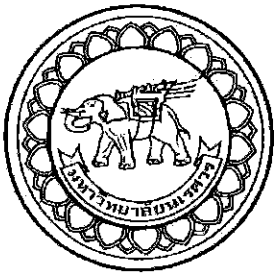
อภินันทนาการ

สำนักหอสมุด



การคัดแยกวัตถุตามสีโดยใช้ตัวรับรู้ TCS230

COLOR-BASED CATEGORIZATION BY USING TCS230 SENSOR

นายกิตติทัต แซ่เฮง รหัส 55363780

นายสุวินัย น้อยวงศ์ รหัส 55364428

สำนักหอสมุด มหาวิทยาลัยนเรศวร

วันลงทะเบียน 17 ก.ค. 2560

เลขทะเบียน.....

เลขเรียกหนังสือ.....

ป
ก17ก
J56ป

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2558

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การคัดแยกวัตถุตามสีโดยใช้ตัวรับรู้ TCS230	
ผู้ดำเนินโครงงาน	นายกิตติทัต แซ่เฮง	รหัส 55363780
	นายสุวินัย น้อยวงศ์	รหัส 55364428
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2558	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. ศุภวรรณ พลพิทักษ์ชัย)

.....กรรมการ
(ดร. สราวุฒิ วัฒนวงค์พิทักษ์)

ชื่อหัวข้อโครงงาน	การคัดแยกวัตถุตามสีโดยใช้ตัวรับรู้ TCS230
ผู้ดำเนินโครงงาน	นายกิตติทัต แซ่เฮง รหัส 55363780
	นายสุวินัย น้อยวงศ์ รหัส 55364428
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2558

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการออกแบบเครื่องคัดแยกวัตถุตามสีโดยใช้ตัวรับรู้ TCS230 โดยสร้างแบบจำลองของเครื่องเพื่อคัดแยกลูกปิงปองสีแดง สีเขียว และสีน้ำเงิน และใช้แอลดีอาร์ในวงจรตรวจจับลูกปิงปอง รวมถึงมอเตอร์เซอร์โวหมุนจานลำเลียงเพื่อส่งลูกปิงปองลงกล่องบรรจุตามเงื่อนไขที่ผู้ใช้ต้องการซึ่งกำหนดได้ 2 แบบวิธีคือ แบบวิธีคัดแยกและแบบวิธีคัดเลือกและควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ ในแบบวิธีคัดแยก เครื่องจะคัดแยกลูกปิงปองสีแดงลงกล่อง A ลูกปิงปองสีน้ำเงินลงกล่อง B และลูกปิงปองสีเขียวลงกล่อง C โดยหน้าจอแอลซีดีจะแสดงจำนวนรวมของลูกปิงปองแต่ละสีที่ตรวจพบ การทำงานในแบบวิธีคัดเลือกจะเริ่มจากการให้ผู้ใช้ระบุจำนวนของลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง A และ B ในขณะที่กล่อง C ถูกกำหนดเป็นกล่อง Spare เพื่อใส่ลูกปิงปองที่ไม่ตรงกับเงื่อนไขของผู้ใช้หรือเมื่อกล่อง A หรือ B มีลูกปิงปองครบตามเงื่อนไขแล้ว ผลการทดสอบแสดงให้เห็นว่าแบบจำลองของเครื่องคัดแยกวัตถุตามสีที่สร้างขึ้นสามารถทำงานได้ถูกต้องตามเงื่อนไขที่ออกแบบไว้สำหรับทั้งสองแบบวิธี

Project title	Color-Based Categorization By-Using TCS230 Sensor
Name	Mr.Kittitat Saeheng ID. 55363780
	Mr.Suwinai Noiwong ID. 55364428
Project advisor	Asst. Prof. Niphat Jantharamin, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2015

Abstract

This thesis presents a design of color-based categorization of objects according to RGB color system by using TCS230 Sensor. The idea leads to construction of a model which, determined by the user, can operate in Separation mode and Selection mode. The system operation is regulated by a microcontroller. The model demonstrates each operating mode by using ping-pongs painted in red, green, and blue. In Separation mode, the ping-pongs are separated according to their color, namely red ones being moved to box A, blue ones to box B, and green ones to box C. In Selection mode, the ping-pongs are chosen according to the number and color of the ping-pongs specified by the user for boxes A and B while box C is programmed to be a Spare box. Ping-pongs are sent to box C if they do not match the user-defined conditions for boxes A and B of after selection task is fulfilled. Experimental results show that the model successfully performs user-defined tasks in both operating modes.

กิตติกรรมประกาศ

ผู้ดำเนินโครงงานขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์ ที่ปรึกษาโครงงาน ซึ่งเอาใจใส่ในรายละเอียดทุกขั้นตอนของการดำเนินโครงงาน โดยให้คำปรึกษาและคำแนะนำในการแก้ไขปัญหาต่าง ๆ อย่างต่อเนื่องจนกระทั่งโครงงานสำเร็จลุล่วง รวมถึงแนะนำหลักการเขียนปริญญานิพนธ์และตรวจทานแก้ไขอย่างละเอียดจนได้ปริญญานิพนธ์เป็นรูปเล่มสมบูรณ์

ขอขอบคุณกรรมการสอบโครงงาน ผู้ช่วยศาสตราจารย์ ดร.ศุภวรรณ พลพิทักษ์ชัย และ ดร. สราวุฒิ วัฒนวงค์พิทักษ์ ซึ่งกรุณาให้คำแนะนำที่เป็นประโยชน์ในการดำเนินโครงงาน

ขอขอบคุณนายกฤษฎิ์ สวัสดี (แบงค์) เพื่อนสาขาวิชาวิศวกรรมไฟฟ้า ที่ให้คำแนะนำในการเขียนโปรแกรมควบคุมการทำงานของเครื่องแยกวัตถุตามสีโดยใช้ตัวรับรู้ TCS230

ขอขอบคุณนายรอนรุจน์ อุ่นฤกษ์ (จอร์ส) เพื่อนสาขาวิศวกรรมไฟฟ้า ที่ให้คำแนะนำในการใช้โปรแกรมตัดต่อวิดีโอ

และขอขอบคุณนายจิรวัฒน์ ช่างเก็บ (ตุ๊ก) เพื่อนสาขาวิชาวิศวกรรมไฟฟ้า ที่ให้ยืมใช้เครื่องมือและอุปกรณ์ช่าง เพื่อใช้ในการทำโครงสร้างเครื่องแยกวัตถุตามสีโดยใช้ตัวรับรู้ TCS230

ในท้ายที่สุดนี้ เหนือสิ่งอื่นใด ผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดาและมารดาที่ให้การสนับสนุนในทุกด้านเกี่ยวกับการศึกษาของผู้ดำเนินโครงงาน รวมทั้งมอบความรัก ความเมตตา และคอยเป็นกำลังใจให้จนประสบความสำเร็จในวันนี้

นายกิตติทัต แซ่เฮง

นายสุวินัย น้อยวงศ์

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 งบประมาณ 3

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 4
2.1 คุณสมบัติของสีที่เกี่ยวข้องกับตัวรับรู้ทางสี 4
2.1.1 สเปกตรัมแสงขาว 4
2.1.2 แม่สีของแสง 5
2.1.3 คุณสมบัติการดูดกลืนสีของวัตถุและสะท้อนสีของแสง 6
2.2 หลักการทำงานของตัวรับรู้หมายเลข TCS230 8
2.3 แผงวงจร Arduino 11
2.4 จอแสดงผลแอลซีดี 13
2.5 ส่วนประกอบของมอเตอร์เซอร์โว 14
2.5.1 ขนาดของมอเตอร์เซอร์โว 15
2.5.2 ประเภทของมอเตอร์เซอร์โว 16
2.6 แผงแป้นตัวเลข 17
2.7 แอลดีอาร์ 18

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและสร้างแบบจำลองการคัดแยกวัตถุตามสี 19
3.1 ส่วนประกอบหลักการทำงานของระบบการคัดแยกวัตถุตามสี 19
3.1.1 ไมโครคอนโทรลเลอร์ 20
3.1.2 วงจรตัวรับรู้สี 21
3.1.3 วงจรแผงแป้นตัวเลข 22
3.1.4 วงจรตรวจจับลูกปิงปอง 22
3.1.5 วงจรคุมค่าแรงดัน 5 V 24
3.1.6 วงจรหน้าจอแสดงผล 24
3.1.7 มอเตอร์เซอร์โว 25
3.2 การออกแบบโครงสร้างและการติดตั้งอุปกรณ์ในแบบจำลอง 27
3.3 การออกแบบการทำงานของเครื่องคัดแยกวัตถุตามสี 32
3.3.1 การออกแบบขั้นตอนการทำงานโดยรวม 32
3.3.2 แบบวิธีคัดแยกสี 32
3.3.3 แบบวิธีคัดเลือกสี 33
3.4 รูปแบบการแสดงผลบนหน้าจอแอลซีดีขณะใช้งาน 35
3.4.1 การแสดงผลขณะใช้งานในแบบวิธีการคัดแยก 35
3.4.2 การแสดงผลในแบบวิธีคัดเลือก 36

บทที่ 4 ผลการทดสอบ 42
4.1 การทดสอบการทำงานของแบบวิธีคัดแยกและแบบวิธีคัดเลือก 42
4.1.1 การทดสอบการคัดแยก 42
4.1.2 การทดสอบการคัดเลือก 44
4.2 การทดสอบหาเวลาที่ใช้ในการคัดแยกสีลูกปิงปองของเครื่อง 50

บทที่ 5 สรุปผลและข้อเสนอแนะ 51
5.1 สรุปผลการดำเนินงาน 51
5.2 ปัญหาและแนวทางการแก้ไข 51
5.3 แนวทางในการพัฒนาต่อไป 52

สารบัญ (ต่อ)

หน้า

เอกสารอ้างอิง 53

ภาคผนวก ก รหัสต้นฉบับของโปรแกรมควบคุมเครื่องคัดแยกวัตถุตามสี ด้วยตัวรับรู้ TCS230.....54
ภาคผนวก ข รายละเอียดข้อมูลของ ATmega2560 73
ภาคผนวก ค รายละเอียดข้อมูลของตัวรับรู้ TCS230 76
ภาคผนวก ง รายละเอียดข้อมูลของแผงแป้นตัวเลข 79
ภาคผนวก จ รายละเอียดข้อมูลของจอแสดงผล 83

ประวัติผู้ดำเนินโครงงาน 86

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงสัญญาณเอาท์พุต....8
2.2 แสดงการขยายสัญญาณ....9
3.1 หน้าที่ของอุปกรณ์ต่างๆในแบบจำลองเครื่องคัดแยกวัตถุตามสี....28
3.2 รายละเอียดของอุปกรณ์ต่างๆในแบบจำลองเครื่องคัดแยกวัตถุตามสี....30
4.1 ผลการทดสอบหาระยะเวลาในการคัดแยกลูกปิงปองแต่ละสี....50

สารบัญรูป

รูปที่ หน้า

2.1 ปรากฏการณ์ของก้อนปริซึม 4
2.2 สเปกตรัมของแสงสีขาว 5
2.3 การผสมของแสงสีปฐมภูมิ 5
2.4 การตกกระทบและสะท้อนของแสงจากวัตถุ 6
2.5 การสะท้อนกลับหมดของแสงสีขาวที่ส่องลงบนวัตถุสีขาว 6
2.6 การสะท้อนและดูดกลืนของแสงสีขาวเมื่อตกกระทบวัตถุที่มีสีทุติยภูมิ 7
2.7 การสะท้อนและดูดกลืนของแสงสีขาวเมื่อตกกระทบวัตถุที่มีสีปฐมภูมิภูมิ 7
2.8 การดูดกลืนของวัตถุสีดำจากการฉายแสงสีขาว 8
2.9 แผนภาพวงจรการเชื่อมต่อใช้งานตัวรับรู้ TCS 230 9
2.10 ตัวรับรู้ TCS230 10
2.11 การควบคุมตัวรับรู้ TCS230 10
2.12 แผงวงจร Arduino - ATmega2560 11
2.13 หน่วยประมวลผลกลางของ ATmega2560 ขนาด 100 ขา 13
2.14 จอแสดงผลแอลซีดี รุ่น Hitachi HD44780 14
2.15 องค์ประกอบหลักของมอเตอร์เซอร์โว 14
2.16 มอเตอร์เซอร์โวขนาดต่าง ๆ 15
2.17 โครงสร้างของมอเตอร์เซอร์โวแบบแอนะล็อก 16
2.18 โครงสร้างมอเตอร์เซอร์โวแบบดิจิตอล 16
2.19 แผงแป้นตัวเลข แบบ 4x4 17
2.20 โครงสร้างของแอลดีอาร์ 18
2.21 ความไวของแอลดีอาร์ที่ความยาวคลื่นต่างกัน 18
3.1 แผนภาพการทำงานของระบบคัดแยกวัตถุตามสี 19
3.2 แผนการเชื่อมต่อใช้งานไมโครคอนโทรลเลอร์กับอุปกรณ์ต่างๆ 20
3.3 แผนการเชื่อมต่อใช้งานตัวรับรู้สีกับไมโครคอนโทรลเลอร์ 21
3.4 แผนการเชื่อมต่อใช้งานแผงแป้นตัวเลขกับไมโครคอนโทรลเลอร์ 22
3.5 แผนภาพวงจรตรวจจับลูกปิงปอง 23
3.6 แผนการเชื่อมต่อใช้งานวงจรตรวจจับลูกปิงปอง 23
3.7 แผนภาพวงจรควบคุมค่าแรงดัน 5 V 24

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.8 แผนการเชื่อมต่อใช้งานแผงวงจรควบคุมค่าแรงดัน 5 V 24
3.9 แผนการเชื่อมต่อใช้งานหน้าจอแสดงผล 25
3.10 แผนการเชื่อมต่อใช้งานมอเตอร์เซอร์โว 26
3.11 โครงสร้างแบบจำลองของเครื่องคัดแยกวัตถุตามสี (มุมมองด้านหน้า) 27
3.12 ตำแหน่งการติดตั้งแอลดีอาร์และตัวรับรู้สีในแบบจำลอง 29
3.13 การติดตั้งกล่องควบคุมไว้ด้านข้างของโครงสร้างแบบจำลอง 29
3.14 แบบจำลองของเครื่องคัดแยกวัตถุตามสี (มุมมองด้านหน้า) 31
3.15 ตำแหน่งการติดตั้งตัวรับรู้สีและแอลดีอาร์ 31
3.16 ขั้นตอนการทำงานในแบบวิธีคัดแยกสี 33
3.17 ขั้นตอนการทำงานในแบบวิธีคัดเลือกสี 34
3.18 รูปแบบการแสดงผลขณะให้เลือกแบบวิธีทำงาน 35
3.19 รูปแบบแสดงผลการตรวจสีในแบบวิธีคัดแยก 36
3.20 รูปแบบการเลือกกล่อง A หรือกล่อง B 36
3.21 รูปแบบแสดงผลการป้อนจำนวนในแต่ละสีของกล่อง A 37
3.22 รูปแบบแสดงการสรุปจำนวนทั้งหมดเมื่อป้อนจำนวนของกล่อง A 37
3.23 รูปแบบแสดงผลการป้อนจำนวนในแต่ละสีของกล่อง B 38
3.24 รูปแบบแสดงการสรุปจำนวนทั้งหมดเมื่อป้อนจำนวนของกล่อง B 38
3.25 รูปแบบการแสดงผลเมื่อตรวจเจอลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง A 39
3.26 รูปแบบการแสดงผลเมื่อตรวจเจอลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง B 40
3.27 รูปแบบการแสดงผลเมื่อตรวจไม่พบลูกปิงปอง 40
3.28 รูปแบบการแสดงผลเมื่อส่งลูกปิงปองไปกล่อง C 41
4.1 หน้าจอแสดงแบบวิธีทำงานให้ผู้ใช้เลือก 42
4.2 การคัดแยกลูกปิงปองสีแดง 43
4.3 การคัดแยกลูกปิงปองสีเขียว 43
4.4 การคัดแยกลูกปิงปองสีน้ำเงิน 44
4.5 การแสดงผลบนหน้าจอเมื่อตรวจไม่พบลูกปิงปองในแบบวิธีคัดแยก 44
4.6 ตัวอย่างการเลือกลูกปิงปองของแต่ละสีในแต่ละกล่อง 44
4.7 การคัดเลือกลูกปิงปองสีแดงไปกล่อง A 45

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.8 การคัดเลือกลูกปิงปองสีน้ำเงินไปกล่อง A 46
4.9 การคัดเลือกลูกปิงปองสีแดงไปกล่อง B 46
4.10 การคัดเลือกลูกปิงปองสีเขียวไปกล่อง A 47
4.11 การคัดเลือกลูกปิงปองสีเขียวไปกล่อง B 47
4.12 การคัดเลือกลูกปิงปองสีแดงไปกล่อง C 48
4.13 การคัดเลือกลูกปิงปองสีน้ำเงินไปกล่อง B 49
4.14 การส่งลูกปิงปองสีเขียวไปกล่อง C หลังจากการคัดเลือกเสร็จสิ้น 49
4.15 การตรวจไม่พบลูกปิงปองหลังจากการคัดเลือกเสร็จสิ้น 50

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันภาคเกษตรกรรมเป็นปัจจัยหลักในการขับเคลื่อนเศรษฐกิจ ซึ่งผลผลิตทางเกษตรกรรมที่ได้จากการเก็บเกี่ยวของเกษตรกรมักมีทั้งผลผลิตที่เกษตรกรต้องการและผลผลิตที่เกษตรกรไม่ต้องการปะปนกันอยู่ ซึ่งสีเป็นคุณลักษณะทางกายภาพอย่างหนึ่งที่บ่งบอกความแตกต่างนี้ได้ ยกตัวอย่างเช่น การใช้สีเป็นตัวคัดแยกสิ่งแปลกปลอมโดยคัดเมล็ดข้าวโพดเสียที่มีสีดำออกจากกลุ่มเมล็ดข้าวโพดที่สมบูรณ์และมีสีเหลือง นอกจากนี้สียังสามารถแบ่งแยกผักผลไม้ที่สุกกับผักผลไม้ที่ยังดิบอยู่ได้ ยกตัวอย่างเช่นการคัดแยกมะเขือเทศที่ยังดิบอยู่ออกจากมะเขือเทศสุก ซึ่งมะเขือเทศสุกมีสีแดง ในขณะที่มะเขือเทศดิบมีสีเขียวจึงทำให้สามารถใช้สีเป็นตัวช่วยในการคัดแยกผลผลิตของเกษตรกรได้ นอกจากนี้สินค้าหรือผลิตภัณฑ์ในภาคอุตสาหกรรมแต่ละประเภทย่อมมีความเหมือนหรือแตกต่างกันในหลายๆด้าน เช่น เรื่องของสี ซึ่งผู้บริโภคสินค้าแต่ละคนมักชื่นชอบสีของสินค้าที่แตกต่างกันไป ยกตัวอย่างเช่น เสื้อผ้า เมื่อถูกตัดเย็บออกมามีขนาดเท่ากัน เนื้อผ้าเดียวกัน ลักษณะลวดลายเหมือนกัน แต่อาจมีความแตกต่างกันในเรื่องของสีของเสื้อผ้าแต่ละตัวเพื่อตอบสนองรสนิยมของผู้บริโภค ไม่เว้นแม้กระทั่งเครื่องเขียน เช่น ปากกายี่ห้อเดียวกันที่มีขนาดเท่ากัน แต่สิ่งสำคัญที่แตกต่างกันคือสีของปากกาที่ผู้ซื้อต้องการ รวมทั้งการใช้สีเป็นสิ่งจำแนกเพื่อบรรจุสินค้าและจำหน่ายต่อไป ดังนั้นในการคัดแยกนี้ก็สามารถใช้คุณสมบัติของสีมาประยุกต์ใช้กับกระบวนการคัดแยกนี้ได้เช่นกัน

ในโครงงานนี้ได้เล็งเห็นถึงความสำคัญดังกล่าวจึงประยุกต์ใช้กลไกในการคัดแยกโดยใช้คุณสมบัติของสีเป็นพื้นฐาน และประมวลผลด้วยไมโครคอนโทรลเลอร์ ระบบการคัดแยกวัตถุตามสีดังกล่าวจะช่วยให้ได้สีของผลิตภัณฑ์ตามที่ต้องการ และถ้าผู้ผลิตต้องการคัดแยกสีจากผลิตภัณฑ์ที่มีจำนวนมากก็สามารถทำได้เช่นกัน เพื่อช่วยทำให้ประหยัดเวลาในการคัดเลือก ประหยัดต้นทุนในการผลิต

1.2 วัตถุประสงค์ของโครงงาน

เพื่อสร้างแบบจำลองการคัดแยกโดยใช้สีของวัตถุเป็นเกณฑ์ภายใต้เงื่อนไขที่ถูกกำหนดโดยผู้ใช้งานและควบคุมด้วยไมโครคอนโทรลเลอร์ที่ทำงานร่วมกับตัวรับรู้ TCS230 โดยสั่งการผ่านทางหน้าจอแอลซีดี

1.3 ขอบเขตของโครงงาน

1) สร้างแบบจำลองการคัดแยกวัตถุตามสีโดยใช้วัตถุทดสอบเป็นลูกปิงปองสีแดง สีเขียว และสีน้ำเงิน โดยมีตัวรับรู้ TCS230 ในการตรวจสอบสีโดยมีแอลดีอาร์ที่สามารถตรวจจับลูกปิงปองโดยทุกขั้นตอนจะแสดงผ่านหน้าจอแอลซีดี
2) ในแบบวิธีคัดแยกเมื่อแอลดีอาร์ตรวจไม่พบลูกปิงปองสามารถแจ้งผ่านหน้าจอแอลซีดีพร้อมทั้งระบุจำนวนรวมของลูกปิงปองในแต่ละสี
3) ในแบบวิธีคัดเลือกผู้ใช้สามารถเลือกสีของลูกปิงปองของแต่ละกล่องได้ตามต้องการในกรณีที่ลูกปิงปองไม่ตรงกับเงื่อนไขหรือในกรณีที่ครบแล้วจะถูกแยกไปลงกล่องต่างหาก

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	พ.ศ. 2558					พ.ศ. 2559			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1) ศึกษาการทำงานของตัวรับรู้ TCS230									
2) ออกแบบโครงสร้างของแบบจำลองการคัดแยกวัตถุตามสี									
3) สร้างแบบจำลองโดยใช้ไมโครคอนโทรลเลอร์ในการควบคุม									
4) ทดสอบและปรับปรุงชิ้นงาน									
5) สรุปผลและจัดทำรูปเล่มปริญญานิพนธ์									

1.5 ประโยชน์ที่คาดว่าจะได้รับ

แบบจำลองการคัดแยกสีของลูกปิงปองนี้ถูกสร้างขึ้นมาให้สามารถนำไปประยุกต์ใช้ในการคัดแยกสินค้าเป็นกลุ่มตามสีของสินค้าหรือคละสีตามรูปแบบที่ผู้ผลิตสินค้าต้องการได้ แล้วสามารถใช้ตรวจสอบคุณภาพสีของสินค้าเพื่อให้ตรงกับมาตรฐานที่ผู้ผลิตกำหนด อีกทั้งยังช่วยลดการใช้แรงงานคนในการคัดแยกอีกด้วย

1.6 งบประมาณ

1) โครงสร้างของแบบจำลอง	1,080 บาท
2) ตัวรับรู้ TCS230	220 บาท
3) มอเตอร์เซอร์โว จำนวน 2 ตัว	300 บาท
4) หน้าจอแอลซีดีขนาด 20×4	200 บาท
5) แผงแป้นตัวเลขขนาด 4×4	40 บาท
6) ไมโครคอนโทรลเลอร์ AT mega 2560	560 บาท
7) ค่าถ่ายเอกสารและเข้าเล่มปริญญานิพนธ์	800 บาท
รวมเป็นเงินทั้งสิ้น (สามพันสองร้อยบาทถ้วน)	3,200 บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

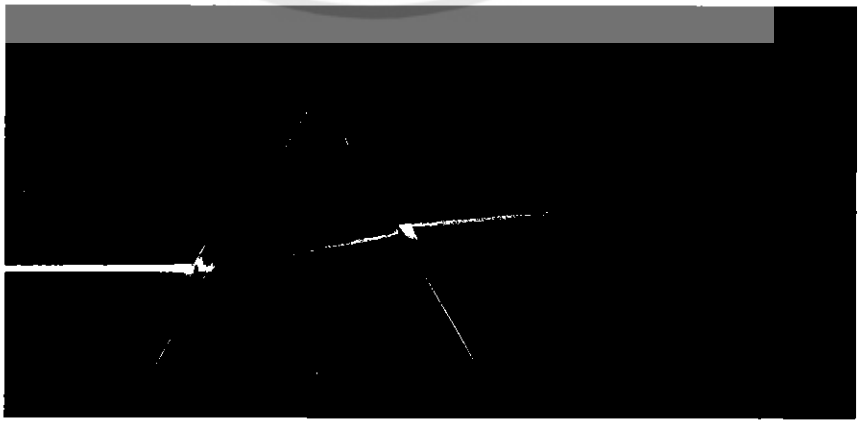
การคัดแยกวัตถุตามสีในโครงงานนี้อาศัยหลักการดูดกลืนและสะท้อนสีของแสงที่ตกกระทบวัตถุ โดยสีของแสงที่ตกกระทบลงบนวัตถุเป็นแสงสีขาวหรือแสงที่มองเห็นได้ด้วยตาเปล่า (Visible light)

2.1 คุณสมบัติของสีที่เกี่ยวข้องกับตัวรับรู้ทางสี

ตัวรับรู้เป็นอุปกรณ์อย่างหนึ่งที่ตอบสนองต่อปริมาณที่นำเข้าโดยการสร้างเอาต์พุตที่เกี่ยวข้องกับการทำงานที่มักจะอยู่ในรูปของสัญญาณไฟฟ้าหรือสัญญาณแสง ความไวของตัวรับรู้แสดงให้เห็นด้วยปริมาณการเปลี่ยนแปลงของเอาต์พุตที่จะเปลี่ยนแปลงเมื่อปริมาณที่กำลังทำการวัดเปลี่ยนแปลงไป โดยโครงงานนี้ได้ใช้ตัวรับรู้สีหมายเลข TCS230 ที่ใช้แสงขาวเป็นตัวคัดแยกสี

2.1.1 สเปกตรัมแสงขาว

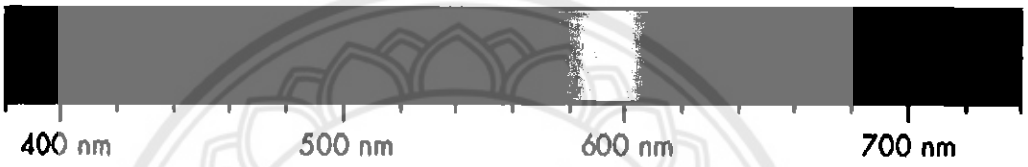
คือช่วงคลื่นแสงที่ทำให้ตาเราสามารถมองเห็นวัตถุเป็นสีต่างๆ ได้ซึ่งแสงสีขาวเป็นส่วนผสมของแสงสีต่างๆแต่ละแสงสีมีความถี่และความยาวคลื่นเฉพาะตัวสีเหล่านี้รวมตัวเป็นสเปกตรัมที่มองเห็นได้ ตาและสมองของเรารับรู้สิ่งต่างๆจากความแตกต่างของความยาวคลื่นของสีที่เรามองเห็นได้ เมื่อฉายแสงสีขาวผ่านก้อนปริซึมจะสามารถแยกได้เป็นแถบสีม่วง สีน้ำเงิน สีเขียว สีเหลือง สีส้ม และสีแดงเป็นแถบต่อเนื่องกันตามลำดับดังรูปที่ 2.1 ซึ่งมีช่วงความยาวคลื่นอยู่ระหว่าง 380-760 nm เท่านั้น และเป็นช่วงที่ตาเราสามารถรับความรู้สึกได้ แหล่งกำเนิดแสงที่ให้แสงขาว เช่นแสงอาทิตย์ แสงจากหลอดไฟเป็นต้น [2]



รูปที่ 2.1 ปรากฏการณ์ของก้อนปริซึม

ที่มา: www.dkfindout.com/uk/science/light/splitting-light

จะเห็นได้ว่า เมื่อเราฉายแสงสีขาวเข้าสู่ก้อนปริซึม แสงสีต่างๆซึ่งซ่อนอยู่ในแสงสีขาวนั้นจะถูกแยกตัวออกมาให้เห็นชัดเจนแสงสีต่างๆเหล่านี้จะมีความยาวคลื่นเฉพาะตัวต่างๆกันออกไป ซึ่งแสดงดังรูปที่ 2.2 จะพบว่าแสงสีม่วงมีความยาวคลื่นอยู่ระหว่าง (380-420 nm) สีครามมีความยาวคลื่นอยู่ระหว่าง (420-440 nm) สีน้ำเงินมีความยาวคลื่นอยู่ระหว่าง (440-490 nm) สีเขียวมีความยาวคลื่นอยู่ระหว่าง (490-560 nm) สีเหลืองมีความยาวคลื่นอยู่ระหว่าง (560-590 nm) สีส้มมีความยาวคลื่นอยู่ระหว่าง (590-630 nm) สีแดงมีความยาวคลื่นอยู่ระหว่าง (630-760 nm) เราอาจกล่าวได้ว่า สีของแสงเกิดจากความไม่สมดุลของแสงสีขาวนั่นเองกล่าวคือ สีของแสงใดที่มีพลังงานสูงที่สุดก็จะมีผลให้สีนั้นปรากฏออกมาเป็นสีดังกล่าวและแสงสีอื่นก็จะถูกดูดกลืนหายไปในวัตถุที่แสงขาวส่องมา

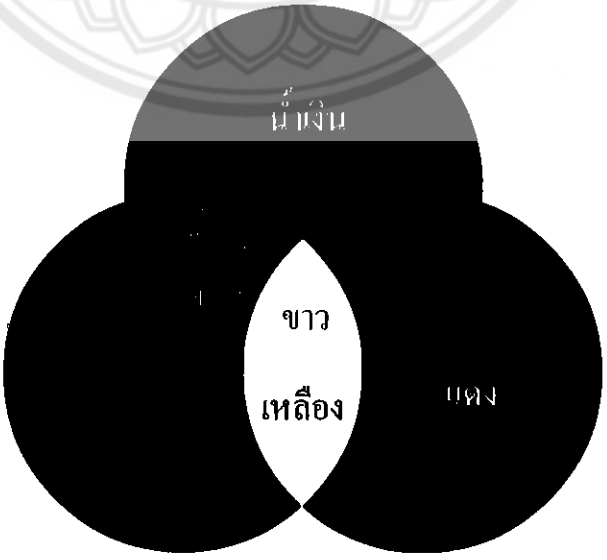


รูปที่ 2.2 สเปกตรัมของแสงสีขาว

ที่มา: www.topicstock.pantip.com/wahkor/topicstock

2.1.2 แม่สีของแสง

แม่สีของแสงหรือเรียกว่า สีปฐมภูมิ (Primary color) คือสีแดง สีเขียว และสีน้ำเงิน ซึ่งต่างจากแม่สีในวิชาทางศิลปะ แม่สีของแสงทั้งสามสีนี้เมื่อนำมาผสมกัน (Additive) โดยการฉายแสงซ้อนกันลงไป เราจะได้แสงชุดที่สองขึ้น เรียกว่า สีทุติยภูมิ (Secondary color) ดังรูปที่ 2.3

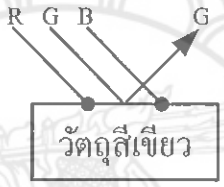


รูปที่ 2.3 การผสมของแสงสีปฐมภูมิ

จากรูปที่ 2.3 จะเห็นได้ว่าแสงสีเหลือง (Yellow) เกิดขึ้นจากการฉายแสงสีแดงลงบนแสงสีเขียว แสงสีม่วงอมแดง (Magenta) เกิดขึ้นจากการฉายแสงสีแดงลงบนแสงสีน้ำเงิน และแสงสีฟ้าอมเขียว (Cyan) ก็เกิดขึ้นจากการฉายแสงสีน้ำเงินลงบนสีเขียว และเมื่อฉายแสงสีชุดที่สองที่เกิดขึ้นนี้พร้อมๆกันลงไป ในอัตราส่วนที่เหมาะสมแล้วเราจะได้แสงออกมาเป็นสีขาว [2]

2.1.3 คุณสมบัติการดูดกลืนสีของวัตถุและสะท้อนสีของแสง

เมื่อแสงตกกระทบวัตถุแสงอาจถูกสะท้อนหรือถูกดูดกลืน พื้นผิวสีจะดูดกลืนทุกสีของแสงยกเว้นสีเดียวกับพื้นผิวนั้นซึ่งถูกสะท้อนดังรูปที่ 2.4 โดยกำหนดให้ R คือสีแดง G คือสีเขียว และ B คือสีน้ำเงิน ดังนั้นสีของวัตถุที่ถูกมองเห็นจะขึ้นอยู่กับแหล่งกำเนิดแสงที่ส่องวัตถุหรืออาจกล่าวได้ว่าสีนั้นต้องปรากฏทั้งในวัตถุและในแหล่งกำเนิดแสง นั่นคือ ถ้าไม่มีสีไม่ว่าจะในวัตถุหรือในแหล่งกำเนิดแสงสีนั้นจะไม่ถูกมองเห็น



รูปที่ 2.4 การตกกระทบและสะท้อนของแสงจากวัตถุ

สีที่เกิดขึ้นนี้จะเป็นปรากฏการณ์ที่เรียกว่า การเลือกดูดกลืน (Selective absorption) ซึ่งเป็นผลของวัตถุที่แสงมาตกกระทบ บางส่วนก็มีการดูดกลืนเข้าไปในวัตถุ บางส่วนก็มีการสะท้อนกลับออกไป หรือบางส่วนอาจกระจายไป ซึ่งจะขึ้นอยู่กลับคุณสมบัติของวัสดุที่เป็นวัตถุนั่นเอง [3]

เมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีขาวที่เกิดจากการรวมกันของสีแดง สีเขียว สีน้ำเงิน จะทำให้เกิดการสะท้อนแสงสีขาวออกมานั่นคือสะท้อนกลับหมดทั้งสามสีตามรูปที่ 2.5

รูปที่ 2.5 การสะท้อนกลับหมดของแสงสีขาวที่ส่องลงบนวัตถุสีขาว [1]

เมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีฟ้าอมเขียว ที่เกิดจากการรวมกันของสีเขียวและสีน้ำเงินจะทำให้วัตถุสะท้อนแสงสีเขียวและแสงสีน้ำเงินแล้วดูดกลืนแสงสีแดง ทำให้สามารถเห็นวัตถุเป็นสีฟ้าอมเขียว

เมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีม่วงอมแดงที่เกิดจากการรวมกันของแดงและสีน้ำเงิน จะทำให้วัตถุสะท้อนแสงสีแดงและแสงสีน้ำเงินแล้วดูดกลืนแสงสีเขียว ทำให้สามารถเห็นวัตถุเป็นสีม่วงอมแดง

เมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีเหลืองที่เกิดจากการรวมกันของสีแดงและสีเขียว จะทำให้วัตถุสะท้อนแสงสีแดงและแสงสีเขียวแล้วดูดกลืนแสงสีน้ำเงิน ทำให้สามารถเห็นวัตถุเป็นสีเหลือง

จากการอธิบายข้างต้นพบว่าสีทุติยภูมิของแสงนั้นจะสะท้อนสีปฐมภูมิที่รวมกันแล้วเกิดเป็นสีนั้นและดูดกลืนสีปฐมภูมิอีกสี ทำให้สามารถมองเห็นสีที่เกิดจากการสะท้อนของสีปฐมภูมิทั้งสองรวมกันตามรูปที่ 2.6

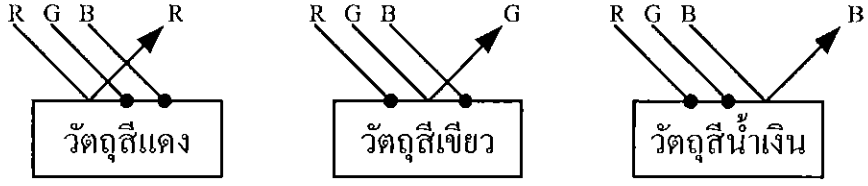
รูปที่ 2.6 การสะท้อนและดูดกลืนของแสงสีขาวเมื่อตกกระทบวัตถุที่มีสีทุติยภูมิ [1]

เมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีแดง จะทำให้วัตถุสะท้อนแสงสีแดงจากนั้นดูดกลืนแสงสีเขียวและแสงสีน้ำเงิน ทำให้สามารถเห็นวัตถุเป็นสีแดงได้

เมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีเขียว จะทำให้วัตถุสะท้อนแสงสีเขียวจากนั้นดูดกลืนแสงสีแดงและแสงสีน้ำเงิน ทำให้สามารถเห็นวัตถุเป็นสีเขียวได้

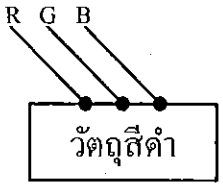
เมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีน้ำเงิน จะทำให้วัตถุสะท้อนแสงสีน้ำเงินจากนั้นดูดกลืนแสงสีแดงและแสงสีเขียว ทำให้สามารถเห็นวัตถุเป็นสีน้ำเงินได้

จากการอธิบายข้างต้นพบว่าเมื่อฉายแสงสีขาวลงบนวัตถุที่มีสีปฐมภูมิ จะทำให้วัตถุสะท้อนสีของแสงที่บ่งบอกถึงสีของวัตถุนั้นออกมาเพียงหนึ่งเดียว และจะดูดกลืนสีของแสงที่เหลือตามรูปที่ 2.7



รูปที่ 2.7 การสะท้อนและดูดกลืนของแสงสีขาวเมื่อตกกระทบวัตถุที่มีสีปฐมภูมิ [1]

เมื่อฉายแสงสีขาวลงบนวัตถุที่มีดำ วัตถุจะไม่เกิดการสะท้อนแสงสีขาวเลย แต่จะเกิดการดูดกลืนแสงสีขาวทั้งหมด [1] ตามรูปที่ 2.8



รูปที่ 2.8 การดูดกลืนของวัตถุสีดำจากการฉายแสงสีขาว [1]

2.2 หลักการทำงานของตัวรับรู้หมายเลข TCS230

ตัวรับรู้ TCS230 มีโฟโตไดโอด (Photodiode) ขนาด 8×8 ตัวโดยมีตัวรับรู้สีน้ำเงิน สีเขียว และสีแดง สีละ 16 ตัว และอีก 16 ตัวสำหรับแสงสีขาว โดยจะให้สัญญาณเอาต์พุตออกมาทางขา S2 และ S3 เช่นถ้า S2 และ S3 ให้สัญญาณต่ำออกมาแสดงว่าโฟโตไดโอดอ่านได้ค่าแสงสีแดง แต่ถ้า S2 และ S3 ให้ค่าออกมาเป็นสัญญาณต่ำและสัญญาณสูงตามลำดับแสดงว่า โฟโตไดโอดอ่านได้ค่าแสงสีน้ำเงินตามตารางที่ 2.1

ตารางที่ 2.1 แสดงสัญญาณเอาท์พุต

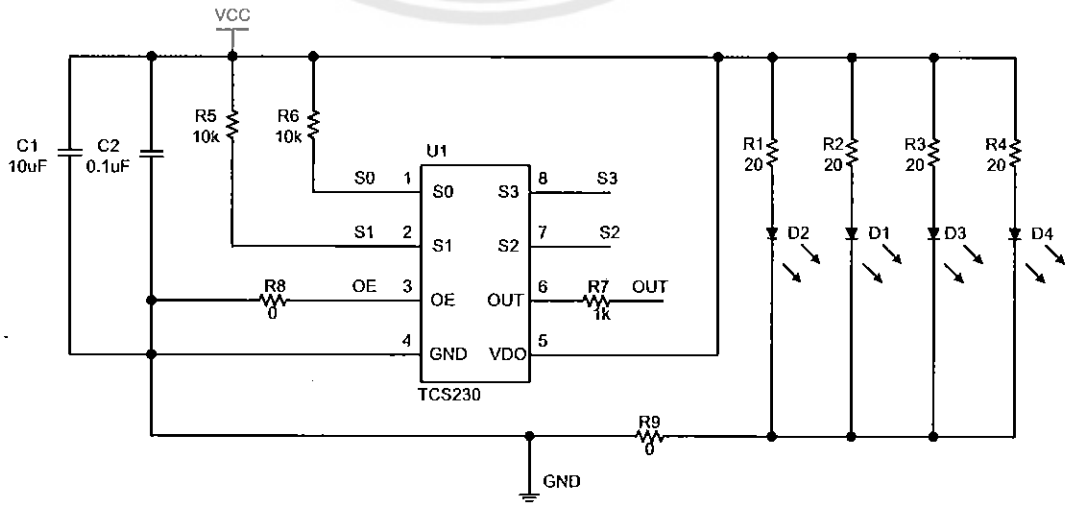
S2	S3	ชนิดของโฟโตไดโอด
สัญญาณต่ำ	สัญญาณต่ำ	แสงสีแดง
สัญญาณต่ำ	สัญญาณสูง	แสงสีน้ำเงิน
สัญญาณสูง	สัญญาณต่ำ	ไม่แสดงผล
สัญญาณสูง	สัญญาณสูง	แสงสีเขียว

การควบคุมขยายสัญญาณที่ขา S0 และที่ขา S1 สามารถทำได้โดยมี 4 ระดับดังต่อไปนี้ คือ ถ้าขา S0 กับขา S1 ให้ค่าออกมาเป็นสัญญาณต่ำทั้งคู่ หรือ (00) แสดงว่าปิดการทำงาน ถ้าขา S0 กับขา S1 ให้ค่าออกมาเป็นสัญญาณสูงและสัญญาณต่ำ หรือ (10) ตามลำดับ แสดงว่า ขยายสัญญาณที่ 20%ตามตารางที่ 2.2

ตารางที่ 2.2 แสดงการขยายสัญญาณ

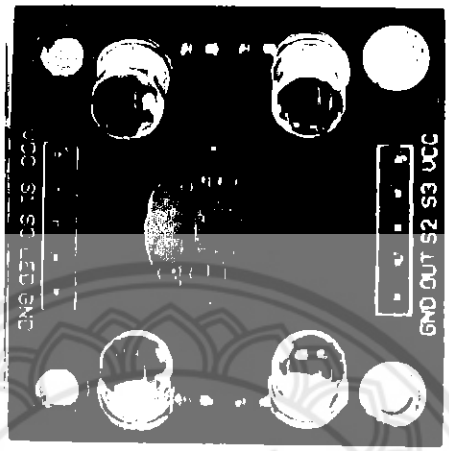
S0	S1	ค่าความถี่เอาต์พุต
สัญญาณต่ำ	สัญญาณต่ำ	ปิดการทำงาน
สัญญาณต่ำ	สัญญาณสูง	ขยายสัญญาณที่ 2%
สัญญาณสูง	สัญญาณต่ำ	ขยายสัญญาณที่ 20%
สัญญาณสูง	สัญญาณสูง	ขยายสัญญาณที่ 100%

จากหลักการของแม่สีของแสงสว่างเบื้องต้น ถ้าสามารถรู้ความถี่ของแสง สีแดง สีเขียว และสีน้ำเงินได้สามารถนำค่าแต่ละค่ามารวมกัน จึงจะได้ออกมาเป็นค่าสีของแสงในแต่ละสีที่บ่งบอกว่าคือสีอะไร และตัวรับรู้สี TCS230 นี้มีตัวรับรู้แสงเหล่านี้มาให้ทั้งหมดแล้วทำให้สามารถแยกประเภทสีได้ ในการประยุกต์ใช้งานตัวรับรู้สี TCS230 จะทำให้ Arduino สามารถมองเห็นสีต่างๆ ของสิ่งรอบตัวเพื่อเอาไปประมวลผลตามที่ต้องการ เช่น ทำหน้าที่เป็นตัวอ่านค่าแม่สีของแสงสว่าง หรือนำค่าแม่สีของแสงสว่างไปแสดงลงบนหน้าจอคอมพิวเตอร์ ตัวรับรู้สีนี้จะช่วยเพิ่มความสามารถให้กับ Arduino ตัวรับรู้สี TCS230 ใช้ไฟเลี้ยง 3.3-5 V ใช้สายสัญญาณ 3 เส้นมีสายสำหรับควบคุมไฟ LED อีก 1 เส้น สามารถควบคุมให้เปิดไฟตอนกำลังอ่านค่าสี และควบคุมให้ปิดไฟเมื่ออ่านค่าสีเสร็จแล้ว ตัวรับรู้สี TCS230 ตัวนี้ใช้แยกความถี่ของแสง โดยโฟโตไดโอด ซึ่งมีหลักการดังนี้ เมื่อแสงตกกระทบลงบนตัวโฟโตไดโอดจะทำให้ศักย์ไฟฟ้าที่คร่อมตัวโฟโตไดโอดเปลี่ยนแปลง แต่การเปลี่ยนแปลงนี้มีค่าน้อยมาก (ระดับ mV) การแปลงแสงเป็นความถี่อ่านได้จากโฟโตไดโอดที่อยู่ด้านในไมโครชิพของตัวรับรู้ ซึ่งมีจำนวนทั้งหมด 8×8 ตัวสามารถอ่านค่า สีแดง สีเขียว และสีน้ำเงิน จากแหล่งกำเนิดแสงโดยตรง ซึ่งค่าที่อ่านมาได้จะเป็นค่าความถี่ของแสง แต่ยังไม่ใช่ค่าแม่สีของแสงสว่างที่ใช้ในคอมพิวเตอร์ทั่วไป



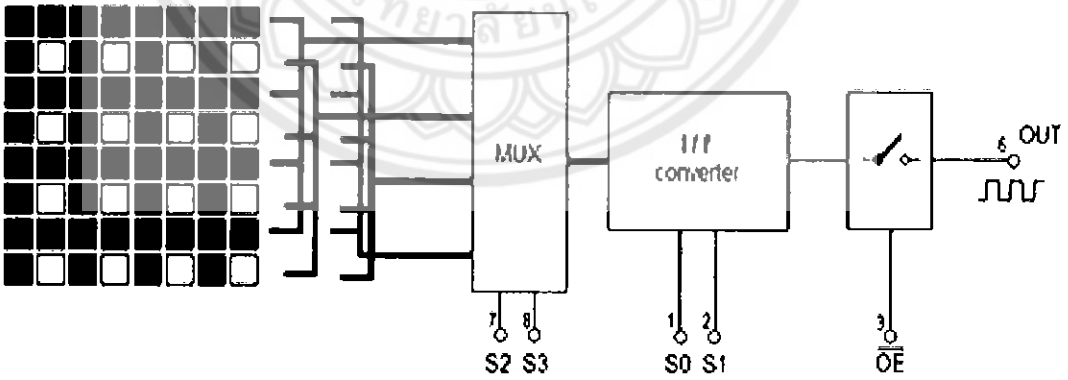
รูปที่ 2.9 แผนภาพวงจรการเชื่อมต่อใช้งานตัวรับรู้ TCS 230

เนื่องจากในการวัดค่าสีนั้น เซนเซอร์จะตรวจสอบค่าสีได้แม่นยำมากขึ้นเมื่อมีแสงสีขาวตรวจจับสีในที่มืด ทำให้เซนเซอร์ตัวนี้มีพื้นหลังเป็นสีขาวและมีกระบอกพลาสติกสีดำครอบ เพื่อไม่ให้สีอื่นไปรบกวนการตรวจสอบสีของเซนเซอร์และจะทำให้มีความแม่นยำในการตรวจสอบสีมากขึ้นอีกด้วยดังรูปที่ 2.10



รูปที่ 2.10 ตัวรับรู้ TCS230

การควบคุมตัวรับรู้ TCS230 นี้ทำได้โดยควบคุมจากขา S2, S3 และขาเอาต์พุต จะแสดงสัญญาณออกมาเป็นสัญญาณเวฟสี่เหลี่ยม (50% duty cycle) เป็นความถี่ที่อ่านได้จากตัวรับรู้แสงโฟโตไดโอดโดยตรงสามารถขยายความเข้มของค่าที่อ่านได้โดยควบคุมอัตราขยายที่ขา S0 และ S1 ตามรูปที่ 2.11

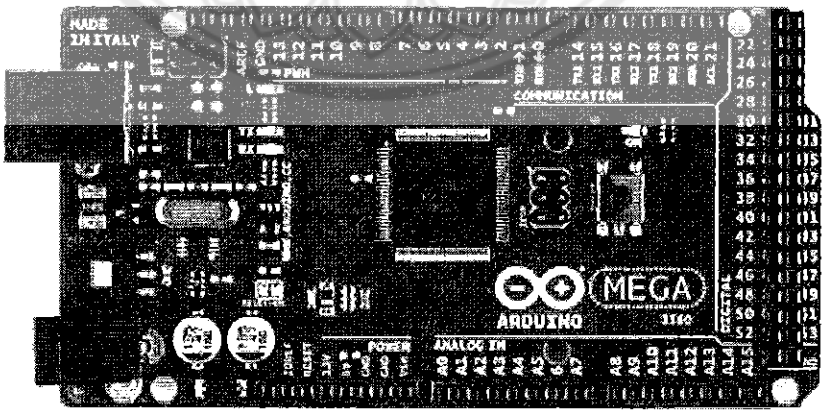


รูปที่ 2.11 การควบคุมตัวรับรู้ TCS230

ที่มา: www.arduinoall.com

2.3 แผงวงจร Arduino

ไมโครคอนโทรลเลอร์ที่ใช้ในโครงงานนี้เป็นแผงวงจร Arduino จัดอยู่ในตระกูล AVR ขนาด 100 ขา ซึ่งใช้ไมโครคอนโทรลเลอร์หมายเลข ATmega2560 แสดงดังรูปที่2.12 เป็นแผงวงจร Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้อินพุตและเอาท์พุตมากกว่า Arduino รุ่นอื่น ๆ เช่น งานที่ต้องรับสัญญาณจากตัวรับรู้ หรือควบคุมมอเตอร์เซอร์โวหลาย ๆ ตัว โดยไมโครคอนโทรลเลอร์ Arduino เป็นแพลตฟอร์ม (Platform) ของอินพุตและเอาท์พุต (I/O) ที่เพียงพอกับการใช้งานและการเรียนรู้ และมีการพัฒนาแบบ Open source คือมีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ แผงวงจรถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นใช้งาน ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวแผงวงจร หรือโปรแกรมต่อได้ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกเข้ามาที่ขา I/O ของแผงวงจร หรือเพื่อความสะดวกสามารถเลือกต่อกับแผงวงจรเสริม (Shield) ประเภทต่าง ๆ เช่น X Bee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาเสียบกับแผงวงจร Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้โดยตัวแผงวงจรมีคำสั่งที่ใช้ควบคุมพอร์ต อินพุตและเอาท์พุต ไม่ว่าจะเป็นพอร์ตดิจิตอล พอร์ตแอนะล็อกพีดับเบิลยูเอ็มและพอร์ตอนุกรมซึ่งแผงวงจร Arduino ทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าการใช้เครื่องคอมพิวเตอร์ ตัวแผงวงจรออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยวและมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้แผงวงจร Arduino สามารถรับสัญญาณจากสวิตช์หรือตัวรับรู้และควบคุมหลอดไฟมอเตอร์หรืออุปกรณ์อื่นๆแผงวงจร Arduino สามารถทำงานอิสระหรือทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ [1]



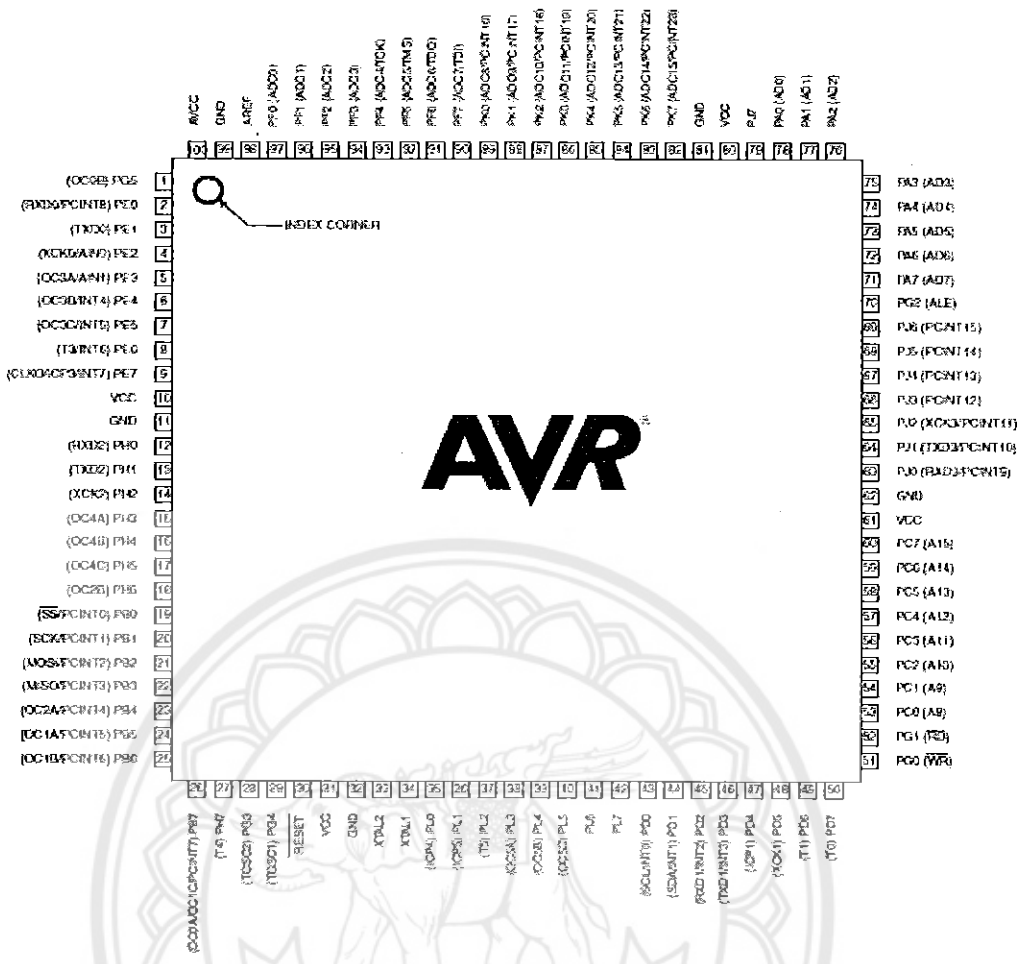
รูปที่ 2.12 แผงวงจร Arduino - ATmega2560

ที่มา: www.arduitronics.com

แผงวงจร Arduino ซึ่งมีจุดเด่นในเรื่องของความง่ายต่อการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่าง ๆ เพื่อสนับสนุนการใช้งานด้วยรูปแบบที่ไม่ซับซ้อนคือเน้นการโปรแกรมไมโครคอนโทรลเลอร์เป็นหลักแผงวงจร Arduino เป็นไมโครคอนโทรลเลอร์โดยใช้ AVR ขนาดเล็กซึ่งเป็นตัวประมวลผลและสั่งงานเหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์ และนำไปประยุกต์เพื่อใช้งานเกี่ยวกับการควบคุมอุปกรณ์ อินพุตและเอาท์พุตต่าง ๆ ได้มากมายทั้งในแบบที่เป็นการทำงานตัวเดียวอิสระ หรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่นเช่น คอมพิวเตอร์ทั้งนี้เนื่องมาจากว่า Arduino สนับสนุนการเชื่อมต่อกับอุปกรณ์อินพุตและเอาท์พุตได้มากมาย ทั้งแบบดิจิตอล (Digital) และแอนะล็อก (Analog) เช่น การรับค่าจากสวิตช์หรือตัวรับรู้แบบต่าง ๆ รวมไปถึงการควบคุมอุปกรณ์เอาท์พุตต่าง ๆ ส่วนภาษาในการเขียนโปรแกรมลงบนแผงวงจร Arduino นั้นจะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ทีมโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) แต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมและยังสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C โดยตรง

ตัวแผงวงจร Arduino ที่ใช้ในโครงงานนี้จะกล่าวถึงสถาปัตยกรรมของเอวีอาร์ (AVR) ขนาด 8 bits โดยเป็นหน่วยประมวลผลกลาง (Central processing unit, CPU) แบบ RISC (Reduced instruction set computer) มีสถาปัตยกรรมการต่อหน่วยความจำแบบฮาวาร์ด (Harvard) ซึ่งแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกันโดยเด็ดขาดดังแสดงในรูปที่ 2.13 โดยใช้หน่วยความจำแบบแฟลช (Flash) สำหรับเป็นหน่วยความจำโปรแกรม ซึ่งมีความจุมากกว่ารุ่น Arduino Uno R3 ทำให้สามารถเขียนโปรแกรมเข้าไปได้มากกว่าในความเร็วของ MCU ที่เท่ากัน และใช้หน่วยความจำแบบ SRAM สำหรับหน่วยความจำข้อมูลและนอกจากนี้ยังมีหน่วยความจำแบบ EEPROM ซึ่งสามารถเก็บข้อมูลเอาไว้ได้โดยไม่จำเป็นต้องมีไฟเลี้ยงอีกด้วย
จากรูปที่ 2.13 แผงวงจรไมโครคอนโทรลเลอร์ ATmega2560 มีคุณสมบัติเด่น [1] ดังนี้

1) ทำงานได้ตั้งแต่ย่านแรงดัน 1.8-5.5 V แรงดันของระบบอยู่ที่ 5 V
2) หน่วยความจำข้อมูลแบบ SRAM ขนาด 8 kb
3) หน่วยความจำข้อมูลแบบ EEPROM ขนาด 4 kb
4) สนับสนุนการเชื่อมต่อแบบ I2C bus
5) พอร์ตอินพุตเอาท์พุตแบบดิจิตอลจำนวน 54 ช่อง
6) พอร์ตเอาท์พุตแบบอนาล็อกจำนวน 16 ช่อง
7) วงจรสื่อสารอนุกรม
8) สนับสนุนช่องสัญญาณสำหรับสร้าง สัญญาณพีดับเบิลยูเอ็ม (PWM) จำนวน 14 ช่อง



รูปที่ 2.13 หน่วยประมวลผลกลางของ ATmega2560 ขนาด 100 ขา

ที่มา: www.atmel.com

2.4 จอแสดงผลแอลซีดี

ในการควบคุมหรือสั่งงานตัวจอแสดงผลแอลซีดี (Liquid crystal display, LCD) นั้นมีส่วนควบคุม (Controller) รวมไว้ในตัวอยู่แล้วสามารถส่งรหัสคำสั่งควบคุมการทำงานของจอแสดงแอลซีดีผ่านไมโครคอนโทรลเลอร์ ว่าต้องการใช้แสดงผลอย่างไร โดยในส่วนควบคุมของจอแสดงผลแอลซีดีของจอตัวนี้เป็น รุ่น Hitachi HD44780 ดังแสดงในรูปที่ 2.14 และขาในการเชื่อมต่อระหว่างจอแสดงผลแอลซีดีกับไมโครคอนโทรลเลอร์มีดังนี้ [5]

1) GND เป็นกราวด์ใช้ต่อระหว่างกราวด์ของระบบไมโครคอนโทรลเลอร์กับแอลจอซีดี
2) VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับจอแอลซีดีขนาด +5 V
3) VO ใช้ปรับความสว่างของหน้าจอ LCD4 RS ใช้บอกให้ตัวควบคุมจอแสดงผลทราบว่ารหัสคำสั่งที่ส่งมาทางขาข้อมูลเป็นคำสั่งหรือข้อมูล
4) RS ใช้บอกส่วนควบคุมจอแสดงผลแอลซีดีทราบว่ารหัสที่ส่งมาทางขาข้อมูลเป็นคำสั่งหรือข้อมูล

5) R และ W ใช้กำหนดว่าจะอ่านหรือเขียนข้อมูลกับส่วนควบคุมจอแสดงผลแอลซีดี

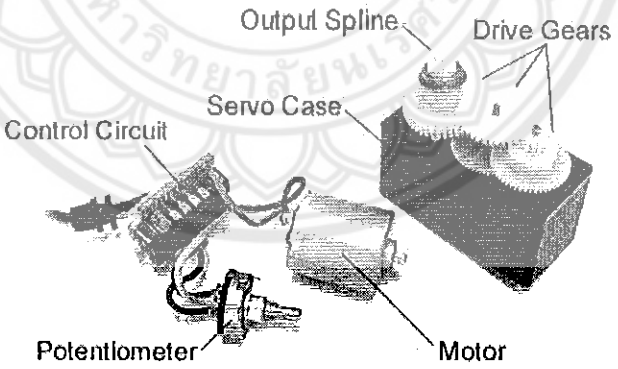
6) E เป็นขา Enable หรือ Chips select เพื่อกำหนดการทำงานให้กับส่วนควบคุมจอแสดงผลแอลซีดี

7) DB0 - DB7 เป็นขาสัญญาณในการเขียนหรืออ่านข้อมูลเพื่อควบคุมจอแสดงผล

รูปที่ 2.14 จอแสดงผลแอลซีดี รุ่น Hitachi HD44780 [5]

2.5 ส่วนประกอบของมอเตอร์เซอร์โว

มอเตอร์เซอร์โว (Servo motor) มีส่วนประกอบต่าง ๆ แสดงให้เห็นแสดงในรูปที่ 2.15



รูปที่ 2.15 องค์ประกอบหลักของมอเตอร์เซอร์โว [6]

องค์ประกอบหลักของมอเตอร์เซอร์โวโดยทั่วไปแล้วจะมีส่วนประกอบหลักดังนี้คือ

1) โครง (Servo case) ซึ่งส่วนใหญ่จะทำมาจากพลาสติก

2) มอเตอร์ (Motor) ซึ่งเป็นส่วนให้กำลังในการหมุนของมอเตอร์เซอร์โว

3) วงจรควบคุม (Control circuit) มีหน้าที่ในการถอดรหัสสัญญาณควบคุมที่รับค่ามาเป็นแบบพีดับเบิลยูเอ็ม และส่งไปควบคุมการทำงานของมอเตอร์ให้หมุนไปอยู่ในตำแหน่งที่ได้ถอดรหัสมา

4) เฟืองขับ (Drive gear) คือ ชุดทดรอบจากการหมุนของมอเตอร์เพื่อให้ได้แรงบิดที่สูง

5) สลัก (Output spline) คือ ส่วนที่ป้องกันการเสียดสีระหว่างโครงและเพลา (Shaft) ซึ่งอาจใช้อุปกรณ์ประเภทลูกปืน (Bearing) เพื่อช่วยลดแรงเสียดทานที่ดี

6) สายไฟและสายสัญญาณ (Servo wire) มีสามเส้นติดเป็นชุดเดียวกัน มีหน้าที่ดังนี้

เส้นที่ 1 จ่ายไฟกระแสตรง ซึ่งแรงดันปกติมีค่า 5-6 V

เส้นที่ 2 เป็นสายกราวด์

เส้นที่ 3 รับสัญญาณพัลส์ควบคุมจากไมโครคอนโทรลเลอร์

2.5.1 ขนาดของมอเตอร์เซอร์โว

มอเตอร์เซอร์โวมีขนาดต่าง ๆ แสดงในรูปที่ 2.16 โดยปกติขนาดของมอเตอร์เซอร์โวจะมีอยู่ด้วยกัน 3 ขนาด คือ ขนาดเล็ก (Micro) ขนาดมาตรฐาน (Standard) และขนาดใหญ่ (Giant) ซึ่งมีการใช้งานที่แตกต่างกันไปแต่ในปัจจุบันก็ได้มีมอเตอร์เซอร์โวที่มีขนาดที่หลากหลายเพื่อครอบคลุมการใช้งานมากขึ้น

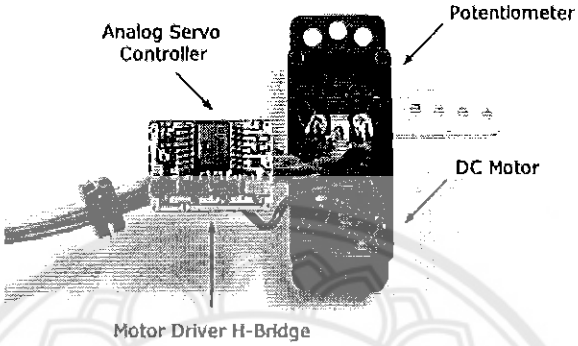
รูปที่ 2.16 มอเตอร์เซอร์โวขนาดต่าง ๆ [4]

นอกเหนือจากขนาดภายนอกของมอเตอร์เซอร์โวที่เราต้องพิจารณาในการใช้งานแล้วยังมีคุณลักษณะที่ต้องพิจารณาอีก คือ ความเร็ว (Speed) และแรงบิด (Torque) การวัดความเร็วของมอเตอร์เซอร์โวเทียบจากเวลาที่ใช้ต่อองศาในการหมุนค่าหนึ่ง ซึ่งมุมมาตรฐานที่ใช้วัดกันทั่วไปคือ 60° จึงกล่าวได้ว่า ความเร็วของมอเตอร์เซอร์โวขึ้นอยู่กับเวลาที่ใช้ในการหมุนแขนของมอเตอร์เซอร์โวไปจากตำแหน่งเดิมเป็นมุม 60° ดังนั้น ตัวเลขเวลาที่มีค่าน้อยเท่าใดมอเตอร์เซอร์โวยิ่งมีความเร็วมากขึ้นเท่านั้น

2.5.2 ประเภทของมอเตอร์เซอร์โว

มอเตอร์เซอร์โวสามารถแบ่งออกได้เป็น 2 ประเภทคือ แบบดิจิตอล (Digital servo) และแบบแอนะล็อก (Analog servo)

1) มอเตอร์เซอร์โวแบบแอนะล็อก มีลักษณะและส่วนประกอบดังในรูปที่ 2.17



รูปที่ 2.17 โครงสร้างของมอเตอร์เซอร์โวแบบแอนะล็อก [6]

ความเร็วและแรงบิดของมอเตอร์แบบแอนะล็อกถูกกำหนดด้วยความกว้างของพัลส์แรงดันขนาด 4.8-6.0 V ความถี่ 50 Hz ซึ่งควบคุมด้วยไมโครคอนโทรลเลอร์ ซึ่งการปรับความกว้างของพัลส์จะเกิดขึ้นทุก ๆ 20 ms ซึ่งอาจไม่สามารถตอบสนองต่อสัญญาณได้เร็วพอหรือสร้างแรงบิดได้ดีพอเมื่อมีการเปลี่ยนแปลงสัญญาณควบคุมเพียงเล็กน้อยหรือเมื่อมีแรงจากภายนอกมากระทำให้ตำแหน่งของมอเตอร์เซอร์โวเลื่อนไปจากตำแหน่งที่ตั้งไว้ อย่างไรก็ตามความเร็วในการตอบสนองดังกล่าวไม่เป็นปัญหาต่อการควบคุม

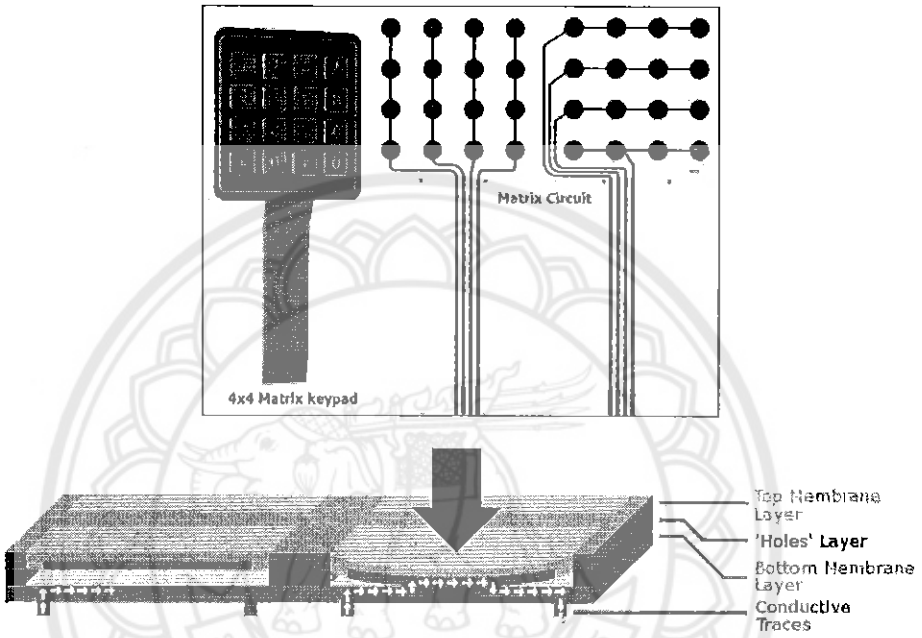
2) มอเตอร์เซอร์โวแบบดิจิตอล (Digital RC servo operation)

ลักษณะและส่วนประกอบของมอเตอร์เซอร์โวแบบดิจิตอลแสดงดังรูปที่ 2.18 พัลส์แรงดันไฟฟ้าที่ป้อนให้กับมอเตอร์เซอร์โวแบบดิจิตอลมีความถี่ในการปรับสัญญาณสูงถึง 300 Hz ทำให้การตอบสนองด้านความเร็วของมอเตอร์ดีขึ้น มีแรงบิดคงที่ และมีอัตราเร่งที่ราบเรียบขึ้น [6]

รูปที่ 2.18 โครงสร้างมอเตอร์เซอร์โวแบบดิจิตอล [6]

2.6 แผงแป้นตัวเลข

แผงแป้นตัวเลข แบบ 4x4 ชนิดนี้ประกอบไปด้วยปุ่ม 16 ปุ่ม ที่เรียงต่อกันเป็นเมตริกซ์แบบ 4 แถว และ 4 หลัก ประกอบไปด้วย เลข 0 – 9 และตัวอักษร A – D และ * และ # ซึ่งปุ่มแต่ละปุ่ม เป็นการกดเพื่อให้หน้าสัมผัสที่เป็นชั้นสีแดงดังรูปที่ 2.19 ไปแตะกันทำให้เป็นการเชื่อมต่อกันทางไฟฟ้าไปอีกด้านหนึ่งของสวิตช์

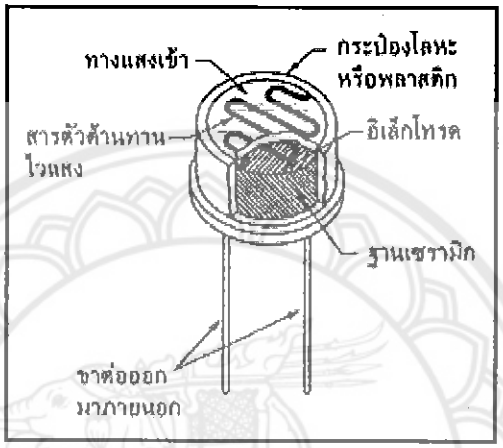


รูปที่ 2.19 แผงแป้นตัวเลข แบบ 4x4 [7]

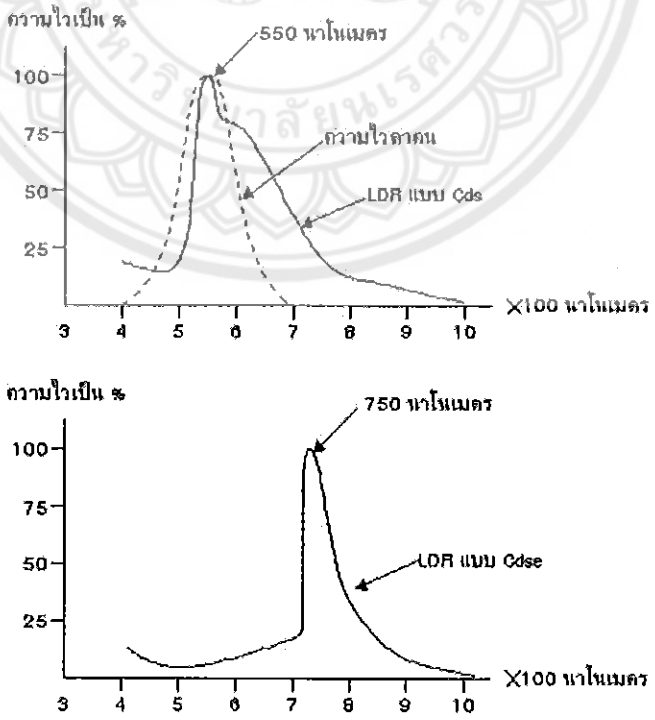
ในการตรวจสอบแป้นตัวเลขการกดปุ่มของผู้ใช้ในขณะนั้น ใช้วิธีการตรวจสอบไปทีละหลัก จนครบทุกหลัก แล้วนำมาตีความว่ามีการตอบสนองออกมาเป็นแบบใดบ้าง เช่น ถ้ามีการกดเลข 1 อยู่ในขณะที่เราจ่ายแรงดัน 5 V ไปที่หลักที่ 1 จะมีเพียงแถวแรกเท่านั้นที่จะอ่านค่าแรงดันได้สูง นอกนั้นจะเป็นแรงดันต่ำ หรือถ้ามีการกดปุ่ม # อยู่ ขณะที่ตรวจสอบไปแต่ละหลักนั้นจะไม่เจอแรงดันสูงที่แถวใดเลยจนกว่าจะตรวจสอบไปถึงหลักที่ 3 ซึ่งจะพบว่ามีการตอบสนองกลับมาจากแถวที่ 4 นั้นเอง ดังนั้นเมื่อพบว่าเป็นการตรวจสอบหลักที่ 3 และมีแถวที่ 4 ตอบสนอง ก็คือปุ่ม # นั่นเอง [7]

2.7 แอลดีอาร์

แอลดีอาร์ (Light dependent resistor, LDR) คือตัวความต้านทานที่สามารถเปลี่ยนสภาพความนำไฟฟ้าได้เมื่อมีแสงมาตกกระทบ โดยสร้างมาจากสารกึ่งตัวนำประเภทแคดเมียมซัลไฟด์ (Cadmium sulfide, CdS) หรือแคดเมียมซิลิไนด์ (Cadmium selenide, CdSe) เอามาฉาบลงบนแผ่นเซรามิกที่ใช้เป็นฐานรองแล้วต่อขาจากสารที่ฉาบไว้ออกมาดังรูปที่ 2.20 แอลดีอาร์ไวต่อแสงช่วงคลื่น 400-1000 nm [8] ดังรูปที่ 2.21



รูปที่ 2.20 โครงสร้างของแอลดีอาร์ [8]



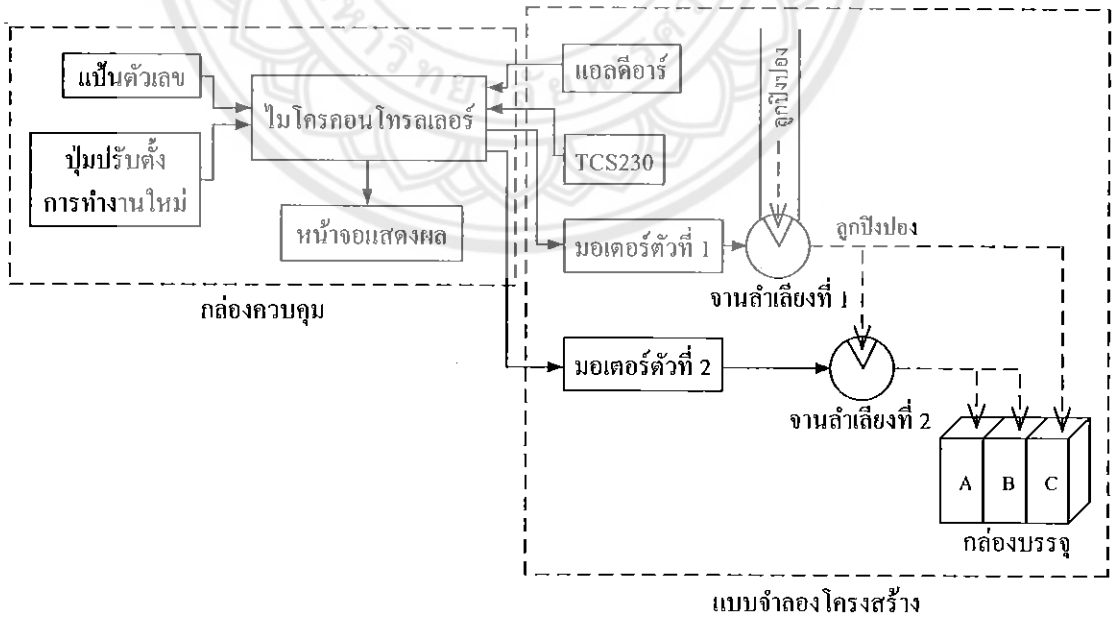
รูปที่ 2.21 ความไวของแอลดีอาร์ที่ความยาวคลื่นต่างกัน [8]

บทที่ 3
การออกแบบและสร้างแบบจำลองการคัดแยกวัตถุตามสี

ในส่วนของบทที่ 3 นี้เป็นการออกแบบการคัดแยกวัตถุตามสีในแต่ละแบบวิธี และสร้างแบบจำลองโดยติดตั้งของอุปกรณ์ทั้งในส่วนของโครงสร้างและในส่วนของการควบคุม

3.1 ส่วนประกอบหลักการทำงานของระบบการคัดแยกวัตถุตามสี

แผนภาพการทำงานของระบบคัดแยกวัตถุตามสีแสดงดังรูปที่ 3.1 โดยเริ่มต้นผู้ใช้ต้องเลือกแบบวิธีทำงานและปรับตั้งค่าต่างๆตามที่ต้องการ โดยผ่านทางแป้นตัวเลข จากนั้นไมโครคอนโทรลเลอร์ทำการประมวลผลซึ่งแสดงผ่านหน้าจอแสดงผล ระบบเริ่มตรวจสอบว่ามีลูกปิงปองหรือไม่โดยใช้แอลดีอาร์ จากนั้นจึงทำการตรวจจับสีของลูกปิงปองโดยใช้ตัวรับรู้สี TCS230 ส่งค่าให้ไมโครคอนโทรลเลอร์ประมวลผลว่าเป็นสีอะไร แล้วจึงสั่งให้มอเตอร์เซอร์โวตัวที่ 1 หมุนจานลำเลียงที่ 1 ส่งลูกปิงปองไปยังจานลำเลียงที่ 2 หรือกล่อง C (ขึ้นอยู่กับสีของลูกปิงปอง) ถ้าลูกปิงปองถูกลำเลียงไปยังจานลำเลียง 2 มอเตอร์เซอร์โวตัวที่ 2 จะหมุนจานลำเลียง 2 ให้ส่งไปยังกล่อง A หรือกล่อง B (ขึ้นอยู่กับสีของลูกปิงปอง) โดยกระบวนการลำเลียงลูกปิงปองนั้นขึ้นอยู่กับเงื่อนไขการทำงานที่ออกแบบให้ไมโครคอนโทรลเลอร์ประมวลผลได้

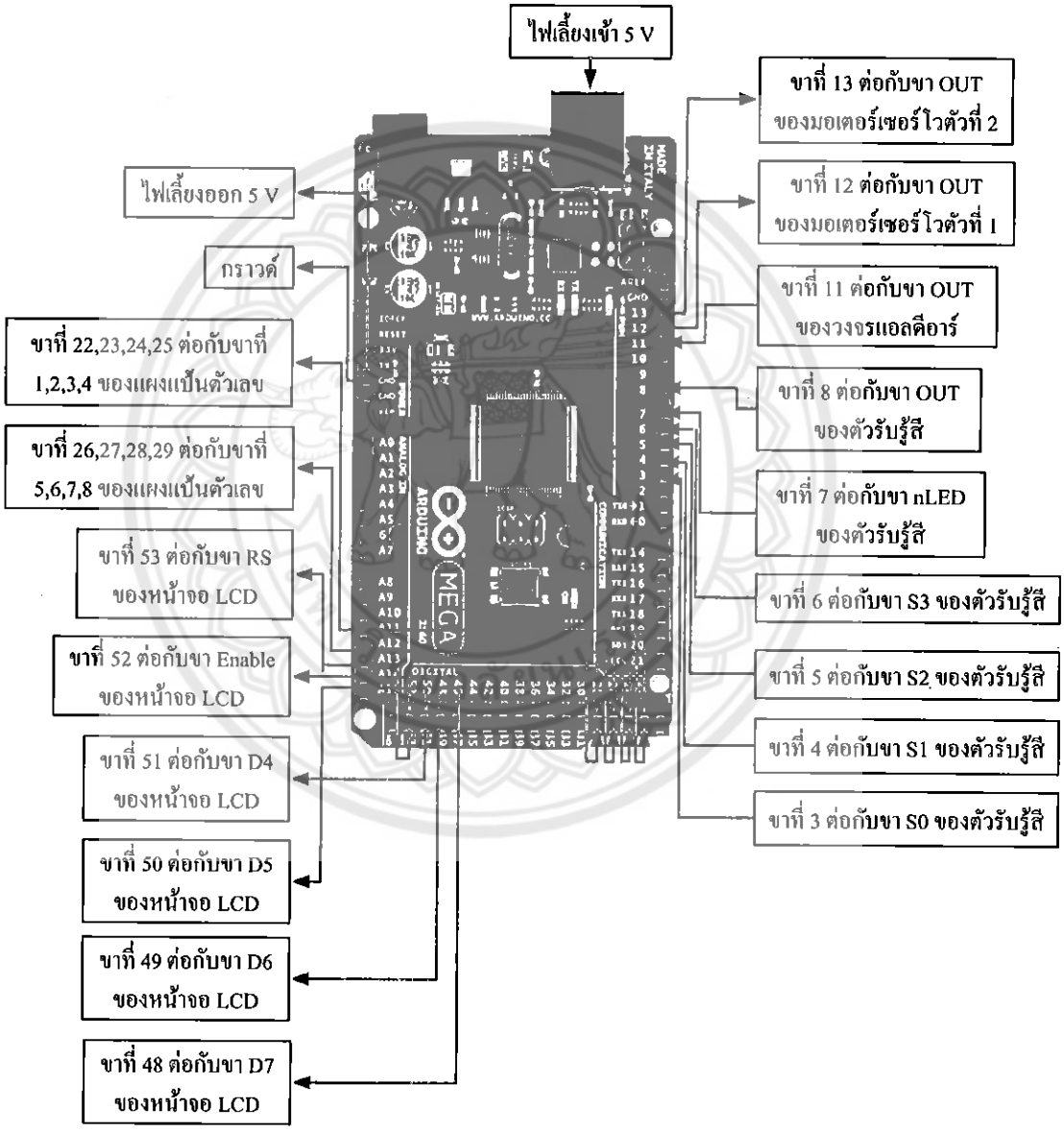


รูปที่ 3.1 แผนภาพการทำงานของระบบคัดแยกวัตถุตามสี

โดยส่วนประกอบหลักที่ส่งผลต่อการทำงานของระบบมีรายละเอียดดังนี้

3.1.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ที่เลือกใช้ในโครงงานนี้คือไมโครคอนโทรลเลอร์ตระกูล Arduino MEGA 2560 ซึ่งมีคุณสมบัติเด่นคือ มีดิจิตอลอินพุตและเอาท์พุตถึง 54 ขาด้วยกันดังรูปที่ 3.2 ที่เลือกใช้แผงวงจร Arduino ชนิดนี้ เนื่องจากโครงงานนี้มีการเชื่อมต่อกับอุปกรณ์ต่างๆได้แก่ สวิตช์ปุ่มกด ตัวรับรู้สี แผงแป้นตัวเลข มอเตอร์เซอร์โว และจอแสดงผลแอลซีดี



รูปที่ 3.2 แผนการเชื่อมต่อใช้งานไมโครคอนโทรลเลอร์กับอุปกรณ์ต่างๆ

จากรูปที่ 3.2 พบว่ามีการเชื่อมต่อดิจิตอลอินพุตและเอาท์พุตเป็นจำนวนมากดังนี้

พอร์ตที่ 3-8 และ 11 เป็นพอร์ตที่รับค่าจากตัวรับรู้สีโดยใช้หลักการสะท้อนของแสงขาว โดยไมโครคอนโทรลเลอร์จะรับค่าสัญญาณดิจิตอลที่เป็นลอจิก 0 และ1 มาจากตัวรับรู้สีแล้วนำค่าไปเก็บไว้เป็นค่าของแต่ละสี หลังจากที่ลูกปิงปองตกมาอยู่ที่ตำแหน่งด้านหน้าของตัวรับรู้

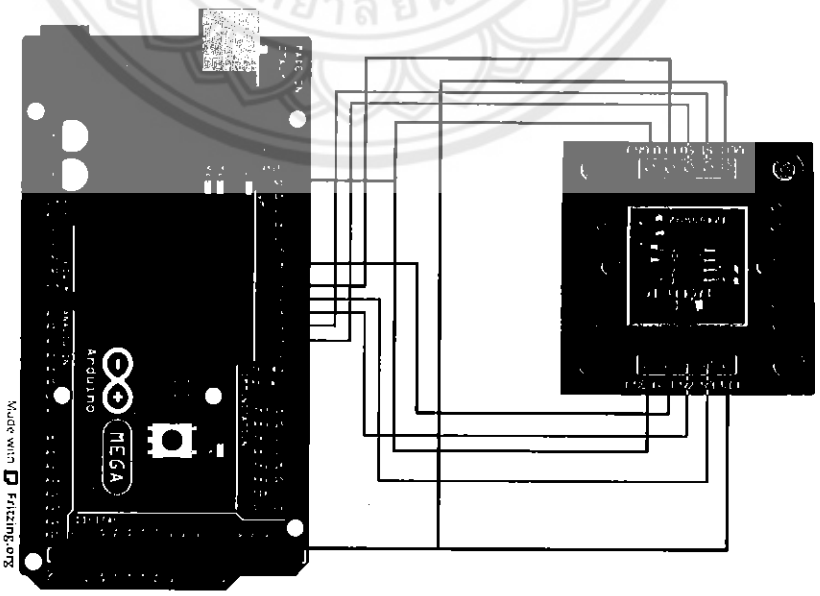
พอร์ตที่ 12 และพอร์ตที่ 13 เป็นพอร์ตที่ควบคุมการทำงานของมอเตอร์ซึ่งจะมีการเขียนโปรแกรมเพื่อสั่งการหมุนซ้ายหรือหมุนขวาเพื่อลำเลียงลูกปิงปอง และควบคุมรูปแบบการหมุนของมอเตอร์ โดยรับคำสั่งมาจากไมโครคอนโทลเลอร์

พอร์ตที่ 48-53 ต่อกับขา D7, D6, D5, D4, Enable และ RS ตามลำดับใช้ควบคุมการทำงานของจอแสดงผล ซึ่งมาจากการประมวลผลของไมโครคอนโทรลเลอร์ที่รับค่ามาจากตัวรับรู้สีและได้เก็บค่าไว้มาแสดงที่จอแอลซีดี

พอร์ตที่ 22-29 ต่อกับแถวและหลักของแผงแป้นตัวเลขทำหน้าที่รับข้อมูลมาจากผู้ใช้งาน

3.1.2 วงจรตัวรับรู้สี

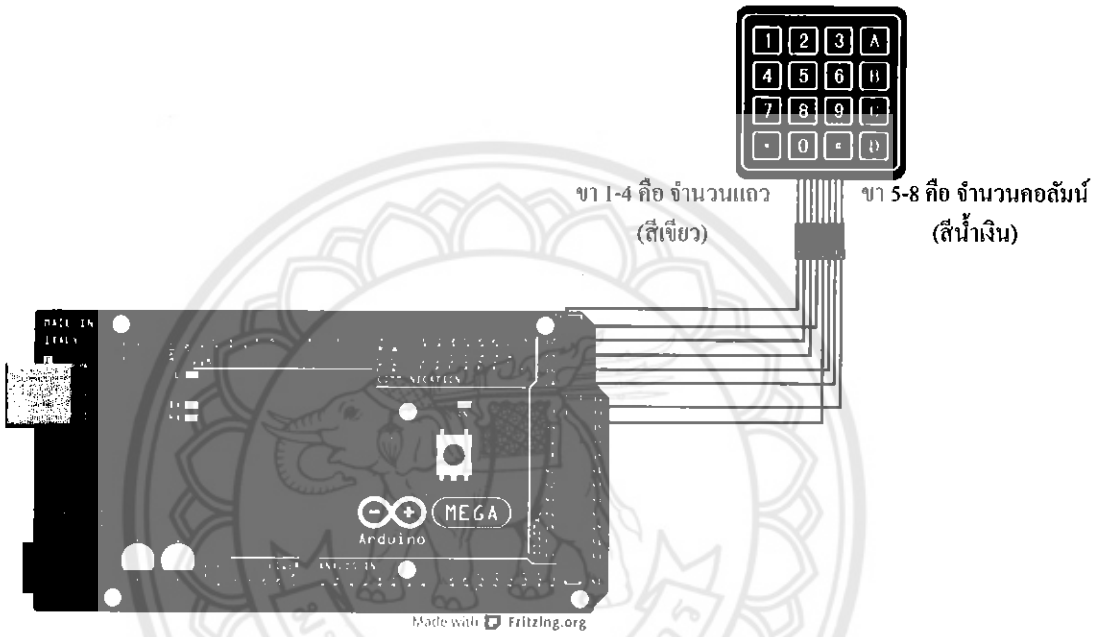
การเชื่อมต่อวงจรตัวรับรู้สีใช้ตัวรับรู้หมายเลข TCS230 แสดงดังรูปที่ 3.3 ซึ่งเป็นตัวรับรู้ที่ใช้ในการตรวจวัดค่าสีแล้วให้ไมโครคอนโทรลเลอร์เป็นตัวประมวลผลซึ่งการต่อใช้งานตัวรับรู้มี 10 ขา คือ ขา V_{CC} มีหน้าที่จ่ายไฟเลี้ยงแก่ตัวรับรู้ ขา GND มีหน้าที่เป็นขากราวด์ ขา OUT มีหน้าที่ส่งความถี่ไปไมโครคอนโทรลเลอร์ ขา LED มีหน้าที่กำหนดการหน่วงเวลาในการเปล่งแสงจากหลอดแอลอีดีสีขาว ขา S0 และ S1 มีหน้าที่ควบคุมอัตราขยายของค่าที่อ่านได้ ขา S3 และ S4 มีหน้าที่ส่งสัญญาณเอาท์พุตของไดโอดเปล่งแสงแต่ละชนิดให้กับไมโครคอนโทรลเลอร์



รูปที่ 3.3 แผนการเชื่อมต่อใช้งานตัวรับรู้สีกับไมโครคอนโทรลเลอร์

3.1.3 วงจรแผงแป้นตัวเลข

แผนการเชื่อมต่อระหว่างวงจรแผงแป้นตัวเลขกับไมโครคอนโทรลเลอร์แสดงดังรูปที่ 3.4 เป็นการเชื่อมต่อวงจรแผงแป้นตัวเลข (Keypad) กับไมโครคอนโทรลเลอร์ซึ่งมีการใช้งานของขาต่างๆดังนี้ ขา 1-4 มีหน้าที่ระบุแถวของแผงแป้นตัวเลข โดยขาที่ 1, 2, 3 และ 4 ระบุแถวของแผงแป้นตัวเลขที่ 1, 2, 3 และ 4 ตามลำดับ ขา 5-8 มีหน้าที่ระบุคอลัมน์ของแผงแป้นตัวเลข โดยขาที่ 5, 6, 7 และ 8 ระบุคอลัมน์ของแผงแป้นตัวเลขที่ 1, 2, 3 และ 4 ตามลำดับ

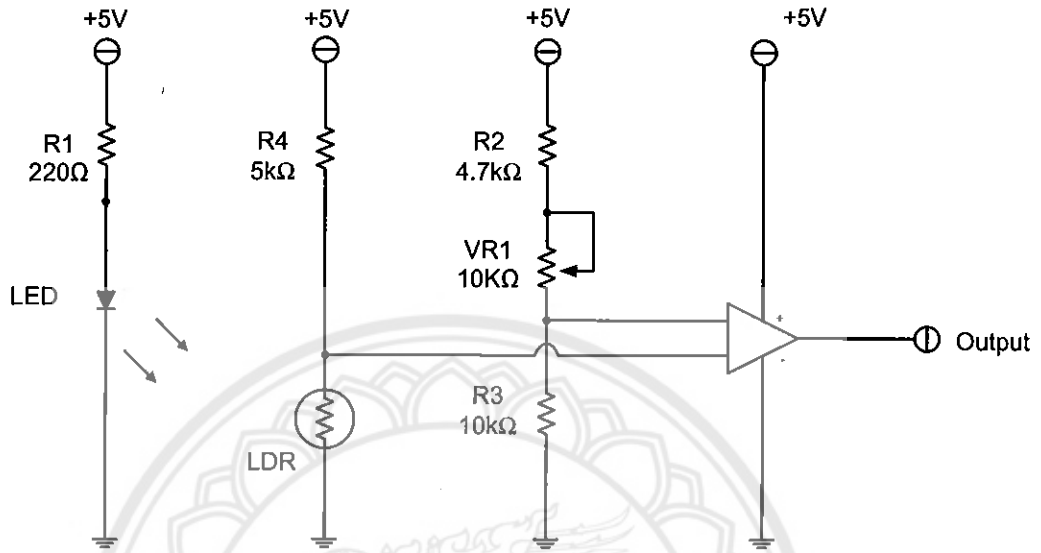


รูปที่ 3.4 แผนการเชื่อมต่อใช้งานแผงแป้นตัวเลขกับไมโครคอนโทรลเลอร์

3.1.4 วงจรตรวจจับลูกปิงปอง

การเชื่อมต่อแผงวงจรตรวจจับลูกปิงปอง แสดงดังรูปที่ 3.5 โดยการทำให้ตัวรับรู้สีสามารถหยุดอ่านค่าสีได้นั้น ต้องใช้ตัวตรวจจับลูกปิงปองที่ทำให้ทราบว่ามีลูกปิงปองอยู่หน้าตัวรับรู้สีหรือไม่ โดยมีแอลอีดีต่ออยู่กับไฟกระแสตรง 5 V ซึ่งทำตัวเป็นแหล่งกำเนิดแสง ให้แสงตกกระทบแอลดีอาร์ และแผนภาพวงจรตรวจจับลูกปิงปองแสดงดังรูปที่ 3.6 เมื่อไม่มีลูกปิงปองมาอยู่หน้าตัวรับรู้สี โดยมีหลักการคือ เมื่อแอลดีอาร์ได้รับแสงทำให้แอลดีอาร์มีค่าความต้านทานลดลง และถ้ามีลูปปิงปองอยู่หน้าตัวรับรู้สี ซึ่งมีแสงตกกระทบแอลดีอาร์น้อยลง ทำให้แอลดีอาร์มีค่าความต้านทานเพิ่มขึ้น การเปลี่ยนแปลงของค่าความต้านทานของแอลดีอาร์ทำให้แรงดันที่ขาข้างหนึ่งของออปแอมป์เปลี่ยนแปลงไปด้วย โดยออปแอมป์ทำการเปรียบเทียบแรงดันระหว่างอินพุตทั้งสองขา โดยแรงดันอินพุตอีกขาหนึ่งเป็นแรงดันอ้างอิง ซึ่งสามารถปรับเพิ่มค่าได้โดยปรับค่าความต้านทานของ VR1 โดยที่ค่าแรงดันอ้างอิงมีค่าแปรผกผันกับค่าความต้านทานของ VR1

ถ้าแรงดันที่ขาอินพุตทั้งสองมีค่าเท่ากัน ออปแอมป์จะส่งสัญญาณดิจิตอลออกเป็นลอจิก 0 แต่ถ้าไม่มีลูกปิงปองมาบังแสง ทำให้แรงดันระหว่างขาทั้งสองมีค่าไม่เท่ากัน ออปแอมป์จะส่งสัญญาณดิจิตอลออกเป็นลอจิก 1 ให้ไมโครคอนโทรลเลอร์ประมวลผล

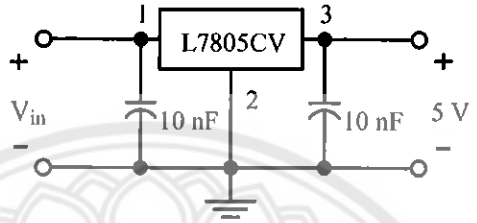


รูปที่ 3.5 แผนภาพวงจรตรวจจับลูกปิงปอง

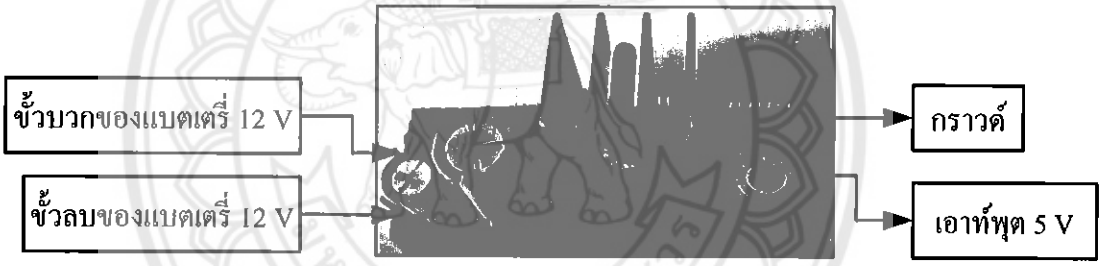
รูปที่ 3.6 แผนการเชื่อมต่อใช้งานวงจรตรวจจับลูกปิงปอง

3.1.5 วงจรคุมค่าแรงดัน 5 V

ในโครงงานนี้แผงแอลอีดี ที่ติดอยู่ข้างตัวรับรู้ที่ทำหน้าที่เพิ่มการสะท้อนสีของลูกปิงปองและเปล่งแสงให้กับแอลดีอาร์ ซึ่งวงจรคุมค่าแรงดันทำงานด้วยไฟกระแสตรง 5 V สร้างขึ้นจากไอซีคุมค่าแรงดันหมายเลข L7805CV โดยมีแผนภาพวงจรแสดงดังรูปที่ 3.7 โดยมีตัวเก็บประจุ 2 ตัว ทำหน้าที่ลดการกระเพื่อมของกระแสทั้งทางด้านอินพุตและเอาท์พุต ซึ่งแผงวงจรที่สร้างขึ้นแสดงดังรูปที่ 3.8 โดยมีตัวระบายความร้อนติดอยู่กับไอซีหมายเลข L7805CV



รูปที่ 3.7 แผนภาพวงจรควบคุมค่าแรงดัน 5 V



รูปที่ 3.8 แผนการเชื่อมต่อใช้งานแผงวงจรควบคุมค่าแรงดัน 5 V

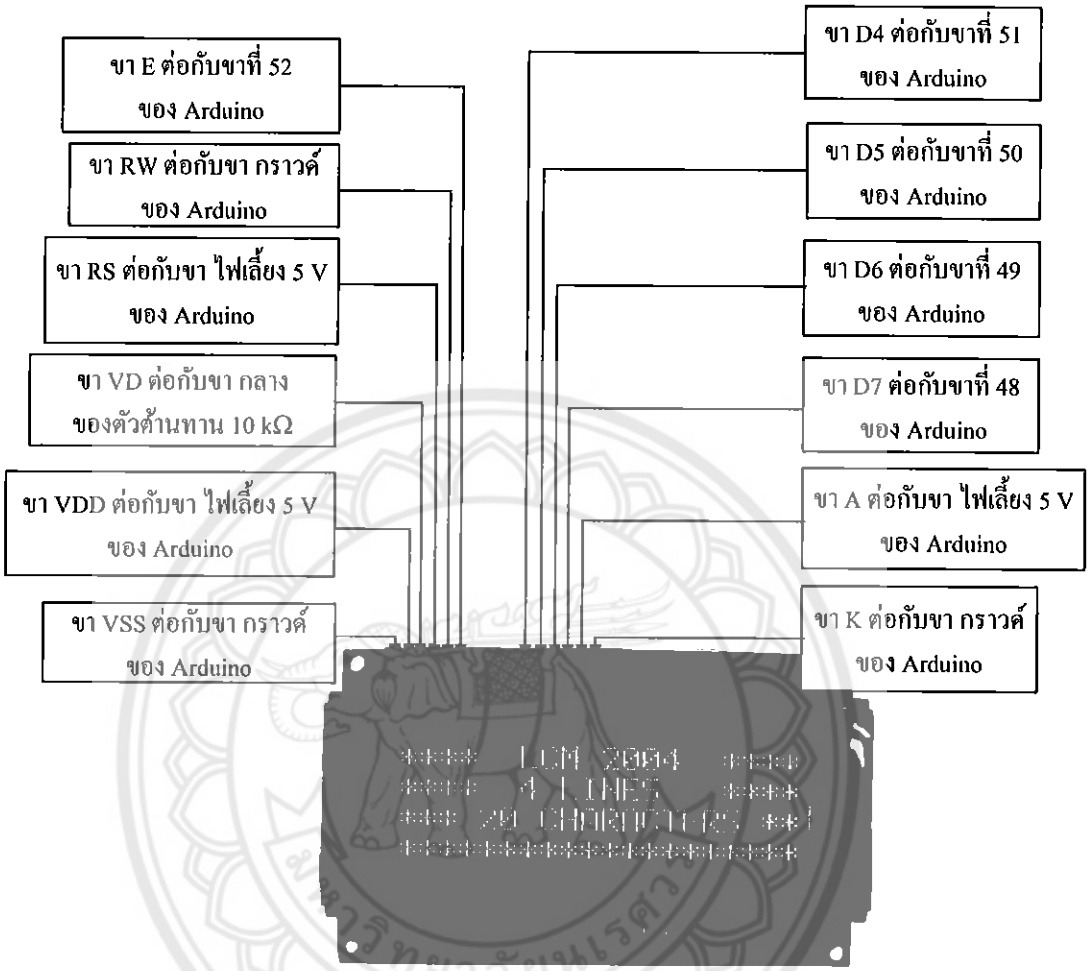
3.1.6 วงจรหน้าจอแสดงผล

วงจรหน้าจอแสดงผลถูกเชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ซึ่งเป็นจอแสดงผลขนาด 20 ตัวอักษร 4 บรรทัด โดยที่ไมโครคอนโทรลเลอร์ทำการประมวลผลคำสั่งจากอินพุตต่างๆได้แก่ แผงแป้นตัวเลข ตัวตรวจจับลูกปิงปอง ตัวรับรู้สีและปุ่มปรับตั้งการทำงานใหม่ ซึ่งหลังจากที่ประมวลผลแล้ว ไมโครคอนโทรลเลอร์จะแสดงผลผ่านทางหน้าจอแสดงผลโดยมีการเชื่อมต่อดังรูปที่ 3.9 จากรูปขา E ของจอแสดงผลที่ทำหน้าที่เป็นขาอินพุตสำหรับสัญญาณพัลส์ เมื่อต้องการเขียนหรืออ่านข้อมูลต่อกับขาที่ 52 ของArduino ขา RW เป็นขาอินพุตสำหรับเลือกแบบวิธีเขียนหรืออ่านข้อมูลต่อกับกราวด์ของ Arduino ขา RS เป็นขาอินพุตสำหรับเลือกเขียนอ่านข้อมูลในรีจิสเตอร์ต่อกับไฟเลี้ยง 5 V ของ Arduino ขา VD สำหรับปรับความเข้มตัวอักษรต่อกับตัวต้านทานปรับค่าได้ขนาด 10kΩ ขา VDD ต่อกับไฟเลี้ยง Arduino 5 V ขา VSS ต่อกับขากราวด์ของ Arduino ขา D4 ขา D5 ขา D6 และขา D7 เป็นขาข้อมูลของหน้าจอซึ่งต่อกับขา 51 50 49 และ 48 ตามลำดับ

ท.๗๑๙๘๕๗ ป
ก.ค.มก
๒๕๕๙

17 ต.ค. 2560

ขา A เป็นขา Vccสำหรับ LED backlight ต่อกับไฟเลี้ยง 5 V ของ Arduino และขา K เป็นขากราวด์สำหรับ LED backlight ต่อกับกราวด์ของ Arduino

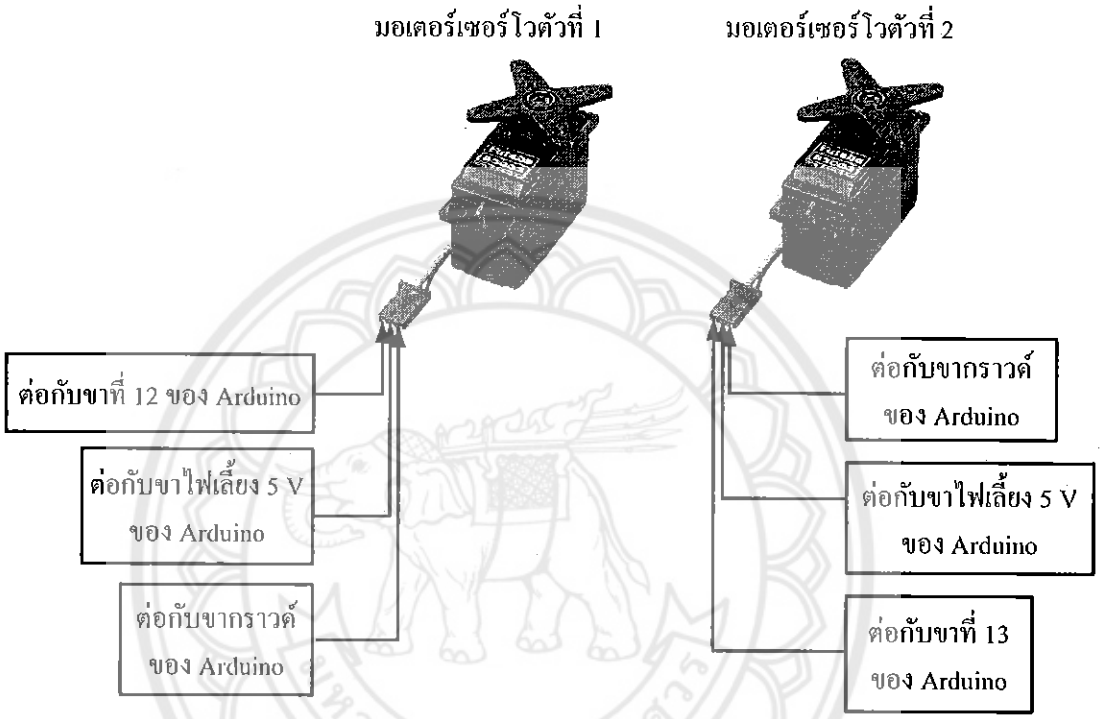


รูปที่ 3.9 แผนการเชื่อมต่อใช้งานหน้าจอแสดงผล

3.1.7 มอเตอร์เซอร์โว

ในโครงงานนี้ได้ใช้มอเตอร์เซอร์โวทั้งหมด 2 ตัว ซึ่งแต่ละตัวนั้นต้องมีการเชื่อมต่อกับไมโครคอนโทรลเลอร์ โดยมอเตอร์เซอร์โวมีหน้าที่เป็นตัวขับเคลื่อนให้จานลำเลียงลูกปิงปองหมุน ซึ่งในการหมุนนี้ส่งผลให้ลูกปิงปองที่อยู่ภายในจานลำเลียงลูกปิงปองถูกขนส่งไปยังจานลำเลียงถัดไปและกล่องต่างๆตามที่ไมโครคอนโทรลเลอร์ประมวลผลได้ โดยการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับมอเตอร์เซอร์โวทั้ง 2 ตัวนั้นผ่านทางสายข้อมูลของมอเตอร์ที่ต่อไปยังพอร์ตที่ 12 และพอร์ตที่ 13 ตามลำดับและนอกจากสายข้อมูลแล้วมอเตอร์เซอร์โวยังมีสายไฟเลี้ยงขนาด 5 V กับสายกราวด์ที่จะต้องเชื่อมต่อกับไฟเลี้ยง 5 V และ กราวด์ของไมโครคอนโทรลเลอร์ด้วยซึ่งได้แสดงดังรูปที่ 3.10 โดยไมโครคอนโทรลเลอร์จะสั่งให้มอเตอร์ทำงานโดยส่งสัญญาณลอจิก 0 และ 1 เพราะต่อกับพอร์ตที่เป็นข้อมูลแบบดิจิตอลแล้วทำให้มอเตอร์เซอร์โวทำงานโดย

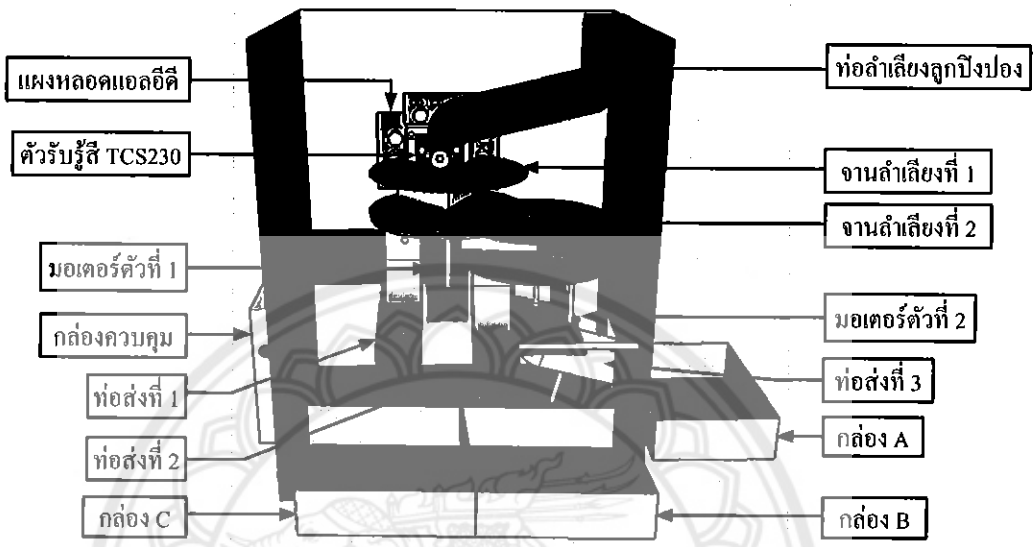
หมุนซึ่งช่วงที่มอเตอร์สามารถหมุนได้อยู่ระหว่าง 0° จนถึง 180° และในโครงงานนี้ต้องการให้มอเตอร์เซอร์โวหมุนได้ทั้งซ้ายและขวาจึงได้กำหนดองศาออกเป็นช่วงคือถ้าต้องการหมุนไปทางขวาจึงกำหนดให้หมุนไปที่ 0° ก่อนแล้วจึงหมุนกลับมาที่ 90° เพื่อกลับมายังตำแหน่งตรงกลางเหมือนเดิม ถ้าต้องการหมุนไปทางซ้ายจึงกำหนดให้หมุนไปที่ 180° ก่อนแล้วหมุนกลับมาที่ 90° เพื่อกลับมายังตำแหน่งตรงกลางเหมือนเดิม



รูปที่ 3.10 แผนการเชื่อมต่อใช้งานมอเตอร์เซอร์โว

3.2 การออกแบบโครงสร้างและการติดตั้งอุปกรณ์ในแบบจำลอง

โครงสร้างของแบบจำลองที่ออกแบบขึ้นในโครงงานนี้แสดงดังรูปที่ 3.11 โดยหน้าที่ของอุปกรณ์ต่างๆถูกอธิบายไว้ในตารางที่ 3.1



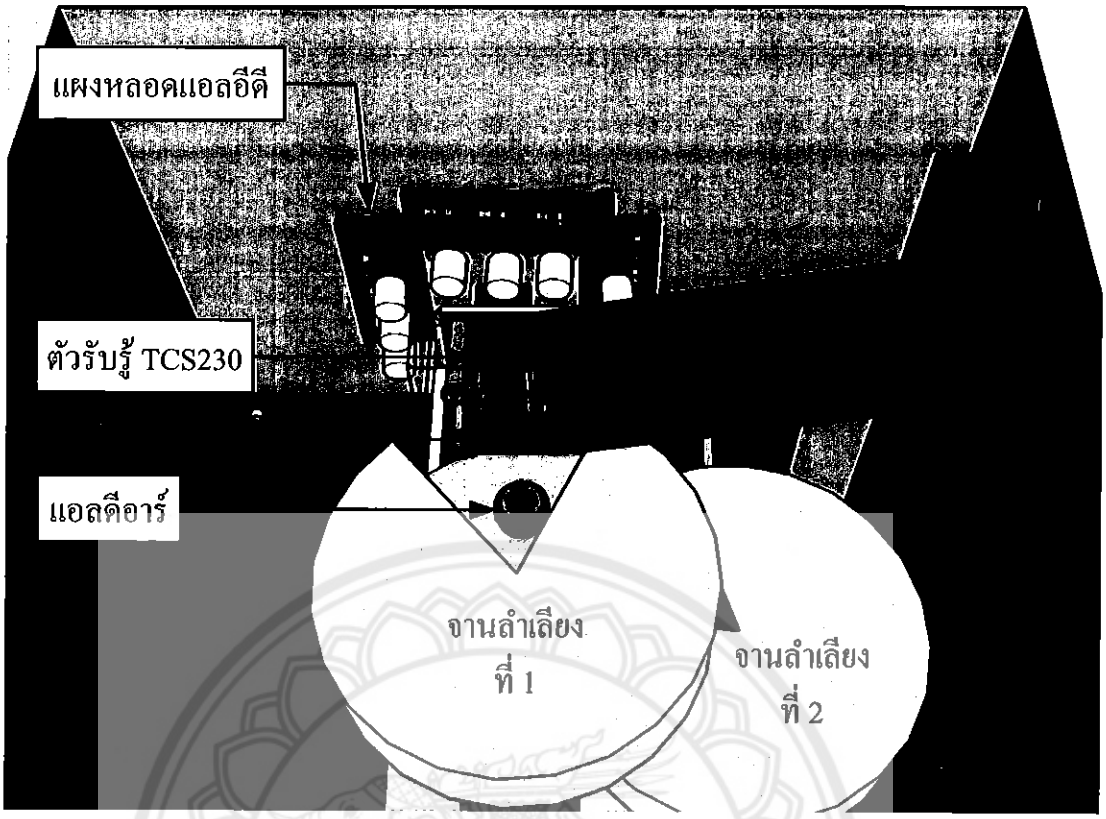
รูปที่ 3.11 โครงสร้างแบบจำลองของเครื่องคัดแยกวัตถุตามสี (มุมมองด้านหน้า)

ตำแหน่งการติดตั้งของแอลดีอาร์แสดงดังรูปที่ 3.12 โดยอยู่ด้านล่างจานลำเลียงที่ 1 ซึ่งแอลดีอาร์จะไม่ได้รับแสงก็ต่อเมื่อมีลูกปิงปองตกลงมาจากท่อปล่อยลูกปิงปอง แต่ถ้าไม่มีลูกปิงปองแอลดีอาร์จึงสามารถรับแสงได้เต็มที่เพราะอยู่ใกล้แอลอีดีโดยแอลดีอาร์ทำหน้าที่ตรวจจับว่าลูกปิงปองถูกลำเลียงมาหรือไม่และส่งสัญญาณไปให้ไมโครคอนโทรลเลอร์ประมวลผล

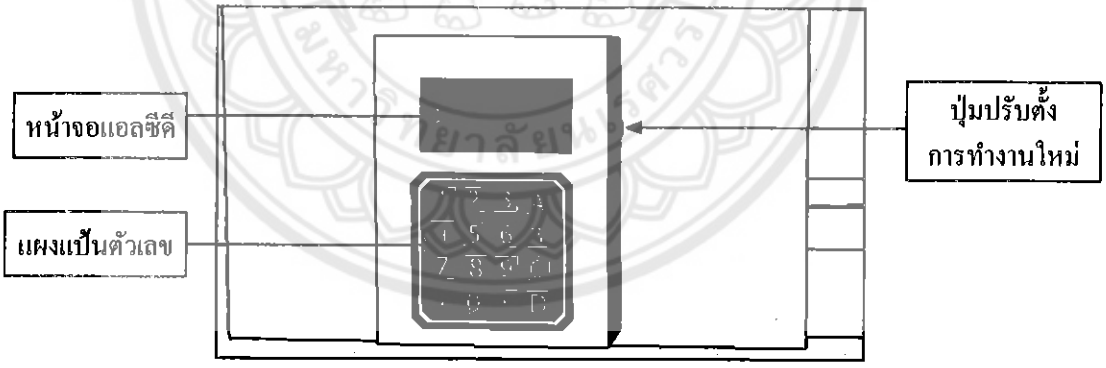
สำหรับด้านข้างของโครงสร้างมีกล่อง A ติดตั้งอยู่ด้านขวาของโครงสร้าง ส่วนด้านซ้ายของโครงสร้างเป็นกล่องควบคุม ดังแสดงในรูปที่ 3.13 ซึ่งในกล่องควบคุมประกอบด้วย วงจรแผงแป้นตัวเลข วงจรคุมค่าแรงดัน 5 V วงจรหน้าจอแสดงผล ปุ่มปรับตั้งการทำงานใหม่ วงจรตรวจจับลูกปิงปอง รวมไปถึงไมโครคอนโทรลเลอร์

ตารางที่ 3.1 หน้าที่ของอุปกรณ์ต่างๆในแบบจำลองเครื่องคัดแยกวัตถุตามสี

อุปกรณ์	หน้าที่
ท่อลำเลียงลูกปิงปอง	ลำเลียงลูกปิงปองไปยังจานลำเลียงที่ 1
จานลำเลียงที่ 1	รับลูกปิงปองจากท่อปล่อยลูกปิงปองแล้วลำเลียงไปยังจานลำเลียงที่ 2 หรือกล่อง C ขึ้นอยู่กับการหมุนของมอเตอร์ตัวที่ 1
จานลำเลียงที่ 2	รับลูกปิงปองจากจานลำเลียงที่ 1 จากนั้นลำเลียงไปยัง กล่อง A หรือ กล่อง B ขึ้นอยู่กับการหมุนของมอเตอร์ตัวที่ 2
มอเตอร์ตัวที่ 1	ขับเคลื่อนจานลำเลียงที่ 1 ให้ลำเลียงลูกตามที่ไมโครคอนโทรลเลอร์ประมวลผลได้
มอเตอร์ตัวที่ 2	ขับเคลื่อนจานลำเลียงที่ 2 ให้ลำเลียงลูกตามที่ไมโครคอนโทรลเลอร์ประมวลผลได้
ท่อส่งที่ 1	รับลูกปิงปองที่มาจากจานลำเลียงที่ 1 แล้วส่งไปยังกล่อง C
ท่อส่งที่ 2	รับลูกปิงปองที่มาจากจานลำเลียงที่ 2 แล้วส่งไปยังกล่อง B
ท่อส่งที่ 3	รับลูกปิงปองที่มาจากจานลำเลียงที่ 2 แล้วส่งไปยังกล่อง A
กล่อง A	รับลูกปิงปองจากท่อส่งที่ 3 แล้วบรรจุไว้ที่กล่อง A ในแบบวิธีคัดเลือกเมื่อกล่อง A ต้องการ และบรรจุลูกปิงปองสีแดงในแบบวิธีคัดแยก
กล่อง B	รับลูกปิงปองจากท่อส่งที่ 2 แล้วบรรจุไว้ที่กล่อง B ในแบบวิธีคัดเลือกเมื่อกล่อง B ต้องการ และบรรจุลูกปิงปองสีน้ำเงินในแบบวิธีคัดแยก
กล่อง C	รับลูกปิงปองจากท่อส่งที่ 1 แล้วบรรจุไว้ที่กล่อง C ในแบบวิธีคัดเลือกและบรรจุลูกปิงปองสีเขียวในแบบวิธีคัดแยก



รูปที่ 3.12 ตำแหน่งการติดตั้งแอลดีอาร์และตัวรับรู้สีในแบบจำลอง

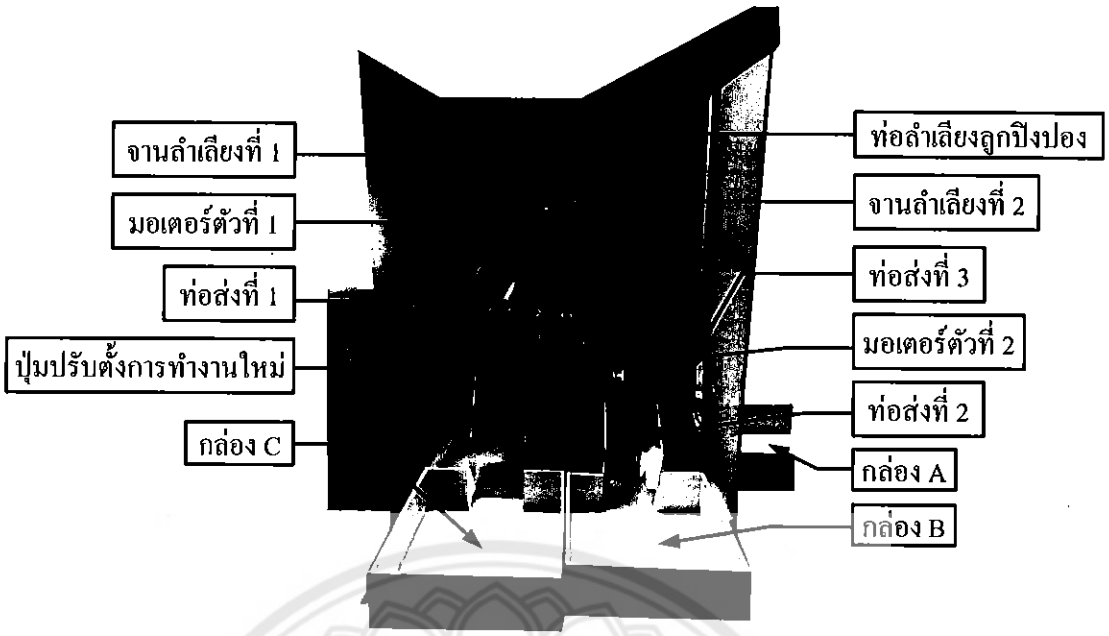


รูปที่ 3.13 การติดตั้งกล่องควบคุมไว้ด้านข้างของโครงสร้างแบบจำลอง

หลังจากที่ได้ออกแบบแล้ว จึงอธิบายรายละเอียดของอุปกรณ์ได้ดังตารางที่ 3.2 และเมื่อดูแบบจำลองในมุมมองด้านหน้าสามารถดูได้จากรูปที่ 3.14

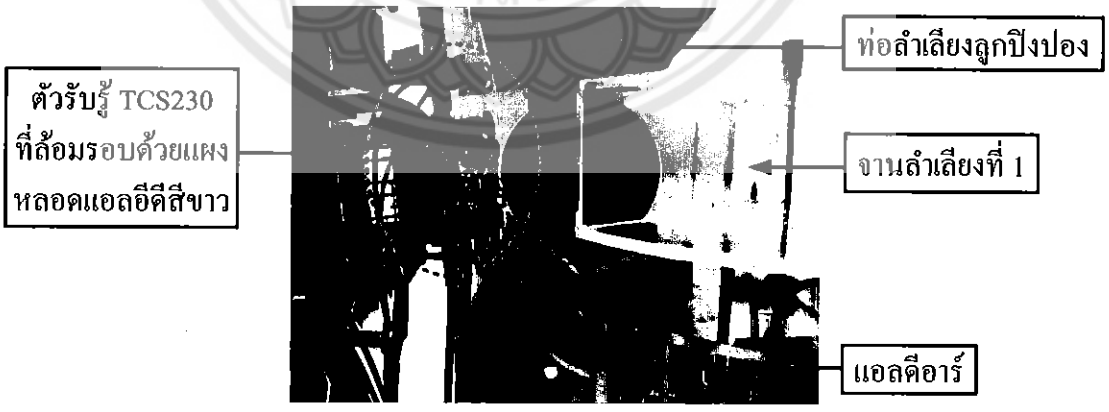
ตารางที่ 3.2 รายละเอียดของอุปกรณ์ต่างๆในแบบจำลองเครื่องคัดแยกวัตถุตามสี

อุปกรณ์	รายละเอียด
ท่อลำเลียงลูกปิงปอง	ติดตั้งไว้กับผนังข้างในของโครงสร้างโดยท่อปล่อยลูกปิงปองทำมาจากอะคริลิคสีดำ ที่มีความยาวประมาณ 30 cm และทำมุมกับผนังด้านข้าง 60°
จานลำเลียงที่ 1	ทำมาจากอะคริลิคใสที่ตัดเป็นรูปวงกลมโดยมีรัศมี 3.5 cm ถูกติดตั้งไว้ด้านหน้าของตัวรับรู้สีเพื่อที่จะรับและลำเลียงลูกปิงปองได้
จานลำเลียงที่ 2	ทำมาจากอะคริลิคใสที่ตัดเป็นรูปวงกลมโดยมีรัศมี 3.5 cm ถูกติดตั้งไว้ด้านล่างเยื้องมาทางขวาของจานลำเลียงที่ 1 เพื่อที่จะรับลูกปิงปองจากจานลำเลียงที่ 1 ได้
มอเตอร์ตัวที่ 1	อยู่ด้านล่างของจานลำเลียงที่ 1 และถูกเชื่อมติดกับแกนของจานลำเลียงที่ 1
มอเตอร์ตัวที่ 2	อยู่ด้านล่างของจานลำเลียงที่ 2 และถูกเชื่อมติดกับแกนของจานลำเลียงที่ 2
ท่อส่งที่ 1	ทำมาจาก อะคริลิคใสความยาวประมาณ 20 cm อยู่ด้านล่างของจานลำเลียงที่ 1 ซึ่งด้านบนของท่อมีวงกลมรัศมี 2.15 cm และลูกปิงปองสามารถผ่านได้
ท่อส่งที่ 2	ทำมาจาก อะคริลิคใสความยาวประมาณ 13 cm อยู่ด้านล่างฝั่งซ้ายจานลำเลียงที่ 2 โดยด้านบนของท่อนี้มีวงกลมรัศมี 2.15 cm และลูกปิงปองสามารถผ่านได้
ท่อส่งที่ 3	ทำมาจาก อะคริลิคใสความยาวประมาณ 13 cm อยู่ด้านล่างฝั่งขวาจานลำเลียงที่ 2 โดยด้านบนของท่อนี้มีวงกลมรัศมี 2.15 cm และลูกปิงปองสามารถผ่านได้
กล่องบรรจุ	ทำมาจากไม้อัดสร้างเป็นรูปสี่เหลี่ยมที่มีความกว้าง 17 cm ยาว 13 cm และสูง 4 cm โดยเปิดด้านบนไว้ เพื่อรับลูกปิงปองจากท่อลำเลียง โดยมีกล่องอยู่ 3 กล่องคือกล่อง A กล่อง B และกล่อง C ซึ่งในแต่ละกล่องสามารถบรรจุลูกปิงปองได้ประมาณ 15 ลูก โดยจะติดตั้งกล่อง A ไว้ด้านขวาของแบบจำลองซึ่งรับลูกปิงปองมาจากท่อส่งที่ 3 ส่วนกล่อง B และ C นั้นถูกติดตั้งไว้ตรงด้านหน้าของแบบจำลอง โดยรับลูกปิงปองมาจากท่อส่งที่ 2 และ 1 ตามลำดับ



รูปที่ 3.14 แบบจำลองของเครื่องคัดแยกวัตถุตามสี (มุมมองด้านหน้า)

ในการติดตั้งตัวรับรู้สี แผงหลอดแอลอีดี และแอลดีอาร์แสดงดังรูป 3.16 ซึ่งแอลดีอาร์ถูกติดตั้งไว้ด้านล่างจานลำเลียงและติดตั้งอยู่ตรงกับช่องที่ลูกปิงปองถูกส่งมาจากท่อลำเลียงลูกปิงปอง เพื่อให้ลูกปิงปองตกลงมาทับแอลดีอาร์พอดีเพื่อบังแสงจากแอลอีดี ในส่วนของตัวรับรู้สีอยู่ด้านหน้าของจานลำเลียงที่ 1 โดยด้านข้างและด้านบนตัวรับรู้สี จะติดตั้งแอลอีดีสีขาวไว้เพื่อช่วยในการให้แสงขาวเพื่อเพิ่มความแม่นยำของการตรวจสี ซึ่งแผงหลอดแอลอีดี มีจำนวนทั้งหมด 3 แผงๆ ละ 3 หลอด โดยเป็นแอลอีดีสีขาวมีวงจรคุมค่าแรงดัน 5 V จ่ายไฟให้



รูปที่ 3.15 ตำแหน่งการติดตั้งตัวรับรู้สีและแอลดีอาร์

3.3 การออกแบบการทำงานของเครื่องคัดแยกวัตถุตามสี

3.3.1 การออกแบบขั้นตอนการทำงานโดยรวม

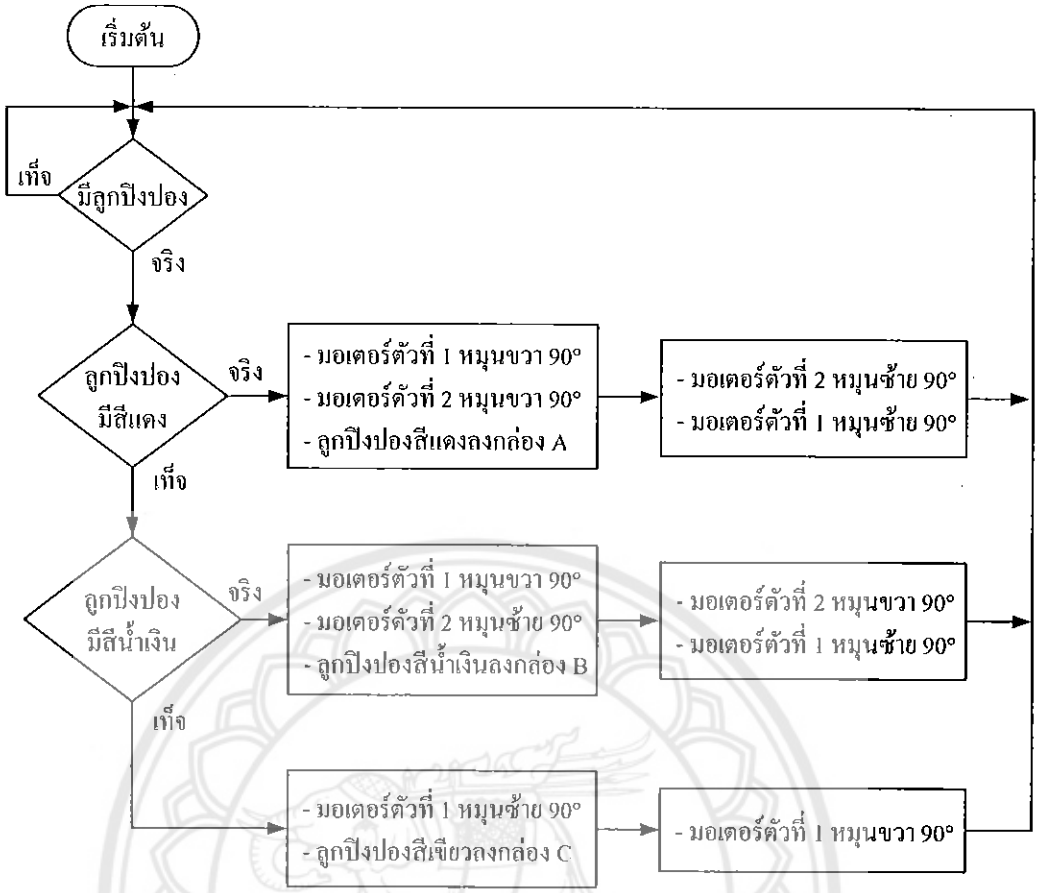
เครื่องคัดแยกวัตถุตามสีในโครงงานนี้ถูกออกแบบให้มีการทำงาน 2 แบบวิธี คือแบบวิธีคัดแยกสี (Separation mode) และแบบวิธีคัดเลือกสี (Sclection mode) เครื่องเริ่มต้นทำงานโดยให้ผู้ใช้เลือกแบบวิธีที่ผู้ใช้ต้องการและเครื่องทำงานในแต่ละแบบวิธีอย่างต่อเนื่องจนกระทั่งผู้ใช้เปลี่ยนแบบวิธีด้วยการกดปุ่มปรับตั้งการทำงานใหม่ (Reset) โดยสถานะการทำงานของเครื่องถูกแสดงบนหน้าจอแอลซีดี การทำงานของเครื่องในแต่ละแบบวิธีมีรายละเอียดดังนี้โดยใช้ลูกปิงปองสีแดง สีเขียว และสีน้ำเงินเป็นวัตถุทดสอบ

3.3.2 แบบวิธีคัดแยกสี

ขั้นตอนการทำงานในแบบวิธีคัดแยกสีมีรายละเอียดดังแสดงในรูปที่ 3.16 โดยเริ่มต้นจากการตรวจจับว่ามีลูกปิงปองในเครื่องหรือไม่ เครื่องจะยังไม่ดำเนินการใดต่อจนกว่าจะตรวจพบลูกปิงปอง ขั้นตอนถัดไปคือการตรวจสอบสีของลูกปิงปองด้วยตัวรับรู้ TCS230 ถ้าพบว่าเป็นลูกปิงปองสีแดง มอเตอร์ตัวที่ 1 จะหมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียง จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางขวา 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง A แล้วมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

ในกรณีที่ตรวจพบว่าเป็นลูกปิงปองสีน้ำเงิน มอเตอร์ตัวที่ 1 จะหมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียง มอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง B จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

เมื่อตรวจพบว่าเป็นลูกปิงปองสีเขียว มอเตอร์ตัวที่ 1 จะหมุนจานลำเลียงที่ 1 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง C และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียง จากนั้นมอเตอร์ตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1 นั่นคือการคัดแยกลูกปิงปองสีเขียวนั้นอาศัยการทำงานมอเตอร์ตัวที่ 1 เพียงลำพัง



รูปที่ 3.16 ขั้นตอนการทำงานในแบบวิธีคัดแยกสี

3.3.3 แบบวิธีคัดเลือกสี

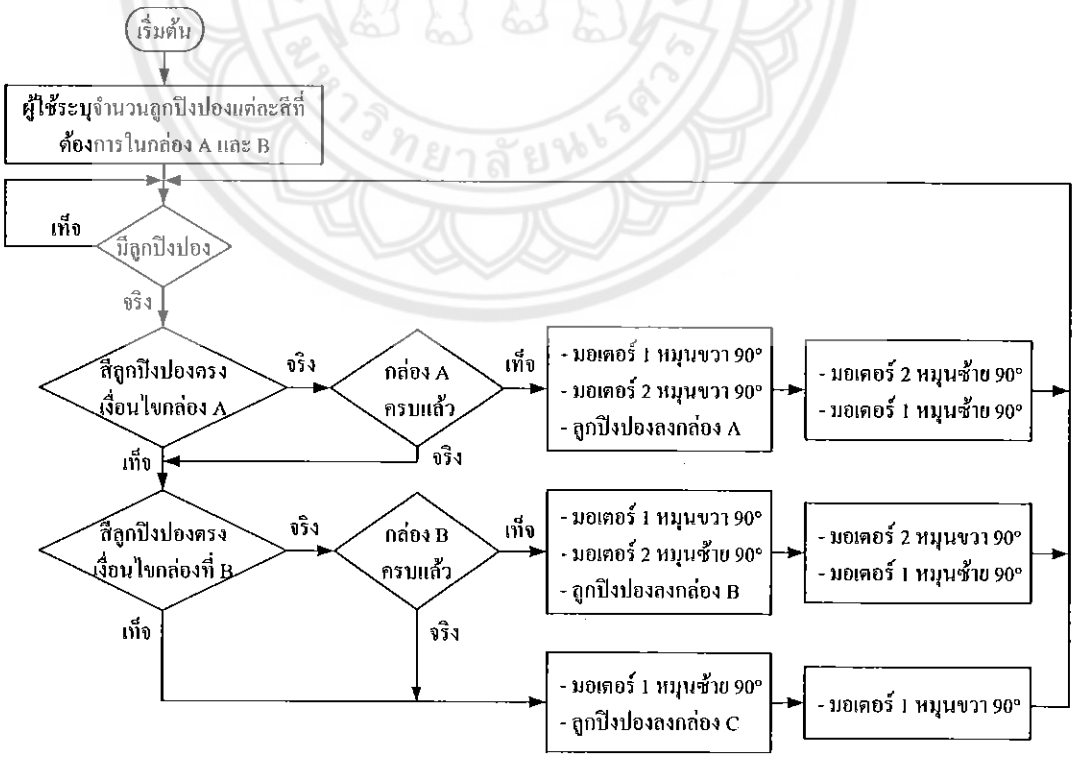
ขั้นตอนการทำงานในแบบวิธีคัดเลือกสีมีรายละเอียดดังแสดงในรูปที่ 3.17 โดยเริ่มต้นจากการให้ผู้ใช้ระบุจำนวนของลูกปิงปองแต่ละสีที่ต้องการในกล่อง A และ B ได้ ในขณะที่กล่อง C ถูกกำหนดเป็นกล่อง Spare เพื่อใส่ลูกปิงปองที่ไม่ตรงกับเงื่อนไขของผู้ใช้หรือเมื่อกล่อง A หรือ B มีลูกปิงปองครบตามเงื่อนไขแล้ว ขั้นตอนถัดไปเป็นการตรวจจับว่ามีลูกปิงปองในเครื่องหรือไม่ เครื่องจะยังไม่ดำเนินการใดต่อจนกว่าจะตรวจพบลูกปิงปอง จากนั้นจึงเป็นการตรวจสอบสีของลูกปิงปองด้วยตัวรับรู้ TCS230 ในระหว่างที่กล่อง A และ B ยังมีจำนวนลูกปิงปองไม่ครบตามเงื่อนไขที่ระบุ ลูกปิงปองที่ตรวจพบว่าสอดคล้องกับเงื่อนไขของทั้งสองกล่องจะถูกคัดเลือกให้ไหลลงกล่อง A ก่อนเพื่อให้กล่อง A มีจำนวนครบตามเงื่อนไขก่อนกล่อง B เสมอ

เมื่อตรวจพบลูกปิงปองที่มีสีตรงกับเงื่อนไขสำหรับกล่อง A และจำนวนลูกปิงปองสีนั้นในกล่อง A ยังไม่ครบตามที่ผู้ใช้ต้องการ เครื่องจะส่งลูกปิงปองนั้นไปยังกล่อง A นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียง จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางขวา 90° เพื่อ

ส่งลูกปิงปองให้ตกลงไปที่กล่อง A จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

หากตรวจสอบพบว่าลูกปิงปองสีนั้นไม่ตรงกับเงื่อนไขสำหรับกล่อง A หรือในกล่อง A มีจำนวนลูกปิงปองสีนั้นครบตามที่ผู้ใช้ต้องการแล้ว เครื่องจะตรวจสอบว่าลูกปิงปองนั้นสอดคล้องกับเงื่อนไขที่ระบุสำหรับกล่อง B หรือไม่ ถ้าสีของลูกปิงปองตรงกับที่ผู้ใช้ต้องการสำหรับกล่อง B และจำนวนลูกปิงปองสีนั้นในกล่อง B ยังไม่ครบตามที่ผู้ใช้ระบุไว้ เครื่องจะส่งลูกปิงปองนั้นไปยังกล่อง B นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียง จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง B จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

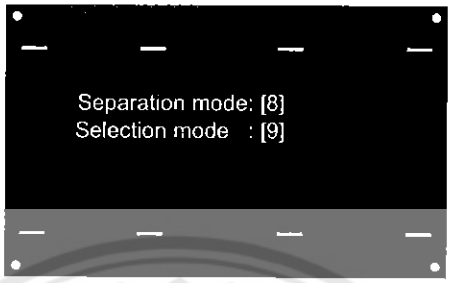
ถ้าตรวจสอบพบว่าลูกปิงปองสีนั้นไม่ตรงกับเงื่อนไขสำหรับทั้งกล่อง A และ B หรือเมื่อพบว่ากล่องทั้งสองมีจำนวนลูกปิงปองสีนั้นครบตามที่ผู้ใช้ต้องการแล้ว มอเตอร์ตัวที่ 1 จะหมุนจานลำเลียงที่ 1 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองลงไปยังกล่อง C และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียง จากนั้นจึงหมุนย้อนกลับไปตำแหน่งเดิมเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1



รูปที่ 3.17 ขั้นตอนการทำงานในแบบวิธีคัดเลือกสี

3.4 รูปแบบการแสดงผลบนหน้าจอแอลซีดีขณะใช้งาน

การทำงานของเครื่องคัดแยกวัตถุตามสี โดยเริ่มต้นการใช้งานของเครื่องจะแสดงผ่านหน้าจอแสดงผลดังรูปที่ 3.18 เมื่อกดปุ่มหมายเลข 8 จะเข้าสู่แบบวิธีคัดแยก และถ้ากดปุ่มหมายเลข 9 จะเข้าสู่แบบวิธีคัดเลือกโดยอธิบายได้ดังนี้



รูปที่ 3.18 รูปแบบการแสดงผลขณะให้เลือกแบบวิธีทำงาน

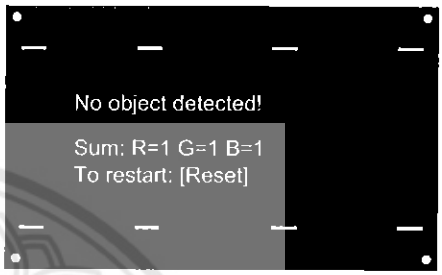
3.4.1 การแสดงผลขณะใช้งานในแบบวิธีการคัดแยก

จากรูปที่ 3.18 เมื่อเลือกแบบวิธีคัดแยก ระบบจึงเริ่มทำการคัดแยกลูกปิงปองโดยมีเงื่อนไขคือ ถ้าเป็นลูกปิงปองสีแดง ไมโครคอนโทรลเลอร์ประมวลผลและสั่งให้หน้าจอแอลซีดีแสดงผลดังรูปแบบในรูปที่ 3.19 ก) ซึ่งบ่งบอกว่าลูกปิงปองสีแดงจะถูกส่งไปกล่อง A พร้อมทั้งแจ้งจำนวนรวมของลูกปิงปองสีแดงที่คัดแยกได้ แต่ถ้าเป็นลูกปิงปองสีเขียวรูปแบบการแสดงผลบนหน้าจอแอลซีดีจะเป็นดังรูปที่ 3.19 ข) โดยบ่งบอกว่าลูกปิงปองสีเขียวจะถูกส่งไปกล่อง C พร้อมทั้งแจ้งจำนวนรวมของลูกปิงปองสีเขียวที่คัดแยกได้ แต่ถ้าเป็นลูกปิงปองสีน้ำเงินรูปแบบการแสดงผลบนหน้าจอแอลซีดีจะเป็นดังรูปที่ 3.19 ค) โดยบ่งบอกว่าลูกปิงปองสีน้ำเงินจะถูกส่งไปกล่อง B พร้อมทั้งแจ้งจำนวนรวมของลูกปิงปองสีน้ำเงินที่คัดแยกได้ แต่ถ้าตรวจไม่พบลูกปิงปองรูปแบบการแสดงผลบนหน้าจอแอลซีดีจะเป็นดังรูปที่ 3.19 ง) โดยระบุว่า 'No object detected!' พร้อมทั้งแสดงจำนวนรวมของลูกปิงปองแต่ละสีที่คัดแยกได้ อย่างไรก็ตามถ้าตรวจพบลูกปิงปองเพิ่มกระบวนการคัดแยกจะดำเนินต่อไปโดยการแสดงผลบนหน้าจอแอลซีดีจะสอดคล้องกับสีที่ตรวจพบดังที่อธิบายมาแล้วข้างต้น อย่างไรก็ตามผู้ใช้สามารถหยุดการทำงานของเครื่องได้โดยกดปุ่มปรับตั้งการทำงานใหม่ (Reset) ซึ่งการแสดงผลบนหน้าจอจะเปลี่ยนกลับไปเป็นดังรูปที่ 3.18

ก) กรณีที่เป็นลูกปิงปองสีแดง

ข) กรณีที่เป็นลูกปิงปองสีเขียว

ค) กรณีที่เป็นลูกปิงปองสีน้ำเงิน

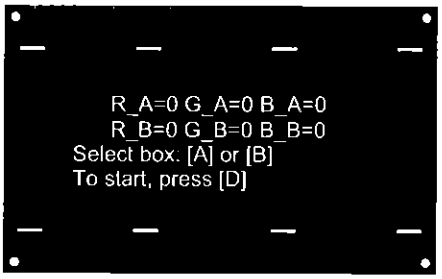


ง) กรณีที่ไม่พบลูกปิงปอง

รูปที่ 3.19 รูปแบบแสดงผลการตรวจสีในแบบวิธีคัดแยก

3.4.2 การแสดงผลในแบบวิธีคัดเลือก

จากรูปที่ 3.18 ถ้าเลือกแบบวิธีคัดเลือก หน้าจอแอลซีดีจะแสดงดังรูปแบบในรูปที่ 3.20 โดยบรรทัดแรกและบรรทัดที่สองแสดงการสรุปจำนวนลูกปิงปองแต่ละสีในกล่อง A และ B ที่ผู้ใช้ต้องการตามลำดับ ในขณะนี้จำนวนของลูกปิงปองถูกแสดงเป็นศูนย์ทั้งหมดเพราะยังไม่มีการป้อนตัวเลขจากผู้ใช้ โดยก่อนที่จะกำหนดสีและจำนวนของลูกปิงปองที่ต้องการผู้ใช้ต้องเลือกกล่องบรรจุเสียก่อนซึ่งสามารถเลือกกล่องใดก่อนก็ได้ โดยกดปุ่ม A เพื่อเลือกกล่อง A หรือกดปุ่ม B เพื่อเลือกกล่อง B ทั้งนี้ผู้ใช้ไม่สามารถเลือกกล่อง C ในแบบวิธีนี้ได้เพราะถูกกำหนดให้เป็นกล่อง Spare สำหรับใส่ลูกปิงปองที่ไม่เข้าเงื่อนไขของกล่อง A และ B หรือในกรณีที่กล่องทั้งสองมีลูกปิงปองครบตามเงื่อนไขแล้ว อย่างไรก็ตามหากผู้ใช้กดปุ่ม D เพื่อให้เครื่องเริ่มการทำงาน ลูกปิงปองทุกลูกที่ตรวจจับได้จะถูกส่งไปกล่อง C เพราะถือว่าผู้ใช้ไม่ต้องการให้มีลูกปิงปองในกล่อง A และ B



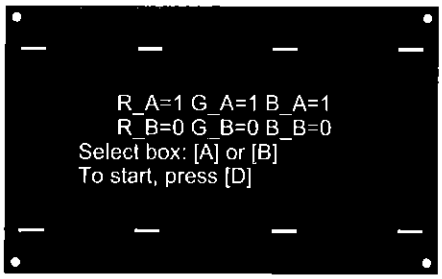
รูปที่ 3.20 รูปแบบการเลือกกล่อง A หรือกล่อง B

ในกรณีที่ผู้ใช้เลือกกล่อง A เป็นลำดับแรก ขั้นตอนถัดไปจึงเป็นการป้อนจำนวนลูกปิงปองในแต่ละสีที่ต้องการในกล่อง A โดยหน้าจอแสดงผลจะปรากฏให้ผู้ใช้ป้อนโดยเรียงจากสีแดง สีเขียว และสีน้ำเงิน ดังรูปแบบที่แสดงในรูปที่ 3.21 ก)-ค) ตามลำดับ โดยในการป้อนจำนวนของแต่ละสีนั้นให้ผู้ใช้กดตัวเลขจำนวนที่ต้องการ (ตั้งแต่ 0 ถึง 5) ตามด้วยปุ่ม '*' เพื่อเป็นการยืนยัน อย่างไรก็ตาม ถ้าป้อนตัวเลขแต่ยังไม่ได้กด '*' ผู้ใช้สามารถเปลี่ยนเลขจำนวนที่ต้องการใหม่ได้ หลังจากป้อนจำนวนของแต่ละครบตามต้องการแล้วหน้าจอจะแสดงจำนวนของแต่ละสีในกล่อง A ที่ผู้ใช้เพิ่งป้อนเสร็จไปโดยมีรูปแบบดังแสดงในรูปที่ 3.21 ง) จากนั้นให้ผู้ใช้กดปุ่ม '#' เพื่อกลับสู่หน้าจอที่แสดงการสรุปจำนวน โดยจะพบว่าเลขจำนวนลูกปิงปองแต่ละสีในกล่อง A ที่ผู้ใช้ป้อนไปแล้วจะถูกแสดงในบรรทัดแรกบนหน้าจอสรุปจำนวนดังแสดงตัวอย่างในรูปที่ 3.23

ก) การป้อนจำนวนของสีแดง ข) การป้อนจำนวนของสีเขียว

ค) การป้อนจำนวนของสีน้ำเงิน ง) การแสดงจำนวนแต่ละสีที่ป้อน

รูปที่ 3.21 รูปแบบแสดงผลการป้อนจำนวนในแต่ละสีของกล่อง A



รูปที่ 3.22 รูปแบบแสดงการสรุปจำนวนทั้งหมดเมื่อป้อนจำนวนของกล่อง A

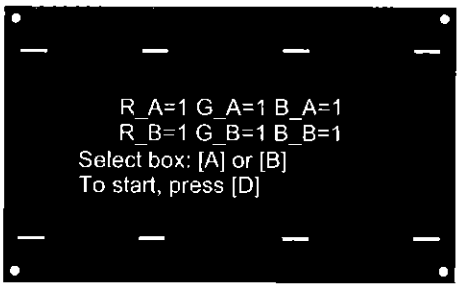
ถ้าขณะนี้ผู้ใช้กดปุ่ม D เครื่องจะเริ่มทำงานโดยไม่มีลูกปิงปองถูกส่งไปยังกล่อง B เพราะจำนวนลูกปิงปองแต่ละสีของกล่อง B ที่ผู้ใช้ต้องการซึ่งเครื่องบันทึกค่าไว้นั้นยังคงเป็นศูนย์อยู่

ในกรณีที่ผู้ใช้เลือกกล่อง B ถัดจากกล่อง A ขั้นตอนถัดไปจึงเป็นการป้อนจำนวนลูกปิงปองในแต่ละสีที่ต้องการในกล่อง B ซึ่งหน้าจอแสดงผลจะปรากฏให้ผู้ใช้ป้อนโดยเรียงจากสีแดง สีเขียว และสีน้ำเงิน ดังรูปแบบที่แสดงในรูปที่ 3.23 ก)-ค) ตามลำดับ โดยในการป้อนจำนวนของแต่ละสีนั้นให้ผู้ใช้กดตัวเลขจำนวนที่ต้องการ (ตั้งแต่ 0 ถึง 5) ตามด้วยปุ่ม '*' เพื่อเป็นการยืนยัน อย่างไรก็ตาม ถ้าป้อนตัวเลขแต่ยังไม่ได้กด '*' ผู้ใช้สามารถเปลี่ยนเลขจำนวนที่ต้องการใหม่ได้ หลังจากป้อนจำนวนของแต่ละสีครบตามต้องการแล้วหน้าจอจะแสดงจำนวนของแต่ละสีในกล่อง B ที่ผู้ใช้เพิ่งป้อนเสร็จไปโดยมีรูปแบบดังแสดงในรูปที่ 3.23 ง) จากนั้นให้ผู้ใช้กดปุ่ม '#' เพื่อกลับสู่หน้าจอที่แสดงการสรุปจำนวน โดยจะพบว่าเลขจำนวนลูกปิงปองแต่ละสีในกล่อง B ที่ผู้ใช้ป้อนไปแล้วจะถูกแสดงในบรรทัดที่สองบนหน้าจอสรุปจำนวนดังแสดงตัวอย่างในรูปที่ 3.24

ก) การป้อนจำนวนของแดง ข) การป้อนจำนวนของสีเขียว

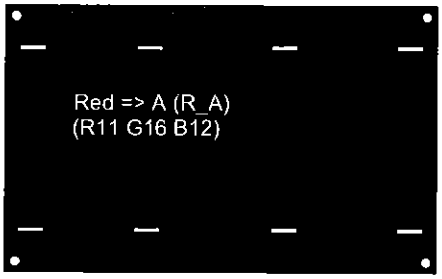
ค) การป้อนจำนวนของสีน้ำเงิน ง) การแสดงจำนวนแต่ละสีที่ป้อน

รูปที่ 3.23 รูปแบบแสดงผลการป้อนจำนวนในแต่ละสีของกล่อง B

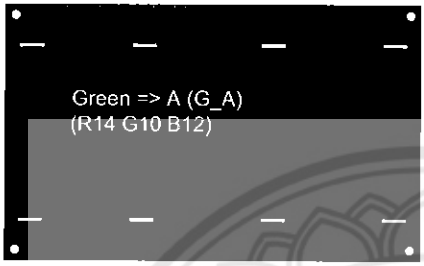


รูปที่ 3.24 รูปแบบแสดงการสรุปจำนวนทั้งหมดเมื่อป้อนจำนวนของกล่อง B

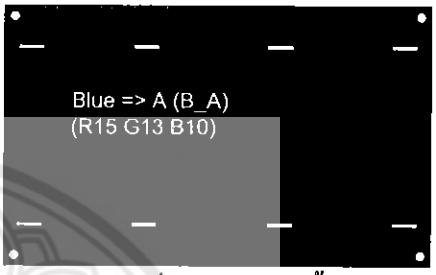
ตราบใดที่จำนวนลูกปิงปองแต่ละสีในกล่อง A ยังไม่ครบตามเงื่อนไข การตรวจจับเพื่อคัดเลือกลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง A จึงเกิดขึ้นอย่างต่อเนื่องโดยมีรูปแบบการแสดงผลบนหน้าจอดังรูปที่ 3.25



ก) กรณีที่ลูกปิงปองสีแดง



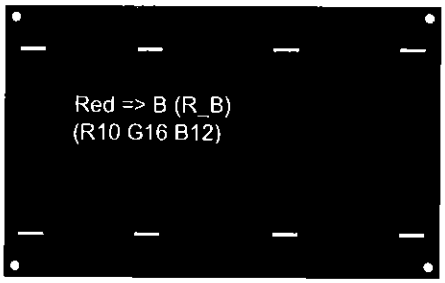
ข) กรณีที่ลูกปิงปองสีเขียว



ค) กรณีที่ลูกปิงปองสีน้ำเงิน

รูปที่ 3.25 รูปแบบการแสดงผลเมื่อตรวจเจอลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง A

ในทำนองเดียวกัน ถ้าจำนวนลูกปิงปองแต่ละสีในกล่อง B ยังไม่ครบตามเงื่อนไข การตรวจจับเพื่อคัดเลือกลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง B จึงเกิดขึ้นอย่างต่อเนื่องโดยมีรูปแบบการแสดงผลบนหน้าจอดังรูปที่ 3.26



ก) กรณีที่ลูกปิงปองสีแดง

ข) กรณีที่ลูกปิงปองสีเขียว

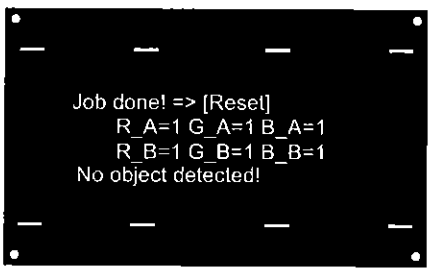
ค) กรณีที่ลูกปิงปองสีน้ำเงิน

รูปที่ 3.26 รูปแบบการแสดงผลเมื่อตรวจเจอลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง B

ในระหว่างที่กระบวนการคัดเลือกยังได้จำนวนลูกปิงปองแต่ละสีไม่ครบตามเงื่อนไขสำหรับกล่อง A และ B ถ้าปรากฏว่าวงจรตัวรับรู้ตรวจไม่พบลูกปิงปอง ได้แก่ กรณีที่ลูกปิงปองหมด หน้าจอแอลซีดีจะแสดงผลตามรูปแบบดังรูปที่ 3.27 ก) โดยบรรทัดแรกแสดงข้อความ 'No object detected!' เพื่อแจ้งให้ผู้ใช้ทราบ และในบรรทัดที่สองแสดงข้อความ 'Add more ping-pongs!' เพื่อแจ้งให้ผู้ใช้ป้อนเพิ่มลูกปิงปองเพื่อให้เครื่องทำการคัดเลือกต่อไป

ถ้าการตรวจไม่พบลูกปิงปองเกิดขึ้นหลังจากคัดเลือกเสร็จสิ้นตามเงื่อนไขแล้วหน้าจอจะแสดงผลดังรูปแบบในรูปที่ 3.27 ข) โดยมีข้อความ 'No object detected!' ในบรรทัดล่างสุด ส่วนบรรทัดแรกมีข้อความ 'Job done! => [Reset]' เพื่อแจ้งให้ผู้ใช้ทราบว่าเครื่องทำการคัดเลือกเสร็จแล้วและผู้ใช้สามารถกดปุ่มปรับตั้งการทำงานใหม่ (Reset) เพื่อกลับไปสู่หน้าจอเลือกแบบวิธีอีกครั้ง ส่วนบรรทัดที่ 2 และ 3 แสดงจำนวนลูกปิงปองแต่ละสีในกล่อง A และ B ตามลำดับ

ก) กรณีที่จำนวนยังไม่ครบตามเงื่อนไข

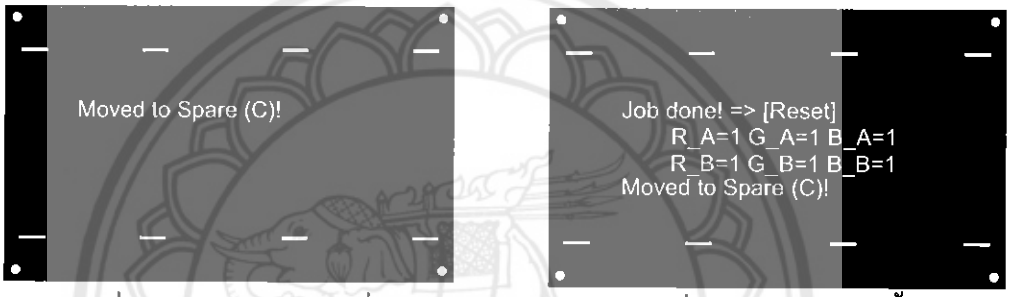


ข) กรณีที่จำนวนครบตามเงื่อนไข

รูปที่ 3.27 รูปแบบการแสดงผลเมื่อตรวจไม่พบลูกปิงปอง

ในระหว่างที่กระบวนการคัดเลือกยังไม่เสร็จสิ้น ถ้าตรวจพบลูกปิงปองที่ไม่ตรงกับเงื่อนไขของทั้งกล่อง A และ B ลูกปิงปองนั้นจะถูกส่งไปยังกล่อง Spare (หรือกล่อง C นั่นเอง) โดยหน้าจอแสดงผลระบุข้อความ 'Move to Spare (C)!' ดังรูปแบบในรูปที่ 3.28 ก)

ถ้าตรวจพบลูกปิงปองที่ไม่ตรงกับเงื่อนไขของทั้งกล่อง A และ B หลังจากที่กระบวนการคัดเลือกเสร็จสิ้นแล้วหน้าจอจะแสดงผลดังรูปแบบในรูปที่ 3.28 ข) โดยแสดงข้อความ 'Move to Spare (C)!' ในบรรทัดล่างสุด ในขณะที่บรรทัดแรกมีข้อความ 'Job done! => [Reset]' เพื่อแจ้งให้ผู้ใช้ทราบว่าเครื่องทำการคัดเลือกเสร็จแล้วและผู้ใช้สามารถกดปุ่มปรับตั้งการทำงานใหม่ (Reset) เพื่อกลับไปสู่หน้าจอเลือกแบบวิธีอีกครั้ง ส่วนบรรทัดที่ 2 และ 3 แสดงจำนวนลูกปิงปองแต่ละสีในกล่อง A และ B ตามลำดับ ซึ่งผู้ใช้ได้ป้อนไว้ก่อนหน้านี้



ก) กรณีที่ลูกปิงปองไม่ตรงกับเงื่อนไข ข) กรณีที่การคัดเลือกเสร็จสิ้นแล้ว

รูปที่ 3.28 รูปแบบการแสดงผลเมื่อส่งลูกปิงปองไปกล่อง C

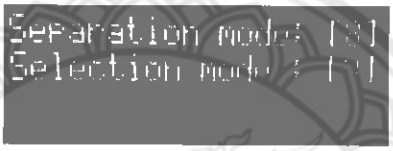
บทที่ 4

ผลการทดสอบ

ในบทนี้จะกล่าวถึงผลการทดสอบการทำงานของแบบจำลองในแบบวิธีคัดแยกและการคัดเลือกลูกปิงปองตามสี รวมทั้งการทดสอบหาระยะเวลาเฉลี่ยในการคัดแยกลูกปิงปองในแต่ละสี

4.1 การทดสอบการทำงานของแบบวิธีคัดแยกและแบบวิธีคัดเลือก

เมื่อเริ่มการทำงานของเครื่อง ผู้ใช้กดเลือกแบบวิธีทำงานที่ต้องการดังแสดงในรูปที่ 4.1 โดยกดปุ่มหมายเลข 8 เพื่อเข้าสู่แบบวิธีคัดแยก หรือกดปุ่มหมายเลข 9 เพื่อเข้าสู่แบบวิธีคัดเลือก

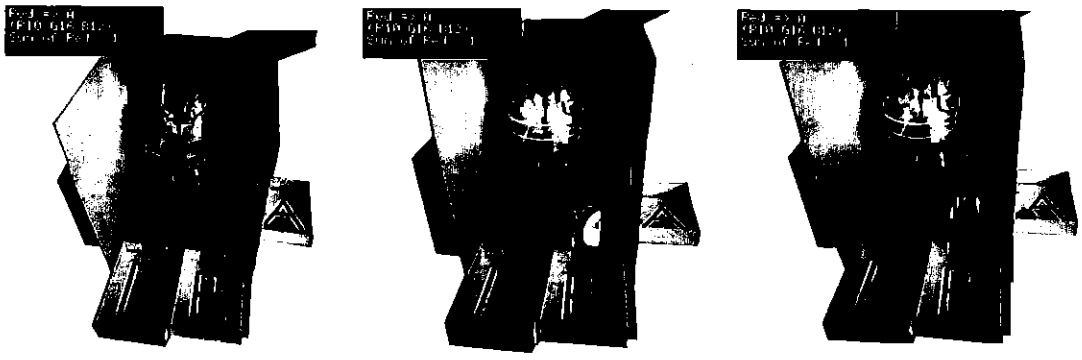


รูปที่ 4.1 หน้าจอแสดงแบบวิธีทำงานให้ผู้ใช้เลือก

4.1.1 การทดสอบการคัดแยก

เมื่อกดปุ่มหมายเลข 8 เพื่อเข้าสู่แบบวิธีคัดแยก เครื่องจะเริ่มอ่านสีของลูกปิงปอง ถ้าตรวจพบว่าเป็นลูกปิงปองสีแดง หน้าจอแอลซีดีจะแจ้งจำนวนรวมของลูกปิงปองสีแดงที่ตรวจพบดังแสดงในรูปที่ 4.2 ก) มอเตอร์ตัวที่ 1 จะหมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังแสดงในรูปที่ 4.2 ข) จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางขวา 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง A ดังรูปที่ 4.2 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

เมื่อตรวจพบว่าเป็นลูกปิงปองสีเขียว หน้าจอแอลซีดีจะแจ้งจำนวนรวมของลูกปิงปองสีเขียวที่ตรวจพบดังแสดงในรูปที่ 4.3 ก) มอเตอร์ตัวที่ 1 จะหมุนจานลำเลียงที่ 1 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง C และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังรูปที่ 4.3 ข) จากนั้นมอเตอร์ตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1 โดยจะเห็นได้ว่าในการคัดแยกลูกปิงปองสีเขียวนั้นไม่ได้อาศัยการทำงานมอเตอร์ตัวที่ 2 ซึ่งมีหน้าที่หมุนจานลำเลียงที่ 2 เลย



ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนขวา

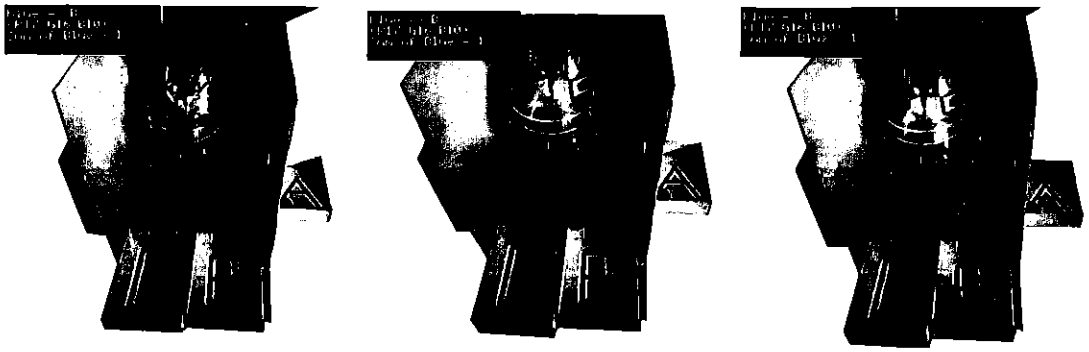
รูปที่ 4.2 การคัดแยกลูกปิงปองสีแดง

ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนซ้าย

รูปที่ 4.3 การคัดแยกลูกปิงปองสีเขียว

ในกรณีที่ตรวจพบว่าเป็นลูกปิงปองสีน้ำเงิน หน้าจอแอลซีดีจะแจ้งจำนวนรวมของลูกปิงปองสีน้ำเงินที่ตรวจพบดังแสดงในรูปที่ 4.4 ก) มอเตอร์ตัวที่ 1 จะหมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียง ดังรูปที่ 4.2 ข) มอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง B ดังรูปที่ 4.4 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

หลังจากที่ไม่มีลูกปิงปองไหลลงมาจากท่อลำเลียง หน้าจอแอลซีดีจะระบุ 'No object detected!' พร้อมทั้งแสดงจำนวนรวมของลูกปิงปองแต่ละสีที่เครื่องได้คัดแยกดังแสดงในรูปที่ 4.5 อย่างไรก็ตามในระหว่างนี้หากตรวจพบลูกปิงปองเพิ่มกระบวนการคัดแยกจะดำเนินต่อไป โดยผู้ใช้สามารถหยุดการทำงานของเครื่องได้ด้วยการกดปุ่มปรับตั้งการทำงานใหม่ (Reset) ซึ่งการแสดงผลบนหน้าจอจะเปลี่ยนกลับไปเป็นดังรูปที่ 4.1



ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนซ้าย

รูปที่ 4.4 การคัดแยกลูกปิงปองสีน้ำเงิน

รูปที่ 4.5 การแสดงผลบนหน้าจอเมื่อตรวจไม่พบลูกปิงปองในแบบวิธีคัดแยก

4.1.2 การทดสอบการคัดเลือก

จากรูปที่ 4.1 เมื่อผู้ใช้กดปุ่มหมายเลข 9 เพื่อให้เครื่องทำงานในแบบวิธีคัดเลือก ผู้ใช้สามารถระบุจำนวนของลูกปิงปองแต่ละสีที่ต้องการในกล่อง A และ B ได้ ในขณะที่กล่อง C ถูกใช้เป็นกล่อง Spare เพื่อใส่ลูกปิงปองที่ไม่ตรงกับเงื่อนไขของผู้ใช้หรือเมื่อกล่อง A หรือ B มีลูกปิงปองครบตามเงื่อนไขแล้ว

หลังจากที่ผู้ใช้ป้อนจำนวนของลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง A และ B ดังแสดงตัวอย่างในรูปที่ 4.6 แล้วจึงกดปุ่ม D เพื่อให้เครื่องเริ่มกระบวนการคัดเลือกลูกปิงปอง

รูปที่ 4.6 ตัวอย่างการเลือกลูกปิงปองของแต่ละสีในแต่ละกล่อง

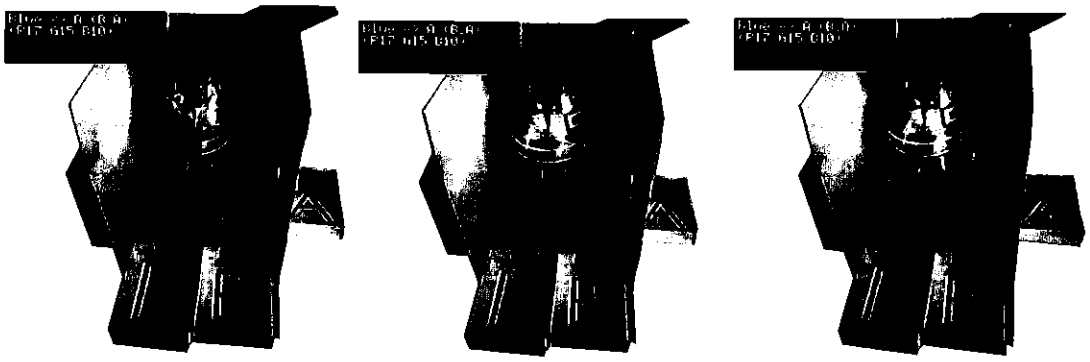
ในระหว่างที่กล่อง A และ B ยังมีจำนวนลูกปิงปองไม่ครบตามเงื่อนไขที่ระบุ ลูกปิงปองที่ตรวจพบว่าสอดคล้องกับเงื่อนไขของทั้งสองกล่องจะถูกคัดเลือกให้ไหลลงกล่อง A ก่อนเพื่อให้กล่อง A มีจำนวนครบตามเงื่อนไขก่อนกล่อง B เสมอ

ถ้าอ้างอิงจากค่าที่ป้อนเป็นตัวอย่างในรูปที่ 4.6 เมื่อตรวจพบลูกปิงปองสีแดงที่ต้องการสำหรับกล่อง A หน้าจอแอลซีดีจึงระบุในบรรทัดแรกว่าเป็นสีแดงและจะส่งไปกล่อง A ดังแสดงในรูปที่ 4.7 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังแสดงในรูปที่ 4.7 ข) จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางขวา 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง A ดังรูปที่ 4.7 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนขวา

รูปที่ 4.7 การคัดเลือกลูกปิงปองสีแดงไปกล่อง A

เมื่อลูกปิงปองลูกต่อมามีสีน้ำเงินซึ่งสอดคล้องกับเงื่อนไขสำหรับกล่อง A หน้าจอแอลซีดีจึงระบุในบรรทัดแรกว่าเป็นสีน้ำเงินและจะส่งไปยังกล่อง A ดังแสดงในรูปที่ 4.8 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังรูปที่ 4.8 ข) จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางขวา 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง A ดังรูปที่ 4.8 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1



ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนขวา

รูปที่ 4.8 การคัดเลือกลูกปิงปองสีน้ำเงินไปกล่อง A

เมื่อตรวจพบว่าลูกปิงปองลูกต่อมามีสีแดงซึ่งไม่สอดคล้องกับเงื่อนไขสำหรับกล่อง A เพราะมีจำนวนลูกปิงปองสีแดงครบตามที่ผู้ใช้ต้องการแล้ว แต่สอดคล้องกับเงื่อนไขสำหรับกล่อง B หน้าจอแอลซีดีจึงระบุในบรรทัดแรกว่าเป็นสีแดงและจะส่งไปยังกล่อง B ดังแสดงในรูปที่ 4.9 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังรูปที่ 4.9 ข) จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง B ดังรูปที่ 4.9 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนซ้าย

รูปที่ 4.9 การคัดเลือกลูกปิงปองสีแดงไปกล่อง B

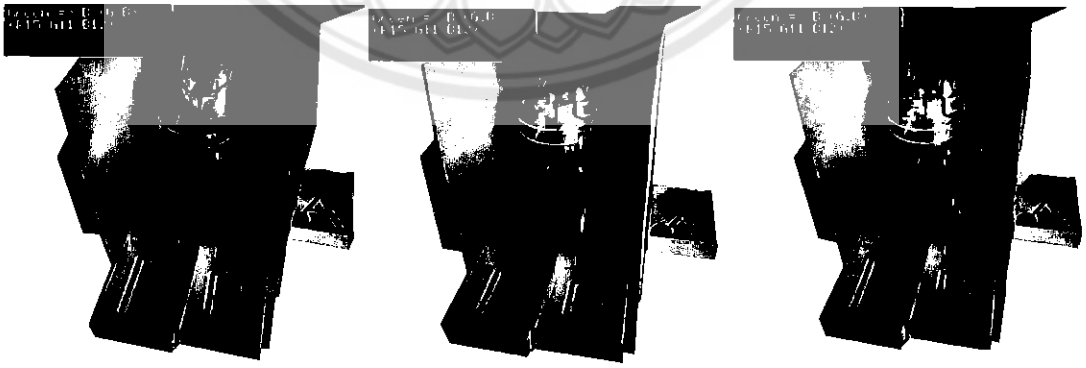
เมื่อลูกปิงปองลูกต่อมามีสีเขียวซึ่งสอดคล้องกับเงื่อนไขสำหรับกล่อง A หน้าจอแอลซีดีจึงระบุในบรรทัดแรกว่าเป็นสีเขียวและจะส่งไปยังกล่อง A ดังแสดงในรูปที่ 4.10 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังรูปที่ 4.10 ข) จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางขวา 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง A ดังรูปที่ 4.10 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัว

ที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1

ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนขวา

รูปที่ 4.10 การคัดเลือกลูกปิงปองสีเขียวไปกล่อง A

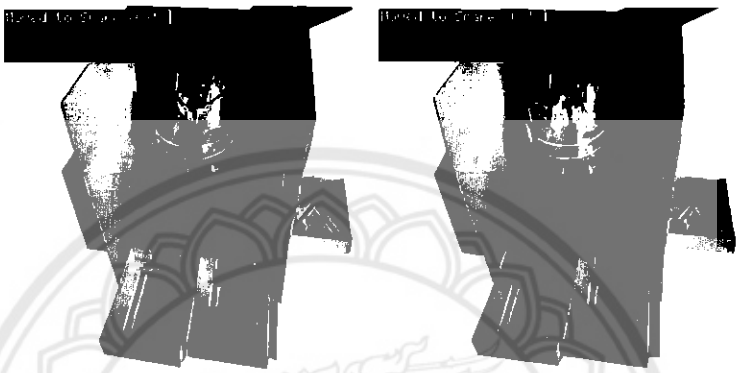
เมื่อตรวจพบว่าลูกปิงปองลูกต่อมามีสีเขียวซึ่งไม่สอดคล้องกับเงื่อนไขสำหรับกล่อง A เพราะมีจำนวนลูกปิงปองสีเขียวครบตามที่ผู้ใช้ต้องการแล้ว แต่สอดคล้องกับเงื่อนไขสำหรับกล่อง B หน้าจอแอลซีดีจึงระบุในบรรทัดแรกว่าเป็นสีเขียวและจะส่งไปยังกล่อง B ดังแสดงในรูปที่ 4.11 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังรูปที่ 4.11 ข) จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง B ดังรูปที่ 4.11 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1



ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนซ้าย

รูปที่ 4.11 การคัดเลือกลูกปิงปองสีเขียวไปกล่อง B

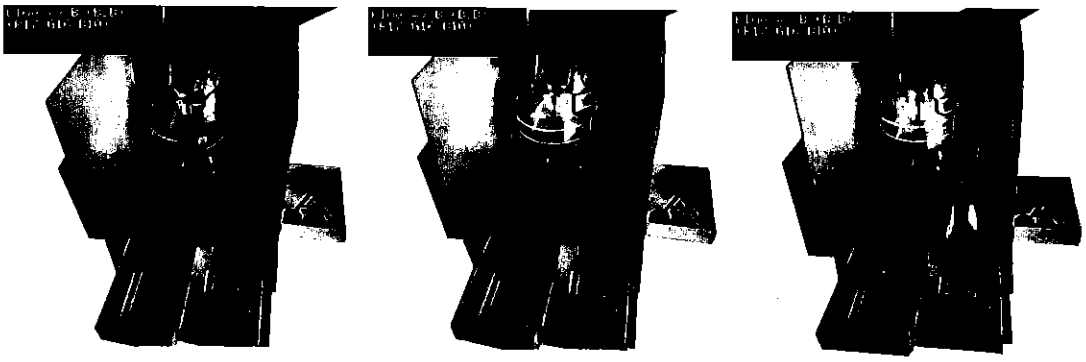
เมื่อตรวจพบว่าลูกปิงปองลูกต่อมามีสีแดงซึ่งไม่สอดคล้องกับเงื่อนไขสำหรับทั้งกล่อง A และ B เพราะมีจำนวนลูกปิงปองสีแดงครบตามที่ผู้ใช้ต้องการแล้วทั้งสองกล่อง หน้าจอแอลซีดีจึงระบุในบรรทัดแรกว่า 'Moved to Spare (C)!'ดังแสดงในรูปที่ 4.12 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองลงไปยังกล่อง C และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังรูปที่ 4.12 ข) จากนั้นจึงหมุนย้อนกลับไปตำแหน่งเดิมเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1



ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนซ้าย

รูปที่ 4.12 การคัดเลือกลูกปิงปองสีแดงไปกล่อง C

เมื่อตรวจพบว่าลูกปิงปองลูกต่อมามีสีน้ำเงินซึ่งไม่สอดคล้องกับเงื่อนไขสำหรับกล่อง A เพราะมีจำนวนลูกปิงปองสีน้ำเงินครบตามที่ผู้ใช้ต้องการแล้ว แต่สอดคล้องกับเงื่อนไขสำหรับกล่อง B หน้าจอแอลซีดีจึงระบุในบรรทัดทัดแรกว่าเป็นสีน้ำเงินและจะส่งไปยังกล่อง B ดังแสดงในรูปที่ 4.13 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางขวา 90° เพื่อส่งลูกปิงปองไปจานลำเลียงที่ 2 และปิดกั้นไม่ให้ลูกถัดไปไหลลงมาจากท่อลำเลียงดังรูปที่ 4.13 ข) จากนั้นมอเตอร์ตัวที่ 2 จึงหมุนจานลำเลียงที่ 2 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองให้ตกลงไปที่กล่อง B ดังรูปที่ 4.13 ค) จากนั้นมอเตอร์ตัวที่ 2 และตัวที่ 1 จึงหมุนย้อนกลับไปตำแหน่งเดิมตามลำดับเพื่อปล่อยให้ลูกปิงปองลูกถัดไปไหลลงสู่จานลำเลียงที่ 1



ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนขวา ค) มอเตอร์ตัวที่ 2 หมุนซ้าย

รูปที่ 4.13 การคัดเลือกลูกปิงปองสีน้ำเงินไปกล่อง B

หลังจากที่กล่อง A และ B มีลูกปิงปองครบตามเงื่อนไขที่ผู้ใช้ต้องการแล้ว หน้าจอจะระบุในบันทัดแรกว่า 'Job done! => [Reset]' เพื่อบอกให้ทราบว่ากระบวนการคัดเลือกเสร็จสิ้นลงแล้ว และให้กดปุ่มปรับตั้งค่าใหม่ (Reset) หากต้องการกลับสู่หน้าจอการเลือกแบบวิธี ส่วนบรรทัดที่ 2 และ 3 แสดงจำนวนลูกปิงปองแต่ละสีในกล่อง A และ B ตามลำดับ ในระหว่างนี้ถ้ายังมีลูกปิงปองไหลลงมาจากท่อลำเลียงอีก หน้าจอจะระบุในบรรทัดสุดท้ายว่า 'Moved to Spare (C)!'ดังรูปที่ 4.14 ก) นั่นคือมอเตอร์ตัวที่ 1 หมุนจานลำเลียงที่ 1 ไปทางซ้าย 90° เพื่อส่งลูกปิงปองลงไปยังกล่อง C ดังรูปที่ 4.14 ข) แล้วจึงหมุนย้อนกลับไปตำแหน่งเดิม

ก) ลูกปิงปองลงจานที่ 1 ข) มอเตอร์ตัวที่ 1 หมุนซ้าย

รูปที่ 4.14 การส่งลูกปิงปองสีเขียวไปกล่อง C หลังจากการคัดเลือกเสร็จสิ้น

ในกรณีที่ตรวจไม่พบลูกปิงปองเพิ่มเติมหลังจากกระบวนการคัดเลือกเสร็จสิ้น หน้าจอจะแสดงข้อความ 'No object detected!' ในบรรทัดล่างสุดดังแสดงในรูปที่ 4.15 ข) และไม่มีการทำงานของมอเตอร์ทั้งสอง

รูปที่ 4.15 การตรวจไม่พบลูกปิงปองหลังจากการคัดเลือกเสร็จสิ้น

4.2 การทดสอบหาเวลาที่ใช้ในการคัดแยกสีลูกปิงปองของเครื่อง

เนื่องจากเวลาที่เครื่องใช้ในการคัดแยกลูกปิงปองแต่ละสีนั้นขึ้นอยู่กับรูปแบบการทำงานของมอเตอร์ตัวที่ 1 และตัวที่ 2 ในการทดสอบแต่ละครั้งกำหนดให้ใส่ลูกปิงปองสีเดียวกันทั้งหมด 5 ลูก แล้วกดเลือกแบบวิธีคัดแยกสีเพื่อสั่งให้มอเตอร์ทำงานและจับเวลาในหน่วยวินาที (s) แล้วคำนวณหาระยะเวลาเฉลี่ยที่ใช้ต่อ 1 ลูก ซึ่งได้ทดสอบทั้งหมด 5 ครั้ง จากนั้นคำนวณหาระยะเวลาเฉลี่ยที่ใช้ต่อ 1 ลูกของแต่ละสีได้ผลการทดสอบดังตารางที่ 4.1 โดยพบว่าระยะเวลาเฉลี่ยในการคัดแยกลูกปิงปองสีเขียวมีค่าน้อยที่สุดเพราะใช้มอเตอร์ตัวที่ 1 เพียงตัวเดียวในการคัดแยก ในขณะที่การคัดแยกลูกปิงปองสีแดงและสีน้ำเงินใช้ระยะเวลาใกล้เคียงกันเพราะมีการทำงานของมอเตอร์ทั้งสองตัวในรูปแบบที่คล้ายกันโดยใช้เวลามากกว่าการคัดแยกลูกปิงปองสีเขียวประมาณ 2 s

ตารางที่ 4.1 ผลการทดสอบหาระยะเวลาในการคัดแยกลูกปิงปองแต่ละสี

ลูกปิงปอง	ระยะเวลาเฉลี่ยต่อ 1 ลูก (s)					ค่าเฉลี่ย (s)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
สีเขียว	4.388	4.404	4.420	4.404	4.404	4.404
สีแดง	6.404	6.420	6.418	6.388	6.388	6.404
สีน้ำเงิน	6.404	6.390	6.390	6.420	6.374	6.396

บทที่ 5
สรุปผลและข้อเสนอแนะ

จากการออกแบบเครื่องคัดแยกวัตถุตามสีและสร้างของแบบจำลอง รวมทั้งการทดสอบการทำงานของแบบจำลอง สามารถสรุปผลได้ดังนี้

5.1 สรุปผลการดำเนินงาน

ในโครงงานนี้ได้ออกแบบเครื่องคัดแยกวัตถุตามสีโดยใช้ตัวรับรู้ TCS230 ในการตรวจจับสีของวัตถุ และสร้างแบบจำลองเพื่อคัดแยกลูกปิงปองสีแดง สีเขียว และสีน้ำเงิน โดยใช้แอลดีอาร์ช่วยในการตรวจจับลูกปิงปอง ใช้ตัวรับรู้ TCS230 ร่วมกับแผงหลอดไฟแอลอีดีสีขาว 9 หลอดในการตรวจสอบสีของลูกปิงปอง ใช้มอเตอร์เซอร์โวจำนวน 2 ตัวขับจานลำเลียงเพื่อส่งลูกปิงปองลงกล่องซึ่งมี 3 กล่องคือกล่อง A B และ C โดยมีไมโครคอนโทรลเลอร์ Arduino MEGA 2560 รับสัญญาณจากส่วนตรวจจับและส่วนตรวจสอบสีของลูกปิงปองเพื่อประมวลผลสำหรับควบคุมมอเตอร์เซอร์โวทั้งสองรวมทั้งการแสดงสถานะการทำงานของเครื่องบนหน้าจอแอลซีดี ซึ่งเครื่องถูกออกแบบให้ทำงานได้ 2 แบบวิธีคือ แบบวิธีคัดแยกสีและแบบวิธีคัดเลือกสี

เมื่อผู้ใช้เลือกการทำงานในแบบวิธีคัดแยก เครื่องจะคัดแยกลูกปิงปองสีแดงลงกล่อง A ลูกปิงปองสีน้ำเงินลงกล่อง B และลูกปิงปองสีเขียวลงกล่อง C โดยหน้าจอแอลซีดีจะแสดงจำนวนรวมของลูกปิงปองแต่ละสีที่ตรวจพบ และในกรณีที่ผู้ใช้เลือกแบบวิธีคัดเลือก การทำงานจะเริ่มจากการให้ผู้ใช้ระบุจำนวนของลูกปิงปองแต่ละสีที่ต้องการสำหรับกล่อง A และ B ในขณะที่กล่อง C ถูกกำหนดเป็นกล่อง Spare เพื่อใส่ลูกปิงปองที่ไม่ตรงกับเงื่อนไขของผู้ใช้หรือเมื่อกล่อง A หรือ B มีลูกปิงปองครบตามเงื่อนไขแล้ว ในระหว่างที่กล่อง A และ B ยังมีจำนวนลูกปิงปองไม่ครบตามเงื่อนไขที่ระบุ ลูกปิงปองที่ตรวจพบว่าสอดคล้องกับเงื่อนไขของทั้งสองกล่องจะถูกคัดเลือกให้ไหลลงกล่อง A ก่อนเพื่อให้กล่อง A มีจำนวนครบตามเงื่อนไขก่อนกล่อง B เสมอ ผลการทดสอบแสดงให้เห็นว่าแบบจำลองของเครื่องคัดแยกวัตถุตามสีที่สร้างขึ้นสามารถทำงานได้ถูกต้องตามเงื่อนไขที่ออกแบบไว้สำหรับทั้งสองแบบวิธี

5.2 ปัญหาและแนวทางการแก้ไข

1) ความไม่สม่ำเสมอของสีบนผิวของวัตถุอาจทำให้ตัวรับรู้ TCS230 ตรวจสอบสีวัตถุผิดพลาดได้ ดังนั้นวัตถุที่นำมาคัดแยกจำเป็นต้องมีความสม่ำเสมอของสีในระดับสูง หรือปรับตั้งค่า

การอ่านสีของตัวรับรู้ในโปรแกรมให้สามารถตีความค่าของแต่ละสีได้ในช่วงกว้างขึ้น หรือเขียนโปรแกรมให้คัดแยกวัตถุที่มีสีไม่สม่ำเสมอออกจากกลุ่ม

2) ระดับความเข้มของสีของวัตถุมีผลต่อการตรวจสอบสีของตัวรับรู้ TCS230 ส่งผลให้เกิดความผิดพลาดในการอ่านสีของวัตถุที่มีสีเดียวกันแต่มีระดับความเข้มของสีต่างกันได้ ในกรณีดังกล่าวจำเป็นต้องปรับตั้งค่าการอ่านสีของตัวรับรู้ในโปรแกรมให้สามารถแยกแยะสีของวัตถุที่มีระดับความเข้มของสีต่างกันได้ละเอียดมากขึ้น

3) การตรวจจับลูกปิงปองที่ไหลลงมาจากท่อลำเลียงในโครงงานนี้ใช้แอลดีอาร์ที่ติดตั้งไว้ด้านล่างของจานลำเลียงนอกจากนี้ยังได้ติดตั้งแผงหลอดแอลอีดีสีขาวล้อมรอบตัวรับรู้ TC230 เพื่อให้ตรวจสอบสีของลูกปิงปองได้แม่นยำมากขึ้น แต่เนื่องจากจานลำเลียงทำมาจากอะคริลิกใส แสงจากแผงหลอดแอลอีดีดังกล่าวรวมทั้งแสงจากภายนอกแบบจำลองจึงอาจทำให้การตรวจจับลูกปิงปองโดยใช้แอลดีอาร์ผิดพลาดได้ในกรณีที่ลูกปิงปองไหลลงมา ดังนั้นจึงจำเป็นการปรับตั้งการหน่วงเวลารับค่าสัญญาณที่ได้จากแอลดีอาร์ให้สัมพันธ์กับค่าความเข้มแสงที่แอลดีอาร์ได้รับ หรือเปลี่ยนใช้ตัวรับรู้ชนิดอื่น เช่น ตัวรับรู้แบบใช้แสง

5.3 แนวทางในการพัฒนาต่อไป

1) การพัฒนาโปรแกรมเพื่อให้ตัวรับรู้สีสามารถแยกแยะสีของวัตถุที่มีระดับความเข้มของสีแตกต่างกันได้ รวมทั้งเพิ่มจำนวนสีที่สามารถแยกแยะได้

2) การพัฒนาโครงสร้างของชิ้นงานและโปรแกรมเพื่อประยุกต์ใช้งานจริงในการคัดแยกสินค้าหรือผลิตภัณฑ์ที่บกพร่อง เช่น คัดแยกเมล็ดข้าวโพดที่เสียออกจากเมล็ดข้าวโพดที่ดี

เอกสารอ้างอิง

[1] John Muir Laws, "RGB Reflection and Absorption", สืบค้นเมื่อวันที่ 26 ตุลาคม 2558 จาก:www.inventorartist.com/primary-colors/

[2] พิบูลย์ ดิษฐอุดม. (2535). การออกแบบระบบแสงสว่าง (Illumination Engineering). กรุงเทพฯ: พิมพ์ที่ หจก.เอช-เอน การพิมพ์.

[3] ศุลี บรรจงจิตร. (2544). วิศวกรรมการส่องสว่าง (Illumination Engineering). กรุงเทพฯ: พิมพ์ที่บริษัท เอช. เอน. กรุ๊ป จำกัด.

[4] Arduino MEGA Board "Arduino MEGA 2560 & Genuino MEGA 2560", สืบค้นเมื่อวันที่ 27 ตุลาคม 2558 จาก http:// arduino.cc/en/Main/ArduinoBoardMega2560

[5] บริษัท วีนัสซัพพลาย จำกัด, "การใช้งาน Character LCD Display กับ Arduino", สืบค้นเมื่อวันที่ 27 ตุลาคม 2558 จาก http://thaieasyelec.com/article-wiki/review-product-article

[6] วรรณยมล กันภัย (tdhobby), "Rc Servo ความรู้พื้นฐานในการใช้งานเซอร์โว", สืบค้นเมื่อวันที่ 27 ตุลาคม 2558 จาก http://www.tdhobby.com

[7] Arduitronics, "Extreme RFID Learning Kit for Arduino II", สืบค้นเมื่อวันที่ 27 ตุลาคม 2558 จาก http://www.arduitronics.com

[8] http://www.mwit.ac.th, สืบค้นเมื่อ 2 พฤษภาคม 2559.

ภาคผนวก ก
รหัสต้นฉบับของโปรแกรมควบคุมเครื่องคัดแยกวัตถุตามสี ด้วยตัวรับรู้ TCS230

```
#include <Keypad.h> //การเรียกใช้งานแผงแป้นตัวเลข
#include <Servo.h> //การเรียกใช้งานมอเตอร์เซอร์โว
#include <LiquidCrystal.h> //การเรียกใช้งานจอแสดงผล
#define servo_pinA 12 //กำหนดพอร์ตของมอเตอร์เซอร์โวตัวที่ 1 และประกาศตัแปร
#define servo_pinB 13 //กำหนดพอร์ตของมอเตอร์เซอร์โวตัวที่ 2 และประกาศตัวแปร
Servo myServo1;
Servo myServo2;
const byte ROWS = 4; //กำหนดจำนวนแถวของแผงแป้นตัวเลข
const byte COLS = 4; //กำหนดจำนวนหลักของแผงแป้นตัวเลข
char keys[ROWS][COLS] = //กำหนดค่าตามแผงแป้นตัวเลข
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {22,23,24,25}; //กำหนดพอร์ตแถวของแผงแป้นตัวเลข
byte colPins[COLS] = {26,27,28,29}; //กำหนดพอร์ตหลักของแผงแป้นตัวเลข
LiquidCrystal lcd(53, 52, 51, 50, 49, 48); //กำหนดพอร์ตขาจอแอลซีดี RS, E, D4, D5, D6, D7
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
const int outputEnabled = 2; // write LOW to turn on Note, may not be hooked up.
const int s0 = 3; //กำหนดพอร์ตของตัวรับรู้สี
const int s1 = 4;
const int s2 = 5;
const int s3 = 6;
const int nLED = 7; //กำหนดพอร์ตขาหลอดแอลอีดีของตัวรับรู้สี
const int out = 8; //กำหนดพอร์ตเอาท์พุตของตัวรับรู้สี
int red = 0; //ประกาศตัวแปร red เป็น int และมีค่าเริ่มต้นเท่ากับ 0
int green = 0; //ประกาศตัวแปร green เป็น int และมีค่าเริ่มต้นเท่ากับ 0
int blue = 0; //ประกาศตัวแปร blue เป็น int และมีค่าเริ่มต้นเท่ากับ 0
int value1 = 0; //ประกาศตัวแปร value1 เป็น int และมีค่าเริ่มต้นเท่ากับ 0
int sensor1 = 11; //กำหนดพอร์ตขาของแอลดีอาร์
```

```
int x; //ประกาศตัวแปร x เป็น int
int red_A=0,green_A=0,blue_A=0; //ประกาศตัวแปร red_A, green_A, และ blue_A มีค่าเริ่มต้น 0
int red_B=0,green_B=0,blue_B=0; //ประกาศตัวแปร red_B, green_B, และ blue_B มีค่าเริ่มต้น 0
int R_A_S=0,G_A_S=0,B_A_S=0; //ประกาศตัวแปร R_A_S, G_A_S, และ B_A_S มีค่าเริ่มต้น 0
int R_B_S=0,G_B_S=0,B_B_S=0; //ประกาศตัวแปร R_B_S, G_B_S, และ B_B_S มีค่าเริ่มต้น 0
int R_s=0,G_s=0,B_s=0; //ประกาศตัวแปร R_S, G_S, และ B_S โดยมีค่าเริ่มต้นเป็น 0
void setup()
{
 myServo1.attach(servo_pinA);
 myServo2.attach(servo_pinB);
 Serial.begin(9600);
 pinMode(outputEnabled, OUTPUT); //กำหนดตัวแปรที่รับค่ามาจากตัวรับรู้ให้เป็นเอาท์พุต
 pinMode(s0, OUTPUT);
 pinMode(s1, OUTPUT);
 pinMode(s2, OUTPUT);
 pinMode(s3, OUTPUT);
 pinMode(nLED, OUTPUT);
 pinMode(out, INPUT);
 digitalWrite(outputEnabled, LOW); //เขียนขาoutputEnabledให้มีค่าLOW
 digitalWrite(s0, HIGH); //เขียนขา s0 ให้มีค่าHIGH
 digitalWrite(s1, HIGH); //เขียนขา s1 ให้มีค่าHIGH
 digitalWrite(nLED, LOW); //เขียนขา nLED ให้มีค่าLOW
 lcd.begin(20, 4); //กำหนดขนาดหน้าจอ 20 ตัวอักษร 4 บรรทัด
}
void loop()
{
 lcd.setCursor(0, 0); //ตำแหน่งตัวอักษรที่แสดงบนหน้าจอตอน
 lcd.print("Separation mode: [8]");
 lcd.setCursor(0, 1);
 lcd.print("Selection mode : [9]");
 while(1)
 {
```

```
char key = keypad.getKey(); //กำหนดตัวแปรที่รับค่ามาจากแผงแป้นตัวเลข
if(key=='9') //เงื่อนไขเมื่อรับค่าจากแผงแป้นตัวเลขเป็นเลข 9 "เข้าสู่แบบวิธีคัดเลือก"
 { //แสดงหน้าจอ "Selection mode..." "R=red,G=green,B=blue" ค้างไว้ 5 วินาที
delay(300);
 lcd.clear();
 lcd.setCursor(2, 1);
 lcd.print("Selection mode...");
 lcd.setCursor(0, 2);
 lcd.print("R=red,G=green,B=blue");
 delay(5000);
 lcd.clear();
 FunctionNum(); //รับค่าจากแผงแป้นตัวเลขที่ผู้ใช้ได้กำหนดในแบบวิธีคัดเลือก
 lcd.clear(); //คำสั่งยกเลิกหน้าจอเพื่อเปลี่ยนไปเป็นการแสดงผลใหม่
 while(1)
 {
  value1 = digitalRead(sensor1); //อ่านค่าที่ส่งมาจากตัวรับรู้แอลดีอาร์
  if (value1 == 0) ) //แอลดีอาร์ถูกบังแสงเมื่อมีลูกปิงปอง
  {
    lcd.clear();
 color(); //ให้ฟังก์ชั่น s2,s3เลือกทำงานส่งข้อมูลให้ไมโครคอนโทรลเลอร์ทำงาน
 lcd.setCursor(0, 1);
 lcd.print("(R");
 lcd.setCursor(2, 1);
 lcd.print(red, DEC);
 lcd.setCursor(4, 1);
 lcd.print(" G");
 lcd.setCursor(6, 1);
 lcd.print(green, DEC);
 lcd.setCursor(8, 1);
 lcd.print(" B");
 lcd.setCursor(10, 1);
 lcd.print(blue, DEC);
```

```
lcd.setCursor(12, 1);
lcd.print(")");
Compare(); //ฟังก์ชั่นที่นำค่าตัวเลขมาเปรียบเทียบเพื่อพิจารณาว่าเป็นสีอะไร
if(x==1&&red_A>0) //พิจารณาเงื่อนไขของสีแดง (เมื่อ x=1) ที่อยู่ภายในกล่อง A
{
 lcd.setCursor(0, 0);
 lcd.print("Red => A (R_A)");
 Box_A();
 red_A--;
}
else if(x==1&&red_B>0) //พิจารณาเงื่อนไขของสีแดง (เมื่อ x=1) ที่อยู่ภายในกล่อง B
{
 lcd.setCursor(0, 0);
 lcd.print("Red => B (R_B)");
 Box_B();
 red_B--;
}
else if(x==2&&blue_A>0) //พิจารณาเงื่อนไขของสีน้ำเงิน (เมื่อ x=2) ที่อยู่ภายในกล่อง A
{
 lcd.setCursor(0, 0);
 lcd.print("Blue => A (B_A)");
 Box_A();
 blue_A--;
}
else if(x==2&&blue_B>0) //พิจารณาเงื่อนไขของสีน้ำเงิน (เมื่อ x=2) ที่อยู่ภายในกล่อง B
{
 lcd.setCursor(0, 0);
 lcd.print("Blue => B (B_B)");
 Box_B();
 blue_B--;
}
else if(x==3&&green_A>0) //พิจารณาเงื่อนไขของสีเขียว (เมื่อ x=3) ที่อยู่ภายในกล่อง A
```

```
{
 lcd.setCursor(0, 0);
 lcd.print("Green => A (G_A)");
 Box_A();
 green_A--;
}
else if(x==3&&green_B>0) //พิจารณาเงื่อนไขของสีเขียว (เมื่อ x=3) ที่อยู่ภายในกล่อง B
{
 lcd.setCursor(0, 0);
 lcd.print("Green => B (G_B)");
 Box_B();
 green_B--;
}
else Spare();
}
else if (value1 == 1) //แอลดีอาร์ได้รับแสงเมื่อไม่มีลูกปิงปอง
 {
  if (red_A==0&&green_A==0&&blue_A==0&&red_B==0&&green_B==0&&blue_B==0)
//ตรวจสอบเงื่อนไขของแต่ละสีในกล่อง A และ B กรณีทำตามเงื่อนไขครบแล้ว
    { //นำค่าที่รับมาจากผู้ใช้ให้แสดงบนหน้าจอแอลซีดีเมื่อทำตามเงื่อนไขครบแล้ว
      lcd.clear();
      lcd.setCursor(2, 1);
      lcd.print("R_A= ");
      lcd.setCursor(6, 1);
      lcd.print(R_A_S);
      lcd.setCursor(8, 1);
      lcd.print("G_A= ");
      lcd.setCursor(12, 1);
      lcd.print(G_A_S);
      lcd.setCursor(14, 1);
      lcd.print("B_A= ");
      lcd.setCursor(18, 1);
```

```
        lcd.print(B_A_S);
        lcd.setCursor(2, 2);
        lcd.print("R_B= ");
        lcd.setCursor(6, 2);
        lcd.print(R_B_S);
        lcd.setCursor(8, 2);
        lcd.print("G_B= ");
        lcd.setCursor(12, 2);
        lcd.print(G_B_S);
        lcd.setCursor(14, 2);
        lcd.print("B_B= ");
        lcd.setCursor(18, 2);
        lcd.print(B_B_S);
        lcd.setCursor(0, 0);
        lcd.print("Job done! => [Reset]");
        lcd.setCursor(0, 3);
        lcd.print("No object detected!"); //พิมพ์คำว่า"No object detected!"ในบรรทัดที่ 4
      }
     else //ตรวจสอบเงื่อนไขของแต่ละสีในกล่อง A และ B กรณียังไม่ครบตามเงื่อนไข
  {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("No object detected!");
        lcd.setCursor(0, 1);
        lcd.print("Add more ping-pongs!");
      }
  }
   delay(3000);
}
}
else if(key=='8') //เงื่อนไขเมื่อรับค่าจากแผงแป้นตัวเลขเป็นเลข 8 "เข้าสู่แบบวิธีคัดเลือก"
 {
```

```
 delay(300);
   lcd.clear();
   lcd.setCursor(2, 1);
   lcd.print("Seperation mode...");
delay(2000);
 while(1)
 {
 value1 = digitalRead(sensor1);
  if (value1 == 0) //แอลดีอาร์ถูกบังแสงเมื่อมีลูกปิงปอง
  {
   lcd.clear();
   color(); //ให้ฟังก์ชั่น s2,s3เลือกทำงานส่งข้อมูลให้ไมโครคอนโทรลเลอร์ทำงาน
   lcd.setCursor(0, 1);
   lcd.print("(R");
   lcd.setCursor(2, 1);
   lcd.print(red, DEC);
   lcd.setCursor(4, 1);
   lcd.print(" G");
   lcd.setCursor(6, 1);
   lcd.print(green, DEC);
   lcd.setCursor(8, 1);
   lcd.print(" B");
   lcd.setCursor(10, 1);
   lcd.print(blue, DEC);
   lcd.setCursor(12, 1);
   lcd.print(")");
if (red < blue && red < green) //พิจารณาเงื่อนไขของสีแดง
 {
      lcd.setCursor(0, 0);
      lcd.print("Red => A");
      R_s++;  //เพิ่มค่าตัวแปร R_s ทีละ 1
      lcd.setCursor(0, 2);
```

```
        lcd.print("Sum of Red = ");
        lcd.setCursor(13, 2);
        lcd.print(R_s);
        Box_A(); //สั่งให้มอเตอร์ตัวที่ 1 และมอเตอร์ตัวที่ 2 ทำงานเพื่อส่งลูกปิงปองไปกล่อง A
    }
    else if (blue < red && blue < green) //พิจารณาเงื่อนไขของสีน้ำเงิน
    {
        lcd.setCursor(0, 0);
        lcd.print("Blue => B");
        B_s++;  //เพิ่มค่าตัวแปร B_s ทีละ 1
        lcd.setCursor(0, 2);
        lcd.print("Sum of Blue = ");
        lcd.setCursor(14, 2);
        lcd.print(B_s);
        Box_B(); //สั่งให้มอเตอร์ตัวที่ 1 และมอเตอร์ตัวที่ 2 ทำงานเพื่อส่งลูกปิงปองไปกล่อง B
    }
    else //พิจารณาเงื่อนไขของสีเขียว
    {
        lcd.setCursor(0, 0);
        lcd.print("Green => C");
        G_s++;  //เพิ่มค่าตัวแปร G_s ทีละ 1
        lcd.setCursor(0, 2);
        lcd.print("Sum of Green = ");
        lcd.setCursor(15, 2);
        lcd.print(G_s);
        Box_C(); //สั่งให้มอเตอร์ตัวที่ 1 ทำงานเพื่อส่งลูกปิงปองไปกล่อง C
    }
     }
      else if (value1 == 1) //แอลดีอาร์ได้รับแสงเมื่อไม่มีลูกปิงปอง
      { //แสดงคำว่า "No object detected!" พร้อมทั้งสรุปจำนวนรวมที่ได้ทั้งหมด
        lcd.clear();
        lcd.setCursor(0, 0);
```

```
      lcd.print("No object detected!");
      lcd.setCursor(0, 2);
      lcd.print("Sum: R=");
      lcd.setCursor(7, 2);
      lcd.print(R_s);
      lcd.setCursor(9, 2);
      lcd.print("G=");
      lcd.setCursor(11, 2);
      lcd.print(G_s);
      lcd.setCursor(13, 2);
      lcd.print("B=");
      lcd.setCursor(15, 2);
      lcd.print(B_s);
      lcd.setCursor(0, 3);
      lcd.print("To restart: [Reset]");
    }
    delay(3000);
   }
  }

}
}
void color() //ประกาศฟังก์ชัน color
{
  digitalWrite(nLED,1);
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); //ความถี่ลูกคลื่นกรณีสีแดง
  digitalWrite(s3, HIGH);
  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); //ความถี่ลูกคลื่นกรณีสีน้ำเงิน
  digitalWrite(s2, HIGH);
  green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); //ความถี่ลูกคลื่นกรณีสีเขียว
```

```
  digitalWrite(nLED,1);
}
 void Compare() //ประกาศฟังก์ชัน Compare
{
 if (red < blue && red < green) //สร้างเงื่อนไขเปลี่ยนค่าตัวแปร x เพื่อนำไปใช้ในแบบวิธีคัดเลือก
  {
  x=1;
  }
  else if (blue < red && blue < green)
 {
  x=2;
 }
  else
 {
   x=3;
 }
}
void Spare() //ประกาศฟังก์ชัน Spare
{
  if (red_A==0&&green_A==0&&blue_A==0&&red_B==0&&green_B==0&&blue_B==0)
//ตรวจสอบเงื่อนไขของแต่ละสีในกล่อง A และ B กรณีทำตามเงื่อนไขครบแล้ว
   { //นำค่าที่รับมาจากผู้ใช้ให้แสดงบนหน้าจอแอลซีดีเมื่อทำตามเงื่อนไขครบแล้ว
       lcd.clear();
       lcd.setCursor(2, 1);
       lcd.print("R_A= ");
       lcd.setCursor(6, 1);
       lcd.print(R_A_S);
       lcd.setCursor(8, 1);
       lcd.print("G_A= ");
       lcd.setCursor(12, 1);
       lcd.print(G_A_S);
       lcd.setCursor(14, 1);
```

```
            lcd.print("B_A= ");
            lcd.setCursor(18, 1);
            lcd.print(B_A_S);
            lcd.setCursor(2, 2);
            lcd.print("R_B= ");
            lcd.setCursor(6, 2);
            lcd.print(R_B_S);
            lcd.setCursor(8, 2);
            lcd.print("G_B= ");
            lcd.setCursor(12, 2);
            lcd.print(G_B_S);
            lcd.setCursor(14, 2);
            lcd.print("B_B= ");
            lcd.setCursor(18, 2);
            lcd.print(B_B_S);
            lcd.setCursor(0, 0);
            lcd.print("Job done! => [Reset]");
            lcd.setCursor(0, 3);
            lcd.print("Moved to Spare (C)!");  //แสดงคำว่า "Moved to Spare"ในบรรทัดที่ 4
            Box_C(); //มอเตอร์หมุนและจานลำเลียงส่งลูกปิงปองไปยังกล่อง C
        }
        else //ตรวจสอบเงื่อนไขของแต่ละสีในกล่อง A และ B กรณียังไม่ครบตามเงื่อนไข
          {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Moved to Spare (C)!"); //พิมพ์คำว่า  "Moved to Spare (C)!"  ในบรรทัดที่ 1
            Box_C(); //มอเตอร์หมุนและจานลำเลียงส่งลูกปิงปองไปยังกล่อง C
          }
}
void FunctionNum() //ประกาศฟังก์ชัน FunctionNum เพื่อรับค่าจากผู้ใช้งาน
{
 Start:
```

```
lcd.clear();
lcd.setCursor(0, 2);
lcd.print("Select box: [A]or[B]");
lcd.setCursor(2, 0);
lcd.print("R_A= ");
lcd.setCursor(6, 0);
lcd.print(red_A);
lcd.setCursor(8, 0);
lcd.print("G_A= ");
lcd.setCursor(12, 0);
lcd.print(green_A);
lcd.setCursor(14, 0);
lcd.print("B_A= ");
lcd.setCursor(18, 0);
lcd.print(blue_A);
lcd.setCursor(2, 1);
lcd.print("R_B= ");
lcd.setCursor(6, 1);
lcd.print(red_B);
lcd.setCursor(8, 1);
lcd.print("G_B= ");
lcd.setCursor(12, 1);
lcd.print(green_B);
lcd.setCursor(14, 1);
lcd.print("B_B= ");
lcd.setCursor(18, 1);
lcd.print(blue_B);
lcd.setCursor(0, 3);
lcd.print("To start, press [D]");
while(1)
{
 char key = keypad.getKey(); //ประกาศตัวแปร key โดยให้รับค่ามาจากแผงแป้นตัวเลข
```

```
if(key=='D') return; //ถ้าผู้ใช้กด [D] โปรแกรมเริ่มทำงานในแบบวิธีคัดเลือกสี
if(key=='#') goto Start; //ถ้าผู้ใช้กด [#] โปรแกรมแสดงค่าที่ผู้ใช้เลือก
if(key=='A') //กรณีผู้ใช้กด [A]
{ //แสดงจำนวนของสีแดงที่ผู้ใช้เลือกในกล่อง A
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Enter R_A no. [0-5]");
 lcd.setCursor(0, 1);
 lcd.print("then press [*]");
 while(1)
  {
   char key = keypad.getKey();
   if(key=='*') goto selectGreen; //ผู้ใช้กด [*] แล้วโปรแกรมจะให้เลือกสีเขียวต่อไป
   if(key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5')  //กำหนดให้ผู้ใช้กดเลข 1-5
   {
    red_A = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร red_A
    R_A_S = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร R_A_S
    lcd.setCursor(5, 2);
    lcd.print("R_A =");
    lcd.setCursor(10, 2);
    lcd.print(red_A);
   }
  }
selectGreen: //ประกาศฟังก์ชัน selectGreen
lcd.clear(); //แสดงจำนวนของสีเขียวที่ผู้ใช้เลือกในกล่อง A
lcd.setCursor(0, 0);
lcd.print("Enter G_A no. [0-5]");
lcd.setCursor(0, 1);
lcd.print("then press [*]");
 while(1)
  {
   char key = keypad.getKey();
```

```
    if(key=='*') goto selectBlue; //ผู้ใช้กด [*] แล้วโปรแกรมจะให้เลือกสีน้ำเงินต่อไป
    if(key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5')
    {
     green_A = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร green_A
     G_A_S = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร G_A_S
     lcd.setCursor(5, 2);
     lcd.print("G_A =");
     lcd.setCursor(10, 2);
     lcd.print(green_A);
    }
   }
selectBlue: //ประกาศฟังก์ชัน selectBlue
lcd.clear(); //แสดงจำนวนของสีน้ำเงินที่ผู้ใช้เลือกในกล่อง A
lcd.setCursor(0, 0);
lcd.print("Enter B_A no. [0-5]");
lcd.setCursor(0, 1);
lcd.print("then press [*]");
 while(1)
   {
    char key = keypad.getKey();
    if(key=='*') goto A; //ผู้ใช้กด [*] แล้วโปรแกรมแสดงผลที่ฟังก์ชัน A
    if(key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5')
    {
     blue_A = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร blue_A
     B_A_S = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร B_A_S
      lcd.setCursor(5, 2);
      lcd.print("B_A =");
      lcd.setCursor(10, 2);
      lcd.print(blue_A);
    }
   }
  A: //ประกาศฟังก์ชัน A เพื่อแสดงผลรวมทั้งสามสีที่ผู้ใช้เลือกในกล่อง A
```

```
    lcd.clear();
    lcd.setCursor(5, 0);
    lcd.print("R_A =");
    lcd.setCursor(10, 0);
    lcd.print(red_A);
    lcd.setCursor(5, 1);
    lcd.print("G_A =");
    lcd.setCursor(10, 1);
    lcd.print(green_A);
    lcd.setCursor(5, 2);
    lcd.print("B_A =");
    lcd.setCursor(10, 2);
    lcd.print(blue_A);
    lcd.setCursor(0, 3);
    lcd.print("To Summary, press[#]");
    }
if(key=='B') //กรณีผู้ใช้กด [B]
 { //แสดงจำนวนของสีแดงที่ผู้ใช้เลือกในกล่อง B
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enter R_B no. [0-5]");
    lcd.setCursor(0, 1);
    lcd.print("then press [*]");
    while(1)
     {
      char key = keypad.getKey();
      if(key=='*') goto selectGreen_B; //ผู้ใช้กด [*] แล้วโปรแกรมจะให้เลือกสีเขียวต่อไป
      if(key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5')
      {
       red_B = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร red_B
       R_B_S = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร R_B_S
       lcd.setCursor(5, 2);
```

```
      lcd.print("R_B =");
      lcd.setCursor(10, 2);
      lcd.print(red_B);
     }
    }
selectGreen_B: //ประกาศฟังก์ชัน selectGreen_B
lcd.clear(); //แสดงจำนวนของสีเขียวที่ผู้ใช้เลือกในกล่อง B
lcd.setCursor(0, 0);
lcd.print("Enter G_B no. [0-5]");
lcd.setCursor(0, 1);
lcd.print("then press [*]");
 while(1)
   {
    char key = keypad.getKey();
    if(key=='*') goto selectBlue_B; //ผู้ใช้กด [*] แล้วโปรแกรมจะให้เลือกสีน้ำเงินต่อไป
    if(key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5')
    {
     green_B = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร green_B
     G_B_S = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร G_B_S
     lcd.setCursor(5, 2);
     lcd.print("G_B =");
     lcd.setCursor(10, 2);
     lcd.print(green_B);
    }
   }
selectBlue_B: //ประกาศฟังก์ชัน selectBlue_B
lcd.clear(); //แสดงจำนวนของสีน้ำเงินที่ผู้ใช้เลือกในกล่อง B
lcd.setCursor(0, 0);
lcd.print("Enter B_B no. [0-5]");
lcd.setCursor(0, 1);
lcd.print("then press [*]");
 while(1)
```

```
    {
     char key = keypad.getKey();
     if(key=='*') goto B; //ผู้ใช้กด [*] แล้วโปรแกรมแสดงผลที่ฟังก์ชัน B
     if(key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5')
     {
      blue_B = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร blue_B
      B_B_S = atoi(&key); //รับค่าจากแผงแป้นตัวเลขประกาศเป็นตัวแปร B_B_S
      lcd.setCursor(5, 2);
      lcd.print("B_B =");
      lcd.setCursor(10, 2);
      lcd.print(blue_B);
     }
    }
 B: //ประกาศฟังก์ชัน B เพื่อแสดงผลรวมทั้งสามสีที่ผู้ใช้เลือกในกล่อง B
   lcd.clear();
   lcd.setCursor(5, 0);
   lcd.print("R_B =");
   lcd.setCursor(10, 0);
   lcd.print(red_B);
   lcd.setCursor(5, 1);
   lcd.print("G_B =");
   lcd.setCursor(10, 1);
   lcd.print(green_B);
   lcd.setCursor(5, 2);
   lcd.print("B_B =");
   lcd.setCursor(10, 2);
   lcd.print(blue_B);
   lcd.setCursor(0, 3);
   lcd.print("To Summary, press[#]");
    }
  }
}
```

```
void Box_A()  //ประกาศคำสั่ง Box_A
{
 myServo1.write(0); //มอเตอร์ตัวที่ 1 หมุนขวา 90°
 delay(1000);
 myServo2.write(0); //มอเตอร์ตัวที่ 2 หมุนขวา 90°
 delay(1000);
 myServo2.write(95); //มอเตอร์ตัวที่ 2 หมุนซ้าย 90°
 delay(1000);
 myServo1.write(85); //มอเตอร์ตัวที่ 1 หมุนซ้าย 90°
 delay(1000);
 }
 void Box_B() //ประกาศคำสั่ง Box_B
{
 myServo1.write(0); //มอเตอร์ตัวที่ 1 หมุนขวา 90°
 delay(1000);
 myServo2.write(180); //มอเตอร์ตัวที่ 2 หมุนซ้าย 90°
 delay(1000);
 myServo2.write(90); //มอเตอร์ตัวที่ 2 หมุนขวา 90°
 delay(1000);
 myServo1.write(85); //มอเตอร์ตัวที่ 1 หมุนซ้าย 90°
 delay(1000);
 }
 void Box_C()
{
 myServo1.write(180); //มอเตอร์ตัวที่ 1 หมุนซ้าย 90°
 delay(1000);
 myServo1.write(78); //มอเตอร์ตัวที่ 1 หมุนขวา 90°
 delay(1000);
 }
```

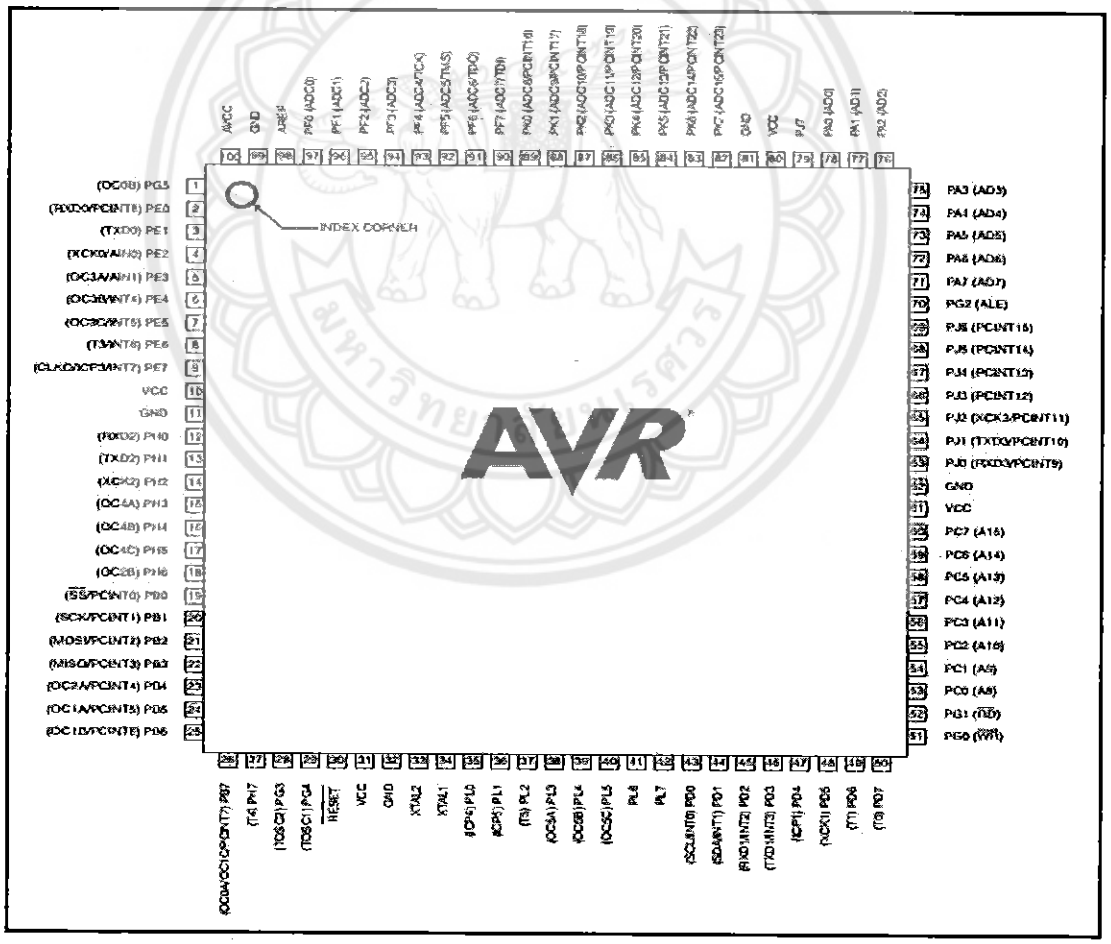
ภาคผนวก ข

รายละเอียดข้อมูลของ ATmega2560

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 64K/128K/256KBytes of In-System Self-Programmable Flash
  - 4Kbytes EEPROM
  - 8Kbytes Internal SRAM
  - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/ 100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
    - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix® acquisition
  - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
  - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change

- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
  - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
  - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
  - RoHS/Fully Green
- Temperature Range:
  - -40°C to 85°C Industrial
- Ultra-Low Power Consumption
  - Active Mode: 1MHz, 1.8V: 500µA
  - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
  - ATmega640V/ATmega1280V/ATmega1281V:
    - 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega2560V/ATmega2561V:
    - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega640/ATmega1280/ATmega1281:
    - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
  - ATmega2560/ATmega2561:
    - 0 - 16MHz @ 4.5V - 5.5V



ภาคผนวก ค

รายละเอียดข้อมูลของตัวรับรู้ TCS230

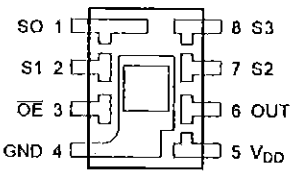
TAOS TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS

TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046 – JANUARY 2003

- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm/ C Temperature Coefficient
- Low-Profile Surface-Mount Package

SOIC PACKAGE
(TOP VIEW)

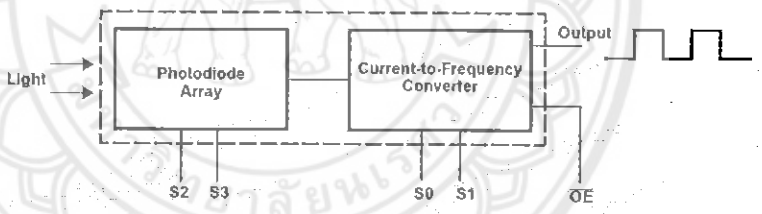


Description

The TCS230 programmable color light-to-frequency converter combines configurable silicon photodiodes and a current-to-frequency converter on single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable ($\overline{OE}$) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

The light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All 16 photodiodes of the same color are connected in parallel and which type of photodiode the device uses during operation is pin-selectable. Photodiodes are 120 µm x 120 µm in size and are on 144-µm centers.

Functional Block Diagram



Terminal Functions

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
GND	4		Power supply ground. All voltages are referenced to GND.
$\overline{OE}$	3	I	Enable for f_o (active low).
OUT	6	O	Output frequency (f_o).
S0, S1	1, 2	I	Output frequency scaling selection inputs.
S2, S3	7, 8	I	Photodiode type selection inputs.
V_{DD}	5		Supply voltage

Table 1. Selectable Options

S0	S1	OUTPUT FREQUENCY SCALING (f_o)		S2	S3	PHOTODIODE TYPE
L	L	Power down		L	L	Red
L	H	2%		L	H	Blue
H	L	20%		H	L	Clear (no filter)
H	H	100%		H	H	Green

Electrical Characteristics at T_A = 25°C, V_{DD} = 5 V (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = –4 mA	4	4.5		V
V_{OL}	Low-level output voltage	I_{OL} = 4 mA		0.25	0.40	V
I_{IH}	High-level input current				5	µA
I_{IL}	Low-level input current				5	µA
I_{DD}	Supply current	Power-on mode		2	3	mA
		Power-down mode		7	15	µA
	Full-scale frequency (See Note 2)	S0 = H, S1 = H	500	600		kHz
		S0 = H, S1 = L	100	120		kHz
		S0 = L, S1 = H	10	12		kHz
	Temperature coefficient of output frequency	$\lambda \le 700$ nm, $-25°C \le T_A \le 70°C$		±200		ppm/°C
k_{SVS}	Supply voltage sensitivity	V_{DD} = 5 V ±10%		±0.5		%/V

NOTE 2. Full-scale frequency is the maximum operating frequency of the device without saturation.

Operating Characteristics at V_{DD} = 5 V, T_A = 25°C, S0 = H, S1 = H (unless otherwise noted) (See Notes 3, 4, 5, 6, and 7).

PARAMETER		TEST CONDITIONS	CLEAR PHOTODIODE S2 = H, S3 = L			BLUE PHOTODIODE S2 = L, S3 = H			GREEN PHOTODIODE S2 = H, S3 = H			RED PHOTODIODE S2 = L, S3 = L			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_O	Output frequency	E_e = 45.6 µW/cm², λ_p = 470 nm	16	20	24	11.2	16.4	21.6							kHz
		E_e = 39.2 µW/cm², λ_p = 524 nm	16	20	24				8	13.6	19.2				kHz
		E_e = 32.8 µW/cm², λ_p = 635 nm	16	20	24							14	19	24	kHz
		E_e = 0		2	12		2	12		2	12		2	12	Hz
R_e	Irradiance responsivity (Note 8)	λ_p = 470 nm		439			360			88			31		Hz/ (µW/ cm²)
		λ_p = 524 nm		510			189			347			46		
		λ_p = 565 nm		548			49			318			110		
		λ_p = 635 nm		610			30			37			579		
	Saturation Irradiance (Note 9)	λ_p = 470 nm		1370			1670								µW/ cm²
		λ_p = 524 nm		1180						1730					
		λ_p = 565 nm		1090						1890					
		λ_p = 635 nm		980									1040		
R_v	Illuminance responsivity (Note 10)	λ_p = 470 nm		585			480			117			41		Hz/ lx
		λ_p = 524 nm		98			36			67			9		
		λ_p = 565 nm		92			8			53			18		
		λ_p = 635 nm		407			20			25			386		
	Nonlinearity (Note 11)	f_O = 0 to 5 kHz		±0.1%			±0.1%			±0.1%			±0.1%		% F.S.
		f_O = 0 to 50 kHz		±0.2%			±0.2%			±0.2%			±0.2%		% F.S.
		f_O = 0 to 500 kHz		±0.5%			±0.5%			±0.5%			±0.5%		% F.S.
	Recovery from power down			100			100			100			100		µs
	Response time to output enable ($\overline{OE}$)			100			100			100			100		ns

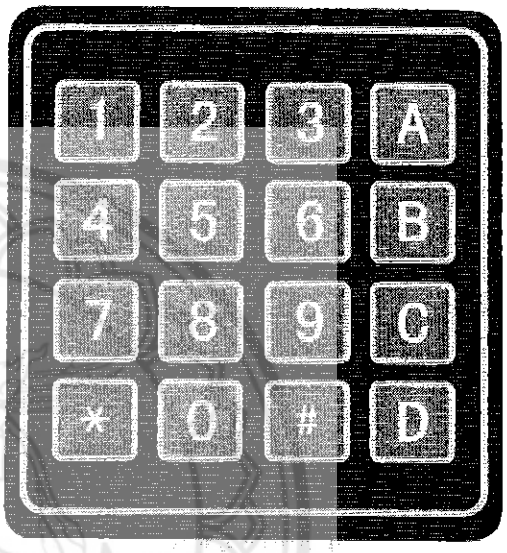
NOTES: 3. Optical measurements are made using small-angle incident radiation from a light-emitting diode (LED) optical source.
4. The 470 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength λ_p = 470 nm, spectral halfwidth $\Delta\lambda_{1/2}$ = 35 nm, and luminous efficacy = 75 lm/W.
5. The 524 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength λ_p = 524 nm, spectral halfwidth $\Delta\lambda_{1/2}$ = 47 nm, and luminous efficacy = 520 lm/W.
6. The 565 nm input irradiance is supplied by a GaP light-emitting diode with the following characteristics: peak wavelength λ_p = 565 nm, spectral halfwidth $\Delta\lambda_{1/2}$ = 28 nm, and luminous efficacy = 595 lm/W.
7. The 635 nm input irradiance is supplied by a AlInGaP light-emitting diode with the following characteristics: peak wavelength λ_p = 635 nm, spectral halfwidth $\Delta\lambda_{1/2}$ = 17 nm, and luminous efficacy = 150 lm/W.
8. Irradiance responsivity R_e is characterized over the range from zero to 5 kHz.
9. Saturation irradiance = (full-scale frequency)/(irradiance responsivity).
10. Illuminance responsivity Rv is calculated from the irradiance responsivity by using the LED luminous efficacy values stated in notes 4, 5, and 6 and using 1 lx = 1 lm/m².
11. Nonlinearity is defined as the deviation of f_O from a straight line between zero and full scale, expressed as a percent of full scale.

ภาคผนวก ง
รายละเอียดข้อมูลของแผงแป้นตัวเลข

PARALLAX INC
Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com
Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

4x4 Matrix Membrane Keypad (#27899)

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.



Features

- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

Key Specifications

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions:
  Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
  Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

Application Ideas

- Security systems
- Menu selection
- Data entry for embedded systems

How it Works

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in Figure 1.

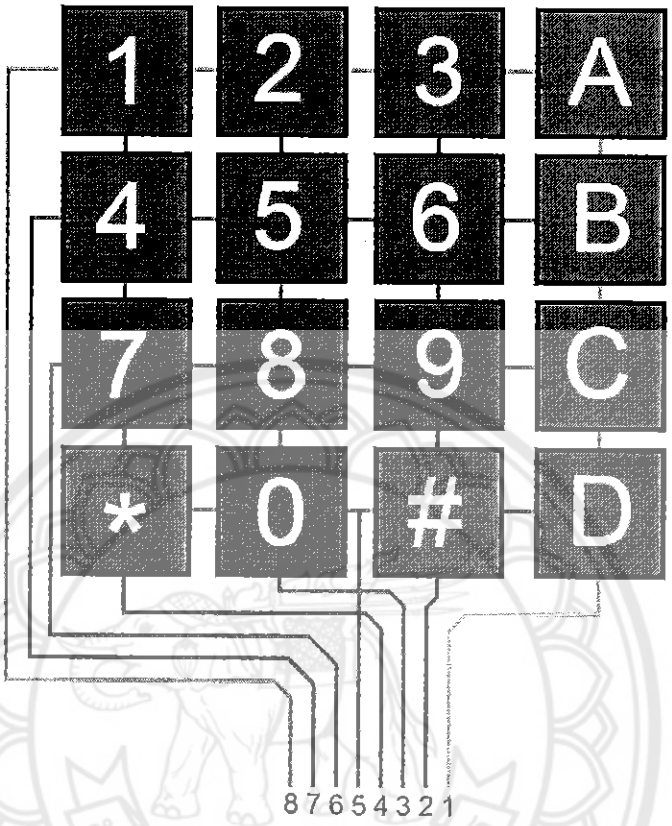


Figure 1: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed.

For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1 high. This means that a contact has been made between column 4 and row 1, so button 'A' has been pressed.

Connection Diagrams

Figure 2

For use with the BASIC Stamp example program listed below.

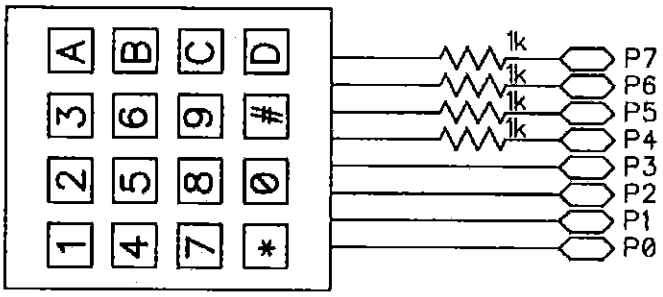
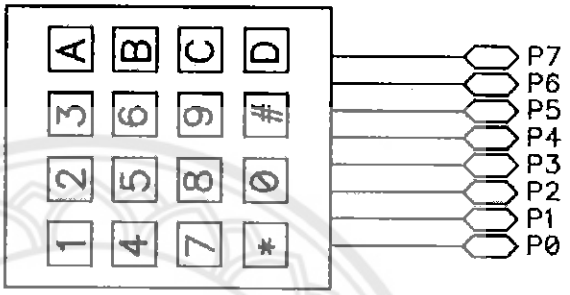


Figure 3

For use with the Propeller P8X32A example program listed below.

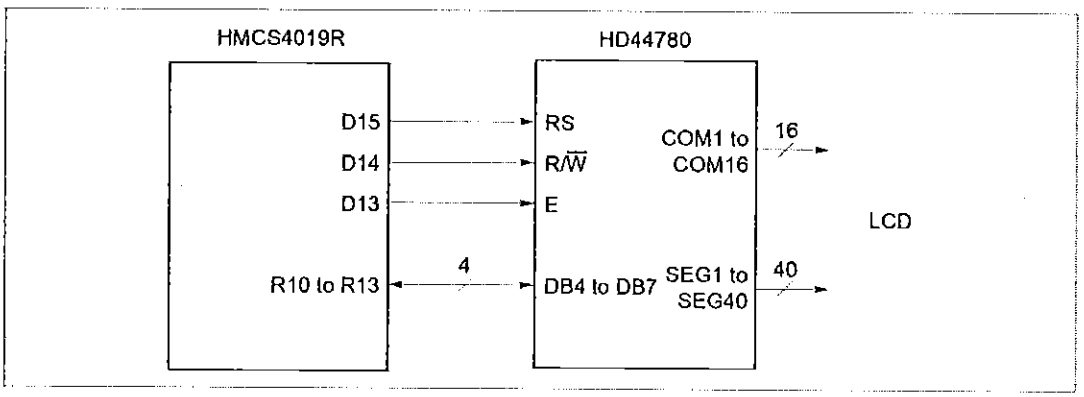


ภาคผนวก จ

รายละเอียดข้อมูลของจอแสดงผล

Pin Functions

Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/$\overline{W}$	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
SEG1 to SEG40	40	O	LCD	Segment signals
V1 to V5	5	—	Power supply	Power supply for LCD drive V_{cc} –V5 = 11 V (max)
V_{cc}, GND	2	—	Power supply	V_{cc}: 2.7V to 5.5V, GND: 0V
OSC1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.



Function Description

Registers

The HD44780U has two 8-bit registers, an instruction register (IR) and a data register (DR).

The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/$\overline{W}$ = 1 (Table 1), the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)

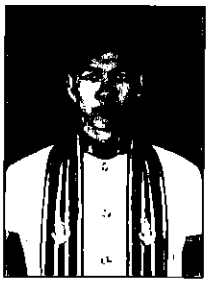
The address counter (AC) assigns addresses to both DDRAM and CGRAM. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when RS = 0 and R/$\overline{W}$ = 1 (Table 1).

Table 1 Register Selection

RS	R/$\overline{W}$	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

ประวัติผู้ดำเนินโครงงาน



ชื่อ นายกิตติทัต แซ่เฮง
ภูมิลำเนา 109 หมู่ 2 ต.บ้านโป่ง อ.เวียงป่าเป้า จ.เชียงราย

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนนุชนาถอนุสรณ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **kittitats55@email.nu.ac.th**

ชื่อ นายสุวินัย น้อยวงศ์
ภูมิลำเนา 73/6 หมู่ 3 ถนนจรดวิถีถ่อง ต.คลองตาล อ.ศรีสำโรง จ.สุโขทัย

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนสวรรค์อนันต์วิทยา
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **suwinain55@email.nu.ac.th**