ตัวแปรที่สำคัญมีดังนี้

1. PWM Period = คาบของสัญญาณคลื่นที่สร้างขึ้น
2. Interrupt Period = คาบของ Timer Interrupt ซึ่งจะเท่ากับความกว้างของลูกคลื่นที่เล็กที่สุด
3. Pulse Width = ค่าความกว้างของลูกคลื่นที่ต้องการ โดยในตัวอย่างแบ่งเป็น 6 ระดับ ตั้งแต่ 0-5

### 2.12.2 ข้อควรระวัง

1. กำหนดความถี่ของสัญญาณ PWM ที่เหมาะสมสำหรับการขับ LED – โดยความถี่ที่สูงก็จะยิ่งทำให้ LED ที่แสดงผลเนียนเรียบ (ไม่กะพริบ) ซึ่งแลกด้วยภาระการทำงานของ CPU ที่หนักขึ้น โดยทั่วไป PWM สำหรับ LED ไม่ควรน้อยกว่า 30 Hz แต่ถ้าให้เนียนจริงๆ ก็จะเป็น 100 Hz ขึ้นไป

2. ใช้ Timer Interrupt สร้าง PWM – ดังรายละเอียดต่อไปนี้

- Duty Cycle ต้องการกี่ระดับ – Algorithm นี้จะกำหนด Duty Cycle เป็นจำนวนระดับ เช่น ถ้ากำหนดว่า 5 ระดับ ก็เท่ากับบอกว่า Duty Cycle ที่จะใช้ได้คือ 0%, 25%, 50%, 75%, 100%

- ตั้งค่า Timer ให้คาบของ Interrupt (Interrupt Period) มีค่าเท่ากับความกว้างของ pulse ที่เล็กที่สุด (ระยะเวลาระหว่างเส้นประในรูป) เช่นในตัวอย่างข้างต้น ถ้าคาบของ PWM (PWM Period) = 10 ms ค่า Interrupt Period ก็จะมีค่า =  $10/5 = 2$  ms (ข้อสังเกต! คาบของ Timer Interrupt เป็นคนละตัวกับคาบของ PWM)

- สร้างตัวแปรไว้สองตัว คือ dutyCycle จะเก็บค่า duty ของสัญญาณ PWM เช่นถ้ากำหนดว่ามี 6 ระดับ (ดังใน diagram ข้างต้น) ค่าตัวแปรนี้เทียบกับ duty จะเป็นดังนี้

0 = 0%

1 = 20%

2 = 40%

3 = 60%

4 = 80%

5 = 100%

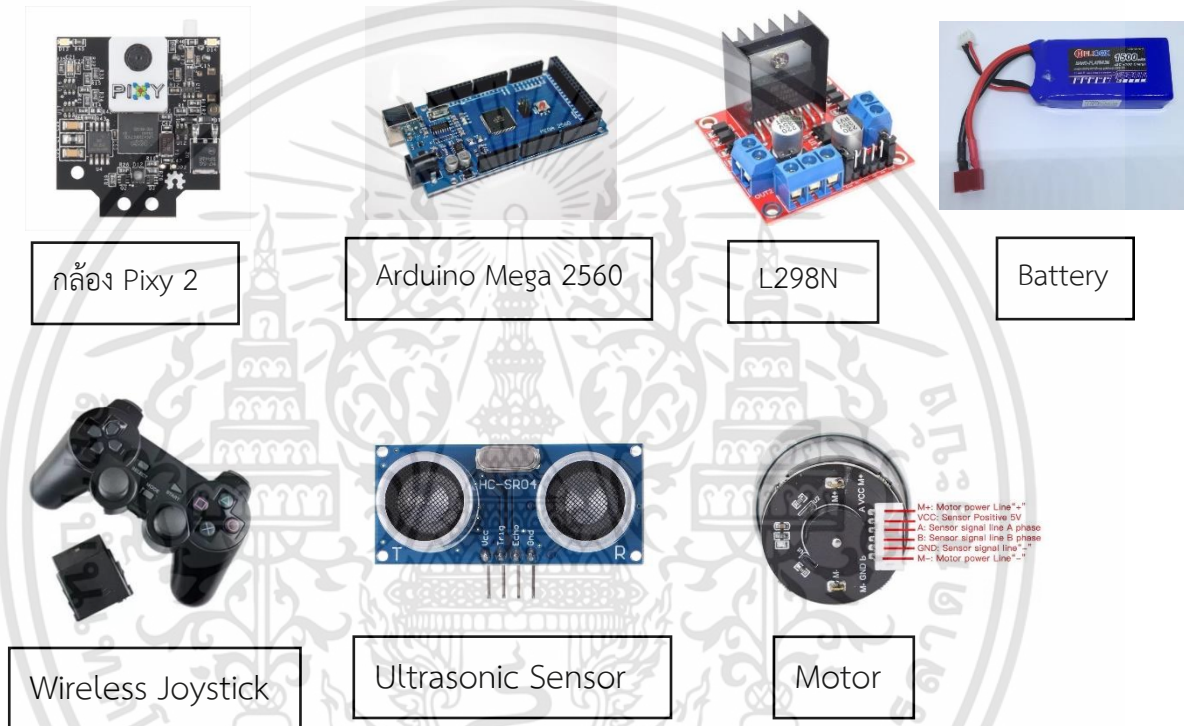
interruptCounter – มีค่าเริ่มต้นเป็น 0 และจะเพิ่มค่าทุกครั้งที่เกิด Timer Interrupt ถ้า interrupt ครบ 6 ครั้ง ค่านี้จะถูก set กลับไปเป็น 0

## บทที่ 3

### การออกแบบและการทำงาน

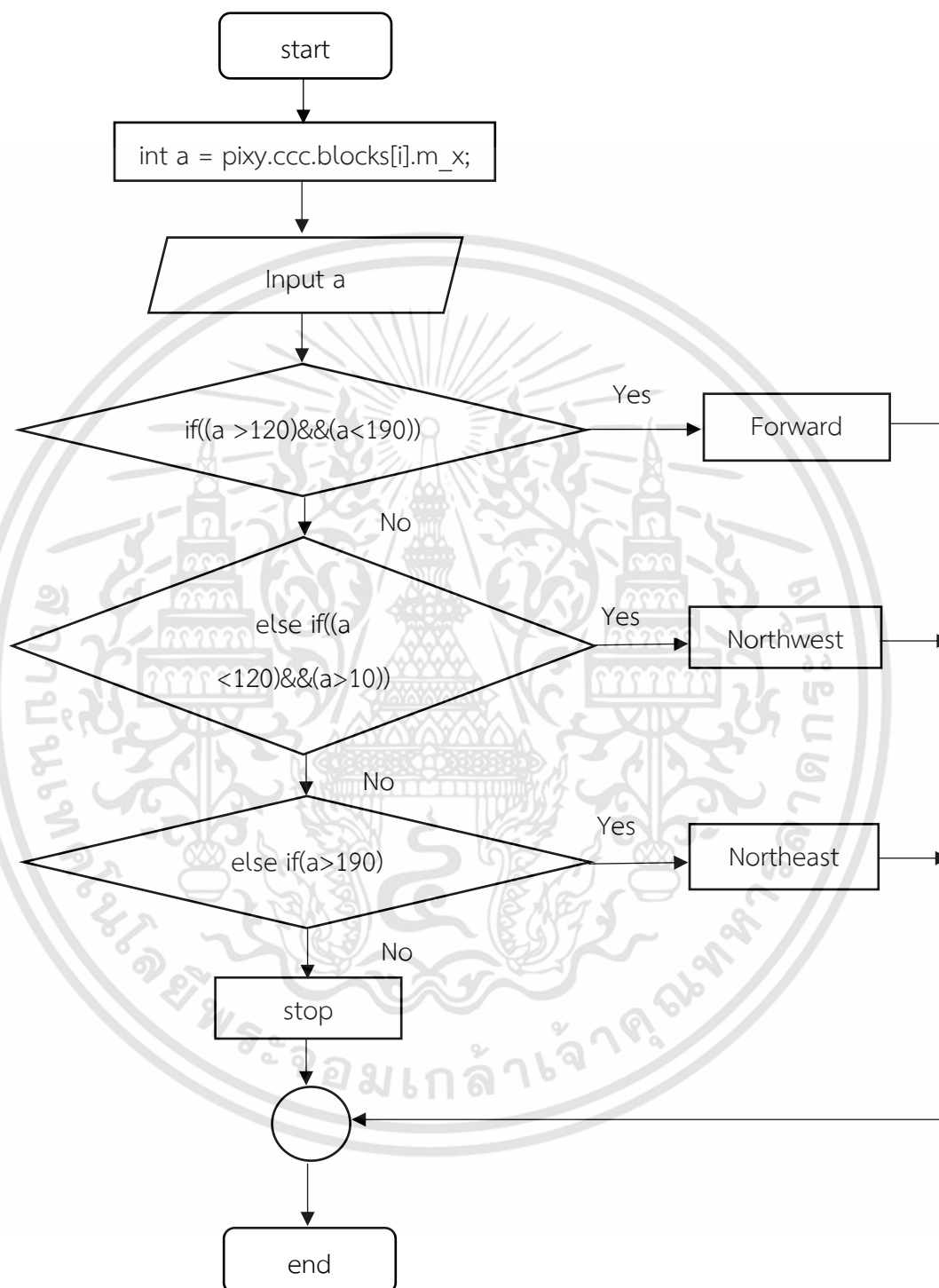
#### 3.1 การออกแบบระบบ

##### 3.1.1 ระบบทั้งหมดประกอบไปด้วย

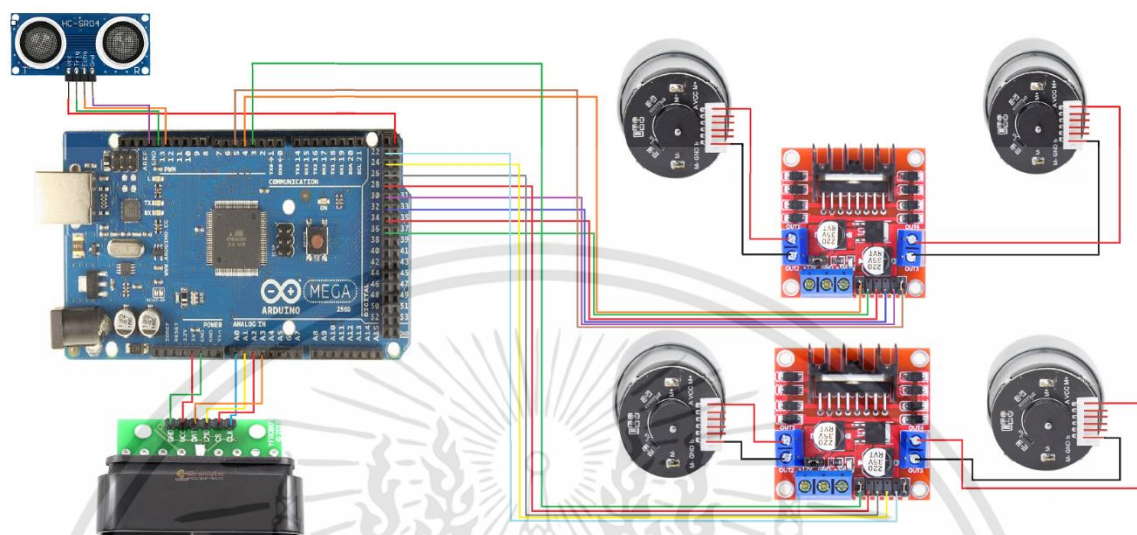


โครงสร้างของระบบโดยรวมประกอบไปด้วยกล้อง Pixy 2 บอร์ดไมโครคอนโทรลเลอร์ วงจรไดรฟ์ควบคุมแบบมอเตอร์ไฟฟ้ากระแสตรง 12 VDC แบตเตอรี่ 12 VDC จอยสติ๊กแบบไร้สายพร้อมตัวปล่อยสัญญาณเชื่อมต่อของจอย และมอเตอร์ไฟฟ้ากระแสตรงขนาด 12 VDC เริ่มจากกล้องทำการจับสีที่ทำการเซตไว้เมื่อกล้องอ่านค่าในแกน X ได้จะทำการส่งค่าไปยังบอร์ดไมโครคอนโทรลเลอร์ให้บอร์ดประมวลผลตามที่เขียนโปรแกรมควบคุมกำหนดไว้เพื่อทำการควบคุมทิศทางในการเคลื่อนที่ของรถให้รถไปถึงจุดหมายที่กำหนด

### 3.1.2 ขั้นตอนการทำงานของระบบ

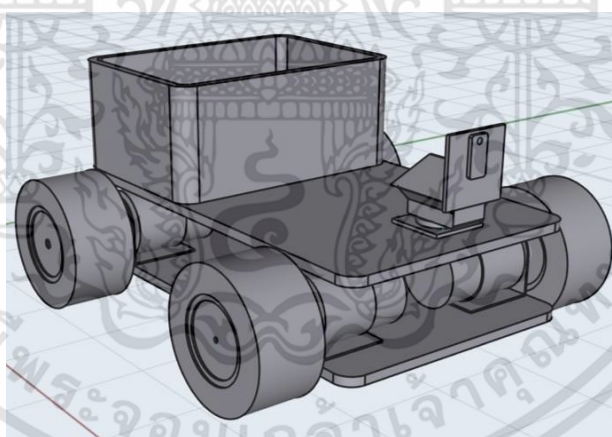


### 3.2 วงจรของระบบ



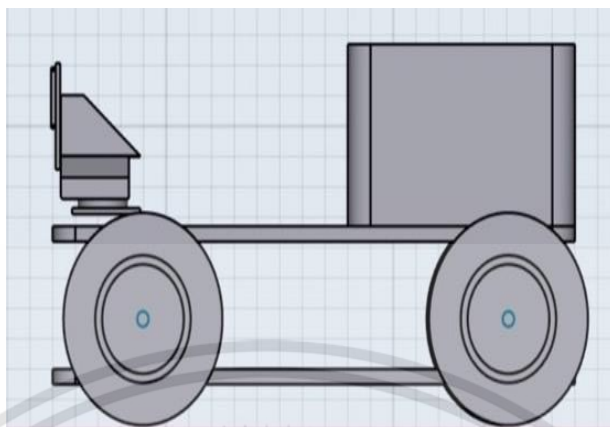
รูปที่ 3.1 วงจรการทำงานของระบบ

### 3.3 โครงสร้างของรถ

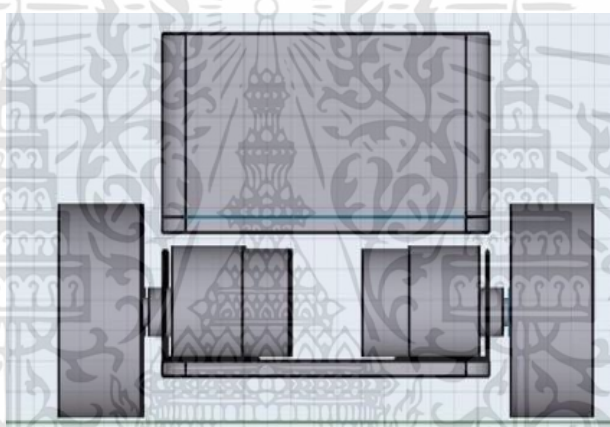


รูปที่ 3.2 โครงสร้างด้านหน้าของรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 โครงสร้างด้านข้างของรถ

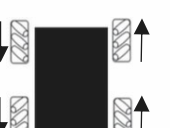
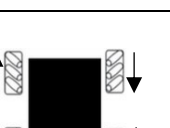
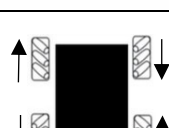


รูปที่ 3.4 โครงสร้างด้านหลังของรถ

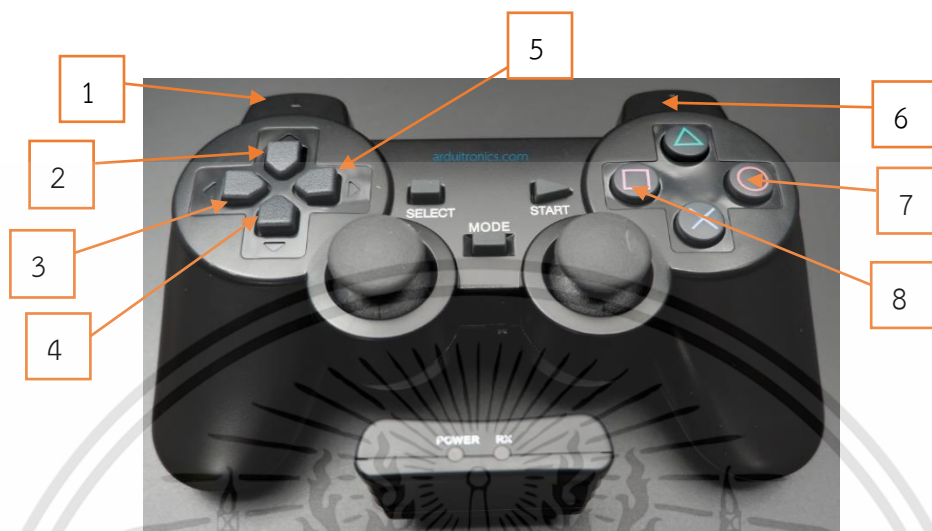
### 3.4 ออกแบบการเคลื่อนที่ของล้อแม็กคานัม(Mecanum)

ล้อแม็กคานัมเป็นการออกแบบให้เป็นล้อทรงกลมที่ลูกกลิ้งอิสระทำมุม 45 องศาติดอยู่ที่ขอบรอบ ๆ ล้อเพื่อให้ล้อสามารถหมุนได้อย่างแม่นยำและสามารถเปลี่ยนทิศทางการวิ่งอย่างอิสระแบบ 90 องศาได้ในทันทีโดยไม่ต้องหมุนรถสามารถให้รถพาหนะขับเคลื่อนอัตโนมัติมีอิสระในการเคลื่อนไหวที่สูงกว่าการใช้ล้อแบบปกติสามารถเคลื่อนที่ในพื้นที่ จำกัด ได้อย่างมีประสิทธิภาพและเพื่อรองรับการหมุนด้านข้างของรถที่ติดตั้งล้อแม็กคานัมต้องมีการสั่งการด้วยระบบคอนโทรลแบบแยก 4 ล้อเพราะการทำงานของล้อแม็กคานัมใช้ความสัมพันธ์ในการหมุนของล้อทั้ง 4 ล้อโดยหลักการทำงานของระบบเลี้ยวแบบล้อแม็กคานัมมีรูปแบบในการทำงานดังตารางที่ 3.1

ตารางที่ 3.1 การเคลื่อนที่และทิศทางของล้อแม็กคานัม

| ทิศทางของล้อ                                                                        | ทิศทางการเคลื่อนที่                                                                              | ทิศทางของล้อ                                                                         | ทิศทางการเคลื่อนที่                                                                                        |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
|    | เดินหน้า<br>    |    | ตะวันตกเฉียงเหนือ<br>   |
|  | ถอยหลัง<br>   |  | ตะวันตกเฉียงเหนือ<br> |
|  | สไลด์ซ้าย<br> |  | ตะวันออกเฉียงใต้<br>  |
|  | สไลด์ขวา<br>  |  | ตะวันตกเฉียงใต้<br>   |
|  | หมุนซ้าย<br>  |  | เลี้ยวขวา<br>         |
|  | หมุนขวา<br>   |  | เลี้ยวซ้าย<br>        |

### 3.5 ออกแบบการทำงานของจอยสติ๊กกับการเคลื่อนที่ของล้อแม็กคาน้ำ



รูปที่ 3.5 บอกตำแหน่งของการควบคุมล้อแม็กคาน้ำ

ตำแหน่งที่ 1 โหมดการทำงานที่ 1 (L1) คือ โหมดที่ทำการควบคุมล้อแม็กคาน้ำด้วยระบบจอยสติ๊ก

ตำแหน่งที่ 2 ให้การเคลื่อนที่ไปข้างหน้า

ตำแหน่งที่ 3 ให้การเคลื่อนที่เลี้ยวซ้าย

ตำแหน่งที่ 4 ให้การเคลื่อนที่ถอยหลัง

ตำแหน่งที่ 5 ให้การเคลื่อนที่เลี้ยวขวา

ตำแหน่งที่ 6 โหมดการทำงานที่ 2 (R1) คือ โหมดระบบอัตโนมัติโดยจะทำงานเมื่อกำลังจับสปีดที่กำหนดไว้และเมื่อเข้าโหมดที่ 2 จะไม่สามารถควบคุมรถผ่านจอยสติ๊กได้

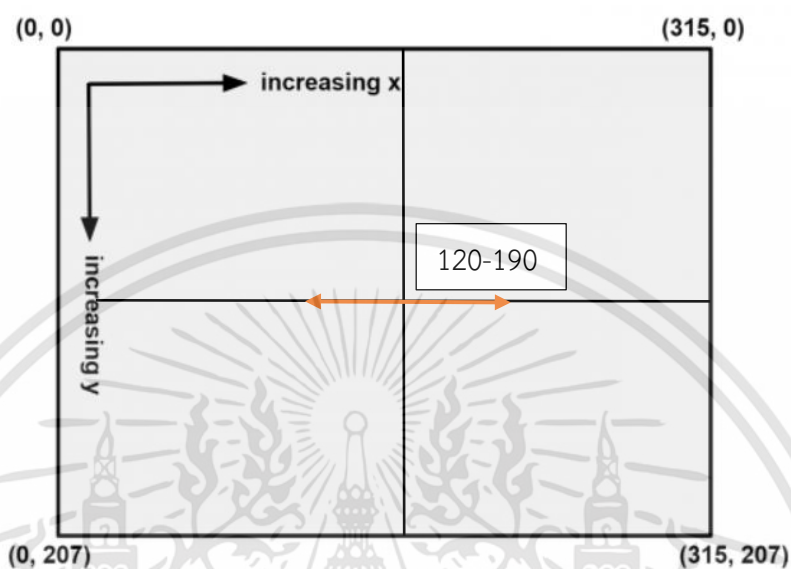
ตำแหน่งที่ 7 ให้การเคลื่อนที่สไลด์ไปทางขวา

ตำแหน่งที่ 8 ให้การเคลื่อนที่สไลด์ไปทางซ้าย

ในการทำงานของจอยสติ๊กจะมีโหมดการทำงาน 2 โหมด คือ Mode manual โดยจะทำการควบคุมทิศทางของล้อแม็กคาน้ำผ่านจอยสติ๊กได้ปกติ และ Mode auto จะไม่สามารถควบคุมผ่านจอยสติ๊กได้จะควบคุมผ่านการตรวจจับสปีดของตัวล้อแทน

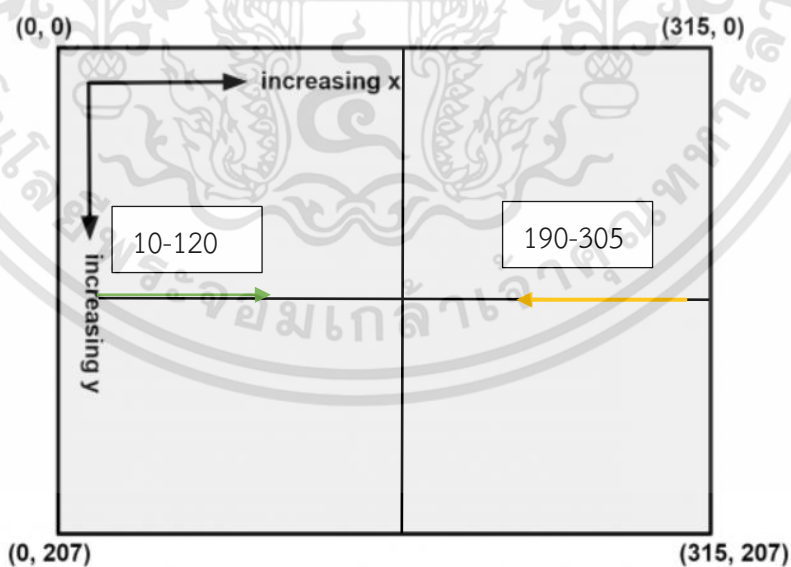
### 3.6 กำหนดช่วงแกน X ของกล้อง PIXY

1. กำหนดแกน X ระยะที่ 120 –190 พิกเซล ให้รถเคลื่อนที่ไปข้างหน้า



รูปที่ 3.6 ช่วงแกน X ที่รถจะทำการเคลื่อนที่ไปข้างหน้า

2. กำหนดแกน X ระยะที่ 10-120 พิกเซล ให้รถเคลื่อนที่ทางทิศตะวันออกเฉียงเหนือ, แกน X ระยะที่มากกว่า 190 พิกเซล ขึ้นไปให้รถเคลื่อนที่ทางทิศตะวันตกเฉียงเหนือ

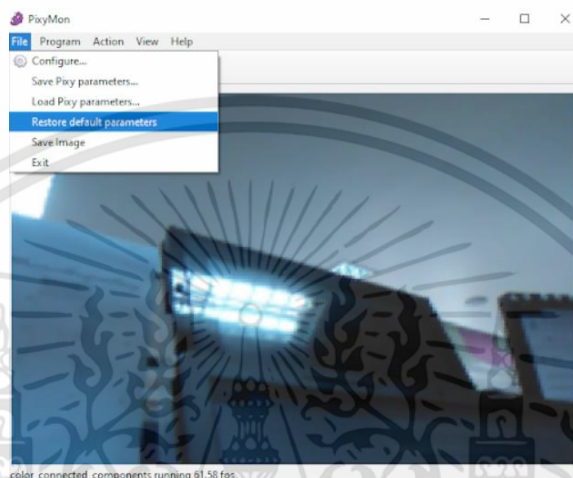


รูปที่ 3.7 ช่วงแกน X ที่รถจะทำการเคลื่อนที่ทางทิศตะวันออกเฉียงเหนือและทางทิศตะวันตกเฉียงเหนือ

### 3.7 กำหนดตัวแปรควบคุมล๊อแมกคานัมผ่านโปรแกรม PIXY MON

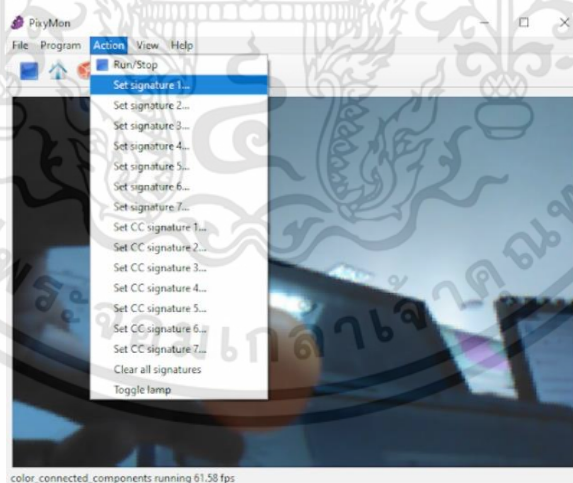
#### 3.7.1 ขั้นตอนการตั้งค่าการตรวจจับสี

1. การตั้งค่าล๊อ Pixy2 แบบ Single object tracking ก่อนอื่นทำการตั้งค่าเป็นค่าเริ่มต้นไปที่ File > Restore default parameter รูปที่ 3.8



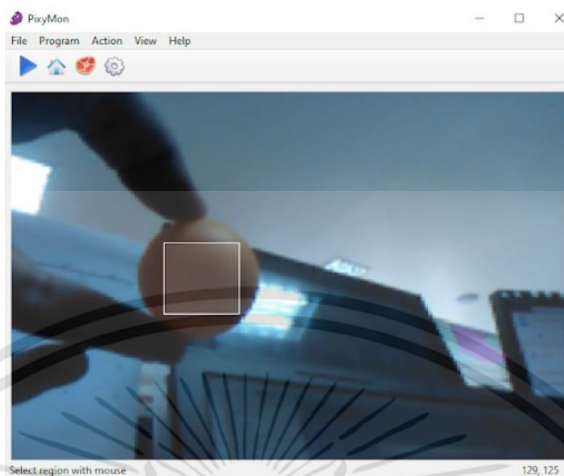
รูปที่ 3.8 เซ็ตค่าของล๊อเป็นค่าเริ่มต้น

2. จากนั้นให้ทำการเลือกสีของวัตถุที่ต้องการจะตรวจจับโดยไปที่เมนู Action > เลือก Set signature 1



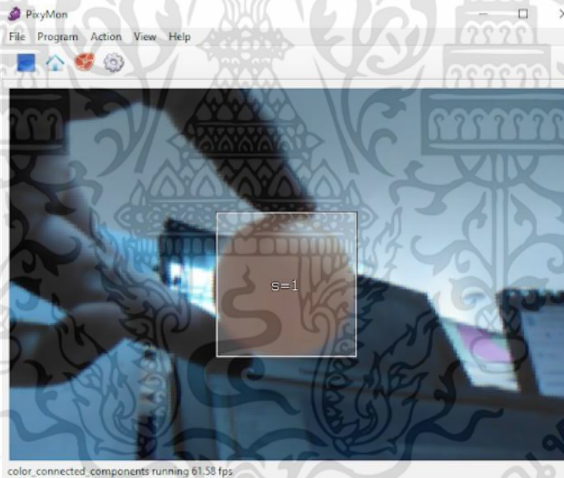
รูปที่ 3.9 เลือก Set signature 1

3. ทำการคลิกเมาส์ไปบริเวณของสีบนวัตถุที่จะทำการกำหนดค่า



รูปที่ 3.10 เลือกวัตถุที่จะทำการตรวจจับสนี

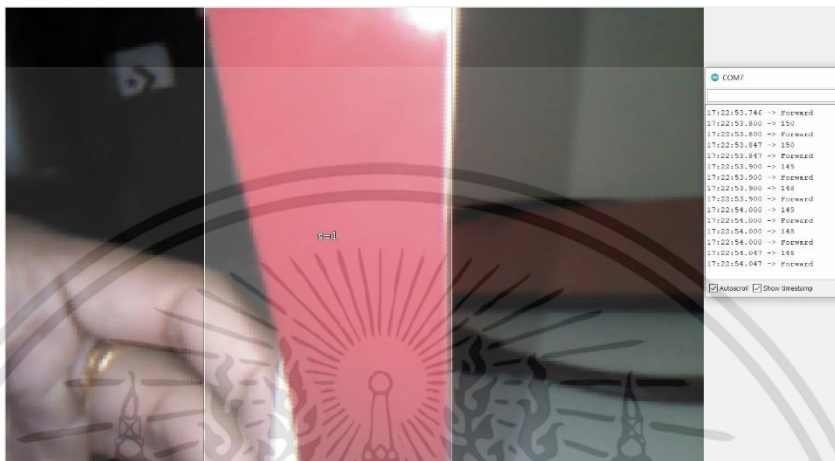
4. จะเห็นว่าเมื่อมีการตรวจจับสนีที่ตั้งค่าได้ จะแสดงให้เห็นการตรวจจับสนีวัตถุ



รูปที่ 3.11 การตรวจจับสนีวัตถุ

### 3.7.2 กำหนดตัวแปรควบคุมให้รถเคลื่อนที่เข้าหาแกน X

1. เมื่อค่าในแกน X ของกล้องอ่านค่าตัวแปร a มีค่ามากกว่า 120 และ a มีค่าน้อยกว่า 190 จะมีคำสั่งให้มอเตอร์ของล้อแมกคานัมเคลื่อนที่ไปข้างหน้า



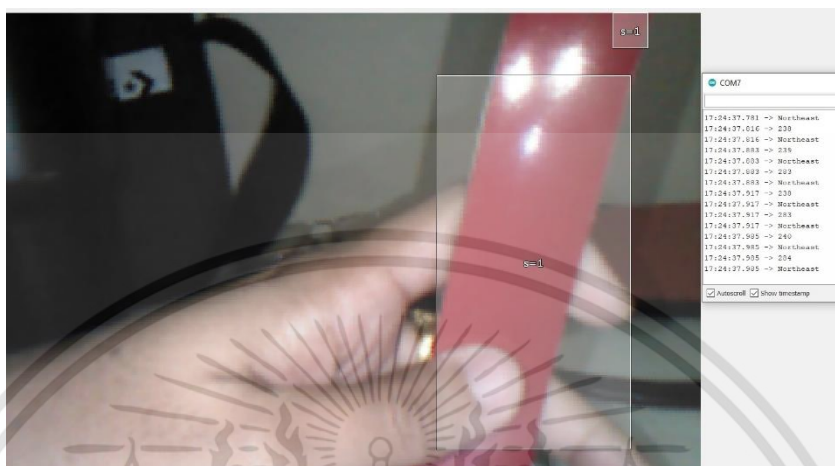
รูปที่ 3.12 แสดงค่าการกำหนดตัวแปรให้รถเคลื่อนที่ไปข้างหน้า

2. เมื่อค่าในแกน X ของกล้องอ่านค่าตัวแปร a มีค่าน้อยกว่า 120 และ a มีค่ามากกว่า 10 จะมีคำสั่งให้มอเตอร์ของล้อแมกคานัมเคลื่อนที่ไปยังทิศตะวันออกเฉียงเหนือ



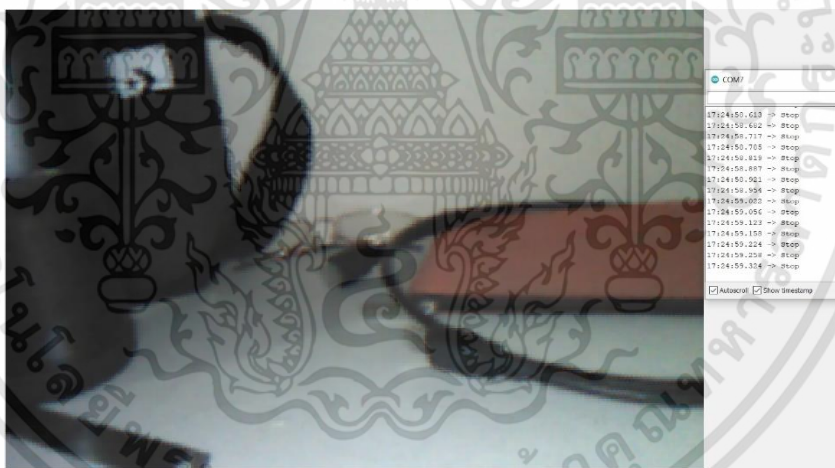
รูปที่ 3.13 แสดงค่าการกำหนดตัวแปรให้รถเคลื่อนที่ไปยังทิศตะวันออกเฉียงเหนือ

3. เมื่อค่าในแกน X ของกล้องอ่านค่าตัวแปร  $a$  มีค่ามากกว่า 190 จะมีคำสั่งให้มอเตอร์ของล้อแมกคานัมเคลื่อนที่ไปยังทิศตะวันตกเฉียงเหนือ



รูปที่ 3.14 แสดงค่าการกำหนดตัวแปรให้รถเคลื่อนที่ไปยังทิศตะวันตกเฉียงเหนือ

4. เมื่อไม่พบตัวแปรที่กำหนด จะมีคำสั่งให้มอเตอร์ของล้อแมกคานัมหยุดการเคลื่อนที่



รูปที่ 3.15 แสดงค่าการกำหนดตัวแปรให้รถหยุดการเคลื่อนที่

### 3.8 โค้ดควบคุมล้อแมกคานัม

```
#define in1_A 22
#define in1_B 28
#define in2_A 24
#define in2_B 26
#define pwm1 2
#define pwm2 3

#define in3_A 32
#define in3_B 30
#define in4_A 34
#define in4_B 36
#define pwm3 5
#define pwm4 4

int speed_ = 100; // กำหนดความเร็วของมอเตอร์

void setup()
{
  Serial.begin(115200); // ประกาศตัวแปรในการรับส่งข้อมูล

  pinMode(in1_A, OUTPUT);
  pinMode(in1_B, OUTPUT);
  pinMode(in2_A, OUTPUT);
  pinMode(in2_B, OUTPUT);
  pinMode(pwm1, OUTPUT);
  pinMode(pwm2, OUTPUT);

  pinMode(in3_A, OUTPUT);
  pinMode(in3_B, OUTPUT);
```

ประกาศขาตัวแปรของไดรฟ์ขับเคลื่อนมอเตอร์ชุดที่ 1

ประกาศขาตัวแปรของไดรฟ์ขับเคลื่อนมอเตอร์ชุดที่ 2

กำหนดหน้าที่ของขาสัญญาณให้เป็นเอาต์พุตของไดรฟ์ชุดที่ 1

กำหนดหน้าที่ของขาสัญญาณให้เป็นเอาต์พุตของไดรฟ์ชุดที่ 2

```

pinMode(in4_A, OUTPUT);
pinMode(in4_B, OUTPUT);
pinMode(pwm3, OUTPUT);
pinMode(pwm4, OUTPUT);
}

void loop()
{
  Forward(); //เดินหน้า
  delay(1000);

  Backward(); //ถอยหลัง
  delay(1000);

  SlideLeft(); //สไลด์ซ้าย
  delay(1000);

  SlideRight(); //สไลด์ขวา
  delay(1000);

  TurnLeft(); //เลี้ยวซ้าย
  delay(1000);

  TurnRight(); //เลี้ยวขวา
  delay(1000);
}

```

กำหนดหน้าที่ของขาสัญญาณให้เป็นเอาต์พุตของ  
ไทรฟชุดที่ 2

```
void Forward()
{
    digitalWrite(in1_A, HIGH);
    digitalWrite(in1_B, LOW);
    analogWrite(pwm1, speed_);
```

```
    digitalWrite(in2_A, HIGH);
    digitalWrite(in2_B, LOW);
    analogWrite(pwm2, speed_);
```

```
    digitalWrite(in4_A, HIGH);
    digitalWrite(in4_B, LOW);
    analogWrite(pwm4, speed_);
```

```
    digitalWrite(in3_A, HIGH);
    digitalWrite(in3_B, LOW);
    analogWrite(pwm3, speed_);
}
```

ส่วนของการป้อนค่าให้รถล้อแมกคาน้ำ  
เคลื่อนที่ไปข้างหน้า

```
void Backward()
{
    digitalWrite(in1_A, LOW);
    digitalWrite(in1_B, HIGH);
    analogWrite(pwm1, speed_);
```

```
    digitalWrite(in2_A, LOW);
    digitalWrite(in2_B, HIGH);
    analogWrite(pwm2, speed_);
```

```
    digitalWrite(in4_A, LOW);
```

ส่วนของการป้อนค่าให้รถล้อแมกคาน้ำ  
ถอยหลัง

```

digitalWrite(in4_B, HIGH);
analogWrite(pwm4, speed_);

digitalWrite(in3_A, LOW);
digitalWrite(in3_B, HIGH);
analogWrite(pwm3, speed_);
}

void SlideLeft()
{
  digitalWrite(in1_A, HIGH);
  digitalWrite(in1_B, LOW);
  analogWrite(pwm1, speed_);

  digitalWrite(in2_A, LOW);
  digitalWrite(in2_B, HIGH);
  analogWrite(pwm2, speed_);

  digitalWrite(in4_A, LOW);
  digitalWrite(in4_B, HIGH);
  analogWrite(pwm4, speed_);

  digitalWrite(in3_A, HIGH);
  digitalWrite(in3_B, LOW);
  analogWrite(pwm3, speed_);
}

```

ส่วนของการป้อนค่าให้รถล้อแมกคาน้ำ  
ถอยหลัง

ส่วนของการป้อนค่าให้รถล้อแมกคาน้ำ  
สไลด์ซ้าย

```
void SlideRight()
{
  digitalWrite(in1_A, LOW);
  digitalWrite(in1_B, HIGH);
  analogWrite(pwm1, speed_);
```

```
  digitalWrite(in2_A, HIGH);
  digitalWrite(in2_B, LOW);
  analogWrite(pwm2, speed_);
```

```
  digitalWrite(in4_A, HIGH);
  digitalWrite(in4_B, LOW);
  analogWrite(pwm4, speed_);
```

```
  digitalWrite(in3_A, LOW);
  digitalWrite(in3_B, HIGH);
  analogWrite(pwm3, speed_);
```

```
}
```

```
void TurnLeft()
```

```
{
```

```
  digitalWrite(in1_A, LOW);
  digitalWrite(in1_B, HIGH);
  analogWrite(pwm1, speed_);
```

```
  digitalWrite(in2_A, HIGH);
  digitalWrite(in2_B, LOW);
  analogWrite(pwm2, speed_);
```

```
  digitalWrite(in4_A, LOW);
```

ส่วนของการป้อนค่าให้รถล้อแมกคาน้ำ  
สไลด์ขวา

ส่วนของการป้อนค่าให้รถล้อแมกคาน้ำ  
เลี้ยวซ้าย

```

digitalWrite(in4_B, HIGH);
analogWrite(pwm4, speed_);

digitalWrite(in3_A, HIGH);
digitalWrite(in3_B, LOW);
analogWrite(pwm3, speed_);
}

void TurnRight()
{
    digitalWrite(in1_A, HIGH);
    digitalWrite(in1_B, LOW);
    analogWrite(pwm1, speed_);

    digitalWrite(in2_A, LOW);
    digitalWrite(in2_B, HIGH);
    analogWrite(pwm2, speed_);

    digitalWrite(in4_A, HIGH);
    digitalWrite(in4_B, LOW);
    analogWrite(pwm4, speed_);

    digitalWrite(in3_A, LOW);
    digitalWrite(in3_B, HIGH);
    analogWrite(pwm3, speed_);
}

```

ส่วนของการป้อนค่าให้รถล้อแมกคานัม  
เลี้ยวซ้าย

ส่วนของการป้อนค่าให้รถล้อแมกคานัม  
เลี้ยวขวา

### 3.9 โค้ดกล้อง PIXY2 รับค่าจากแกน X ส่งไปควบคุมยังบอร์ดไมโครคอนโทรลเลอร์

```
#include <Pixy2.h>

int a ;

// This is the main Pixy object
Pixy2 pixy;

void setup()
{
  Serial.begin(115200);
  Serial.print("Starting...\n");
  pixy.init();
}

void loop()
{
  pixy.setLamp(1, 1);
  int i;
  // grab blocks!
  pixy.ccc.getBlocks();
  // If there are detect blocks, print them!
  if (pixy.ccc.numBlocks)
  {
    // Serial.print("Detected ");
    // Serial.println(pixy.ccc.numBlocks);
    for (i=0; i<pixy.ccc.numBlocks; i++)
    {
      // Serial.print("  block ");
      // Serial.print(i);
      // Serial.print(": ");
      // pixy.ccc.blocks[i].print();
      a = pixy.ccc.blocks[i].m_x;
```

```

Serial.println(a);
if((a >120)&&(a<190))
{
    Forward();
    Serial.println("Forward");
}
else if((a <120)&&(a>10))
{
    // Northwest();//ตะวันตกเฉียงเหนือ
    Northeast();//ตะวันออกเฉียงใต้
    Serial.println("Northwest");
}
else if(a>190)
{
    // Northeast();//ตะวันออกเฉียงใต้
    Northwest();//ตะวันตกเฉียงเหนือ
    Serial.println("Northeast");
}
}
}
else {
    Stop();
    Serial.println("Stop");
}
}
}

```

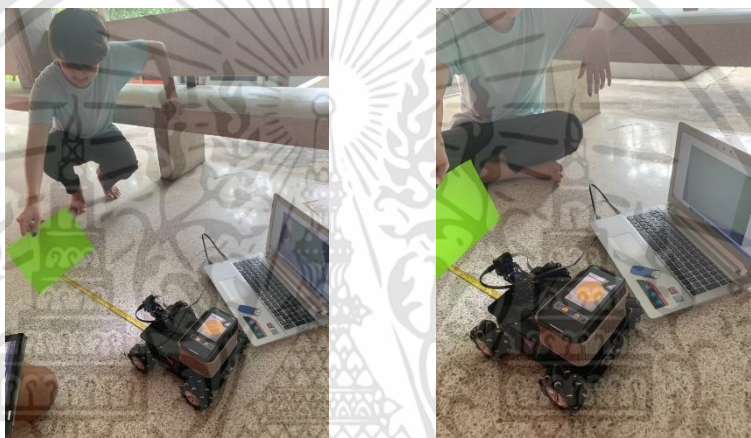
## บทที่ 4

### ผลการทดลอง

#### 4.1 การทดลองระยะการตรวจจับสีของกล้องกับแสงสว่างแนวตั้งฉากกับพื้น

##### 4.1.1 การทดลองวัดปริมาณของแสงกับวัตถุขนาด 26x19 เซนติเมตร แนวตั้งฉากกับพื้น

โดยการนำกล้อง PIXY2 ตั้งไว้อยู่กับที่แล้วนำวัตถุที่เราเตรียมไว้มาถูกระยะทางที่กล้อง PIXY2 จะทำการจับได้ในขณะที่มีแสงที่แตกต่างกัน แสดงดังรูปภาพที่ 4.1



รูปที่ 4.1 วัดปริมาณของแสงกับวัตถุแนวตั้งฉากกับพื้นขนาด 26x19 เซนติเมตร

##### ตารางที่ 4.1.1 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 26x19 เซนติเมตร

| ระยะทาง<br>(เซนติเมตร) | ความสว่างของแสง (Lux) |     |     |     |     |
|------------------------|-----------------------|-----|-----|-----|-----|
|                        | 1                     | 136 | 173 | 280 | 310 |
| 10                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 20                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 40                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 80                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 120                    | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 140                    | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 180                    | ✗                     | ✓   | ✓   | ✓   | ✓   |

หมายเหตุ : 1. ✓ แทนการตรวจจับสีของวัตถุได้

2. ✗ แทนการตรวจจับสีของวัตถุไม่พบ

#### 4.1.2 การทดลองวัดปริมาณของแสงกับวัตถุขนาด 13x19 เซนติเมตร แนวตั้งฉากกับพื้น

โดยการนำกล้อง PIXY2 ตั้งไว้อยู่กับที่แล้วนำวัตถุที่เราเตรียมไว้มาดูระยะทางที่กล้อง PIXY2 จะทำการจับได้ในขณะที่มีแสงที่ต่างกัน แสดงดังรูปภาพที่ 4.2



รูปที่ 4.2 วัดปริมาณของแสงกับวัตถุแนวตั้งฉากกับพื้นขนาด 13x19 เซนติเมตร

ตารางที่ 4.1.2 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x19 เซนติเมตร

| ระยะทาง<br>(เซนติเมตร) | ความสว่างของแสง (Lux) |     |     |     |     |
|------------------------|-----------------------|-----|-----|-----|-----|
|                        | 1                     | 136 | 173 | 280 | 310 |
| 10                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 20                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 40                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 80                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 120                    | ✗                     | ✗   | ✓   | ✓   | ✓   |
| 140                    | ✗                     | ✗   | ✗   | ✓   | ✓   |
| 180                    | ✗                     | ✗   | ✗   | ✗   | ✗   |

หมายเหตุ : 1. ✓ แทนการตรวจจับสีของวัตถุได้  
2. ✗ แทนการตรวจจับสีของวัตถุไม่พบ

### 4.1.3 การทดลองวัดปริมาณของแสงกับวัตถุขนาด 13x10 เซนติเมตร แนวตั้งฉากกับพื้น

โดยการนำกล้อง PIXY2 ตั้งไว้อยู่กับที่แล้วนำวัตถุที่เราเตรียมไว้มาดูระยะทางที่กล้อง PIXY2 จะทำการจับได้ในขณะที่มีแสงที่ต่างกัน แสดงดังรูปภาพที่ 4.3



รูปที่ 4.3 วัดปริมาณของแสงกับวัตถุแนวตั้งฉากกับพื้นขนาด 13x10 เซนติเมตร

ตารางที่ 4.1.3 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x10 เซนติเมตร

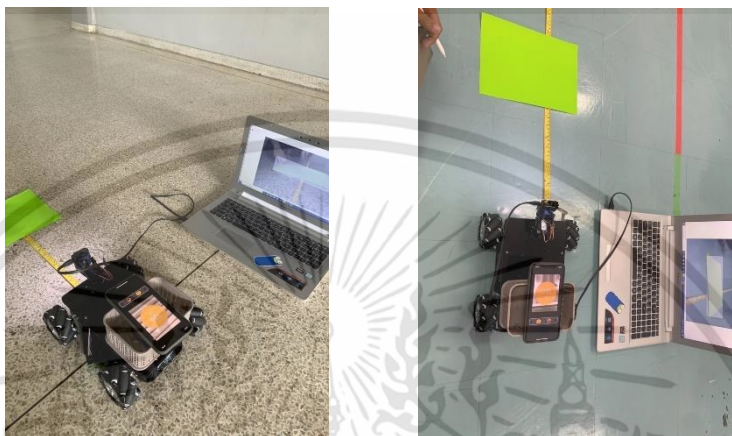
| ระยะทาง<br>(เซนติเมตร) | ความสว่างของแสง (Lux) |     |     |     |     |
|------------------------|-----------------------|-----|-----|-----|-----|
|                        | 1                     | 136 | 173 | 280 | 310 |
| 10                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 20                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 40                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 80                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 120                    | ✗                     | ✗   | ✗   | ✓   | ✓   |
| 140                    | ✗                     | ✗   | ✗   | ✗   | ✓   |
| 180                    | ✗                     | ✗   | ✗   | ✗   | ✗   |

หมายเหตุ : 1. ✓ แทนการตรวจจับสีของวัตถุได้  
2. ✗ แทนการตรวจจับสีของวัตถุไม่พบ

## 4.2 การทดลองระยะการตรวจจับสีของกล้องกับแสงสว่างแนวราบกับพื้น

### 4.2.1 การทดลองวัดปริมาณของแสงกับวัตถุขนาด 26x19 เซนติเมตร แนวราบกับพื้น

โดยการนำกล้อง PIXY2 ตั้งไว้อยู่กับที่แล้วนำวัตถุที่เราเตรียมไว้มาดูระยะทางที่กล้อง PIXY2 จะทำการจับได้ในขณะที่มีแสงที่ต่างกัน แสดงดังรูปภาพที่ 4.4



รูปที่ 4.4 วัดปริมาณของแสงกับวัตถุแนวราบกับพื้นขนาด 26x19 เซนติเมตร

ตารางที่ 4.2.1 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 26x19 เซนติเมตร

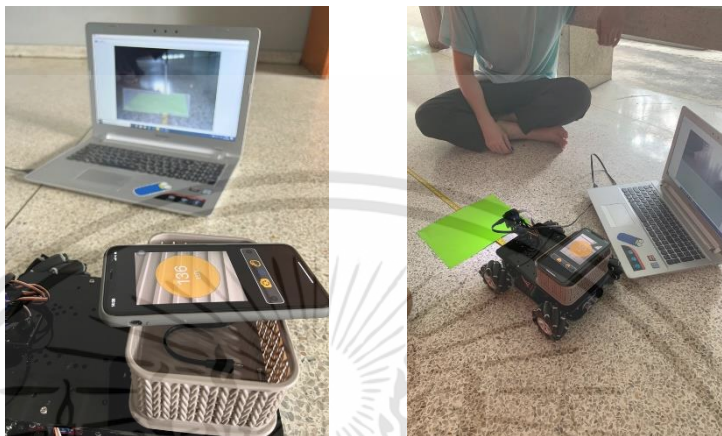
| ระยะทาง<br>(เซนติเมตร) | ความสว่างของแสง (Lux) |     |     |     |     |
|------------------------|-----------------------|-----|-----|-----|-----|
|                        | 1                     | 136 | 173 | 280 | 310 |
| 10                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 20                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 30                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 40                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 50                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 60                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 70                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 80                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 90                     | ✗                     | ✗   | ✗   | ✓   | ✓   |
| 100                    | ✗                     | ✗   | ✗   | ✓   | ✓   |

หมายเหตุ : 1. ✓ แทนการตรวจจับสีของวัตถุได้

2. ✗ แทนการตรวจจับสีของวัตถุไม่พบ

#### 4.2.2 การทดลองวัดปริมาณของแสงกับวัตถุขนาด 13x19 เซนติเมตร แนวราบกับพื้น

โดยการนำกล้อง PIXY2 ตั้งไว้อยู่กับที่แล้วนำวัตถุที่เราเตรียมไว้มาดูระยะทางที่กล้อง PIXY2 จะทำการจับได้ในขณะที่มีแสงที่แตกต่างกัน แสดงดังรูปภาพที่ 4.5



รูปที่ 4.5 วัดปริมาณของแสงกับวัตถุแนวราบกับพื้นขนาด 13x19 เซนติเมตร

ตารางที่ 4.2.1 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x19 เซนติเมตร

| ระยะทาง<br>(เซนติเมตร) | ความสว่างของแสง (Lux) |     |     |     |     |
|------------------------|-----------------------|-----|-----|-----|-----|
|                        | 1                     | 136 | 173 | 280 | 310 |
| 10                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 20                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 30                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 40                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 50                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 60                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 70                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 80                     | ✗                     | ✗   | ✓   | ✓   | ✓   |
| 90                     | ✗                     | ✗   | ✗   | ✓   | ✓   |
| 100                    | ✗                     | ✗   | ✗   | ✓   | ✓   |

หมายเหตุ : 1. ✓ แทนการตรวจจับสีของวัตถุได้

2. ✗ แทนการตรวจจับสีของวัตถุไม่พบ

#### 4.2.3 การทดลองวัดปริมาณของแสงกับวัตถุขนาด 13x10 เซนติเมตร แนวราบกับพื้น

โดยการนำกล้อง PIXY2 ตั้งไว้อยู่กับที่แล้วนำวัตถุที่เราเตรียมไว้มาดูระยะทางที่กล้อง PIXY2 จะทำการจับได้ในขณะที่มีแสงที่แตกต่างกัน แสดงดังรูปภาพที่ 4.5



รูปที่ 4.6 วัดปริมาณของแสงกับวัตถุแนวราบกับพื้นขนาด 13x10 เซนติเมตร

ตารางที่ 4.2.1 ความสว่างในการทดลองกับการตรวจจับสีของวัตถุขนาด 13x10 เซนติเมตร

| ระยะทาง<br>(เซนติเมตร) | ความสว่างของแสง (Lux) |     |     |     |     |
|------------------------|-----------------------|-----|-----|-----|-----|
|                        | 1                     | 136 | 173 | 280 | 310 |
| 10                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 20                     | ✓                     | ✓   | ✓   | ✓   | ✓   |
| 30                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 40                     | ✗                     | ✓   | ✓   | ✓   | ✓   |
| 50                     | ✗                     | ✗   | ✓   | ✓   | ✓   |
| 60                     | ✗                     | ✗   | ✗   | ✓   | ✓   |
| 70                     | ✗                     | ✗   | ✗   | ✓   | ✓   |
| 80                     | ✗                     | ✗   | ✗   | ✗   | ✓   |
| 90                     | ✗                     | ✗   | ✗   | ✗   | ✓   |
| 100                    | ✗                     | ✗   | ✗   | ✗   | ✗   |

- หมายเหตุ : 1. ✓ แทนการตรวจจับสีของวัตถุได้  
2. ✗ แทนการตรวจจับสีของวัตถุไม่พบ

### 4.3 การทดสอบแรงดันไฟฟ้าที่ทำการจ่ายให้ล้อแมกคานัมโดยการควบคุม PWM มอเตอร์

ในการทดลองนี้เป็นการทดสอบแรงดันไฟฟ้าที่ทำการจ่ายให้มอเตอร์โดยการควบคุมแบบ PWM โดยจะมี 3 ระดับ คือ 100 120 150 โดยการควบคุมทั้ง 3 ระดับจะมีการจ่ายแรงดันทั้งหมด 3 ระดับ โดยจะทำการวัดทุกความเร็วที่ทำการควบคุมโดยทำการวัด 3 ครั้ง

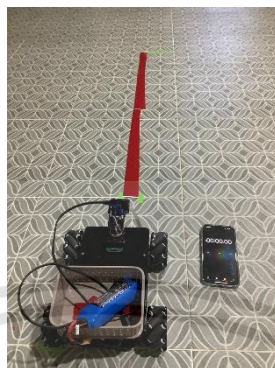


รูปที่ 4.7 วัดแรงดันไฟฟ้าทั้ง 3 ความเร็วของมอเตอร์

ตารางที่ 4.3.1 การทดสอบวัดแรงดันไฟฟ้าที่ความเร็วในการควบคุมมอเตอร์ต่างกัน

| ความเร็วของรถ<br>(PWM) | แรงดันไฟฟ้า(Volt) |            |            | ค่าเฉลี่ย<br>(Volt) |
|------------------------|-------------------|------------|------------|---------------------|
|                        | ครั้งที่ 1        | ครั้งที่ 2 | ครั้งที่ 3 |                     |
| 100                    | 4.68              | 4.71       | 4.69       | 4.69                |
| 120                    | 5.74              | 5.69       | 5.72       | 5.72                |
| 150                    | 7.05              | 7.01       | 7.05       | 7.04                |

#### 4.4 การทดสอบหาความเร็วในการเคลื่อนที่ของรถ



รูปที่ 4.8 การทดสอบหาความเร็วในการเคลื่อนที่ของรถ

ในการทดลองระบบนำทางยานขนส่งอัจฉริยะได้ทำการแบ่งความถี่ฐานออกเป็น 3 ระดับโดยทำการควบคุม PWM มอเตอร์ คือ 100 120 150 ซึ่งความถี่ที่กำหนดมานั้นจะอ้างอิงจากการวิ่งของการที่ดีที่สุดโดยควบคุมผ่านจอยสติ๊กโดยใช้ระยะทางในการทดสอบ 1 เมตร

ตารางที่ 4.4.1 การทดสอบหาความเร็วในการเคลื่อนที่ของรถ

| จำนวนพัลส์<br>(PWM) | เวลา(s)    |            |            | ค่าเฉลี่ย<br>(s) |
|---------------------|------------|------------|------------|------------------|
|                     | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 3 |                  |
| 100                 | 2.70       | 2.60       | 2.23       | 2.51             |
| 120                 | 2.16       | 1.98       | 1.99       | 2.04             |
| 150                 | 1.73       | 1.76       | 1.68       | 1.72             |

การคำนวณหาความเร็วของรถ

$$v = \frac{S}{t}$$

$V$  = ความเร็ว หน่วย เมตร/วินาที (m/s)

$S$  = การกระจัด หน่วย เมตร (m)

$t$  = เวลา หน่วย วินาที (s)

- ความเร็วในการควบคุมมอเตอร์ 100 PWM จะได้ค่าอัตราเร็ว คือ 0.40 m/s
- ความเร็วในการควบคุมมอเตอร์ 120 PWM จะได้ค่าอัตราเร็ว คือ 0.49 m/s
- ความเร็วในการควบคุมมอเตอร์ 150 PWM จะได้ค่าอัตราเร็ว คือ 0.58 m/s

#### 4.5 การทดลองความเร็วในการเคลื่อนที่ตามสีแนวแกน X ของกล้อง PIXY

ระบบนำทางอัจฉริยะจะแบ่งความเร็วจะแบ่งออกเป็น 3 ระดับ คือ 0.40 เมตร/วินาที 0.50 เมตร/วินาที 0.58 เมตร/วินาที โดยจะมีแนวเส้นทางในการวิ่ง 4 แนวเส้น คือ แนวเส้นตรง แนวเส้นโค้งไปทางซ้าย แนวเส้นโค้งไปทางขวา และทั้งสามแนวเส้นทำการวิ่งร่วมกัน



รูปที่ 4.9 เส้นทางตรงที่ใช้ในการทดลอง



รูปที่ 4.10 เส้นทางโค้งไปทางซ้ายที่ใช้ในการทดลอง



รูปที่ 4.11 เส้นทางโค้งไปทางขวาที่ใช้ในการทดลอง

```
int speed_1 = 100;
int speed_2 = 100;
int speed_3 = 100;
int speed_4 = 100;
```

รูปที่ 4.12 ความเร็วในการควบคุมมอเตอร์ของรถระดับที่ 1

```
int speed_1 = 120;
int speed_2 = 120;
int speed_3 = 120;
int speed_4 = 120;
```

รูปที่ 4.13 ความเร็วในการควบคุมมอเตอร์ของรถระดับที่ 2

```
int speed_1 = 150;
int speed_2 = 150;
int speed_3 = 150;
int speed_4 = 150;
```

รูปที่ 4.14 ความเร็วในการควบคุมมอเตอร์ของรถระดับที่ 3

#### 4.5.1 การทดลองวิ่งแนวเส้นตรง

ตารางที่ 4.5.1 การทดลองวิ่งแนวเส้นตรง

| ความเร็วของรถ<br>(m/s) | จำนวนครั้งที่ทำการทดลอง |   |   |   |   |   |
|------------------------|-------------------------|---|---|---|---|---|
|                        | 1                       | 2 | 3 | 4 | 5 | 6 |
| 0.40                   | ✓                       | ✓ | ✓ | ✓ | ✓ | ✓ |
| 0.48                   | ✓                       | ✓ | ✓ | ✓ | ✓ | ✗ |
| 0.58                   | ✓                       | ✓ | ✗ | ✗ | ✗ | ✗ |

หมายเหตุ : 1. ✓ แทนการที่รถวิ่งไปถึงจุดหมาย

2. ✗ แทนการที่รถวิ่งไม่ถึงจุดหมาย

$$\%error = \frac{\text{Exact value} - \text{Approximate value}}{\text{Exact value}} \times 100$$

- ความเร็วของรถที่ 0.40 m/s มีค่าความผิดพลาดเท่ากับ 0 %
- ความเร็วของรถที่ 0.48 m/s มีค่าความผิดพลาดเท่ากับ 16.67 %
- ความเร็วของรถที่ 0.58 m/s มีค่าความผิดพลาดเท่ากับ 66.67 %

#### 4.5.2 การทดลองวิ่งแนวโค้งไปทางซ้าย

ตารางที่ 4.5.2 การทดลองวิ่งแนวโค้งไปทางซ้าย

| ความเร็วของรถ<br>(m/s) | จำนวนครั้งที่ทำการทดลอง |   |   |   |   |   |
|------------------------|-------------------------|---|---|---|---|---|
|                        | 1                       | 2 | 3 | 4 | 5 | 6 |
| 0.40                   | ✓                       | ✓ | ✓ | ✓ | ✓ | ✓ |
| 0.48                   | ✓                       | ✗ | ✓ | ✗ | ✓ | ✗ |
| 0.58                   | ✓                       | ✗ | ✗ | ✗ | ✗ | ✗ |

หมายเหตุ : 1. ✓ แทนการที่รถวิ่งไปถึงจุดหมาย

2. ✗ แทนการที่รถวิ่งไม่ถึงจุดหมาย

$$\%error = \frac{\text{Exact value} - \text{Approximate value}}{\text{Exact value}} \times 100$$

- ความเร็วของรถที่ 0.40 m/s มีค่าความผิดพลาดเท่ากับ 0 %
- ความเร็วของรถที่ 0.48 m/s มีค่าความผิดพลาดเท่ากับ 66.67 %
- ความเร็วของรถที่ 0.58 m/s มีค่าความผิดพลาดเท่ากับ 83.34 %

### 4.5.3 การทดลองวิ่งแนวโค้งไปทางขวา

ตารางที่ 4.5.3 การทดลองวิ่งแนวโค้งไปทางขวา

| ความเร็วของรถ<br>(m/s) | จำนวนครั้งที่ทำการทดลอง |   |   |   |   |   |
|------------------------|-------------------------|---|---|---|---|---|
|                        | 1                       | 2 | 3 | 4 | 5 | 6 |
| 0.40                   | ✓                       | ✓ | ✓ | ✗ | ✓ | ✓ |
| 0.48                   | ✗                       | ✗ | ✓ | ✓ | ✗ | ✗ |
| 0.58                   | ✗                       | ✗ | ✗ | ✓ | ✗ | ✗ |

หมายเหตุ : 1. ✓ แทนการที่รถวิ่งไปถึงจุดหมาย

2. ✗ แทนการที่รถวิ่งไม่ถึงจุดหมาย

$$\%error = \frac{\text{Exact value} - \text{Approximate value}}{\text{Exact value}} \times 100$$

- ความเร็วของรถที่ 0.40 m/s มีค่าความผิดพลาดเท่ากับ 16.67 %
- ความเร็วของรถที่ 0.48 m/s มีค่าความผิดพลาดเท่ากับ 66.67 %
- ความเร็วของรถที่ 0.58 m/s มีค่าความผิดพลาดเท่ากับ 83.34 %

#### 4.5.4 การทดลองวิ่งทางตรงแนวทางโค้งไปทางซ้ายและแนวโค้งไปทางขวา

ตารางที่ 4.5.4 การทดลองวิ่งทางตรงแนวทางโค้งไปทางซ้ายและแนวโค้งไปทางขวา

| ความเร็วของรถ<br>(m/s) | จำนวนครั้งที่ทำการทดลอง |   |   |   |   |   |
|------------------------|-------------------------|---|---|---|---|---|
|                        | 1                       | 2 | 3 | 4 | 5 | 6 |
| 0.40                   | ✓                       | ✗ | ✓ | ✗ | ✗ | ✓ |
| 0.48                   | ✗                       | ✗ | ✗ | ✓ | ✗ | ✗ |
| 0.58                   | ✗                       | ✗ | ✗ | ✗ | ✗ | ✗ |

หมายเหตุ : 1. ✓ แทนการที่รถวิ่งไปถึงจุดหมาย

2. ✗ แทนการที่รถวิ่งไม่ถึงจุดหมาย

$$\%error = \frac{\text{Exact value} - \text{Approximate value}}{\text{Exact value}} \times 100$$

- ความเร็วของรถที่ 0.48 m/s มีค่าความผิดพลาดเท่ากับ 50 %
- ความเร็วของรถที่ 0.48 m/s มีค่าความผิดพลาดเท่ากับ 83.34 %
- ความเร็วของรถที่ 0.58 m/s มีค่าความผิดพลาดเท่ากับ 100 %

## บทที่ 5

# สรุปผลการทดลอง

### 5.1 สรุปผลการทดลอง

จากการทดลองการทดลองระยะการตรวจจับสีของกล้องกับแสงสว่างพบว่าแสงที่ใช้สำหรับการตรวจจับนั้นจะต้องใช้แสงไฟส่องสว่างจากหลอดไฟภายในห้อง เนื่องจากถ้าใช้แสงจากสภาพแวดล้อมหรือแสงแดดจะมีการเปลี่ยนแปลงโทนสีโดยไม่ตรงกับค่าที่ตั้งไว้ในโปรแกรม PIXY MON ทำให้การควบคุมรถในโหมดอัตโนมัตินั้นมีการเกิดการรบกวนของค่าสีที่เปลี่ยนแปลงเมื่อเจอแสงแดดที่ทำการส่องมายังตัวกล้องได้

สำหรับการเคลื่อนที่และระบบควบคุมโหมดอัตโนมัติเป็นการใช้ค่าแกน X จากกล้อง PIXY2 ควบคุมทิศทางการเคลื่อนที่ของรถให้ไปถึงจุดหมายด้วยความเร็ว 3 ระดับจะมีค่าเท่ากับ 100 120 150 PWM ความเร็วที่เหมาะสมสำหรับการวิ่งของรถที่มีค่าความผิดพลาดน้อยที่สุด คือ 100 PWM รถต้องการความเร็วในการเคลื่อนที่ที่ต่ำเนื่องจากความผิดพลาดที่กล้องนั้นมีค่าความผิดพลาดที่สูงในการจะนำที่นำมาใช้ในระบบนำทางเมื่อรถทำการเคลื่อนที่แกน X จะเป็นตัวควบคุมทิศทางเมื่อมีความเร็วที่สูงรถก็จะมีการหลุดออกจากเส้นทางที่ทำการตั้งเป้าหมายไว้เนื่องจากการรับค่าแกน X ส่งไปประมวลผลยังบอร์ดไมโครคอนโทรลเลอร์ไม่ได้

### 5.2 ปัญหาที่พบขณะทำการทดลอง

1. พอร์ตของบอร์ด Arduino Uno ที่ต่อเข้ากับบอร์ด HR8833 นั้นพอร์ตไม่เพียงพอต่อการประยุกต์ใช้งานต่อกับอุปกรณ์อื่นได้ เนื่องจากเป็นบอร์ดควบคุมมอเตอร์เพียงอย่างเดียว
2. มีแสงแดดแทรกซ้อนระหว่างการทดลองทำให้บางที่รถเกิดการร่นไปทำงานที่คำสั่งอื่น
3. สายไฟหลวมเกิดการช็อตกันภายในวงจรทำให้ตัวส่งสัญญาณของ Joystick PS2 เกิดความเสียหายควบคุมรถไม่ได้
4. ใช้ความเร็วของรถที่สูงทำให้รถมีการไปไม่ถึงยังเป้าหมายที่กำหนด

### 5.3 วิธีการแก้ไขปัญหา

1. เปลี่ยนใช้บอร์ด Arduino Mega2560 ร่วมกับไดร์ฟขับเคลื่อนมอเตอร์ L298N เพื่อให้พืดเพิ่มขึ้นสามารถประยุกต์ใช้งานต่อร่วมกับอุปกรณ์เพิ่มเติมอื่น ๆ ได้

2. ใช้แสงสว่างภายในห้องเพื่อไม่ให้เกิดการแทรกซ้อนของแสงแดดไปรบกวนเปลี่ยนสีของระบบคำสั่งการเคลื่อนที่ของรถ
3. ทำการสั่งซื้อตัวส่งสัญญาณของ Joystick PS2 ใหม่ ตรวจสอบสายไฟทั้งระบบใหม่เพื่อไม่ให้เกิดข้อผิดพลาดได้
4. ทำการปรับลดความเร็วของรถให้อยู่ในความเร็วที่ต่ำและเหมาะสมกับพื้นที่ที่ทำการทดลอง

#### 5.4 ข้อจำกัดการใช้งาน

1. รถไม่สามารถใช้งานในพื้นที่ผิวขรุขระได้ เนื่องจากล้อไม่ได้ออกแบบมาให้ใช้งานกับพื้นที่ผิวขรุขระ
2. รถไม่สามารถออกวิ่งตามพื้นที่สภาพแวดล้อมที่มีแสงแดดได้มาก เนื่องจากถ้าตัวกล้องรับแสงแดดมากอาจมีการเปลี่ยนสีของวัตถุที่รับค่าจากกล้องทำให้ไม่สามารถส่งคำสั่งไปยังระบบคอนโทรลเลอร์ได้

#### 5.5 ข้อเสนอแนะเพิ่มเติม

สำหรับการพัฒนาต่อในอนาคต รถนี้สามารถเพิ่มเติมความสามารถได้โดยอาจจะเปลี่ยนโมดูลกล้องที่ใช้ในการรับค่าคอนโทรลรถให้มีประสิทธิภาพมากกว่าตัวกล้องที่ใช้อยู่ เพิ่มเติมระบบการหลบหลีกสิ่งกีดขวางให้มีการควบคุมที่ดีขึ้นไปในอนาคต

## บรรณานุกรม

ThaiEasyElec enable your design. บทความอาณูโนคืออะไร,  
<https://blog.thaieasyelec.com/what-is-arduino-ch1/>

PIXY. PIXY2, <https://pixycam.com/pixy2/>

TAWUN ALL ELECTRONIC. บทความการใช้งาน การใช้งาน L298N motor drive,  
<http://gamerbloggerport.blogspot.com/2017/09/l298n-motor-drive.html>

Innovative Experiment Co.,Ltd. Wireless Joystick kit,  
<https://inex.co.th/store/manual/WirelessPSX140913.pdf>

สรายุทธ นาขมิ้น,ณิชนันท์ ขาวมงคล. ระบบขับเคลื่อนล้อแม่กลอน,  
[http://digital\\_collect.lib.buu.ac.th/project/b00254411.pdf](http://digital_collect.lib.buu.ac.th/project/b00254411.pdf)

บริษัท แคลิเบรชั่น แลบบอราทอรี จำกัด. เครื่องมือวัดแสง, <https://www.cal-laboratory.com/>

บริษัท สุพรีมไลน์ จำกัด. อัลตราโซนิกเซนเซอร์, <https://www.supremelines.co.th/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <Pixy2.h>

int a ;

// This is the main Pixy object
Pixy2 pixy;

/////MOTOR/////

#define in1_A 22
#define in1_B 28
#define in2_A 24
#define in2_B 26
#define pwm1 2
#define pwm2 3

#define in3_A 32
#define in3_B 30
#define in4_A 34
#define in4_B 36
#define pwm3 5
#define pwm4 4

int speed_1 = 100;
int speed_2 = 100;
int speed_3 = 100;
int speed_4 = 100;

void setup()
{
    Serial.begin(115200);
    Serial.print("Starting...\n");
    ///////////////////////////////////
    pinMode(in1_A, OUTPUT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pinMode(in1_B, OUTPUT);
pinMode(in2_A, OUTPUT);
pinMode(in2_B, OUTPUT);
pinMode(pwm1, OUTPUT);
pinMode(pwm2, OUTPUT);
```

```
pinMode(in3_A, OUTPUT);
pinMode(in3_B, OUTPUT);
pinMode(in4_A, OUTPUT);
pinMode(in4_B, OUTPUT);
pinMode(pwm3, OUTPUT);
pinMode(pwm4, OUTPUT);
pixy.init();
}

void loop()
{
    pixy.setLamp(1, 1);
    int i;
    // grab blocks!
    pixy.ccc.getBlocks();

    // If there are detect blocks, print them!
    if (pixy.ccc.numBlocks)
    {
        // Serial.print("Detected ");
        // Serial.println(pixy.ccc.numBlocks);
        for (i=0; i<pixy.ccc.numBlocks; i++)
        {
            // Serial.print("  block ");
```

```

// Serial.print(i);
// Serial.print(": ");
// pixy.ccc.blocks[i].print();
a = pixy.ccc.blocks[i].m_x;
Serial.println(a);
if((a >120)&&(a<190))
{
    Forward();
    Serial.println("Forward");
}
else if((a <120)&&(a>10))
{
    // Northwest();//ตะวันตกเฉียงเหนือ
    Northeast();//ตะวันออกเฉียงเหนือ
    Serial.println("Northwest");
}
else if(a>190)
{
    // Northeast();//ตะวันออกเฉียงใต้
    Northwest();//ตะวันตกเฉียงเหนือ
    Serial.println("Northeast");
}
}

}

else {
    Stop();
    Serial.println("Stop");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

void Forward()
{
    digitalWrite(in1_A, HIGH);
    digitalWrite(in1_B, LOW);
    analogWrite(pwm1, speed_1);

    digitalWrite(in2_A, HIGH);
    digitalWrite(in2_B, LOW);
    analogWrite(pwm2, speed_2);

    digitalWrite(in3_A, HIGH);
    digitalWrite(in3_B, LOW);
    analogWrite(pwm3, speed_3);

    digitalWrite(in4_A, HIGH);
    digitalWrite(in4_B, LOW);
    analogWrite(pwm4, speed_4);
}

void Backward()
{
    digitalWrite(in1_A, LOW);
    digitalWrite(in1_B, HIGH);
    analogWrite(pwm1, speed_1);

    digitalWrite(in2_A, LOW);
    digitalWrite(in2_B, HIGH);

```

```

analogWrite(pwm2, speed_2);

digitalWrite(in3_A, LOW);
digitalWrite(in3_B, HIGH);
analogWrite(pwm3, speed_3);

digitalWrite(in4_A, LOW);
digitalWrite(in4_B, HIGH);
analogWrite(pwm4, speed_4);
}

void SlideRight()
{
  digitalWrite(in1_A, HIGH);
  digitalWrite(in1_B, LOW);
  analogWrite(pwm1, speed_1);

  digitalWrite(in2_A, LOW);
  digitalWrite(in2_B, HIGH);
  analogWrite(pwm2, speed_2);

  digitalWrite(in3_A, LOW);
  digitalWrite(in3_B, HIGH);
  analogWrite(pwm3, speed_3);

  digitalWrite(in4_A, HIGH);
  digitalWrite(in4_B, LOW);
  analogWrite(pwm4, speed_4);
}

```

```

void SlideLeft()
{
    digitalWrite(in1_A, LOW);
    digitalWrite(in1_B, HIGH);
    analogWrite(pwm1, speed_1);

    digitalWrite(in2_A, HIGH);
    digitalWrite(in2_B, LOW);
    analogWrite(pwm2, speed_2);

    digitalWrite(in3_A, HIGH);
    digitalWrite(in3_B, LOW);
    analogWrite(pwm3, speed_3);

    digitalWrite(in4_A, LOW);
    digitalWrite(in4_B, HIGH);
    analogWrite(pwm4, speed_4);
}

void TurnRight()
{
    digitalWrite(in1_A, LOW);
    digitalWrite(in1_B, HIGH);
    analogWrite(pwm1, speed_1);

    digitalWrite(in2_A, HIGH);
    digitalWrite(in2_B, LOW);
    analogWrite(pwm2, speed_2);

    digitalWrite(in3_A, LOW);

```

```
digitalWrite(in3_B, HIGH);
analogWrite(pwm3, speed_3);
```

```
digitalWrite(in4_A, HIGH);
digitalWrite(in4_B, LOW);
analogWrite(pwm4, speed_4);
}
```

```
void TurnLeft()
{
    digitalWrite(in1_A, HIGH);
    digitalWrite(in1_B, LOW);
    analogWrite(pwm1, speed_1);

    digitalWrite(in2_A, LOW);
    digitalWrite(in2_B, HIGH);
    analogWrite(pwm2, speed_2);

    digitalWrite(in3_A, HIGH);
    digitalWrite(in3_B, LOW);
    analogWrite(pwm3, speed_3);

    digitalWrite(in4_A, LOW);
    digitalWrite(in4_B, HIGH);
    analogWrite(pwm4, speed_4);
}
```

```
void Stop()
{
    digitalWrite(in1_A, LOW);
```

```

digitalWrite(in1_B, LOW);
analogWrite(pwm1, speed_1);

digitalWrite(in2_A, LOW);
digitalWrite(in2_B, LOW);
analogWrite(pwm2, speed_2);

digitalWrite(in3_A, LOW);
digitalWrite(in3_B, LOW);
analogWrite(pwm3, speed_3);

digitalWrite(in4_A, LOW);
digitalWrite(in4_B, LOW);
analogWrite(pwm4, speed_4);
}

void Northwest();//ตะวันตกเฉียงเหนือ
{
    digitalWrite(in1_A, LOW);
    digitalWrite(in1_B, LOW);
    analogWrite(pwm1, speed_1);

    digitalWrite(in2_A, HIGH);
    digitalWrite(in2_B, LOW);
    analogWrite(pwm2, speed_2);

    digitalWrite(in3_A, HIGH);
    digitalWrite(in3_B, LOW);
    analogWrite(pwm3, speed_3);

```

```
digitalWrite(in4_A, LOW);
digitalWrite(in4_B, LOW);
analogWrite(pwm4, speed_4);
}
```

```
void Northeast();//ตะวันออกเฉียงเหนือ
```

```
{
digitalWrite(in1_A, HIGH);
digitalWrite(in1_B, LOW);
analogWrite(pwm1, speed_1);

digitalWrite(in2_A, LOW);
digitalWrite(in2_B, LOW);
analogWrite(pwm2, speed_2);

digitalWrite(in3_A, LOW);
digitalWrite(in3_B, LOW);
analogWrite(pwm3, speed_3);

digitalWrite(in4_A, HIGH);
digitalWrite(in4_B, LOW);
analogWrite(pwm4, speed_4);
}
```