เทคโนโลยีการประมวลผลภาพสำหรับการอ่านป้ายทะเบียนรถยนต์
IMAGE PROCESSING FOR CAR LICENSE PLATE READER

กิตติภัทท์ เพชรประดับ
กิตติภัทร์ แสงโชติหิรัญ
ศุภวิชญ์ จันทร์ศิริ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

เทคโนโลยีการประมวลผลภาพสำหรับการอ่านป้ายทะเบียนรถยนต์

IMAGE PROCESSING FOR CAR LICENSE PLATE READER

กิตติภัทท์ เพชรประดับ
กิตติภัทร์ แสงโชติหิรัญ
ศุภวิชญ์ จันทร์ศิริ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

IMAGE PROCESSING FOR CAR LICENSE PLATE READER

KITTIPHAT PETCHPRADUB

KITTIPHAT SAENGCHOTHIRAN

SUPAVITH CHANSIRI

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2017

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เทคโนโลยีการประมวลผลภาพสำหรับการอ่านป้ายทะเบียนรถยนต์

IMAGE PROCESSING FOR CAR LICENSE PLATE READER

ผู้จัดทำ	นายกิตติภัทท์	เพชรประดับ	57010099
	นายกิตติภัทร์	แสงโชติหิรัญ	57010100
	นายศุภวิชญ์	จันทร์ศิริ	57011274

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตหิรัญรัตน์)

เทคโนโลยีการประมวลผลภาพสำหรับการอ่านป้ายทะเบียนรถยนต์

โดย
นายกิตติภัทท์ เพชรประดับ 57010099
นายกิตติภัทร์ แสงโชติหิรัญ 57010100
นายศุภวิชญ์ จันทร์ศิริ 57011274

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตหิรัญรัตน์

ปีการศึกษา 2560

บทคัดย่อ

ในปัจจุบันเป็นยุคที่เทคโนโลยีมีการพัฒนาขึ้นอย่างต่อเนื่อง โดยมุ่งเน้นให้เกิดความสะดวกสบาย และความปลอดภัยที่เพิ่มขึ้นโดยเฉพาะในชีวิตประจำวัน ในปัจจุบันการรักษาความปลอดภัยโดยกล้องวงจรปิดนั้นเป็นสิ่งที่ใช้กันอย่างกว้างขวาง และแพร่หลาย ดังนั้นการพัฒนาความสามารถหรือประสิทธิภาพของระบบประมวลผลภาพจึงเป็นเรื่องที่ควรให้ความสำคัญ และไม่ควรมองข้ามเพราะจะสามารถนำไปใช้อำนวยความสะดวก และเพิ่มประสิทธิภาพในการรักษาความปลอดภัยให้ดียิ่งขึ้น ดังนั้นจึงได้ศึกษาออกแบบ และสร้างแบบจำลองการประมวลผลภาพในการอ่านป้ายทะเบียนยนต์ ซึ่งได้นำมาประยุกต์ให้สามารถเป็นซอฟต์แวร์โปรแกรม เพื่อนำมาใช้งานร่วมกับกล้องวงจรปิดในการอ่านป้ายทะเบียนรถยนต์ และนำข้อมูลที่ได้จากการประมวลผลไปประยุกต์ใช้เพื่ออำนวยความสะดวกและรักษาความปลอดภัยในชีวิตประจำวัน โดยในโครงงานนี้ เราจะนำข้อมูลที่ได้จากการประมวลผลนี้ไปใช้ในการตรวจสอบความถูกต้องของป้ายทะเบียนรถยนต์ที่เข้าออก เพื่อเป็นการรักษาความปลอดภัยและสามารถตรวจสอบได้ในกรณีที่เกิดปัญหา และสั่งการทำงานเปิดและปิดไม้กั้นรถยนต์ (Barrier Gate) อัตโนมัติเพื่อเป็นกรณีศึกษา โดยระบบสามารถตรวจสอบว่าหมายเลขป้ายทะเบียนรถยนต์ที่อ่านได้นั้นตรงกับหมายเลขที่อนุญาตหรือไม่ เพื่อสั่งการไม้กั้นให้เปิดและปิด

IMAGE PROCESSING FOR CAR LICENSE PLATE READER

By

Mr.Kittiphat Petchpradub 57010099

Mr.Kittiphat Saengchothiran 57010100

Mr.Supavith Chansiri 57011274

Advisor

Asst.Prof.Dr.Kongsak Anuntahiranrat

Academic Year 2017

ABSTRACT

Nowadays, technologies have been developing in order to improve convenience and security especially in daily life. These day, CCTV system is used widely and extensively so improving and developing this technology are crucial and should not be overlooked because developing this technology is going to make our daily life more convenience and increase efficiency in security. This project is designed and invented for improving security and convenience for barrier gate by using software that has image processing algorithm. The algorithm takes an input image of the number plate (number plate should be dominant in the image) and after filtering the image, it performs region based operations. Then it tries to capture the characters regions in a processed binary image and with the aid of template matching outputs the string of number plate characters. Then it checks the string of number plate characters. The barrier gate will be opened when the string of number plate characters with permission are detected.

กิตติกรรมประกาศ

ปริญญานิพนธ์บับนี้ประสบความสำเร็จไปได้ด้วยดี ด้วยความกรุณาจากอาจารย์ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตหิรัญรัตน์ อาจารย์สาขาวิชาวิศวกรรมแมคคาทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และอาจารย์ที่ปรึกษาโครงงานร่วม ดร.เฉลิมภัณฑ์ ฟองสมุทร อาจารย์สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา ที่ได้ให้คำแนะนำและเสนอแนวทางการแก้ปัญหาโครงงานนี้ให้มีความสมบูรณ์ยิ่งขึ้น

สุดท้ายนี้คณะผู้จัดทำ ขอขอบพระคุณ บิดา มารดา และบุคคลในครอบครัว ที่ได้ให้ความช่วยเหลือทุกๆ ด้าน รวมทั้งเพื่อนๆ พี่ๆ ที่ให้คำแนะนำและกำลังใจตลอดระยะเวลาในการทำโครงงานนี้

ผู้จัดทำ

นายกิตติภัทท์ เพชรประดับ

นายกิตติภัทร์ แสงโชติหิรัญ

นายศุภวิชญ์ จันทร์ศิริ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของการศึกษา	2
1.4 วิธีการดำเนินงานวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีงานวิจัยที่เกี่ยวข้อง	3
2.1 เมืองอัจฉริยะ (Smart City)	3
2.2 การประมวลผลภาพ (Image Processing)	3
2.3 แมตแล็บ (MATLAB : Matrix Laboratory)	4
2.4 ซีมูลิงค์ (Simulink)	5
2.5 พิกเซล (Pixel)	5
2.6 ความละเอียด (Resolution)	6
2.7 อาจีบี (RGB)	7
2.8 เกรสเกล (Gray Scale)	8
2.9 ไบนารีสเกล (Binary Scale)	9
2.10 อาดูโน่ (Arduino)	9

สารบัญ (ต่อ)

	หน้า
2.10.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม	10
2.11 เซอร์โวมอเตอร์ (Servo Motor)	10
2.11.1 หลักการทำงานของ Servo Motor	11
บทที่ 3 วิธีการทดลอง	12
3.1 การปรับปรุงรูปภาพ	12
3.1.1 คำสั่ง imread	12
3.1.2 คำสั่ง imresize	12
3.1.3 คำสั่ง rgb2gray	13
3.1.4 คำสั่ง medfilt2	13
3.1.5 คำสั่ง imdilate	14
3.1.6 คำสั่ง imerode	15
3.1.7 คำสั่ง imsubstract	16
3.1.8 คำสั่ง imadjust	16
3.1.9 คำสั่ง imfill	17
3.1.10 คำสั่ง bwareaopen	18
3.2 การดึงข้อมูลอักษรจากรูปภาพ	19
3.2.1 การวัดคุณลักษณะรูปภาพเพื่อแยกตัวอักษร	19
3.3 การเปลี่ยนข้อมูลเชิงรูปภาพให้เป็นข้อมูลเชิงอักขระ	21
3.3.1 การนำข้อมูลตัวอักษรมาเทียบกับฐานข้อมูล	21
3.3.2 การหาค่าดัชนีเพื่อเลือกใช้ตัวอักษร	22
3.3.3 การนำค่าดัชนีเปลี่ยนเป็นข้อมูลอักขระ	22
3.3.4 การเก็บค่าตัวอักษรจากป้ายทะเบียน	23
3.4 การต่อวงจร	25

สารบัญ (ต่อ)

	หน้า
3.4.1 การต่อวงจรของบอร์ด Arduino กับ Servo Motor	25
3.4.2 การต่อวงจรของบอร์ด Arduino กับสวิตช์	25
3.5 การเชื่อมต่อบอร์ด Arduino กับ Matlab	26
3.6 การใช้ Matlab และ Simulink เพื่อควบคุมเซอร์โวมอเตอร์	33
3.7 แนวคิดด้านการออกแบบ	33
3.8 แบบจำลอง	34
บทที่ 4 ผลการดำเนินงาน	35
4.1 ผลการทดลองการอ่านป้ายทะเบียน	35
4.1.1 ผลการทดลองการอ่านป้ายทะเบียนที่สมบูรณ์	35
4.1.2 ผลการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์	37
4.1.3 สรุปผลการทดลองการอ่านป้ายทะเบียน	38
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	40
5.1 สรุปผลการดำเนินงาน	40
5.2 ข้อเสนอแนะ	40
5.3 ปัญหาในการดำเนินงาน	41
เอกสารอ้างอิง	42

สารบัญรูป

รูปที่	หน้า
2.1 รูปภาพอธิบายความหมายของ Pixel	6
2.2 รูปภาพอธิบายความหมายของ Resolution	7
2.3 รูปภาพอธิบายความหมายของ RGB	8
2.4 รูปภาพอธิบายความหมายของ Gray Scale	8
2.5 รูปภาพอธิบายความหมายของ Binary Scale	9
3.1 ตัวอย่างการใช้คำสั่ง rgb2gray	13
3.2 ตัวอย่างการใช้คำสั่ง medfilt2	14
3.3 ตัวอย่างการใช้คำสั่ง imdilate	15
3.4 ตัวอย่างการใช้คำสั่ง imerode	15
3.5 ตัวอย่างการใช้คำสั่ง imsubstract	16
3.6 ตัวอย่างการใช้คำสั่ง imadjust	17
3.7 ตัวอย่างการใช้คำสั่ง imfill	18
3.8 ป้ายทะเบียนก่อนใช้คำสั่ง bwareaopen	19
3.9 ป้ายทะเบียนหลังใช้คำสั่ง bwareaopen	19
3.10 รูปตัวอย่างการวัดคุณสมบัติรูปภาพ	20
3.11 การแบ่งตัวอักษรโดยการวัดคุณลักษณะ BoundingBox	21
3.12 ตัวอย่างรูปภาพในฐานข้อมูล	22
3.13 Flow Chart การเปลี่ยน Binary Image ให้เป็นข้อมูลรูปแบบ Char	23
3.14 ผลการประมวลผลภาพ	24
3.15 การต่อวงจรของบอร์ด Arduino และ Servo Motor	25
3.16 การต่อวงจรของบอร์ด Arduino กับสวิตช์	25
3.17 ตัวอย่างการลง Get Hardware Support Packages	26
3.18 ตัวอย่างการเลือก Install From Internet	26

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.19 ตัวอย่างการเลือก Install	27
3.20 ตัวอย่างการกดเริ่ม Simulink	27
3.21 ตัวอย่างการเปิด Library Browser ของ Simulink	28
3.22 ตัวอย่างการเลือก Simulink Library	28
3.23 ตัวอย่างเมื่อลาก Standard Servo Write ลงในหน้าต่าง Untitled–Simulink	29
3.24 ตัวอย่างการเลือก Chart	29
3.25 ตัวอย่างการเลือก Constant	30
3.26 ตัวอย่างการเลือก Digital Input	30
3.27 ตัวอย่างหลังจากโยงเส้นเชื่อมกัน	31
3.28 ตัวอย่างเมื่อกดเข้ามาใน Chart	31
3.29 ตัวอย่างการเขียนเงื่อนไขลงใน Chart	32
3.30 ผลการทดลองการอ่านป้ายทะเบียน	33
3.31 ประตูกั้นทาง	33
3.32 เซอร์โวมอเตอร์และรูปตัวอย่างประตูกั้นทางจำลอง	34
3.33 แบบจำลองประตูกั้นทาง	34
4.1 ผลการทดลองการอ่านป้ายทะเบียน	35
4.2 ผลการทดลองการอ่านป้ายทะเบียน	36
4.3 ผลการทดลองการอ่านป้ายทะเบียน	36
4.4 ผลการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์	37
4.5 ผลการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์	37
4.6 ผลการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์	38
4.7 กราฟแสดงเวลาที่ใช้ในการทดลอง	39

สารบัญตาราง

ตารางที่	หน้า
4.1 ผลการทดลองการอ่านป้ายทะเบียน	39

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันเป็นยุคที่เทคโนโลยีมีการพัฒนาขึ้นอย่างต่อเนื่อง ทำให้ได้ยินและคุ้นเคยกับคำว่าเมืองอัจฉริยะ (Smart City) และมีแนวโน้มที่จะได้ยินบ่อยมากขึ้น โดยการพัฒนาเมืองหรือชุมชนให้เป็นเมืองอัจฉริยะนั้นก็มีเป้าหมาย เพื่อพัฒนาและมุ่งเน้นให้เกิดความสะดวกสบาย และความปลอดภัยที่เพิ่มขึ้นโดยเฉพาะในชีวิตประจำวัน จึงได้เลือกหนึ่งในเทคโนโลยีที่น่าสนใจและคาดว่าจะเข้ามามีบทบาทและมีความสำคัญต่อการพัฒนาเมืองหรือชุมชนให้มีความอัจฉริยะมากขึ้นคือ ระบบการประมวลผลภาพ

ในปัจจุบันการรักษาความปลอดภัยโดยกล้องวงจรปิดนั้นเป็นสิ่งที่ใช้กันอย่างกว้างขวาง และแพร่หลาย เช่น ใช้เพื่อบันทึกเหตุการณ์ต่างๆ ภายในพื้นที่ที่ต้องการ หรือใช้เพื่อบันทึกรถยนต์และป้ายทะเทียนรถยนต์ที่เข้ามาภายในอาคารหรือห้างสรรพสินค้า อีกทั้งเทคโนโลยีกล้องยังสามารถพัฒนาเพื่อนำมาใช้ร่วมกับเทคโนโลยีหรือเครื่องจักรต่างๆ ในปัจจุบัน ดังนั้นการพัฒนาความสามารถหรือประสิทธิภาพของระบบประมวลผลภาพจึงเป็นเรื่องที่ควรให้ความสำคัญ และไม่ควรมองข้ามเพราะสามารถนำไปใช้อำนวยความสะดวก และเพิ่มประสิทธิภาพในการรักษาความปลอดภัยให้ดียิ่งขึ้น

โดยในวิทยานิพนธ์นี้จะทำการศึกษาการประมวลผลภาพที่สามารถเขียนได้โดยซอฟต์แวร์โปรแกรม โดยจะทำการสาธิตการประมวลผลภาพ และนำข้อมูลที่ได้จากการประมวลผลไปประยุกต์ใช้ในแบบจำลองเพื่อเป็นกรณีศึกษา โดยมีจุดมุ่งหมายเพื่อให้สามารถพัฒนาต่อยอดให้เป็นสิ่งอำนวยความสะดวกและเพิ่มความปลอดภัยได้

1.2 วัตถุประสงค์

1. เพื่อศึกษาเทคโนโลยีและซอฟต์แวร์โปรแกรมการประมวลผลภาพ
2. เพื่อศึกษาการประยุกต์ใช้ข้อมูลที่ได้จากการประมวลผล
3. จำลองแบบการประยุกต์ใช้การประมวลผลภาพเพื่อเป็นกรณีศึกษา

1.3 ขอบเขตของการศึกษา

1. สามารถใช้ซอฟต์แวร์โปรแกรมเพื่ออ่านภาพป้ายทะเบียนรถยนต์ได้

2. นำข้อมูลจากการประมวลผลภาพไปใช้ร่วมกับไมโครคอนโทรลเลอร์เพื่อสร้างแบบจำลองกรณีศึกษา

1.4 วิธีการดำเนินการวิจัย

1. ศึกษาและทดลองการใช้งานโปรแกรม MATLAB เพื่อเรียนรู้คำสั่งในการใช้เขียน และทำความรู้จักกับ Library ในการประมวลผลภาพ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้เรื่องเทคโนโลยีการประมวลผลภาพและการนำไปประยุกต์ใช้

2. สามารถทำแบบจำลองการประยุกต์ใช้เทคโนโลยีการประมวลผลภาพเพื่อเป็นกรณีศึกษา

3. แนวคิดและต้นแบบในการนำไปพัฒนาต่อยอดในอนาคต

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 เมืองอัจฉริยะ (Smart City)

เมืองที่ได้รับการออกแบบโดยให้ความสำคัญในองค์ประกอบหลักคือ การพัฒนารูปแบบโครงสร้างของเมืองที่สอดรับกับแนวคิดของเมืองอัจฉริยะ การส่งเสริมการอนุรักษ์พลังงานและสิ่งแวดล้อม รวมทั้งการส่งเสริมการใช้พลังงานทดแทน ประกอบกับการนำเทคโนโลยีสารสนเทศและข้อมูล มาช่วยในการบริหารจัดการทรัพยากรของเมืองเพื่อให้เกิดประโยชน์สูงสุด อาทิ ระบบควบคุมการจราจรอัจฉริยะ ระบบควบคุมอาคารอัจฉริยะ และระบบตรวจจับป้ายทะเบียนรถยนต์

ในปัจจุบัน Smart City หรือเมืองอัจฉริยะเป็นกระแสที่กำลังมีการพัฒนากันทั่วโลก เพราะเป็นเรื่องของการนำเทคโนโลยีดิจิทัลมาปรับใช้กับเมืองเพื่อให้มีความน่าอยู่มากขึ้น มีการพัฒนาคุณภาพชีวิตที่ดีขึ้น สร้างความปลอดภัยได้มากขึ้น แนวคิดในการนำเทคโนโลยีมาพัฒนาเมืองให้มีความอัจฉริยะนั้นจึงเป็นหัวข้อที่น่าสนใจ และได้นำเรื่องเกี่ยวกับเทคโนโลยีการประมวลผลภาพมาศึกษาทำปริญญานิพนธ์

2.2 การประมวลผลภาพ (Image Processing)

การนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่ต้องการทั้งในเชิงคุณภาพและปริมาณ โดยมีขั้นตอนต่างๆ ที่สำคัญคือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ จากนั้นจะสามารถนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์ และสร้างเป็นระบบ เพื่อใช้ประโยชน์ในงานด้านต่างๆ เช่น ระบบจดจำลายนิ้วมือเพื่อตรวจสอบว่าภาพลายนิ้วมือที่มีอยู่นั้นเป็นของผู้ใด ระบบตรวจสอบคุณภาพของผลิตภัณฑ์ในกระบวนการผลิตของโรงงานอุตสาหกรรม ระบบคัดแยกเกรดหรือคุณภาพของพืชผลทางการเกษตร ระบบอ่านรหัสไปรษณีย์อัตโนมัติ เพื่อคัดแยกปลายทางของจดหมายที่มีจำนวนมากในแต่ละวันโดยใช้ภาพถ่ายของรหัสไปรษณีย์ที่อยู่บนซอง และระบบอ่านป้ายทะเบียนรถที่เข้าและออกอาคาร โดยใช้ภาพถ่ายของป้ายทะเบียนรถเพื่อประโยชน์ในด้านความปลอดภัย ซึ่งเป็นกรณีศึกษาในปริญญานิพนธ์นี้ โดยงานในลักษณะเหล่านี้หากให้มนุษย์วิเคราะห์เอง มักต้องใช้เวลามากและใช้แรงงานสูง อีกทั้งหากจำเป็นต้องวิเคราะห์ภาพเป็นจำนวนมาก ผู้วิเคราะห์ภาพอาจเกิดอาการล้า ส่งผลให้เกิดความผิดพลาดขึ้นได้ ดังนั้นคอมพิวเตอร์จึงมีบทบาทสำคัญในการทำหน้าที่

เหล่านี้แทนมนุษย์ อีกทั้งเป็นที่ทราบโดยทั่วกันว่า คอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมหาศาลได้ในเวลาอันสั้น จึงมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพการประมวลผลภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบต่างๆ ดังกล่าวข้างต้น

2.3 แมตแล็บ (MATLAB : Matrix Laboratory)

เป็นซอฟต์แวร์ในการคำนวณและการเขียนโปรแกรมที่มีความสามารถครอบคลุม ตั้งแต่การพัฒนาอัลกอริธึม การสร้างแบบจำลองทางคณิตศาสตร์ และการทำซิมูเลชั่นของระบบ การสร้างระบบควบคุม และโดยเฉพาะเรื่อง Image Processing และ Wavelet การสร้างเมตริกซ์

MATLAB เป็นโปรแกรมสำเร็จรูปที่ใช้กันอย่างแพร่หลายในแวดวงของนักวิทยาศาสตร์และวิศวกรในปัจจุบัน ชื่อโปรแกรม MATLAB นั้นย่อมาจาก Matrix Laboratory MATLAB ได้เริ่มต้น ขึ้นเพื่อต้องการให้สามารถแก้ปัญหาตัวแปรที่มีลักษณะเป็นเมทริกซ์ได้ง่ายขึ้น MATLAB เริ่มพัฒนาครั้งแรกโดย Dr. Cleve Molor ซึ่งเขียนโปรแกรมนี้ขึ้นมาด้วยภาษาฟอร์แทรน โดยโปรแกรมนี้ได้ พัฒนาภายใต้โครงการ LINPACK และ EISPACK

MATLAB เป็นโปรแกรมสำหรับการคำนวณเชิงตัวเลข และ Visualization ที่มีประสิทธิภาพสูง และเป็นเครื่องหมายการค้าของบริษัท MathWorks ซึ่งการทำงานภายในโปรแกรม MATLAB อยู่บนพื้นฐานของการคำนวณทางเมตริกซ์ (Matrix Manipulation and Computation) เป็นแกนหลักโปรแกรม MATLAB สามารถทำงานแบบโต้ตอบ ซึ่งมีลักษณะคล้ายกับภาษา Basic ในโปรแกรม QBasic และแบบ Compiled Mode ซึ่งคล้ายกับภาษา C และ Pascal นอกจากนี้สามารถใช้ MATLAB เสมือนเครื่องคิดเลข ทำการคำนวณทางคณิตศาสตร์ได้ทันที การเขียนโปรแกรม MATLAB สามารถทำได้ง่ายเมื่อเทียบกับภาษาโปรแกรมอื่นๆ สามารถสร้างและกำหนดค่าตัวแปรได้ทันทีโดยไม่ต้องประกาศตัวแปรก่อน ซึ่งสะดวกในการเขียนโปรแกรม เนื่องจากผู้ใช้ไม่ต้องกังวลในเรื่องโครง สร้างของภาษา เช่น ในเรื่องการต้องประกาศตัวแปรก่อนใช้งาน แต่มาสนใจอัลกอริทึมในการแก้โจทย์ปัญหา เป็นหลัก ข้อมูลไม่ว่าจะเป็นตัวเลข หรือตัวอักษร (Strings) จะถูกจัดเก็บในรูปแบบของแถวและหลัก หรือ Array ซึ่งก็คือ Matrix นั่นเอง เช่น จำนวนสเกลล่าร์ (Scalar) จะถูกแทนที่ด้วยเมตริกซ์ขนาด 1 x 1 ข้อมูลที่เป็นเวกเตอร์จะถูกแทนที่ด้วยเมตริกซ์ที่มีเพียง 1 แถว ในกรณีที่เป็น เวกเตอร์แบบแถว (Row Vector) หรือถูกแทนที่ด้วยเมตริกซ์ที่มีเพียง 1 หลักในกรณีที่เป็นเวกเตอร์แบบหลัก (Column Vector) เป็นต้น การที่ MATLAB ถูกออกแบบมาให้มีการทำงานภายในเช่นนี้ ทำให้การเขียนโปรแกรมแก้ปัญหาที่มีลักษณะของ Vector และ Matrix เป็นเรื่องง่าย

2.4 ซีมูลิงค์ (Simulink)

โปรแกรม MATLAB สามารถจำลอง ทดสอบ และวิเคราะห์การทำงานของระบบพลศาสตร์ ในเชิงเวลา ได้โดยการใช้ Simulink ซึ่งเป็นเครื่องมือ Toolbox ที่อยู่ในโปรแกรม MATLAB โดยจะทำงานภายใต้หน้าต่างที่เป็นการเชื่อมต่อทางรูปภาพ (GUI) ของ Simulink เท่านั้น คำว่า Simulink มาจากคำสองคำคือ Simulation และ Link การใช้งาน Simulink จะกระทำโดยการนำ Block ในหน้าต่าง Library-Simulink มาต่อกันตามที่ต้องการและสามารถจำลองระบบได้ทั้งระบบที่เป็นเชิงเส้น ไม่เป็นเชิงเส้น ระบบเวลาต่อเนื่องและไม่ต่อเนื่อง การจำลองระบบสามารถกระทำได้โดยป้อนอินพุตที่ป้อนเข้าไป

เมื่อเริ่มต้นใช้งาน Simulink โปรแกรม MATLAB จะกำหนดชื่อหน้าต่าง Simulink โดยอัตโนมัติเป็น Untitled หน้าต่าง Simulink นี้จะทำงานเชื่อมต่อกับหน้าต่างคำสั่งของโปรแกรม MATLAB โดย Simulink สามารถรับส่งข้อมูลผ่าน Workspace ของหน้าต่างคำสั่ง ในกรณีที่ระบบใช้ฟังก์ชันหรือ Block ที่เป็น To Workspace ตลอดจน Block ของหมวดหมู่ต่างๆ ที่ไม่ได้กำหนดค่าตัวเลขแต่กำหนดเป็นค่าตัวแปรในหน้าต่าง Simulink และเมื่อเก็บ (Save As) ระบบที่จำลองได้ด้วย Simulink โปรแกรม MATLAB จะกำหนดไฟล์ที่ทำการ Simulink เป็นชื่อไฟล์นามสกุล .mdl

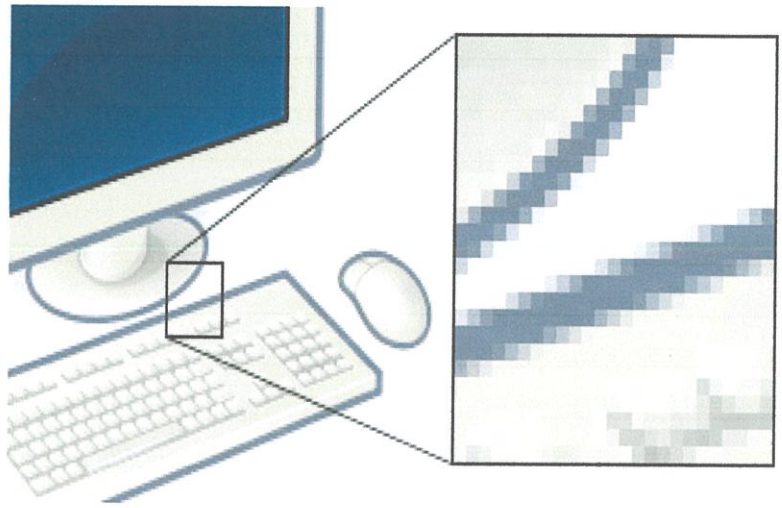
เมื่อเริ่มใช้งาน Simulink จะพบหน้าต่าง Library Simulink ที่ประกอบด้วย Block Diagram ที่เป็นโหมดของอุปกรณ์ต่างๆ ที่จะนำไปใช้ในการจำลองระบบ ซึ่งอุปกรณ์เหล่านี้ไม่สามารถทำงานได้เพียงตัวเดียวโดดๆ ต้องเชื่อมต่อกับอุปกรณ์ตัวอื่นถึงจะทำงานได้ การเลือกใช้งานโหมดใดโหมดหนึ่งในหน้าต่าง Library Simulink ให้ดับเบิลคลิกไปที่โหมดนั้น แล้วจะปรากฏหน้าต่าง แสดงรายละเอียดของ Block ต่างๆ บนหน้าต่าง Library Simulink ของโหมดที่เลือกไว้ออกมา

2.5 พิกเซล (Pixel)

หน่วยพื้นฐานซึ่งเล็กที่สุดของภาพดิจิทัล เทียบได้กับจุดสีของภาพ 1 จุด หลากหลายสี หลายๆ จุดที่เรียงชิดติดกันถูกรวมกันทำให้เกิดเป็นภาพนั่นเอง แต่ 1 Pixel จะเป็นสีหนึ่งสีใดเพียงสีเดียวเท่านั้นจะมีสีอื่นไม่ได้ เนื่องจากว่าเป็นส่วนที่เล็กที่สุดของการแสดงผล Pixel นั้นมีความสำคัญต่อการสร้างกราฟฟิกของคอมพิวเตอร์มาก เพราะทุกๆ ส่วนของกราฟฟิก เช่น จุด เส้น แบบลาย และสีของภาพล้วนเกิดจาก Pixel ทั้งสิ้น ดังรูปที่ 2.1

นอกเหนือจากหน่วยของการแสดงที่เป็น Pixel ยังมีคำว่า BPP ย่อมาจากคำว่า Bits Per Pixel คือค่าของจำนวนสีที่สามารถแสดงได้ในหนึ่งพิกเซล ดังที่บอกไปในข้างต้นว่า 1 Pixel จะสามารถแสดงผลได้เพียงสีเดียวเท่านั้น ดังนั้นค่า BPP จะเป็นตัวบอกว่าใน 1 Pixel จะสามารถเปลี่ยนสีได้กี่สี

ดังนั้นจึงสามารถสรุปได้ว่า หากค่าพิกเซลยิ่งสูงมากเท่าไหร่ภาพที่ได้ก็จะยิ่งมีความละเอียดสูงมากตามไปด้วย



รูปที่ 2.1 รูปภาพอธิบายความหมายของ Pixel

2.6 ความละเอียด (Resolution)

ความละเอียดในการแสดงผลของภาพ โดยจะกำหนดเป็น Mega Pixel ยิ่งมีค่ามากก็จะยิ่งมีความละเอียดสูง โดยกำหนดจากจำนวนเม็ดสี (Pixel) ต่อหนึ่งหน่อยความยาวของภาพ เช่น หากภาพนั้นมีค่า Resolution = 150 Pixels/Inches แสดงว่าในพื้นที่ 1 ตารางนิ้วของภาพนั้นประกอบไปด้วยเม็ดสีจำนวน 150 สี (Pixels) เป็นต้น

ค่า Resolution สามารถกำหนดค่าได้หลายรูปแบบ เช่น

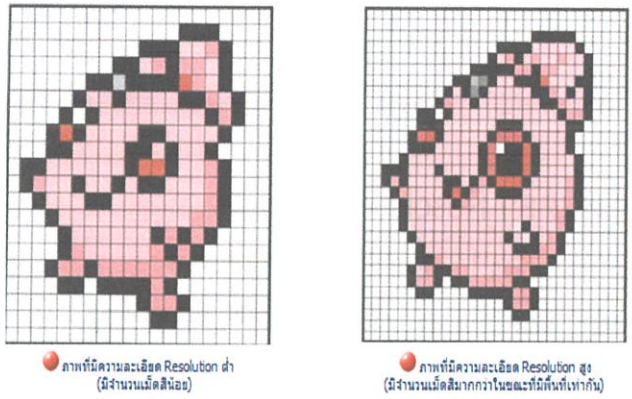
1. แสดงผลเป็นจำนวนจุด หรือ Pixel ในแนวนอนและแนวตั้ง เช่น จอแสดงผล Resolution 1920 x 1080 ซึ่งหมายถึง มีจำนวนจุดสำหรับแสดงผลในแนวนอน 1,920 จุด และมีจุดแสดงผลในแนวตั้งจำนวน 1,080 จุด ซึ่งการแสดงผลด้วยความละเอียดขนาดเท่านี้ นิยมเรียกว่าการแสดงผลแบบ Full HD นั่นเอง นอกจากนี้ยังมีการแสดงค่าอื่นในลักษณะนี้อีกมากมาย เช่น 1280 x 720, 1366 x 768 และ 800 x 600 เป็นต้น การแสดงค่า Resolution แบบนี้ยิ่งค่าในแต่ละแนวมีค่าสูง ก็จะสามารถแสดงผลได้ละเอียด และคมชัดมากขึ้น

2. การแสดงค่า Resolution แบบการนับความหนาแน่นของจุด หรือจำนวน Pixel ต่อตารางนิ้ว เช่น 326 จุดต่อนิ้ว (Pixel Per Inch, PPI) หรือ 470 PPI เป็นต้น ยิ่งค่า PPI สูง ก็จะยิ่งแสดงภาพได้ละเอียด และคมชัดมากกว่าตัวเลขน้อยๆ

3. การแสดงค่า Resolution เป็นจำนวนจุดทั้งหมดที่มีอยู่ในจอแสดงผล เช่น ถ้าจอแสดงผลนั้นสามารถแสดงผลได้ที่ความละเอียด 1366 x 768 ก็จะหมายความว่าหน้าจอนี้สามารถแสดงภาพได้จำนวน 1,049,088 Pixel หรือ 1.05 Mega Pixel ส่วนจอแสดงผลแบบ 4K (4096 x 2160) จะมี

จำนวนจุดแสดงผลทั้งสิ้น 8,847,360 Pixel หรือ 8.8 Mega Pixel ในระบบนี้ก็เช่นเดียวกับระบบอื่น กล่าวคือยิ่งตัวเลขมาก ก็ยิ่งแสดงผลได้ละเอียดมากขึ้น

ค่า Resolution นั้น ปัจจุบันนิยมนำมาใช้กับอุปกรณ์ที่มีหน้าจอแสดงผลแทบทุกชนิด เช่น LCD TV, LED TV, Plasma TV, Computer Monitor, แทบเล็ต และสมาร์ทโฟน เป็นต้น ดังรูปที่ 2.2



รูปที่ 2.2 รูปภาพอธิบายความหมายของ Resolution

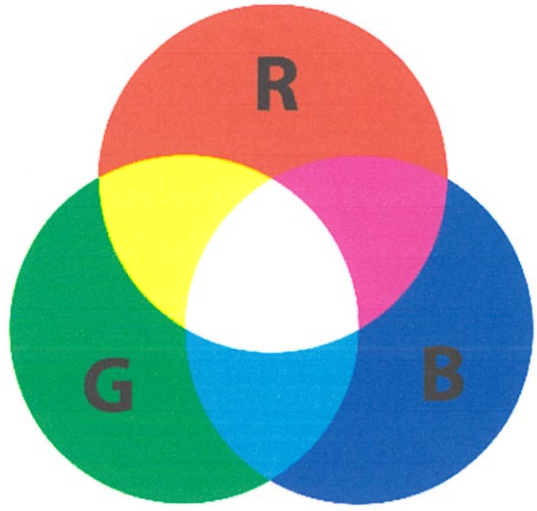
2.7 อาร์จีบี (RGB)

RGB Image หรือ Truecolor Image เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติ ขนาด m x n x 3 โดยที่ m คือ ความยาว และ n คือ ความกว้างของภาพในหน่วยพิกเซล ส่วนมิติสุดท้ายนั้นในแต่ละมิติจะเก็บค่าสีแยกกัน คือ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue)

RGB ย่อมาจาก Red, Green และ Blue คือ ระบบสีของแสง เกิดจากการหักเหของแสงกลายเป็นสีรุ้ง ด้วยกัน 7 สี ซึ่งเป็นช่วงแสงที่ตาของคนเราสามารถมองเห็นได้ แสงสีม่วงจะมีความถี่สูงสุดเรียกว่าอุลตร้าไวโอเรต และแสงสีแดงจะมีความถี่ต่ำสุด เรียกว่า อินฟาเรต คลื่นแสงที่มีความถี่สูงกว่าสีม่วง และต่ำกว่าสีแดงนั้น สายตาของมนุษย์ไม่สามารถรับได้ แสงสีทั้งหมดเกิดจากแสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้งสามสีถือเป็นแม่สีของแสง ดังรูปที่ 2.3

คุณสมบัติที่โดดเด่นของช่วงสี RGB คือ สีที่เห็นจากการผสมของสีต่างๆ ของแสง ส่งผลให้ RGB คือโมเดลผสมสีแบบเพิ่ม ยิ่งใช้สีมากรูปภาพสุดท้ายก็ยิ่งสว่างมากขึ้น

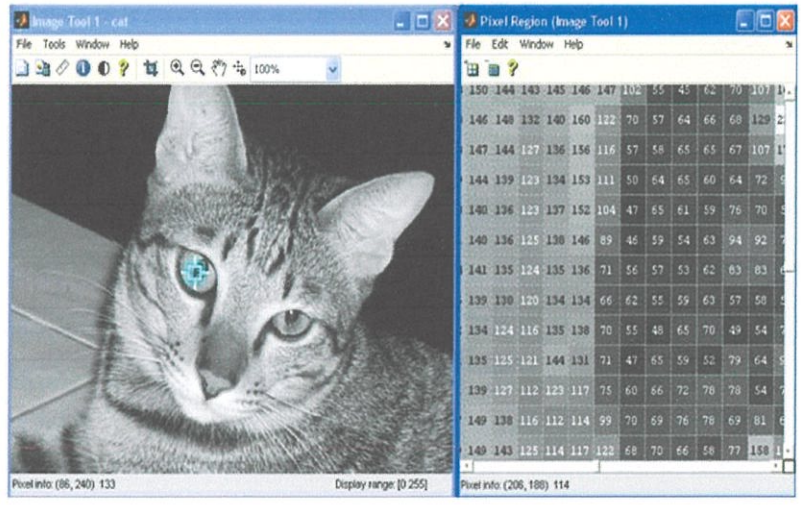
เมื่อนำแม่สีของแสงทั้ง 3 มาผสมกัน ในปริมาณความสว่างเท่ากันก็จะได้เป็นแสงที่มีสีขาว แต่ถ้าผสมกันระหว่างแสงระดับความสว่างต่างกัน ก็จะได้ผลเป็นแสงสีๆ มากมายเป็นล้านสี ส่วนใหญ่การใช้สีลักษณะนี้จะใช้ในอุปกรณ์ที่เกี่ยวกับแสง เช่น จอภาพ กล้องดิจิตอล สแกนเนอร์ เป็นต้น



รูปที่ 2.3 รูปภาพอธิบายความหมายของ RGB

2.8 เกรย์สเกล (Gray Scale)

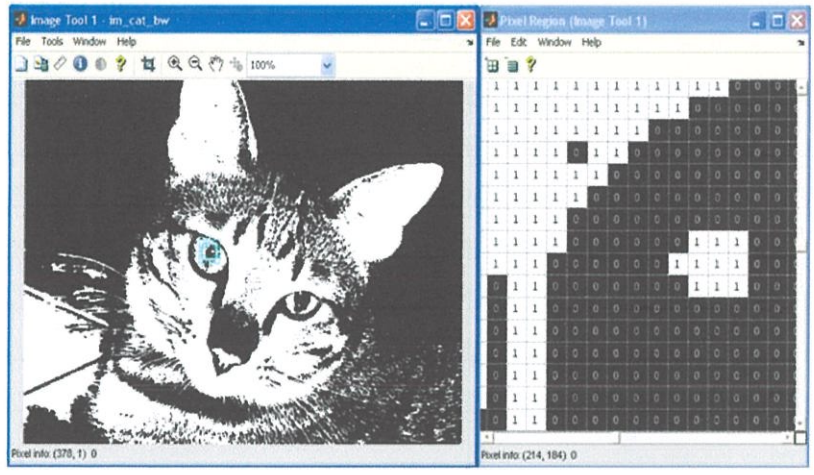
เป็นโหมดสีที่มีการไล่เฉดสีของสีเทา จากสีเทาอ่อนไปจนถึงสีเทาแก่ได้ถึง 256 สี ซึ่งแตกต่างกับภาพขาว-ดำ ที่มีเพียง 2 สีคือ ขาวกับดำ สีใน Gray Scale นี้ แสดงถึงความเข้มของสี (Intensity) ในระดับต่างๆ โดยสีดำเป็นส่วนที่มีความเข้มของสีน้อย และสีขาวจะมีความเข้มของสีมาก จำนวนระดับของสีขึ้นอยู่กับขนาดของบิตที่ใช้เก็บค่าสี โดยทั่วไปแล้วจะเก็บข้อมูลสีประเภทนี้ด้วยข้อมูลขนาด 8 บิต หรือ 1 ไบต์ ภาพที่ได้ในโหมดนี้เหมาะสำหรับการนำภาพที่ตกแต่งไปใช้งานกับเครื่องพิมพ์แบบขาว-ดำ ข้อดีของภาพในโหมด Gray Scale คือ จะได้ไฟล์ที่มีขนาดเล็กแต่ยังมีการไล่เฉดสีให้เห็นอยู่ในภาพ ดังรูปที่ 2.4



รูปที่ 2.4 รูปภาพอธิบายความหมายของ Gray Scale

2.9 ไบนารีสเกล (Binary Image)

เป็นโหมดสีที่มีเฉพาะสีขาวและสีดำ ใช้พื้นที่เพียง 1 Bit โดยสีขาวจะมีค่าพิกเซลเป็น 1 และสีดำจะมีค่าพิกเซลเป็น 0 เหมาะที่จะใช้งานประเภทภาพวาดหรืองานที่เป็นลายเส้น ไม่ต้องการรายละเอียดของสีอื่นๆ ภาพที่ได้จะมีความละเอียดน้อยมาก แต่ก็จะมีขนาดของไฟล์ที่เล็กมากตามไปด้วย ดังรูปที่ 2.5



รูปที่ 2.5 รูปภาพอธิบายความหมายของ Binary Scale

2.10 อาดูโน่ (Arduino)

เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือ ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา Input/Output ของบอร์ดหรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้

Arduino ถูกใช้ประโยชน์ในลักษณะเดียวกับ MCU คือ ใช้ติดต่อสื่อสารและควบคุมอุปกรณ์ไฟฟ้าอื่นๆ ด้วยการเขียนโปรแกรมให้กับ MCU เพื่อควบคุมการรับส่งสัญญาณทางไฟฟ้าตามเงื่อนไขต่างๆ ตัวอย่าง การประยุกต์ใช้ Arduino ในชีวิตประจำวัน เช่น ระบบเปิด/ปิดไฟในบ้านอัตโนมัติ,

ระบบรดน้ำต้นไม้อัตโนมัติ, ระบบเปิด/ปิดประตูอัตโนมัติ, ระบบเครื่องซักผ้าหยอดเหรียญ หรือใช้ควบคุมความเร็วและทิศทางการหมุนของมอเตอร์ เป็นต้น

2.10.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

1. Arduino กำลังเป็นที่นิยม และเป็นที่สนใจสำหรับนักอิเล็กทรอนิกส์ทั้งมือใหม่ และมือเก่า ทำให้สามารถหาอ่านคู่มือ วิธีใช้และวิธีแก้ปัญหาต่างๆ ได้ง่ายบนอินเทอร์เน็ต

2. Arduino พร้อมใช้งานทันที เพราะบอร์ด Arduino ติดตั้งอุปกรณ์จำเป็นพื้นฐานมาให้หมดแล้ว

3. Arduino สามารถเขียนโปรแกรมสั่งงานด้วยไวยากรณ์ภาษา C/C++ ซึ่งง่ายสำหรับผู้ที่มีพื้นฐานด้านการเขียนโปรแกรมอยู่บ้างแล้ว แต่สำหรับผู้ที่ไม่เคยเขียนโปรแกรมมาก่อนเลย ก็สามารถเริ่มต้นศึกษา และหาหนังสืออ่านได้ไม่ยาก นอกจากนี้ยังมี Library ให้เลือกใช้มากมาย ทำให้การเขียนโปรแกรมทำได้ง่ายและรวดเร็ว

4. Arduino ราคาไม่แพงเกินไปสำหรับผู้ที่อยากจะเริ่มต้นใช้งาน

5. การอัปโหลดโปรแกรมที่เขียนบนคอมพิวเตอร์ลงไปที่ Arduino ก็ทำได้โดยง่าย แค่ใช้สาย USB ต่อบอร์ด Arduino เข้ากับคอมพิวเตอร์ แล้วอัปโหลดด้วยโปรแกรม Arduino IDE

2.11 เซอร์โวมอเตอร์ (Servo Motor)

เป็นมอเตอร์ที่มีการควบคุมการเคลื่อนที่ของ (State) ไม่ว่าจะเป็นระยะ ความเร็ว มุมการหมุน โดยใช้การควบคุมแบบป้อนกลับ (Feedback Control) เป็นอุปกรณ์ที่สามารถควบคุมเครื่องจักรกลหรือระบบการทำงานนั้นๆ ให้เป็นไปตามความต้องการ เช่น ควบคุมความเร็ว (Speed), ควบคุมแรงบิด (Torque), ควบคุมแรงตำแหน่ง (Position), ระยะทางในการเคลื่อนที่(หมุน) (Position Control) ของตัวมอเตอร์ได้ ซึ่งมอเตอร์ทั่วไปไม่สามารถควบคุมในลักษณะงานเบื้องต้นได้ โดยให้ผลลัพธ์ตามความต้องการที่มีความแม่นยำสูง

โดยทั่วจะมีทั้งดีซีและเอซีเซอร์โว ในเครื่องจักรรุ่นเก่าๆ จะพบว่า DC Servo Motor มีการใช้เครื่องจักรกลอุตสาหกรรมมากกว่า AC Servo Motor เนื่องจากช่วงที่ผ่านมาการควบคุมกระแสกระแสสูงๆ นั้นจะต้องใช้ SCRs แต่ปัจจุบันทรานซิสเตอร์ได้พัฒนาขีดความสามารถให้ตัดต่อกระแสสูงและใช้งานที่ความถี่ได้สูงๆ ขึ้น จึงทำให้ระบบควบคุมทางเอซีและระบบเซอร์โวได้ถูกนำมาใช้งานมากขึ้น

2.11.1 หลักการทำงานของเซอร์โวมอเตอร์

การทำงานของเซอร์โวมอเตอร์ชนิดนี้จะคล้ายกับการทำงานของซิงโครนัสมอเตอร์ 3 เฟส กล่าวคือ เมื่อมีการควบคุมให้คอนโทรลเลอร์จ่ายกระแสไฟฟ้าเข้าไปยังขดลวดที่สเตเตอร์ แกนเหล็กของสเตเตอร์จะกลายเป็นแม่เหล็กไฟฟ้า และหมุนเคลื่อนที่ด้วยความเร็วที่แปรผันตามความถี่ ซึ่งเรียกว่า ความเร็วซิงโครนัส หรือความเร็วสนามแม่เหล็กหมุน และจะดูดให้โรเตอร์ซึ่งเป็นแม่เหล็กถาวรหมุนเคลื่อนที่ตาม

บทที่ 3
วิธีการทดลอง

3.1 การปรับปรุงรูปภาพ

3.1.1 คำสั่ง imread

เริ่มต้นจะทำการรับข้อมูลที่ได้จากกล้องเข้ามาโดยใช้คำสั่ง imread ซึ่งสามารถรับประเภทของรูปภาพได้หลากหลาย เช่น .JPG, .PNG, .PSD และ .TIFF เป็นต้น

F = imread('Picture Name.jpg')

F คือ ตัวแปรสำหรับการเก็บข้อมูลรูปภาพ

Imread คือ คำสั่งสำหรับการรับข้อมูลรูปภาพจากกล้องเข้าสู่โปรแกรม MATLAB

Picture Name.jpg คือ ชื่อไฟล์ของรูปภาพที่ต้องการนำมาประมวลผล

3.1.2 คำสั่ง imresize

รูปภาพที่รับเข้ามาจากกล้องจะมีขนาดไฟล์ของรูปภาพที่ใหญ่มาก เนื่องจากกล้องมีความละเอียด 1080p หรือมีขนาด 1920 x 1080 Pixels ซึ่งหากใช้รูปภาพที่มีไฟล์ขนาดใหญ่มาใช้ประมวลผลในโปรแกรม MATLAB จะทำให้การประมวลผลใช้เวลานาน จึงต้องทำการลดขนาดไฟล์ของรูปภาพโดยการลดความละเอียดของรูปภาพลง โดยใช้คำสั่ง imresize ในที่นี้จะทำการลดขนาดให้เหลือ 400 x 800 Pixels ซึ่งเป็นขนาดที่ง่ายต่อการวิเคราะห์ และใช้เวลาในการประมวลผลน้อย

F = imresize(f,[400 800])

F คือ ตัวแปรสำหรับการเก็บข้อมูลรูปภาพ

imresize คือ คำสั่งในการปรับเพิ่มหรือลดขนาดของรูปภาพ โดยเป็นการปรับความละเอียดของรูปภาพ เพื่อลดขนาดไฟล์ของรูปภาพลง

f คือ ข้อมูลรูปภาพที่จะนำมาปรับขนาดโดยคำสั่ง imresize โดยสามารถกำหนดขนาดรูปภาพใหม่ที่ต้องการเป็นเชิงตัวเลข

3.1.3. คำสั่ง rgb2gray

รูปภาพที่รับเข้ามาจากกล้อง นอกจากไฟล์ของรูปภาพจะมีขนาดใหญ่เนื่องจากความละเอียดของรูปภาพแล้ว สีของรูปภาพที่มาก (RGB) จะส่งผลให้ขนาดไฟล์ของรูปภาพมีขนาดใหญ่กว่าภาพสีเทา (Gray Scale) ที่มีสีของรูปภาพน้อยอีกด้วย จึงทำการปรับสีของรูปภาพจากภาพสี (RGB) ให้เป็นภาพสีเทา (Gray Scale) เพื่อให้ได้รูปภาพที่มีขนาด Pixel ตั้งแต่ 0-255 เนื่องจากภาพเทาเป็นรูปภาพที่มีขนาด 8 bit โดยใช้คำสั่ง Rgb2gray ซึ่งมีรูปแบบคำสั่ง “g=rgb2gray(f)” ดังรูปที่ 3.1

G = rgb2gray(f)

G คือ ตัวแปรสำหรับเก็บข้อมูลรูปภาพหลังจากการปรับสีโดยคำสั่ง rgb2gray

rgb2gray คือ คำสั่งในการเปลี่ยน Pixel ของรูปภาพ โดยจากรูปภาพสีให้เป็นรูปภาพเทา

f คือ ข้อมูลรูปภาพที่ต้องการจะนำมาปรับสีโดยคำสั่ง rgb2gray

รูปที่ 3.1 ตัวอย่างการใช้คำสั่ง rgb2gray

3.1.4 คำสั่ง medfilt2

ภาพสีเทาที่ได้จากการปรับสีนั้นยังปรากฏสัญญาณรบกวนอยู่ในภาพ ดังนั้นจึงต้องกำจัดสัญญาณรบกวนออกก่อน เพื่อให้ภาพที่ได้มีความคมชัดมากขึ้น ง่ายต่อการประมวลผลและวิเคราะห์ โดยการกำจัดสัญญาณรบกวนในภาพจะใช้ตัวกรองแบบ Median Filter เนื่องจากตัวกรองนี้มีความละเอียดในการกำจัดสัญญาณรบกวนในภาพมากกว่าตัวกรองอื่นๆ เช่น Mean Filter เป็นต้น โดยใช้คำสั่ง medfilt2 ดังรูปที่ 3.2

G = medfilt2(f)

G คือ ตัวแปรสำหรับเก็บข้อมูลรูปภาพหลังจากการกำจัดสัญญาณรบกวนโดยคำสั่ง medfilt2

medfilt2 คือ คำสั่งในการกำจัดสัญญาณรบกวนของรูปภาพ โดยใช้ตัวกรองแบบ Median Filter เพื่อกำจัดสัญญาณรบกวนที่ได้จากการถ่ายรูปของกล้อง

f คือ ข้อมูลรูปภาพที่ต้องการนำมากำจัดสัญญาณรบกวนโดยคำสั่ง medfilt2

รูปที่ 3.2 ตัวอย่างการใช้คำสั่ง medfilt2

3.1.5 คำสั่ง imdilate

หาขอบของตัวอักษรและตัวเลขจาก Pixel ในรูปภาพ โดยเปลี่ยนค่าของ Pixel ซึ่งขั้นตอนนี้โปรแกรมจะทำการเลือก Pixel ที่มีจำนวนเลขรอบข้างที่มากที่สุดเพื่อหาขอบของตัวอักษรและตัวเลข โดยใช้คำสั่ง imdilate ดังรูปที่ 3.3

Gi = imdilate(g)

Gi คือ ตัวแปรสำหรับเก็บค่าข้อมูลรูปภาพ

imdilate คือ คำสั่งในการเปลี่ยนค่าของ Pixel โดยโปรแกรมจะเลือก Pixel ที่มีจำนวนเลขรอบข้างที่มากที่สุดเพื่อหาขอบของตัวอักษรและตัวเลข

g คือ ข้อมูลรูปภาพที่ต้องการใช้คำสั่ง imdilate

รูปที่ 3.3 ตัวอย่างการใช้คำสั่ง imdilate

3.1.6 คำสั่ง imerode

หาขอบของตัวอักษรและตัวเลขจาก Pixel ในรูปภาพ โดยเปลี่ยนค่าของ Pixel ซึ่งขั้นตอนนี้โปรแกรมจะทำการเลือก Pixel ที่มีจำนวนเลขรอบข้างที่น้อยที่สุดเพื่อหาขอบของตัวอักษรและตัวเลข โดยใช้คำสั่ง Imerode ดังรูปที่ 3.4

Ge = imerode(g)

Ge คือ ตัวแปรสำหรับการรับข้อมูลรูปภาพหลังใช้คำสั่ง imerode

imerode คือ คำสั่งในการเปลี่ยนค่าของ Pixel โดยโปรแกรมจะเลือก Pixel ที่มีจำนวนเลขรอบข้างที่น้อยที่สุดเพื่อหาขอบของตัวอักษรและตัวเลข

g คือ ข้อมูลรูปภาพที่ต้องการใช้คำสั่ง imerode

รูปที่ 3.4 ตัวอย่างการใช้คำสั่ง imerode

3.1.7 คำสั่ง imsubstract

เพื่อให้ได้รูปภาพที่มีขอบของตัวอักษรและตัวเลขที่ชัดเจนขึ้น นำรูปภาพที่ได้จากขั้นตอนที่ 5 (ภาพที่ได้จากการ imdilate) มาลบกับรูปภาพที่ได้จากขั้นตอนที่ 6 (ภาพที่ได้จากการ imerode) โดยใช้คำสั่ง imsubstract ดังรูปที่ 3.5

รูปที่ 3.5 ตัวอย่างการใช้คำสั่ง imsubstract

Gdiff = imsubtract(gi,ge)

Gdiff คือ ตัวแปรสำหรับเก็บข้อมูลรูปภาพหลังใช้คำสั่ง imsubtract

imsubtract คือ คำสั่งในการนำค่า Pixel ของรูปภาพหนึ่ง ลบออกด้วยค่า Pixel ของอีกรูปภาพหนึ่งในตำแหน่งที่ตรงกัน ซึ่งในที่นี้กระทำเพื่อหาขอบของตัวอักษรและตัวเลข

gi คือ ข้อมูลรูปภาพที่จะนำมาใช้คำสั่ง imsubtract

ge คือ ข้อมูลรูปภาพที่จะนำมาใช้คำสั่ง imsubtract

3.1.8 คำสั่ง imadjust

รูปภาพได้จากการหาขอบของตัวอักษรและตัวเลขมีขนาด Pixel ตั้งแต่ 0-253 ซึ่งมีขนาดลดลงจากเดิมที่มีขนาด Pixel ตั้งแต่ 0-255 ดังนั้นจึงต้องมีการปรับขนาดของ Pixel ให้มีขนาดเท่ากับขนาดเดิม โดยเพิ่มความหนาแน่นของสีในรูปภาพให้มีความคมชัดมากยิ่งขึ้น โดยใช้คำสั่ง imadjust ดังรูปที่ 3.6

Gdiff = imadjust(gdiff)

Gdiff คือ ตัวแปรสำหรับเก็บค่าข้อมูลรูปภาพหลังใช้คำสั่ง imadjust

imadjust คือ คำสั่งในการเพิ่มความหนาแน่นของสีในรูปภาพ เพื่อให้รูปภาพมีความคมชัดมากขึ้น

gdiff คือ ข้อมูลรูปภาพที่ต้องการใช้คำสั่ง imadjust

รูปที่ 3.6 ตัวอย่างการใช้คำสั่ง imadjust

3.1.9 คำสั่ง imfill

รูปภาพที่ได้จากการหาขอบของตัวอักษรและตัวเลขอีกทั้งการเพิ่มความหนาแน่นของสีนั้นยังเป็นรูปภาพที่ไม่ตรงกับตัวอักษรและตัวเลขในฐานข้อมูล ดังนั้นจึงต้องทำการเปลี่ยน Pixel ที่อยู่ภายในวงปิดของตัวอักษรและตัวเลขให้กลายเป็น Pixel 1 (สีขาว) ทั้งหมด เพื่อให้ตัวอักษรและตัวเลขเป็นเนื้อเดียวกันซึ่งตรงกับตัวอักษรและตัวเลขในฐานข้อมูลที่มีอยู่ โดยใช้คำสั่ง imfill ดังรูปที่ 3.7

F = imfill(gdiff,'holes')

F คือ ตัวแปรสำหรับการเก็บข้อมูลรูปภาพหลังจากใช้คำสั่ง imfill

imfill คือ คำสั่งในการเปลี่ยน Pixel ที่อยู่ภายในวงปิดของตัวอักษรและตัวเลขให้กลายเป็น Pixel 1 (สีขาว) ทั้งหมด เพื่อให้ตัวอักษรและตัวเลขเป็นเนื้อเดียวกันซึ่งตรงกับตัวอักษรและตัวเลขในฐานข้อมูลที่มีอยู่

Gdiff คือ ข้อมูลรูปภาพที่ต้องการใช้คำสั่ง imfill

holes คือ การระบุรูปแบบของคำสั่ง imfill

รูปที่ 3.7 ตัวอย่างการใช้คำสั่ง imfill

3.1.10 คำสั่ง bwareaopen

กำจัดส่วนที่ไม่ต้องการออกจากภาพป้ายทะเบียน โดยขั้นตอนนี้มีจุดประสงค์เพื่อลดโอกาสในการอ่านป้ายทะเบียนที่ผิดพลาด เนื่องจากมีสัญญาณรบกวนจากร่องรอยต่างๆ ที่เกิดขึ้น เช่น รอยน็อต และอักษรข้อความอื่นๆ ที่ไม่ใช้ป้ายทะเบียน จากขั้นตอนก่อนนี้จะทำให้ได้รูปทะเบียนดังรูปที่ 3.8 ซึ่งจะทำการลบส่วนที่ไม่ต้องการคือ ข้อความและร่องรอยต่างๆ ที่ไม่ใช่เลขทะเบียน โดยจะใช้คำสั่งจาก Library คือ bwareaopen โดยคำสั่งนี้จะทำการลบกลุ่มของ Pixel ที่มีขนาดน้อยกว่าค่า Pixel ที่กำหนด ดังนั้นจึงกำหนดค่า Pixel ให้มีขนาดใกล้เคียงกับเลขทะเบียนรถยนต์ จึงทำให้กลุ่ม Pixel ที่มีขนาดเล็กกว่าขนาดของอักษรป้ายทะเบียนถูกกำจัดออกไป จึงได้เป็นดังรูปที่ 3.9 และมีการเขียนคำสั่งการกำจัดส่วนที่ไม่ต้องการออกจากภาพนี้คือ

Output = bwareaopen(Input, Pixel);

Output คือ ตัวแปรสำหรับเก็บข้อมูลรูปภาพหลังจากใช้คำสั่ง bwareaopen

bwareaopen คือ ชื่อฟังก์ชันที่ใช้ในการลบสิ่งรบกวนออกจากป้ายทะเบียน

Input คือ ข้อมูลรูปภาพที่ต้องการทำการลบสิ่งรบกวนออกจากภาพ

Pixel คือ ค่าขนาดของ Pixel ของกลุ่มรูปภาพย่อยๆ ในข้อมูลรูปภาพที่จะไม่ถูกลบ โดยกำหนดเป็นค่าเชิงตัวเลข (หากไม่กำหนดจะมีค่า Default = 50)

รูปที่ 3.8 ป้ายทะเบียนก่อนใช้คำสั่ง bwareaopen

รูปที่ 3.9 ป้ายทะเบียนหลังใช้คำสั่ง bwareaopen

3.2 การดึงข้อมูลตัวอักษรจากรูปภาพ

3.2.1 การวัดคุณลักษณะรูปภาพเพื่อแยกตัวอักษร

ในขั้นตอนนี้จะทำการดึงอักษรแต่ละตัวออกมาจากรูปภาพเพื่อทำการประมวลผล โดยจะใช้คำสั่ง regionprops เพื่อเรียกใช้การวัดคุณลักษณะ เช่น การหาจุด Centroid การเติมส่วนที่หายไปของกลุ่ม Pixel หรือการหาพื้นที่ เป็นต้น ซึ่งตัวอย่างคำสั่งการวัดคุณลักษณะเพิ่มเติมนั้น จะแสดงได้ดังรูปที่ 3.10 โดยในกรณีนี้จะเรียกใช้การวัดคุณลักษณะที่มีชื่อว่า BoundingBox ซึ่งจะทำการตรวจสอบกลุ่ม Pixel ที่อยู่ในรูปและทำการแบ่งเพื่อแยกตัวอักษร โดยสามารถแสดงลักษณะการแบ่ง

ได้ดังรูปที่ 3.11 และใช้การวัดคุณลักษณะอีกอย่างที่เรียกว่า Image เพื่อทำการคืนค่ารูปภาพที่ถูกแยกนั้นให้เป็นรูปภาพแบบ Binary Image และจะนำรูปภาพตัวอักษรที่ถูกแยกเหล่านี้ (Box) ไปเก็บไว้ในตัวแปรชื่อ boxes จากนั้นจะนำไปเปลี่ยนจากข้อมูลรูปภาพให้เป็นข้อมูลเชิงอักขระในขั้นตอนต่อไป

boxes = regionprops(input,'BoundingBox','Image')

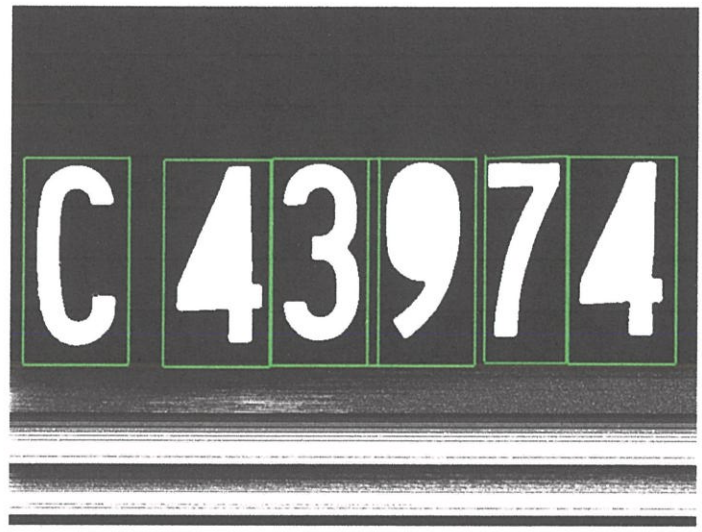
boxes คือ ตัวแปรที่รับข้อมูลรูปภาพหลังจากการวัดคุณลักษณะและรูปแบบของภาพ

regionprops คือ ชื่อของฟังก์ชั่นที่เรียกใช้เพื่อวัดคุณลักษณะและรูปแบบของภาพ

input คือ ข้อมูลรูปภาพที่ต้องการจะวัดคุณลักษณะและรูปแบบของภาพ

Property Name	Description	N-D Support	GPU Support	Code Generation
`'Area'`	Actual number of pixels in the region, returned as a scalar. (This value might differ slightly from the value returned by `bwarea`, which weights different patterns of pixels differently.) To find the equivalent to the area of a 3-D volume, use the `'Volume'` property of `regionprops3`.	Yes	Yes	Yes
`'BoundingBox'`	Smallest rectangle containing the region, returned as a 1-by-`Q*2` vector, where `Q` is the number of image dimensions. For example, in the vector `[ul_corner width]`, `ul_corner` specifies the upper-left corner of the bounding box in the form `[x y z ...]`. `width` specifies the width of the bounding box along each dimension in the form `[x_width y_width ...]`. `regionprops` uses `ndims` to get the dimensions of label matrix or binary image, `ndims(L)`, and `numel` to get the dimensions of connected components, `numel(CC.ImageSize)`.	Yes	Yes	Yes
`'Centroid'`	Center of mass of the region, returned as a 1-by-Q vector. The first element of `Centroid` is the horizontal coordinate (or *x*-coordinate) of the center of mass. The second element is the vertical coordinate (or *y*-coordinate). All other elements of `Centroid` are in order of dimension. This figure illustrates the centroid and bounding box for a discontiguous region. The region consists of the white pixels; the green box is the bounding box, and the red dot is the centroid.	Yes	Yes	Yes

รูปที่ 3.10 รูปตัวอย่างการวัดคุณสมบัติรูปภาพ



รูปที่ 3.11 การแบ่งตัวอักษรโดยการวัดคุณลักษณะ BoundingBox

3.3 การเปลี่ยนข้อมูลเชิงรูปภาพให้เป็นเชิงอักขระ

3.3.1 การนำข้อมูลตัวอักษรมาเทียบกับฐานข้อมูล

เมื่อได้ทำการแยกอักษรแต่ละตัวแล้ว จะทำการนำแต่ละตัวอักษร (Box) ที่อยู่ในตัวแปร Boxes มาทำการเปรียบเทียบกับฐานข้อมูล (Data Base) ที่มี ซึ่งฐานข้อมูลนั้นเป็นข้อมูลที่อยู่ในรูปแบบ Binary Image ดังรูปที่ 3.12 จากนั้นจึงเรียกใช้คำสั่ง corr2 (Correlation) ซึ่งจะทำการตรวจสอบความสัมพันธ์ของรูปตัวอักษรที่ได้นำมาทำการเปรียบเทียบกับฐานข้อมูล โดยคำสั่ง corr2 จะคืนค่าความสัมพันธ์ของตัวอักษรแต่ละตัวที่ถูกเปรียบเทียบกับฐานข้อมูล และถูกเก็บไว้อยู่ในตัวแปร sem

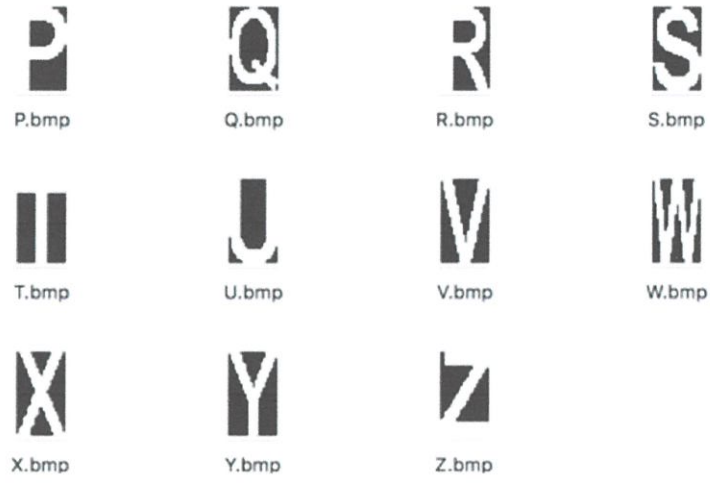
sem=corr2(Templates,box)

sem คือ ตัวแปรสำหรับเก็บค่าความสัมพันธ์ของการเปรียบเทียบตัวอักษร

corr2 คือ ฟังก์ชั่นที่ใช้ในการเปรียบเทียบความสัมพันธ์ของตัวอักษรกับรูปภาพฐานข้อมูล

Templates คือ ตัวแปรสำหรับเก็บฐานข้อมูลรูปภาพทั้งหมดที่ใช้ในการเปรียบเทียบ

Box คือ รูปภาพตัวอักษรที่ได้จากรูปภาพป้ายทะเบียน



รูปที่ 3.12 ตัวอย่างรูปภาพในฐานข้อมูล

3.3.2 การหาค่าดัชนีเพื่อใช้เลือกตัวอักษร

จากขั้นตอน 3.3.1 จะนำค่าตัวแปร sem ที่ได้จากการเปรียบเทียบตัวอักษรที่มีค่าความสัมพันธ์กันในแต่ละตัวอักษร เพื่อนำมาเป็นค่าดัชนีในการเลือกตัวอักษรโดยการบันทึกทุกค่า sem ลงในตัวแปร comp จากนั้นจะทำการเลือกค่า comp ที่มีค่าความสัมพันธ์กันมากที่สุดเพื่อนำไปเป็นค่าดัชนี โดยค่า comp จะถูกจำกัดให้ค่าความสัมพันธ์ที่นำไปเป็นดัชนีมีค่าความถูกต้องไม่ต่ำกว่าร้อยละเจ็ดสิบ (comp >= 0.7) เพื่อป้องกันไม่ให้เกิดการอ่านป้ายทะเบียนที่มีโอกาสผิดพลาดสูง

vd=find(comp==max(comp))

vd คือ ตัวแปรสำหรับเก็บค่าดัชนี

find คือ ฟังก์ชันที่ใช้ในการหาตำแหน่งข้อมูล และคืนค่าออกมาเป็นดัชนีบอกตำแหน่ง

max คือ ฟังก์ชันที่ใช้เลือกค่าที่มากที่สุดจากข้อมูลจำนวนหนึ่ง

comp คือ ตัวแปรที่ใช้เก็บข้อมูลความสัมพันธ์ของการเทียบอักษรแต่ละตัว

3.3.3 การนำค่าดัชนีไปเปลี่ยนเป็นข้อมูลเชิงอักขระ

ในขั้นตอนนี้จะนำเอาค่าดัชนีที่ได้จากขั้นตอน 3.3.2 เพื่อนำมาเลือกค่าตัวอักษรจากฐานข้อมูล โดยดัชนีจะมีความสัมพันธ์กับตำแหน่งของรูปอักษรในฐานข้อมูล ซึ่งเขียนคำสั่งและสร้างเงื่อนไขได้ดังตัวอย่าง

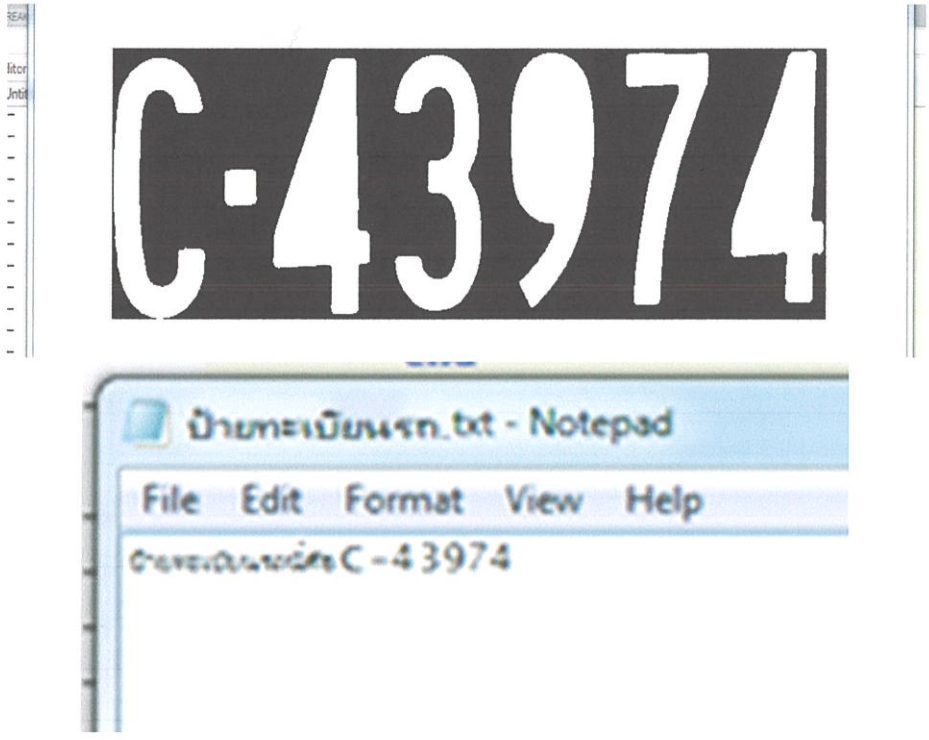
```
if vd==9
letter='F';
```

vd คือ ค่าดัชนีที่ใช้เลือกตำแหน่งตัวอักษร

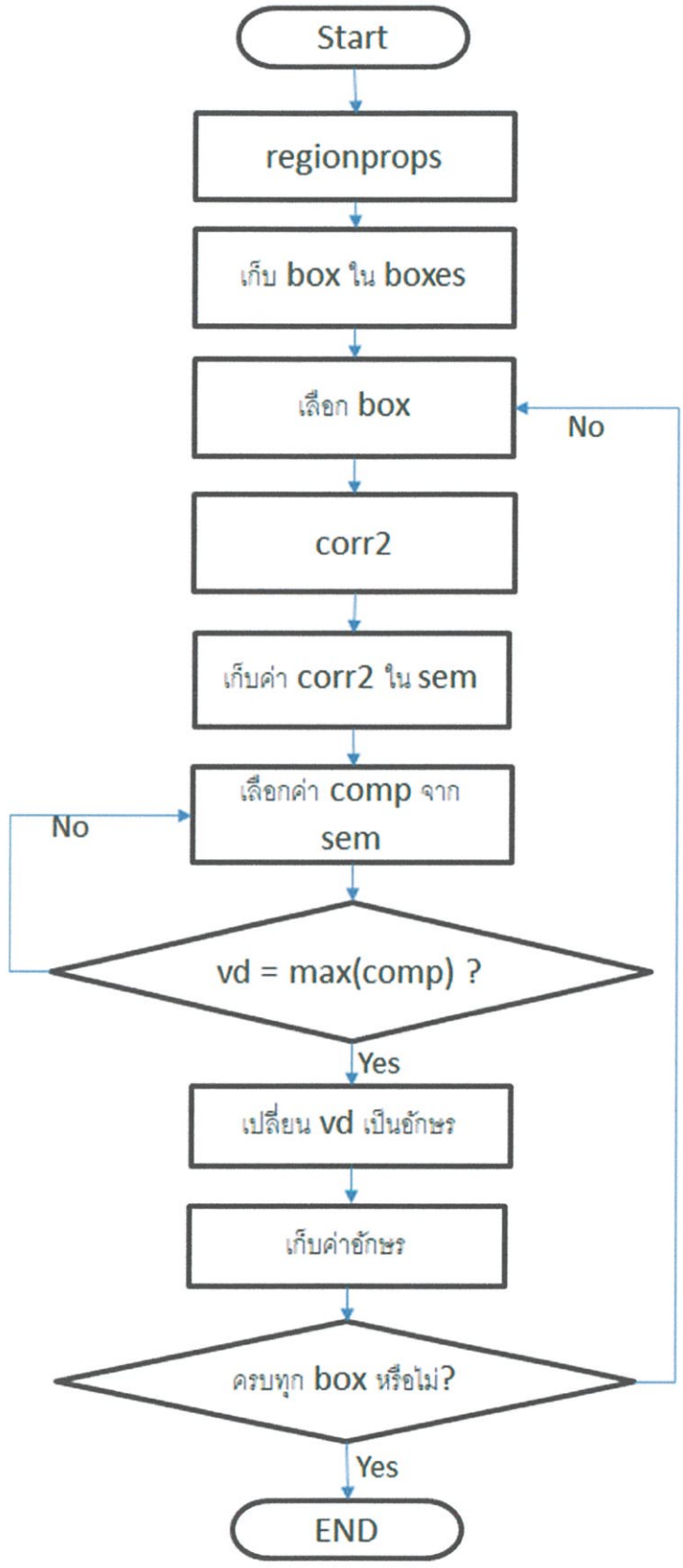
letter คือ ตัวแปรสำหรับเก็บค่าอักษรที่ดัชนีเลือก

3.3.4 การเก็บค่าตัวอักษรจากป้ายทะเบียน

เมื่อทำการเทียบตัวอักษรกับฐานข้อมูลครบทุกตัวแล้ว ชุดตัวอักษรจากป้ายทะเบียนจะถูกเก็บไว้ในตัวแปรใหม่ ซึ่งในขณะนี้ข้อมูลเชิงรูปภาพได้ถูกเปลี่ยนให้อยู่ในรูปข้อมูลแบบอักขระแล้ว ดังนั้นจึงเป็นการสิ้นสุดการทำการประมวลผลภาพ ดังรูปที่ 3.13 และสามารถนำข้อมูลเชิงอักขระนี้ไปประยุกต์ใช้ต่อได้ในส่วนถัดไป



รูปที่ 3.13 ผลการประมวลผลภาพ

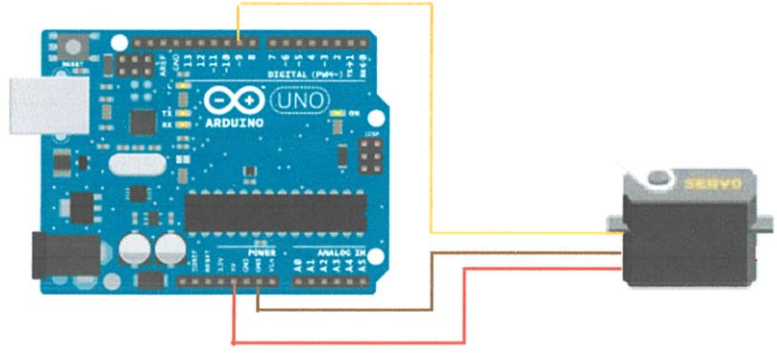


รูปที่ 3.14 Flow Chart การเปลี่ยน Binary Image ให้เป็นข้อมูลรูปแบบ Char

3.4 การต่อวงจร

3.4.1 การต่อวงจรของบอร์ด Arduino และ Servo Motor

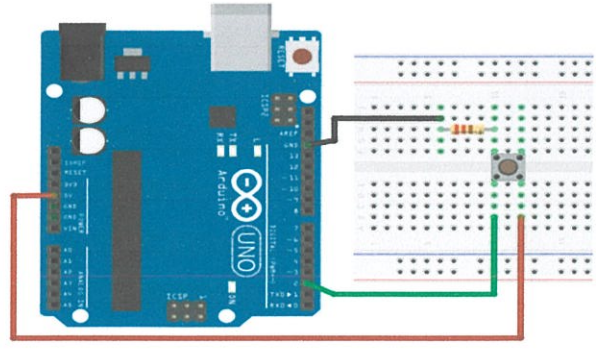
Servo Motor จะมีสายทั้งหมดสามสายคือ สายไฟ (สายสีแดง), สายดิน (สายสีน้ำตาล) และสายสัญญาณ (สายสีเหลืองหรือสีส้ม) ในส่วนของ Arduino Board จะมีสายส่งไฟฟ้า (5 V, 3.3 V) สายดิน (Ground) และ Port ต่างๆ ไว้ใช้ส่งสัญญาณ เริ่มแรกทำการต่อสายดิน (สายสีน้ำตาล) เข้ากับ GND (Ground) ต่อสายไฟ (สายสีแดง) เข้ากับช่องไฟ 5 V สุดท้ายต่อสายสัญญาณกับ Port ที่ 9 (สามารถใช้ได้ตั้งแต่ Port ที่ 1-13) ดังรูปที่ 3.15



รูปที่ 3.15 การต่อวงจรของบอร์ด Arduino และ Servo Motor

3.4.2 การต่อวงจรของบอร์ด Arduino กับสวิตช์

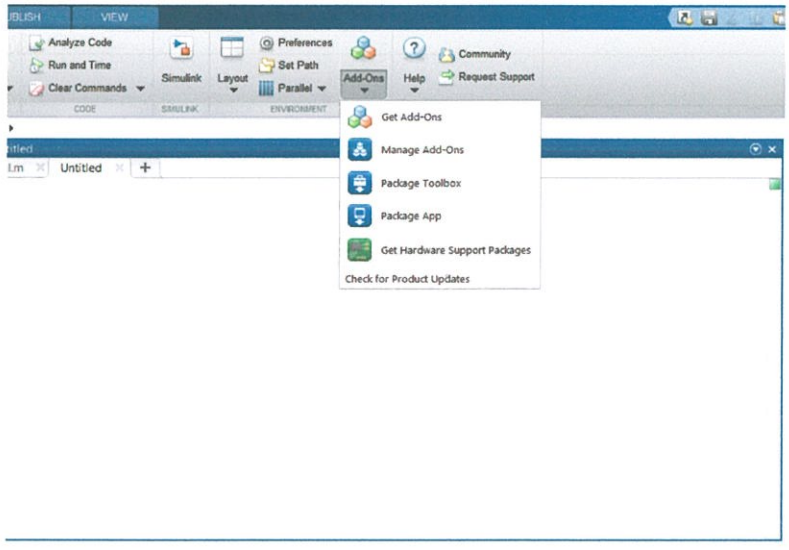
อุปกรณ์ที่ใช้ในการต่อคือ สายไฟจำนวน 3 เส้น, สวิตช์ปุ่มกดจำนวน 1 ตัวและตัวต้านทานขนาด 10K Ω จำนวน 1 ตัว เริ่มโดยนำสายไฟขนาด 5 V ต่อเข้าที่ขาของสวิตช์ข้างใดข้างหนึ่งหลังจากนั้นต่อสายสัญญาณจากข้างอีกข้างเข้าสู่บอร์ด Arduino ที่ Port 2 (สามารถใช้ Port ใดก็ได้ตั้งแต่ 1-13) จากขาด้านที่เข้าสายสัญญาณอีกฝั่งต่อเข้ากับตัวต้านทานขนาด 10K Ω และต่อเข้ากับ GND ของบอร์ด Arduino ดังรูปที่ 3.16



รูปที่ 3.16 การต่อวงจรของบอร์ด Arduino กับสวิตช์

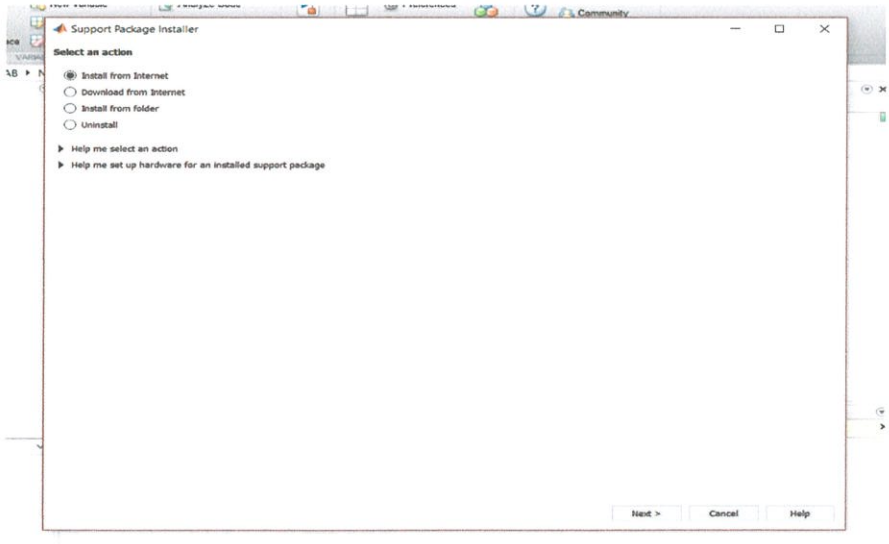
3.5 การเชื่อมต่อบอร์ด Arduino กับโปรแกรม MATLAB

1. เข้าไปที่ Add-Ons => Get Hardware Support Packages (เพื่อที่จะได้ลง Add-On สำหรับ Aduino) ดังรูปที่ 3.17



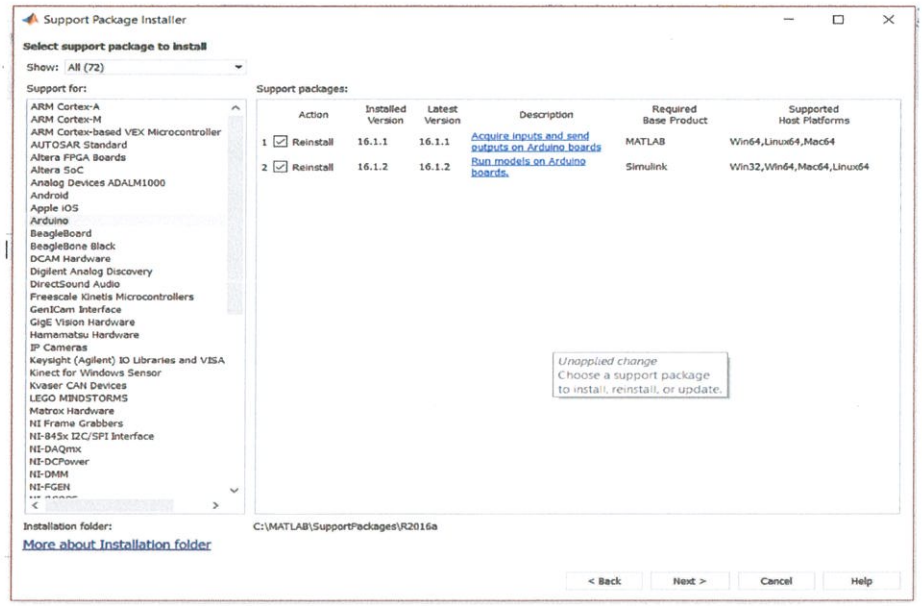
รูปที่ 3.17 ตัวอย่างการลง Get Hardware Support Packages

2. เลือก Install From Internet แล้วกด Next ดังรูปที่ 3.18



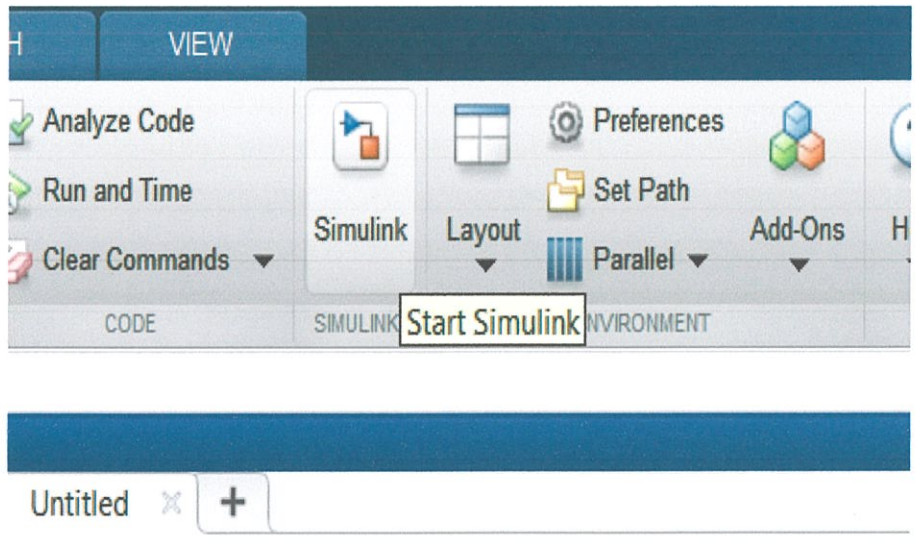
รูปที่ 3.18 ตัวอย่างการเลือก Install From Internet

3. เลือก Install ตามรูป กด Next ดังรูปที่ 3.19



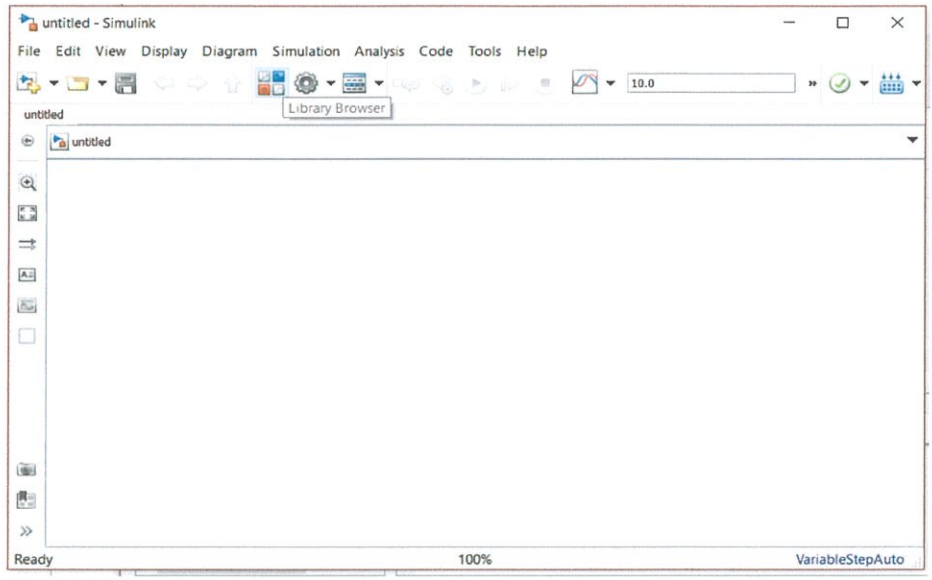
รูปที่ 3.19 ตัวอย่างการเลือก Install

4. จากนั้นกดเข้ามาที่ Simulink เพื่อเริ่มต้นการวาด Block Diagram ดังรูปที่ 3.20



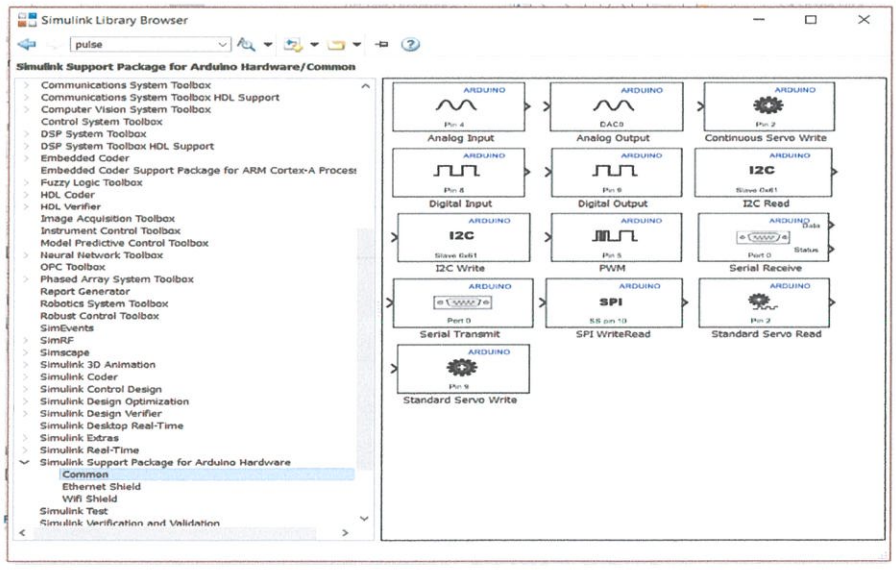
รูปที่ 3.20 ตัวอย่างการกดเริ่ม Simulink

5. กดเข้าไปที่ Library Browser เพื่อเลือก Block Diagram ดังรูปที่ 3.21



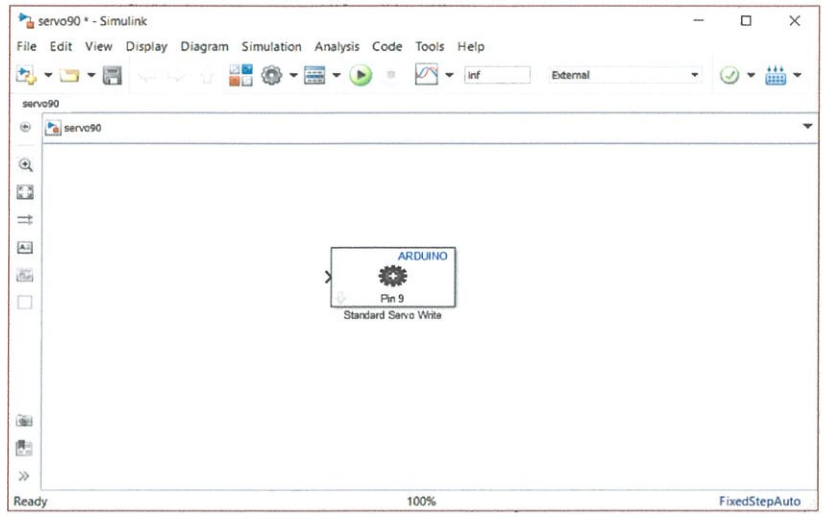
รูปที่ 3.21 ตัวอย่างการเปิด Library Browser ของ Simulink

6. เลือกที่คำสั่ง Simulink Support Package for Arduino Hardware => Common ดังรูปที่ 3.22 (Block เหล่านี้คือ Add-On ที่โหลดมาดังขั้นตอนก่อนหน้า เพื่อไว้ใช้กับ Arduino)



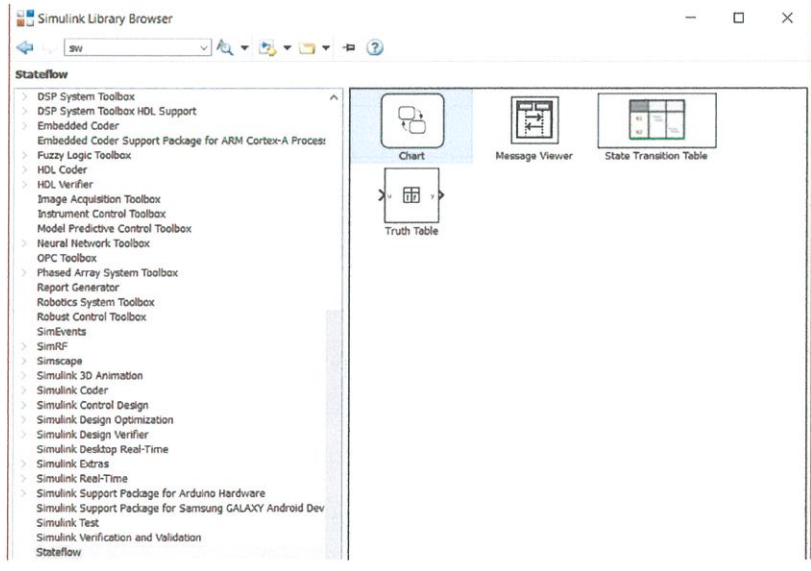
รูปที่ 3.22 ตัวอย่างการเลือก Simulink Library

7. กดที่ Standard Servo Write (เป็น Block ที่ ควบคุมว่า Servo Motor จะเคลื่อนที่ไปเป็นมุมเท่าใด) แล้วลากมาที่หน้าต่าง untitled–Simulink ดังรูปที่ 3.23

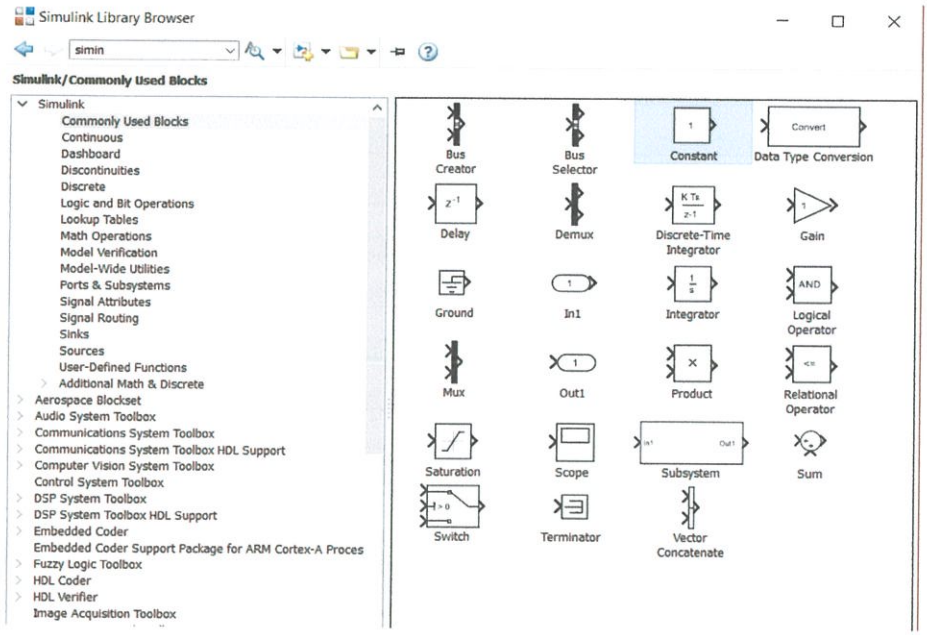


รูปที่ 3.23 ตัวอย่างเมื่อลาก Standard Servo Write ลงในหน้าต่าง untitled–Simulink

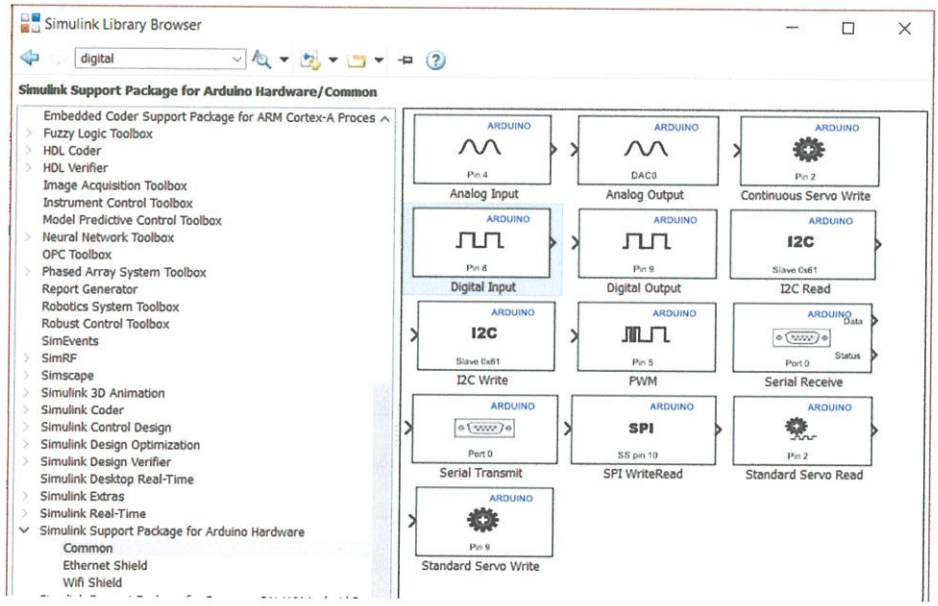
8. เลือก Chart (เป็น Block ที่ใช้ในการเขียน Loop เพื่อส่งค่าไปให้ Standard Servo Write จะแสดงวิธีการเขียน Loop ในขั้นตอนต่อไป), Digital Input (เป็น Block ที่ใช้รับค่า Digital จากสวิตช์), Constant (ใช้เมื่อ Simulink เริ่มทำงานจะส่งค่า 1 ให้กับ Chart ทันที เพื่อสั่งให้ Servo Motor เลื่อนไปที่มุม 90 องศา) มาวางที่หน้าต่าง Untitled–Simulink ดังรูปที่ 3.24, ดังรูปที่ 3.25, ดังรูที่ 3.26 และนำทุก Block มาเชื่อมกันดังรูปที่ 3.27



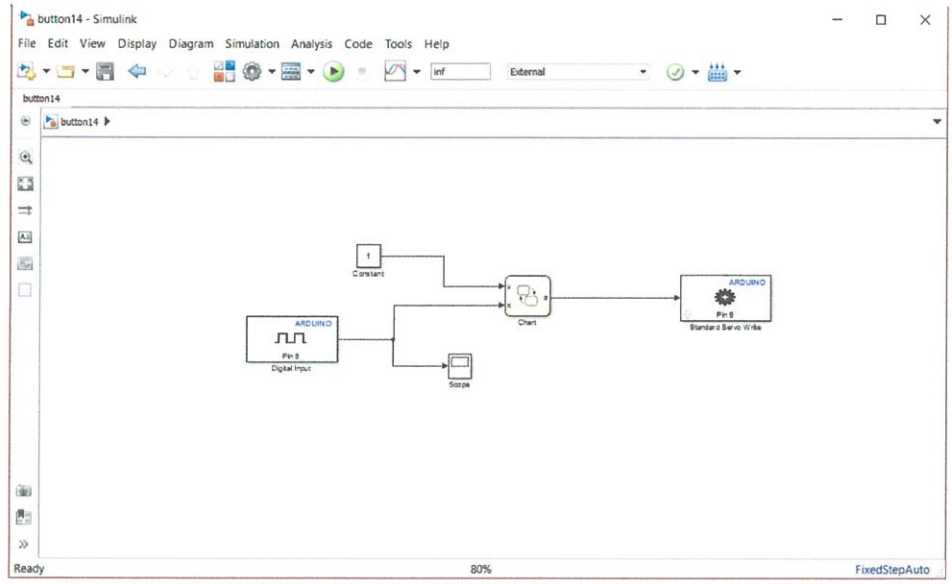
รูปที่ 3.24 ตัวอย่างการเลือก Chart



รูปที่ 3.25 ตัวอย่างการเลือก Constant

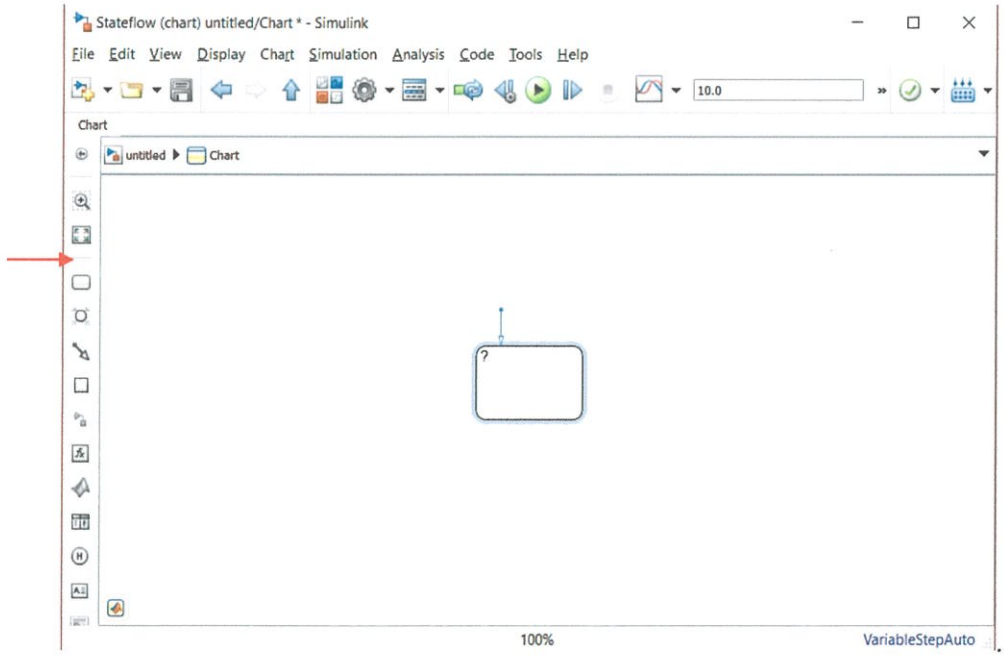


รูปที่ 3.26 ตัวอย่างการเลือก Digital Input



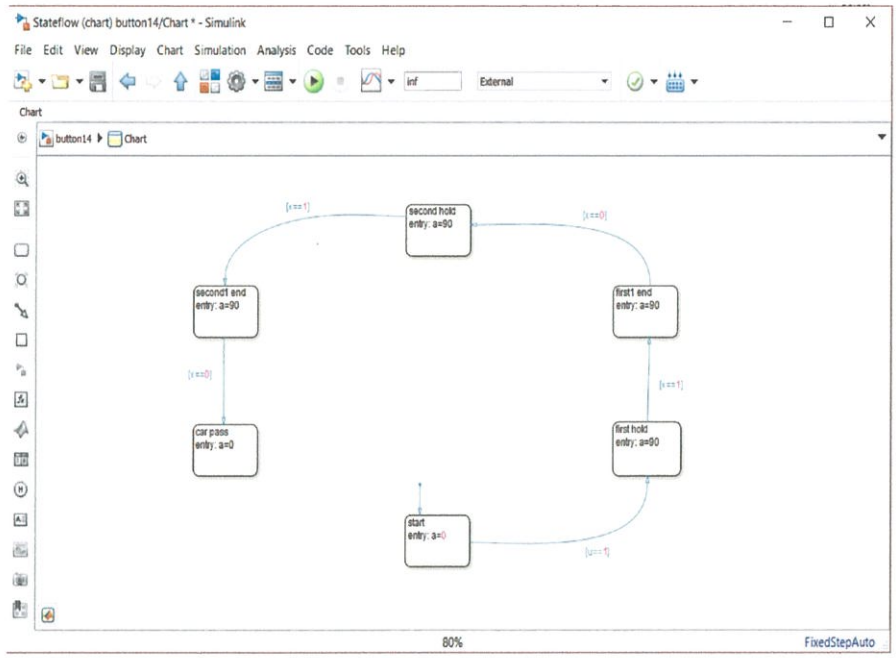
รูปที่ 3.27 ตัวอย่างหลังจากโยงเส้นเชื่อมกัน

9. กด Double Click เข้ามาที่ Chart => เลือก State จาก Icon ที่มีลูกศรชี้ โดย State คือ Block ที่ไว้เขียนเงื่อนไขหรือคำสั่งต่างๆ จะมีทั้งค่า In-Put, Out-Put ใน Block เดียวกัน ค่า In-Put จะกำหนดให้เป็นตัวแปร x ซึ่งได้มาจากการกดสวิตช์ --> Digital In-Put และตัวแปร u ซึ่งได้มาจาก Block Constant ที่กำหนดค่าให้เป็น 1 ส่วนของ Out-Put จะกำหนดให้เป็นตัวแปร a ซึ่งจะส่งต่อไปให้ Standard Servo Write เพื่อบังคับมุมการหมุนของ Servo Motor ต่อไป ดังรูปที่ 3.28



รูปที่ 3.28 ตัวอย่างเมื่อกดเข้ามาใน Chart

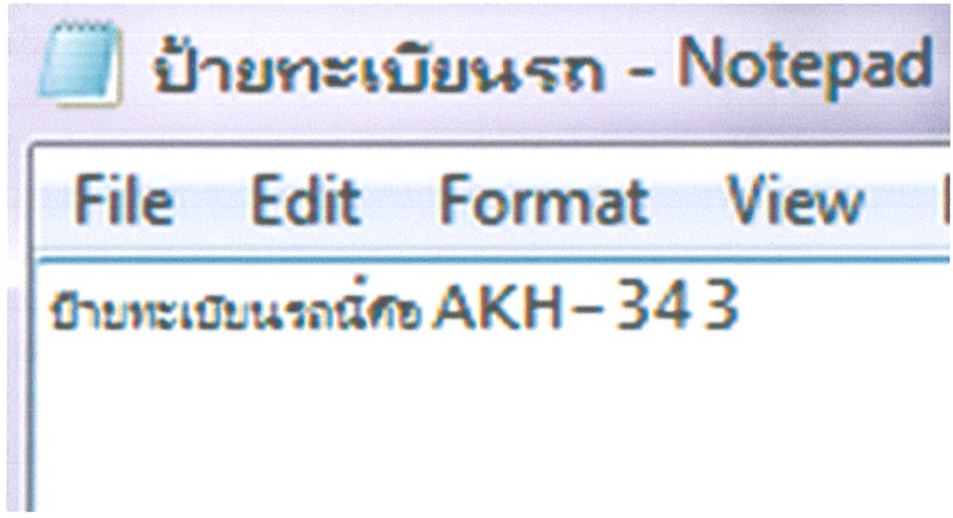
10. กำหนดเงื่อนไขตามรูปข้างล่าง โดยเริ่มจาก State ที่มีลูกศรชี้ บ่งบองว่า State นี้คือ State ที่เริ่มต้นการทำงาน ซึ่งให้ค่า a = 0 ซึ่ง Servo Motor จะอยู่ที่มุม 0 State ต่อไปตั้งชื่อว่า First Hold (สามารถตั้งชื่ออะไรก็ได้ตามต้องการ) จะสังเกตุได้ว่ามีลูกศรที่เชื่อมระหว่างสอง State เส้นนี้จะเป็นตัวกำหนดค่า In-Put ของ State First Hold (ให้ตัวแปรเป็นตัว u ซึ่งค่า u นี้ได้มาจาก Constant ที่กำหนดให้มีค่าเป็น 1 เมื่อเลื่อนไปสู่ State First Hold, Out-Put คือ Entry : a = 90 (ค่า a = 90 คือการสั่งให้มุมของ Servo Motor หมุมไปที่ตำแหน่ง 90 องศา) และเมื่อกดสวิตช์ 1 ครั้ง(แทนการที่ล้อหน้ารถยนต์ผ่านสวิตช์) State จะเปลี่ยนไปเป็น State ที่มีชื่อว่า First End และรูปแบบการทำงานจะเป็นอย่างนี้ต่อไปจนกระทั้งจบที่ State ที่มีชื่อว่า Car Pass ดังรูปที่ 3.29



รูปที่ 3.29 ตัวอย่างการเขียนเงื่อนไขลงใน Chart

3.6 การใช้ MATLAB และ Simulink เพื่อควบคุมเซอร์โวมอเตอร์

หลังจากที่ได้ทะเบียนรถ ดังรูปที่ 3.30 โปรแกรมจะทำการตรวจสอบข้อมูลว่าตรงตามฐานข้อมูลหรือไม่ ถ้าทะเบียนนั้นอยู่ในฐานข้อมูลโปรแกรมจะสั่งให้ Simulink เริ่มทำงานหลังจากนั้นเซอร์โวมอเตอร์จะเลื่อนขึ้นไปอยู่ที่ตำแหน่ง 90 องศา และเมื่อรถยนต์ได้ผ่านไปเซอร์โวมอเตอร์จะเลื่อนกลับมาอยู่ที่ตำแหน่ง 90 องศาดังเดิม

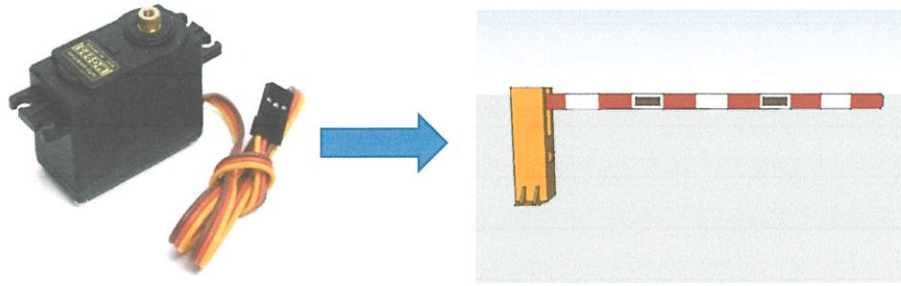


รูปที่ 3.30 ผลการทดลองอ่านป้ายทะเบียน

3.7 แนวคิดด้านการออกแบบ

รูปร่างตัวอย่างของปริญญานิพนธ์นี้จะเป็นประตูกั้นทาง ดังรูปที่ 3.31 ซึ่งจะใช้เซอร์โวมอเตอร์จำลองและมีลักษณะดังรูปที่ 3.32

รูปที่ 3.31 ประตูกั้นทาง



รูปที่ 3.32 เซอร์โวมอเตอร์และรูปตัวอย่างประตูกั้นทางจำลอง

3.8 แบบจำลอง

จากที่ได้กล่าวมา รูปร่างตัวอย่างจะเป็นดังรูปที่ 3.33 แบบจำลองตึกด้านหลังจะเป็นอาคารตัวอย่าง กล่องสีขาวแทนป้อมยามและประตูกั้นทาง เพื่อจำลองในกรณีศึกษาที่อาคารในเมืองอัจฉริยะมีระบบรักษาความปลอดภัยของการอนุญาตให้รถยนต์ที่จะเข้ามาภายในพื้นที่ ต้องเป็นรถยนต์ที่มีป้ายทะเบียนที่ได้รับอนุญาตเท่านั้น โดยในกล่องสีขาวจะประกอบไปด้วย เซอร์โวมอเตอร์, บอร์ด Arduino และแผงวงจรต่างๆ

รูปที่ 3.33 แบบจำลองประตูกั้นทาง

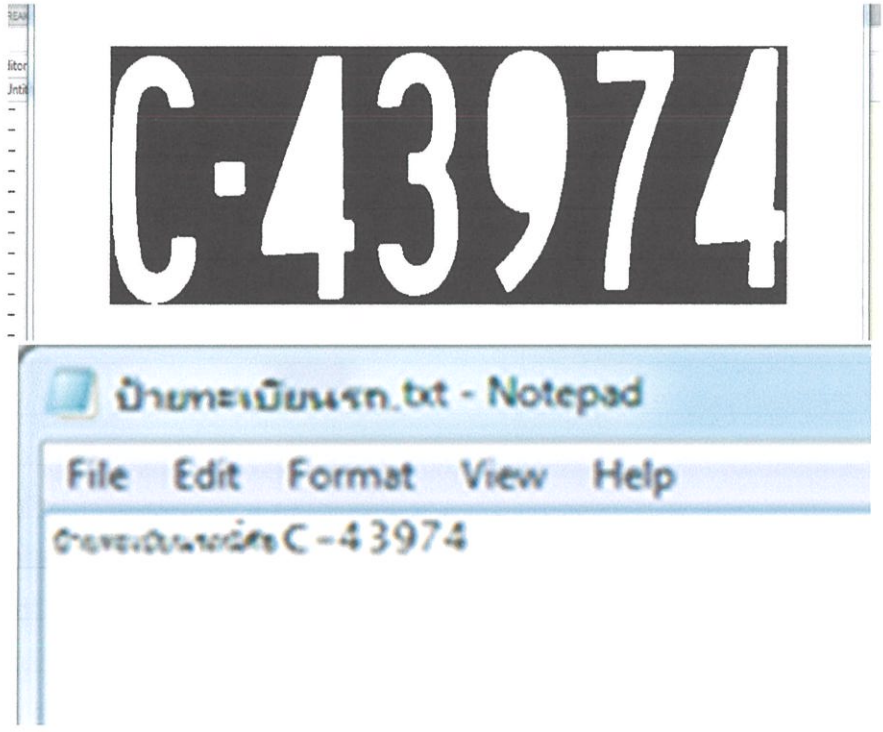
บทที่ 4
ผลการดำเนินงาน

4.1 ผลการทดลองการอ่านป้ายทะเบียน

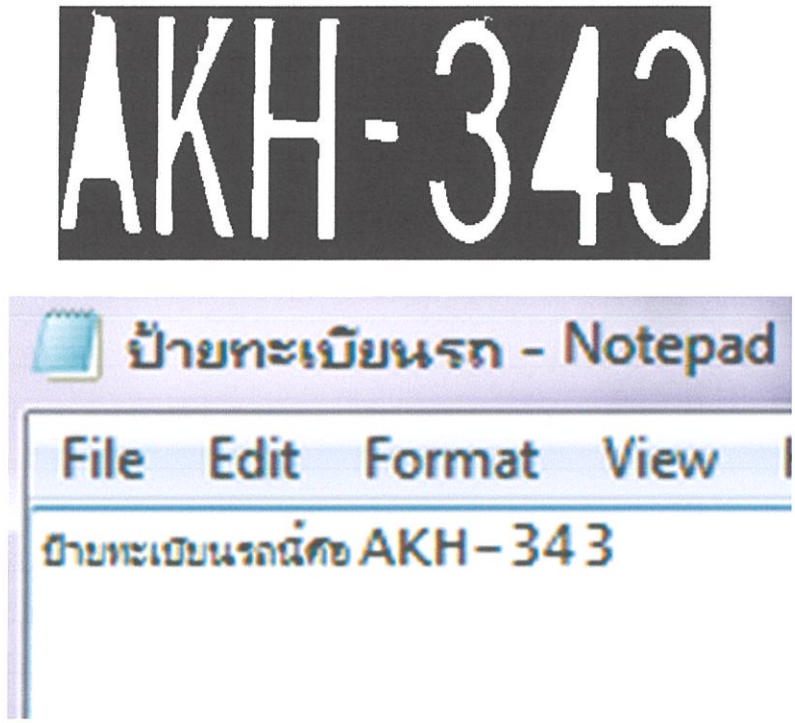
การทดลองโปรแกรมการอ่านป้ายทะเบียนจะใช้ป้ายทะเบียนตัวอย่างทั้งหมด 6 รูป แบ่งเป็นป้ายทะเบียนที่สมบูรณ์ 3 รูป และไม่สมบูรณ์ 3 รูป

4.1.1 ผลการทดลองการอ่านป้ายทะเบียนที่สมบูรณ์

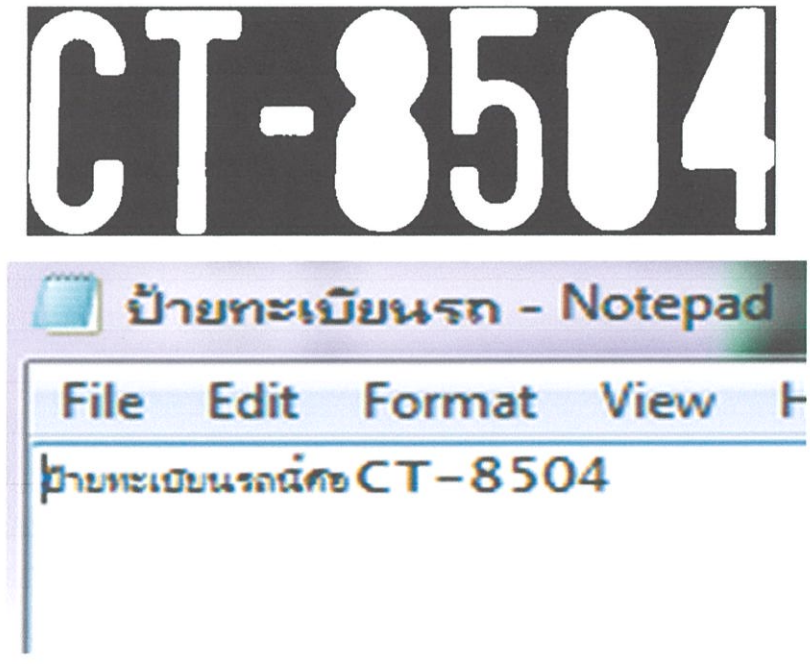
จากการอ่านป้ายทะเบียนที่มีความสมบูรณ์แบบ ทำให้พบว่าโปรแกรมอ่านป้ายทะเบียนรถยนต์นั้นสามารถอ่านเลขทะเบียนที่มีความสมบูรณ์ทั้ง 3 รูป ได้อย่างแม่นยำและถูกต้องดังรูปที่ 4.1, รูปที่ 4.2 และรูปที่ 4.3



รูปที่ 4.1 ผลการทดลองการอ่านป้ายทะเบียน



รูปที่ 4.2 ผลการทดลองการอ่านป้ายทะเบียน



รูปที่ 4.3 ผลการทดลองการอ่านป้ายทะเบียน

4.1.2 ผลการทดลองอ่านป้ายทะเบียนที่ไม่สมบูรณ์

การอ่านป้ายทะเบียนที่ไม่สมบูรณ์อาจเกิดขึ้นได้ในกรณีตัวอย่างเช่น ป้ายทะเบียนชำรุด มีรอยขีดข่วน หรือมีรอยที่มาทำให้เกิดความไม่ชัดเจน ซึ่งอาจทำให้การอ่านป้ายทะเบียนมีความผิดพลาดเกิดขึ้นได้ จึงได้มีการเขียนคำสั่งเพื่อลดโอกาสความผิดพลาดดังกล่าว และจากการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์นั้นพบว่า การอ่านป้ายทะเบียนบางครั้งมีความผิดพลาดเกิดขึ้นดังรูปที่ 4.4, รูปที่ 4.5 รูปที่ 4.6

รูปที่ 4.4 ผลการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์

รูปที่ 4.5 ผลการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์

C – 4--74

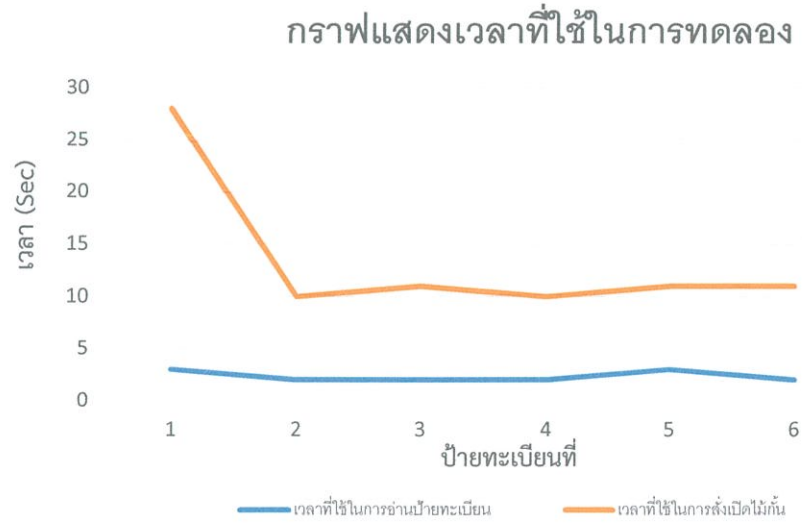
รูปที่ 4.6 ผลการทดลองการอ่านป้ายทะเบียนที่ไม่สมบูรณ์

4.1.3 สรุปผลการทดลองการอ่านป้ายทะเบียน

จากการทดลองพบว่า โปรแกรมการอ่านป้ายทะเบียนรถยนต์นี้ มีข้อจำกัดการใช้งานในการอ่านป้ายทะเบียนเฉพาะแบบที่มีความชัดเจน และมีความโดดเด่นในรูปภาพที่จะนำมาประมวลผล ดังนั้นจึงจำเป็นจะต้องมีการจำกัด และควบคุมตัวแปรสำหรับรูปภาพที่จะนำมาประมวผล เช่น ความคมชัด ความสว่าง มุมองศาและความโดดเด่นของป้ายทะเบียน ทำให้จำเป็นต้องใช้รูปภาพป้ายทะเบียนที่ถูกตัดมาเพียงแผ่นป้ายทะเบียนเท่านั้น โดยการอ่านป้ายทะเบียนที่สมบูรณ์นั้นทำได้อย่างแม่นยำและถูกต้อง แต่สำหรับการอ่านป้ายทะเบียนที่ไม่สมบูรณ์นั้น โปรแกรมสามารถอ่านป้ายทะเบียนที่มีความคลาดเคลื่อนของตัวอักษรที่มีความคลาดเคลื่อนได้ไม่เกินร้อยละสามสิบ โดยจะคืนค่าอักษรที่มีความคล้ายกันมากที่สุด และได้แสดงเวลาที่ใช้ในการทดลองได้ดังตารางที่ 4.1 และรูปที่ 4.6 ซึ่งจะแสดงให้เห็นว่า การสั่งงานเพื่อควบคุมไม้กั้นผ่านโปรแกรม Simulink นั้นต้องใช้เวลาค่อนข้างนาน โดยเฉพาะครั้งแรกที่เริ่มทำการอ่านป้ายทะเบียนหลังจากเปิดโปรแกรม และการประมวลผลป้ายทะเบียนที่ไม่สมบูรณ์จะไม่ส่งผลต่อเวลาในการประมวลผล จึงสรุปการทำปริญญานิพนธ์นี้ได้ว่าจากข้อจำกัด และเวลาที่ใช้ในการประมวลผลการสั่งการต่างๆ ทำให้ยังไม่เหมาะสมในการนำไปใช้จริงในปัจจุบัน หรือกล่าวคือยังเป็นขั้นทดลองและต้องศึกษาวิจัยเพิ่มเติม

ตารางที่ 4.1 ผลการทดลองการอ่านป้ายทะเบียน

ครั้งที่	ทะเบียน	ผล	เวลาที่ใช้อ่าน (วินาที)	เวลาที่ใช้เปิดไม้กั้น (วินาที)
1		ถูกต้อง	3	28
2		ถูกต้อง	2	10
3		ถูกต้อง	2	11
4		ผิดพลาด	2	10
5		ถูกต้อง	3	11
6		ผิดพลาด	2	11



รูปที่ 4.7 กราฟแสดงเวลาที่ใช้ในการทดลอง

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการทดลองเขียนคำสั่งโปรแกรมสามารถสรุปได้ว่า โปรแกรมสามารถเรียกรูปภาพป้ายทะเบียนจากที่เก็บข้อมูลมาประมวลผลผ่านคอมพิวเตอร์ จนสามารถอ่านข้อความเลขทะเบียนจากป้ายทะเบียนรถยนต์ได้ จากนั้นจึงนำข้อมูลจากการประมวลผลป้ายทะเบียนนี้ส่งไปใช้ควบคุมเซอร์โวมอเตอร์ ให้ทำงานผ่านไมโครคอนโทรลเลอร์ซึ่งรับค่าสัญญาณมาจากโปรแกรม MATLAB ในคอมพิวเตอร์ และนำไปใช้ควบคุมการเปิด-ปิดไม้กั้นได้ ในส่วนของความผิดพลาดที่อาจจะเกิดขึ้นในการอ่านป้ายทะเบียน เช่น อักษรบนป้ายทะเบียนมีการถูกบิดเบือนนั้น โปรแกรมจะทำการพิจารณาค่าความสัมพันธ์ระหว่างตัวอักษรที่อ่านกับฐานข้อมูล ว่าหากมีค่าความแม่นยำต่ำกว่าร้อยละเจ็ดสิบ โปรแกรมจะไม่อ่านอักษรตัวนั้นและบันทึกเป็นอักษรขีด เพื่อป้องกันการอ่านป้ายทะเบียนที่ไม่ถูกต้องและเพื่อให้ผู้ที่เกี่ยวข้องเข้ามาแก้ปัญหาต่อไป

5.2 ข้อเสนอแนะ

เทคโนโลยีการประมวลผลภาพเป็นเรื่องที่ควรให้ความสำคัญและไม่ควรมองข้าม เพราะเทคโนโลยีในการประมวลผลภาพนั้น เป็นเทคโนโลยีที่สามารถนำไปประยุกต์ใช้กับเทคโนโลยีอื่นๆ และจะเข้ามามีบทบาทมากในอนาคต เพราะการรวมเทคโนโลยีการประมวลผลภาพเข้าร่วมกับเทคโนโลยีอื่นๆ ก็เปรียบเสมือนกับการเพิ่มความสามารถในด้านวิสัยทัศน์ในการมองเห็น และวิเคราะห์ภาพให้กับเทคโนโลยีนั้นๆ เช่น การติดตั้งกล้องและการประมวลผลภาพให้กับแขนกลในอุตสาหกรรมทั่วไป ทำให้แขนกลนั้นสามารถทำงานและเคลื่อนที่ไปยังจุดที่ต้องการได้อย่างอัตโนมัติโดยผ่านการวิเคราะห์ตำแหน่งจากเทคโนโลยีการประมวลผลภาพ เป็นต้น

เทคโนโลยีโปรแกรมการประมวลผลภาพนั้นสามารถศึกษา และเขียนได้โดยหลายภาษาเช่น Python, C++, JAVA เป็นต้น โดยในบางภาษาจะมีผู้พัฒนาเขียนฟังก์ชันและ Library ในการประมวลผลภาพ เพื่อความสะดวกในการเขียนโปรแกรมประมวลผลภาพ เช่น MATLAB เป็นต้น

การประมวลผลภาพเป็นกระบวนการที่ค่อนข้างใช้เวลา และทรัพยากรในการประมวลผลมาก ดังนั้นจึงควรเลือกใช้คอมพิวเตอร์เป็นตัวประมวลผล ซึ่งเป็นตัวเลือกที่ดีกว่าไมโครคอนโทรลเลอร์

5.3 ปัญหาในการดำเนินงาน

การประมวลผลภาพในปริญญานิพนธ์นี้มีข้อด้อยคือ รูปภาพที่นำมาประมวลผลนั้นจำเป็นต้องมีความชัดเจน และเลขในป้ายทะเบียนต้องมีความโดดเด่นมากกว่าสิ่งอื่น ซึ่งเป็นผลให้การทดลองและการทำปริญญานิพนธ์นี้จำเป็นต้องมีการควบคุมตัวแปร เช่น ความชัดเจนของรูปภาพ มุมของรูปภาพป้ายทะเบียน เป็นต้น จึงทำให้เกิดข้อจำกัดในการเลือกรูปภาพเพื่อประมวลผล โดยจำเป็นต้องลดขอบเขตของการทำปริญญานิพนธ์นี้คือ ยกเลิกการทำระบบกล้องถ่ายรูปเพื่อนำมาใช้ประมวลผลซึ่งสามารถควบคุมตัวแปรต่างๆ ในการทำปริญญานิพนธ์ได้ยาก จึงจำเป็นต้องนำรูปภาพป้ายทะเบียนที่ชัดเจนและได้มีการบันทึกไว้แล้วมาประมวลผลแทน

การประมวลผลภาพเป็นการประมวลผลที่ค่อนข้างใช้เวลาและทรัพยากรในการประมวลผลมาก ดังนั้นไมโครคอนโทรลเลอร์จึงไม่เหมาะจะนำมาใช้แทนคอมพิวเตอร์ในปริญญานิพนธ์นี้

เอกสารอ้างอิง

[1] **"Smart City"** (Online). Available :

https://www.applicadthai.com/editor-talks/smart-city

[2] **"Image Processing"** (Online). Available :

http://jaratcyberu.blogspot.com/2009/10/image-processing.html

[3] **"Pixel"** (Online). Available :

www.printtosme.com

[4] **"Resolution"** (Online). Available :

www.comgeeks.net/resolution

[5] **"RGB"** (Online). Available :

http://www.infinityprinting.co.th/main/content.php?page=sub&category

[6] **"MATLAB"** (Online). Available :

http://www.mindphp.com/-what-is-matlab.html

[7] **"Aduino"** (Online). Available :

https://poundxi.com/arduino

[8] **"Servo Motor"** (Online). Available :

http://chanaphinp.blogspot.com/