

เครื่องจำแนกสีและนับจำนวนโคโลนีอัตโนมัติโดยใช้การประมวลผลภาพ
AUTOMATIC COLONY COUNTING AND COLOR CLASSIFICATION
DEVICE BASED ON IMAGE PROCESSING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

เครื่องจำแนกสีและนับจำนวนโคโลนีอัตโนมัติโดยใช้การประมวลผลภาพ
AUTOMATIC COLONY COUNTING AND COLOR CLASSIFICATION
DEVICE BASED ON IMAGE PROCESSING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC COLONY COUNTING AND COLOR CLASSIFICATION

DEVICE BASED ON IMAGE PROCESSING



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN MECHATRONIC ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องจำแนกสีและนับจำนวนโคโลนีอัตโนมัติโดยใช้การประมวลผลภาพ
AUTOMATIC COLONY COUNTING AND COLOR CLASSIFICATION
DEVICE BASED ON IMAGE PROCESSING

ผู้จัดทำ นายเจษฎา บุญชัย 55010197

นายนันทวุฒิ ทองทรง 55010655

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.นพดล มณีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจำแนกสีและนับจำนวนโคโลนีอัตโนมัติโดยใช้การ ประมวลผลภาพ

โดย

นายเจษฎา บุญชัย 55010197

นายฉันทวุฒิ ทองทรง 55010655

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.นพดล มณีรัตน์

ปีการศึกษา 2558

บทคัดย่อ

การเพาะเลี้ยงเชื้อแบคทีเรียเพื่อจำแนกและนับจำนวนโคโลนี เป็นขั้นตอนในการสำคัญในการหาชนิดและปริมาณของแบคทีเรียที่ปนเปื้อนในอาหาร ในปัจจุบันการจำแนกและนับจำนวนโคโลนีต้องใช้บุคลากรผู้เชี่ยวชาญในการปฏิบัติงาน ถึงแม้จะเป็นวิธีที่มีประสิทธิภาพสามารถนับจำนวนโคโลนีได้อย่างแม่นยำ แต่มีความยุ่งยากและใช้เวลามากในการนับจำนวนโคโลนีแต่ละจานเพาะเชื้อ ในปริญญาโทนี้ได้มีวิธีการตรวจนับโคโลนีโดยใช้กระบวนการประมวลผลภาพรวมกับการทำงานอัตโนมัติของอุปกรณ์ ทำให้กระบวนการทำงานมีความสะดวกมากขึ้นและใช้เวลาเฉลี่ยนลง ในขั้นตอนการออกแบบจะประกอบด้วย 2 ส่วนหลักคือ ฮาร์ดแวร์ และซอฟต์แวร์ ส่วนของซอฟต์แวร์ การประมวลผลภาพจะรับภาพจากกล้องนำผ่านการทำเทรซโฮลด์ การหาขอบของภาพ การกำจัดสัญญาณรบกวน และการนับจำนวนโคโลนี ส่วนของฮาร์ดแวร์จะแบ่ง 3 ส่วน คือส่วนรับจานเพาะเชื้อ ส่วนถ่ายรูปประมวลผล และส่วนพิกจานที่นับแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC COLONY COUNTING AND COLOR CLASSIFICATION DEVICE BASED ON IMAGE PROCESSING

By

Mr.Jessada Boonchai 55010197

Mr.Nantawut Thongtrong 55010655

Advisor

Asst.Prof.Dr. Noppadol Maneerat

Academic Year 2015

ABSTRACT

Growing bacteria on petri dish for counting colony is important process to classify and estimate amount of bacteria contaminated in foods. At the present, the method of classification and colony counting are used by personel who expert in process. Although it high efficiently can count number of colony correctly but process are complicated and take time. This thesis present a method of colony counting by using image processing and automatic machine. Method of image processing will be convenient and use less time. The design and development of colony counting device comprises of two part which hardware and software. Software part used image processing unit to analyze image. Main step is receive image from camera , multilevel threshold, edge detection and counting algorithm. The hardware of device are petri dish receiver, camera and petri dish keeper after counting.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปฏิญญานิพนธ์เรื่องเครื่องจำแนกสีและนับจำนวนโคลนอีเอ็มดีโดยใช้การประมวลผลภาพ สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับการสนับสนุนและความช่วยเหลือเป็นอย่างดีจากอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.นพดล มณีรัตน์ อาจารย์ประจำภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้คำปรึกษาชี้แนะช่วยแก้ปัญหา ตลอดจนให้ความรู้

ขอขอบคุณ นายรัตนพล ยุทธวิริยะ ที่คอยให้คำปรึกษา ช่วยเหลือหาทางแก้ปัญหา ช่วยจัดทำอุปกรณ์ และให้กำลังใจอยู่เสมอ

ขอขอบคุณ ศูนย์ปฏิบัติการวิจัยทางการแพทย์และการเกษตรแห่งเอเชีย จำกัด (AMARC) ที่ได้ให้ความอนุเคราะห์ในการเข้าไปดูงานศึกษาตัวอย่างโคลนอีเอ็มดี

ขอขอบคุณเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือ ให้คำแนะนำในการทำปฏิญญานิพนธ์และให้กำลังใจในการทำปฏิญญานิพนธ์ตลอดมา

สุดท้ายนี้ขอขอบคุณครอบครัวของผู้ทำปฏิญญานิพนธ์ สำหรับการสนับสนุนและให้กำลังใจในทุกด้านจนปฏิญญานิพนธ์นี้สำเร็จลุล่วงได้เป็นอย่างดี คุณค่าและประโยชน์อันพึงมีจากปฏิญญานิพนธ์นี้ขอมอบแต่ผู้มีพระคุณทุกท่าน

ผู้จัดทำ

นายเจษฎา บุญชัย

นายนันทวุฒิ ทองทรง

สารบัญ

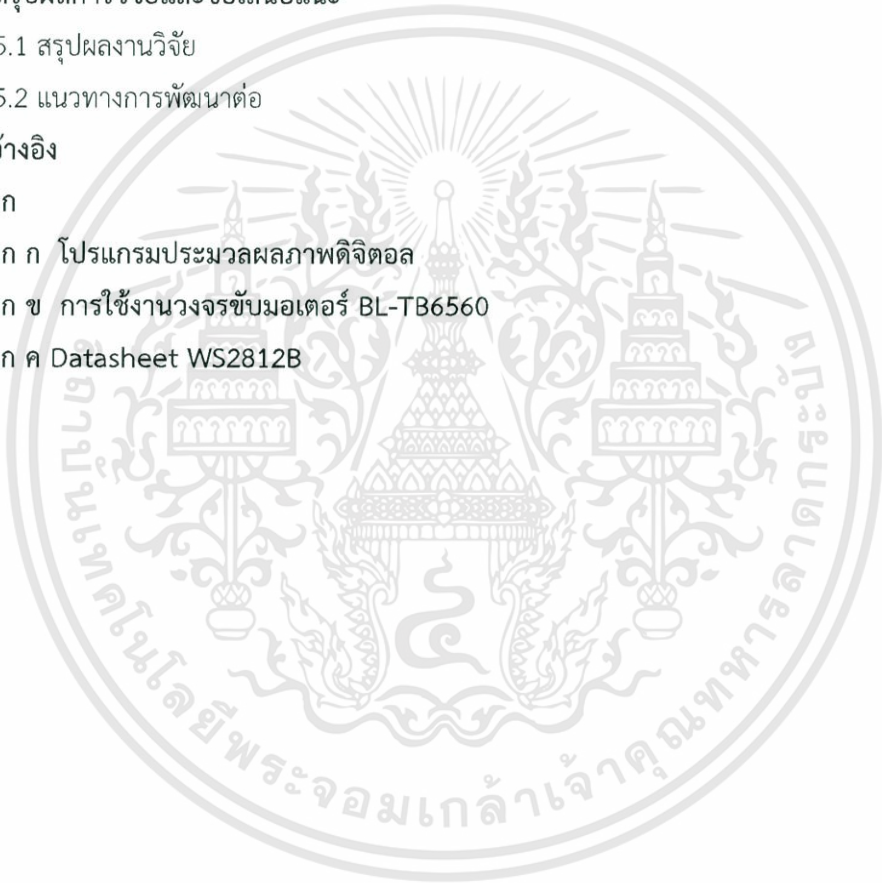
	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของปริญญานิพนธ์	1
1.3 สมมติฐานของงานปริญญานิพนธ์	2
1.4 ขอบเขตปริญญานิพนธ์	2
1.5 รายละเอียดปริญญานิพนธ์	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	4
2.1 บทนำ	4
2.2 การเตรียมและการเพาะเชื้อโคลนีย์	4
2.2.1 วิธีการเพาะเชื้อ Streak – Plate Technique	4
2.2.2 วิธีการนับเชื้อโคลนีย์แบบดั้งเดิม	5
2.3 การประมวลผลภาพดิจิทัล	6
2.3.1 การประมวลผลภาพเชิงตัวเลข	6
2.3.1.1 มาตรฐานแสดงค่าความสว่าง	6
2.3.1.2 พิกัดตำแหน่งจุดภาพ	6
2.3.2 การตัดระดับเทรชโฮลด์ (Threshold)	7
2.3.3 การทำเทรชโฮลด์หลายระดับ (Multilevel Threshold)	8
2.3.4 ความสัมพันธ์ระหว่างพิกเซล	9
2.3.5 การหาขอบภาพ (Edge Detection)	9
2.3.6 การทำมอร์โฟโลยี (Morphology)	10
2.4 สเต็ปมอเตอร์	10
2.4.1 สเต็ปเปอร์มอเตอร์แบบไบโพลาร์	11
2.4.2 สเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์	11
2.4.3 การควบคุมการหมุนของสเต็ปมอเตอร์	12

สารบัญ (ต่อ)

	หน้า
2.4.3.1 การกระตุ้นสเต็ปเปอร์มอเตอร์แบบพัลสเต็มป์ 1 เฟส	12
2.4.3.2 การกระตุ้นแบบพัลสเต็มป์ 2 เฟส	12
2.4.3.3 การกระตุ้นแบบครึ่งสเต็ปหรือฮาล์ฟสเต็ป	13
2.4.4 วงจรขับสเต็ปเปอร์มอเตอร์อย่างง่าย	14
2.5 Arduino	14
2.6 ไอซี LED เบอร์ WS2812B	15
2.6.1 โครงสร้างของโมดูลไอซี LED เบอร์ WS2812B	15
2.6.2 การควบคุมทำงานของไอซี LED เบอร์ WS2812B	16
บทที่ 3 โครงสร้างและการออกแบบ	18
3.1 การออกแบบโครงสร้าง	18
3.1.1 การเลือกวัสดุ	18
3.1.2 การออกแบบชิ้นส่วนต่างๆ ของเครื่อง	18
3.1.2.1 การออกแบบ In Feed Stack	18
3.1.2.2 การออกแบบ Pusher Petri Dish	19
3.1.2.3 การออกแบบ Gripper	19
3.1.2.4 การออกแบบ Lift	20
3.1.2.5 การออกแบบ Discharge Stack	20
3.2 ภาพรวมโครงสร้างและหลักการทำงาน	21
3.3 การออกแบบซอฟต์แวร์ประมวลผลภาพ	22
3.3.1 การออกแบบหน้าจออินเตอร์เฟส	22
3.3.2 การออกแบบกระบวนการประมวลผลภาพ	24
บทที่ 4 การทดลองและผลการทดลอง	27
4.1 ทดสอบซอฟต์แวร์ประมวลผลภาพ	27
4.1.1 ทดลองใช้งานซอฟต์แวร์ประมวลผลภาพ	28
4.1.2 ทดสอบการทำงานของโปรแกรมโดยใช้พื้นหลังสีขาว	29
4.1.3 ทดสอบการทำงานของโปรแกรมโดยใช้สีพื้นให้เหมาะสม	30
4.1.4 วิเคราะห์และสรุปผลการทดลอง	32
4.2 ทดสอบการทำงานของสเต็ปมอเตอร์	33

สารบัญ (ต่อ)

	หน้า
4.2.1 ทดสอบทิศทางการหมุนของสเต็ปมอเตอร์ โดยใช้วงจรไดร์ BL-TB6560	34
4.2.2 ทดสอบองศาการหมุนของสเต็ปมอเตอร์	35
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	36
5.1 สรุปผลงานวิจัย	36
5.2 แนวทางการพัฒนาต่อ	37
เอกสารอ้างอิง	38
ภาคผนวก	39
ภาคผนวก ก โปรแกรมประมวลผลภาพดิจิทัล	40
ภาคผนวก ข การใช้งานวงจรขับเคลื่อน BL-TB6560	47
ภาคผนวก ค Datasheet WS2812B	49



สารบัญรูป

รูปที่	หน้า
2.1 ขั้นตอนการเพาะเชื้อด้วยวิธี Streak – Plate Technique	5
2.2 การแทนภาพด้วยเมตริกซ์สองมิติ	7
2.3 ฮิสโตแกรมและตำแหน่งเทรชโฮลด์	8
2.4 ฮิสโตแกรมและตำแหน่งเทรชโฮลด์	8
2.5 ภาพผลลัพธ์ที่ได้จากการหาขอบภาพวิธีต่างๆ	9
2.6 โครงสร้างอย่างง่ายของสแต็ปมอเตอร์	11
2.7 ลักษณะการพันขดลวดแบบยูนิโพลาร์	11
2.8 การต่อ Arduino กับ LED	14
2.9 การต่อ Arduino กับ Xbee Shield	14
2.10 ตำแหน่งขาต่อใช้งานไอซี LED เบอร์ WS2812B	15
2.11 ข้อกำหนดสำหรับสัญญาณดิจิทัลที่ใช้กำหนดค่าบิตแต่ละบิต สำหรับ WS2812B	16
2.12 การจัดเรียงบิตแบบ GRB	16
3.1 โครงสร้าง In Feed Stack	18
3.2 โครงสร้าง Pusher Petri Dish	19
3.3 โครงสร้าง Gripper	19
3.4 โครงสร้าง Lift	20
3.5 โครงสร้าง Discharge Stack	20
3.6 เครื่องนับและจำแนกสีโคลนี	21
3.7 วางถาดเพาะเชื้อบน In Feed Stack	21
3.8 ปุ่ม Start Camera	22
3.9 การเลือกรูปแบบการทำงาน	22
3.10 การเลือกสีโคลนีที่ต้องการนับ	23
3.11 หน้าจอแสดงผลการนับโคลนี	23
3.12 หน้าจออินเตอร์เฟซของซอฟต์แวร์ประมวลผลภาพ	23
3.13 แผนผังแสดงการทำงานของโปรแกรมประมวลผลภาพ	24
3.14 การแยกองค์ประกอบสีของหยดสีน้ำสีแดง	25
3.15 ภาพหลังผ่านกระบวนการเทรชโฮลด์	25
3.16 การตีกรอบรอบจุดที่ทำการนับแล้ว	26

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.1 ถาดหยดสีน้ำที่ใช้สำหรับการทดลอง	27
4.2 คอมพิวเตอร์โน้ตบุ๊ก LENOVO Z510	28
4.3 หน้าต่างโปรแกรมหลังจากการกดปุ่ม CAMERA	28
4.4 หน้าต่างการใช้โปรแกรม MANUAL	29
4.5 การต่อสแต็ปมอเตอร์กับวงจรขับเคลื่อนมอเตอร์ BL-TB6560	33
4.6 การต่อวงจรไมโครคอนโทรลเลอร์กับวงจรขับเคลื่อนมอเตอร์ BL-TB6560	33
4.7 วงจรควบคุมสแต็ปมอเตอร์ด้วยวงจรขับเคลื่อนมอเตอร์ BL-TB6560	34
4.8 การปรับตั้งสวิทช์วงจรขับเคลื่อนมอเตอร์ BL-TB6560	34



สารบัญตาราง

ตารางที่	หน้า
2.1 รูปแบบการกระตุ้นขดลวดไฟฟ้าแบบเวฟหรือฟูลสเต็ป	12
2.2 รูปแบบการกระตุ้นขดลวดไฟฟ้าแบบฟูลสเต็ป 2 เฟส	13
2.3 รูปแบบการกระตุ้นแบบครึ่งสเต็ปหรือฮาล์ฟสเต็ป	13
4.1 ผลการทดสอบโดยใช้พื้นวางงานสีขาว	29
4.2 ตารางแสดงสีพื้นหลังที่เหมาะสมกับหยดสีต่างๆ	31
4.3 ผลการทดสอบโปรแกรมโดยเปลี่ยนสีพื้นหลังของส่วนวางงานเพาะเชื้อ	31
4.4 ตารางเปรียบเทียบใช้พื้นหลังสีขาวกับการใช้พื้นหลังที่เหมาะสม	32
4.5 ตารางแสดงผลการทดลอง CW+ เป็น HIGH และ LOW ด้วย BL-TB6560	35
4.6 อนุกรมการหมุนของสเต็ปมอเตอร์เมื่อหมุนตามจำนวน Step ที่กำหนด	35



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้มีการคิดค้นยารักษาโรคขึ้นมามากมายเพื่อใช้สำหรับป้องกันและรักษาโรคภัยไข้เจ็บที่เกิดขึ้น โดยยาแต่ละประเภทยานั้นจะมีสารตั้งต้นและวิธีการผลิตที่แตกต่างกัน ซึ่งสารแอนติไบโอติก (Antibiotic) เป็นหนึ่งในสารที่สำคัญที่ใช้ในการผลิตยา เนื่องจากสารนี้มีความสามารถในการยับยั้งจุลินทรีย์ได้หลายชนิด จึงสามารถยับยั้งการเจริญเติบโตของ แบคทีเรีย รา ไวรัส โปรโตซัว รวมทั้งสามารถต้านเซลล์มะเร็งและเซลล์เนื้องอกได้ โดยที่ 2 ใน 3 ของสารแอนติไบโอติกทั้งหมดที่มีในปัจจุบัน สามารถพบได้ในแบคทีเรียตระกูล Actinomycete ดังนั้นแบคทีเรียในตระกูล Actinomycete นี้จึงจัดว่าเป็นแบคทีเรียที่มีความสำคัญทางการแพทย์และทางเภสัชกรรมเป็นอย่างมาก

อีกหนึ่งความสำคัญของการเพาะเลี้ยงเชื้อแบคทีเรียคือ เพื่อหาสิ่งที่เป็นเพื่อนในอาหารซึ่งจะมีทั้งแบคทีเรียที่มีประโยชน์และให้โทษ เนื่องจากการบริโภคอาหารที่ไม่สะอาดเข้าไป อาจทำให้เกิดการเจ็บป่วยได้ การนำเข้าและส่งออกอาหารจำเป็นต้องผ่านการตรวจสอบคุณภาพและรับรองตามกฎหมายของประเทศนั้นๆ ดังนั้นความสะอาดในอุตสาหกรรมอาหารจึงเป็นสิ่งสำคัญ โดยที่การเพาะเลี้ยงเชื้อแบคทีเรียในงานเพาะเชื้อ เพื่อจำแนกชนิดและนับจำนวนโคโลนีจึงเป็นขั้นตอนพื้นฐาน ซึ่งการตรวจนับดังกล่าวจะสามารถรายงานค่าที่เกี่ยวข้องออกมาได้หลายค่า คือชนิดของเชื้อแบคทีเรียที่เป็นเพื่อนอยู่ในอาหาร ปริมาณของเชื้อที่เป็นเพื่อนอยู่ โดยเชื้อ Staphylococcus Aureus เป็นสาเหตุของโรคอาหารเป็นพิษ ผู้ป่วยจะมีอาการคลื่นไส้ อาเจียน ปวดท้อง อ่อนเพลีย ในรายที่รุนแรงมีอาการปวดศีรษะ ปวดกล้ามเนื้อ ความดันโลหิตเปลี่ยนแปลงเป็นต้น นอกจากนี้แล้วการเพาะเชื้อแบคทีเรียยังช่วยให้ทราบว่าอาหารที่ผลิตขึ้นมานั้นสามารถเก็บรักษาไว้ได้กี่วัน และควรเก็บรักษาที่อุณหภูมิใด

1.2 ความมุ่งหมายและวัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาความรู้ด้านการประมวลผลภาพดิจิทัล (Digital Image Processing)
2. เพื่อจำแนกสีและนับจำนวนโคโลนี โดยการวิเคราะห์จากภาพถ่ายได้
3. เพื่อเปรียบเทียบผลการทดลองที่ได้จากงานวิจัยนี้กับผลจากผู้เชี่ยวชาญ
4. เพื่อปรับปรุงการนับจำนวนโคโลนีให้มีประสิทธิภาพ และสะดวกสบายยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เพื่อออกแบบอุปกรณ์ให้มีราคาถูกและสามารถเคลื่อนย้ายให้ใช้งานนอกเหนือจากห้องวิจัยได้

6. เพื่อเสนออีกแนวทางหนึ่งสำหรับใช้ในการจำแนกสีและนับจำนวนโคโลนี

1.3 สมมติฐานของงานปริญญานิพนธ์

จากการทำการศึกษาเบื้องต้น พบว่าการจำแนกสีและนับจำนวนโคโลนีที่มีสีแตกต่างกันไปนั้นทำได้ค่อนข้างลำบาก เนื่องจากสีของโคโลนีหลายชนิดจะมีสีที่ใกล้เคียงกับสีของพื้นหลัง อีกทั้งยังมีโคโลนีบางชนิดที่สีมีความคล้ายคลึงกัน จึงทำให้การประมวลผลภาพมีความผิดพลาด ดังนั้นการออกแบบเครื่องจำแนกสีและนับจำนวนโคโลนีจึงใช้หลอดแอลอีดีชนิด RGB มาติดตั้งที่ใต้พื้นสำหรับวางจานเพาะเชื้อ เพื่อช่วยสร้างความแตกต่างระหว่างสีของโคโลนีแต่ละชนิดกับสีของพื้นหลัง แล้วจึงนำเข้ากระบวนการประมวลผลภาพ เพื่อทำการจำแนกสีและนับจำนวนโคโลนีต่อไป ด้วยวิธีนี้จึงน่าจะมีความถูกต้องในการจำแนก และนับจำนวนโคโลนีโดยใช้พื้นหลังที่มีสีเดียวกัน ดังนั้นส่วนของความถูกต้องของการจำแนกและนับจำนวนโคโลนีจะมีมากหรือน้อยเพียงใดนั้น จึงขึ้นกับกระบวนการเลือกสีพื้นหลังที่ถูกต้องด้วย

1.4 ขอบเขตปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้มีการใช้ไมโครคอนโทรลเลอร์สำหรับควบคุมการทำงานของหลอดแอลอีดีชนิด RGB ที่ใช้ในการสร้างสีของพื้นสำหรับวางจานเพาะเชื้อ ซึ่งทำงานร่วมกับซอฟต์แวร์การประมวลผลภาพ ที่พัฒนาผ่านโปรแกรม Visual Studio และการควบคุมสเต็ปมอเตอร์เพื่อรับและส่งจานเพาะเชื้อ

1.5 รายละเอียดปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 5 บท โดยมีรายละเอียดดังนี้

บทที่ 1 ความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ สมมติฐาน และขอบเขตของการวิจัย

บทที่ 2 ขั้นตอนการเพาะเลี้ยงเชื้อแบคทีเรียในจานเพาะเชื้อ วิธีการนับเชื้อโคโลนีแบบดั้งเดิม ความรู้พื้นฐานของการประมวลผลภาพดิจิทัล การควบคุมการทำงานของหลอดแอลอีดี WS2812B

บทที่ 3 การออกแบบและพัฒนาทั้งในส่วนของซอฟต์แวร์ประมวลผลภาพ และเครื่องนับโคโลนีและส่วนประกอบต่างๆ

บทที่ 4 การทดลองและผลการทดลอง แสดงผลลัพธ์ของการทดลอง

บทที่ 5 สรุปผลการดำเนินงาน และแนวทางการพัฒนาต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการและทฤษฎีที่เกี่ยวข้อง

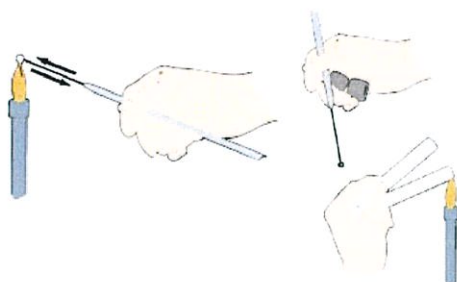
2.1 บทนำ

ในการพัฒนาเครื่องจำแนกสีและนับจำนวนโคโลนีโดยใช้การประมวลผลภาพดิจิทัลซึ่งจะกล่าวถึง ขั้นตอนการเพาะเชื้อแบคทีเรียในงานเพาะเชื้อ การตรวจนับตัวอย่างโคโลนีซึ่งในปริญญานิพนธ์ฉบับนี้เรียกว่าวิธีการนับเชื้อแบคทีเรียแบบดั้งเดิม การออกแบบเครื่องนับเชื้อที่ต้องใช้การควบคุมมอเตอร์ร่วมกับการใช้ไมโครคอนโทรลเลอร์ควบคุมหลอด LED รวมไปถึงหลักการการประมวลผลภาพดิจิทัล

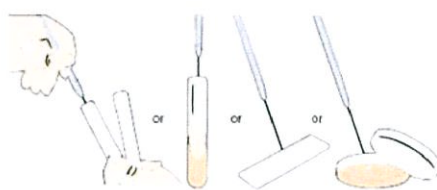
2.2 การเตรียมและการเพาะเชื้อโคโลนี

2.2.1 วิธีการเพาะเชื้อ Streak – Plate Technique

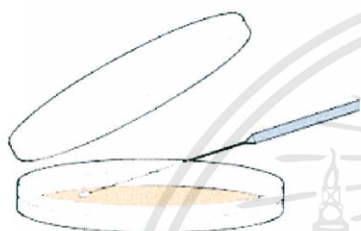
การแยกเชื้อให้บริสุทธิ์เป็นเทคนิคพื้นฐานทางจุลชีววิทยาที่มีความสำคัญอย่างมากเนื่องจากสิ่งแวดล้อมต่างๆ เช่น น้ำ อากาศ พื้น หรือแม้แต่ร่างกายของคนก็มีเชื้อจุลินทรีย์อยู่หลายชนิดที่อาจปนเปื้อนในหลอดเพาะเชื้อได้ หลักการของการแยกเชื้อให้บริสุทธิ์บนอาหารเลี้ยงเชื้อ (Culture Medium) คือ จะต้องแยกเชื้อให้ได้โคโลนีเดี่ยวๆ (Single Colony) จำนวนมาก จากนั้นจึงนำเชื้อที่เป็นโคโลนีเดี่ยวไปศึกษารูปร่างลักษณะและคุณสมบัติต่างๆ เพื่อให้ทราบว่าเป็นเชื้อชนิดใด เทคนิคที่นิยมใช้ในการแยกเชื้อบริสุทธิ์คือ วิธี Cross Streak Plate ซึ่งทำได้โดยใช้ห่วงเขี่ยเชื้อ (Loop)แตะตัวอย่างหรือสิ่งส่งตรวจแล้วลากหรือขีด (Streak) ลงบนจานเพาะเชื้อที่มีอาหารแข็ง (Agar Plate) อยู่ให้ได้แนวระนาบติดต่อกัน 4-5 เส้น ติดกันโดยรอยขีดของเชื้อจะไม่ทับกับระนาบแรกอีก หลังจากนั้นก็ทำเช่นเดียวกันกับระนาบที่สองจนครบทั่วทั้งจานเพาะเชื้อซึ่งมีประมาณสี่ระนาบ เมื่อได้เชื้อบริสุทธิ์แล้วจะมีการศึกษาเชื้อต่อไปในด้านต่างๆ ซึ่งจะต้องมีการถ่ายเชื้อจากอาหารเดิมไปยังอาหารใหม่ หรือมีการเพาะเชื้อลงในอาหารเพื่อการทดสอบและการวิเคราะห์ต่างๆ ดังนั้นวิธีการที่ถูกต้องในการถ่ายเชื้อจึงมีความสำคัญเป็นอย่างยิ่ง ซึ่งต้องอาศัยหลักการของ Aseptic Technique เพื่อป้องกันการปนเปื้อนของเชื้อชนิดอื่นซึ่งจะทำให้ผลการทดสอบผิดพลาดได้ โดยในขั้นตอนการถ่ายเชื้อจุลินทรีย์จำพวกแบคทีเรียจะใช้ห่วงเขี่ยเชื้อ และเข็มเขี่ยเชื้อปลายตรง ส่วนเชื้อราที่เป็นเส้นสาย (Filamentous Fungi) จะใช้เข็มเขี่ยปลายงอ



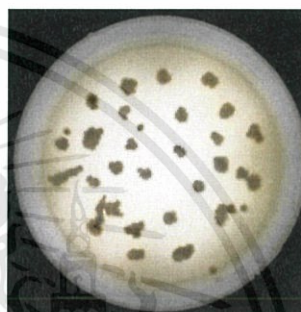
(ก) ขั้นตอนการเผาไหม้เพื่อฆ่าเชื้อ



(ข) การใช้ Loop เชี่ยเชื้อแบคทีเรีย



(ค) การ Streak เชื้อบนอาหารเพาะเชื้อ



(ง) เชื้อโคโลนีที่ได้

รูปที่ 2.1 ขั้นตอนการเพาะเชื้อด้วยวิธี Streak – Plate Technique

2.2.2 วิธีการนับเชื้อโคโลนีแบบดั้งเดิม

การนับจำนวนโคโลนีเป็นวิธีการพื้นฐานในการหาปริมาณของเชื้อแบคทีเรีย โดยอาศัยหลักการในการหาปริมาณของเชื้อแบคทีเรีย โดยอาศัยหลักการว่าแบคทีเรียที่มีชีวิตเมื่อได้รับสารอาหารจะสามารถเจริญเติบโต และแบ่งเซลล์เพิ่มจำนวนทับถมกันจนเป็นโคโลนี ดังนั้น 1 โคโลนีจึงหมายถึงการเจริญเติบโตและแบ่งตัวจากเซลล์แบคทีเรียเริ่มต้น 1 เซลล์ วิธีการทั่วไปที่ใช้ในทางปฏิบัติคือ การนับจำนวนของแบคทีเรียบนจานอาหารเลี้ยงเชื้อซึ่งดังขั้นตอนต่อไปนี้

1. เพาะตัวอย่างเชื้อแบคทีเรียบนจานเลี้ยงเชื้อ
2. ตรวจสอบจานเลี้ยงเชื้อที่มีโคโลนีเกิดขึ้น หลังจากที่ได้บ่มในระยะเวลาที่เหมาะสมที่ดี
3. เลือกจานที่โคโลนีปรากฏขึ้นระหว่าง 30 – 300 โคโลนี เพื่อจะเป็นตัวอย่างที่ดีที่สุด
4. ใช้ Light Box หรือ Colony Counter แล้วใช้ปากกาทำเครื่องหมายบริเวณโคโลนี ที่นับแล้วเพื่อป้องกันการนับซ้ำ โดยปกติแล้วจะนับโคโลนีที่อยู่ในช่วง 30 – 300 โคโลนี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการบันทึกข้อมูลโดยมีรายละเอียดคือหมายเลขของงานเพาะเชื้อและจำนวนโคโลนีที่ตรวจนับได้

2.3 การประมวลผลภาพดิจิทัล

2.3.1 การประมวลผลภาพเชิงตัวเลข

โดยการแทนจุดภาพใดๆ ด้วยฟังก์ชันสองมิติชนิดไม่ต่อเนื่อง $F(x,y)$ ซึ่งเกิดจากการแบ่งซอยฟังก์ชันต่อเนื่อง $F(x,y)$ เป็นช่วง ผลลัพธ์ที่ได้ประกอบด้วยสององค์ประกอบคือ

2.3.1.1 มาตรฐานแสดงค่าความสว่าง

คือ มาตรฐานแสดงค่าความสว่างของจุดภาพใดๆ จากการวัดแล้วแปลงเป็นค่าตัวเลข (Amplitude Digitization) ซึ่งจะเรียกว่าความเข้ม ในกรณีของภาพสี จะหมายถึงค่าระดับของแม่สีที่ผสมกันเป็นสีต่างๆ ส่วนภาพขาวดำจะแสดงค่าระดับความเข้มของสีขาว โดยแทนด้วยเลขจำนวนเต็มบวก L จำนวน การแบ่งชั้นของระดับค่า L จะบอกถึงความละเอียดของภาพในเชิงตัวเลข เช่น ภาพที่มีระดับความเข้มหลายระดับจะถูกเรียกว่าภาพระดับสีเทา (Gray Scale Image) ในทางคอมพิวเตอร์นิยมแบ่งค่าระดับความเข้ม L ให้เหมาะสมกับจำนวนบิต เช่น ภาพระดับสีเทา 8 บิตสามารถแบ่งระดับความเข้มออกเป็น 256 ระดับ ($2^8 = 256$) แสดงได้ดังสมการที่ (2.1)

$$F(x, y) = L, 0 \leq L \leq 2^n - 1 \quad (2.1)$$

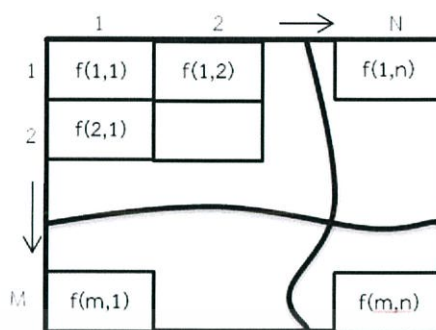
โดยที่ X, Y คือ พิกัดตำแหน่งจุดภาพ

N คือ จำนวนบิตของรหัสแทนค่าความเข้มของจุดภาพ

L คือ ค่าแสดงค่าความเข้มจุดภาพ

2.3.1.2 พิกัดตำแหน่งจุดภาพ

โดยการแบ่งภาพที่ต่อเนื่องกันให้เป็นตาราง (Spatial Digitization) โดยแต่ละช่องของตารางภาพคือ จุดภาพ (Pixel) และแทนตำแหน่งที่อยู่ของจุดภาพด้วยฟังก์ชันสองมิติในระนาบแกน X, Y หรือใช้เมตริกซ์สองมิติขนาด $M \times N$ แสดงดังรูปที่ 2.2 เรียกวิธีการนี้ว่าการสุ่มจุดภาพ (Image Sampling)



(ก) ภาพต้นฉบับ

(ข) ภาพที่แปลงเป็นดิจิตอล

รูปที่ 2.2 การแทนภาพด้วยเมตริกซ์สองมิติ

2.3.2 การตัดระดับเทรชโฮลด์ (Threshold)

การตัดระดับเทรชโฮลด์ (Threshold) เป็นวิธีการที่นิยมใช้กัน เพื่อแยกส่วนที่เป็นวัตถุและพื้นหลังออกจากกัน ซึ่งมีวิธีการคือ นำจุดภาพใดๆ บนภาพต้นฉบับ $F(x,y)$ มาเปรียบเทียบกับค่าแบ่งระดับ ความเข้ม T ถ้าจุดภาพมีค่าความเข้มมากกว่าหรือเท่ากับค่าแบ่งระดับความเข้ม จะกำหนดให้ภาพผลลัพธ์ $F(x,y)$ มีค่าเป็น 1 นอกนั้นให้ค่าเป็น 0

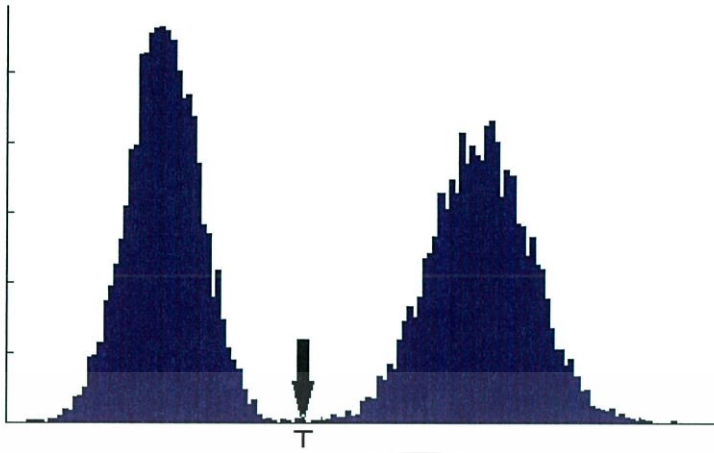
$$F(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) < T \end{cases} \quad (2.2)$$

T คือ ค่าเทรชโฮลด์

$f(x, y)$ คือ จุดภาพต้นฉบับ

$f'(x, y)$ คือ จุดภาพผลลัพธ์ชนิดความเข้มสองระดับ

ภาพความเข้มสองระดับจะให้ค่า 1 แทนวัตถุ หรือเนื้อภาพ (Foreground) และให้ค่า 0 แทน ส่วนพื้นหลัง (Background) เนื่องจากการแทนค่าความเข้มด้วยตัวเลขของภาพชนิดนี้ใช้เพียงสองค่าเท่านั้นจึงเรียกว่า ภาพไบนารี (Binary Image)



รูปที่ 2.3 ฮิสโตแกรมและตำแหน่งเทรชโฮลด์

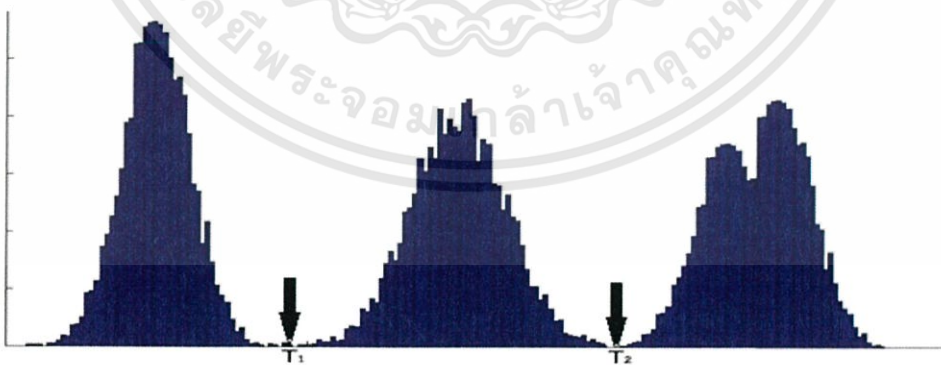
2.3.3 การทำเทรชโฮลด์หลายระดับ (Multilevel Threshold)

ในบางครั้งโคโลนอาจมีค่าระดับความเข้มแสงที่แตกต่างกันออกไปเพื่อให้การแยกโคโลนกับพื้นหลังมีประสิทธิภาพ จึงจำเป็นต้องกำหนดค่าระดับเทรชโฮลด์ (T) มากกว่าหนึ่งค่าขึ้นไปในที่นี้กำหนดค่าคือ T_1 และ T_2

ถ้า $f(x, y) < T_1$ แล้ว $g(x, y) = 0$

ถ้า $T_1 \leq f(x, y) \leq T_2$ แล้ว $g(x, y) = 255$

ถ้า $f(x, y) > T_2$ แล้ว $g(x, y) = 0$



รูปที่ 2.4 ฮิสโตแกรมและตำแหน่งเทรชโฮลด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 ความสัมพันธ์ระหว่างพิกเซล

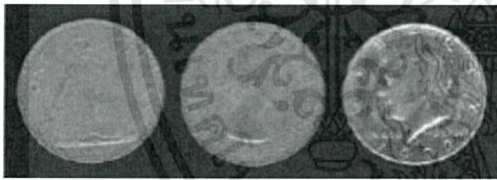
หลังผ่านกระบวนการตัดระดับเทรชโอล์ดมาแล้วจะได้รับภาพไบนารี โดยพิกเซลมีค่า 1 แทนส่วนวัตถุ และพิกเซลที่มีค่า 0 แทนพื้นหลัง กำหนดให้ภาพแทนด้วย (x,y) และเมื่ออ้างอิงถึงพิกเซลใดๆ จะแทนด้วยตัวอักษรเช่น p q

2.3.4.1 จุดรอบข้างของพิกเซล

พิกเซล p ที่ตำแหน่ง (x,y) จะมีจุดรอบข้างในแนวตั้งละแนวนอน 4 จุด คือ $(x+1,y)$ $(x-1,y-1)$, $(x,y+1)$, $(x-1,y-1)$ เรียกว่า 4-neighbors ของจุด p แทนด้วย $N(p)$ โดยแต่ละพิกเซลมีระยะห่างจาก (x,y) 1 หน่วย ส่วนจุดรอบข้างในแนวทแยงมุมของ p คือ $(x+1,y+1)$, $(x+1,y-1)$, $(x-1,y+1)$, $(x-1,y-1)$ เมื่อรวมกับ 4-neighbors เรียกว่า 8-neighbors ของจุด p

2.3.5 การหาขอบภาพ (Edge Detection)

การหาขอบภาพ (Edge Detection) เป็นวิธีการสำหรับการประมวลผลภาพที่มีจุดประสงค์เพื่อการหาขอบเขตในภาพ โดยการทำให้ขอบของภาพเด่นชัดขึ้นมา เพื่อทำการหาขอบเขตของภาพต่างๆ ได้โดยขอบเขตภาพที่เกิดความเด่นชัดขึ้นมาจากความแตกต่างระหว่างความเข้มของแสงจากจุดหนึ่งไปยังภาพอีกจุดหนึ่งที่มีความต่อเนื่องกัน



(ก) ภาพต้นฉบับ



(ข) Laplacian Operator



(ค) Sobel Operator



(ง) Prewitt Operator

รูปที่ 2.5 ภาพผลลัพธ์ที่ได้จากการหาขอบภาพวิธีต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 การทำมอร์โฟโลยี (Morphology)

2.3.6.1 การเซาะ (Erosion)

การเซาะ (Erosion) เป็นพื้นฐานของมอร์โฟโลยี (Morphology) มีลักษณะที่คล้ายการย่อภาพ โดยลบบริเวณของขอบภาพออก การเซาะภาพ A ด้วย เทมเพลต B ดังสมการที่ (2.3)

$$A \ominus B = \bigcap \{A + b : b \in -B\} \quad (2.3)$$

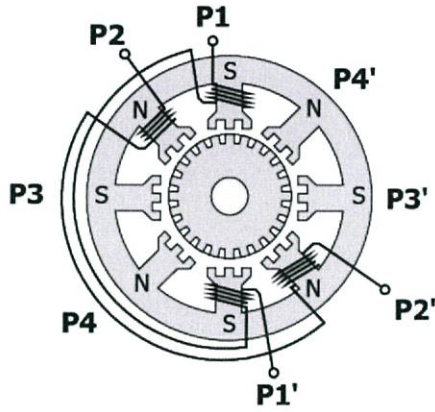
2.3.6.2 การขยาย (Dilation)

การขยาย (Dilation) หรือการขยายภาพเป็นกระบวนการที่คู่กับการเซาะ (Erosion) ในที่นี้ การขยาย (Dilation) ของภาพอินพุต A ด้วยเทมเพลต B ดังสมการที่ (2.4)

$$A \oplus B = \bigcup \{A + b : b \in B\} \quad (2.4)$$

2.4 สเต็ปมอเตอร์

สเต็ปมอเตอร์เป็นมอเตอร์ไฟฟ้ากระแสสลับชนิดหนึ่ง que เปลี่ยนสัญญาณดิจิทัลเป็นการเคลื่อนที่ทางกล จึงเหมาะสำหรับการเชื่อมต่อกับอุปกรณ์ดิจิทัลหรือคอมพิวเตอร์ ในอดีตมีการแบ่งชนิดของสเต็ปเปอร์มอเตอร์ตามลักษณะโครงสร้างซึ่งแบ่งได้เป็น 3 ชนิดคือ แบบแม่เหล็กถาวรหรือ PM (Permanent Magnet), แบบปรับค่าความต้านทานแม่เหล็กได้หรือ VR (Variable Reluctance) และแบบผสมหรือไฮบริด (Hybrid) ซึ่งเป็นการผสมกันระหว่างแบบ PM และ VR ในปัจจุบันนี้สเต็ปเปอร์มอเตอร์ส่วนใหญ่เป็นแบบไฮบริด เนื่องจากสามารถทำให้มีความละเอียดในการเคลื่อนที่ของแกนได้สูงถึง 0.9 องศาต่อสเต็ป (ซึ่งเป็นข้อดีของแบบ VR) และให้แรงบิดหรือทอร์กที่สูง โดยใช้พลังงานต่ำ (เป็นข้อดีของแบบ PM) ในรูปที่ 2.6 แสดงโครงสร้างอย่างง่ายของสเต็ปเปอร์มอเตอร์แบบไฮบริด



รูปที่ 2.6 โครงสร้างอย่างง่ายของสเต็ปมอเตอร์

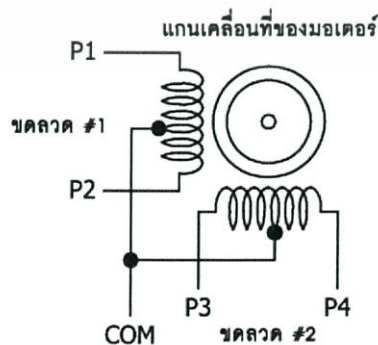
ดังนั้นการกำหนดชนิดของสเต็ปเปอร์มอเตอร์ในยุคต่อมาจนถึงปัจจุบันจึงพิจารณาที่ลักษณะของการพันขดลวด, การต่อสายออกมาใช้งาน และวงจรขับ ซึ่งแบ่งเป็น 2 ชนิดหลักๆ คือ ชนิดไบโพลาร์ (Bipolar) และชนิดยูนิโพลาร์ (Uni-polar)

2.4.1 สเต็ปเปอร์มอเตอร์แบบไบโพลาร์

ลักษณะการพันขดลวดของมอเตอร์แบ่งออกเป็น 2 ชุดที่ไม่มีแท่งกลาง ทำให้บางครั้งจึงเรียกสเต็ปเปอร์มอเตอร์แบบนี้ว่า เป็นสเต็ปเปอร์มอเตอร์แบบ 2 เฟส การขับให้มอเตอร์แบบนี้หมุนจะต้องป้อนแรงดันต่างขั้วกันให้แก่ขดลวดแต่ละชุด ทำให้วงจรขับสเต็ปเปอร์มอเตอร์แบบนี้ค่อนข้างซับซ้อน

2.4.2 สเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์

มีลักษณะการพันขดลวดของมอเตอร์แสดงในรูปที่ 2.7 มีด้วยกัน 2 แบบคือ แบบ 5 และ 6 สาย บางครั้งเรียกสเต็ปเปอร์มอเตอร์แบบนี้ว่า เป็นสเต็ปเปอร์มอเตอร์แบบ 4 เฟส การขับจะต้องป้อนสัญญาณเข้าที่ขั้วหรือเฟสของมอเตอร์ให้เรียงลำดับอย่างถูกต้อง มอเตอร์จึงจะสามารถหมุนได้อย่างราบรื่น



รูปที่ 2.7 ลักษณะการพันขดลวดแบบยูนิโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 การควบคุมการหมุนของสเต็ปมอเตอร์

การกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีควเอนเชียลในรูปแบบที่ถูกต้องด้วย สามารถแบ่งได้เป็น 3 รูปแบบคือ แบบเวฟ (Wave) หรือแบบฟูลสเต็ป 1 เฟส (Full Step 1-Phase), แบบฟูลสเต็ป 2 เฟส และแบบครึ่งสเต็ป (Half Step)

2.4.3.1 การกระตุ้นสเต็ปเปอร์มอเตอร์ฟูลสเต็ป 1 เฟส

เป็นการกระตุ้นที่มีรูปแบบง่ายที่สุด โดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป เช่น เริ่มต้นที่ขดที่ 1, 2, 3, 4 แล้ววนกลับมาขดที่ 1 วนไปเรื่อยๆ หรือเริ่มที่ขดที่ 1 แล้วย้อนไปยังขดที่ 4, 3, 2 แล้วกลับมาขดที่ 1 อีกครั้ง ซึ่งทำให้ทิศทางของการหมุนสวนกัน ในการกระตุ้นรูปแบบนี้จึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบนี้มีราคาถูกและง่าย ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.1

ตารางที่ 2.1 แสดงรูปแบบการกระตุ้นขดลวดไฟฟ้าแบบเวฟหรือฟูลสเต็ป

สเต็ปที่	ขดลวดที่ 1	ขดลวดที่ 2	ขดลวดที่ 3	ขดลวดที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

2.4.3.2 การกระตุ้นแบบฟูลสเต็ป 2 เฟส

เป็นการกระตุ้นซึ่งคล้ายกับแบบฟูลสเต็ปหนึ่งเฟส แต่การกระตุ้นแบบนี้จ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปเช่นเดียวกับแบบฟูลสเต็ป 1 เฟส ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 ถัดไปเป็นขดที่ 4 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนี้ หรือเริ่มที่ขด 1 และ 4 ตามด้วยขดที่ 4 และ 3 ถัดไปเป็นขดที่ 3 และ 2 ต่อไปเป็นขดที่ 2 และ 1 แล้ววนกลับมาที่ขดที่ 1 และ 4 ทิศทางการหมุนจะสวนทางกัน การกระตุ้นสเต็ปเปอร์มอเตอร์แบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบฟูลสเต็ป 1 เฟส โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และเคลื่อนที่ต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือการกระตุ้นแบบนี้ต้องใช้กำลังไฟฟ้ามกขึ้น ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.2

ตารางที่ 2.2 แสดงรูปแบบการกระตุ้นขดลวดไฟฟ้าแบบฟูลสเต็ป 2 เฟส

สเต็ปที่	ขดลวดที่ 1	ขดลวดที่ 2	ขดลวดที่ 3	ขดลวดที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

2.4.3.3 การกระตุ้นแบบครึ่งสเต็ปหรือฮาล์ฟสเต็ป

เป็นรูปแบบที่ผสมผสานระหว่างการกระตุ้นแบบฟูลสเต็ป 1 และ 2 เฟส เพื่อเพิ่มจำนวน ของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ เริ่มจากขดลวดที่ 1, 1 และ 2, 2, 2 และ 3, 3, 3 และ 4, 4, 4 และ 1 แล้ววนกลับมาขดลวดที่ 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องพึงระวังไว้อีกประการหนึ่งว่าเมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องทำการหมุนถึง 2 สเต็ป จึงจะได้เท่ากับระยะเท่ากับ 1 สเต็ปเต็มของการควบคุมใน 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้ขนาดเท่ากับแบบ 2 เฟสเป็นอย่างน้อยจึงจะเพียงพอ ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.3

ตารางที่ 2.3 แสดงรูปแบบการกระตุ้นแบบครึ่งสเต็ปหรือฮาล์ฟสเต็ป

สเต็ปที่	ขดลวดที่ 1	ขดลวดที่ 2	ขดลวดที่ 3	ขดลวดที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

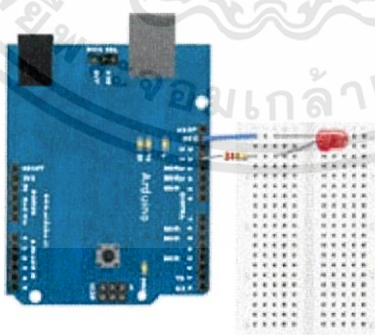
2.4.4 วงจรขับสแต็ปเปอร์มอเตอร์อย่างง่าย

จากการทำงานที่อธิบายมาสามารถใช้วงจรขับสแต็ปเปอร์มอเตอร์ได้หลายแบบ โดยมีหลักการที่ตรงกันคือ ป้อนสัญญาณกระตุ้นอย่างเป็นลำดับที่ถูกต้องแก่ขั้วหรือเฟสของสแต็ปเปอร์มอเตอร์ และถ้าหากต้องการให้มอเตอร์หมุนเร็วหรือช้าให้ทำการกำหนดหรือปรับแต่งที่ความถี่ของสัญญาณกระตุ้นที่ใช้ขับมอเตอร์ในรูปที่ 2.8 แสดงวงจรขับสแต็ปเปอร์มอเตอร์แบบยูนิโพลาร์อย่างง่าย แบบพูลสแต็ป 1 เฟสที่ไม่ต้องใช้ไมโครคอนโทรลเลอร์ เพียงมีวงจรกำเนิดสัญญาณพัลส์ซึ่งในวงจรใช้ ไอซีเบอร์ 4093 (อาจใช้เบอร์ 4011 แทนได้) ส่งสัญญาณไปยังวงจรจัดการลำดับของสัญญาณกระตุ้นซึ่งในวงจรใช้ไอซีเบอร์ 4017 เข้ามาจัดการเพื่อให้มีการส่งสัญญาณกระตุ้นไปยังมอเตอร์ที่ละเฟส

2.5 Arduino

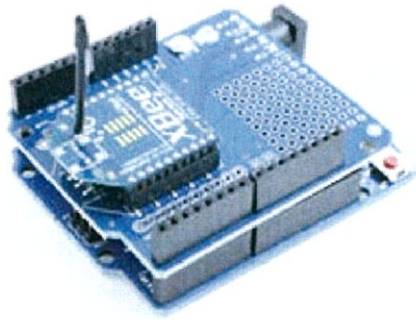
Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด (ดูตัวอย่างรูปที่ 2.8) หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ (ดูตัวอย่างรูปที่ 2.9) เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเปรียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย



รูปที่ 2.8 การต่อ Aduino กับ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

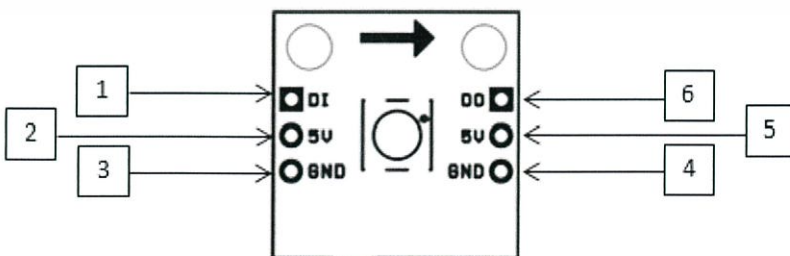


รูปที่ 2.9 การต่อ Arduino กับ Xbee Shield

2.6 ไอซี LED เบอร์ WS2812B

WS2812B เป็นอุปกรณ์อิเล็กทรอนิกส์ซึ่งได้รวมหลอดแอลอีดีชนิด RGB กับไอซี WS2811 ไว้ด้วยกัน โดยภายในไอซี WS2812B จะประกอบไปด้วย วงจรปรับรูปสัญญาณ, วงจร Drive, วงจรควบคุมพิกเซล RGB, วงจรรีเซ็ต, วงจร Power Lose Reset และภายในแอลอีดีแต่ละหลอดก็จะประกอบด้วยแม่สี 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน ซึ่งจะให้ความละเอียดบิตสีได้สูงถึง 24 บิตสี RGB ($3 \times 8 \text{ บิต} = 24 \text{ บิต}$) โดยในแต่ละสีจะแสดงความสว่างได้ 256 ระดับ นั้นหมายความว่าหลอดแอลอีดีแบบ RGB นี้สามารถแสดงสีได้ทั้งหมด 16,777,216 สี และเมื่อใช้อัตราการ Refresh อยู่ที่ 30 fps จะสามารถนำโมดูลนี้มาต่อกันแบบ Cascade หรือแบบอนุกรมได้ทั้งหมด 1024 หลอด โดยไม่จำเป็นต้องเพิ่มวงจรใดๆ ในส่วนของขั้นตอนการควบคุมจะใช้การอินเตอร์เฟส (Interface) ด้วยสายเพียงเส้นเดียวแบบ NRZ (Non Return to Zero) คือส่งข้อมูล (Data) ในแบบอนุกรม (Serial) โดยข้อมูล 0 หรือ 1 นั้น จะถูกกำหนดด้วยคาบเวลาซึ่งมีลักษณะเหมือนกับสัญญาณพัลส์ (Pulse) โดยจะใช้สัญญาณ 1 คาบเวลาต่อ 1 บิตข้อมูล (Bit Data) เป็นตัวกำหนดข้อมูล 0 หรือ 1 ตัวโมดูลนี้สามารถรองรับอินพุต ได้ทั้งในแบบ TTL 5 โวลต์ และ 3.3 โวลต์ แต่ไฟเลี้ยงโมดูลนี้จะต้องเป็นตีสี่ 5 โวลต์ เท่านั้น

2.6.1 โครงสร้างของโมดูล ไอซี LED เบอร์ WS2812B



รูปที่ 2.10 ตำแหน่งขาต่อใช้งานไอซี LED เบอร์ WS2812B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

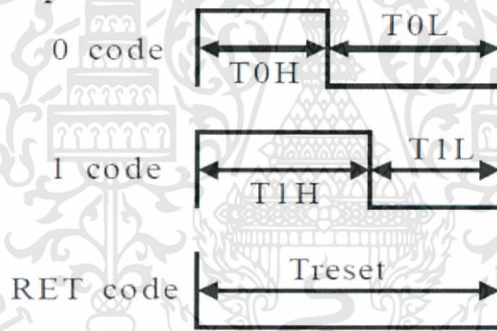
โดยที่

- | | |
|---------|--|
| 1 = DI | คือ ขาอินพุตรับสัญญาณ Serial Data Bit Color/รับสัญญาณ TTL 5V |
| 2 = 5V | คือ ไฟเลี้ยงโมดูล DC 5V |
| 3 = GND | คือ Ground |
| 4 = GND | คือ Ground |
| 5 = 5V | คือ ไฟเลี้ยง DC 5V ให้กับโมดูลตัวต่อไป |
| 6 = DO | คือ ขาเอาต์พุตสำหรับส่ง Serial Data Bit Color |

2.6.2 การควบคุมทำงานของไอซี LED เบอร์ WS2812B

โดยสิ่งสำคัญในการควบคุมนี้ จะอยู่ที่วิธีการสร้างสัญญาณลอจิก 0, 1 และรีเซต ซึ่งจะขึ้นอยู่กับความสามารถของทรัพยากรของไมโครคอนโทรลเลอร์ ว่าสามารถสร้างสัญญาณได้ตามที่โมดูลต้องการหรือไม่ โดยจะใช้การส่งข้อมูลของ SPI เป็นตัวสร้างสัญญาณ ซึ่งมีกระบวนการดังนี้

1. เริ่มต้นจะต้องสร้างสัญญาณขึ้นมา 3 แบบ คือ สัญญาณ ลอจิก 0, ลอจิก 1 และ สัญญาณ รีเซต โดยสัญญาณทั้ง 3 รูปแบบนี้จะเกี่ยวข้องกับเวลาในการ On/Off ของสัญญาณ



รูปที่ 2.11 ข้อกำหนดสำหรับสัญญาณดิจิทัลที่ใช้กำหนดค่าบิตแต่ละบิต สำหรับ WS2812B

2. ส่งข้อมูลลอจิกโค้ด (Logic Data Code) สี 24 บิตไปยังขา DI ของโมดูลหลอดแอลอีดี โดยการส่ง Code 0 หรือ 1 ตามตารางที่ 2.1 ออกไป 1 คาบเวลา ก็จะหมายถึงการส่งข้อมูล 0 หรือ 1 ออกไปให้โมดูลแอลอีดี 1 บิต นั่นเอง ซึ่ง Code สี 24 บิต ที่ส่งออกไปนั้นจะมีการจัดเรียงบิตแบบ GRB แสดงดังรูปที่ 2.12 และจำเป็นต้องส่งข้อมูลบิต MSB ออกไปเป็นบิตแรก

G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	B	B	B	B	B	B	B		
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

รูปที่ 2.12 การจัดเรียงบิตแบบ GRB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ฟังก์ชันสำหรับส่ง Frame Data โดยการส่งข้อมูล 1 Frame ก็คือ การส่งข้อมูลสี 24 บิต ในข้อ 2 นั้นเอง ซึ่งการส่งข้อมูล 1 Frame นี้จะเป็นการกำหนดสีให้กับแอลอีดี 1 หลอดนั่นเอง ดังนั้น จำนวน Frame ที่ส่งแต่ละครั้งก็ต้องเท่ากับจำนวนหลอดแอลอีดีที่ต่อ Cascade กันอยู่ทั้งหมด โดย Frame ที่ส่งออกไป Frame แรก จะเป็นการกำหนดสีให้กับแอลอีดีหลอดแรก Frame ที่ส่งออกไป Frame ที่ 2 ก็จะเป็นการกำหนดสีให้กับแอลอีดีหลอดที่ 2 ที่ต่อ Cascade อยู่ตามลำดับเช่นนี้ไปเรื่อยๆ เมื่อส่ง Frame ครบตามจำนวนหลอดแอลอีดี ที่ต่อแล้วก็ต้องปิดท้ายด้วยสัญญาณรีเซต เสมอ เพื่อให้ Frame ที่ส่งต่อมา หลังสัญญาณรีเซตจะเริ่มต้นกำหนดค่าสีให้กับแอลอีดีหลอดแรกใหม่อีกครั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

โครงสร้างและการออกแบบ

ในส่วนของการออกแบบและหลักการทำงานของเครื่องนับโคโลนีแบบอัตโนมัตินั้นได้แบ่งออกเป็น 3 ส่วนที่สำคัญดังนี้ ส่วนของโครงสร้างของเครื่อง ภาพรวมโครงสร้างและการทำงานของเครื่อง และการออกแบบซอฟต์แวร์การประมวลผลภาพ ซึ่งจะแยกอธิบายแต่ละส่วนดังต่อไปนี้

3.1 การออกแบบโครงสร้าง

3.1.1 การเลือกวัสดุ

เนื่องจากโครงงานนี้เป็นการสร้างเครื่องนับโคโลนีแบบอัตโนมัติ ที่มีราคาที่เหมาะสมกว่า ดังนั้นการเลือกวัสดุจึงพิจารณาถึงการใช้งาน ความคงทนแข็งแรงและราคาเป็นหลัก จึงเลือกใช้สแตนเลสเป็นชิ้นส่วนเพื่อป้องกันการทำปฏิกิริยากับสารเคมีในห้องปฏิบัติการ

3.1.2 การออกแบบชิ้นส่วนต่างๆของเครื่อง

3.1.2.1 การออกแบบ In Feed Stack

In Feed Stack จะทำหน้าที่บรรจุจานเพาะเชื้อเป็นแถวเรียงกันในแนวตั้ง เพื่อเตรียมเข้าสู่การนับ โดยจะมีเซ็นเซอร์แสงเพื่อตรวจสอบจานเพาะเชื้อในแต่ละแถว เมื่อแถวไหนหมด In Feed Stack ก็จะหมุนเพื่อเปลี่ยนแถวใหม่ลำเรียงงานเข้าสู่กระบวนการนับต่อไป



รูปที่ 3.1 โครงสร้าง In Feed Stack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 การออกแบบ Pusher Petri Dish

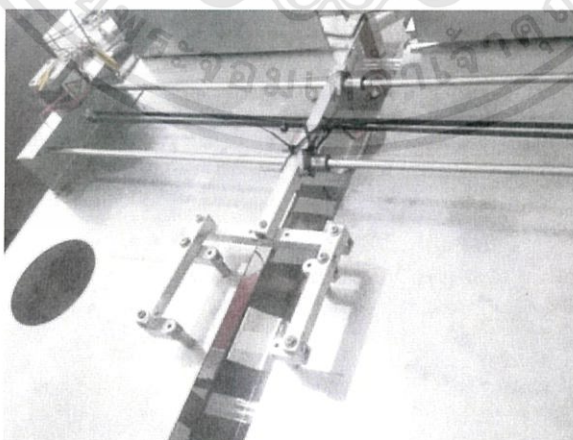
Pusher Petri Dish จะทำหน้าที่ลำเลียงงานเพาะเชื้อจาก In Feed Stack เข้าสู่กระบวนการนับและลำเลียงไปส่งให้ Lift ยกไปเก็บไว้ที่ Discharge Stack โดยจะมีตำแหน่งการลำเลียง 5 ตำแหน่งต่อการเคลื่อนของ Pusher Petri Dish 1 ครั้ง การเคลื่อนที่ของ Pusher Petri Dish จะเคลื่อนที่ในแกนแนวนอน (x) และแกนแนวตั้งใช้ Limit Switch ในการกำหนดความแม่นยำของการเคลื่อนที่



รูปที่ 3.2 โครงสร้าง Pusher Petri Dish

3.1.2.3 การออกแบบ Gripper

Gripper จะทำหน้าที่ลำเลียงงานเพาะเชื้อจากตำแหน่งของ Pusher Petri Dish ไปเข้าสู่ส่วนของกระบวนการนับที่มีกล้องถ่ายภาพ และลำเลียงจากส่วนของกระบวนการนับกลับสู่แนวการเคลื่อนที่ของ Pusher Petri Dish ใช้ Limit Switch ในการกำหนดความแม่นยำของการเคลื่อนที่



รูปที่ 3.3 โครงสร้าง Gripper

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.4 การออกแบบ Lift

Lift จะทำหน้าที่ลำเลียงงานเพาะเชื้อจาก Pusher Petri Dish โดยยกขึ้นไปเก็บที่ Discharge Stack ใช้ Limit Switch ในการกำหนดความแม่นยำของการเคลื่อนที่



รูปที่ 3.4 โครงสร้าง Lift

3.1.2.5 การออกแบบ Discharge Stack

Discharge Stack จะทำหน้าที่เก็บงานเพาะเชื้อที่ผ่านกระบวนการนับแล้ว เข้ามาเก็บเป็นแถวในแนวตั้ง โดยจะมีเดือยที่ทำหน้าที่ในการป้องกันไปให้งานเพาะเชื้อร่วงลงด้านล่าง มีเซ็นเซอร์แสงไวคอยตรวจสอบแถวของงานเพาะเชื้อ เมื่อแถวเต็มแล้วก็จะหมุนเปลี่ยนแถวใหม่เพื่อเก็บงานเพาะเชื้อต่อไป



รูปที่ 3.5 โครงสร้าง Discharge Stack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ภาพรวมโครงสร้างและหลักการทำงาน



รูปที่ 3.6 เครื่องนับและจำแนกสีโคลน

1. หลักการทำงานของเครื่องนับโคลนแบบอัตโนมัติเริ่มจาก In Feed Stack, Discharge Stack, Pusher Petri Dish, Gripper และ Lift เลื่อนเข้าสู่ตำแหน่งเริ่มต้น



รูปที่ 3.7 วางถาดเพาะเชื้อบน In Feed Stack

2. จากนั้น Pusher Petri Dish เลื่อนขึ้นในแกนตั้ง ยกงานเพาะเชื้อขึ้น แล้วเลื่อนไปข้างหน้าในแนวแกนนอน ไป 1 ตำแหน่ง จากนั้นจึงเลื่อนลงในแนวแกนตั้ง และเลื่อนกลับในแนวแกนนอน (x) เพื่อให้ Pusher Petri Dish กลับสู่ตำแหน่งเดิม งานเพาะเชื้อถูกเครื่องสแกนบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Pusher Petri Dish ทำงานแบบเดิมอีกครั้ง Gripper เลื่อนในแนวแกนนอน (y) พาจานเพาะเชื้อเข้าสู่ส่วนถ่ายภาพและประมวลผลภาพเพื่อนับโคโลนีและทำการเก็บข้อมูล

4. Gripper เลื่อนกลับในแนวแกนนอน (y) พาจานเพาะเชื้อกลับมาที่ตำแหน่งเดิม Pusher Petri Dish ทำงานแบบเดิมอีก 2 ครั้ง จนจานเพาะเชื้อไปถึง Lift จึงทำการยกจานเพาะเชื้อขึ้นไปเก็บที่ Discharge Stack โดยที่ In Feed Stack และ Discharge จะมีเซ็นเซอร์แสงคอยตรวจจำนวนจานเพาะเชื้อเพื่อหมุนแถวในการลำเลียงเข้าไปนับและเก็บ เครื่องจะทำงานวนไปอย่างนี้จนทำการนับจานเพาะเชื้อครบ

3.3 การออกแบบซอฟต์แวร์ประมวลผลภาพ

ในการออกแบบซอฟต์แวร์ประมวลผลภาพได้แบ่งออกเป็น 2 ส่วน คือการออกแบบหน้าจออินเตอร์เฟซสำหรับติดต่อผู้ใช้งาน และการออกแบบโปรแกรมวิเคราะห์ประมวลผลภาพ โดยทั้ง 2 การออกแบบได้ใช้ Visual Studio C#2012 เป็นโปรแกรมในการออกแบบซึ่งสามารถทำหน้าจออินเตอร์เฟซพร้อมกับทำโปรแกรมวิเคราะห์ได้พร้อมกัน

3.3.1 การออกแบบหน้าจออินเตอร์เฟซ

หน้าจออินเตอร์เฟซเป็นส่วนสำคัญที่ให้ผู้ใช้งานสามารถควบคุมเครื่อง และดูการทำงานของเครื่อง ทั้งนี้ผลลัพธ์จากการติดต่อกับกล้องหรือการประมวลผลภาพก็ยังคงแสดงออกทางหน้าจออินเตอร์เฟซนี้เช่นกัน โดยมีการออกแบบดังนี้

1. ออกแบบให้มีปุ่มสำหรับเชื่อมต่อกับกล้อง โดยใช้ชื่อ Start Camera



Start Camera

รูปที่ 3.8 ปุ่ม Start Camera

2. ออกแบบให้สามารถทำงานได้ทั้งในระบบอัตโนมัติ และกำหนดเอง

Manual Auto

Run Manual

Run Auto

รูปที่ 3.9 การเลือกรูปแบบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ให้สามารถเลือกสีของโคโลนีที่จะนับ

Colony Color

- Red Yellow
- Green Black
- Blue

รูปที่ 3.10 การเลือกสีโคโลนีที่ต้องการนับ

4. การแสดงผลการนับจำนวนโคโลนีที่ตรวจนับได้

RESULT
TOTAL COLONY

รูปที่ 3.11 หน้าจอแสดงผลการนับโคโลนี

5. มีหน้าต่างแสดงภาพจากกล้องและภาพที่ประมวลผลได้ ซึ่งจะได้นหน้าต่างโปรแกรมตามรูปที่ 3.12

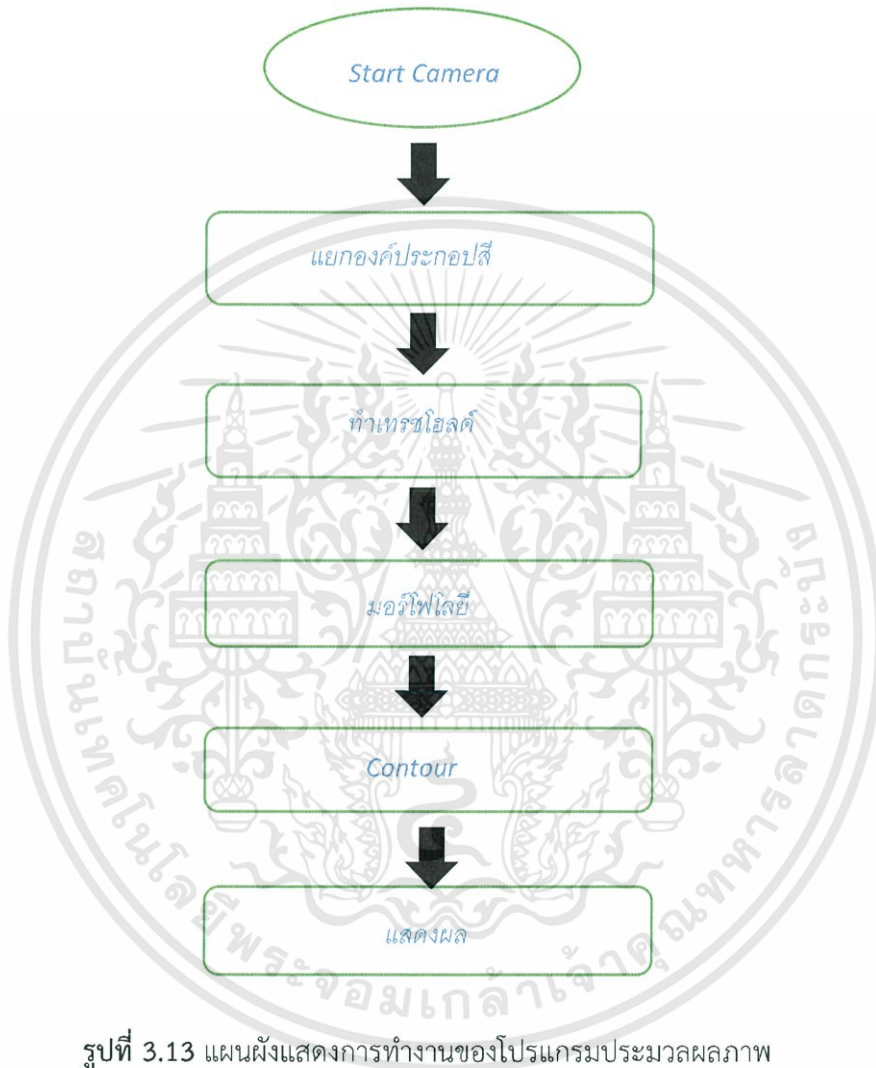


รูปที่ 3.12 หน้าจออินเทอร์เน็ตเฟสของซอฟต์แวร์ประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

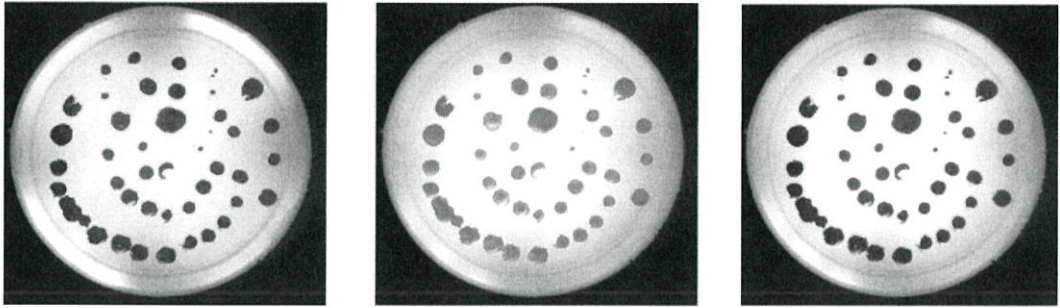
3.3.2 การออกแบบกระบวนการประมวลผลภาพ

การประมวลผลภาพดิจิทัลเป็นขั้นตอนสำคัญในงานวิจัยนี้ เพราะเป็นการทำให้ทราบว่า ภาวะเชื้อที่สนใจมีโคโลนีจำนวนเท่าไร โดยการออกแบบใช้หลักการประมวลผลภาพที่กล่าวมา ในบทที่ 2 ร่วมกับการเขียนโปรแกรม Visual C# 2012 โดยมีรายละเอียดดังนี้



รูปที่ 3.13 แผนผังแสดงการทำงานของโปรแกรมประมวลผลภาพ

1. การแยกองค์ประกอบสีคือ การมองภาพในสีที่กำหนดเท่านั้นโดยในการออกแบบโปรแกรมได้ใช้ 3 สี คือ แดง เขียว และน้ำเงิน ซึ่งในขั้นตอนการทำงานจะทำการแยกองค์ประกอบทั้ง 3 สี และนำมาพิจารณาว่าสีใดทำให้มีความแตกต่างระหว่างพื้นหลังกับวัตถุมากที่สุด ซึ่งภาพที่ได้ออกมาจะเป็นภาพระดับสีเทา ของแต่ละองค์ประกอบสี



(ก) การแยกสีแดง

(ข) การแยกน้ำเงิน

(ค) การแยกเขียว

รูปที่ 3.14 การแยกองค์ประกอบสีของหยดสีน้ำสีแดง

2. เทรซโฮลด์ ภาพของโคโลนีที่ผ่านการทำเป็นภาพระดับสีเทาในองค์ประกอบของสีต่างๆ มาแล้วนั้น ค่าที่เก็บในแต่ละจุดภาพจะเป็นค่าของระดับความสว่างซึ่งจะมีทั้งหมด 256 ระดับ โดยพื้นสีขาวนั้นจะแทนด้วยค่าความสว่าง 255 ส่วนพื้นสีดำจะแทนด้วยค่าความสว่างเป็น 0 โดยในขั้นตอนนี้จะทำให้ภาพระดับสีเทามีแค่สีขาว กับสีดำ โดยการกำหนดค่าเทรซโฮลด์ ซึ่งค่าเทรซโฮลด์นี้จะเป็นตัวกำหนดว่าจุดภาพใดๆ ที่มีค่าระดับความสว่างไม่ได้อยู่ในช่วงที่กำหนดของค่าเทรซโฮลด์ จะเป็นส่วนของพื้นหลัง ส่วนจุดภาพซึ่งมีค่าความสว่างอยู่ในช่วงที่กำหนดนั้น จะเป็นส่วนของโคโลนี

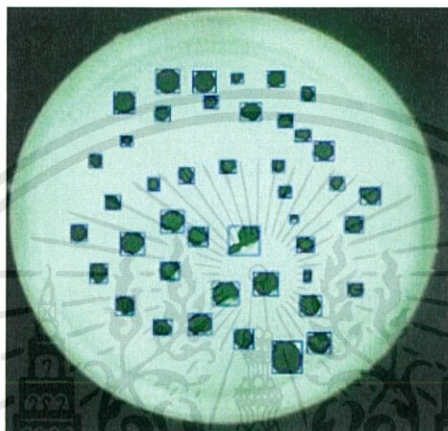


รูปที่ 3.15 ภาพหลังผ่านกระบวนการเทรซโฮลด์

3. มอร์โฟโลยี เป็นขั้นตอนการลบข้อมูลส่วนที่ไม่ต้องการออกจากภาพ ทำได้โดยนำภาพที่ผ่านขั้นตอนทำเทรซโฮลด์หลายระดับมาวิเคราะห์ โดยบริเวณที่เป็นสีขาว ซึ่งในส่วนนี้จะเป็นส่วนของโคโลนีที่ต้องการ แต่จะมีส่วนที่ไม่ใช่โคโลนีปรากฏอยู่ด้วยซึ่งก็คือสัญญาณรบกวน ดังนั้นจึงจำเป็นต้องลบส่วนที่เป็นสัญญาณรบกวนนี้ออกไป ซึ่งใช้ทฤษฎีการเปิด (Opening) โดยกำหนดให้ทำการเซาะ (Erosion) และตามด้วยการขยาย (Dilation) เพื่อให้ขนาดของโคโลนี กลับมามีขนาดที่ใกล้เคียงขนาดเดิม

4. Contour คือ การหารายชื่อของเส้นขอบทุกเส้นภายในรูปโดยจะทำการเก็บข้อมูลไว้สำหรับในการวาด Contour ซึ่งหลังจากการทำเทรซโฮลด์ภาพที่ได้จะมีแค่วัตถุ และพื้นหลังทำให้สามารถแบ่งได้ชัดเจนระหว่างโคโลนิกับพื้นของงานเพาะเชื้อ การหา Contour จะทำการเก็บข้อมูลเส้นขอบของทุกวัตถุจึงมีเท่ากับจำนวนโคโลนิ

5. การแสดงผลภาพ เป็นการนำภาพหลังจากการประมวลผลมาแสดงออกทางหน้าจอ โดยได้ทำการตีกรอบสี่เหลี่ยมรอบจุดที่ทำการนับ



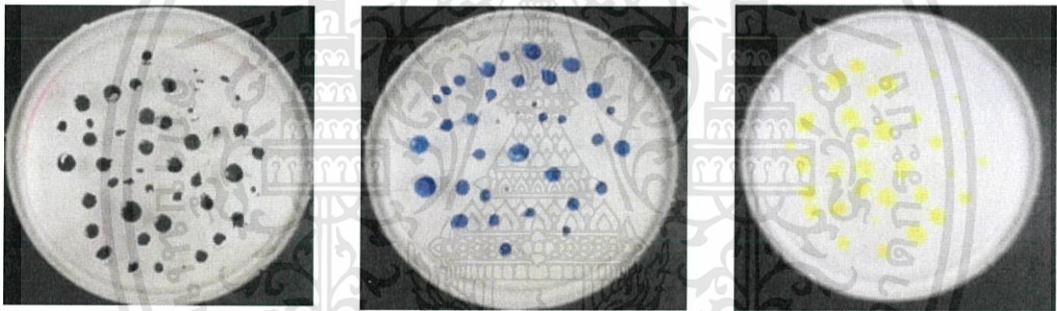
รูปที่ 3.16 การตีกรอบรอบจุดที่ทำการนับแล้ว

บทที่ 4

การทดลองและผลการทดลอง

4.1 ทดสอบซอฟต์แวร์ประมวลผลภาพ

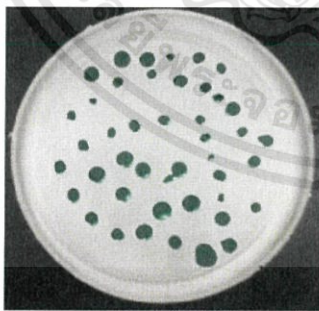
งานวิจัยนี้ได้มีการออกแบบและพัฒนาอัลกอริธึมต่างๆ สำหรับซอฟต์แวร์ประมวลผลภาพ โดยมีการพัฒนาโปรแกรมด้วย Microsoft Visual C# 2012 รวมทั้งมีการออกแบบหน้าต่างติดต่อผู้ใช้ให้ใช้งานและทำความเข้าใจได้ง่าย โดยในการทดลองใช้โปรแกรมจะมีการติดต่อกับกล้อง รับภาพประมวลผล โดยใช้ปุ่มต่างๆ ของโปรแกรม ภาพที่ใช้ทดลองจะใช้สีน้ำยดลงบนจานเพาะเชื้อซึ่งมีความใกล้เคียงกับเชื้อโคลีฟอร์มในห้องปฏิบัติการ ลักษณะการหยดสีจะเป็นแบบสุ่มทั้งขนาดและจำนวนในแต่ละจานเพาะเชื้อ ใช้ทั้งหมด 5 สีคือ แดง ดำ น้ำเงิน เขียว เหลือง



(ก) ใช้แทนโคลีฟอร์มสีดำ

(ข) ใช้แทนโคลีฟอร์มสีน้ำเงิน

(ค) ใช้แทนโคลีฟอร์มสีเหลือง



(ง) ใช้แทนโคลีฟอร์มสีเขียว



(จ) ใช้แทนโคลีฟอร์มสีแดง

รูปที่ 4.1 ภาดหยดสีน้ำที่ใช้น้ำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 ทดลองใช้งานซอฟต์แวร์ประมวลผลภาพ

ในการทดลองการใช้งานเป็นการทดลองว่าโปรแกรมสามารถทำงานได้หรือไม่ทั้งในส่วนการติดต่อกล้อง รับภาพ การประมวลผล และในส่วนของการติดต่อผู้ใช้งานการทำงานของปุ่มต่างๆ โดยในการทดลองใช้คอมพิวเตอร์โน้ตบุ๊ก LENOVO Z510 ในการลงโปรแกรม และใช้กล้อง Microsoft Life Cam HD 3000 เป็นส่วนรับภาพ



รูปที่ 4.2 คอมพิวเตอร์โน้ตบุ๊ก LENOVO Z510

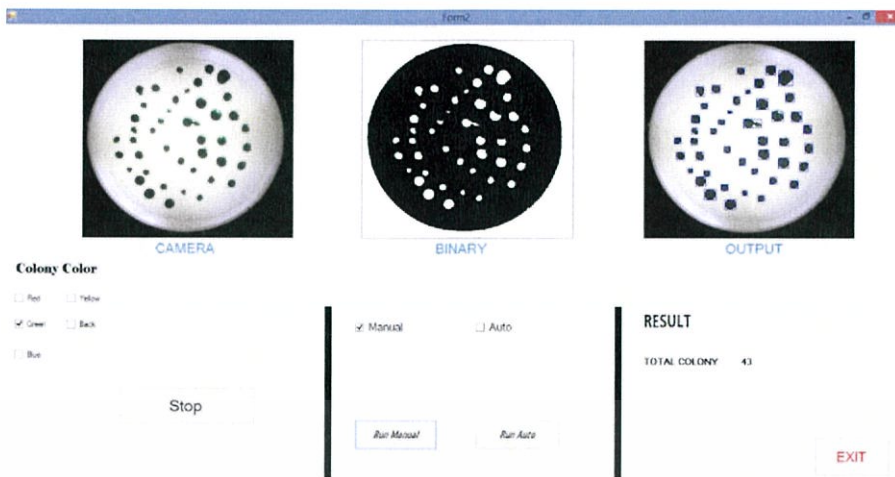
การทำงานของโปรแกรมเริ่มจากการกดปุ่ม Start Camera เพื่อรับภาพจากกล้อง ถ้าโปรแกรมทำงานถูกต้องในช่อง CAMERA จะมีภาพของจานเพาะเชื้อปรากฏตลอดเวลา



รูปที่ 4.3 หน้าต่างโปรแกรมหลังจากการกดปุ่ม CAMERA

ต่อมาผู้ใช้งานสามารถเลือกรูปแบบการทำงานของโปรแกรม โดยมีแบบ AUTO และ MANUAL โดยถ้าเลือกใช้งานแบบ MANUAL ผู้ใช้งานต้องกำหนดสีของโคโลนีเอง และเมื่อกดปุ่มทำงานจะมีจำนวนโคโลนีที่นับได้ในช่อง RESULT โดยสามารถปิดโปรแกรมได้ด้วยปุ่ม EXIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 หน้าต่างการใช้โปรแกรม MANUAL

4.1.2 ทดสอบการทำงานของโปรแกรมโดยใช้พื้นหลังสีขาว


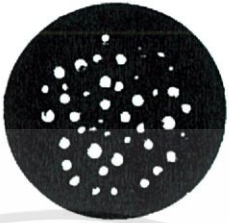
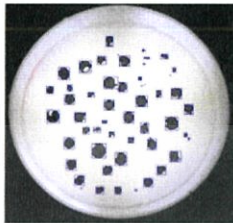
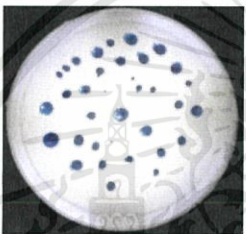

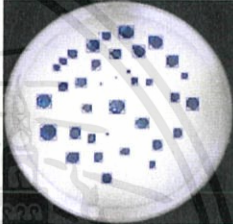
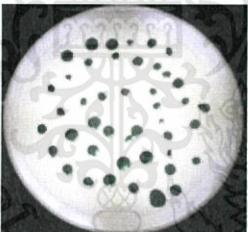

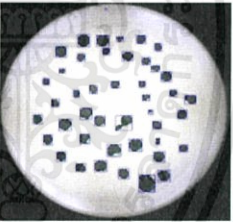
ในการทดลองนี้เป็นการนับโคโลนีโดยไม่ทำการเปลี่ยนสีพื้นของส่วนวางจานเพาะเชื้อคือ ใช้สีขาว โดยใช้จุดสีน้ำจำนวน 5 สี คือ แดง ดำ น้ำเงิน เขียว เหลือง ซึ่งได้ผลลัพธ์ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดสอบโดยใช้พื้นวางจานสีขาว

สี	จำนวน	ภาพจากกล้อง	ภาพเทรชโฮลด์	ภาพผลลัพธ์	จำนวนที่นับได้
แดง	43				44
เหลือง	36				35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทดสอบโปรแกรมโดยใช้พื้นวางงานเพาะเชื้อสีขาว(ต่อ)

สี	จำนวน	ภาพจากกล้อง	ภาพเทอร์ชโฮลด์	ภาพผลลัพธ์	จำนวน ที่นับได้
ดำ	50				47
น้ำเงิน	37				39
เขียว	45				44

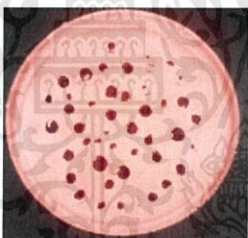

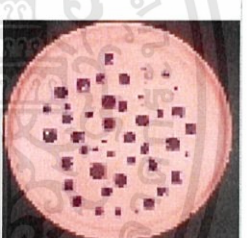
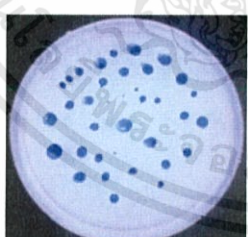

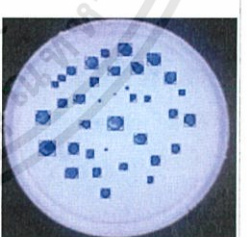
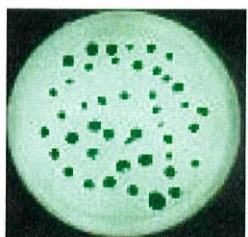

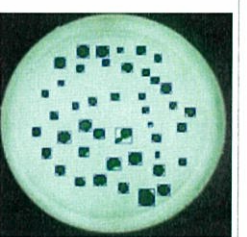
4.1.3 ทดสอบการทำงานของโปรแกรมโดยใช้สีพื้นให้เหมาะสม

ในการทดลองนี้เป็นการนับโคโลนีโดยทำการเปลี่ยนสีพื้นหลังของส่วนวางงานเพาะเชื้อให้เหมาะสมกับสีของหยดสีน้ำ โดยจากการศึกษาพบว่าสีต่างๆ มีพื้นหลังที่เหมาะสมต่างกันการใช้สีพื้นที่เหมาะสม จะทำให้ภาพที่ได้มีความแตกต่างระหว่างพื้นหลังและวัตถุที่สนใจ โดยจากการศึกษาสามารถเลือกสีพื้นหลังที่เหมาะสมตามตารางที่ 4.2 และผลการทดลองตามตารางที่ 4.3

ตารางที่ 4.2 ตารางแสดงสีพื้นหลังที่เหมาะสมกับหยดสีต่างๆ

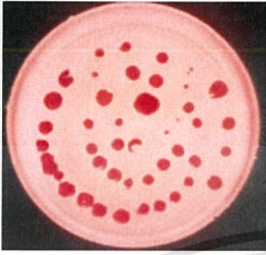

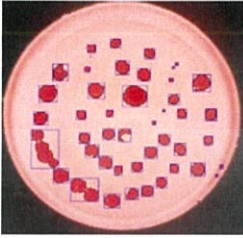
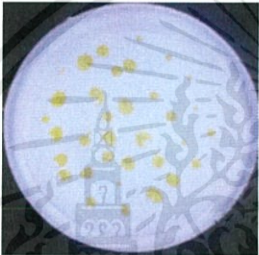
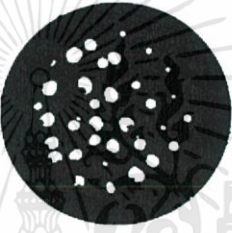
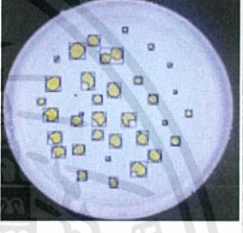
สีของกรดเพาะเชื้อ	สีพื้นหลังที่เหมาะสม
กรดเพาะเชื้อสีแดง	แดง
กรดเพาะเชื้อสีน้ำเงิน	น้ำเงิน
กรดเพาะเชื้อสีเขียว	เขียว
กรดเพาะเชื้อสีดำ	แดง
กรดเพาะเชื้อสีเหลือง	น้ำเงิน

ตารางที่ 4.3 ผลการทดสอบโปรแกรมโดยเปลี่ยนสีพื้นหลังของส่วนวางจานเพาะเชื้อ

สี	จำนวน	ภาพจากกล้อง	ภาพเทอร์ชโฮลด์	ภาพผลลัพธ์	จำนวนที่นับได้
ดำ	50				49
น้ำเงิน	37				37
เขียว	45				43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลการทดสอบโปรแกรมโดยเปลี่ยนสีพื้นหลังของส่วนวางงานเพาะเชื้อ(ต่อ)

สี	จำนวน	ภาพจากกล้อง	ภาพเทอร์ชโฮลด์	ภาพผลลัพธ์	จำนวนที่นับได้
แดง	43				42
เหลือง	36				35

4.1.4 วิเคราะห์และสรุปผลการทดลอง

จากผลการทดลองโปรแกรมประมวลผลภาพเพื่อตรวจนับจำนวนโคโลนีได้ผลลัพธ์คือโปรแกรมสามารถทำงานได้ดีมีความต่อเนื่องในการทำงาน ทั้งในส่วนของการรับภาพจากกล้อง การประมวลผลภาพ การติดต่อกับผู้ใช้งานผ่านปุ่มต่างๆ สามารถประมวลผลได้รวดเร็ว ต่อมาในส่วนของความแม่นยำในการนับจำนวนโคโลนีจากผลการทดลองพบว่า การใช้พื้นหลังสีขาวและการเปลี่ยนสีพื้นหลังให้เหมาะสมกับเชื้อโคโลนีนั้น ให้ผลลัพธ์ที่มีความใกล้เคียงกันมากเปรียบเทียบได้จากตารางที่ 4.1 และตารางที่ 4.3 ซึ่งการที่ผลลัพธ์ใกล้เคียงกันเป็นผลมาจากการที่โปรแกรมประมวลผลภาพได้มีการแยกองค์ประกอบสีมาพิจารณา เป็นองค์ประกอบสีแดง สีเขียว และสีน้ำเงินทำให้ความแม่นยำในการนับเพิ่มขึ้นแม้ว่าใช้พื้นหลังสีขาว ได้ผลตามตารางที่ 4.4

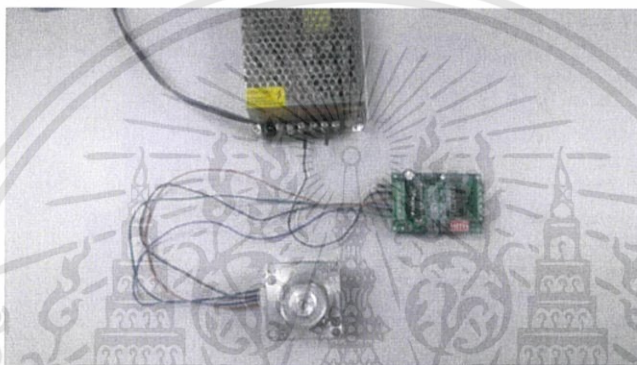
ตารางที่ 4.4 ตารางเปรียบเทียบใช้พื้นหลังสีขาวกับการใช้พื้นหลังที่เหมาะสม

ลักษณะพื้นหลัง	หยดสีแดง	หยดสีเหลือง	หยดสีดำ	หยดสีน้ำเงิน	หยดสีเขียว
ขาว	44	35	47	39	44
สีพื้นที่เหมาะสม	42	35	49	37	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

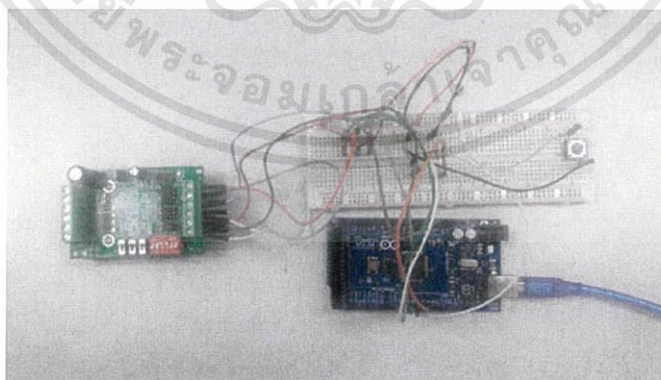
4.2 ทดสอบการทำงานของสเต็ปมอเตอร์

ในงานวิจัยนี้ได้มีการออกแบบและพัฒนาส่วนต่างๆ ของเครื่องนับโคโลนิโดยมีส่วนประกอบหลายส่วน เช่น ส่วนที่วางลาดเพาะเชื้อ ส่วนถ่ายรูป ส่วนเก็บจานเพาะเชื้อ โดยจะมีการเคลื่อนที่ของจานเพาะเชื้อจากที่หนึ่งไปยังอีกที่หนึ่ง ซึ่งได้มีการออกแบบระบบทางกลไว้สำหรับทำหน้าที่ต่างๆ โดยมีต้นกำลังหลักคือ สเต็ปมอเตอร์แบบไบโพลาร์ โดยใช้วงจรมอเตอร์ BL-TB6560 ควบคุมกับการเขียนโปรแกรมควบคุมผ่านไมโครคอนโทรลเลอร์ Arduino mega 2560 โดยการใช้วงจรมอเตอร์ BL-TB6560 แบ่งเป็น 2 ฝั่งคือ ฝั่งต่อมอเตอร์ และฝั่งไมโครคอนโทรลเลอร์ การต่อวงจรมอเตอร์ใช้สายไฟจากตัวมอเตอร์ต่อเข้า A+ A- B+ B- และไฟ 12 VDC จากเพาเวอร์ซัพพลาย



รูปที่ 4.5 การต่อสเต็ปมอเตอร์กับวงจรมอเตอร์ BL-TB6560

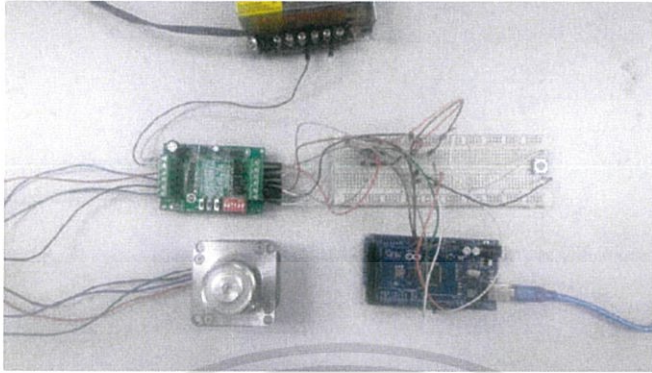
การต่อวงจรทางด้านไมโครคอนโทรลเลอร์กับวงจรมอเตอร์ BL-TB6560 ให้ขา CW- CLK- EN- ต่อเชื่อมกับไฟลบทั้งหมด และขา CW+ ,CLK+ ,EN+ ต่อกับขา OUTPUT ของไมโคร-คอนโทรลเลอร์



รูปที่ 4.6 การต่อวงจรไมโครคอนโทรลเลอร์กับวงจรมอเตอร์ BL-TB6560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำวงจรทั้ง 2 มารวมกันสามารถควบคุมสเต็ปมอเตอร์ได้ตามโค้ดคำสั่งที่เขียนให้กับ Arduino โดยสามารถเขียนคำสั่งด้วยโปรแกรม Arduino Sketch



รูปที่ 4.7 วงจรควบคุมสเต็ปมอเตอร์ ด้วยวงจรขับมอเตอร์ BL-TB6560

4.2.1 ทดสอบทิศทางการหมุนของสเต็ปมอเตอร์ โดยใช้วงจรไดร์ BL-TB6560

การทดลองนี้เป็นการเขียนคำสั่งให้กับ Arduino เพื่อส่ง OUTPUT ไปควบคุมวงจรถับสเต็ปมอเตอร์ โดยทำการทดสอบทิศทางการหมุนของมอเตอร์ว่าตรงกับโปรแกรมที่เขียนหรือไม่

ขั้นตอนการทดลอง

1. ปรับการตั้งวงจรถับมอเตอร์ BL-TB6560 ให้ SW1 ,SW2 , s3 ,s4 เป็น ON และให้SW3 ,s1 ,s2 ,s5, s6 เป็น OFF ตามรูปที่ 4.8



รูปที่ 4.8 การปรับตั้งสวิทช์วงจรถับมอเตอร์ BL-TB6560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เขียนโปรแกรมคำสั่งให้มอเตอร์หมุนโดย CW+ เป็น HIGH

3. เขียนโปรแกรมคำสั่งให้มอเตอร์หมุนโดย CW+ เป็น LOW

สรุปผลการทดลอง

การทดลองนี้เป็นการสั่งให้สเต็ปมอเตอร์หมุนตามเข็มนาฬิกาและหมุนทวนเข็มนาฬิกา โดยวงจรของ BL-TB6560 ที่ควบคุมทิศทางหมุนคือ CW+ ถ้าผู้ใช้งานให้สัญญาณเป็นบวกมอเตอร์จะหมุนตามเข็มนาฬิกา และเมื่อให้สัญญาณลบจะเป็นการสั่งให้หมุนทวนเข็มนาฬิกา ตามตารางที่ 4.5

ตารางที่ 4.5 ตารางแสดงผลการทดลอง CW+ เป็น HIGH และ LOW ด้วย BL-TB6560

สถานะ CW+	ทิศการหมุน(200)	ทิศการหมุน(300)
HIGH	ตามเข็ม	ตามเข็ม
LOW	ทวนเข็ม	ทวนเข็ม

4.2.2 ทดสอบองศาการหมุนของสเต็ปมอเตอร์

การทดลองนี้เป็นการเขียนคำสั่งให้กับ Arduino เพื่อส่ง OUTPUT ไปควบคุมวงจรขับ สเต็ปมอเตอร์ โดยทำการทดสอบองศาการหมุนของมอเตอร์ว่าตรงกับโปรแกรมที่เขียนหรือไม่

ขั้นตอนการทดลอง

1. ปรับการตั้งวงจรขับมอเตอร์ BL-TB6560 ให้ SW1 ,SW2, s3 ,s4 เป็น ON และให้ SW3 s1 ,s2 ,s5, s6 เป็น OFF ตามรูปที่ 4.8

2. เขียนโปรแกรมให้เมื่อกดปุ่มมอเตอร์หมุนไป 50 Step, 100 Step, 150 Step 200 Step, 250 Step ตามลำดับ

สรุปผลการทดลอง

จากการทดลองเขียนโปรแกรมให้สเต็ปมอเตอร์หมุนไปตามจำนวน Step ที่ต้องการสามารถทำให้รู้ว่า สเต็ปมอเตอร์ที่เลือกใช้เคลื่อนที่ครั้งละ 1.8 องศาต่อ 1 Step การเคลื่อนที่ซึ่งตรงตามข้อมูลที่ของสเต็ปมอเตอร์ระบุไว้ ผลการทดลองเป็นไปตามตารางที่ 4.6

ตารางที่ 4.6 องศาการหมุนของสเต็ปมอเตอร์เมื่อหมุนตามจำนวน Step ที่กำหนด

จำนวน Step	50	100	150	200	250
องศาที่เปลี่ยนแปลง	90°	180°	270°	360°	450°

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการวิจัยและข้อเสนอแนะ

เนื้อหาในบทนี้เป็นการสรุปงานวิจัยเครื่องจำแนกสีและนับจำนวนโคโลนีโดยใช้การประมวลผลภาพ สรุปผลการทดลองการใช้โปรแกรมในการนับจำนวนหยดสีน้ำบนภาตเพาะเชื้อ และความแม่นยำในการประมวลผล ตลอดจนการขับสเต็ปมอเตอร์ให้ทำงานตามที่ต้องการโดยมีรายละเอียดดังนี้

5.1 สรุปผลงานวิจัย

งานวิจัยนี้นำเสนอการศึกษาและพัฒนาเครื่องจำแนกสีและนับจำนวนโคโลนีโดยเทคนิคการประมวลผลภาพ เพื่อเป็นการประหยัดเวลาและลดความยุ่งยากในการนับโคโลนี โดยได้แบ่งการออกแบบเป็น 2 ส่วนคือ โปรแกรมประมวลผลภาพ และตัวเครื่องสำหรับลำเลียงภาตเพาะเชื้อมาเก็บภาพ โดยในส่วนของโปรแกรมประมวลผลภาพ โปรแกรมสามารถทำงานได้มีความแม่นยำและมีความเสถียร ทั้งในส่วนของารรับภาพจากกล้องสามารถรับภาพและแสดงทางหน้าจอได้ถูกต้อง การประมวลผลภาพจากการทดลองนับหยดสีน้ำ 5 สีพบว่าสามารถแสดงจำนวนหยดสีได้ใกล้เคียงจำนวนจริงโดยมีความผิดพลาดอยู่ที่ 3.1% โดยที่การเปลี่ยนสีพื้นหลังให้เป็นสีที่เหมาะสมกับการใช้พื้นหลังสีขาว ผลการนับจำนวนหยดสีมีค่าความผิดพลาดใกล้เคียงกัน เนื่องจากการแยกองค์ประกอบสีในกระบวนการวิเคราะห์ ส่วนการติดต่อผู้ใช้งานปุ่มต่างๆ ทำงานได้ถูกต้องตอบสนองรวดเร็ว ต่อมาในส่วนของตัวเครื่องนับโคโลนีการออกแบบเน้นให้มีความแข็งแรง เพียงพอที่จะรับน้ำหนักอุปกรณ์ต่างๆ ที่ติดตั้งมีโดยใช้สแตนเลสเป็นวัสดุหลักในการทำ ความแข็งแรง และลดโอกาสการเกิดปฏิกิริยากับสารเคมีในห้องปฏิบัติการ อุปกรณ์รับภาพใช้กล้อง Logitech c920 ซึ่งมีความละเอียด 2.1 ล้านพิกเซลสามารถต่อกับคอมพิวเตอร์ได้โดยตรงด้วยสายของกล้อง การเปลี่ยนสีพื้นหลังของภาตวางจานเพาะเชื้อใช้ไอซี LED WS2812B ซึ่งสามารถแสดงสีได้ 16 ล้านสี ควบคุมการเปลี่ยนสีได้โดยการใช้ไมโครคอนโทรลเลอร์ Arduino ร่วมกับไลบรารี NeoPixelmaster และส่วนการขับเคลื่อนต่างๆ ใช้สเต็ปมอเตอร์ร่วมกับวงจรขับมอเตอร์ BL-TB6560 สามารถเขียนโปรแกรมควบคุมด้วยไมโครคอนโทรลเลอร์ Arduino โดย CW+ ควบคุมทิศทางหมุน และ CLK+ ควบคุมจำนวนสเต็ปของการหมุนโดยสามารถควบคุมได้ทั้งหมดตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา

5.2 แนวทางการพัฒนาต่อ

1. ในงานวิจัยนี้เกิดปัญหาเกี่ยวกับการเจริญเติบโตของโคโลนีที่บริเวณขอบของจานเพาะเชื้อ ซึ่งการวิเคราะห์ของซอฟต์แวร์ประมวลผลไม่สามารถตรวจนับได้ เนื่องจากซอฟต์แวร์อาศัยภาพจากมุมด้านบนเป็นหลักในการวิเคราะห์ จึงทำให้เกิดกับนับจำนวนผิดพลาด ดังนั้นจึงควรติดตั้งกล้องเพิ่มเพื่อรับภาพจากบริเวณขอบด้านข้างของจานเพาะเชื้อเพื่อนำภาพมาประมวลผลร่วมกับภาพจากกล้องมุมด้านบน

2. ในการออกแบบหน้าต่างติดต่อผู้ใช้ของโปรแกรมประมวลผลภาพควรมีการเพิ่มคำสั่งให้มากขึ้น เช่น 1. เลือกขนาดของโคโลนี 2. เลือกกรุปปร่าง 3. เลือกนับบริเวณที่ต้องการนับ เพื่อให้โปรแกรมมีความสามารถเพิ่มมากขึ้นสามารถทำงานได้อย่างหลากหลายตามความต้องการของผู้ใช้งาน

3. เครื่องนับโคโลนีถูกออกแบบมาให้ตรวจนับเฉพาะโคโลนีที่สามารถมองเห็นได้ด้วยตาเปล่าเท่านั้น แต่ในความเป็นจริงมีเชื้อที่ไม่สามารถมองเห็นได้ด้วยตาเปล่า ซึ่งสามารถมองได้ด้วยการใช้ แบล็คไลท์ ดังนั้นในการออกแบบตัวเครื่องนับโคโลนีจึงควรมีการเพิ่ม แบล็ค-ไลท์เข้าไปด้วยเพื่อให้สามารถทำงานได้ครอบคลุมกับความต้องการของผู้ใช้งาน

4. การทำงานของเครื่องนับโคโลนีเป็นแบบอัตโนมัติโดยการกดปุ่มเริ่มทำงานเพียงปุ่มเดียวทำให้ไม่มีการหยุดในตำแหน่งที่ผู้ใช้งานต้องการ ดังนั้นจึงควรมีการออกแบบเพิ่มความสามารถให้เครื่องนับโคโลนีทำงานทีละขั้นตอนเพื่อให้ผู้ใช้งานสามารถใช้งานได้ง่าย

5. พัฒนาซอฟต์แวร์ประมวลผลภาพให้มีความสามารถมากยิ่งขึ้นลดความผิดพลาดจากการนับ โดยอาจใช้การวิเคราะห์ในโครงสร้างสีแบบอื่นแทน RGB เช่น HSV หรือ LAB

เอกสารอ้างอิง

- [1] ผศ.ดร.สมเกียรติ อุดมธรรษากุล. การประมวลผลภาพเบื้องต้น พิมพ์ครั้งที่ 1 ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สจล., กรุงเทพมหานคร. 2550
- [2] รัตน์พล ยุทธวิริยะ. “เครื่องจำแนกสีและนับจำนวนโคโลนีโดยใช้การประมวลผลภาพ” วิทยานิพนธ์ วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมระบบควบคุม บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2558
- [3] ภิญโญ แยมพราย. “การคัดแยกเซลล์เม็ดเลือดขาวโดยอ้างอิงการวิเคราะห์ส่วนประกอบพื้นฐาน” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2550
- [4] ปุระชัย จงสมชัย. “ชุดตรวจนับเชื้อแบคทีเรียในอากาศโดยอาศัยหลักการการประมวลผลภาพ” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมระบบควบคุม บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2557
- [5] ประจัน พลังสันติกุล. เรียนรู้และใช้งาน CSSและC คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : อินโน-เวทีฟ เอ็กเพอริเมนต์ 2547
- [6] Jiamin Liu, Jacob M. White, Ronald M. Summers “AUTOMATED DETECTION OF BLOB STRUCTURED BY HESSIAN ANALYSIS AND OBJECT SCALE” processing of 2010 IEEE 17th International Conference on Image Processing, September 26-29, 2010 Hong Kong
- [7] Hong Men, Yujie Wu, Xiaoying Li Zhen Kou, Shanrang Yang, “Counting Method of Heterotrophic Bacteria Base on Image Processing” IEEE Publication Year 2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมประมวลผลภาพดิจิทัล

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Threading;

using Emgu.CV;
using Emgu.Util;
using Emgu.CV.UI;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using Emgu.CV.Util;

namespace test_emgu
{
    public partial class oo : Form
    {
        private Capture capture;
        private bool captureInProgress;
        private bool autoclik = true;
        private bool checkAuto = true;
        public oo()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {

            capture = new Capture(1);

            if (capture != null)
            {
                if (captureInProgress)
                { //if camera is getting frames then stop the capture and set
                    button Text
                    // "Start" for resuming capture
                    mm.Text = "Start Camera"; //
                    Application.Idle -= ProcessFrame;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
    //if camera is NOT getting frames then start the capture
and set button
    // Text to "Stop" for pausing capture
    mm.Text = "Stop";
    Application.Idle += ProcessFrame;
}

captureInProgress = !captureInProgress;
}

}

private void ProcessFrame(object sender, EventArgs arg)
{
    Mat cannyImage = new Mat();
    Mat realtime = new Mat();
    Mat gray = new Mat();
    Mat gray2 = new Mat();
    Mat result = new Mat();
    Mat thresImage = new Mat();
    VectorOfMat constorge = new VectorOfMat();
    Mat elem = new Mat();
    elem = CvInvoke.GetStructuringElement(ElementShape.Rectangle, new
Size(3, 3), new Point(-1, -1));

    realtime = capture.QueryFrame();

    Mat roiImage = new Mat(realtime, new Rectangle(116, 36, 400, 390));
    camera.Image = roiImage;
}

private void ReleaseData()
{
    if (capture != null)
        capture.Dispose();
}

private void bt2_Click(object sender, EventArgs e)
{
    if (Manual.Checked)
    {
        Manualprogram();
    }
    else { }
}

private void Manualprogram()
{
    Mat OrgImage = new Mat();
    Mat OrgImage2 = new Mat();
    Mat bandcolor = new Mat();
    Mat binImage = new Mat();
    Mat binImage2 = new Mat();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Mat gray = new Mat();
Mat ero = new Mat();
Mat ero2 = new Mat();
Mat dilate = new Mat();
Mat dilate2 = new Mat();
Mat dilate3 = new Mat();
Mat rect = new Mat();
Mat elem = new Mat();
int total=0;
int j = 0;
VectorOfMat constorge = new VectorOfMat();
VectorOfMat splitImage = new VectorOfMat();

OrgImage = capture.QueryFrame(); //Load Image
Mat roiImage = new Mat(OrgImage, new Rectangle(116, 36, 400, 390));
OrgImage2 = roiImage.Clone();

// splite Image
CvInvoke.Split(OrgImage2, splitImage);
if (checkblue.Checked || checkblack.Checked)
bandcolor = splitImage[2].Clone();
else if (checkred.Checked || checkyellow.Checked)
bandcolor = splitImage[0].Clone();
else if (checkgreen.Checked )
    bandcolor = splitImage[1].Clone();
else if (!checkblue.Checked && !checkred.Checked &&
!checkgreen.Checked)
{ bandcolor = roiImage.Clone(); }

//Threshold
CvInvoke.Threshold(bandcolor, binImage, 140, 255,
ThresholdType.BinaryInv | ThresholdType.Otsu);

//create Structure
elem = CvInvoke.GetStructuringElement(ElementShape.Rectangle , new
Size(3, 3), new Point(-1, -1));

binImage2 = binImage.Clone();

//Opening
CvInvoke.Dilate(binImage2, dilate, elem, new Point(-1, -1), 1,
BorderType.Default, new MCvScalar(0, 0, 0));
CvInvoke.Erode(dilate, ero, elem, new Point(-1, -1), 1,
BorderType.Default, new MCvScalar(0, 0, 0));
// camera3.Image = ero; //show Image
ero2 = ero.Clone();
//Find Countours
CvInvoke.FindContours(ero2, constorge, null, RetrType.List,
ChainApproxMethod.ChainApproxSimple);

//Draw all countours
int k = constorge.Size;

for (int i = 0; i < k; i++)
{
    double conA = CvInvoke.ContourArea(constorge[i]);

    if (conA < 1500)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            CvInvoke.DrawContours(OrgImage2, constorge, i, new
MCvScalar(0, 0, 0), 1, LineType.FourConnected, null, 1);

            Rectangle rec = CvInvoke.BoundingRectangle(constorge[i]);

            CvInvoke.Rectangle(OrgImage2, rec, new MCvScalar(255, 0,
0), 1, LineType.EightConnected, 0);
        }
        else
        {
            j++;
        }
    }
    total = (constorge.Size) -j;
    t1.Text = total.ToString();
    camera2.Image = binImage;
    camera3.Image = OrgImage2;

}

private void im_Click(object sender, EventArgs e)
{
}

private void camera_MouseMove(object sender, MouseEventArgs e)
{
    int offsetX = (int)(e.Location.X / camera.ZoomScale);
    int offsetY = (int)(e.Location.Y / camera.ZoomScale);
    int horizontalScrollBarValue = camera.HorizontalScrollBar.Visible ?
(int)camera.HorizontalScrollBar.Value : 0;
    int verticalScrollBarValue = camera.VerticalScrollBar.Visible ?
(int)camera.VerticalScrollBar.Value : 0;
    // tb2.Text = Convert.ToString(offsetX + horizontalScrollBarValue) +
"." + Convert.ToString(offsetY + verticalScrollBarValue);
}

private void autobt_Click(object sender, EventArgs e)
{
    if (Auto.Checked)
    {
        autoclik = !autoclik;
        if (autoclik == false)
        {
            autobt.Text = " Pause";
            Application.Idle += Autoprogram;
        }
        else{ autobt.Text = " Run Auto";
            Application.Idle -= Autoprogram;
        }
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private async void Autoprogram(object sender, EventArgs arg)
{
    Mat OrgImage = new Mat();
    Mat OrgImage2 = new Mat();
    Mat bandcolor = new Mat();
    Mat binImage = new Mat();
    Mat binImage2 = new Mat();
    Mat ThresImage = new Mat();
    Mat ThresImage2 = new Mat();
    Mat gray = new Mat();
    Mat ero = new Mat();
    Mat ero2 = new Mat();
    Mat dilate = new Mat();
    Mat dilate2 = new Mat();
    Mat dilate3 = new Mat();
    Mat rect = new Mat();
    Mat elem = new Mat();
    int j = 0;
    int total = 0;
    VectorOfMat constorgeA = new VectorOfMat();
    VectorOfMat constorgeA2 = new VectorOfMat();
    VectorOfMat constorgeAmain = new VectorOfMat();
    VectorOfMat splitImage = new VectorOfMat();

    //Auto
    VectorOfMat splitband = new VectorOfMat();
    VectorOfMat contourred = new VectorOfMat();
    VectorOfMat contourgreen = new VectorOfMat();
    VectorOfMat contourblue = new VectorOfMat();

    Mat testAuto = new Mat();
    Mat redband = new Mat();
    Mat greenband = new Mat();
    Mat blueband = new Mat();
    Mat Autobandcolor = new Mat();
    Mat thresAuto = new Mat();
    Mat AutobinImage = new Mat();
    Mat AutobinImage2 = new Mat();
    Mat redbinImage = new Mat();
    Mat greenbinImage = new Mat();
    Mat bluebinImage = new Mat();

    //Find right picture
    OrgImage = capture.QueryFrame(); //Load Image
    Mat roiImage = new Mat(OrgImage, new Rectangle(116, 36, 400, 390));
    CvInvoke.CvtColor(roiImage, binImage, ColorConversion.Bgr2Gray);
    CvInvoke.Threshold(binImage, ThresImage, 0, 255,
ThresholdType.Otsu);
    CvInvoke.FindContours(ThresImage, constorgeA, null, RetrType.List,
ChainApproxMethod.ChainApproxSimple);

    //Auto capture
    if (constorgeA.Size >30){
        await Task.Delay(7000);
        testAuto = capture.QueryFrame();

        Mat testAutoROI =new Mat(testAuto, new Rectangle(116, 36, 400,
390));

        CvInvoke.Split(testAutoROI,splitband );
        redband = splitband[0].Clone();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

greenband = splitband[1].Clone();
blueband = splitband[2].Clone();

CvInvoke.Threshold(redband, redbinImage, 140, 255,
ThresholdType.BinaryInv | ThresholdType.Otsu);
CvInvoke.Threshold(greenband, greenbinImage, 140, 255,
ThresholdType.BinaryInv | ThresholdType.Otsu);
CvInvoke.Threshold(blueband, bluebinImage, 140, 255,
ThresholdType.BinaryInv | ThresholdType.Otsu);

CvInvoke.FindContours(redbinImage, contourred, null,
RetrType.List, ChainApproxMethod.ChainApproxSimple);
CvInvoke.FindContours(greenbinImage, contourgreen, null,
RetrType.List, ChainApproxMethod.ChainApproxSimple);
CvInvoke.FindContours(bluebinImage, contourblue, null,
RetrType.List, ChainApproxMethod.ChainApproxSimple);

if ((contourred.Size > contourgreen.Size) && (contourred.Size >
contourblue.Size))
{
    Autobandcolor = splitband[0];
}
else if ((contourgreen.Size > contourred.Size) &&
(contourgreen.Size > contourblue.Size))
{
    Autobandcolor = splitband[1];
}
else if ((contourblue.Size > contourred.Size) &&
(contourblue.Size > contourgreen.Size))
{
    Autobandcolor = splitband[2];
}

CvInvoke.Threshold(Autobandcolor, AutobinImage, 140, 255,
ThresholdType.BinaryInv | ThresholdType.Otsu);

AutobinImage2 = AutobinImage.Clone();
CvInvoke.FindContours(AutobinImage2, constorgeAmain, null,
RetrType.List, ChainApproxMethod.ChainApproxSimple);

//Draw all countours
int k = constorgeAmain.Size;

for (int i = 0; i < k; i++)
{
    double conA = CvInvoke.ContourArea(constorgeAmain[i]);

    if (conA < 1500)
    {
        CvInvoke.DrawContours(testAutoROI, constorgeAmain, i, new
MCvScalar(0, 0, 0), 1, LineType.FourConnected, null, 1);

        Rectangle rec =
CvInvoke.BoundingRectangle(constorgeAmain[i]);

        CvInvoke.Rectangle(testAutoROI, rec, new MCvScalar(255,
0, 0), 1, LineType.EightConnected, 0);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
        {
            j++;
        };
    }

    total = (constorgeAmain.Size) - j;
    t1.Text = total.ToString();
    camera2.Image = AutobinImage;
    camera3.Image = testAutoROI;
}

}

private void oo_Load(object sender, EventArgs e)
{
}

private void Reset_Click(object sender, EventArgs e)
{
    Application.Exit();
}

}

}
```





ภาคผนวก ข

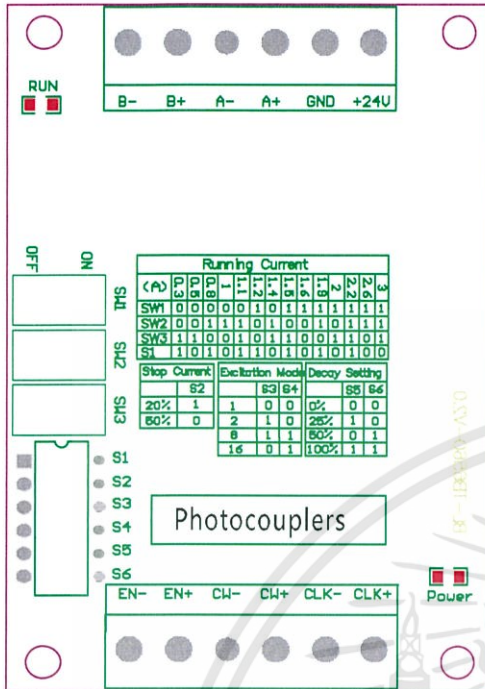
การใช้งาน วงจรขับเคลื่อน BL-TB6560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tb6560 stepping motor driver V20

Warning:

1. Check the connection twice! The Tb6560 chipset can be damaged if the motor or the power supply are not connected properly.
2. Don't apply a motor that its rated current is more than 3A to this driver.
3. Do not set the current more than the motor rated current!



Wiring Terminal symbol	Description
+24V, GND	Power positive and negative
A+, A-	Motor phase A
B+, B-	Motor phase B
CLK+, CLK-	Pulse positive and negative
CW+, CW-	Direction positive and negative
EN+, EN-	Enable positive and negative

Note:

1. 6 input terminals, can be connected as common anode or cathode.
2. The normal input voltage is 5V, if it is more than 5V, than a series resistor is needed. this resistance is 1K case 12V and 2.4K case 24V.
3. when pulse is applied to CLK, the stepping motor will rotate, and stop when there is none, and the motor driver will change its current to the half current mode as setting to hold the motor still.
4. Motor rotate clockwise when CW is low level and counterclockwise when CW is high level.
5. Motor is enable when EN is low level and disable when EN is high level.

Running Current														
(A)	0.3	0.5	0.8	1	1.1	1.2	1.4	1.5	1.6	1.9	2	2.2	2.6	3
SW1	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	ON	ON	ON	ON	ON	ON
SW2	OFF	OFF	ON	ON	ON	OFF	ON	OFF	OFF	ON	OFF	ON	ON	ON
SW3	ON	ON	OFF	OFF	ON	OFF	ON	ON	OFF	OFF	ON	ON	OFF	ON
S1	ON	OFF	ON	OFF	ON	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF

Stop Current	
	S2
20%	ON
50%	OFF

Excitation Mode		
Step	S3	S4
whole	OFF	OFF
half	ON	OFF
1/8	ON	ON
1/16	OFF	ON

Decay Setting		
	S5	S6
0%	OFF	OFF
25%	ON	OFF
50%	OFF	ON
100%	ON	ON



ภาคผนวก ค

Datasheet WS2812B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



WS2812B

Intelligent control LED
integrated light source

Features and Benefits

- Intelligent reverse connect protection, the power supply reverse connection does not damage the IC.
- The control circuit and the LED share the only power source.
- Control circuit and RGB chip are integrated in a package of 5050 components, form a complete control of pixel point.
- Built-in signal reshaping circuit, after wave reshaping to the next driver, ensure wave-form distortion not accumulate.
- Built-in electric reset circuit and power lost reset circuit.
- Each pixel of the three primary color can achieve 256 brightness display, completed 16777216 color full color display, and scan frequency not less than 400Hz/s.
- Cascading port transmission signal by single line.
- Any two point the distance more than 5m transmission signal without any increase circuit.
- When the refresh rate is 30fps, cascade number are not less than 1024 points.
- Send data at speeds of 800Kbps.
- The color of the light were highly consistent, cost-effective..

Applications

- Full-color module, Full color soft lights a lamp strip.
- LED decorative lighting, Indoor/outdoor LED video irregular screen.

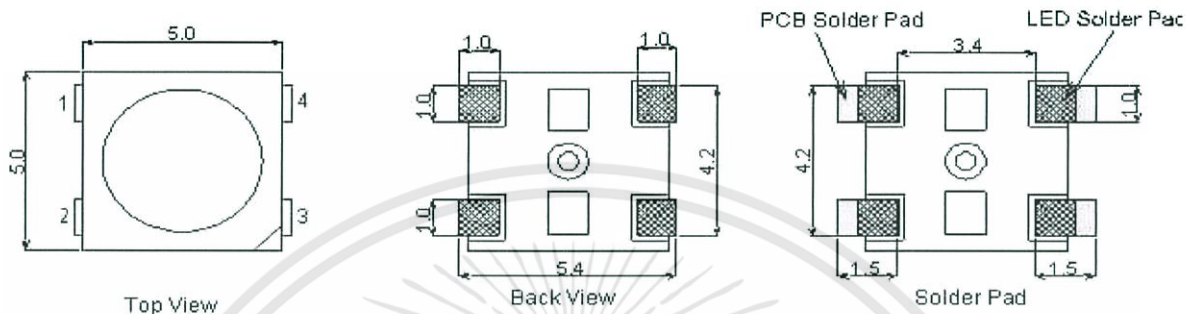
General description

WS2812B is a intelligent control LED light source that the control circuit and RGB chip are integrated in a package of 5050 components. It internal include intelligent digital port data latch and signal reshaping amplification drive circuit. Also include a precision internal oscillator and a 12V voltage programmable constant current control part, effectively ensuring the pixel point light color height consistent.

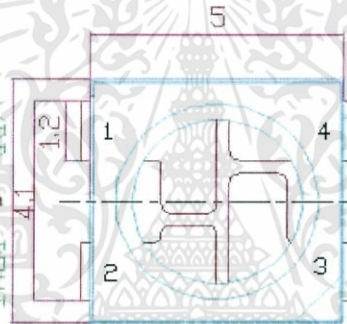
The data transfer protocol use single NZR communication mode. After the pixel power-on reset, the DIN port receive data from controller, the first pixel collect initial 24bit data then sent to the internal data latch, the other data which reshaping by the internal signal reshaping amplification circuit sent to the next cascade pixel through the DO port. After transmission for each pixel, the signal to reduce 24bit. pixel adopt auto reshaping transmit technology, making the pixel cascade number is not limited the signal transmission, only depend on the speed of signal transmission.

LED with low driving voltage, environmental protection and energy saving, high brightness, scattering angle is large, good consistency, low power, long life and other advantages. The control chip integrated in LED above becoming more simple circuit, small volume, convenient installation.

Mechanical Dimensions



PIN configuration



PIN function

NO.	Symbol	Function description
1	VDD	Power supply LED
2	DOUT	Control data signal output
3	VSS	Ground
4	DIN	Control data signal input

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Power supply voltage	V _{DD}	+3.5~+5.3	V
Input voltage	V _I	-0.5~V _{DD} +0.5	V
Operation junction temperature	T _{opt}	-25~+80	°C
Storage temperature range	T _{stg}	-55~+150	°C

Electrical Characteristics (T_A=-20~+70°C, V_{DD}=4.5~5.5V, V_{SS}=0V, unless otherwise specified)

<http://www.world-semi.com> บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



WS2812B

Intelligent control LED
integrated light source

Parameter	Symbol	conditions	Min	Tpy	Max	Unit
Input current	I_I	$V_I=V_{DD}/V_{SS}$	—	—	± 1	μA
Input voltage level	V_{IH}	D_{IN}, SET	$0.7V_{DD}$	—	—	V
	V_{IL}	D_{IN}, SET	—	—	$0.3V_{DD}$	V
Hysteresis voltage	V_H	D_{IN}, SET	—	0.35	—	V

Switching characteristics ($T_A=-20\sim+70^\circ C$, $V_{DD}=4.5\sim 5.5V, V_{SS}=0V$, unless otherwise specified)

Parameter	Symbol	Condition	Min	Tpy	Max	Unit
Operation frequency	F_{osc2}	—	—	800	—	KHz
Transmission delay time	t_{PLZ}	$CL=15pF, D_{IN}\rightarrow D_{OUT}, RL=10K\Omega$	—	—	300	ns
Fall time	t_{THZ}	$CL=300pF, OUTR/OUTB \rightarrow TG/OUTB$	—	—	120	μs
Data transmission rate	F_{MAX}	Duty ratio 50%	400	—	—	Kbps
Input capacity	C_I	—	—	—	15	pF



WS2812B

Intelligent control LED
integrated light source

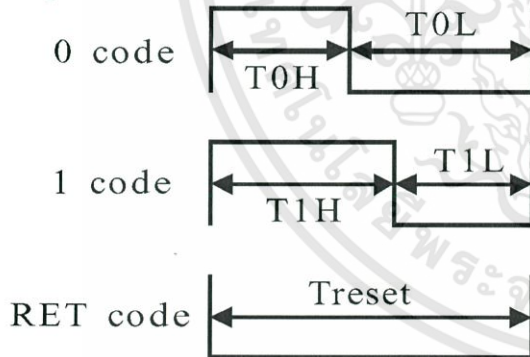
LED characteristic parameter

Emitting color	Model	Wavelength(nm)	Luminous intensity(mcd)	Voltage(V)
Red	13CBAUP	620-630	550-700	1.8-2.2
Green	13CGAUP	515-530	1100-1400	3.0-3.2
Blue	10R1MUX	465-475	200-400	3.0-3.4

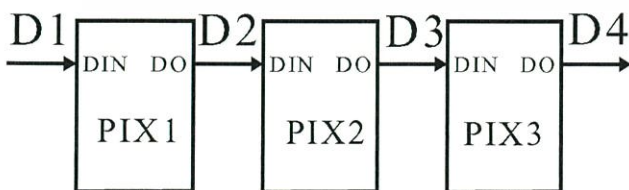
Data transfer time(TH+TL=1.25μs±150ns)

T0H	0 code ,high voltage time	0.35us	±150ns
T1H	1 code ,high voltage time	0.9us	±150ns
T0L	0 code , low voltage time	0.9us	±150ns
T1L	1 code ,low voltage time	0.35us	±150ns
RES	low voltage time	Above 50μs	

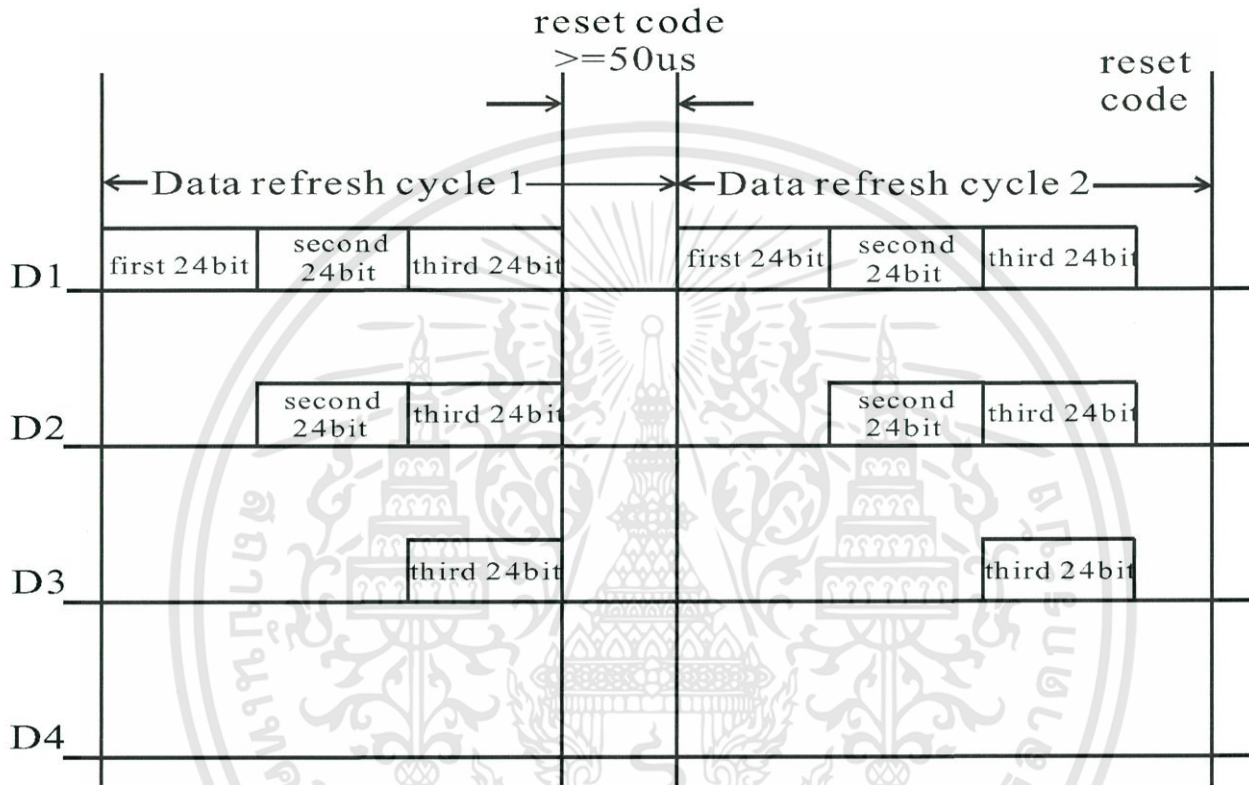
Sequence chart:



Cascade method:



Data transmission method:



Note: The data of D1 is send by MCU, and D2, D3, D4 through pixel internal reshaping amplification to transmit.

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Follow the order of GRB to sent data and the high bit sent at first.

Typical application circuit:

