

เครื่องแสดงผลอักษรเบรลล์
Braille Display



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เครื่องแสดงผลอักษรเบรลล์

Braille Display



ฐิตีทรัพย์ ศรีวโล
Thitisub Sriwalo

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องแสดงผลอักษรเบรลล์

Braille Display

โดย

ฐิติทรัพย์ ศรีวโล



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2559
ภาควิชา วิศวกรรมอิเล็กทรอนิกส์
คณะ วิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง เครื่องแสดงผลอักษรเบรลล์
Braille Display
ผู้จัดทำ นางสาว ฐิติทรัพย์ ศรีวิไล รหัสนักศึกษา 56010336

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว


.....
(รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์)
อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องแสดงผลอักษรเบรลล์
นักศึกษา	นางสาว รุติทรัพย์ ศรีวิไล รหัสประจำตัว 56010336
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2559
อาจารย์ที่ปรึกษาโครงการ	รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการออกแบบ และสร้างเครื่องแสดงผลอักษรเบรลล์ ที่เขียนด้วยโปรแกรม Halcon และ บอร์ด Arduino ที่ควบคุมขดลวดโซลินอยด์ หลักการทำงานคือ รับภาพที่มีตัวอักษรภาษาอังกฤษจากกล้อง โดยเปลี่ยนภาพให้เป็นข้อความตัวอักษรด้วยกระบวนการ Optical Character Recognition ในโปรแกรม Halcon แล้วแปลงให้อยู่ในรูปของรหัส ASCII จากนั้นส่งรหัส ASCII ไปยังบอร์ด Arduino ที่ควบคุมขดลวดโซลินอยด์ ผ่าน Serial Port ซึ่งจะทำให้ขดลวดโซลินอยด์จำนวน 6 ชุด แสดงผลตามลักษณะนูนตามรูปแบบของอักษรเบรลล์ โดยเครื่องแสดงผลอักษรเบรลล์ที่สร้างขึ้นสามารถนำไปใช้เพื่อเป็นประโยชน์แก่ผู้พิการทางสายตา ได้



Project Title	Braille Display
Student	Miss. Thitisub Sriwalo Student ID 56010336
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2016
Project Advisor	Assoc. Prof. Suraphan Airphaiboon

ABSTRACT

This thesis presents the design and make a braille display by written code with the Halcon program connected the Arduino board that controlled the solenoid coil. The work of the Braille display is to get pictures that has English characters from the camera, converts the images into text messages with the Optical Character Recognition processes in the Halcon program to the form of ASCII code, and then sends the ASCII code to the Arduino board that controls the 6 solenoid coils through the serial port. The Braille display can be used benefitly of the visually impaired.



กิตติกรรมประกาศ

ปริญญานิพนธ์จากการศึกษาค้นคว้าอย่างอิสระฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ซึ่งผู้มีพระคุณท่านแรกที่คุณศึกษาใคร่ขอกราบพระคุณคือ รองศาสตราจารย์ ดอกเตอร์ สุรพันธุ์ เอื้อไพบูลย์ อาจารย์ผู้สอนที่ได้ให้ความรู้และคณาจารย์ทุกท่าน ที่ให้คำแนะนำตรวจทาน และแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่ทุกขั้นตอน เพื่อให้การเขียนปริญญานิพนธ์ค้นคว้าอย่างอิสระฉบับนี้สมบูรณ์ที่สุด และขอขอบพระคุณ คุณพ่อและคุณแม่ ที่ทำให้ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี สนับสนุนเงินทุน เป็นแรงผลักดันและให้กำลังใจ ขอขอบคุณรุ่นพี่และเพื่อนๆที่คอยให้คำปรึกษา รวมถึง และให้กำลังใจ ทำให้ปริญญานิพนธ์นี้สำเร็จไปได้ด้วยดี

ฐิติทรัพย์ ศรีวโล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไป III

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	V
สารบัญตาราง.....	VI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมุติฐานของการศึกษา.....	1
1.4 ขอบเขตการวิจัย.....	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	1
บทที่ 2 หลักการและทฤษฎี.....	2
2.1 องค์ประกอบของตา (Anatomy of eye).....	2
2.2 ความพิการทางการมองเห็น.....	3
2.2.1 ตาบอด.....	3
2.2.2 ตาเห็นเลือนราง.....	3
2.3 อักษรเบรลล์ (Braille).....	4
2.3.1 ประวัติ.....	4
2.3.2 สัญลักษณ์.....	4
2.3.3 Slate and Stylus.....	5
2.4 การเขียน - อ่านอักษรเบรลล์.....	6
2.5 โปรแกรม Halcon.....	7
2.6 Digital Image Processing.....	8
2.6.1 ขั้นตอนพื้นฐานของการประมวลผลภาพ.....	8
2.6.1.1 การรับข้อมูลภาพ (Image Acquisition).....	8
2.6.1.2 การปรับปรุงภาพ (Enhancement).....	9
2.6.1.3 การแยกภาพ (Segmentation).....	9
2.6.1.4 การจดจำและการตีความ (Recognition and Interpretation).....	9
2.7 ชนิดของภาพ.....	10
2.7.1 ภาพความเข้มแสง (Intensity Image).....	10
2.7.2 ภาพขาว-ดำ (Binary image).....	10
2.7.3 Grayscale Image.....	11
2.7.4 ภาพสี (RGB Image).....	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไป IV

สารบัญ(ต่อ)

หน้า

2.7.5 Indexed Image.....	12
2.8 Optical Character Recognition (OCR)	12
2.9 Arduino.....	13
2.9.1 ส่วนประกอบของ Arduino.....	13
2.9.2 โปรแกรมเขียนชุดคำสั่งของ Arduino.....	14
2.9.3 โครงสร้างภาษา C ที่ใช้ในโปรแกรม Arduino เบื้องต้น.....	15
2.9.3.1. ปรีโพรเซสเซอร์ไดเรกทีฟ (Preprocessor directives).....	15
2.9.3.2. ส่วนของการกำหนดค่า (Global declarations).....	16
2.9.3.3. ฟังก์ชัน setup () และฟังก์ชัน loop ().....	16
2.9.3.4. การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน (Users-defined function).....	16
2.9.3.5. ส่วนอธิบายโปรแกรม (Program comments).....	17
2.9.4 ชนิดของข้อมูล.....	18
2.10 American Standard Code for Information Interchange (ASCII).....	19
2.11 Serial Port.....	20
2.12 ขดลวด Solenoids.....	21
บทที่ 3 วิธีการดำเนินงาน.....	22
3.1 ขั้นตอนการดำเนินงาน.....	22
3.1.1 Flow Chart	22
3.2 การปฏิบัติงานบนโปรแกรม HALCON.....	23
3.2.1 ขั้นตอนการปฏิบัติการบนโปรแกรม HALCON.....	23
3.3 การปฏิบัติงานบน Arduino.....	33
3.4 ส่วนแสดงผลอักษรเบรลล์.....	34
บทที่ 4 การทดลอง และผลการทดลอง.....	35
4.1 การทดลอง.....	35
4.2 ผลการทดลอง.....	35
บทที่ 5 สรุปผลและวิจารณ์.....	42
5.1 สรุปผลการทดลอง.....	42
5.2 ปัญหาที่เกิดขึ้น.....	42
5.3 แนวทางในการพัฒนา.....	42
5.4 ประโยชน์ที่ได้รับ.....	42
บรรณานุกรม.....	43
ภาคผนวก	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 องค์ประกอบของตา (1).....	2
2.2 องค์ประกอบของตา (2).....	3
2.3 Braille Louis	4
2.4 องค์ประกอบของอักษรเบรลล์.....	4
2.5 อักษรเบรลล์ ในภาษาอังกฤษ และ ตัวเลข.....	5
2.6 Slate and Stylus	5
2.7 การอ่านอักษรเบรลล์.....	6
2.8 การเขียน – อ่านอักษรเบรลล์	6
(ก) วิธีการเขียนอักษรเบรลล์ เขียนจากขวาไปซ้าย.....	6
(ข) วิธีการอ่านอักษรเบรลล์ ใช้นิ้วสัมผัส อ่านจากซ้ายไปขวา.....	6
2.9 ส่วนประกอบของ HALCON.....	7
2.10 วิธีการรับภาพจากกล้องเชื่อมต่อกับโปรแกรม HALCON.....	8
2.11 ทฤษฎีและการประยุกต์ใช้งานกับภาพดิจิทัล.....	8
2.12 ขั้นตอนพื้นฐานของการประมวลผลภาพ.....	9
2.13 ทฤษฎีและการประยุกต์ใช้งานกับภาพดิจิทัล.....	10
2.14 ภาพขาว-ดำ (Binary image)	10
2.15 Grayscale Image	11
2.16 ภาพสี (RGB Image)	11
2.17 Indexed Image.....	12
2.18 ผลการนำ OCR ไปใช้งาน.....	12
2.19 บอร์ด Arduino.....	13
2.20 ส่วนประกอบของบอร์ด Arduino.....	14
2.21 โปรแกรมเขียนชุดคำสั่งของ Arduino (Arduino IDE).....	14
2.22 หน้าต่างการทำงานของ Arduino IDE version 1.8.2.....	14
2.23 ตาราง ASCII Code.....	19
2.24 ตัวอย่าง Serial Port.....	20
(ก) Serial Port.....	20
(ข) USB Port.....	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.25 โครงสร้างภายในขดลวด Solenoids.....	21
2.26 ขดลวด Solenoids ที่ใช้ในการทดลอง.....	21
3.1 แสดงขั้นตอนการเชื่อมต่อระหว่างกล้องกับโปรแกรม HALCON (Image Acquisition).....	24
(ก) ค้นหากล้องที่เชื่อมต่อกับโปรแกรมHALCON.....	25
(ข) ประมวลผลออกมาเป็นcode.....	25
(ค) Code ที่ได้จากการ Image Acquisition.....	25
3.2 ชุดคำสั่งแสดงพารามิเตอร์ในส่วนของ open_framgrabber.....	26
3.3 ชุดคำสั่งแสดงพารามิเตอร์ ของการรับภาพจากกล้องที่เชื่อมต่อ.....	27
3.4 แสดงค่าพารามิเตอร์ของการรับภาพจากกล้องที่เชื่อมต่อที่จับภาพแบบไม่พร้อมกัน.....	27
3.5 ขั้นตอนการเชื่อมต่อและตั้งค่า Serial Port	
(ก) ชุดคำสั่งเชื่อมต่อกับ Serial Port.....	27
(ข) ชุดคำสั่งกำหนดค่าที่เชื่อมต่อกับ Serial Port.....	28
(ค) ชุดคำสั่งการตั้งค่าเพื่ออ่านข้อมูลจาก Serial Port.....	28
(ง) ชุดคำสั่งล้างค่าที่รับมา ใน Serial Port.....	28
3.6 แสดงชุดคำสั่งการปรับปรุงภาพ การเปลี่ยนภาพสีให้เป็นภาพ Grey Scale.....	29
3.7 แสดงการกำหนดค่าตัวอักษรที่จะแสดงผลบนหน้าต่าง	
(ก) ชุดคำสั่งรับข้อมูลภาพเข้ามา.....	29
(ข) ชุดคำสั่งกำหนดค่าพารามิเตอร์ของจอแสดงผล.....	30
(ค) ชุดคำสั่งกำหนดค่าพารามิเตอร์ ของตัวอักษรที่จะแสดงผล.....	30
3.8 แสดงขั้นตอนการค้นหาตัวอักษรบนรูปภาพ และการตั้งค่า Parameter	
(ก) ชุดคำสั่งการกำหนดเลือกใช้ไฟล์ OCR ที่ชื่อ Industrial_0-9_NoRej.....	31
(ข) ชุดคำสั่งการกำหนดขอบเขต ของค่าพารามิเตอร์.....	31
(ค) ชุดคำสั่งแสดงชุดคำสั่งการค้นหาตัวอักษรบนภาพ.....	31
3.9 แสดงชุดคำสั่งการแปลงข้อมูลเพื่อเก็บค่าตัวอักษรที่ได้รับจากวัตถุ.....	32
3.10 แสดงชุดคำสั่งการแปลงข้อมูลเพื่อเก็บค่าตัวอักษรที่ได้รับจากการทำการแบ่งข้อความ (Text segmentation)	32
3.11 การแปลงข้อมูลจาก ข้อมูล 'String' เป็น ASCII Code.....	32
3.12 บอร์ด Arduino ที่ใช้ในการทดลอง.....	33
3.13 บอร์ด Arduino และ ขดลวด Solenoids ที่ใช้ในการทดลอง.....	33
3.14 แสดงส่วนแสดงผลอักษรเบรลล์.....	34
3.15 แสดงส่วนเชื่อมต่อระหว่างขดลวดโซลินอยด์ กับ บอร์ด Arduino.....	34
4.1 แสดงผลโปรแกรม HALCON สามารถแยกแยะตัวอักษรภาษาอังกฤษ.....	35
4.2 LEDs Dot Matrix และ ขดลวด Solenoids ที่ทำงานตามที่ได้เขียนคำสั่งควบคุมไว้.....	35

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงชนิดของข้อมูล การเก็บข้อมูล และ ขนาด ของชนิดข้อมูลที่ใช้ในโปรแกรม Arduino เบื้องต้น..18	
3.1 คำสั่งที่ใช้ในการปฏิบัติงานบนโปรแกรม HALCON พร้อมคำอธิบาย.....24	
4.1 แสดงผลการทดลองเมื่อนำกล้องส่องไปยังภาพที่มีตัวหนังสือภาษาอังกฤษ โดยแสดงผลสองแบบคือ แบบLEDs Dot Matrix และ แบบแสดงผลด้วยขดลวด Solenoids โดยแยกทีละตัวอักษร.....36	



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

มนุษย์ดำเนินชีวิตโดยใช้ประสาทสัมผัสทั้ง 5 ซึ่ง “ดวงตา” เป็นอวัยวะในการมองเห็น ทำให้เราสามารถใช้ชีวิตได้เป็นไปอย่างปกติ เช่น อ่าน เรียน เขียน หนังสือ ดูโทรทัศน์ เลือกซื้อสินค้า เดินทางไปยังสถานที่ต่างๆ และอื่นๆที่มนุษย์พึงจะทำได้ แต่ถ้าหากเราลองหลับตาจำลองว่าไม่มีดวงตาหรือมองไม่เห็น เราจะใช้ชีวิตอย่างไร? ยังสามารถ เขียน-อ่านหนังสือ หรือเลือกสินค้าได้อย่างเดิมหรือไม่? แค่การอ่านหนังสือพิมพ์ในตอนเช้าอาจไม่ใช่เรื่องง่ายอีกต่อไป และนี่คือสิ่งที่ผู้พิการทางสายตาคบเจอ

ในปีค.ศ.1824 นายหลุยส์ เบรลล์ คุณครูตาบอดชาวฝรั่งเศสได้ประดิษฐ์อักษรเบรลล์ สำหรับผู้พิการทางสายตาคบอ่าน โดยใช้การสัมผัสจุดนูนของนิ้วมือ ที่พัฒนาจากวิธีการส่งข่าวสารลับของทหาร ซึ่งใช้ในการสื่อสาร การศึกษา อย่างแพร่หลาย ถึงแม้ในปัจจุบันจะมีสิ่งอำนวยความสะดวก เช่น Braille Block (ทางเดินคนตาบอด) แต่ป้ายบอกทาง หรือ บรรจุภัณฑ์ส่วนใหญ่ ยังไม่มีฉลากสำหรับผู้พิการทางสายตา อาจเป็นอุปสรรคในการดำเนินชีวิตเหมือนคนสายตาปกติ

ข้าพเจ้าจึงมีแรงบันดาลใจที่จะสร้างอุปกรณ์ ที่ทำให้ผู้พิการทางสายตาสามารถรับรู้ เข้าใจ สิ่งต่างๆ รอบตัว ที่เป็นอักษรคนตาบอดได้โดยนำความรู้ด้านอิเล็กทรอนิกส์ การออกแบบ และการเขียนโปรแกรม มาประยุกต์ใช้ เพื่อให้ผู้พิการทางสายตา สามารถนำไปใช้ในชีวิตประจำวันได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

สามารถสร้างอุปกรณ์ที่สามารถอ่าน อักษรคนตาบอดแล้วเปลี่ยน ให้เป็น จุดนูนแบบอักษรเบรลล์ โดยใช้ความรู้ทางด้านอิเล็กทรอนิกส์ และอื่นๆมาประยุกต์ใช้งานได้

1.3 สมมุติฐานของการศึกษา

สามารถเขียนโปรแกรม จับภาพอักษรคนตาบอด โดยรับภาพจากกล้อง แล้วแปลเป็นอักษรเบรลล์ บนวงจรที่สร้างจุดนูนที่มีลักษณะเป็นอักษรเบรลล์ได้

1.4 ขอบเขตการวิจัย

ศึกษาเกี่ยวกับอักษรเบรลล์ การเขียนโปรแกรม วงจรอิเล็กทรอนิกส์ และหลักการ Image Processing

1.5 ประโยชน์ที่คาดว่าจะได้รับ

นักศึกษาสามารถเข้าใจหลักการทำงาน วิธีการนำหลักการ Image Processing มาใช้งาน สามารถสร้างวงจรอิเล็กทรอนิกส์ และสามารถเขียนโปรแกรมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 1

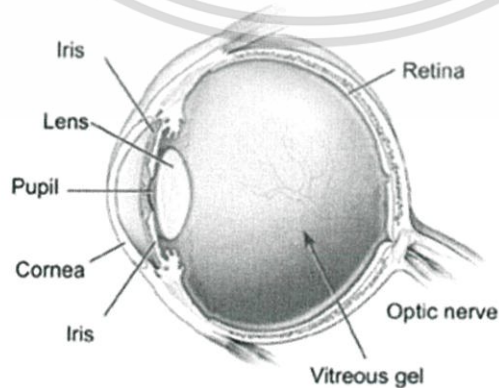
บทที่ 2

หลักการและทฤษฎี

2.1 องค์ประกอบของตา (Anatomy of eye)

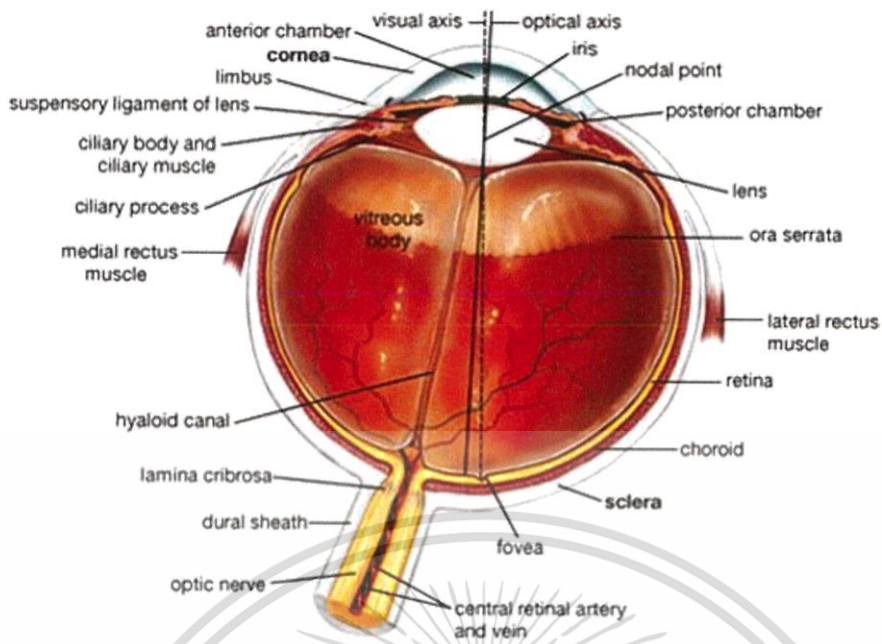
แบ่งออกเป็น

1. กระจกตา (Cornea) เป็นส่วนของตาดำ บางคนเปรียบเปรยว่าเป็นหน้าต่างของดวงตา เพราะตามปกติกระจกตาจะมีความใส และทำให้เกิดการหักเหของแสงมากที่สุด กระจกตาประกอบด้วยเนื้อเยื่อทั้งหมด 5 ชั้น ชั้นบนที่สุดเรียกว่า epithelium เป็นปราการด่านแรกที่ ปกป้องดวงตาจากการได้รับอันตราย
2. ม่านตา (Iris) เป็นส่วนที่มีสีซึ่งแตกต่างกันขึ้นอยู่กับเชื้อชาติ บ้างเป็นสีเขียว บ้างเป็นสีฟ้า สีน้ำตาล หรือสีดำ หน้าที่ของม่านตาคือช่วยในการควบคุมขนาดของรูม่านตา โดยการหดตัวหรือขยายตัวของกล้ามเนื้อม่านตา
3. เลนส์แก้วตา (Lens) เป็นส่วนที่ใสอยู่หลังม่านตา หน้าที่ของเลนส์แก้วตาคือช่วยโฟกัสเพื่อการมองเห็นที่ชัดเจนมากขึ้นสำหรับการอ่านหรือการมองระยะใกล้ โดยการปรับรูปร่างของเลนส์ให้เหมาะสม
4. รูม่านตา (Pupil) เป็นรูกลม ๆ ที่เห็นในดวงตา มีหน้าที่ควบคุมปริมาณแสงที่ผ่านเข้าสู่ลูกตา เมื่ออยู่ในสภาพแวดล้อมที่มีแสงจ้า ม่านตาจะหดทำให้แสงเข้าตาได้น้อยลง ส่วนในที่มืดม่านตาจะขยายเพื่อให้แสงเข้าตาได้มากขึ้น
5. น้ำวุ้นตา (Vitreous) น้ำวุ้นตาลักษณะคล้ายเจล อยู่ด้านหลังลูกตา ทำหน้าที่ให้ลูกตากคงรูปลักษณะกลมตลอดเวลา ในผู้ที่มีสายตาสั้นมาก ๆ บางครั้งอาจสังเกตเห็นว่ามีจุดดำหรือใยสีดำลอยไปมาในน้ำวุ้นตา ซึ่งเกิดจากการที่น้ำวุ้นตาเสื่อมนั่นเอง
6. เส้นประสาท (Optic Nerve) เป็นตัวส่งผ่านการกระตุ้นของการมองเห็นจากจอประสาทตมายังสมอง
7. จอประสาทตา (Retina) จอประสาทตาประกอบด้วยเส้นประสาทตาที่มีความละเอียดสูงอยู่ในผนังชั้นในของลูกตา ทำหน้าที่คล้ายกับฟิล์มถ่ายรูปโดยจะส่งผ่านรูปไปยังสมอง ซึ่งในภาวะสายตาสกิด การหักเหของแสงจะลงมาตกกระทบที่จอประสาทตาพอดี
8. นัยน์ตาขาว (Sclera) เป็นส่วนสีขาวของลูกตา ทำหน้าที่ป้องกันอันตรายต่อลูกตา ช่วยให้เกิดความแข็งแรง



รูปที่ 2.1 องค์ประกอบของตา(1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 2



รูปที่ 2.2 องค์ประกอบของตา(2)

2.2 ความพิการทางการมองเห็น

- 2.1.1 ตาบอด หมายถึง การที่บุคคลมีข้อจำกัดในการปฏิบัติกิจกรรมในชีวิตประจำวันหรือการเข้าไปมีส่วนร่วมในกิจกรรมทางสังคม ซึ่งเป็นผลมาจากการมีความบกพร่องในการเห็น เมื่อตรวจวัดการเห็นของสายตาสั้นที่ดีกว่าเมื่อใช้แว่นสายตารธรรมดาแล้ว อยู่ในระดับแย่กว่า 3 ส่วน 60 เมตร หรือ 20 ส่วน 400 ฟุต ลงมาจนกระทั่งมองไม่เห็นแม้แต่แสงสว่าง หรือมีลานสายตาแคบกว่า 10 องศา
- 2.1.2 ตาเห็นเลือนราง หมายถึง การที่บุคคลมีข้อจำกัดในการปฏิบัติกิจกรรมในชีวิตประจำวันหรือการเข้าไปมีส่วนร่วมในกิจกรรมทางสังคม ซึ่งเป็นผลมาจากการมีความบกพร่องในการเห็น เมื่อตรวจวัดการเห็นของสายตาสั้นที่ดีกว่า เมื่อใช้แว่นสายตารธรรมดาแล้ว อยู่ในระดับตั้งแต่ 3 ส่วน 60 เมตร หรือ 20 ส่วน 400 ฟุต ไปจนถึงแย่กว่า 6 ส่วน 18 เมตร หรือ 20 ส่วน 70 ฟุต หรือ มีลานสายตาแคบกว่า 30 องศา

2.3 อักษรเบรลล์ (Braille)

2.3.1 ประวัติ

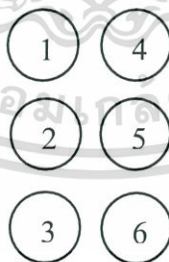
หลุยส์ เบรลล์ (Louis Braille) เกิดที่เมือง Coupvray ใกล้กับปารีส ในประเทศฝรั่งเศส แต่เติบโตที่เมือง Lisle บิดาคือ ไซมอน เรนเน่ เบรลล์ (Simon-René Braille) มีอาชีพทำขนมปัง เมื่ออายุได้ 3 ปี เบรลล์ประสบอุบัติเหตุจากเข็มของบิดา ทำให้ตาข้างซ้ายบอด เมื่ออายุได้ 4 ปี โรคตาอีกเสบอย่างรุนแรงทำให้เบรลล์ตาบอดทั้ง 2 ข้าง แต่เบรลล์ก็ยังได้เข้าเรียน ด้วยการสนับสนุนจากพ่อ ในปี 1821 กัปตันชาร์ล บาบิแอร์ นายทหารแห่งกองทัพบกฝรั่งเศสได้มาเยี่ยมโรงเรียน และนำวิธีการส่งข่าวสารของทหารในเวลากลางคืนเรียกว่า night-writing มาลองใช้ ซึ่งเป็นรหัสที่ใช้จุด 12 จุด และใช้ค่อนข้างยาก ในปีนั้นเอง เบรลล์ได้เริ่มประดิษฐ์อักษรที่ใช้ระบบจุดเช่นกัน เบรลล์ใช้จุดเพียง 6 จุด และใช้เพียงนิ้วเดียววางบนจุดทั้งหมด



รูปที่ 2.3 Braille Louis

2.3.2 สัญลักษณ์

ประกอบไปด้วยจุดเบรลล์ 6 จุด โดยเรียงแบบ 2 แถวๆ ละ 3 จุด เรียกว่า Braille Cell



รูปที่ 2.4 องค์ประกอบของอักษรเบรลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 4

ภาษาอังกฤษ

1	2	3	4	5
A	B	C	D	E
⠁	⠃	⠉	⠋	⠑
6	7	8	9	0
F	G	H	I	J
⠕	⠗	⠈	⠊	⠚
K	L	M	N	O
⠕	⠇	⠓	⠝	⠛
P	Q	R	S	T
⠏	⠒	⠞	⠘	⠜
U	V	W	X	Y
⠥	⠧	⠪	⠬	⠮
Z				
⠵				

รูปที่ 2.5 อักษรเบรลล์ ในภาษาอังกฤษ และ ตัวเลข

2.3.3 Slate and Stylus

อุปกรณ์สำหรับเขียนอักษรเบรลล์, แผ่นและเข็ม สำหรับเขียนอักษรเบรลล์ลงบนกระดาษ อุปกรณ์นี้ ประดิษฐ์ โดย Charles Barbier and Louis Braille ประกอบด้วย แผ่นรองเขียนอักษรเบรลล์ ทำด้วยพลาสติก เหล็ก หรือไม้ และเข็มดันกระดาษ ตัวแผ่นเจาะเป็นช่องอักษรเบรลล์สำหรับสร้างจุดเบรลล์ 2 แถวๆ ละ 3 จุด แต่ละช่องเรียกว่า Braille Cell รวม 28 ช่องต่อ 1 บรรทัด แผ่นหลังเป็นที่วางกระดาษ มีบานพับปิดเปิดระหว่าง 2 แผ่น

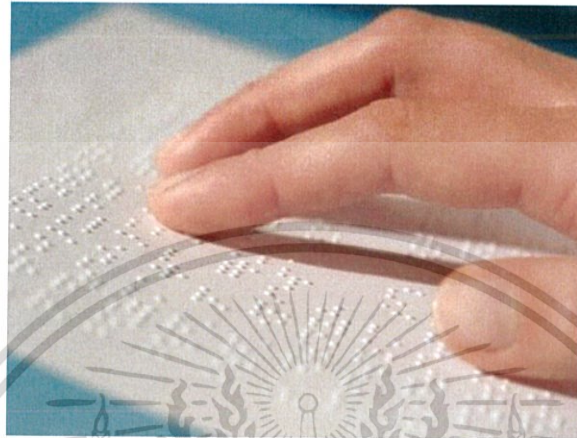


รูปที่ 2.6 Slate and Stylus

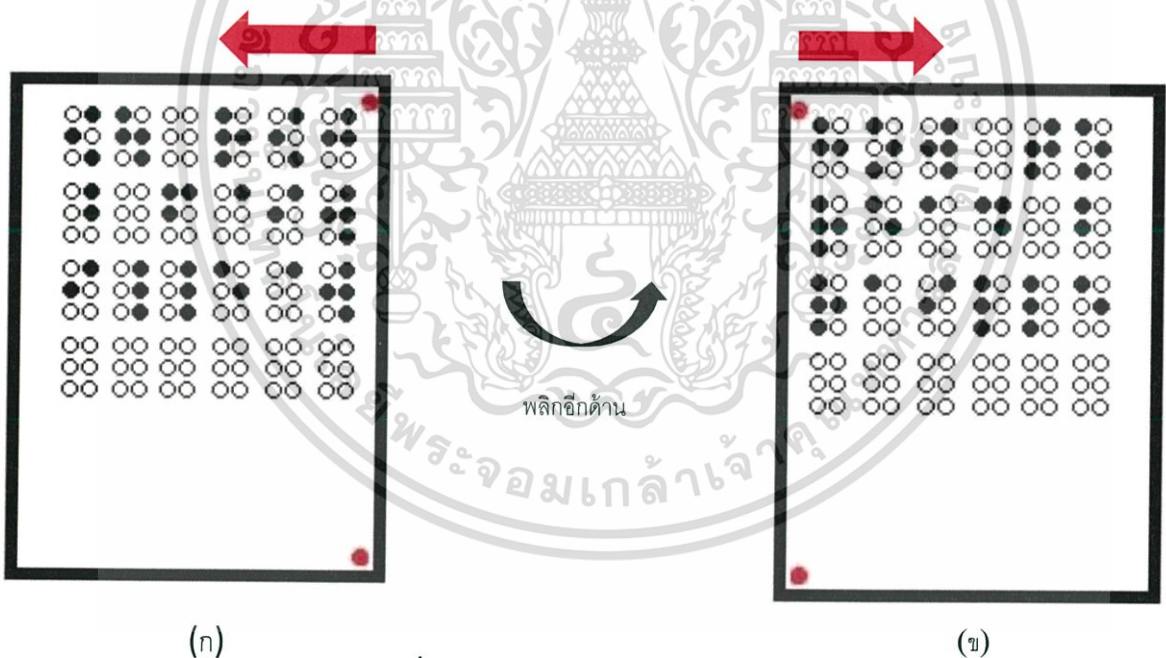
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 5

2.4 การเขียน - อ่านอักษรเบรลล์

การเขียนอักษรเบรลล์ พลิกกระดาษอีกด้าน ใช้ Slate and Stylus ใช้ stylus กดจุดจาก ขวาไปซ้าย เขียนแบบ ตรงข้ามกับตัวอักษรเบรลล์ เช่น อักษร A คือ จุดหมายเลข 1 เมื่อเวลาเขียนจะพลิกกระดาษอีกด้าน ใช้ stylus กดที่ตำแหน่งจุดหมายเลข 4 แทน เมื่อพลิกกระดาษกลับมา จุดนั้นจะอยู่ในตำแหน่งหมายเลข 1



รูปที่ 2.7 การอ่านอักษรเบรลล์



รูปที่ 2.8 การเขียน - อ่านอักษรเบรลล์

(ก) วิธีการเขียนอักษรเบรลล์ (เขียนจากขวาไปซ้าย)

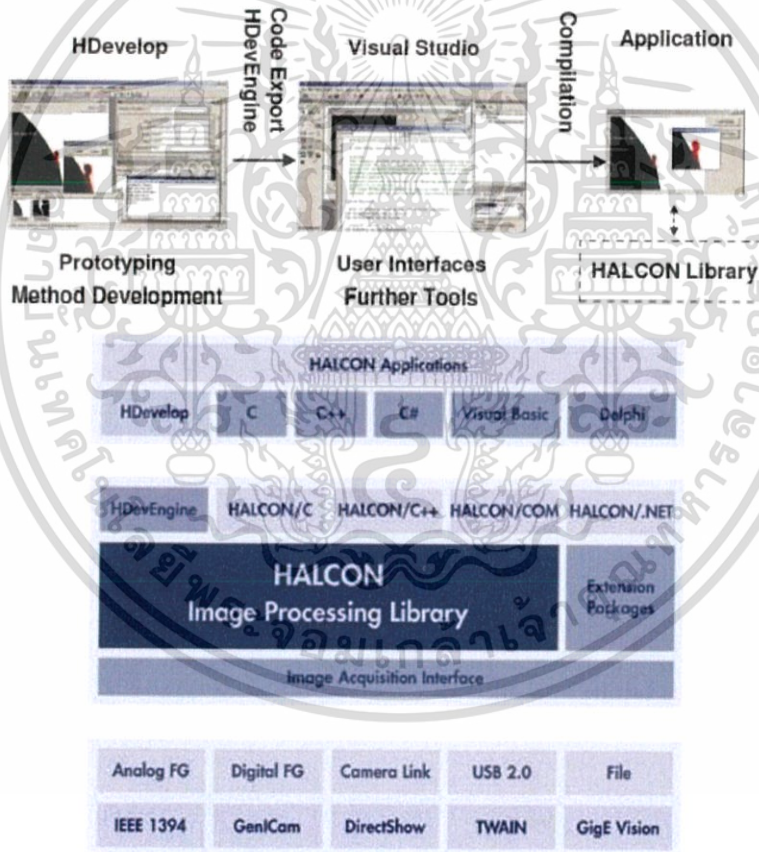
(ข) วิธีการอ่านอักษรเบรลล์ ใช้นิ้วสัมผัส (อ่านจากซ้ายไปขวา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 6

2.5 โปรแกรม Halcon

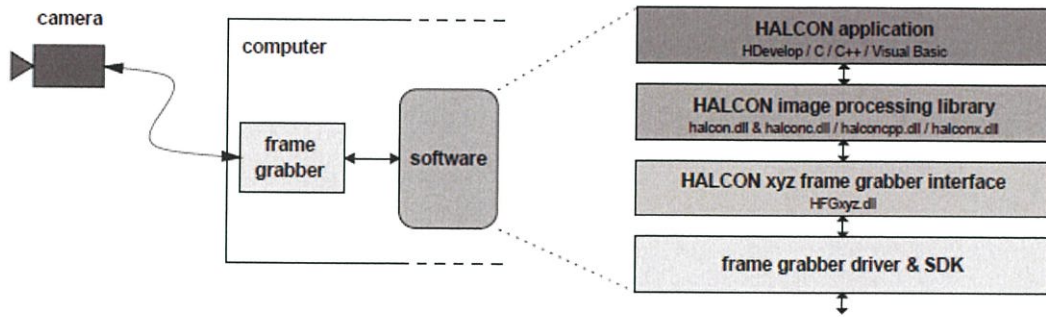
การศึกษาเกี่ยวกับการประมวลผลภาพในปัจจุบัน มีเครื่องมือ หรือว่าโปรแกรมสำเร็จรูปให้เลือกใช้มากมาย หรือถ้าจะมีการเขียนโปรแกรมด้วยตนเองก็สามารถทำได้ เช่นการใช้โปรแกรม MatLAB, LabView, RoboRelm, SimpleCV, OpenCV, ภาษา Python, VB, C++, C# หรือแม้กระทั่ง Java บนระบบปฏิบัติการต่างๆ เครื่องมือเหล่านี้ จะสามารถแสดงให้เห็นถึงผลของภาพหลังจากการประมวลผลได้ทันที แต่เมื่อนำทฤษฎีการประมวลผลที่สูงขึ้นมาใช้งานร่วมกันด้วย ก็จะทำให้เกิดการประยุกต์ใช้งานได้จริงในหลายๆภาคส่วน เช่น ในทางอุตสาหกรรม ทางการแพทย์ ทางด้านความมั่นคง เป็นต้น

Halcon เป็นชุดโปรแกรมสำหรับพัฒนาและสร้างซอฟต์แวร์การตรวจสอบอย่างละเอียดด้วยภาพ ที่มีความยืดหยุ่นสูง ใช้งานง่าย และราคาไม่สูงมาก สามารถพัฒนาโดยเริ่มจาก การออกแบบระบบด้วยการใช้งานและปรับค่า พารามิเตอร์ต่างๆบนโปรแกรม HDevelop ที่สามารถเชื่อมต่อกับกล้องจับภาพได้หลายชนิด เช่น กล้อง Webcam ที่นิยมใช้กัน หลังจากนั้นทำการ export คำสั่งที่ได้ทดลองจนถูกต้องแล้ว ออกไปใช้งานร่วมกันกับภาษาที่ต้องการในการสร้างส่วนของ user interface ต่อไป

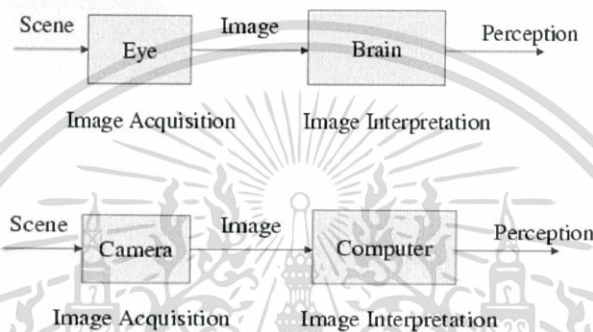


รูปที่ 2.9 ส่วนประกอบของ HALCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 วิธีการรับภาพจากกล้องเชื่อมต่อกับโปรแกรม HALCON



รูปที่ 2.11 ทฤษฎีและการประยุกต์ใช้งานกับภาพดิจิทัล

2.6 Digital Image Processing

การประมวลผลภาพดิจิทัล สองมิติโดยใช้เครื่องคอมพิวเตอร์ หรือ การประมวลผลทางดิจิทัลของข้อมูลสองมิติใดๆ ส่วนภาพดิจิทัล หมายถึง อาร์เรย์สองมิติของจำนวนจริงหรือ จำนวนเชิงซ้อนที่ถูกแทนด้วยจำนวนบิตที่จำกัด ภาพที่มีค่าระดับความเข้ม เท่ากับ $2^8 = 256$ ระดับ โดยที่ 0 แทนระดับความเข้มต่ำสุด และ 255 แทนระดับความเข้มสูงสุด

2.6.1 ขั้นตอนพื้นฐานของการประมวลผลภาพ

2.6.1.1 การรับข้อมูลภาพ (Image Acquisition)

หน่วยรับข้อมูลภาพแบ่งออกเป็นสองหน่วยพื้นฐาน ส่วนแรกเป็นอุปกรณ์ที่ไวต่อสเปกตรัมของคลื่นแม่เหล็กไฟฟ้า ได้แก่ x-rays ,Ultraviolet ,Visible Light หรือ Infrared และให้สัญญาณไฟฟ้าเป็นสัดส่วนกับระดับพลังงานของคลื่นแม่เหล็กไฟฟ้าที่ตรวจวัด หน่วยที่สองเป็นหน่วยที่ทำการเปลี่ยนแปลงเอาต์พุตที่เป็นสัญญาณไฟฟ้าให้อยู่ในรูปแบบดิจิทัล ซึ่งสามารถนำไปประมวลผลในเครื่องคอมพิวเตอร์ได้ ตัวอย่างของอุปกรณ์ ประเภทนี้ได้แก่ Charge-Coupled Device (CCD) เป็นต้น CCD เป็นอุปกรณ์ที่สำคัญในกล้องดิจิทัลและเครื่องสแกนเนอร์

2.6.1.2. การปรับปรุงภาพ (Enhancement)

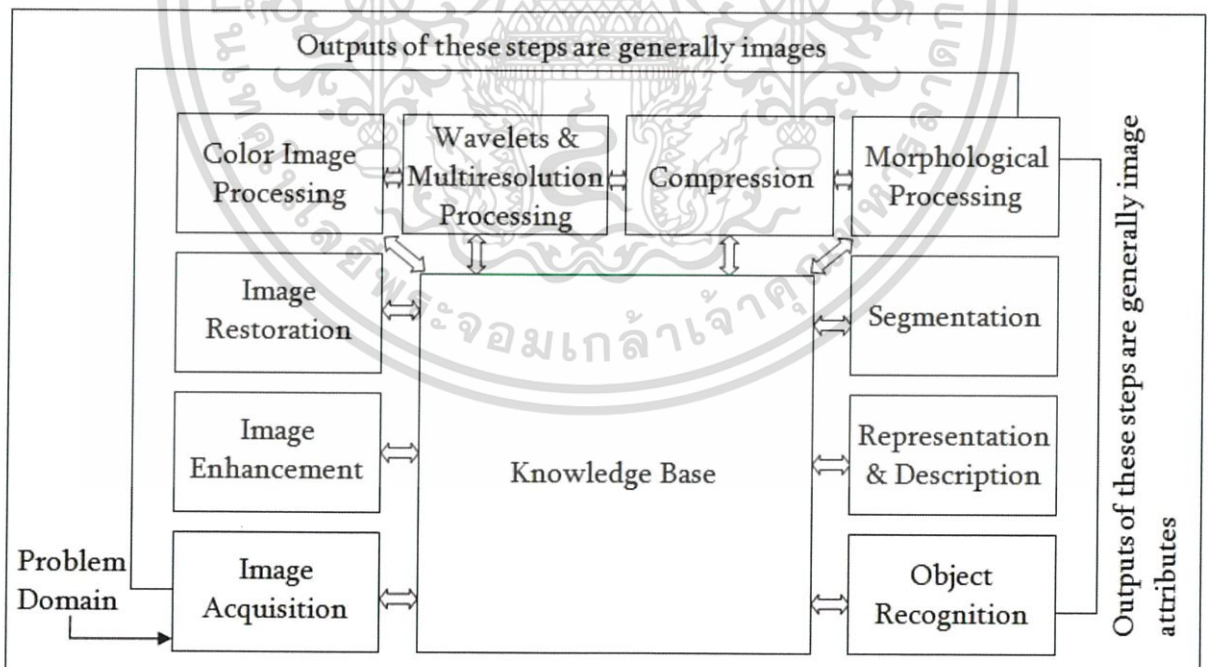
เป็นขบวนการก่อนประมวลผล เพื่อเพิ่มโอกาสที่จะประสบความสำเร็จของขั้นตอนการประมวลผล (Enhancement) เช่นการเพิ่มคอนทราสต์ การกำจัดนอยส์ การแยกบริเวณที่เป็นตัวอักษร เป็นต้น

2.6.1.3. การแยกภาพ (Segmentation)

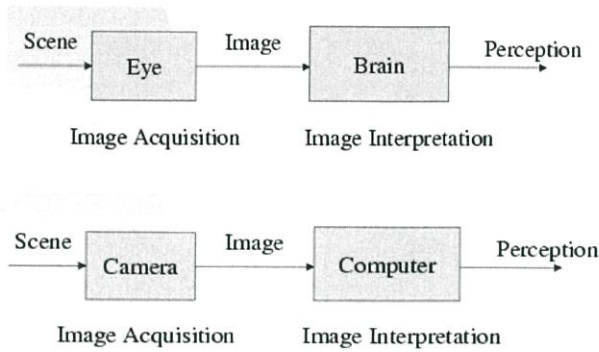
คือการแบ่งส่วนภาพออกตามองค์ประกอบหรือตามวัสดุที่ประกอบภาพ โดยทั่วไปการแยกส่วนภาพโดยอัตโนมัติเป็นงานที่ยากลำบากในการประมวลผลภาพดิจิทัล และ ความสำเร็จของการประมวลผลขึ้นกับประสิทธิภาพของการแยกส่วนภาพ เอกลักษณ์ที่ได้จากการขบวนการแบ่งส่วนภาพเป็นข้อมูลพิกเซลดิบบๆ ซึ่งรวมส่วนที่เป็นขอบภาพ (Boundary) และจุดทุกจุดภายในบริเวณที่แยกได้ ไม่ว่าจะ เป็นกรณีใดการเปลี่ยนแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมสำหรับการประมวลผลด้วยคอมพิวเตอร์ เป็นสิ่งจำเป็น โดยต้องกำหนดขอบเขตของข้อมูลในการแทน (Representation) ซึ่งเป็นเพียงขั้นตอนหนึ่งที่ใช้ในการแปลงข้อมูลดิบบให้อยู่ในรูปแบบที่เหมาะสมแก่การประมวลผลด้วยคอมพิวเตอร์ ขั้นตอนต่อไปคือการระบุวิธีใช้ในการอธิบาย (Describe) ข้อมูลเพื่อให้ลักษณะที่เราสนใจเด่นชัดขึ้น การอธิบายหรือที่เรียกว่าการเลือกลักษณะพิเศษเกี่ยวข้องกับการแยกลักษณะเพื่อให้ได้เป็นข้อมูลเชิงปริมาณ หรือให้ได้ลักษณะที่ใช้ในการแยกความแตกต่างวัตถุกลุ่มหนึ่งออกจากอีกกลุ่มหนึ่ง

2.6.1.4. การจดจำและการตีความ (Recognition and Interpretation)

การจดจำเป็นขบวนการทำฉลาก ให้กับวัตถุโดยใช้ลักษณะพิเศษที่ได้จากขั้นตอนการอธิบาย การตีความเกี่ยวกับการให้ความหมายกับกลุ่มของวัตถุที่จำได้



รูปที่ 2.12 ขั้นตอนพื้นฐานของการประมวลผลภาพ



รูปที่ 2.13 ทฤษฎีและการประยุกต์ใช้งานกับภาพดิจิทัล

2.7 ชนิดของภาพ

ชนิดของภาพมีผลต่อขบวนการ Digital Image Processing มีความแตกต่าง ของคลาส ชนิดของมูล และมีความแตกต่างในการประกาศตัวแปร

2.7.1 ภาพความเข้มแสง (Intensity Image)

ภาพความเข้มแสงคือเมทริกซ์ของข้อมูลซึ่งถูกลดสเกลลงให้เป็นค่าความเข้มแสง เมื่อสมาชิกของค่าในภาพ ความเข้มแสงเป็นชนิด uint8 หรือ uint16 ค่าของภาพจะอยู่ในช่วง [0,255] และ [0,65535] ตามลำดับถ้าค่า ในภาพความเข้มแสงชนิด double ค่าของภาพจะอยู่ในช่วง [0,1]

2.7.2 ภาพขาว-ดำ (Binary image)

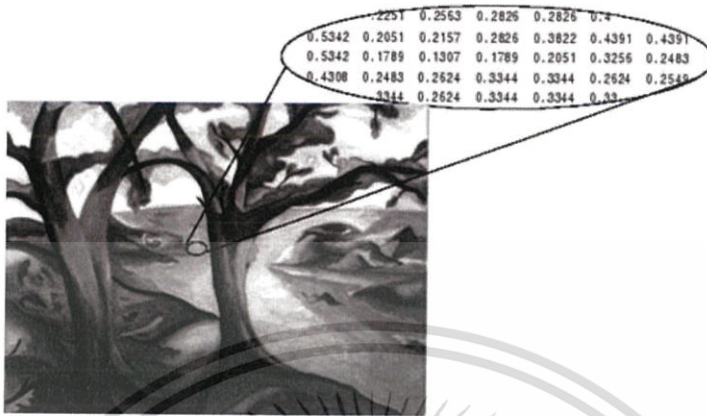
เป็นรูปที่ใช้เนื้อที่เพียง 1 บิต ต่อ พิกเซล โดยค่าสีจะมีแค่สองค่าคือ 0 หรือสีดำ และ 1 หรือสีขาว

1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

รูปที่ 2.14 ภาพขาว-ดำ (Binary image)

2.7.3 Grayscale Image

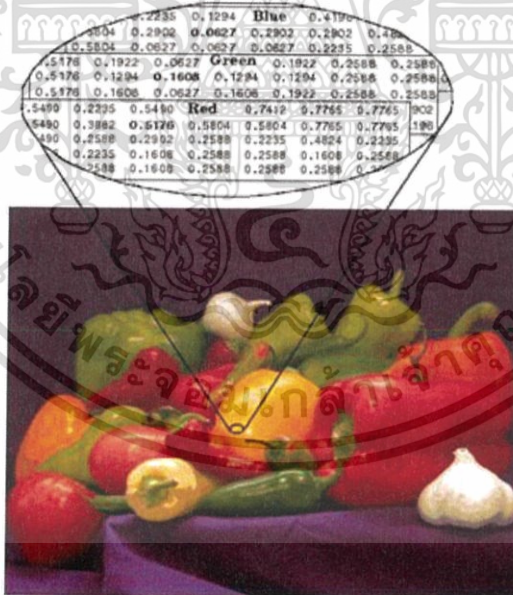
เป็นรูปที่เก็บโดยใช้รูปแบบของอาร์เรย์ 2 มิติ โดยค่าที่เก็บจะมีค่าอยู่ในช่วงๆหนึ่ง ซึ่งระดับของสีขึ้นอยู่กับขนาดของบิตที่ใช้เก็บค่าสี



รูปที่ 2.15 Grayscale Image

2.7.4 ภาพสี (RGB Image)

เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติ ขนาด $m \times n \times 3$ โดยที่ m คือความยาว และ n คือความกว้างของภาพในหน่วยพิกเซล ส่วนมิติสุดท้ายนั้น ในแต่ละมิติจะเก็บค่าสีแยกกัน คือสีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue)

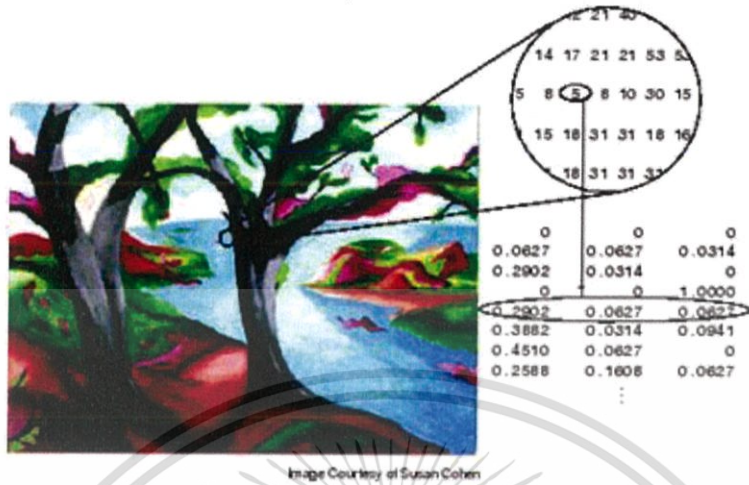


รูปที่ 2.16 ภาพสี (RGB Image)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.5 Indexed Image

เป็นรูปที่มีรูปแบบการเก็บแบบ indexed คือ ภาพประเภทนี้จะเก็บค่าสีเป็น indexed และในแต่ละช่องอาร์เรย์ จะเก็บตำแหน่งของสีใน indexed นั้นๆไว้



รูปที่ 2.17 Indexed Image

2.8 Optical Character Recognition (OCR)

การรู้จำอักขระด้วยแสง หรือมักเรียกอย่างย่อว่า โอซีอาร์ คือกระบวนการทางกลไกหรือทางอิเล็กทรอนิกส์เพื่อแปลภาพของข้อความจากการเขียนหรือจากการพิมพ์ ไปเป็นข้อความที่สามารถแก้ไขได้โดยเครื่องคอมพิวเตอร์ การจับภาพอาจทำได้โดยเครื่องสแกนเนอร์ กล้องดิจิทัล การรู้จำอักขระด้วยแสง (การใช้เทคนิคทางแสง เช่นกระจกและเลนส์) การรู้จำอักขระทางดิจิทัล (การใช้เทคนิคทางดิจิทัล เช่นสแกนเนอร์และอัลกอริธึมคอมพิวเตอร์) เดิมเคยเป็นสาขาที่แยกจากกัน แต่เนื่องจากเหลือการใช้งานน้อยมากที่ใช้เฉพาะเทคนิคทางแสง คำว่า โอซีอาร์ ในปัจจุบันจึงกินความกว้างถึงการประมวลผลภาพทางดิจิทัล ด้วยเช่นกัน

ระบบในสมัยเริ่มแรกต้องการการฝึกฝน (สอนตัวอย่างที่รู้จักแล้วของแต่ละตัวอักษร) เพื่ออ่านฟอนต์หนึ่ง ๆ ปัจจุบัน ระบบ "อัจฉริยะ" ที่สามารถอ่านฟอนต์ส่วนใหญ่ได้แม่นยำสูงนั้นสามารถพบได้ทั่วไป บางระบบถึงกับสามารถคงรูปแบบการจัดหน้าเดิมไว้ได้เกือบหมด ซึ่งรวมถึง รูปภาพ การแบ่งคอลัมน์ และส่วนประกอบที่ไม่ใช่ข้อความอื่น ๆ



รูปที่ 2.18 ผลการนำ OCR ไปใช้งาน

2.9 Arduino

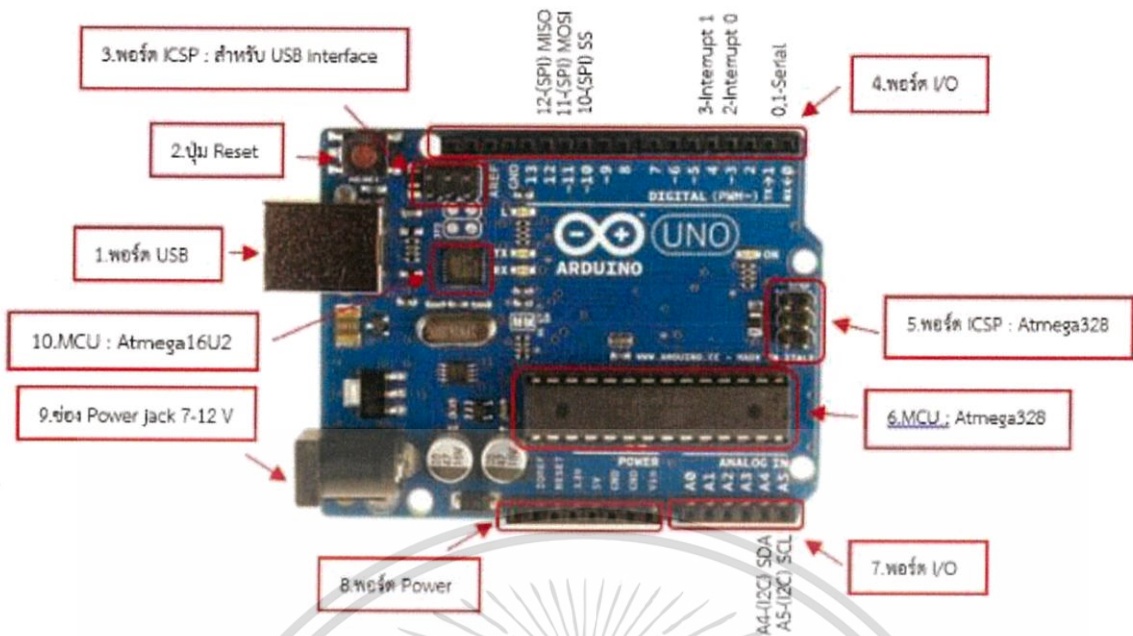
เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย เหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดหรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆเช่น Arduino XBee Shield, Arduino Music Shield มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย



รูปที่ 2.19 บอร์ด Arduino

2.9.1 ส่วนประกอบของ Arduino

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 บน Atmega16U2
4. I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
7. I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V_{in}
9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2



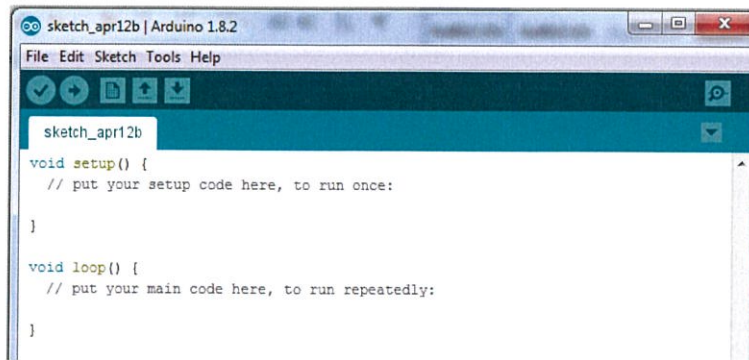
รูปที่ 2.20 ส่วนประกอบของบอร์ด Arduino

2.9.2 โปรแกรมเขียนชุดคำสั่งของ Arduino

บอร์ด Arduino ที่มีอยู่หลายแบบนี้จะใช้โปรแกรมตัวเดียวกันในการเขียนชุดคำสั่ง โดยโปรแกรมนี้มีชื่อว่า Arduino IDE (IDE : Integrated Development Environment) ซึ่งใช้งานได้ทั้งบนระบบปฏิบัติการ Window (XP Vista 7 8) ทั้ง 32 และ 64 บิต, Mac OS X และ Linux



รูปที่ 2.21 โปรแกรมเขียนชุดคำสั่งของ Arduino (Arduino IDE)



รูปที่ 2.22 หน้าต่างการทำงานของ Arduino IDE version 1.8.2

2.9.3 โครงสร้างภาษา C ที่ใช้ในโปรแกรม Arduino เบื้องต้น

โครงสร้างโปรแกรมภาษา C บน Arduino จะมีลักษณะแบบเดียวกับ C ทั่วไปประกอบไปด้วย

1. ปรีโพรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)
2. ส่วนของการกำหนดค่า (Global declarations)
3. ฟังก์ชัน setup () และ ฟังก์ชัน loop()
4. การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน (Users-defined function)
5. ส่วนอธิบายโปรแกรม (Program comments)

2.9.3.1. ปรีโพรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)

โดยปกติแล้วเกือบทุกโปรแกรมต้องมี โดยส่วนนี้จะเป็นส่วนที่คอมไพเลอร์จะมีการประมวลผลและทำตามคำสั่งก่อนที่จะมีการคอมไพล์โปรแกรม ซึ่งจะเริ่มต้นด้วยเครื่องหมายไดเรกทีฟ (directive) หรือเครื่องหมายสี่เหลี่ยม # แล้วจึงตามด้วยชื่อคำสั่งที่ต้องการเรียกใช้ หรือกำหนด โดยปกติแล้วส่วนนี้จะอยู่ในส่วนบนสุด หรือส่วนหัวของโปรแกรม และต้องอยู่นอกฟังก์ชันหลักใดๆก็ตาม #include เป็นคำสั่งที่ใช้อ้างอิงไฟล์ภายนอก เพื่อเรียกใช้ฟังก์ชัน หรือตัวแปรที่มีการสร้างหรือกำหนดไว้ในไฟล์นั้น รูปแบบการใช้งานคือ

```
#include <ชื่อไฟล์.h>
```

การอ้างอิงไฟล์จากภายใน หรือการอ้างอิงไฟล์ไลบรารีที่มีอยู่แล้วใน Arduino หรือเป็นไลบรารีที่เราเพิ่มเข้าไปเอง จะใช้เครื่องหมาย <> ในการคร่อมชื่อไฟล์ไว้ เพื่อให้โปรแกรมคอมไพเลอร์เข้าใจว่าควรไปหาไฟล์เหล่านี้จากในโฟลเดอร์ไลบรารี แต่หากต้องการอ้างอิงไฟล์ที่อยู่ในโฟลเดอร์โปรเจกต์ จะต้องใช้เครื่องหมาย "" คร่อมแทน ซึ่งคอมไพเลอร์จะวิ่งไปหาไฟล์นี้โดยอ้างอิงจากไฟล์โปรแกรมที่คอมไพเลอร์อยู่ เช่น

```
#include <Wire.h>    #include <Time.h>
```

#define เป็นคำสั่งที่ใช้ในการแทนชื่อความที่กำหนดไว้ ด้วยชื่อความที่กำหนดไว้ ซึ่งการใช้คำสั่งนี้ข้อดีคือจะไม่มีการอ้างอิงกับตัวโปรแกรมเลยเช่น

```
#define LEDPIN 13
```

2.9.3.2. ส่วนของการกำหนดค่า (Global declarations)

ส่วนนี้จะเป็นส่วนที่ใช้ในการกำหนดชนิดตัวแปรแบบนอกฟังก์ชัน หรือประกาศฟังก์ชัน เพื่อให้ฟังก์ชันที่ประกาศสามารถกำหนด หรือเรียกใช้ได้จากทุกส่วนของโปรแกรม เช่น

```
int pin = 13;
void blink(void);
```

2.9.3.3. ฟังก์ชัน setup () และฟังก์ชัน loop()

ฟังก์ชัน setup() และฟังก์ชัน loop() เป็นคำสั่งที่ถูกบังคับให้ต้องมีในทุกโปรแกรม โดยฟังก์ชัน setup() จะเป็นฟังก์ชันแรกที่ถูกเรียกใช้ นิยมใช้กำหนดค่า หรือเริ่มต้นใช้งานไลบรารีต่างๆ เช่น ในฟังก์ชัน setup() จะมีคำสั่ง pinMode() เพื่อกำหนดให้ขาใดๆก็ตามเป็นดิจิตอลอินพุต หรือเอาต์พุต ส่วนฟังก์ชัน loop() จะเป็นฟังก์ชันที่ทำงานหลังจากฟังก์ชัน setup() ได้ทำงานเสร็จสิ้นไปแล้ว และมีการวนรอบแบบไม่รู้จบ เมื่อฟังก์ชัน loop() งานครบตามคำสั่งแล้ว ฟังก์ชัน loop() ก็จะถูกเรียกขึ้นมาใช้อีก เช่น

```
int pin = 13;
void setup() {
  pinMode(pin, OUTPUT);}
void loop() {
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(1000);}
```

2.9.3.4. การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน (Users-defined function)

ในการสร้างฟังก์ชันขึ้นมา คำสั่งต่างๆที่อยู่ภายในฟังก์ชัน ต้องอยู่ภายใต้เครื่องหมายปีกกาเปิด { และปีกกาปิด } เท่านั้น ภายใต้เครื่องหมาย {} เราสามารถนำฟังก์ชันหรือคำสั่งใดๆก็ได้มาใส่ไว้ แต่จะต้องคั่นแต่ละคำสั่งด้วยเครื่องหมายเซมิโคลอน ; โดยจะนำคำสั่งทั้งหมดไว้บรรทัดเดียวกันเลย หรือแยกบรรทัดกันก็ได้เพื่อความสวยงามของโค้ด (ไม่มีผลกับขนาดของโปรแกรมหลังคอมไพล์) เช่น

```
void Mode(int pin) {
  pinMode(pin, OUTPUT);
}
void setup() {
  Mode(13);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไป 16

2.9.3.5. ส่วนอธิบายโปรแกรม (Program comments)

ส่วนอธิบายโปรแกรม หรือการคอมเมนต์โปรแกรมเป็นส่วนที่สำคัญอย่างมากที่จะช่วยให้ผู้ที่ไม่ได้เขียนโปรแกรม หรือเป็นผู้เขียนโปรแกรมเข้าใจโปรแกรมได้ง่ายขึ้นโดยอ่านจากคอมเมนต์ แทนการทำความเข้าใจโปรแกรมโดยอ่านแต่ละฟังก์ชัน ส่วนอธิบายโปรแกรม หรือส่วนคอมเมนต์นี้ จะไม่มีผลใดๆกับขนาดของโปรแกรมหลังคอมไพล์ เนื่องจากส่วนนี้จะถูกตัดทิ้งทั้งหมดเนื่องจากไม่ได้ถูกนำไปใช้งาน มีผลเพียงแค่ว่าไฟล์โค้ดโปรแกรมจะใหญ่ขึ้นมา หากมีการคอมเมนต์โค้ดเยอะๆ แต่ขนาดก็จะเพิ่มขึ้นตามตัวอักษร ดังนั้นการคอมเมนต์โค้ดจึงไม่คิดพื้นที่มากนัก แต่ผู้เขียนแนะนำให้คอมเมนต์โค้ดให้สั้น และกระชับ เพื่อให้เกิดความรวดเร็วในการทำความเข้าใจ และไม่ยาวจนต้องเลื่อนสกรีนไปทางขวาเพื่ออ่านคอมเมนต์เพิ่มเติมอีกการคอมเมนต์โค้ดมีอยู่ 2 รูปแบบ คือ เปิดด้วย /* และปิดด้วย */ เป็นการคอมเมนต์โค้ดแบบข้ามบรรทัด คือตราบใดที่ยังไม่มี */ตรงส่วนนั้นจะเป็นคอมเมนต์ทั้งหมด เช่น

```
/*  
*/  
void setup() { ... }
```

และแบบที่ 2 เป็นการคอมเมนต์บรรทัดเดียว คือเปิดด้วยเครื่องหมาย // และปิดด้วยการขึ้นบรรทัดใหม่ เช่น

```
void setup() {  
  pinMode(13, OUTPUT); // Set pin 13 to output  
}
```

2.9.4. ชนิดของข้อมูล

ชนิดของข้อมูลใน Arduino มีชนิดของข้อมูลให้เลือกใช้ดังนี้

ตารางที่ 2.1 แสดงชนิดของข้อมูล การเก็บข้อมูล และ ขนาด ของชนิดข้อมูลที่ใช้ในโปรแกรม Arduino เบื้องต้น

ชนิดข้อมูล	การเก็บข้อมูล	ขนาด
boolean	จริง (True) หรือ เท็จ (False)	1 บิต
char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ -128 ถึง 127
unsigned char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
byte	ไบต์	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ -32,768 ถึง 32,767
unsigned int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 65,535 ($2^{16} - 1$)
long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
unsigned long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 4,294,967,295 ($2^{32} - 1$)
float	ตัวเลขทศนิยมใช้ในการคำนวณ	4 ไบต์ ใส่ค่าได้ตั้งแต่ $3.4028235E+38$ ถึง $-3.4028235E+38$ มีทศนิยมได้ 6 ถึง 7 ตำแหน่ง
String	ข้อความ	ไม่ระบุ

2.10 American Standard Code for Information Interchange (ASCII)

แอสกี(ASCII) หรือ รหัสมาตรฐานของสหรัฐอเมริกาเพื่อการแลกเปลี่ยนสารสนเทศ เป็นรหัสอักขระที่ประกอบด้วยอักษรละติน เลขอารบิก เครื่องหมายวรรคตอน และสัญลักษณ์ต่างๆ รหัสแอสกีมีใช้ในระบบคอมพิวเตอร์ และเครื่องมือสื่อสารแบบดิจิทัลต่างๆ พัฒนาขึ้นโดยคณะกรรมการ X3 ซึ่งอยู่ภายใต้การดูแลของสมาคมมาตรฐานอเมริกา (American Standards Association) ภายหลังกลายเป็น สถาบันมาตรฐานแห่งชาติอเมริกา (American National Standard Institute : ANSI) ในปี ค.ศ. 1969 โดยเริ่มต้นใช้ครั้งแรกในปี ค.ศ. 1967 ซึ่งมีอักขระทั้งหมด 128 ตัว (7 บิต) โดยจะมี 33 ตัวที่ไม่แสดงผล (unprintable/control character) โดยแต่ละรหัสจะแทนด้วยตัวอักษรหนึ่งตัว เช่น รหัส 65 (เลขฐานสิบ) ใช้แทนอักษรเอ (A) พิมพ์ใหญ่ เป็นต้น

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	(NULL)	32	20	(SPACE)	64	40	@	96	60	`
1	1	(START OF HEADING)	33	21	!	65	41	A	97	61	a
2	2	(START OF TEXT)	34	22	"	66	42	B	98	62	b
3	3	(END OF TEXT)	35	23	#	67	43	C	99	63	c
4	4	(END OF TRANSMISSION)	36	24	\$	68	44	D	100	64	d
5	5	(ENQUIRY)	37	25	%	69	45	E	101	65	e
6	6	(ACKNOWLEDGE)	38	26	&	70	46	F	102	66	f
7	7	(BELL)	39	27	'	71	47	G	103	67	g
8	8	(BACKSPACE)	40	28	(72	48	H	104	68	h
9	9	(HORIZONTAL TAB)	41	29)	73	49	I	105	69	i
10	A	(LINE FEED)	42	2A	*	74	4A	J	106	6A	j
11	B	(VERTICAL TAB)	43	2B	+	75	4B	K	107	6B	k
12	C	(FORM FEED)	44	2C	,	76	4C	L	108	6C	l
13	D	(CARRIAGE RETURN)	45	2D	-	77	4D	M	109	6D	m
14	E	(SHIFT OUT)	46	2E	.	78	4E	N	110	6E	n
15	F	(SHIFT IN)	47	2F	/	79	4F	O	111	6F	o
16	10	(DATA LINK ESCAPE)	48	30	0	80	50	P	112	70	p
17	11	(DEVICE CONTROL 1)	49	31	1	81	51	Q	113	71	q
18	12	(DEVICE CONTROL 2)	50	32	2	82	52	R	114	72	r
19	13	(DEVICE CONTROL 3)	51	33	3	83	53	S	115	73	s
20	14	(DEVICE CONTROL 4)	52	34	4	84	54	T	116	74	t
21	15	(NEGATIVE ACKNOWLEDGE)	53	35	5	85	55	U	117	75	u
22	16	(SYNCHRONOUS IDLE)	54	36	6	86	56	V	118	76	v
23	17	(END OF TRANS. BLOCK)	55	37	7	87	57	W	119	77	w
24	18	(CANCEL)	56	38	8	88	58	X	120	78	x
25	19	(END OF MEDIUM)	57	39	9	89	59	Y	121	79	y
26	1A	(SUBSTITUTE)	58	3A	:	90	5A	Z	122	7A	z
27	1B	(ESCAPE)	59	3B	;	91	5B	[123	7B	{
28	1C	(FILE SEPARATOR)	60	3C	<	92	5C	\	124	7C	
29	1D	(GROUP SEPARATOR)	61	3D	=	93	5D]	125	7D	}
30	1E	(RECORD SEPARATOR)	62	3E	>	94	5E	^	126	7E	~
31	1F	(UNIT SEPARATOR)	63	3F	?	95	5F	_	127	7F	[DEL]

รูปที่ 2.23 ตาราง ASCII Code

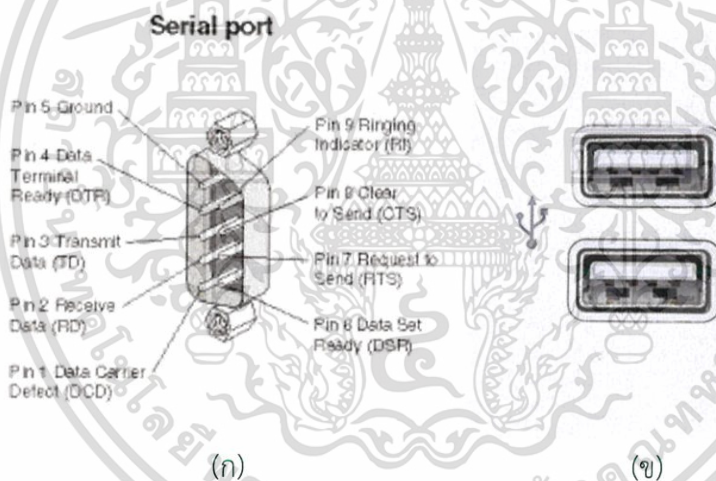
2.11 Serial Port

พอร์ตอนุกรม ในการสื่อสารข้อมูลนั้นพอร์ตอนุกรมจะมีความเร็วในการสื่อสารที่ช้ากว่าแบบขนาน เพราะการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลที่ละหลายๆ บิตพร้อมๆกันได้ แต่ข้อดีของการสื่อสารข้อมูลแบบอนุกรมคือ สามารถส่งข้อมูลได้ในระยะทางที่ไกลกว่าแบบขนาน และใช้สายสัญญาณที่น้อยกว่าการสื่อสารข้อมูลแบบขนาน

2.11.1 ประเภทของ Serial Port แบ่งตามลักษณะสัญญาณในการส่งแบ่งได้ 2 แบบ คือ

- 1.) การสื่อสารแบบซิงโครนัส (Synchronous) เป็นการสื่อสารข้อมูลโดยใช้สัญญาณนาฬิกาในการควบคุมจังหวะของการรับส่งสัญญาณ
- 2.) การสื่อสารแบบอะซิงโครนัส (Asynchronous) เป็นการสื่อสารที่ใช้สายข้อมูลเพียงตัวเดียว จะใช้รูปแบบของการส่งข้อมูล(Bit Pattern) เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นตัวตรวจสอบความถูกต้องของข้อมูล และส่วนไหนเป็นส่วนปิดท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาคส่งและภาครับ

ปัจจุบันมีพอร์ตอนุกรมที่มีความเร็วสูงแล้ว ที่เรียกว่า USB (Universal Serial Bus) คือ พอร์ต หรือช่องทางในการเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์อื่นๆ ไม่ว่าจะเป็น Printer, Modem , Mouse, Keyboard, Digital Camera เป็นต้น



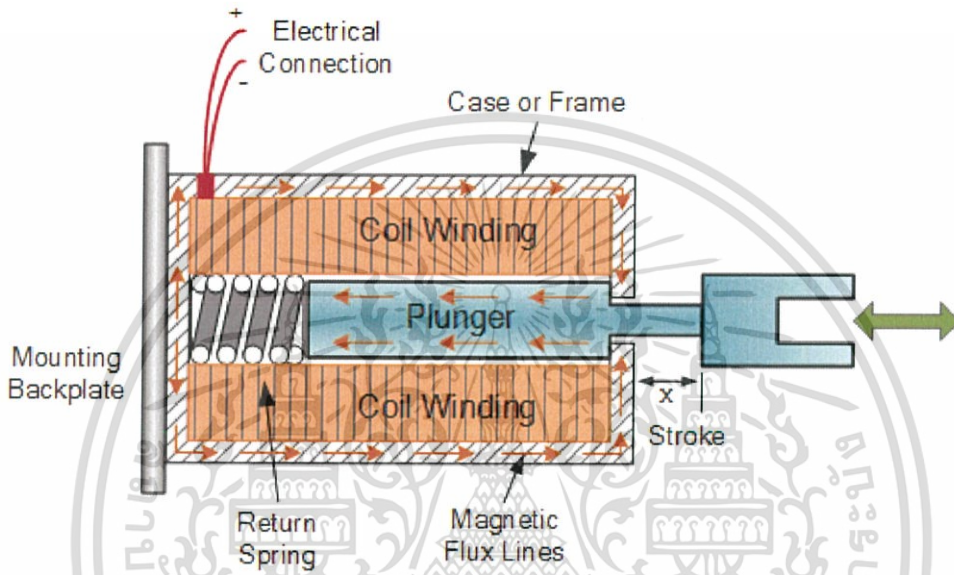
รูปที่ 2.24 ตัวอย่าง Serial Port

(ก) Serial Port

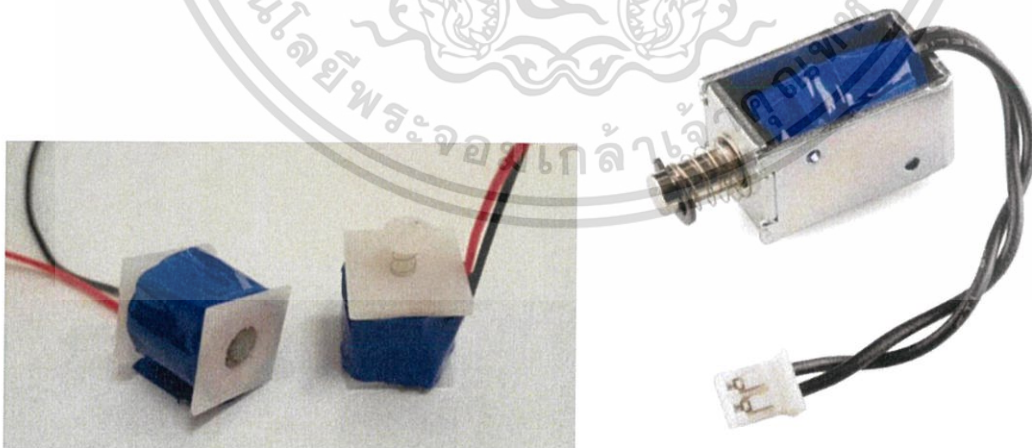
(ข) USB Port

2.12 ขดลวด Solenoids

ขดลวดโซลินอยด์ หมายถึง ขดลวดตัวนำที่มีฉนวนหุ้ม นำมาพันเป็นขดลวดวงกลมหลายรอบเรียงซ้อนกันเป็นทรงกระบอกโดยมีรัศมีคงที่เมื่อให้กระแสไฟฟ้าขนาดคงที่ไหลผ่านขดลวดโซลินอยด์ จะเกิดสนามแม่เหล็กรอบขดลวดโซลินอยด์ สนามแม่เหล็กบริเวณแกนกลางจะมีค่าเพิ่มขึ้นถ้าจำนวนรอบเพิ่มขึ้น ถ้าใส่แกนเหล็กอ่อนไว้ในขดลวดเมื่อมีกระแสไฟฟ้าผ่าน แกนเหล็กอ่อนจะกลายเป็นแม่เหล็ก แม่เหล็กที่เกิดขึ้นนี้ เรียกว่าแม่เหล็กไฟฟ้า ซึ่งค่าของสนามแม่เหล็กไฟฟ้า จะขึ้นอยู่กับค่าของ กระแสไฟฟ้าที่ไหลผ่าน และ จำนวนรอบของขดลวด



รูปที่ 2.25 โครงสร้างภายในขดลวด Solenoids



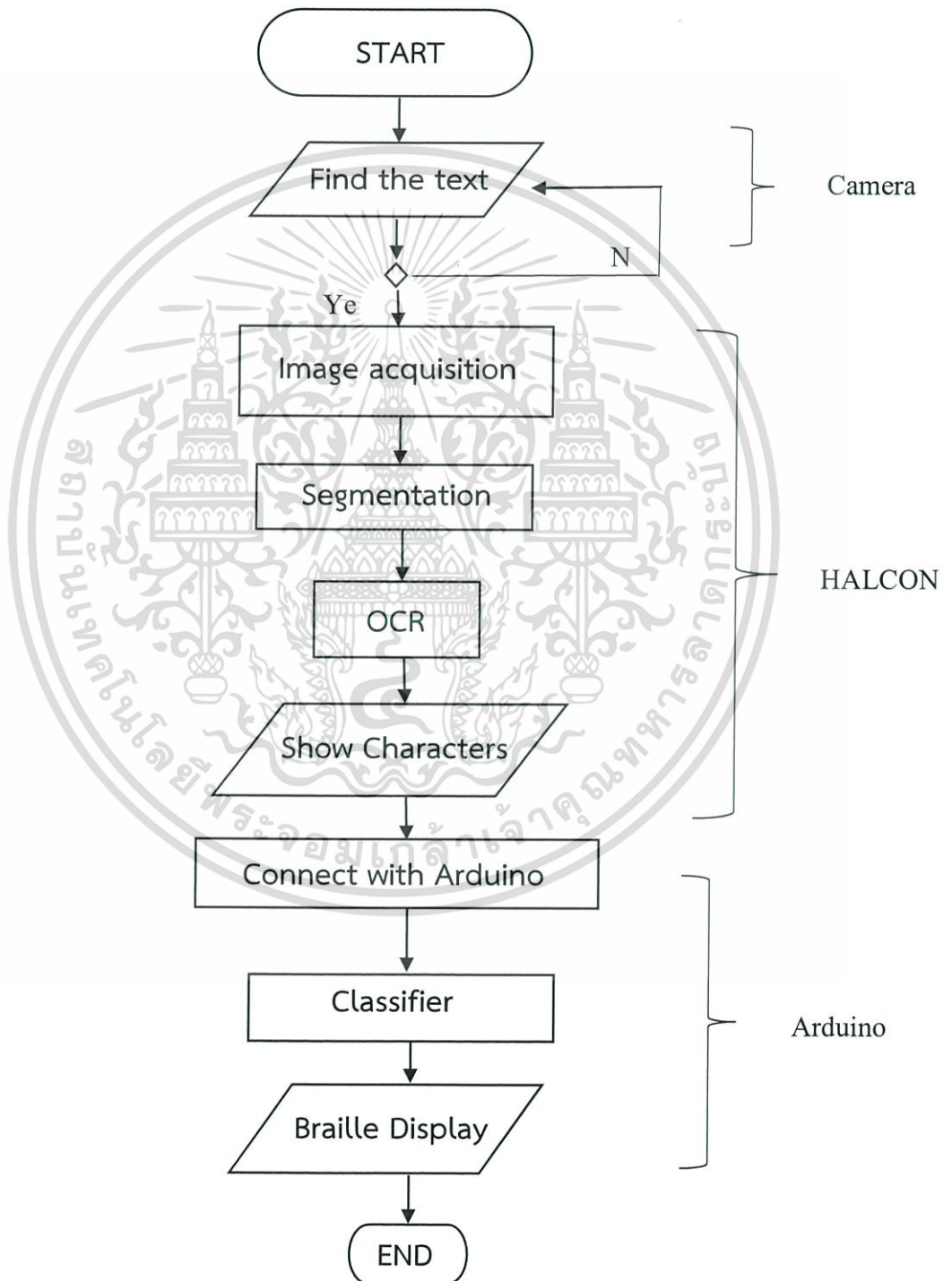
รูปที่ 2.26 ขดลวด Solenoids ที่ใช้ในการทดลอง

บทที่ 3

วิธีการดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

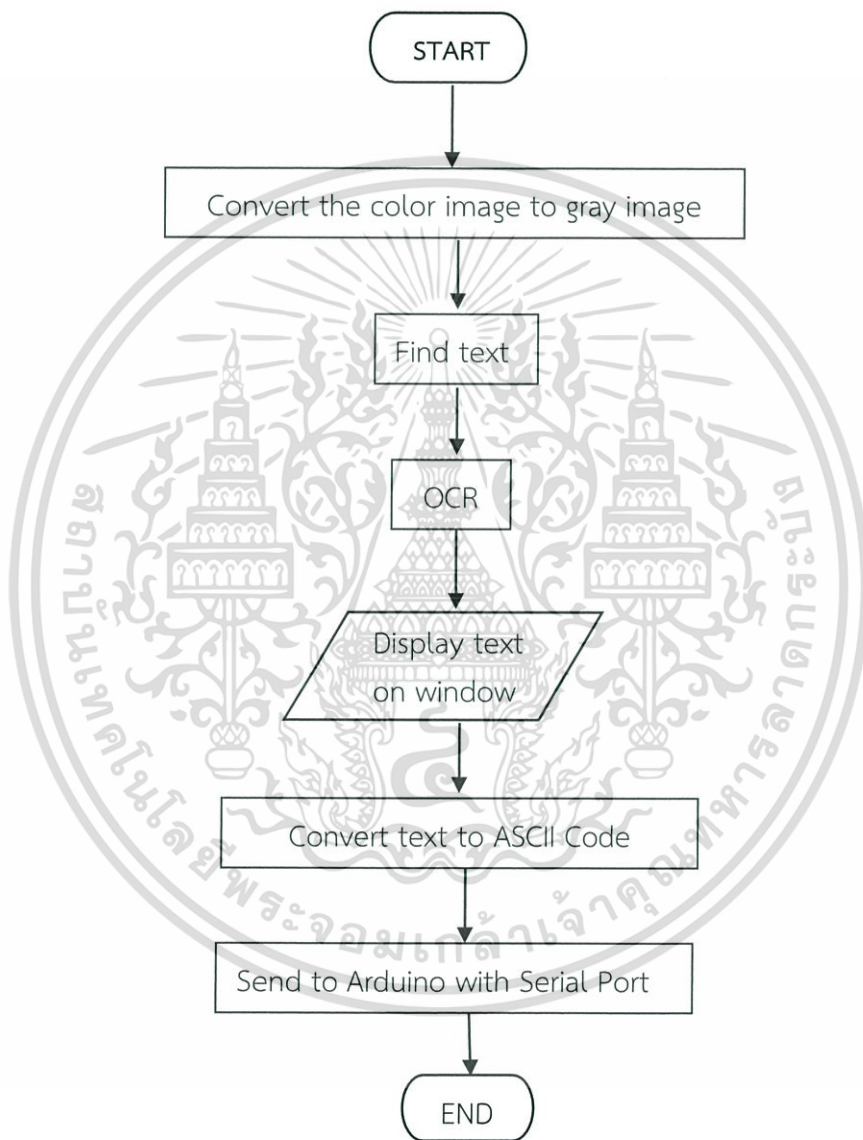
3.1.1 Flow Chart



3.2 การปฏิบัติงานบนโปรแกรม HALCON

HALCON มีการใช้งานในลักษณะ GUI (Graphic User Interface) เหมือนกับโปรแกรมประเภทวิชชวลซ์ (Visual Programming) ซึ่งทำให้ผู้ใช้สามารถติดต่อกับโปรแกรมได้ง่ายผ่านคอมพิวเตอร์ต่าง บนเมนูบาร์ทำได้โดยพิมพ์ guide ที่ช่อง “Enter Operator or Procedure” ในหน้าต่าง Operator Window โปรแกรม HALCON สามารถเชื่อมต่อกับกล้อง, รับภาพในโหมดต่างๆ และกำหนดค่าที่ซับซ้อนได้

3.2.1 ขั้นตอนการปฏิบัติการบนโปรแกรม HALCON



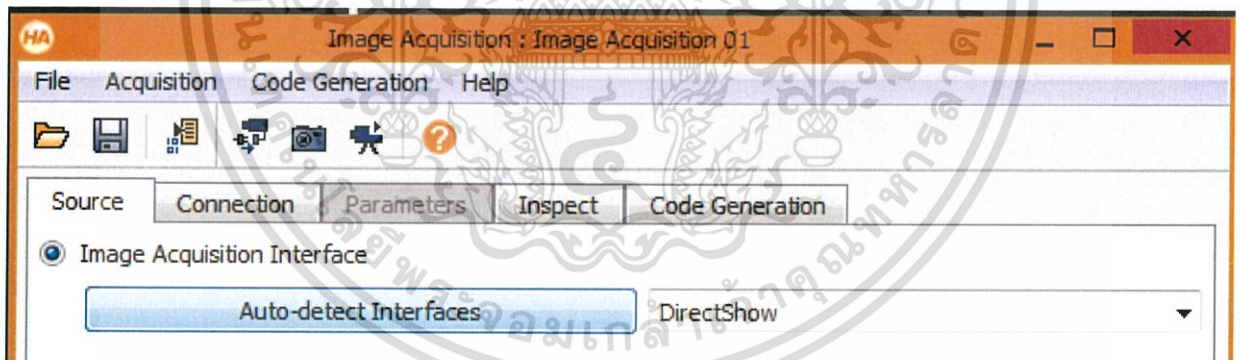
ตารางที่ 3.1 คำสั่งที่ใช้ในการปฏิบัติงานบนโปรแกรม HALCON พร้อมคำอธิบาย

Line	Code	คำอธิบาย
1	FontName := 'Industrial_0-9_NoRej'	เปิดใช้งานไฟล์ตัวอักษรที่ชื่อ 'Industrial_0-9_NoRej'
2	open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', '[0] USB Camera', 0, -1, AcqHandle)	ค้นหากล้องที่เชื่อมต่อกับโปรแกรม HALCON แล้วเก็บค่าไว้ในชื่อ "AcqHandle"
3	grab_image_start (AcqHandle, -1)	จับภาพจากกล้อง
4	open_serial ('COM3', SerialHandle)	เปิดใช้งานserial port กับ port ที่ต้องการสื่อสารด้วย 'COM3' ในที่นี้คือ Arduino
5	set_serial_param (SerialHandle, 9600, 8, 'none', 'none', 1, 200, 200)	กำหนดค่า ใน serial port เช่น Buad rate
6	while (true)	คำสั่งที่ใช้นซ้ำ โดยจะตรวจสอบเงื่อนไขก่อนและทำงานเมื่อเป็นจริง ตามเงื่อนไข
7	read_serial (SerialHandle, 1, Data)	อ่านค่าจาก Serial port
8	clear_serial (SerialHandle, 'input')	เคลียร์คำสั่งการทำงาน
9	if (Data = 0)	ถ้า ข้อมูลเริ่มต้นจาก 0
10	grab_image_async (Image, AcqHandle, -1)	การจับภาพเป็นช่วงๆ
11	rgb1_to_gray (Image, Bottle)	ปรับปรุงภาพจากภาพสีเป็นภาพgray scale
12	get_image_size (Bottle, Width, Height)	รับค่า ขนาดของรูปภาพ
13	dev_open_window (0, 0, Width, Height, 'black', WindowID)	จอแสดงผลพื้นหลังเป็นสีดำ
14	set_display_font (WindowID, 24, 'mono', 'true', 'false')	ตั้งค่า ตัวอักษรที่จะแสดงผล
15	dev_display (Bottle)	แสดงผลภาพที่จับได้
16	create_text_model_reader('auto', 'Document_0-9A-Z_NoRej.omc', TextModel)	สร้าง ตัวอักษร ที่อ่านได้อัตโนมัติ
17	set_text_model_param(TextModel, 'min_stroke_width', 4)	ตั้งค่าความหนาน้อยสุดของตัวอักษร
18	set_text_model_param(TextModel, 'max_stroke_width', 'auto')	ตั้งค่าความหนามากสุดของตัวอักษร
19	find_text(Bottle, TextModel, TextResultID)	ค้นหาตัวอักษรบนภาพ
20	get_text_object (Characters, TextResultID, 'all_lines')	รับค่าตัวอักษรจากภาพมาเก็บไว้
21	dev_display (Characters)	แสดงผลตัวอักษร

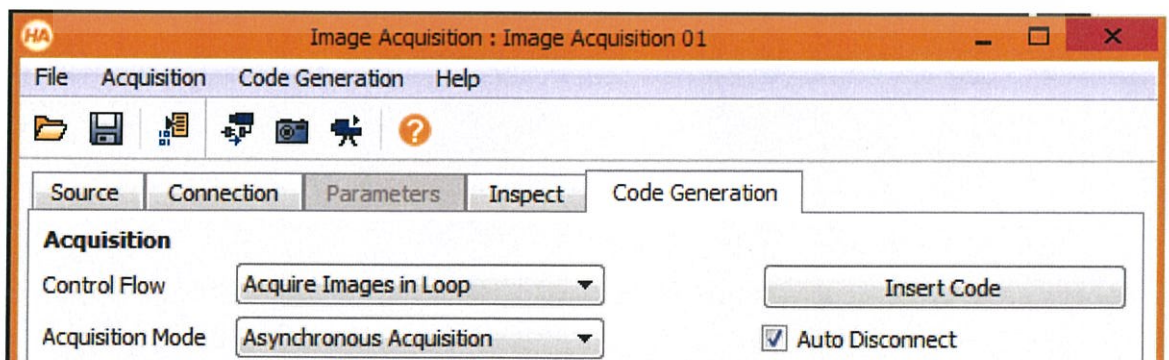
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 24

22	get_text_result(TextResultID,'class', Classes)	รับค่าข้อความที่อ่านค่าได้
23	area_center(Characters,Area,Row,Column)	กำหนดพื้นที่/บริเวณที่ตัวอักษรควรปรากฏบนจอ
24	disp_message (WindowID, Classes, 'image', 80, Column - 3, 'green', 'false')	แสดงผลจากการOCRบนหน้าจอ
25	tuple_ord(Classes,asd)	การเปลี่ยน string ให้เป็น ASCII Code
26	if(asd>1) write_serial(SerialHandle, asd)	ส่งค่าตามเงื่อนไขที่กำหนดไปยังบอร์ด Arduino
27	endif	จบการทำงาน if
28	clear_text_result (TextResultID)	ล้างตัวแปร TextResultID
29	clear_text_model (TextModel)	ล้างตัวแปร TextModel
30	endif	จบการทำงาน if
31	endwhile	จบการทำงาน while
32	close_framegrabber (AcqHandle)	ปิดการเชื่อมต่อกับกล้อง

ขั้นแรกต่ออุปกรณ์ ซึ่งในที่นี้เราใช้กล้อง USB Camera เป็นแหล่งที่มาของภาพที่จะ เชื่อมต่อกับโปรแกรม HALCON การที่จะส่ง/รับข้อมูลได้จะต้อง เริ่มจากขั้นตอน Image Acquisition ซึ่งจะช่วยให้โปรแกรม กับกล้องสื่อสารกันได้ โดยใช้ เมนู Assistants >> Open New Image Acquisition



(ก)



(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไป 25

```

Program Window - main* () - Main Thread: 2420
main (:::):
1 * Image Acquisition 01: Code generated by Image Acquisition 01
2 open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', '[0] USB Camera', 0, -1, AcqHandle)
3 grab_image_start (AcqHandle, -1)
4 while (true)
5     grab_image_async (Image, AcqHandle, -1)
6     * Image Acquisition 01: Do something
7 endwhile
8 close_framegrabber (AcqHandle)

```

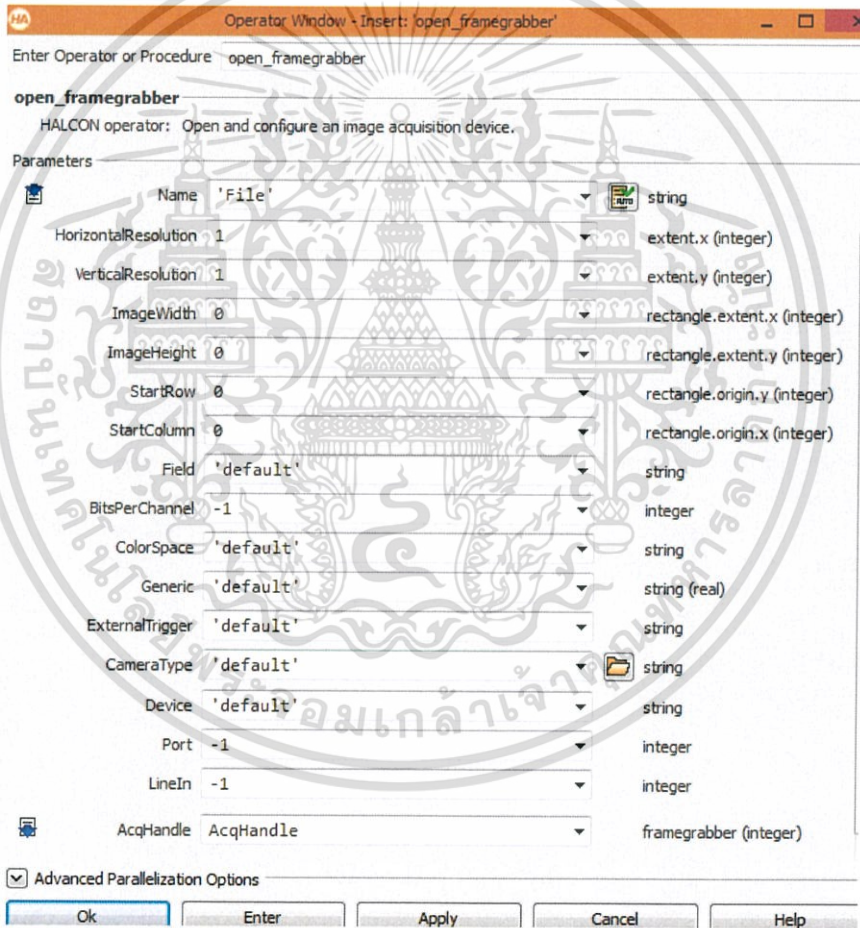
(ค)

รูปที่3.1 แสดงขั้นตอนการเชื่อมต่อระหว่างกล้องกับโปรแกรม HALCON (Image Acquisition)

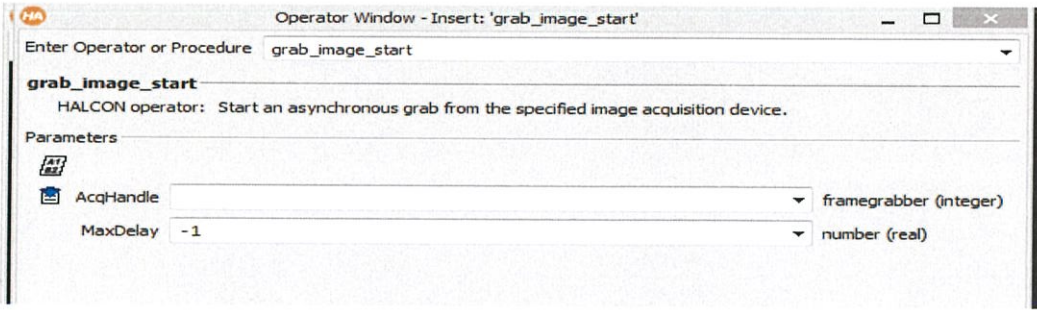
(ก) ค้นหากล้องที่เชื่อมต่อกับโปรแกรมHALCON

(ข) ประมวลผลออกมาเป็น code

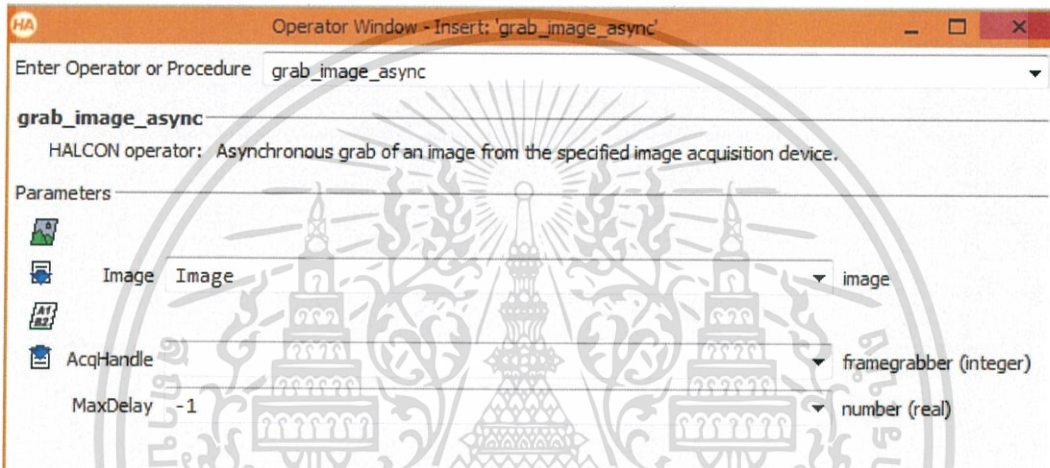
(ค) Code ที่ได้จากการ Image Acquisition



รูปที่3.2 ชุดคำสั่งแสดงพารามิเตอร์ในส่วนของ open_framgrabber

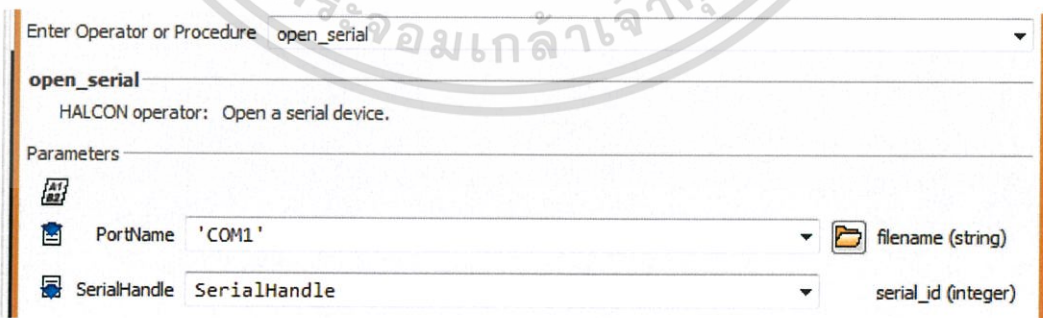


รูปที่ 3.3 ชุดคำสั่งแสดงพารามิเตอร์ ของการรับภาพจากกล้องที่เชื่อมต่อ

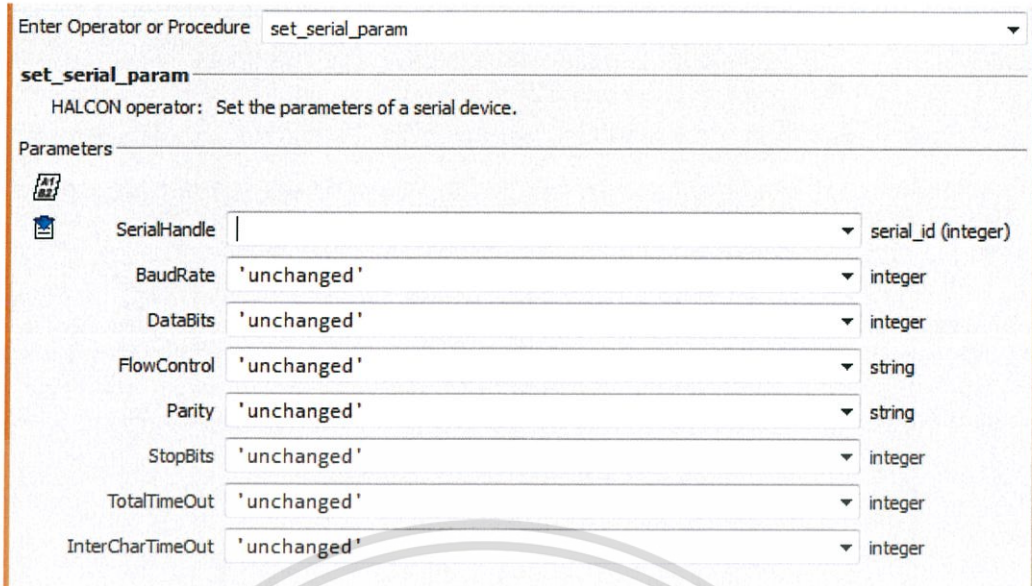


รูปที่ 3.4 แสดงค่าพารามิเตอร์ของการรับภาพจากกล้องที่เชื่อมต่อที่จับภาพแบบไม่พร้อมกัน

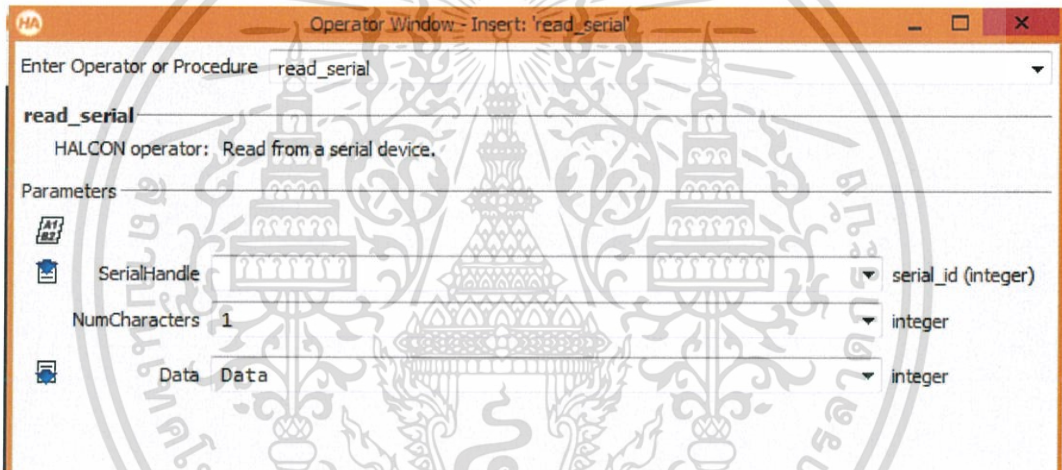
การทดลองครั้งนี้ได้ใช้โปรแกรม เชื่อมต่อกับ บอร์ด Arduino โดยเราจะติดต่อ กับบอร์ด โดย ใช้ Serial Port



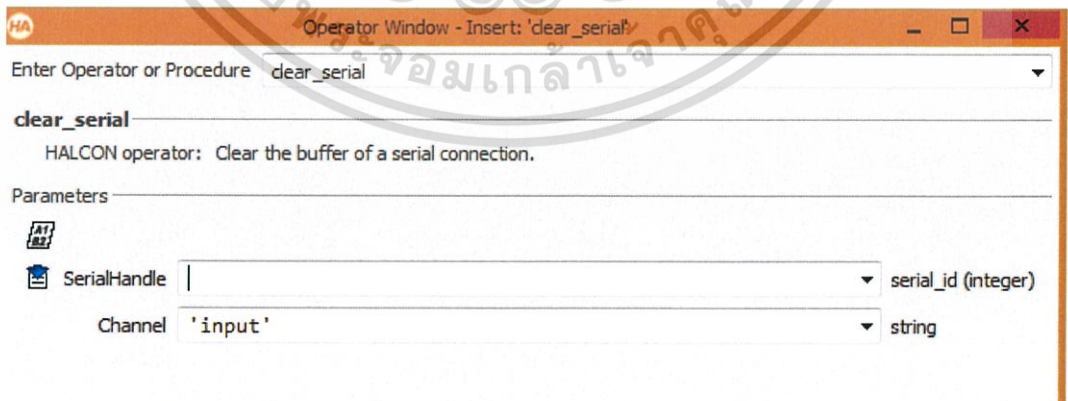
(ก)



(จ)



(ค)



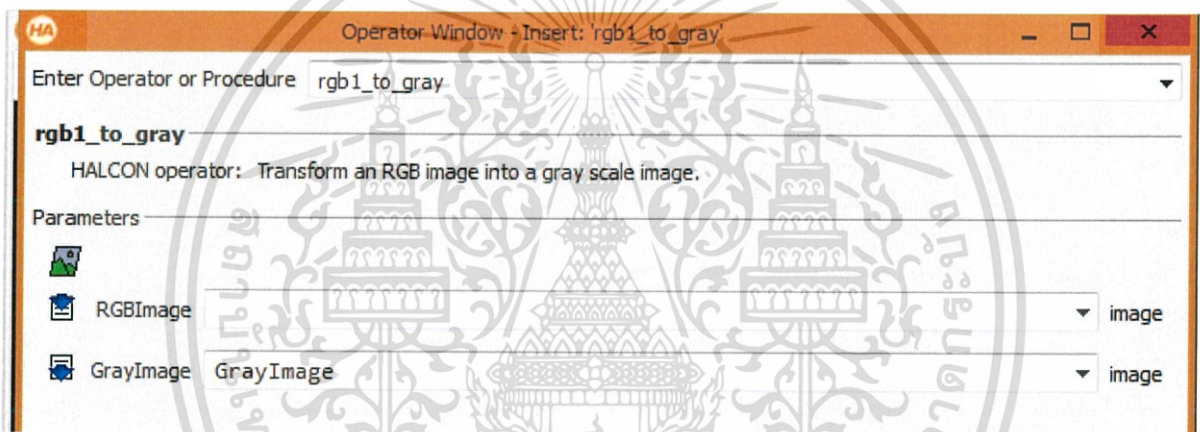
(ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 28

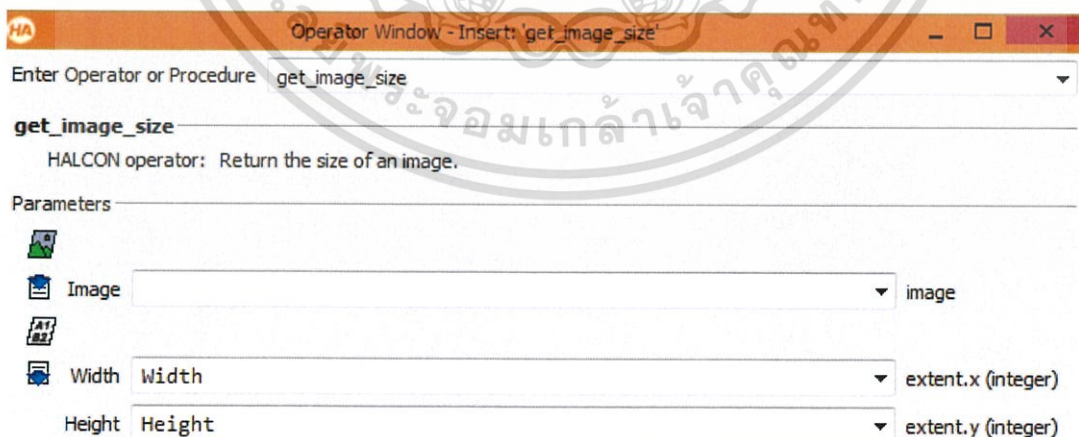
รูปที่3.5 ขั้นตอนการเชื่อมต่อและตั้งค่า Serial Port

- (ก) ชุดคำสั่งเชื่อมต่อกับ Serial Port
- (ข) ชุดคำสั่งกำหนดค่าที่เชื่อมต่อกับ Serial Port
- (ค) ชุดคำสั่งการตั้งค่าเพื่ออ่านข้อมูลจาก Serial Port
- (ง) ชุดคำสั่งล้างค่าที่รับมา ใน Serial Port

ขบวนการก่อนประมวลผล เป็นขบวนการที่เพิ่มโอกาสประสบความสำเร็จของขั้นตอนการประมวลผล โดยภาพเริ่มต้นที่รับมาจะเป็นภาพสี (RGB Image) ก่อนการแยกส่วนของภาพ จะต้องปรับภาพให้เป็นภาพ Grey Scale เพราะการแยกส่วนของภาพโดยอัตโนมัติเป็นงานที่ยากลำบากในการประมวลผลภาพดิจิทัลจึงต้องแปลงข้อมูลดิบให้อยู่ในรูปแบบที่เหมาะสมเพื่อที่เราจะสามารถตรวจสอบความถูกต้องว่า โปรแกรม สามารถ แปล/อ่านข้อความจากภาพได้ถูกต้องหรือไม่ จึงต้องแสดงผลที่โปรแกรมอ่านค่าให้เป็นตัวอักษรบนหน้าต่าง โปรแกรม HALCON



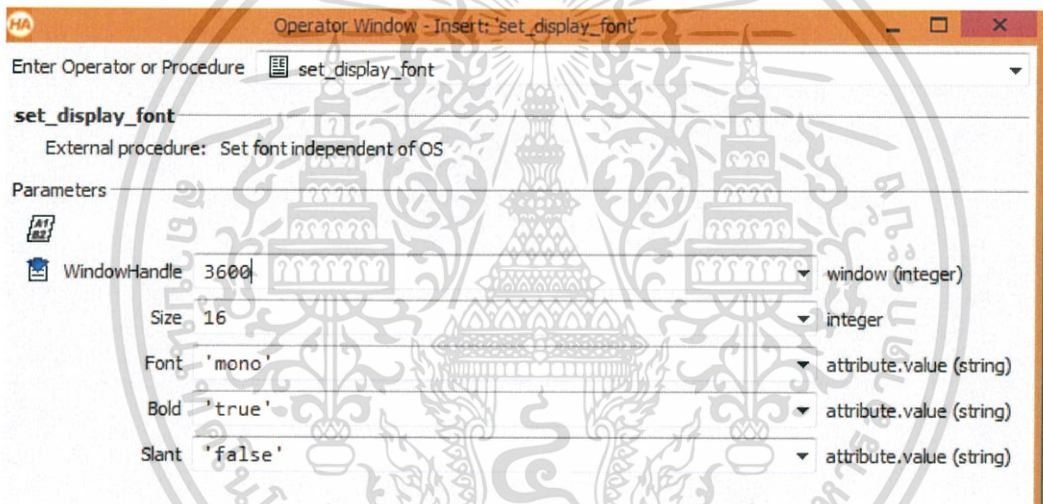
รูปที่3.6 แสดงชุดคำสั่งการปรับปรุงภาพ การเปลี่ยนภาพสีให้เป็นภาพ Grey Scale



(ก)



(ข)

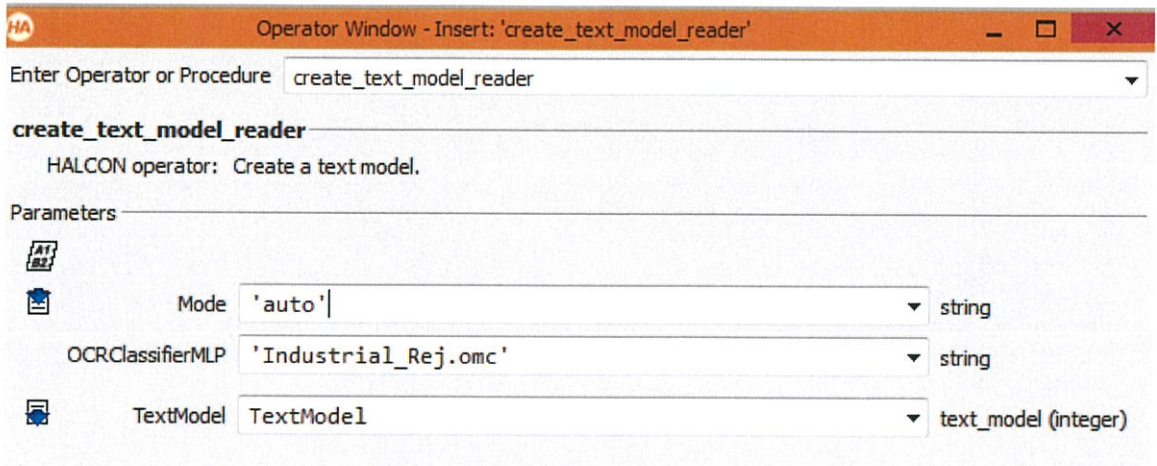


(ค)

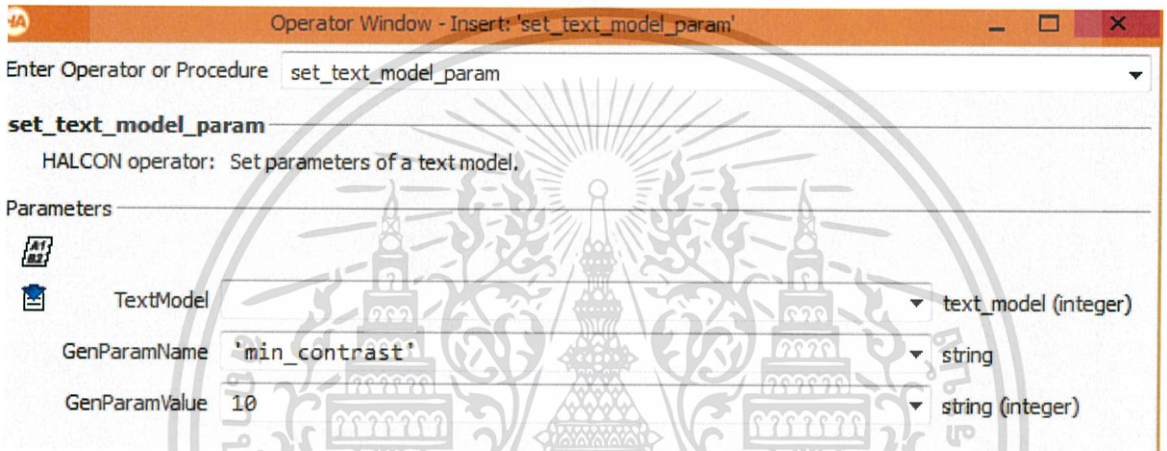
รูปที่3.7 แสดงการกำหนดค่าตัวอักษรที่จะแสดงผลบนหน้าต่าง

- (ก) ชุดคำสั่งรับข้อมูลภาพเข้ามา
- (ข) ชุดคำสั่งกำหนดค่าพารามิเตอร์ของจอแสดงผล
- (ค) ชุดคำสั่งกำหนดค่าพารามิเตอร์ ของตัวอักษรที่จะแสดงผล

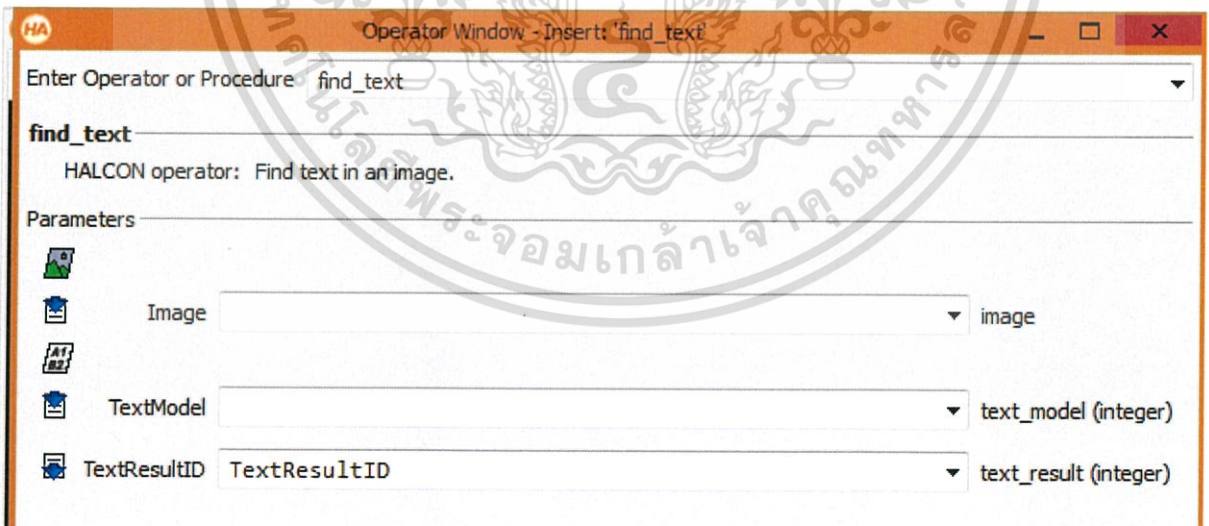
ต่อไปเป็นกระบวนการตั้งค่าเพื่อกำหนดขอบเขต ลักษณะ ของตัวอักษรบนภาพเพื่อที่จะทำการ OCR ระบบในสมัยเริ่มแรกต้องการการฝึกฝน (สอนตัวอย่างที่รู้จักแล้วของแต่ละตัวอักษร) เพื่ออ่านพอนต์หนึ่ง ๆ ปัจจุบัน ระบบ "อัจฉริยะ" ที่สามารถอ่านพอนต์ส่วนใหญ่ได้แม่นยำสูงนั้นสามารถพบได้ทั่วไป โดยในครั้งนี้นะเราจะใช้ พอนต์ที่มีชื่อว่า "Industrial_0-9_NoRej" นอกจากนี้ เรายังต้องกำหนด ความหนาของตัวอักษรเริ่มต้น จำนวนตัวอักษรขั้นต่ำในหนึ่งบรรทัด เป็นต้น



(ก)



(ข)



(ค)

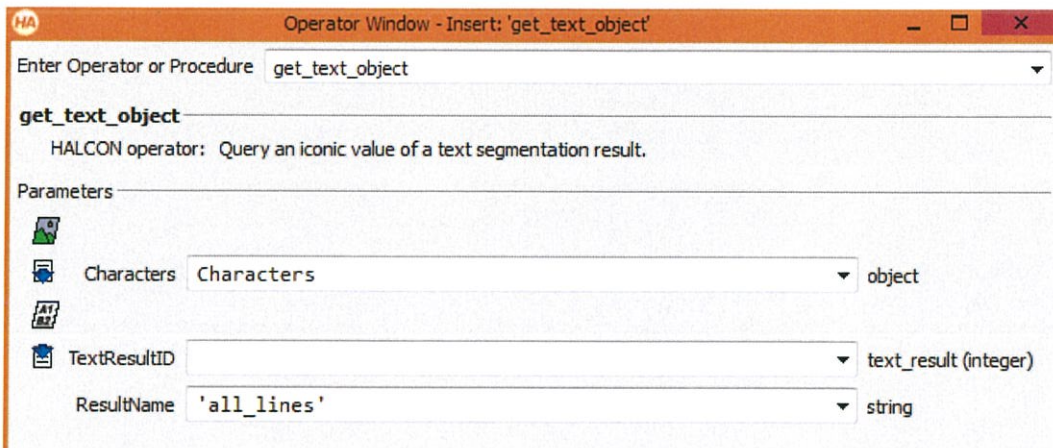
รูปที่ 3.8 แสดงขั้นตอนการค้นหาคำอักขรบนรูปภาพ และการตั้งค่า Parameter

(ก) ชุดคำสั่งการกำหนดเลือกใช้ไฟล์ OCR ที่ชื่อ Industrial_0-9_NoRej

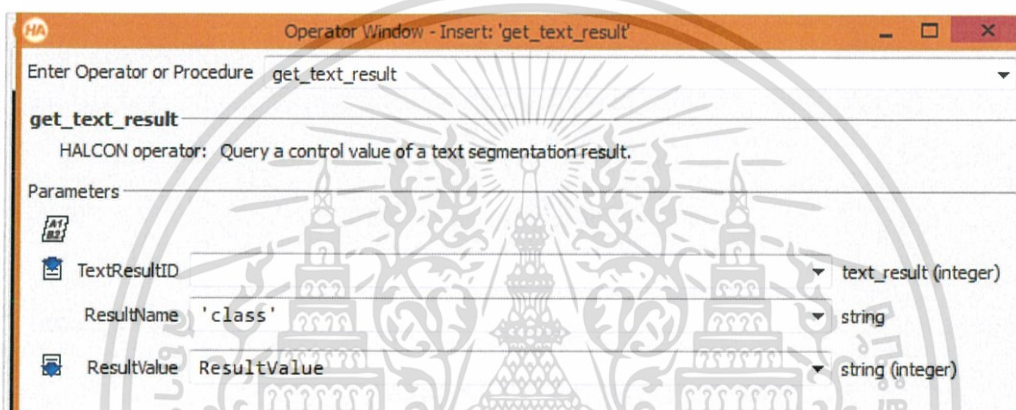
(ข) ชุดคำสั่งการกำหนดขอบเขต ของค่าพารามิเตอร์

(ค) ชุดคำสั่งแสดงชุดคำสั่งการค้นหาคำอักขรบนรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร หากมีข้อผิดพลาดประการใด ขออภัยและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 31

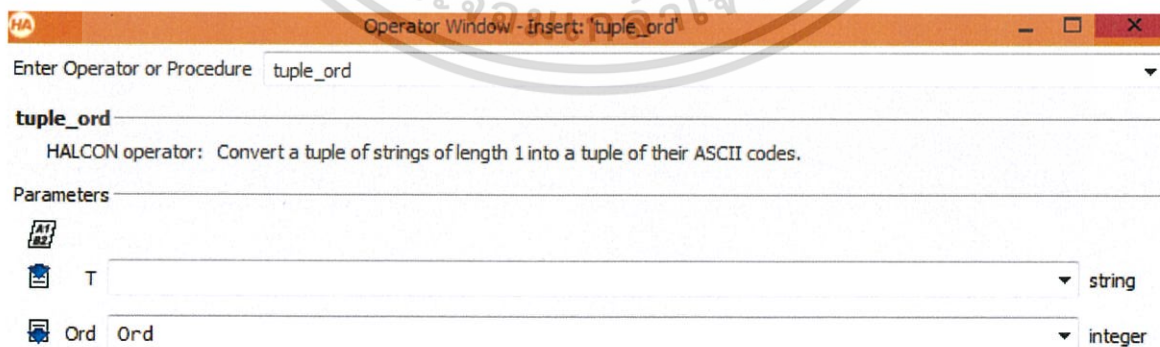


รูปที่3.9 แสดงชุดคำสั่งการแปลงข้อมูลเพื่อเก็บค่าตัวอักษรที่ได้รับจากวัตถุ



รูปที่3.10 แสดงชุดคำสั่งการแปลงข้อมูลเพื่อเก็บค่าตัวอักษรที่ได้รับจากการทำการแบ่งข้อความ (Text segmentation)

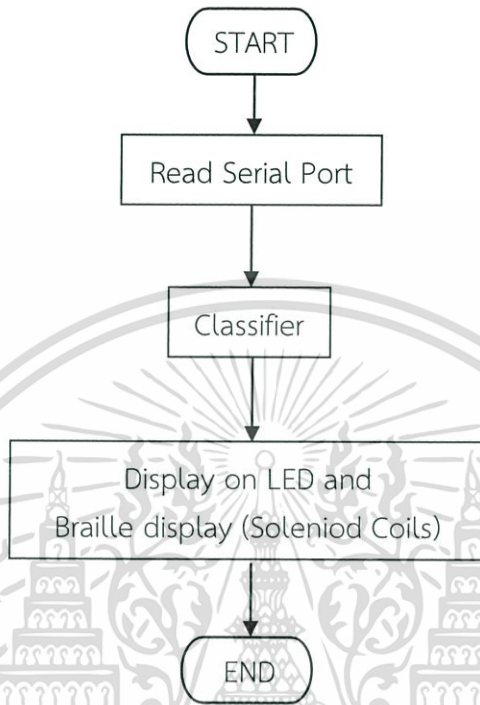
เราไม่สามารถส่งข้อมูลแบบ String จากโปรแกรม HALCON ไปยังบอร์ด Arduino ได้ เราจึงต้องแปลงข้อมูลให้เป็นรหัส ASCII เพื่อติดต่อสื่อสาร - ส่งข้อมูล ซึ่งเป็นรหัสมาตรฐานที่ไว้ใช้ควบคุมคอมพิวเตอร์



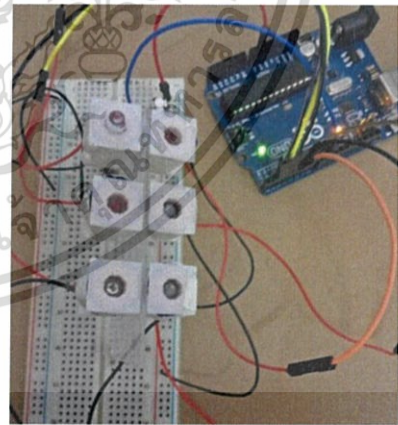
รูปที่3.11 การแปลงข้อมูลจาก ข้อมูล 'String' เป็น ASCII Code

3.3 การปฏิบัติงานบน Arduino

ในงานปริญญานิพนธ์นี้ได้ใช้บอร์ด Arduino รับค่าจากโปรแกรม HALCON แล้วใช้ I/O Port ของ Arduino ในการจ่ายไฟให้แก่ ขดลวด Solenoids เพื่อให้ขดลวด Solenoids ทำงาน โดยเขียนโปรแกรมลงบน Arduino IDE ซึ่งสามารถดูรายละเอียดของโปรแกรมได้ในส่วนของภาคผนวก ข.



รูปที่3.12



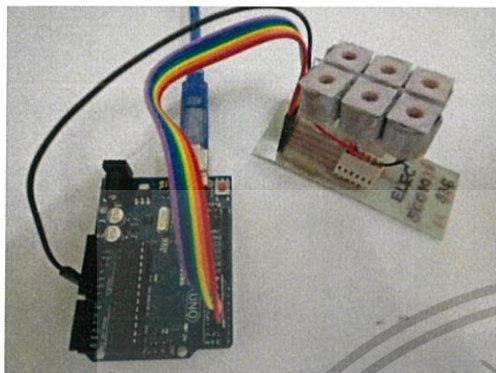
รูปที่3.13

รูปที่3.12 บอร์ด Arduinoที่ใช้ในการทดลอง

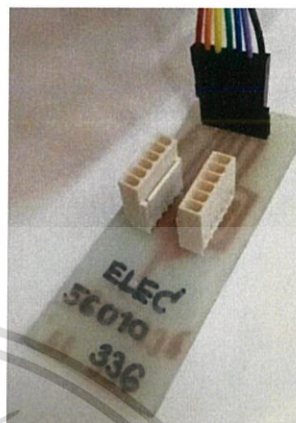
รูปที่3.13 บอร์ด Arduino และ ขดลวดSolenoids ที่ใช้ในการทดลอง

3.4 ส่วนแสดงผลอักษรเบรลล์

ในส่วนของการแสดงผลอักษรเบรลล์ ได้ใช้ ขอลวดโซลินอยด์ 6 ตัวต่อกับ Connector ที่เชื่อมไปยังบอร์ด Arduino



รูปที่ 3.14 แสดงส่วนแสดงผลอักษรเบรลล์



รูปที่ 3.15 แสดงส่วนเชื่อมต่อระหว่างขดลวดโซลินอยด์ กับ บอร์ด Arduino

บทที่ 4

การทดลอง และ ผลการทดลอง

4.1 การทดลอง

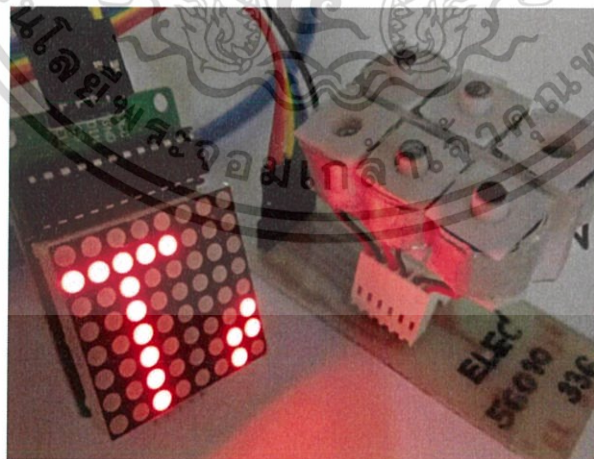
นำกล้องที่เชื่อมต่อกับโปรแกรม HALCON ไปจับตัวอักษรบนวัตถุ และส่งข้อมูลไปยังบอร์ด Arduino โดย I/O Port ของบอร์ด Arduino จะทำให้ขดลวด Solenoids ทำงาน เกิดเป็นจุดนูน ตามลักษณะอักษร เบลล์ เมื่อสัมผัส และทำให้แผงหลอด LEDs Dot Matrix ทำงานตามที่ได้เขียนคำสั่งควบคุมไว้

4.2 ผลการทดลอง

จากการทดลองเมื่อกำลังจับตัวอักษรบนวัตถุได้แล้ว โปรแกรมสามารถแยกแยะตัวอักษรภาษาอังกฤษ และสามารถแสดงผลเป็นไปตามคำสั่งที่เขียนไว้ใน Arduino โดยมีรูปแบบลักษณะเป็นอักษรเบลล์ใน ภาษาอังกฤษ มีความเร็วในการแสดงผลเท่ากับ 200 ms.







รูปที่4.1 แสดงผลโปรแกรม HALCON สามารถแยกแยะตัวอักษรภาษาอังกฤษ


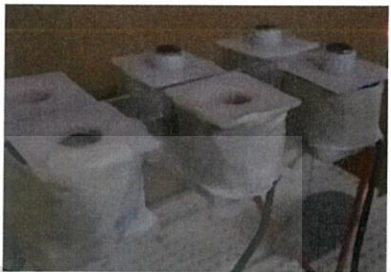



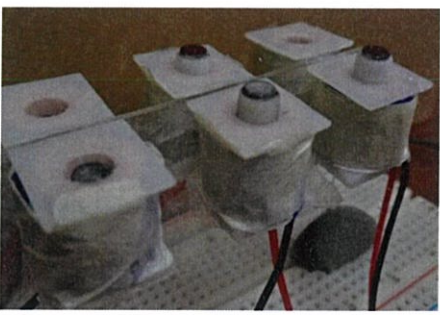


รูปที่4.2 LEDs Dot Matrix และ ขดลวด Solenoids ที่ทำงานตามที่ได้เขียนคำสั่งควบคุมไว้



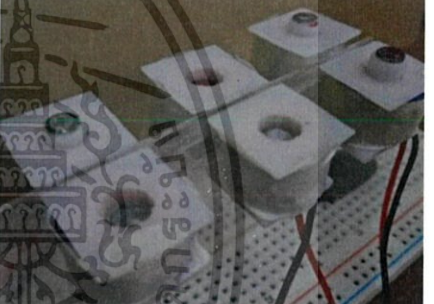
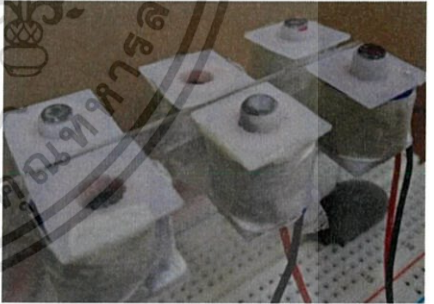

ตารางที่4.1 แสดงผลการทดลองเมื่อนำกล่องส่งไปยังภาพที่มีตัวหนังสือภาษาอังกฤษ โดยแสดงผลสองแบบคือ แบบLEDs Dot Matrix และ แบบแสดงผลด้วยขดลวด Solenoids โดยแยกทีละตัวอักษร






ตัวอักษร	การแสดงผลด้วย ขดลวด Solenoids	
A	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
B	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
C	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ทำงาน Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
D	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ทำงาน Solenoids ตำแหน่งที่ 5 : ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไป 36






E	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน Solenoids ตำแหน่งที่ 5 : ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
F	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ทำงาน Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
G	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ทำงาน Solenoids ตำแหน่งที่ 5 : ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
H	Solenoids ตำแหน่งที่ 1 : ทำงาน Solenoids ตำแหน่งที่ 2 : ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน Solenoids ตำแหน่งที่ 5 : ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
I	Solenoids ตำแหน่งที่ 1 : ไม่ทำงาน Solenoids ตำแหน่งที่ 2 : ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ทำงาน Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	
J	Solenoids ตำแหน่งที่ 1 : ไม่ทำงาน Solenoids ตำแหน่งที่ 2 : ทำงาน Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน Solenoids ตำแหน่งที่ 4 : ทำงาน Solenoids ตำแหน่งที่ 5 : ทำงาน Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน	


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ทางการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไป 37

K	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
L	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
M	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
N	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
O	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	

P	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
Q	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
R	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
S	<p>Solenoids ตำแหน่งที่ 1 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
T	<p>Solenoids ตำแหน่งที่ 1 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 39

U	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ทำงาน</p>	
V	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ทำงาน</p>	
W	<p>Solenoids ตำแหน่งที่ 1 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ทำงาน</p>	
X	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	
Y	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ไม่ทำงาน</p>	

Z	<p>Solenoids ตำแหน่งที่ 1 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 2 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 3 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 4 : ไม่ทำงาน</p> <p>Solenoids ตำแหน่งที่ 5 : ทำงาน</p> <p>Solenoids ตำแหน่งที่ 6 : ทำงาน</p>	
---	---	--



สรุปผลและวิจารณ์

5.1 สรุปผลการทดลอง

จากการทดลองเมื่อกำลังจับตัวอักษรบนวัตถุได้แล้ว ส่งสัญญาณภาพมายังโปรแกรม HALCON บนคอมพิวเตอร์ได้ โปรแกรมสามารถแยกแยะตัวอักษรภาษาอังกฤษ และส่งข้อมูลไปยังบอร์ด Arduino ที่เชื่อมต่อกับขดลวดโซลินอยด์ 6 ตัว โดยสามารถแสดงผลเป็นไปตามคำสั่งที่เขียนไว้ใน Arduino โดยมีรูปแบบลักษณะเป็นอักษรเบรลล์ในภาษาอังกฤษได้ โดยค่าความผิดพลาดขึ้นอยู่กับการจัดเรียงบรรทัด ความเข้มแสง การสะท้อนของแสง และ ความแตกต่างของสีพื้นหลังกับสีตัวอักษร

5.2 ปัญหาที่พบ

สำหรับปัญหาที่เกิดขึ้น เนื่องจากโปรแกรม HALCON เป็นโปรแกรมที่ไม่ได้มีคู่มือทั่วไปให้ศึกษาจึงต้องใช้เวลาในการศึกษาค้นคว้า การเขียนโปรแกรม และการแก้ไขปัญหา ค่อนข้างมาก ส่วนการเชื่อมต่อระหว่างโปรแกรม และ บอร์ด Arduino มีปัญหาเรื่องการติดต่อกับพอร์ตอนุกรม ในส่วนของบอร์ด Arduino และ ส่วนแสดงผลอักษรเบรลล์ ขดลวดโซลินอยด์ที่ใช้มีสนามแม่เหล็กไฟฟ้า เมื่อนำมาวางใกล้เกิดจึงการดึงดูดกัน ทำให้แสดงผลคลาดเคลื่อน อีกทั้งยังไม่สามารถทนแรงกดของนิ้วมือได้ จึงต้องเปลี่ยนชนิดของขดลวดซึ่งมีราคาสูงขึ้น นอกจากการโปรแกรมสามารถจับตัวอักษรได้ที่ละบรรทัด เพื่อป้องกันการอ่านและแยกแยะผิดพลาด

5.3 แนวทางในการพัฒนา

เขียนโปรแกรม HALCON ให้สามารถรับค่าตัวอักษรจากภาพที่มีจำนวนหลายบรรทัดได้ และส่งข้อมูลที่ละบรรทัดได้ ในส่วนของการแสดงผลควรเพิ่มจำนวนส่วนแสดงผลให้สามารถแสดงผลเป็นประโยค หรือหลายตัวอักษรพร้อมกัน โดยพัฒนาการเขียนโปรแกรม และเพิ่มวงจร เพื่อให้สามารถส่งข้อมูลเป็นแบบขนาน นอกจากนี้อาจทำให้สามารถพกพาได้โดยอาจใช้อุปกรณ์ที่ทำหน้าที่เป็นคอมพิวเตอร์ขนาดเล็กที่มีหน่วยประมวลผล และหน่วยความจำเพียงพอ เช่น Raspberry Pi, LattePanda

5.4 ประโยชน์ที่ได้รับ

- 1) มีความรู้ ความเข้าใจเกี่ยวกับการเขียนโปรแกรม HALCON และ โปรแกรม Arduino
- 2) มีความรู้ ความเข้าใจเกี่ยวกับการเชื่อมต่อพอร์ตบนคอมพิวเตอร์
- 3) มีความรู้ ความเข้าใจเกี่ยวกับการทำงานให้เป็นระบบ
- 4) เกิดการศึกษาค้นคว้า และ รู้จักการแก้ไขปัญหาต่างๆได้ด้วยตนเอง
- 5) เป็นแนวทางไปสู่การศึกษา ค้นคว้าวิจัย ในระดับการศึกษาที่สูงกว่าต่อไป

บรรณานุกรม

- [1] รศ.ดร.ชูชาติ พิณฑวิรุจน์. (พฤษภาคม 2555). การประมวลผลภาพดิจิทัลด้วยMatlab. ครั้งที่2. กรุงเทพฯ:คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [2] Maria Petrou, Panagiota Bosdogianni. (1999). Image processing: the fundamentals. Chichester : John Wiley.
- [3] John C. Russ. (2002). The image processing handbook. Boca Raton FL: CRC Press.
- [4] ณัชชา โสคติยานุรักษ์. 2542. “Digital Image Processing”. ครั้งที่1. กรุงเทพฯ : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.
- [5] นันทน์ภัส โตอดิเทพย์. 2554. “ภาษาซีและโครงสร้างข้อมูล”. นครปฐม : โรงพิมพ์มหาวิทยาลัยศิลปากร
- [6] Charles Dodge and Thamos A.Jerse. 1985. “Digital image processing”. New York : Schirmer Books.





ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 44



ภาคผนวก ก.
Code ที่ใช้ในการปฏิบัติงานบนโปรแกรม HALCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FontName := 'Industrial_0-9_NoRej'
open_framegrabber ('DirectShow', 1, 1, 0, 0, 0, 0, 'default', 8, 'rgb', -1, 'false', 'default', '[0] USB
Camera', 0, -1, AcqHandle)
grab_image_start (AcqHandle, -1)

open_serial ('COM3', SerialHandle)
set_serial_param (SerialHandle, 9600, 8, 'none', 'none', 1, 200, 200)

while (true)
    read_serial (SerialHandle, 1, Data)
    clear_serial (SerialHandle, 'input')

if (Data = 0)
    grab_image_async (Image, AcqHandle, -1)
    dev_update_window ('off')

    rgb1_to_gray (Image, Bottle)
    get_image_size (Bottle, Width, Height)

    dev_close_window ()
    dev_open_window (0, 0, Width, Height, 'black', WindowID)
    set_display_font (WindowID, 24, 'mono', 'true', 'false')
    dev_display (Bottle)

    create_text_model_reader ('auto', 'Document_0-9A-Z_NoRej.omc', TextModel)

    set_text_model_param (TextModel, 'min_stroke_width', 4)
    set_text_model_param (TextModel, 'max_stroke_width', 'auto')

    find_text (Bottle, TextModel, TextResultID)
    get_text_object (Characters, TextResultID, 'all_lines')
    dev_display (Characters)

    get_text_result (TextResultID, 'class', Classes)
    area_center (Characters, Area, Row, Column)

    disp_message (WindowID, Classes, 'image', 80, Column - 3, 'blue', 'false')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tuple_ord(Classes,asd)

if(asd>1)
    write_serial(SerialHandle, asd)
endif
clear_text_result (TextResultID)
clear_text_model (TextModel)

endif
```

```
endwhile
close_framegrabber (AcqHandle)
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The seal of Rajabhat Buriram University is a circular emblem. It features a central sunburst with a flame-like base, flanked by two traditional Thai stupas. The entire design is set against a background of intricate floral and scrollwork patterns. The Thai text "มหาวิทยาลัยราชภัฏบุรีรัมย์" (Mahavithayalai Rajabhat Buriram) is written around the perimeter of the seal.

ภาคผนวก ข.
Code ที่ใช้ในการปฏิบัติงานบนโปรแกรม Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int pushButton = 8;
unsigned int a;
#include "LedControl.h"
LedControl lc=LedControl(12,11,10);
unsigned long delaytime=400;
byte A[8]={ B00000000,
            B00000100,
            B00000000,
            B00011111,
            B00101000,
            B01001000,
            B00101000,
            B00011111,};
byte B[8]={ B00000000,
            B00000110,
            B00000000,
            B00110110,
            B01001001,
            B01001001,
            B01001001,
            B01111111,};
byte C[8]={ B00000100,
            B00000100,
            B00000000,
            B00100010,
            B01000001,
            B01000001,
            B01000001,
            B00111110,};
byte D[8]={ B00000110,
            B00000100,
            B00000000,
            B00011100,
            B00100010,
            B01000001,
            B01000001,
            B01111111,};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
byte E[8]={ B00000010,  
            B00000100,  
            B00000000,  
            B01000001,  
            B01001001,  
            B01001001,  
            B01001001,  
            B01111111,};
```

```
byte F[8]={ B00000100,  
            B00000110,  
            B00000000,  
            B01000000,  
            B01001000,  
            B01001000,  
            B01001000,  
            B01111111,};
```

```
byte G[8]={ B00000110,  
            B00000110,  
            B00000000,  
            B00110000,  
            B01000100,  
            B01000111,  
            B01000001,  
            B00111110,};
```

```
byte H[8]={ B00000010,  
            B00000110,  
            B00000000,  
            B01111111,  
            B00001000,  
            B00001000,  
            B00001000,  
            B01111111,};
```

```
byte I[8]={ B00000100,  
            B00000010,  
            B00000000,  
            B01000001,  
            B01000001,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
B01111111,  
B01000001,  
B01000001,};  
byte J[8]={ B00000010,  
B00000110,  
B00000000,  
B01000000,  
B01111110,  
B01000001,  
B01000001,  
B01000110,  
};
```

```
byte K[8]={ B00000000,  
B00000101,  
B00000000,  
B01000001,  
B00100010,  
B00010100,  
B0000100,  
B01111111,};
```

```
byte L[8]={ B00000000,  
B00000111,  
B00000000,  
B00000001,  
B00000001,  
B00000001,  
B00000001,  
B01111111,};
```

```
byte M[8]={ B00000100,  
B00000101,  
B00000000,  
B01111111,  
B00100000,  
B00010000,  
B00100000,  
B01111111,  
};
```

```
byte N[8]={ B00000110,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
B00000101,  
B00000000,  
B01111111,  
B01000110,  
B00011000,  
B00110000,  
B01111111,};
```

```
byte O[8]={ B00000010,  
B00000101,  
B00000000,  
B00111110,  
B01000001,  
B01000001,  
B01000001,  
B00111110,};
```

```
byte P[8]={ B00000100,  
B00000111,  
B00000000,  
B00110000,  
B01001000,  
B01001000,  
B01001000,  
B01111111,  
};
```

```
byte Q[8]={ B00000110,  
B00000111,  
B00000000,  
B00000011,  
B00111110,  
B01000001,  
B01000001,  
B00111110,};
```

```
byte R[8]={ B00000010,  
B00000111,  
B00000000,  
B00000011,  
B00110100,  
B01001000,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
B01001000,  
B01111111,};  
byte S[8]={ B00000100,  
B00000011,  
B00000000,  
B00100110,  
B01001001,  
B01001001,  
B01001001,  
B00110010,  
};
```

```
byte T[8]={ B00000110,  
B00000011,  
B00000000,  
B01000000,  
B01000000,  
B01111111,  
B01000000,  
B01000000,};
```

```
byte U[8]={ B00000001,  
B00000101,  
B00000000,  
B01111110,  
B00000001,  
B00000001,  
B00000001,  
B01111110,};
```

```
byte V[8]={ B00000001,  
B00000111,  
B00000000,  
B01111000,  
B00000110,  
B00000001,  
B00000110,  
B01111000,  
};
```

```
byte W[8]={ B00000111,  
B00000010,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

B00000000,
B01111110,
B00000001,
B00000110,
B00000001,
B01111110,};
byte X[8]={ B00000101,
B00000101,
B00000000,
B00100001,
B00010010,
B00001100,
B00010010,
B00100001,};
byte Y[8]={ B00000111,
B00000101,
B00000000,
B01100000,
B00011000,
B00000111,
B00011000,
B01100000,
};
byte Z[8]={ B00000011,
B00000101,
B00000000,
B01100001,
B01010001,
B01001001,
B01000101,
B01000011,};

```

```

void setup()
{
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pinMode(7,OUTPUT);
pinMode(pushButton, INPUT);
Serial.begin(9600);
lc.shutdown(0,false);
lc.setIntensity(0,2);
lc.clearDisplay(0);
}
```

```
void loop()
{
  int buttonState = digitalRead(pushButton);
  Serial.write(buttonState);
  delay(60);
  if(Serial.available()>0)
  {
    a=Serial.read();
    if(a==65)
    {digitalWrite(2,HIGH);
      digitalWrite(3,LOW);
      digitalWrite(4,LOW);
      digitalWrite(5,LOW);
      digitalWrite(6,LOW);
      digitalWrite(7,LOW);
      lc.setRow(0,0,A[0]);
      lc.setRow(0,1,A[1]);
      lc.setRow(0,2,A[2]);
      lc.setRow(0,3,A[3]);
      lc.setRow(0,4,A[4]);
      lc.setRow(0,5,A[5]);
      lc.setRow(0,6,A[6]);
      lc.setRow(0,7,A[7]);
    }else if(a==66)
    {digitalWrite(2,HIGH);
      digitalWrite(3,HIGH);
      digitalWrite(4,LOW);
      digitalWrite(5,LOW);
      digitalWrite(6,LOW);
      digitalWrite(7,LOW);
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lc.setRow(0,0,B[0]);
lc.setRow(0,1,B[1]);
lc.setRow(0,2,B[2]);
lc.setRow(0,3,B[3]);
lc.setRow(0,4,B[4]);
lc.setRow(0,5,B[5]);
lc.setRow(0,6,B[6]);
lc.setRow(0,7,B[7]);
}else if(a==67)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
lc.setRow(0,0,C[0]);
lc.setRow(0,1,C[1]);
lc.setRow(0,2,C[2]);
lc.setRow(0,3,C[3]);
lc.setRow(0,4,C[4]);
lc.setRow(0,5,C[5]);
lc.setRow(0,6,C[6]);
lc.setRow(0,7,C[7]);
}else if(a==68)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,D[0]);
lc.setRow(0,1,D[1]);
lc.setRow(0,2,D[2]);
lc.setRow(0,3,D[3]);
lc.setRow(0,4,D[4]);
lc.setRow(0,5,D[5]);
lc.setRow(0,6,D[6]);
lc.setRow(0,7,D[7]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}else if(a==69)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,E[0]);
lc.setRow(0,1,E[1]);
lc.setRow(0,2,E[2]);
lc.setRow(0,3,E[3]);
lc.setRow(0,4,E[4]);
lc.setRow(0,5,E[5]);
lc.setRow(0,6,E[6]);
lc.setRow(0,7,E[7]);
}else if(a==70)
{digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
lc.setRow(0,0,F[0]);
lc.setRow(0,1,F[1]);
lc.setRow(0,2,F[2]);
lc.setRow(0,3,F[3]);
lc.setRow(0,4,F[4]);
lc.setRow(0,5,F[5]);
lc.setRow(0,6,F[6]);
lc.setRow(0,7,F[7]);
}else if(a==71)
{digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,G[0]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lc.setRow(0,1,G[1]);
lc.setRow(0,2,G[2]);
lc.setRow(0,3,G[3]);
lc.setRow(0,4,G[4]);
lc.setRow(0,5,G[5]);
lc.setRow(0,6,G[6]);
lc.setRow(0,7,G[7]);
}else if(a==72)
{digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,H[0]);
lc.setRow(0,1,H[1]);
lc.setRow(0,2,H[2]);
lc.setRow(0,3,H[3]);
lc.setRow(0,4,H[4]);
lc.setRow(0,5,H[5]);
lc.setRow(0,6,H[6]);
lc.setRow(0,7,H[7]);
}else if(a==73)
{digitalWrite(2,LOW);
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
lc.setRow(0,0,I[0]);
lc.setRow(0,1,I[1]);
lc.setRow(0,2,I[2]);
lc.setRow(0,3,I[3]);
lc.setRow(0,4,I[4]);
lc.setRow(0,5,I[5]);
lc.setRow(0,6,I[6]);
lc.setRow(0,7,I[7]);
}else if(a==74)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,LOW);
  lc.setRow(0,0,J[0]);
  lc.setRow(0,1,J[1]);
  lc.setRow(0,2,J[2]);
  lc.setRow(0,3,J[3]);
  lc.setRow(0,4,J[4]);
  lc.setRow(0,5,J[5]);
  lc.setRow(0,6,J[6]);
  lc.setRow(0,7,J[7]);
}else if(a==75)
{digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  lc.setRow(0,0,K[0]);
  lc.setRow(0,1,K[1]);
  lc.setRow(0,2,K[2]);
  lc.setRow(0,3,K[3]);
  lc.setRow(0,4,K[4]);
  lc.setRow(0,5,K[5]);
  lc.setRow(0,6,K[6]);
  lc.setRow(0,7,K[7]);
}else if(a==76)
{digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  lc.setRow(0,0,L[0]);
  lc.setRow(0,1,L[1]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lc.setRow(0,2,L[2]);
lc.setRow(0,3,L[3]);
lc.setRow(0,4,L[4]);
lc.setRow(0,5,L[5]);
lc.setRow(0,6,L[6]);
lc.setRow(0,7,L[7]);
}else if(a==77)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
lc.setRow(0,0,M[0]);
lc.setRow(0,1,M[1]);
lc.setRow(0,2,M[2]);
lc.setRow(0,3,M[3]);
lc.setRow(0,4,M[4]);
lc.setRow(0,5,M[5]);
lc.setRow(0,6,M[6]);
lc.setRow(0,7,M[7]);
}else if(a==78)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,N[0]);
lc.setRow(0,1,N[1]);
lc.setRow(0,2,N[2]);
lc.setRow(0,3,N[3]);
lc.setRow(0,4,N[4]);
lc.setRow(0,5,N[5]);
lc.setRow(0,6,N[6]);
lc.setRow(0,7,N[7]);
}else if(a==79)
{digitalWrite(2,HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,O[0]);
lc.setRow(0,1,O[1]);
lc.setRow(0,2,O[2]);
lc.setRow(0,3,O[3]);
lc.setRow(0,4,O[4]);
lc.setRow(0,5,O[5]);
lc.setRow(0,6,O[6]);
lc.setRow(0,7,O[7]);
}else if(a==80)
{digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
lc.setRow(0,0,P[0]);
lc.setRow(0,1,P[1]);
lc.setRow(0,2,P[2]);
lc.setRow(0,3,P[3]);
lc.setRow(0,4,P[4]);
lc.setRow(0,5,P[5]);
lc.setRow(0,6,P[6]);
lc.setRow(0,7,P[7]);
}else if(a==81)
{digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,Q[0]);
lc.setRow(0,1,Q[1]);
lc.setRow(0,2,Q[2]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lc.setRow(0,3,Q[3]);
lc.setRow(0,4,Q[4]);
lc.setRow(0,5,Q[5]);
lc.setRow(0,6,Q[6]);
lc.setRow(0,7,Q[7]);
}else if(a==82)
{digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,R[0]);
lc.setRow(0,1,R[1]);
lc.setRow(0,2,R[2]);
lc.setRow(0,3,R[3]);
lc.setRow(0,4,R[4]);
lc.setRow(0,5,R[5]);
lc.setRow(0,6,R[6]);
lc.setRow(0,7,R[7]);
}else if(a==83)
{digitalWrite(2,LOW);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
lc.setRow(0,0,S[0]);
lc.setRow(0,1,S[1]);
lc.setRow(0,2,S[2]);
lc.setRow(0,3,S[3]);
lc.setRow(0,4,S[4]);
lc.setRow(0,5,S[5]);
lc.setRow(0,6,S[6]);
lc.setRow(0,7,S[7]);
}else if(a==84)
{digitalWrite(2,LOW);
digitalWrite(3,HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
lc.setRow(0,0,T[0]);
lc.setRow(0,1,T[1]);
lc.setRow(0,2,T[2]);
lc.setRow(0,3,T[3]);
lc.setRow(0,4,T[4]);
lc.setRow(0,5,T[5]);
lc.setRow(0,6,T[6]);
lc.setRow(0,7,T[7]);
}else if(a==85)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
lc.setRow(0,0,U[0]);
lc.setRow(0,1,U[1]);
lc.setRow(0,2,U[2]);
lc.setRow(0,3,U[3]);
lc.setRow(0,4,U[4]);
lc.setRow(0,5,U[5]);
lc.setRow(0,6,U[6]);
lc.setRow(0,7,U[7]);
}else if(a==86)
{digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
lc.setRow(0,0,V[0]);
lc.setRow(0,1,V[1]);
lc.setRow(0,2,V[2]);
lc.setRow(0,3,V[3]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lc.setRow(0,4,V[4]);
lc.setRow(0,5,V[5]);
lc.setRow(0,6,V[6]);
lc.setRow(0,7,V[7]);
}else if(a==87)
{digitalWrite(2,LOW);
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
lc.setRow(0,0,W[0]);
lc.setRow(0,1,W[1]);
lc.setRow(0,2,W[2]);
lc.setRow(0,3,W[3]);
lc.setRow(0,4,W[4]);
lc.setRow(0,5,W[5]);
lc.setRow(0,6,W[6]);
lc.setRow(0,7,W[7]);
}else if(a==88)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
lc.setRow(0,0,X[0]);
lc.setRow(0,1,X[1]);
lc.setRow(0,2,X[2]);
lc.setRow(0,3,X[3]);
lc.setRow(0,4,X[4]);
lc.setRow(0,5,X[5]);
lc.setRow(0,6,X[6]);
lc.setRow(0,7,X[7]);
}else if(a==89)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
lc.setRow(0,0,Y[0]);
lc.setRow(0,1,Y[1]);
lc.setRow(0,2,Y[2]);
lc.setRow(0,3,Y[3]);
lc.setRow(0,4,Y[4]);
lc.setRow(0,5,Y[5]);
lc.setRow(0,6,Y[6]);
lc.setRow(0,7,Y[7]);
}else if(a==90)
{digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
lc.setRow(0,0,Z[0]);
lc.setRow(0,1,Z[1]);
lc.setRow(0,2,Z[2]);
lc.setRow(0,3,Z[3]);
lc.setRow(0,4,Z[4]);
lc.setRow(0,5,Z[5]);
lc.setRow(0,6,Z[6]);
lc.setRow(0,7,Z[7]);
}
else
{digitalWrite(2,LOW);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
}
}
delay(1000);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * LedControl.h - A library for controlling Leds with a MAX7219/MAX7221
 * Copyright (c) 2007 Eberhard Fahle
 *
 * Permission is hereby granted, free of charge, to any person
 * obtaining a copy of this software and associated documentation
 * files (the "Software"), to deal in the Software without
 * restriction, including without limitation the rights to use,
 * copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the
 * Software is furnished to do so, subject to the following
 * conditions:
 *
 * This permission notice shall be included in all copies or
 * substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
 * HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
 * FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
 * OTHER DEALINGS IN THE SOFTWARE.
 */
```

```
#ifndef LedControl_h
#define LedControl_h
```

```
#include <avr/pgmspace.h>
```

```
#if (ARDUINO >= 100)
```

```
#include <Arduino.h>
```

```
#else
```

```
#include <WProgram.h>
```

```
#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
 * Segments to be switched on for characters and digits on
 * 7-Segment Displays
 */

```

```

const static byte charTable [] PROGMEM = {
    B01111110,B00110000,B01101101,B01111001,
    B00110011,B01011011,B01011111,B01110000,
    B01111111,B01111011,B01110111,B00011111,
    B00001101,B00111101,B01001111,B01000111,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B10000000,B00000001,B10000000,B00000000,
    B01111110,B00110000,B01101101,B01111001,
    B00110011,B01011011,B01011111,B01110000,
    B01111111,B01111011,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B01110111,B00011111,B00001101,
    B00111101,B01001111,B01000111,B00000000,
    B00110111,B00000000,B00000000,B00000000,
    B00001110,B00000000,B00000000,B00000000,
    B01100111,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00001000,
    B00000000,B01110111,B00011111,B00001101,
    B00111101,B01001111,B01000111,B00000000,
    B00110111,B00000000,B00000000,B00000000,
    B00001110,B00000000,B00010101,B00011101,
    B01100111,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000,
    B00000000,B00000000,B00000000,B00000000

```

```

};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

class LedControl {
private :
    /* The array for shifting the data to the devices */
    byte spidata[16];
    /* Send out a single command to the device */
    void spiTransfer(int addr, byte opcode, byte data);

    /* We keep track of the led-status for all 8 devices in this array */
    byte status[64];
    /* Data is shifted out of this pin*/
    int SPI_MOSI;
    /* The clock is signaled on this pin */
    int SPI_CLK;
    /* This one is driven LOW for chip selectzion */
    int SPI_CS;
    /* The maximum number of devices we use */
    int maxDevices;

public:
    /*
    * Create a new controller
    * Params :
    * dataPin      pin on the Arduino where data gets shifted out
    * clockPin     pin for the clock
    * csPin        pin for selecting the device
    * numDevices   maximum number of devices that can be controled
    */
    LedControl(int dataPin, int clkPin, int csPin, int numDevices=1);

    /*
    * Gets the number of devices attached to this LedControl.
    * Returns :
    * int         the number of devices on this LedControl
    */
    int getDeviceCount();

    /*
    * Set the shutdown (power saving) mode for the device

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* Params :
* addr The address of the display to control
* status If true the device goes into power-down mode. Set to false
*       for normal operation.
*/
void shutdown(int addr, bool status);

/*
* Set the number of digits (or rows) to be displayed.
* See datasheet for sideeffects of the scanlimit on the brightness
* of the display.
* Params :
* addr address of the display to control
* limit number of digits to be displayed (1..8)
*/
void setScanLimit(int addr, int limit);

/*
* Set the brightness of the display.
* Params:
* addr the address of the display to control
* intensity the brightness of the display. (0..15)
*/
void setIntensity(int addr, int intensity);

/*
* Switch all Leds on the display off.
* Params:
* addr address of the display to control
*/
void clearDisplay(int addr);

/*
* Set the status of a single Led.
* Params :
* addr address of the display
* row the row of the Led (0..7)
* col the column of the Led (0..7)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* state If true the led is switched on,
*      if false it is switched off
*/
void setLed(int addr, int row, int col, boolean state);

/*
* Set all 8 Led's in a row to a new state
* Params:
* addr address of the display
* row row which is to be set (0..7)
* value each bit set to 1 will light up the
*      corresponding Led.
*/
void setRow(int addr, int row, byte value);

/*
* Set all 8 Led's in a column to a new state
* Params:
* addr address of the display
* col column which is to be set (0..7)
* value each bit set to 1 will light up the
*      corresponding Led.
*/
void setColumn(int addr, int col, byte value);

/*
* Display a hexadecimal digit on a 7-Segment Display
* Params:
* addr address of the display
* digit the position of the digit on the display (0..7)
* value the value to be displayed. (0x00..0x0F)
* dp sets the decimal point.
*/
void setDigit(int addr, int digit, byte value, boolean dp);

```

```

* Display a character on a 7-Segment display.
* There are only a few characters that make sense here :

```

```

*'0','1','2','3','4','5','6','7','8','9','0',

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
* 'A','b','c','d','E','F','H','L','P',  
* ' ','-', '_'  
* Params:  
* addr  address of the display  
* digit  the position of the character on the display (0..7)  
* value  the character to be displayed.  
* dp     sets the decimal point.  
void setChar(int addr, int digit, char value, boolean dp);  
};  
  
#endif //LedControl.h
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

