

ปริญญาโทปีการศึกษา 2549

ภาควิชาวิศวกรรมเกษตร

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องคัดขนาดและสีถั่วระอู๋ปุ่นเพื่อการส่งออกโดยวิธีการประมวลผลภาพ

Green Soybean for Export Size and Color Grader Using Image Processing Technique

ผู้จัดทำ

1. โปรดเกล้า เทศลงทอง รหัสประจำตัว 46010469
2. พัชรีย์ คำหอม รหัสประจำตัว 46010509
3. เพ็ญธิดา คูหาแสนห์ รหัสประจำตัว 46010548



..... อาจารย์ที่ปรึกษา
(รศ.ดร.ปานมนัส ศิริสมบุญ)

..... อาจารย์ที่ปรึกษา
(ผศ.ดร.ธนารัตน์ ชลิตาพงศ์)

..... อาจารย์ที่ปรึกษา
(ผศ. พรสุข รติโรจน์อนันต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคัดขนาดและสีถั่วระอู๋ปุ่นเพื่อการส่งออกโดยวิธีการประมวลผลภาพ

โปรดเกล้า เทศลงทอง	46010469
พัชรี คำหอม	46010509
เพ็ญธิดา กุหาเสน่ห์	46010548
รศ.ดร.ปานมนัส ศิริสมบุรณ์	อาจารย์ที่ปรึกษา
ผศ.ดร. ธนาวัฒน์ ชลิตาพงศ์	อาจารย์ที่ปรึกษา
ผศ. พรสุข รติโรจน์อนันต์	อาจารย์ที่ปรึกษา
ปีการศึกษา 2549	

บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อศึกษาวิธีประยุกต์ใช้เทคนิคประมวลผลภาพในการคัดขนาดและคุณภาพสี เพื่อออกแบบและสร้างเครื่องคัดขนาดและคุณภาพถั่วระอู๋ปุ่นเพื่อการส่งออกโดยวิธีการประมวลผลภาพในระดับศึกษาเบื้องต้น โดยส่วนแรกใช้โปรแกรม Visual Basic C/C++ เวอร์ชัน 6.0 ร่วมกับ OpenCV คัดขนาดและคุณภาพสีของถั่วระอู๋ปุ่น โปรแกรม Mat lab เวอร์ชัน 6 คำนวณค่าสี Hue เฉลี่ยและค่าเบี่ยงเบนมาตรฐานของสีถั่ว ส่วนที่สองคือส่วนการออกแบบและสร้างเครื่องคัดขนาดและคุณภาพสีของถั่วระอู๋ปุ่น โดยการถ่ายภาพถั่วระอู๋ปุ่นแล้ววิเคราะห์ผลโดยใช้โปรแกรมข้างต้นแล้วส่งสัญญาณให้ชุดไมโครคอนโทรลเลอร์ส่งชุดหัวพ่นลมทำงานตรงตามตำแหน่งที่ต้องการ ซึ่งประสิทธิภาพของโปรแกรมคัดแยกถูกต้อง 100 เปอร์เซ็นต์ ในการทดสอบสมรรถนะของเครื่องคัดขนาดและคุณภาพสีของถั่วระอู๋ปุ่นพบว่ากลไกทำงานต่าง ๆ ประสานการทำงานได้ มีความสามารถในการคัดแยกได้ 11,482 ฝักต่อชั่วโมง โดยป้อนถั่วระอู๋ปุ่นในลักษณะตามแนวสายพานมีความเร็ว 0.63 เมตร/วินาที หน้ากว้างในการป้อนถั่วระอู๋ปุ่น 28 เซนติเมตร จำนวน 1 แถว ประกอบด้วย 17 ฝัก ถั่วที่มีขนาดฝักเล็กและขนาดกลางควรป้อนในตามแนวการเคลื่อนที่สายพานเพื่อให้มีประสิทธิภาพการคัดแยกสูงสุดคือ คัดถูกต้อง 90 และ 87.5 เปอร์เซ็นต์ ตามลำดับ ส่วนขนาดใหญ่ควรวางในลักษณะขวางการเคลื่อนที่ของสายพานเพื่อให้มีประสิทธิภาพการคัดแยกสูงสุดคือ คัดถูกต้อง 75 เปอร์เซ็นต์

**GREEN SOYBEAN FOR EXPORT SIZE AND COLOR GRADER USING IMAGE
PROCESSING TECHNIQUE**

Prodklao Teslongthong 46010469

Phatcharee Khumhom 46010509

Penthida Koohasaneh 46010548

Assoc. Prof. Dr. Panmanas Sirisomboon Advisor

Assist. Prof. Dr. Thanarat Chalidabhongse Advisor

Assist. Prof. Phornsuk Ratiroch-Anant Advisor

2006

Abstract

The purposes of this project were to apply the image processing techniques in green soybean grading by size and color and to preliminary design and fabricate the grader of the green soybean for export using image processing. The program for grading developed by Visual Basic C/C++ version 6.0 and OpenCV was used. Another program developed using Mat lab version 6 was used for calculating mean of hue and standard deviation of green soybean color. The grader fabricated initially captured green soybean pods on belt images, sent to compile using the grading program that had efficiency 100 percentages and then sent signal to microcontroller for control nozzle to shoot green soybeans which had unacceptable size and color. From performance testing, the systems worked fairly well and had capacity of 11,482 pods/hour by feeding the pods on the belt which had the velocity of 0.63 m/s and the width of feeding of 28 centimeters and the pattern of the pods was 17 pods in one row. It was indicated from the test that small and medium size green soybeans should be fed in the direction that their length parallel to the direction of belt movement for the maximum grading efficiency, 90 and 87.5 percentages, respectively. The large size of green soybeans should be fed transverse to the direction of belt movement to obtain the maximum grading efficiency of 75 percentages.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ไม่อาจจะสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือให้คำแนะนำจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญอย่างยิ่งที่ทำให้ปริญญาบัตรนี้เสร็จสมบูรณ์ได้ ขอกราบขอบพระคุณ รศ.ดร.ปานมนัส ติริสมบูรณ์, ผศ. ดร. ธนารัตน์ ชลิตาพงศ์ และ ผศ. พรสุข รติโรจน์-อนันต์ ที่ช่วยแนะแนวทาง ให้คำปรึกษาตลอดจนดูแลเอาใจใส่ ตั้งแต่ต้นจนเสร็จสิ้นปริญญาบัตร ซึ่งเป็นส่วนสำคัญอย่างยิ่งในการทำปริญญาบัตรในครั้งนี้

ขอขอบพระคุณ บริษัท Interroll (Thailand) จำกัด ที่ให้ความอนุเคราะห์มอบระบบชุดสายพานลำเลียง จำนวน 1 ชุด

ขอขอบคุณ เพื่อนๆ พี่น้อง เจ้าหน้าที่ และคณาจารย์ทุกท่าน ในภาควิชาวิศวกรรมเกษตร ที่ช่วยเหลือและให้กำลังใจเสมอมา

ขอขอบพระคุณ นายคานววัฒน์ แสงผล คณะเทคโนโลยีสารสนเทศ ที่คอยช่วยเหลือให้คำแนะนำในเรื่องของการประมวลผลภาพ (Image Processing)

ขอขอบคุณ นายเอกศักดิ์ สุวรรณโณ และนายมนัส วงษ์ลา ภาควิชาโทรคมนาคม ที่ช่วยเหลือและให้คำปรึกษาในส่วนวงจรการควบคุมสัญญาณ (Microcontroller)

ขอขอบพระคุณ นายกฤษณ์ ผลโพธิ์ (นายช่าง) ที่คอยช่วยเหลือให้คำแนะนำสร้างเครื่องคัดขนาดและสีของถั่วแระญี่ปุ่นเพื่อการส่งออก

ข้าพเจ้าขอกราบขอบพระคุณอย่างสุดซึ้ง แค่ คุณพ่อ คุณแม่ ของข้าพเจ้า ซึ่งเป็นบุคคลที่สำคัญที่สุดของชีวิตของข้าพเจ้า ทำให้ข้าพเจ้าได้มีวันนี้ ที่เลี้ยงข้าพเจ้าจนเติบโตใหญ่ดูแลเอาใจใส่และเป็นที่กำลังใจแก่ข้าพเจ้าเป็นอย่างดีที่สุด พร้อมทั้งให้โอกาสในการศึกษาแก่ข้าพเจ้าอย่างเต็มที่ตลอดมา ข้าพเจ้าจะขอระลึกพระคุณอันสุดประมาณและขอกราบขอบพระคุณ ณ ที่นี้

ขอคุณพระศรีรัตนตรัยจงช่วย คลบับดาล ปกป้องรักษาให้ทุกท่านที่ข้าพเจ้าเอ๋ยถึงมีสุขภาพและพลานามัยที่แข็งแรงและมีความสุข ตลอดไป

ขอขอบคุณ

นายโปรดเกล้า เทศลงทอง

นางสาวพัชรี คำหอม

นางสาวเพ็ญธิดา คูหาสนนท์

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญภาพ	ช
สารบัญตาราง	ฉ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของการศึกษา	2
1.4 ผลที่คาดว่าจะได้รับ	2
1.5 วิธีการดำเนินการ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 การประมวลผลภาพ	4
2.1.1 ระบบของสี	4
1. ระบบสีแบบ RGB (Red/Green/Blue)	4
2. ระบบสีแบบ HSI (Hue/Saturation/Intensity)	4
2.1.2 การจำแนกภาพ (Image Segmentation)	4
2.1.3 มอร์โฟโลยีทางคณิตศาสตร์ (Mathematical Mophology)	5
1. อีรชัน (Erosion)	5
2. ไดเลชัน (Dilation)	5
3. โอเพนนิ่ง (Opening)	6
4. โคลสซิง (Closiing)	6
2.1.4 การวิเคราะห์องค์ประกอบที่เชื่อมต่อกันของภาพ (Connected Components)	7
2.2 ไมโครคอนโทรลเลอร์	8
2.2.1 คุณลักษณะพื้นฐานของ MCS-51	8
2.2.2 ลักษณะการจัดขาของ MCS-51	9
2.2.3 พอร์ทอินพุตและพอร์ทเอาต์พุต	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

1. การใช้งานพอร์ทเป็นอินพุท	12
2. การใช้งานพอร์ทเป็นเอาต์พุท	13
2.2.4 หน่วยความจำโปรแกรมของ MCS-51	13
2.2.5 หน่วยความจำข้อมูลของ MCS-51	14
1. หน่วยความจำข้อมูลภายใน	15
2. หน่วยความจำขนาด 128 ไบต์แรก	15
3. หน่วยความจำขนาด 128 ไบต์ถัดไป	15
2.2.6 รีจิสเตอร์หน้าที่พิเศษ	16
1. แอควิวมูลเตอร์ (Accumulator)	16
2. รีจิสเตอร์ B	17
3. โปรแกรมเคาน์เตอร์ (Program Counter)	17
4. สแตคพอยน์เตอร์ (Stack Pointer)	17
5. ตัวชี้ข้อมูลหรือค่าพอยน์เตอร์ (Data Pointer)	17
6. โปรแกรมสแตตัสเวิร์ด (PSW)	17
7. รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ท (Port Register)	17
8. รีจิสเตอร์ SBUF	17
9. รีจิสเตอร์ PCON	17
10. รีจิสเตอร์ IP, IE, TMOD, SCON	17
2.2.7 การใช้งานพอร์ทสื่อสารอนุกรมแบบ Single Processor	18
1. พอร์ทสื่อสารอนุกรม	18
2.3 งานวิจัยที่เกี่ยวข้อง	20
บทที่ 3 การออกแบบและการสร้าง	23
3.1 แผนผังขั้นตอนการทำงาน	23
3.2 ขั้นตอนการเขียนโปรแกรม	24
3.2.1 ตัวอย่างภาพถั่วระอูญี่ปุ่นจากกล้องถ่ายภาพ CCD progressive scan	25
3.2.2 การแยกส่วนของถั่วระอูญี่ปุ่นออกจากพื้นหลัง (Segmentation)	25
1. การสร้างแบบจำลองค่า Hue ของถั่วระอูญี่ปุ่น	25
2. การหาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน	26
3. การแยกส่วนของถั่วระอูญี่ปุ่นออกจากพื้นหลัง	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

3.2.3	การกำจัดสัญญาณรบกวน	27
3.2.4	การหาเซตคอร์ดของถั่วระเฉีปน	28
3.2.5	การหาขนาดของถั่วระเฉีปน	29
3.2.6	การคำนวณหาจุด Centroid ของฝักถั่ว	30
3.2.7	การคำนวณตำแหน่งหัว nozzle และเวลาในการยิงของหัว nozzle	31
3.2.8	ลักษณะของโปรแกรม GBA	31
3.3	วงจรการควบคุมสัญญาณ	32
3.3.1	วงจรส่งค่าผ่าน serial port	33
3.3.2	วงจรถับสัญญาณ	34
3.3.3	วงจรถบคุมภาคส่งสัญญาณและเชื่อมต่อกับอุปกรณ์พ่นลม	34
3.4	การออกแบบอุปกรณ์และการสร้างเครื่องคัดขนาดและคุณภาพถั่วระเฉีปน	37
3.4.1	อุปกรณ์ที่ใช้ในการสร้างเครื่องคัดแยกถั่วระเฉีปน	37
	1. ชุดอุปกรณ์ระบบสายพานลำเลียง (Belt conveyer)	37
	2. ชุดอุปกรณ์ระบบนิวแมติกส์	38
	3. ชุดอุปกรณ์ระบบการประมวลผลภาพ	42
3.4.2	การติดตั้งอุปกรณ์	43
	1. การติดตั้งอุปกรณ์ในส่วนของชุดอุปกรณ์สายพานลำเลียง	43
	2. การติดตั้งอุปกรณ์ในส่วนของระบบนิวแมติก	44
	3. การติดตั้งอุปกรณ์ในส่วนของระบบการประมวลผลภาพ	44
3.4.3	การคำนวณจุดถ่ายเทวัสดุออกจากหัวสายพาน	45
3.4.4	การคำนวณแรงดันลมที่จะใช้ในการคัดแยกถั่วระเฉีปน	47
3.4.5	การคำนวณอัตราเร็วในการถ่ายรูป	48
บทที่ 4	การทดสอบและผลการทดสอบ	50
4.1	สีของถั่วระเฉีปน	50
4.2	ขนาดของถั่วระเฉีปน	51
4.3	สมรรถนะของเครื่องคัด	53
	4.3.1 การทำงาน	53
	4.3.2 ความสามารถ	53
	4.3.3 ประสิทธิภาพ	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
บทที่ 5 การสรุปผลการทดสอบ	55
5.1 บทสรุป	56
5.2 วิจารณ์	56
ภาคผนวก	58
ก ส่วนของโปรแกรม GBA	59
ข ส่วนของโปรแกรม camera	84
ค ส่วนของโปรแกรม Finding Color Bean	89
ง ส่วนของโปรแกรม ควบคุม Microcontroller	90
จ ตารางแสดงผลการทำงานของ โปรแกรม GBA	82
ฉ ตารางพื้นที่ภาพถ่ายของถั่วแต่ละขนาด	103
เอกสารอ้างอิง	105

สารบัญภาพ

รูปที่	หน้าที่
1.1 ถั่วระเฉี้ปุ่น	1
2.1 การทำอิรอรุ่น	5
2.2 การทำไคเลรุ่น	6
2.3 การทำโอเพนนิ่ง	6
2.4 การทำโคลสซิ่ง	7
2.5 องค์ประกอบของภาพ	7
2.6 การระบุหมายเลขให้กับองค์ประกอบของภาพโดยพิจารณาใน 4 ทิศทาง	8
2.7 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89xx	9
2.8 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ของ ATMEL	10
2.9 การใช้ไมโครคอนโทรลเลอร์เป็นอินพุตและเอาต์พุตพอร์ท	12
2.10 การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51	14
2.11 ลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล	15
2.12 ตำแหน่งของหน่วยความจำแบบไบต์และแบบบิท	16
2.13 Timing Diagram การส่งข้อมูล โหมด 0	19
2.14 Timing Diagram การส่งข้อมูล โหมด 1	19
2.15 Timing Diagram การส่งข้อมูล โหมด 2	20
2.16 Timing Diagram การส่งข้อมูล โหมด 3	20
3.1 แผนผังขั้นตอนการทำงานของเครื่องคัดขนาดและคุณภาพสีของถั่วระเฉี้ปุ่น	23
3.2 Flow chart การทำงานของโปรแกรมหลัก	24
3.3 ภาพถั่วระเฉี้ปุ่นบนสายพานสีขาว	25
3.4 ฮีสโตแกรมค่า Hue รูปถั่วระเฉี้ปุ่นกับพื้นหลังสีขาว มี 2 รูป รูปที่ 1 เป็นค่า Hue ของถั่วระเฉี้ปุ่น ส่วนรูปที่ 2 เป็นรูปของพื้นหลัง	26
3.5 ภาพถั่วระเฉี้ปุ่นที่ตัดบางส่วนเพื่อนำมาวิเคราะห์สี	26
3.6 ภาพของถั่วระเฉี้ปุ่นที่ถูกแยกออกจากพื้นหลัง	27
3.7 รูปภาพที่ถูกกำจัดสัญญาณรบกวนแล้ว โดยใช้วิธี Opening	27
3.8 (a) ถั่วระเฉี้ปุ่นที่มีหลายสีปะปนกันอยู่ (b) ถั่วเหลืองที่มีสี Hue น้อยกว่าค่าสีเรด โซลด์ถูกคัดออก	28
3.9 ลักษณะแกนของสายพาน มีแกนตามแนวการเคลื่อนที่ของสายพาน (T-cor) และแกนขวางการเคลื่อนที่ของสายพาน (N-cor)	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่	หน้าที่
3.10 (a) คือรูปภาพถั่วระเฉี้ปุ่นในแนวแกน N-coordinate	30
(b) คือรูปภาพถั่วระเฉี้ปุ่นในแนวแกน T-coordinate	30
3.11 จำลองภาพถั่วรูปถั่วมีสเกลเป็นพิกเซล	31
3.12 ลักษณะของโปรแกรม GBA	32
3.13 วงจรการควบคุมสัญญาณ	32
3.14 (a) วงจรส่งค่าผ่าน Serial port	33
(b) สาย Serial Port (RS232)	33
3.15 อุปกรณ์ Microcontroller	34
3.16 (a) หม้อแปลง	35
(b) วงจรควบคุมภาคส่งสัญญาณ	35
3.17 วงจรควบคุมภาคส่งสัญญาณและเชื่อมต่อกับอุปกรณ์พ่นลม	35
3.18 วงจรการควบคุมสัญญาณรวม	36
3.19 (a) แบบชุดสายพานลำเลียง	37
(b) ชุดสายพานลำเลียง	37
3.20 อุปกรณ์ปรับความเร็วรอบ	38
3.21 หัวพ่นลม	38
3.22 ข้อต่อ 3 ทาง	39
3.23 อุปกรณ์ต่อตรง	39
3.24 Reducing valve	40
3.25 สายท่อลม	40
3.26 ชุดหัวพ่นลม	40
3.27 Power supply	41
3.28 เครื่องอัดอากาศ	41
3.29 Regulator	41
3.30 กล้องถ่ายภาพ CCD progressive scan	42
3.31 Light Box	43
3.32 ชุดอุปกรณ์สายพานลำเลียง	43
3.33 ชุดอุปกรณ์ระบบนิวแมติกส์	44
3.34 ชุดอุปกรณ์ประมวลผลภาพ	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่		หน้าที่
3.35	กราฟแสดงลักษณะการตกของตัวระฆังปุ่น	46
3.36	ลักษณะการเคลื่อนที่ของวัสดุบนสายพาน	47
3.37	การออกแบบทิศทางการยิงของตัวระฆังปุ่นเมื่อถูกยิง	48



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้าที่
2.1 การเลือกโหมคการทำงานในการรับส่งข้อมูล	18
3.1 ค่าที่รับตามตำแหน่งหัวพ่นลมที่ต้องการ	33
3.2 ตำแหน่งขาที่ส่งสัญญาณตามตำแหน่งหัวพ่นลมที่ต้องการ	36
3.3 ระยะตามแนวแกน X และระยะตามแนวแกน Y ที่ตำแหน่งต่างๆ	46
4.1 ค่าสี Hue เฉลี่ยของถั่วระอู๋ปุ่นทั้ง 30 ฝัก	50
4.2 ขนาดของถั่วระอู๋ปุ่น โดยใช้โปรแกรมทั้ง 30 ฝัก	52
4.3 ประสิทธิภาพในการคัดแยกสูงสุด	54
4.4 ประสิทธิภาพในการคัดแยกต่ำสุด	54
จ. 1 ผลการคัดสีของถั่วระอู๋ปุ่นจาก โปรแกรม GBA	92
จ. 2 ผลการคัดขนาดของถั่วระอู๋ปุ่นจาก โปรแกรม GBA	95
จ. 3 ผลการคัดสีและคัดขนาดของถั่วระอู๋ปุ่นจาก โปรแกรม GBA	98
ฉ. 1 พื้นที่ภาพฉายของถั่วแต่ละขนาด	103

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากถั่วแระญี่ปุ่นดังแสดงในรูปที่ 1.1 เป็นพืชที่ให้ผลตอบแทนสูง และเป็นแหล่งโปรตีนราคา ถูกเมื่อเทียบกับโปรตีนจากเนื้อสัตว์มีไขมันสูง มีวิตามิน เอ บี และซี มีแร่ธาตุที่ร่างกายต้องการ เช่น เหล็ก แคลเซียม ฟอสฟอรัส มี Isoflavones ซึ่งเป็นสารช่วยลดความเสี่ยงต่อการเป็นโรคหัวใจ โรคมะเร็ง เต้านม มะเร็งต่อมลูกหมากและลดอาการวัยทอง



รูปที่ 1.1 ถั่วแระญี่ปุ่น

แต่สิ่งที่เป็นปัญหา คือ การส่งออกถั่วแระญี่ปุ่นเป็นการแข่งขันเพื่อบุกเบิกตลาดนานาชาติ ซึ่งประเทศคู่แข่ง ได้แก่ ไต้หวัน และจีน ซึ่งส่งออกถั่วแระญี่ปุ่นไปยังประเทศญี่ปุ่นถึงปีละประมาณ 40,000-55,000 ตัน [1] เนื่องจากถั่วแระญี่ปุ่นเป็นสินค้าส่งออกประเภทผัก ซึ่งมีความเข้มงวดในแง่คุณภาพ มีมาตรฐานเฉพาะ เช่น ขนาด สี กลิ่น รส และจะต้องปลอดโรคและแมลง ตลอดจนสารพิษตกค้าง และสารปนเปื้อนอื่นๆ ประเทศไทยจำเป็นต้องพัฒนาเทคโนโลยีเพื่อให้มีส่วนแบ่งตลาดเพิ่มขึ้น

การตัดฝักจำเป็นต้องทำอย่างประณีต และใช้เวลานาน ปัจจุบันการตัดแยกฝักเริ่มจากในไร่เป็นการตัดอย่างหยาบเพื่อตัดฝักที่ไม่ต้องการออก และเมื่อมาถึงโรงงานจะทำการคัดขนาดโดยใช้เครื่องคัดแบบตะแกรงเขย่า ซึ่งฝักดิบบางส่วนจะถูกแยกออกไป แต่เครื่องคัดแบบนี้ยังมีปัญหาอยู่มากเนื่องจากมีฝักถั่วอุดรูตะแกรงจำนวนมากทำให้การคัดขนาดมีประสิทธิภาพต่ำ หลังจากนั้นจะมีการคัดคุณภาพก่อนที่จะบรรจุและเก็บรักษาก่อนที่จะส่งขาย การคัดคุณภาพนี้จะเป็นการคัดสี คัดฝักที่มีจุดค้างดำ ฝักบิดงอ หรือแม้กระทั่งฝักเล็กที่หลงคิดมา การคัดนี้ใช้คนคัดจำนวนมากโดยให้คนยืนเรียงคิๆ กันทั้งสองข้างของสายพานที่เคลื่อนตัวอย่างช้าๆ ในระยะทางยาวประมาณ 3 เมตร ซึ่งจะทำ 2 ครั้ง ทำให้นอกจากจะใช้คนมากยังคัดช้ามากด้วย การมีเครื่องคัดขนาดและคุณภาพดี ที่มีประสิทธิภาพจะช่วยให้ลดต้นทุนและเพิ่มผลผลิตอีกด้วย

1.2 วัตถุประสงค์

1. เพื่อศึกษาวิธีประยุกต์ใช้เทคนิคประมวลผลภาพในการคัดขนาดและสีตัวระบุเพื่อการส่งออก
2. เพื่อออกแบบและสร้างเครื่องคัดขนาดและคุณภาพตัวระบุเพื่อการส่งออกโดยวิธีการประมวลผลภาพในระดับศึกษาเบื้องต้น

1.3 ขอบเขตการศึกษา

เครื่องที่ออกแบบและสร้างจะทำการคัดแยกเฉพาะขนาดและสีเท่านั้น

1.4 ผลที่คาดว่าจะได้รับ

1. นักศึกษาได้รับการฝึก/มีประสบการณ์ตามความคาดหวัง
เนื่องจากนักศึกษามีความประสงค์ที่จะเข้าร่วมโครงการนี้ซึ่งจะทำให้มีความรู้และทักษะเกี่ยวกับการเขียน โปรแกรมคัดขนาดและคุณภาพตัวระบุ โปรแกรมเกี่ยวกับการควบคุมเครื่องคัดแยก และการออกแบบและสร้างเครื่องคัดขนาดและคุณภาพตัวระบุเพื่อการส่งออกโดยวิธีการประมวลผลภาพ
2. ได้ผลงานที่โรงงาน/บริษัทจะนำไปใช้ได้
เนื่องจากเครื่องคัดขนาดและคุณภาพตัวระบุเพื่อการส่งออกโดยวิธีการประมวลผลภาพในโครงการนี้เป็นการสร้างต้นแบบในระดับศึกษาเบื้องต้น จำเป็นต้องมีการพัฒนาต่อไปเพื่อให้ใช้ได้ในระดับอุตสาหกรรม ซึ่งจะต้องใช้งบประมาณที่สูงกว่านี้ แต่โครงการนี้จะเป็นจุดเริ่มต้นที่ดีในการสร้างต้นแบบในระดับศึกษาเบื้องต้นก่อนที่จะมีการพัฒนาต่อไป
3. ผลงานมีโอกาสขยายผลต่ออุตสาหกรรมทั้ง sectors
เนื่องจากเครื่องคัดขนาดและคุณภาพตัวระบุเพื่อการส่งออกโดยวิธีการประมวลผลภาพในโครงการนี้เป็นการสร้างต้นแบบในระดับศึกษาเบื้องต้นจึงมีโอกาขยายผลต่ออุตสาหกรรมทั้ง sectors นอกจากจะสามารถนำผลงานไปใช้กับการคัดแยกตัวระบุแล้วยังสามารถประยุกต์ใช้กับพืชผักอื่นที่มีลักษณะคล้ายคลึงกัน เช่น พริก กระจับเขียว ข้าวโพดฝักอ่อน เป็นต้นซึ่งเป็นผลผลิตเพื่อการส่งออกเช่นกัน ทำให้อุตสาหกรรมการส่งออกผักหรือแม้แต่ผลไม้ไทยมีเทคโนโลยีที่ทันสมัยมาใช้ ทำให้คุณภาพของผลผลิตดีขึ้นสามารถแข่งขันกับประเทศผู้ส่งออกรายอื่นได้และมีส่วนแบ่งตลาดมากขึ้น
4. สามารถพัฒนาผู้ประกอบการให้เข้าใจงานวิจัยมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 วิธีการดำเนินงาน

วิธีดำเนินการในการออกแบบเครื่องคัดขนาดและคุณภาพสีของถั่วระดูญี่ปุ่น ซึ่งในส่วนของขั้นตอนการดำเนินการได้แบ่งส่วนการทำงานออกเป็นสองส่วน คือการทำงานในส่วนของโปรแกรมคัดแยกถั่วระดูญี่ปุ่น และในส่วนของเครื่องคัดแยกถั่วระดูญี่ปุ่น ในส่วนของโปรแกรมคัดแยกถั่วนั้น เป็นขั้นตอนในการเขียนโปรแกรมคัดขนาดและคุณภาพสีของถั่วระดูญี่ปุ่น ใช้โปรแกรม Visual Basic C/C++ โปรแกรมนี้จะทำการวิเคราะห์ผลจากภาพถั่วระดูญี่ปุ่น ซึ่งรับภาพจากกล้องถ่ายภาพ CCD progressive scan ส่งผ่านมาทำการประมวลผลในโปรแกรมที่คอมพิวเตอร์ จากนั้นจะส่งสัญญาณไปให้ชุดไมโครคอนโทรลเลอร์ (Microcontroller) ส่วนนี้ก็มีเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์รับค่าจากคอมพิวเตอร์ และส่งสัญญาณไปยังตำแหน่งที่ต้องการ ซึ่งจะมี 2 คำสั่งด้วยกัน คำสั่งแรกคือให้อุปกรณ์พ่นลมทำงาน และอีก 1 วินาทีต่อมาก็ให้อุปกรณ์พ่นลมหยุดทำงาน เมื่อได้คำสั่งทำงานดังนี้แล้ว อุปกรณ์พ่นลมก็จะอัดลมเข้าไปด้วยความดันที่สามารถเป่าให้เมล็ดถั่วที่ต้องการคัดออกถูกคัดออกไป



บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การประมวลผลภาพ

2.1.1 ระบบของสี

ภาพดิจิทัลจะแทนด้วยเมตริกซ์ใน 2 มิติ โดยแต่ละหน่วยในเมตริกซ์จะเรียกว่าพิกเซล (Pixel) โดยแต่ละพิกเซลจะมีค่าเฉพาะตัวแสดงถึงสีต่างๆ

1. ระบบสีแบบ RGB (Red/Green/Blue)

ระบบสีแบบ RGB (Red/Green/Blue) จะบอกค่าสีในลักษณะ (R, G, B) โดยแต่ละค่าจะมีค่าตั้งแต่ 0 – 255 เช่น พิกัด (255, 255, 255) คือสีขาว พิกัด (0, 0, 0) เป็นสีดำ พิกัด (255, 0, 0) เป็นสีแดง พิกัด (128, 0, 255) เป็นสีแดงอมชมพูเข้ม เป็นต้น

2. ระบบสีแบบ HSI (Hue/Saturation/Intensity)

ระบบ HSI เป็นระบบที่สร้างมาจากสัญชาตญาณการนึกคิดของมนุษย์ในการกำหนดค่าสี โดยที่ค่า Hue (สี) จะเป็นค่าตัวแทนสีตั้งแต่ 0 – 360 ส่วนค่า Saturation (ค่าความเข้มของสี) มีระดับความเข้มตั้งแต่ 0 – 100 % สำหรับค่า Intensity (ค่าความสว่างของสี) ซึ่งมีช่วงตั้งแต่ 0% (มืด หรือดำ) จนถึง 100% (สว่าง หรือขาว) โดยเฉพาะค่า Hue ซึ่งเป็นค่าสีที่สามารถนำมาคำนวณวิเคราะห์ประมวลผลระบบสีได้

2.1.2 การจำแนกภาพ (Image Segmentation)

ภาพถ่ายโดยทั่วไปจะประกอบด้วยส่วนวัตถุ (Foreground) และส่วนพื้นหลัง (Background) ซึ่งเป็นฉากของภาพ ในการประมวลผลวัตถุของภาพจะต้องมีการจำแนกภาพเพื่อแยกวัตถุออกจากพื้นหลังของภาพ ซึ่งในการจำแนกภาพด้วยความคล้ายคลึงกันของค่าสี เป็นวิธีหนึ่งที่ย่างในการนำมาพิจารณา นั่นคือ วัตถุเดียวกันจะมีช่วงค่าสีที่ใกล้เคียงกัน และจะมีช่วงค่าสีที่แตกต่างจากส่วนอื่น ๆ ในการพิจารณาช่วงค่าสีของแต่ละส่วนในภาพสามารถใช้ฮิสโตแกรมของสี (Color Histogram) มาช่วยในการพิจารณาภาพได้ โดยฮิสโตแกรมจะแสดงออกมาเป็นกราฟแท่งเพื่อแสดงจำนวนจุดภาพของแต่ละค่าสีในภาพซึ่งจะไม่ได้สื่อถึงตำแหน่งในภาพ

ในการจำแนกส่วนประกอบต่าง ๆ ของภาพด้วยฮิสโตแกรมของค่าสี จะต้องมีจุดที่ใช้ในการแบ่งแยกช่วงค่าสีที่แตกต่างกันออกมาเป็นองค์ประกอบต่าง ๆ ซึ่งเรียกวิธีนี้ว่าการทำ เธรชโฮลด์ (Thresholding Technique) นั่นคือจะใช้ค่าเธรชโฮลด์ในการเปรียบเทียบเพื่อใช้เป็นการแบ่งแยกช่วงกลุ่มค่าสีที่แตกต่างกัน

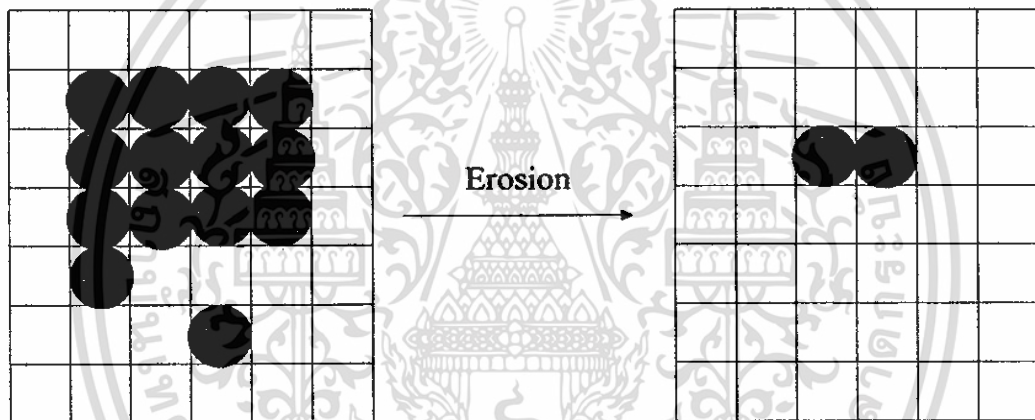
2.1.3 มอร์โฟโลยีทางคณิตศาสตร์ (Mathematical Morphology)

ภาพของวัตถุที่ได้จากการสแกนด้วยเครื่องสแกน (Scanner) หรือได้จากการถ่ายภาพของกล้องดิจิทัล (Digital Camera) เพื่อนำมาประมวลผลภาพด้วยคอมพิวเตอร์นั้น ภาพที่ได้อาจมีสัญญาณรบกวนเกิดขึ้น ทำให้คุณภาพของภาพนั้นลดลงไป ดังนั้นก่อนที่จะประมวลผลภาพต่อไปจึงต้องมีการวิเคราะห์เพื่อกำจัดสัญญาณรบกวนที่เกิดขึ้นนั้นออกไป โดยการวิเคราะห์มีหลายแบบต่างๆ กันไป

การทำงานโดยพื้นฐานของมอร์โฟโลยีจะประกอบด้วย อีรอสัน (Erosion) และไดเลชัน (Dilation) ซึ่งมีการทำงานดังนี้

1. อีรอสัน (Erosion) [2]

เป็นการลดจำนวนจุดภาพดำ ซึ่งจะเพิ่มช่วงห่างให้กว้างขึ้นระหว่างแต่ละกลุ่มของจุดภาพดำที่อยู่ตำแหน่งใกล้กัน หรือเป็นการกำจัดจุดภาพดำที่มีขนาดเล็กมากออกไป ดังรูปที่ 2.1

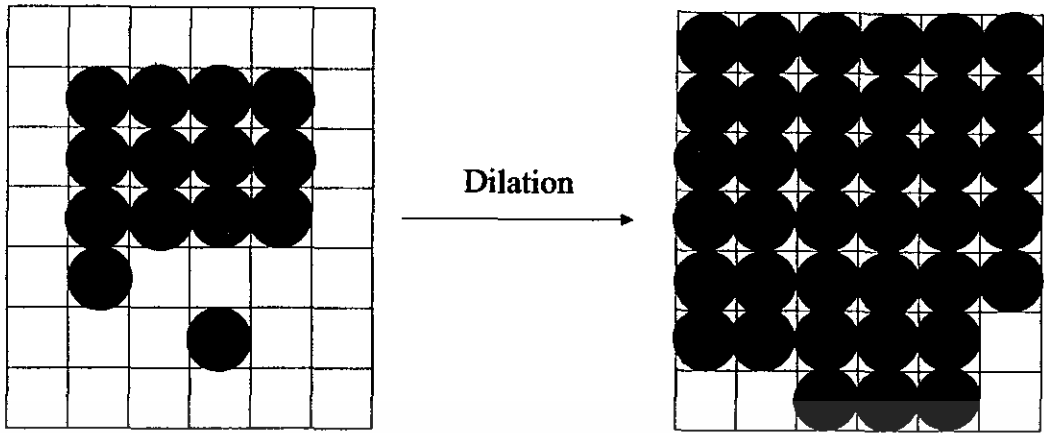


รูปที่ 2.1 การทำอีรอสัน [2]

2. ไดเลชัน (Dilation) [2]

การทำไดเลชันเป็นการทำงานที่ตรงข้ามกับอีรอสันคือ เป็นการเพิ่มจำนวนจุดภาพดำเพื่อขยายขนาดกลุ่มของจุดภาพดำ ทำให้เป็นการลดช่วงห่างให้แคบลง หรืออาจเกิดการเชื่อมต่อกันระหว่างแต่ละกลุ่มจุดภาพดำที่อยู่ตำแหน่งใกล้กัน ดังรูปที่ 2.2

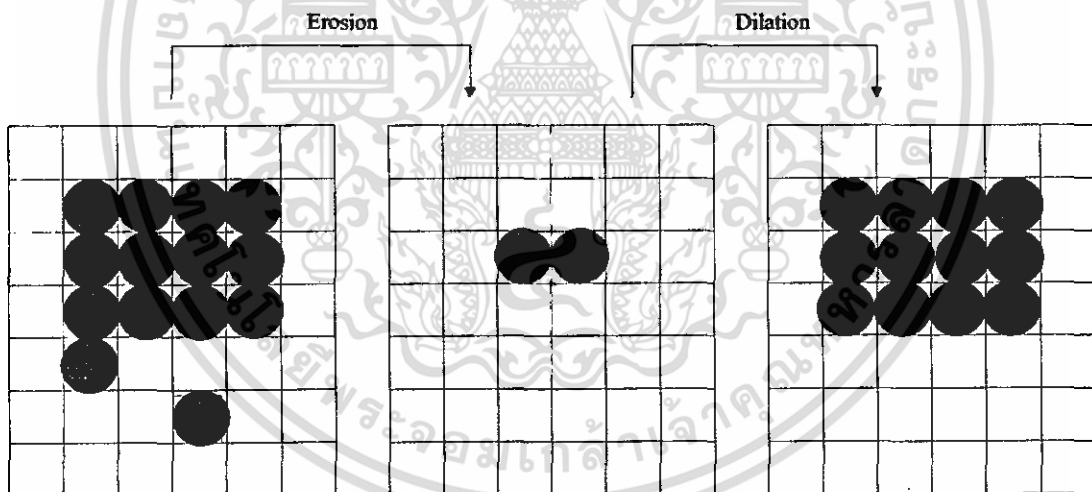
ในการเพิ่มประสิทธิภาพของมอร์โฟโลยีทางคณิตศาสตร์สามารถรวมการทำงานของทั้ง 2 พื้นฐานเข้าด้วยกัน ทำให้เกิดวิธีการ โอเพนิง (Opening) และ โคลสซิง (Closing) ซึ่งมีการทำงานดังนี้



รูปที่ 2.2 การทำไคเลชั่น [2]

3. โอเพนนิ่ง (Opening) [2]

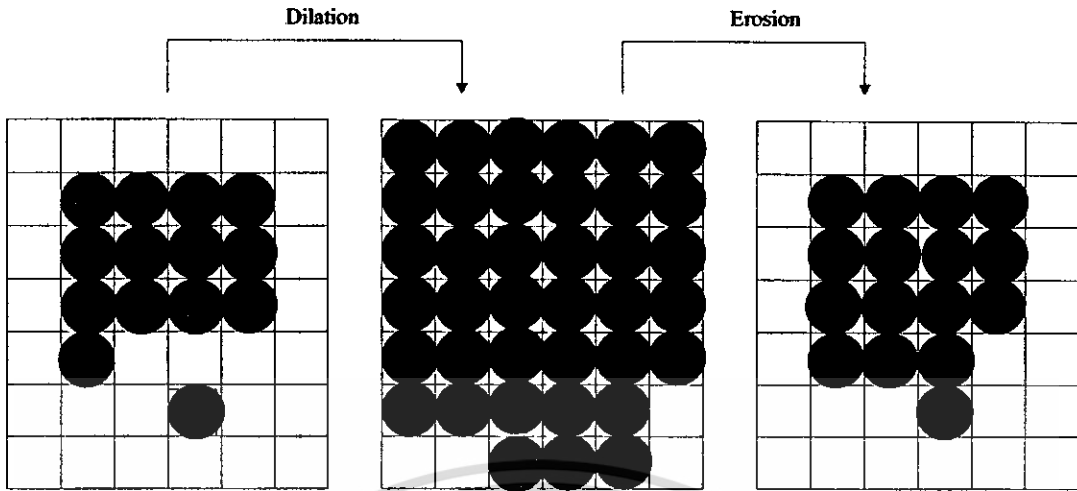
เป็นการทำงานร่วมกันระหว่างการทำอีรอสันและไคเลชั่นตามลำดับ คือจะเริ่มจากการทำอีรอสัน โดยลดจำนวนจุดภาพดำ แล้วตามด้วยการทำไคเลชั่น โดยเพิ่มจำนวนจุดภาพดำลงไปในภาพให้เท่ากับจำนวนของการทำอีรอสัน ดังรูปที่ 2.3



รูปที่ 2.3 การทำโอเพนนิ่ง [2]

4. โคลสซิง (Closing) [2]

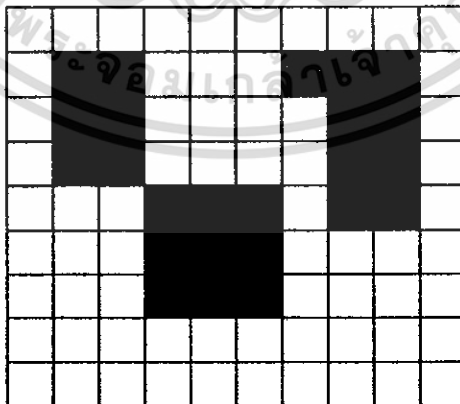
เป็นการทำงานที่มีลำดับตรงข้ามกับการทำโอเพนนิ่ง โดยจะทำไคเลชั่นและอีรอสันตามลำดับ คือจะเริ่มจากการทำไคเลชั่น โดยเพิ่มจำนวนจุดภาพดำ แล้วตามด้วยการทำอีรอสัน โดยลดจำนวนจุดภาพดำของภาพให้เท่ากับจำนวนรอบของการทำไคเลชั่น ดังรูปที่ 2.4



รูปที่ 2.4 การทำโคลสซิง [2]

2.1.4 การวิเคราะห์องค์ประกอบที่เชื่อมต่อกันของภาพ (Connected Components) [2]

ในภาพๆ หนึ่งอาจจะมียอดองค์ประกอบหรือวัตถุอยู่วัตถุเดียวหรือมากกว่าหนึ่งวัตถุขึ้นไป โดยในแต่ละวัตถุอาจจะอยู่ในตำแหน่งที่ต่อเนื่องกัน ซ้อนทับกัน หรือห่างจากกัน ซึ่งในภาพที่เป็นแบบไบนารีจะมีการพิจารณาการเชื่อมต่อของวัตถุ โดยพิจารณาจากพิกเซลที่เป็นส่วนของวัตถุกับพิกเซลรอบข้าง ๆ (Neighborhoods) ว่ามีการเชื่อมต่อกันหรือไม่ ซึ่งสามารถพิจารณาได้ทั้งใน 4 ทิศทาง (4-connected Neighborhoods) หรือใน 8 ทิศทาง (8-connected Neighborhoods) ซึ่งถ้าพิกเซลที่เป็นส่วนของวัตถุมีการเชื่อมต่อกันจะถือว่าเป็นวัตถุชนิดเดียวกัน แต่ถ้าพิกเซลที่เป็นส่วนของวัตถุขาดออกจากกันหรือไม่เชื่อมต่อกันจะกำหนดให้เป็นวัตถุที่ต่างกัน ดังแสดงในรูปตัวอย่างดังรูปที่ 2.5



รูปที่ 2.5 องค์ประกอบของภาพ [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่แสดงดังตัวอย่างข้างบนนั้นกำหนดให้พิกเซลสีขาวเป็นส่วนของพื้นหลัง และพิกเซลสีดำเป็นส่วนของวัตถุ เมื่อนำภาพมาวิเคราะห์ห่องค์ประกอบที่เชื่อมต่อกันของภาพทำให้สามารถระบุหมายเลข (Labeling of Connected Components) ให้กับองค์ประกอบของภาพได้ดังรูปที่ 2.6

0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	3	3	3	0
0	1	1	0	0	0	0	3	3	0
0	1	1	0	0	0	0	3	3	0
0	0	0	2	2	2	0	3	3	0
0	0	0	2	2	2	0	0	0	0
0	0	0	2	2	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0

รูปที่ 2.6 การระบุหมายเลขให้กับองค์ประกอบของภาพโดยพิจารณาใน 4 ทิศทาง [2]

จากรูปข้างบนเป็นผลการระบุหมายเลขให้กับองค์ประกอบของภาพ ซึ่งภาพจะประกอบไปด้วย 4 องค์ประกอบด้วยกัน โดยหมายเลข 0 แสดงส่วนของพื้นหลัง หมายเลข 1 แสดงวัตถุหนึ่ง หมายเลข 2 แสดงวัตถุที่สอง และหมายเลข 3 แสดงวัตถุที่สามของภาพ ซึ่งจากผลที่ได้ข้างเป็นการวิเคราะห์องค์ประกอบที่เชื่อมต่อกันของภาพโดยใช้ 4 ทิศทาง แต่ถ้าเป็นการวิเคราะห์โดยใช้ 8 ทิศทางจะทำให้ภาพข้างต้นประกอบไปด้วย 2 วัตถุเท่านั้น นั่นคือวัตถุที่หนึ่งและวัตถุที่สองเป็นวัตถุที่เชื่อมต่อกันในแนวทแยงจึงถือว่าเป็นวัตถุเดียวกัน

2.2 ไมโครคอนโทรลเลอร์ [3]

2.2.1 คุณลักษณะพื้นฐานของ MCS-51

คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

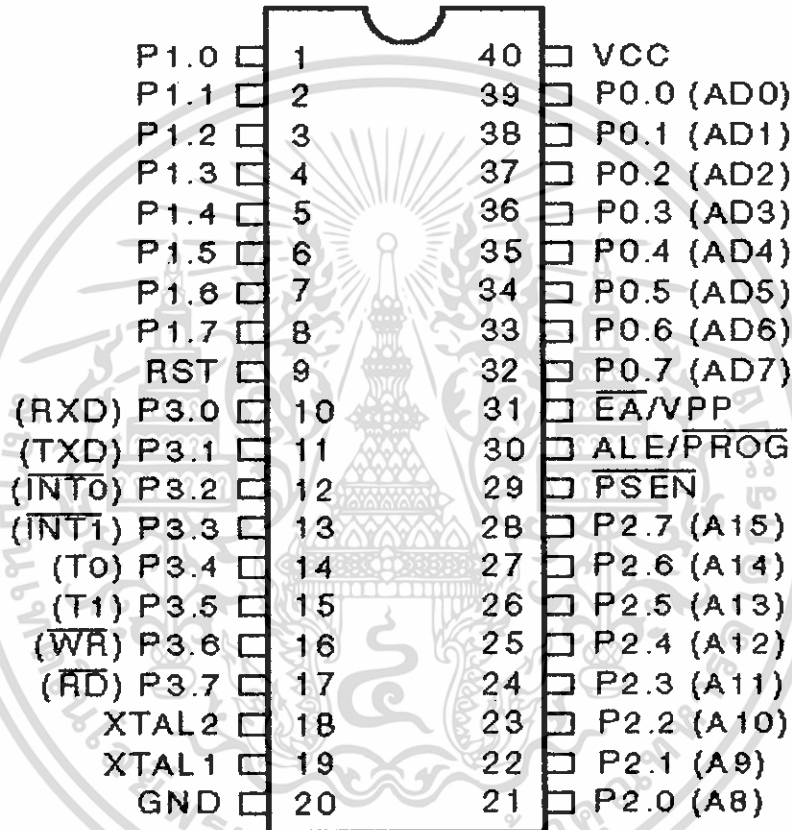
1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ชิพขนาด 8 บิต
2. ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้พันครั้ง
3. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีพรอมเพิ่มเติม
4. ขาพอร์ทเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
5. มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
6. ไทมเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
7. สามารถรองรับแหล่งกำเนิดอินเตอร์รัพต์ได้ 6 ประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

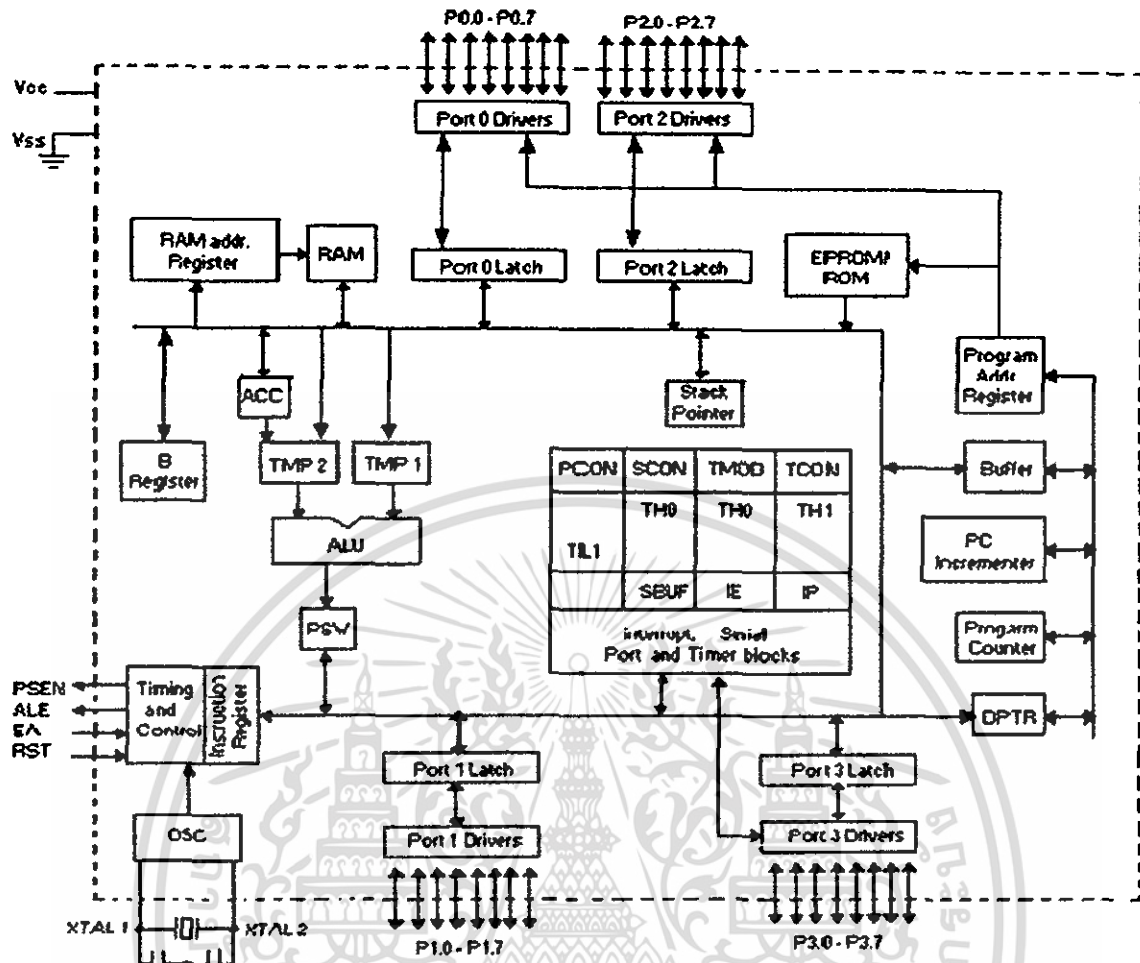
8. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
9. มีวงจรกำเนิดสัญญาณพิกาทูภายในชิป

2.2.2 ลักษณะการจัดขาของ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2.7 และ 2.8 โดยมีรายละเอียดขั้นต้นดังนี้



รูปที่ 2.7 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89xx [3]



รูปที่ 2.8 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ของ ATMEL [3]

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขาราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มีขา 8 ขา สามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อกับสาย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนี้ขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อกับแอดเดรส และขาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ท 1 (P1.0-P1.7) มีขา 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ทใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย

ขาพอร์ท 2 (P2.0-P2.7) มีขา 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ทใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ทนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ทอินพุตได้ นอกจากนี้ขาพอร์ทนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ท 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ทใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ทนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ทอินพุตได้ นอกจากนี้ขาพอร์ท 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นตอนต่อไปนี้

- P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD
- P3.1 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD
- P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัพต์จากภายนอกช่อง 0 หรือขา $\overline{\text{INT0}}$
- P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัพต์จากภายนอกช่อง 1 หรือขา $\overline{\text{INT1}}$
- P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทมเมอร์จากภายนอกช่อง 0 หรือขา T0
- P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทมเมอร์จากภายนอกช่อง 1 หรือขา T1
- P3.6 ใช้เป็นขาสัญญาณ $\overline{\text{WR}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก
- P3.7 ใช้เป็นขาสัญญาณ $\overline{\text{RD}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขา รีเซ็ต (Reset) ใช้ในการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ โดยใช้การป้อนสัญญาณเพื่อรีเซ็ต สถานะที่ขานี้ต้องอยู่ในระดับรีเซ็ตอย่างน้อย 2 แมกซ์อินไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา $\overline{\text{ALE}}/\text{PROG}$ (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ท 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนี้ขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา $\overline{\text{PSEN}}$ (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัว

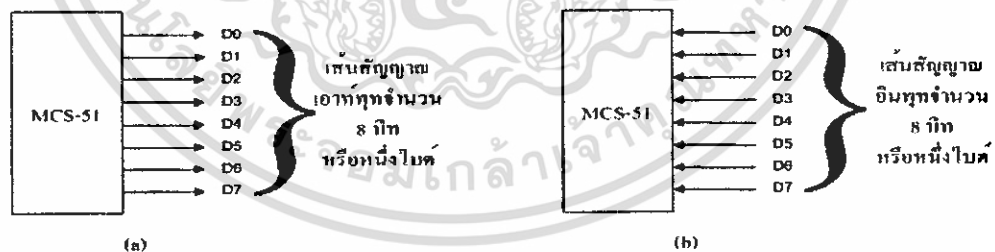
ไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขาที่ 2 ครั้ง ในแต่ละแมกซ์ซินไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอกขาที่นี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อกับหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าขาขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขาขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขาขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +5V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.2.3 พอร์ตอินพุตและพอร์ตเอาต์พุต

พอร์ต คือ แอแดคเรสหนึ่งที่ได้รับกำหนดไว้เพื่อการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อกับขึ้นอยู่กับการไหลของข้อมูล เมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก จากรูปที่ 2.9(a) เป็นการใช้นิโครคอนโทรลเลอร์เป็นเอาต์พุตพอร์ต และจากรูป 2.9 (b) เป็นการใช้นิโครคอนโทรลเลอร์เป็นอินพุตพอร์ต



รูปที่ 2.9 การใช้ไมโครคอนโทรลเลอร์เป็นอินพุตและเอาต์พุตพอร์ต [3]

1. การใช้งานพอร์ตเป็นอินพุต

การใช้งานพอร์ตเป็นการอินพุตข้อมูลจะต้องเริ่มต้นด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตสั้ก่อนเป็นอันดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในซึ่งมีผลทำให้บิตนั้นของพอร์ต 1, 2 และ 3 เป็นสถานะลอจิกสูง ตัวต้านทานนี้มีค่าประมาณ 50 k Ohm ซึ่งเป็นค่าที่สูงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาก และทำให้อุปกรณ์ภายนอกสามารถจับสัญญาณของพอร์ทเหล่านี้เป็นลอจิกค่าได้ง่าย สำหรับบิตของพอร์ท 0 นั้นแม้ว่าจะมีหลักการการทำงานที่คล้ายคลึงกันกับบิตของพอร์ทอื่นๆ แต่เนื่องจากไม่มีตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในไว้ ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่จับสัญญาณเอาต์พุตนั้นหยุดการทำงาน ก็จะเป็นผลให้สัญญาณนี้อยู่ในสถานะอิมพีแดนซ์สูงแทน

2. การใช้งานพอร์ทเป็นเอาต์พุต

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ททุกพอร์ท ข้อมูลนี้จะถูกส่งให้กับ ฟลิปฟลอปซึ่งจะค้างค่านีไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่จับสัญญาณเอาต์พุตนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสถานะลอจิกเป็นลอจิกค่าด้วย

ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ท 1, 2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่จับสัญญาณเอาต์พุตนั้นหยุดทำงาน มีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัวต้านทานที่ Pull-up อยู่ภายในนั้น แต่สำหรับการใช้งานในแต่ละบิตทางพอร์ท 0 นั้นจะมีผลแตกต่างออกไป โดยขาสัญญาณจะมีสถานะอิมพีแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านทานภายในเชื่อมต่ออยู่นั่นเอง ดังนั้นการใช้งานพอร์ท 0 เป็นการนำข้อมูลออกจากเอาต์พุต จึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับลอจิกสูงแทน

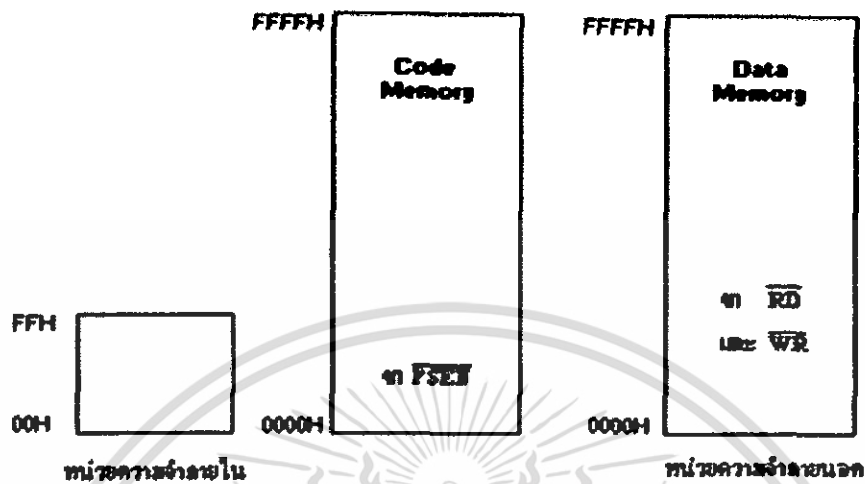
2.2.4 หน่วยความจำโปรแกรมของ MCS-51

หน่วยความจำโปรแกรม (Program Memory) มีไว้เพื่อบรรจุคำสั่งหรือโปรแกรมที่ผู้ใช้พัฒนาขึ้นจัดเก็บไว้ในหน่วยความจำ โดยอาจจะประกอบอยู่ภายในตัวของไอซี 89C51 เอง หรือเป็นไอซีหน่วยความจำ EPROM หรือ ROM แยกออกต่างหากได้ ในกรณีหลังจำเป็นต้องมีการใช้พอร์ทอินพุตเอาต์พุต ทำหน้าที่เป็นบัสแอดเดรส และบัสข้อมูลเพื่อให้สามารถติดต่อกับหน่วยความจำมาตรฐานทั่วไปได้

หน่วยความจำโปรแกรมของ 89C51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้ระบบข้อมูลเหล่านี้ก็ยังไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ใน ไอซีหน่วยความจำประเภทต่างๆ เช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory)

การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 89C51 จะมีการจัดพื้นที่ดังรูปที่ 2.10 โดยที่ไมโครคอนโทรลเลอร์ 89C51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมสูงสุดได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ

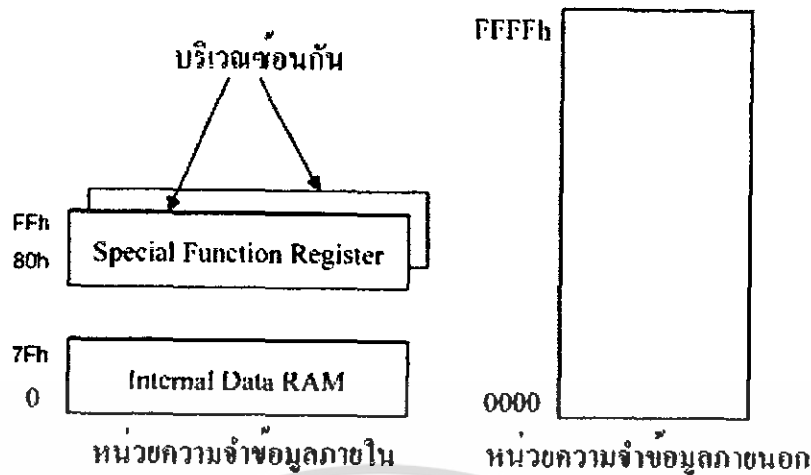
EPROM ที่อยู่ภายในตัวไอซี ไมโครคอนโทรลเลอร์เองและหน่วยความจำภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ



รูปที่ 2.10 การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 [3]

2.2.5 หน่วยความจำข้อมูลของ MCS-51

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยที่หน่วยความจำข้อมูลจะมีลักษณะเป็นหน่วยความจำ RAM แบบสแตติก (Static) ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ในหน่วยความจำนี้สูญหายไป พื้นที่ของหน่วยความจำข้อมูลของ 89C51 สามารถมีได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตำแหน่งที่ตั้งของหน่วยความจำนั้นดังลักษณะในรูปที่ 2.11 คือ หน่วยความจำโปรแกรมภายใน (Internal Data Memory) หรือ RAM ที่อยู่ภายในตัวไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 2.11 ลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล [3]

1. หน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายใน 89C51 มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกออกได้เป็นสองลักษณะคือ พื้นที่เฉพาะสำหรับตัวประมวลผลกลางใช้งานเท่านั้นซึ่งเรียกวารีจิสเตอร์ และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา จากรูปที่ 2.12 แสดงถึงการจัดพื้นที่ของหน่วยความจำข้อมูลภายในของ 89C51 ซึ่งแบ่งเป็นสองส่วน คือ หน่วยความจำขนาด 128 ไบต์แรก และหน่วยความจำขนาด 128 ไบต์ถัดไป

2. หน่วยความจำขนาด 128 ไบต์แรก

บริเวณนี้จะมีตำแหน่งแอดเดรสอยู่ในช่วง 00H ซึ่งแบ่งได้เป็นอีกสามส่วนดังนี้

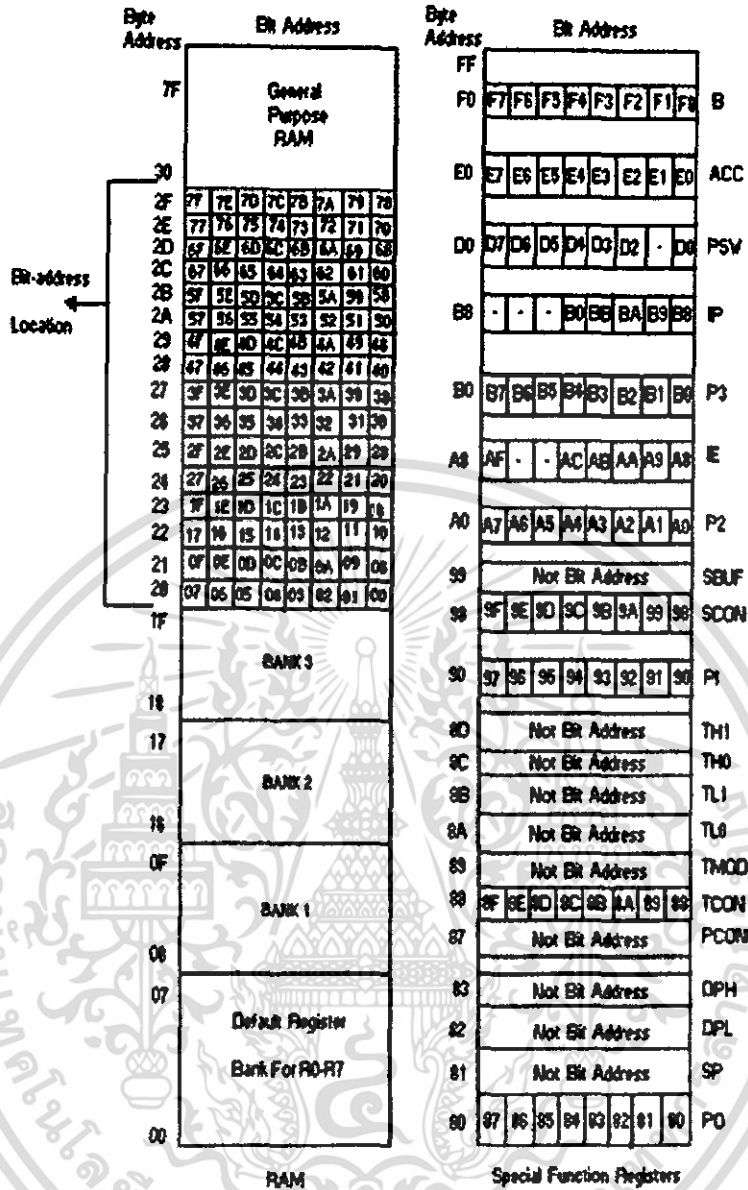
- บริเวณแอดเดรส 00H-1FH จำนวน 32 ไบต์ จำแนกออกเป็นกลุ่มหรือแบ่งก็ข้อมูลจำนวน 8 ไบต์ รวมทั้งหมดสี่กลุ่ม พื้นที่ข้อมูลในแต่ละกลุ่มจะถูกใช้งานในฐานะของรีจิสเตอร์ทั่วไป เรียกว่า รีจิสเตอร์ R0-R7

- บริเวณแอดเดรส 20H-2FH จำนวน 16 ไบต์ จะเป็นพื้นที่ส่วนสำหรับผู้ใช้ที่สามารถเข้าถึงข้อมูลได้ทั้งแบบ ไบต์และแบบบิต

- บริเวณแอดเดรส 30H-7FH เป็นบริเวณที่สามารถนำไปใช้งานได้โดยอิสระ โดยสามารถอ้างอิงได้เฉพาะในลักษณะของ ไบต์ข้อมูลตามปกติเท่านั้น

3. หน่วยความจำขนาด 128 ไบต์ถัดไป

พื้นที่ตั้งแต่แอดเดรส 80H-FFFH เป็นหน่วยความจำที่นำมาใช้งานเป็นรีจิสเตอร์หน้าที่พิเศษ แต่ก็ยังมีบริเวณของหน่วยความจำที่อยู่บริเวณนี้ที่ผู้ใช้สามารถเก็บข้อมูลได้ แต่การเรียกใช้งานจะต้องมีการเข้าถึงข้อมูลแบบ โดยอ้อมเท่านั้น



รูปที่ 2.12 ตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต [3]

2.2.6 รีจิสเตอร์หน้าที่พิเศษ

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์หรือพอร์ทของ 89C51 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

1. แอควิวมูลเตอร์ (Accumulator)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บข้อมูลที่ส่งให้หน่วยทำงานในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณหารตัวเลข ในกรณีที่ไมใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

3. โปรแกรมเคาน์เตอร์ (Program Counter)

เป็นรีจิสเตอร์ที่ใช้สำหรับการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะออกไปทำงานในลำดับถัดไป การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

4. สแตกพอยน์เตอร์ (Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแตก สำหรับเก็บข้อมูลแอดเดรสของรีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม ค่าเริ่มต้นของสแตกจะอยู่ที่ตำแหน่ง 07H การใช้งานในโปรแกรมจะเรียกว่ารีจิสเตอร์ SP

5. ตัวชี้ข้อมูลหรือค่าตัวพอยน์เตอร์ (Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็น รีจิสเตอร์ขนาด 8 บิต สองตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจเป็นแอดเดรสของอุปกรณ์ภายนอก

6. โปรแกรมสแตตัสเวิร์ด (PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กส์สภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์(Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

7. รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ท (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ทอินพุทเอาต์พุทโดยตรงซึ่งจะเป็น รีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะการอินพุทหรือการเอาต์พุทข้อมูลได้

8. รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล

9. รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ควบคุมการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรแกรมเซ็นเซอร์ การกำหนดอัตราทวิคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรมและแฟล็กส์ สภาวะการทำงานทั่วไป

10. รีจิสเตอร์ IP,IE,TMOD,SCON

เป็นกลุ่มรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอินเทอร์รัพต์ต่างๆ

72031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 การใช้งานพอร์ตสื่อสารอนุกรมแบบ Single Processor

1. พอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) สามารถรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน

- (1) ทางด้านส่งใช้ขา TxD (พอร์ต 3.1)
- (2) ทางด้านรับใช้ขา RxD (พอร์ต 3.0)

Serial Port Buffer (SBUF) ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรม โดยมีอยู่ 2 ตัว การส่งข้อมูล ข้อมูลที่จะส่งให้ใส่ใน SBUF โดยใช้คำสั่ง MOV SBUF, A โดยเตรียมข้อมูลที่จะส่งเข้า A ก่อน

การรับข้อมูล ข้อมูลที่รับได้จะอยู่ใน SBUF การถ่ายข้อมูลออกมาใช้คำสั่ง MOV A, SBUF แล้วจึงนำข้อมูลใน A ไปใช้

พอร์ตสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกัน โดยเลือกที่บิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรม มีดังนี้

SM0, SM1 บิตเลือกโหมดการทำงาน

M0	M1	โหมด	การทำงาน
		0	Shift register ความเร็วในการรับหรือส่งข้อมูลเท่ากับ (1/12) ของ CPU OSC
		1	8 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดได้จาก Timer 1, 2
		2	9 Bit UART ความเร็วในการรับหรือส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU OSC โดยขึ้นกับบิต SMOD ใน PCON
		3	9 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดที่ Timer 1, 2

ตารางที่ 2.1 การเลือกโหมดการทำงานในการรับส่งข้อมูล [3]

REN (Receive Enable) บิตควบคุมให้รับหรือไม่รับข้อมูล

- 1 : ให้รับข้อมูลได้
0 : ห้ามรับข้อมูล

หมายเหตุ (การรับข้อมูลสามารถห้ามได้ แต่การส่งข้อมูลห้ามไม่ได้)

TI แฟล็กซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

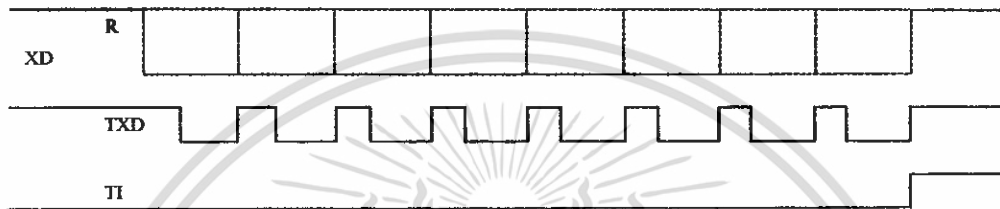
RI แฟล็กซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมควบคุมการรับและส่งข้อมูลทำได้ 2 วิธี

- การตรวจสอบบิต TI หรือ RI โดยใช้คำสั่งตรวจสอบบิต เช่น ใช้คำสั่ง WAIT: JNB TI, WAIT คำสั่งนี้หมายความว่า ถ้า TI=0 ใหวนไปยังแอดเดรสชื่อ WAIT
ถ้า TI=1 ถือว่าส่งข้อมูลเสร็จแล้วให้ทำคำสั่งถัดไป
- การใช้อินเตอร์รัพต์ควบคุม

โหมด 0 : พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งข้อมูลจะเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อนทางขา RxD เนื่องจากไม่มีการส่ง Start bit แต่จะส่ง shift clock ทางขา TxD [ความเร็ว (1/12) เท่าของ CPU Clock]

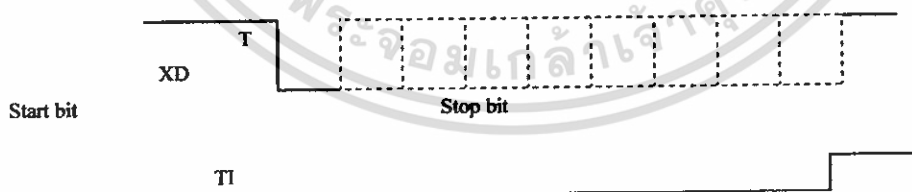


รูปที่ 2.13 Timing Diagram การส่งข้อมูลโหมด 0 [3]

โหมด 1 : พอร์ตสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และ อัตราโอเวอร์โพล์ของ Timer 1, 2

$$\text{Baud Rate Mode 1, 3} = \frac{2^{\text{SMOD}} \times \text{CPU OSC}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ Timer 1}$$

$$\text{Baud Rate Mode 1, 3} = \frac{\text{CPU OSC}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ Timer 2}$$

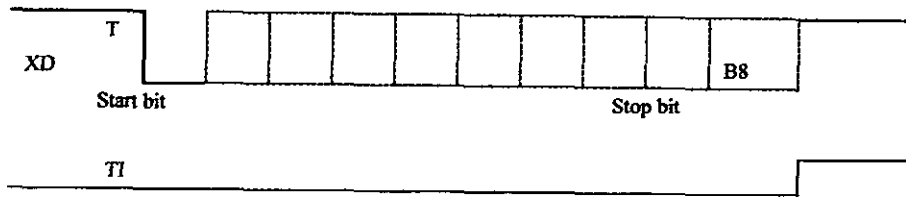


รูปที่ 2.14 Timing Diagram การส่งข้อมูลโหมด 1 [3]

โหมด 2 : พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 start bit และ 1 stop bit (TB8 นิยมนำมาใช้ส่ง Parity bit) ความเร็วในการรับส่งข้อมูลเท่ากับ (1/32) หรือ (1/64) เท่าของ CPU OSC โดยขึ้นกับบิต SMOD ใน PCON

$$\text{-Baud Rate (Mode 2)} = (1/32) \text{ CPU OSC} \quad \text{เมื่อ SMOD} = 1$$

$$\text{-Baud Rate (Mode 2)} = (1/64) \text{ CPU OSC} \quad \text{เมื่อ SMOD} = 0$$

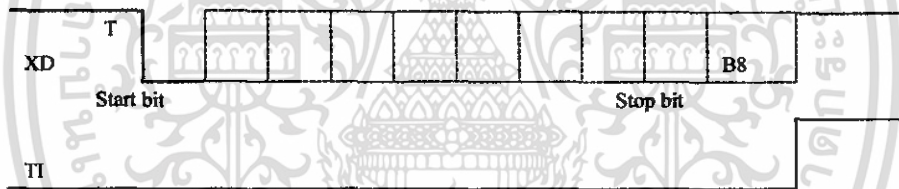


รูปที่ 2.15 Timing Diagram การส่งข้อมูลโหมด 2 [3]

โหมด 3 : พอร์ตสื่อสารอนุกรม 11 bit UART โดย DATA 8 bit, 1 start bit และ 1 stop bit เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1 สำหรับ 8051 หรือ อัตราโอเวอร์โพล์ของ Timer 2 (สำหรับ 80C154D)

$$\text{Baud Rate Mode 3} = \frac{2^{\text{SMOD}} \times \text{CPU OSC}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ Timer 1}$$

$$\text{Baud Rate Mode 3} = \frac{\text{CPU OSC}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ Timer 2}$$



รูปที่ 2.16 แสดง Timing Diagram การส่งข้อมูลโหมด 3 [3]

2.3 งานวิจัยที่เกี่ยวข้อง

ฉัฐพล และดาวพระสุภร์ [2] ศึกษาความเป็นไปได้ในการคัดแยกถั่วแระญี่ปุ่นโดยใช้กระบวนการประมวลผลภาพ ใช้พัฒนาโปรแกรมเพื่อใช้ในการคัดแยกขนาด สี และรูปร่าง โดยใช้โปรแกรม Mat lab เวอร์ชัน 6 โดยเริ่มจากการแยกวัตถุออกจากพื้นหลัง คำนวณพื้นที่ที่ภาพฉายของถั่วแระญี่ปุ่น แล้วนำมาเปรียบเทียบกับพื้นที่ภาพฉายมาตรฐาน ซึ่งฝักที่มีขนาดเล็กกว่ามาตรฐานจะถูกคัดออก จากนั้นใช้แบบจำลองค่าสีมาใช้ในการประมาณค่า ค่าสีของถั่วแระญี่ปุ่น เพื่อคัดแยกฝักที่มีสีไม่ต้องการออก และขั้นสุดท้ายคือการคัดแยกถั่วแระญี่ปุ่นจึงมีลักษณะบิดงอ โดยทำการหาค่าความแปรปรวนของแต่ละจุดบนเส้นขอบของถั่วแระญี่ปุ่น ฝักที่บิดงอจะมีค่าความแปรปรวนของมุมสูงกว่าฝักปกติ ในการวิเคราะห์ใช้ภาพถ่ายของถั่วแระญี่ปุ่นพันธุ์ AGS 292 ทั้งแบบที่ต้องการและไม่ต้องการผลคือ

1. การหาพื้นที่ภาพฉายยังได้ค่าสูงกว่าที่วัดได้โดย Planimeter 9.8% ซึ่งในส่วนนี้ไม่มีผลต่อการคัด

แยกโดยใช้การประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คัดผิด โดยคัดผิดที่ต้องการออกไป 8% และผิดที่ไม่ต้องการยังอยู่ 21.4%

Chalidabhong-, Sirisomboon and Yimyam. [4] วิเคราะห์ลักษณะทางกายภาพของมะม่วงโดยใช้หลักการประมวลผลภาพและคอมพิวเตอร์วิชัน ประกอบด้วย

1. การถ่ายภาพวัตถุในหลายมุมมอง และจัดเก็บอยู่ในรูปแบบสี RGB ทำให้ได้ภาพผลมะม่วงในมุมมองที่แตกต่างกัน
2. การแยกส่วนของวัตถุออกจากพื้นหลัง ประกอบด้วย การสร้างแบบจำลองค่า Hue ของผลมะม่วง การแยกส่วนของมะม่วงออกจากพื้นหลังโดยพิจารณาแบบจำลองค่า Hue ของผลมะม่วง การสร้างเส้นขอบมะม่วง
3. การเทียบวัตถุกล้อง จะทำให้ได้ค่าตัวแปรภายในกล้องและตัวแปรภายนอกกล้อง โดยวิธีการถ่ายภาพตารางกระดาษขาวค่า
4. การสร้างวัตถุในเชิง 3 มิติจากโครงร่าง 2 มิติของมุมมอง
5. การวิเคราะห์ลักษณะทางกายภาพ แบ่งเป็นการวิเคราะห์ลักษณะทางกายภาพในเชิง 2 มิติ เช่น พื้นที่ภาพฉาย ความยาว ความกว้าง ความหนา สี และการวิเคราะห์ลักษณะทางกายภาพในเชิง 3 มิติ เช่น ปริมาตรและพื้นที่ผิว ผลการทดลองพบว่างานวิจัยที่น่าเสนอนี้เป็นวิธีการวัดลักษณะทางกายภาพและคัดแยกขนาดของมะม่วงที่มีประสิทธิภาพและสะดวกรวดเร็วเมื่อเปรียบเทียบกับการทำงานต่างๆ เหล่านี้โดยใช้คนหรือเครื่องมือวัดในห้องปฏิบัติการ

พูนพัฒน์ และ อัมพวัน [5] ศึกษาการคัดคุณภาพของมะม่วงในทุกด้าน ได้แก่ขนาด รูปร่าง ความอ่อนแก่ ความสุก และปริมาณค้ำหนิโดยประยุกต์ใช้เทคโนโลยีแมชชีนวิชันซึ่งมีการใช้งานอย่างแพร่หลายในต่างประเทศ ซึ่งเริ่มจากความสัมพันธ์ระหว่างข้อมูลถ่ายภาพกับลักษณะคุณภาพกับผลมะม่วง แล้วนำข้อมูลที่ได้มาออกแบบการทำงานของโปรแกรมควบคุม ซึ่งระบบจะถูกประเมินความสามารถโดยเปรียบเทียบผลการคัดคุณภาพของระบบกับการคัดด้วยคน

ในการทดสอบนั้น ใช้มะม่วงพันธุ์น้ำดอกไม้ โดยมีข้อมูลการคัดดังนี้

- การคัดขนาด พิจารณาความสัมพันธ์ระหว่างน้ำหนักของผลมะม่วงกับความยาวหรือความกว้างของผลมะม่วง (โดยวัดในแนวเส้นแกนหลักและแกนรองตามลำดับ)หรือจุดสีของภาพถ่ายผลมะม่วง
- การคัดรูปร่าง จะกำหนดให้ระบบแมชชีนวิชันแสดงตำแหน่งที่สำคัญของผลมะม่วงจำนวน 4 จุด (ขั้ว ปลาย ด้านท้อง ด้านหลัง) โดยกำหนดคู่ลำดับพิกัดเชิงเส้น (x, y) ซึ่งคู่ลำดับดังกล่าวใช้เป็นข้อมูลเบื้องต้นในการคำนวณค่าสัดส่วนสำหรับการประเมินรูปร่างของผลมะม่วง
- การคัดความแก่โดยวิธีการในการประเมินความแก่ของผลมะม่วงพันธุ์น้ำดอกไม้พบว่าความถ่วงจำเพาะของผลมะม่วงสามารถใช้เป็นดัชนีในการคัดแยกได้
- การคัดความสุก ทำได้โดยการหาค่าสัดส่วนของปริมาณของพื้นที่สีเหลืองที่ปรากฏอยู่บนเปลือกของผลมะม่วงต่อปริมาณพื้นที่ทั้งหมด เพื่อหาค่าสัดส่วนที่เหมาะสมในการแยกผลมะม่วงสุก ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าสัดส่วนดังกล่าวจะถูกเปรียบเทียบกับความหนาแน่นเนื้อ ซึ่งทำการทดสอบโดยใช้เครื่องวิเคราะห์เนื้อสัมผัส

- การประเมินปริมาณตำหนิ ซึ่งตำหนิบนเปลือกของผลมะม่วงส่วนใหญ่จะเป็นสีดำหรือสีน้ำตาลเข้ม โดยใช้ระบบแมชชีนวิชันจะรายงานพื้นที่ตำหนิเป็นจำนวนพิกเซล โปรแกรมคอมพิวเตอร์จะเปรียบเทียบค่าดังกล่าวกับปริมาณพื้นที่ทั้งหมดแล้วแบ่งชั้นคุณภาพตามที่มาตรฐานกำหนด

จากผลทดสอบการทำงานของเครื่องคัดคุณภาพมะม่วงระบบแมชชีนวิชัน ที่ออกแบบให้ประเมินผลคุณภาพของผลมะม่วงโดยใช้ข้อมูลจากภาพถ่ายดิจิทัล ได้แก่ จำนวนจุดสีที่กำหนดและค่าคู่ลำดับที่แสดงถึงตำแหน่งสำคัญของมะม่วงจำนวน 4 จุด สรุปได้ว่าเครื่องคัดคุณภาพมะม่วงที่สร้างขึ้นมีอัตราการคัดคุณภาพที่ 797 ผลต่อชั่วโมง สามารถคัดคุณภาพในด้านขนาด ความสุก และตำหนิ ได้ถูกต้องมากกว่าร้อยละ 80 ส่วนการตรวจสอบรูปร่างและความอ่อนแก่ของมะม่วงระบบจะมีความถูกต้องในการคัดแยกต่ำกว่าร้อยละ 80 จึงต้องมีการปรับปรุงและเพิ่มจุดตรวจสอบและพัฒนาระบบตัดสินใจ

Aleixos et al. [6] ได้ทดสอบสมรรถนะและความแม่นยำของระบบคัดขนาดและคุณภาพของมะนาวโดยใช้ระบบแมชชีนวิชันดังต่อไปนี้ การตรวจสอบความแม่นยำของการคัดขนาดทำโดยวัดค่า Correlation coefficient และ Standard error การตรวจสอบความแม่นยำของการคัดสีทำโดยหาเปอร์เซ็นต์ Coincidence และการตรวจสอบความแม่นยำของการคัดตำหนิทำโดยทดสอบขั้นตอนการ Segmentation โดยการเปรียบเทียบระหว่างใช้คนกับการ Segmentation ของระบบและการเปรียบเทียบการคัดตำหนิโดยหาเปอร์เซ็นต์ Coincidence โดยการคัดทั้ง 3 อย่างเป็นการคัดโดยระบบที่ออกแบบกับการคัดโดยคน การประมาณเวลาในการคำนวณของระบบแมชชีนวิชัน เวลาที่ใช้ของแต่ละ Processor และเวลาที่ใช้ในการคัดแยกของกระบวนการต่างๆ เช่น กระบวนการคัดขนาด กระบวนการคัดสีและตำหนิของระบบที่ออกแบบ นอกจากนี้ยังมีการวัดความเร็วในการตรวจสอบ (Inspection speed, fruits/s)

บทที่ 3

การออกแบบและการสร้าง

การออกแบบและการสร้างระบบแบ่งออกเป็น 2 ส่วน

1. ส่วน Software

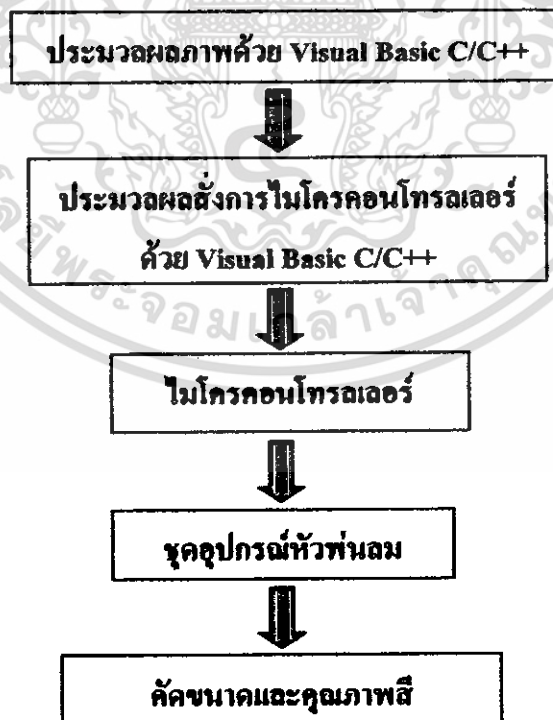
- เขียนโปรแกรมควบคุมการประมวลผลภาพและควบคุมระบบเครื่องจักรด้วยภาษา C/C++ โดยใช้ OpenCV

2. ส่วน Hardware

- การออกแบบระบบควบคุม
- การติดตั้งอุปกรณ์ต่างๆ ฯลฯ

3.1 แผนผังขั้นตอนการทำงาน

แผนผังขั้นตอนการทำงานในการออกแบบและสร้างเครื่องจักรขนาดและคุณภาพสี่ตัวระฆังญี่ปุ่น ขั้นตอนแรกคือการเขียนโปรแกรมประมวลผลภาพโดยใช้โปรแกรม GBA ประมวลผลคัดแยกตัวระฆังญี่ปุ่นแล้วส่งสัญญาณให้กับชุดอุปกรณ์ไมโครคอนโทรลเลอร์ทำงานสั่งชุดอุปกรณ์หัวพันลม เพื่อคัดแยกตัวระฆังญี่ปุ่น ดังรูปที่ 3.1

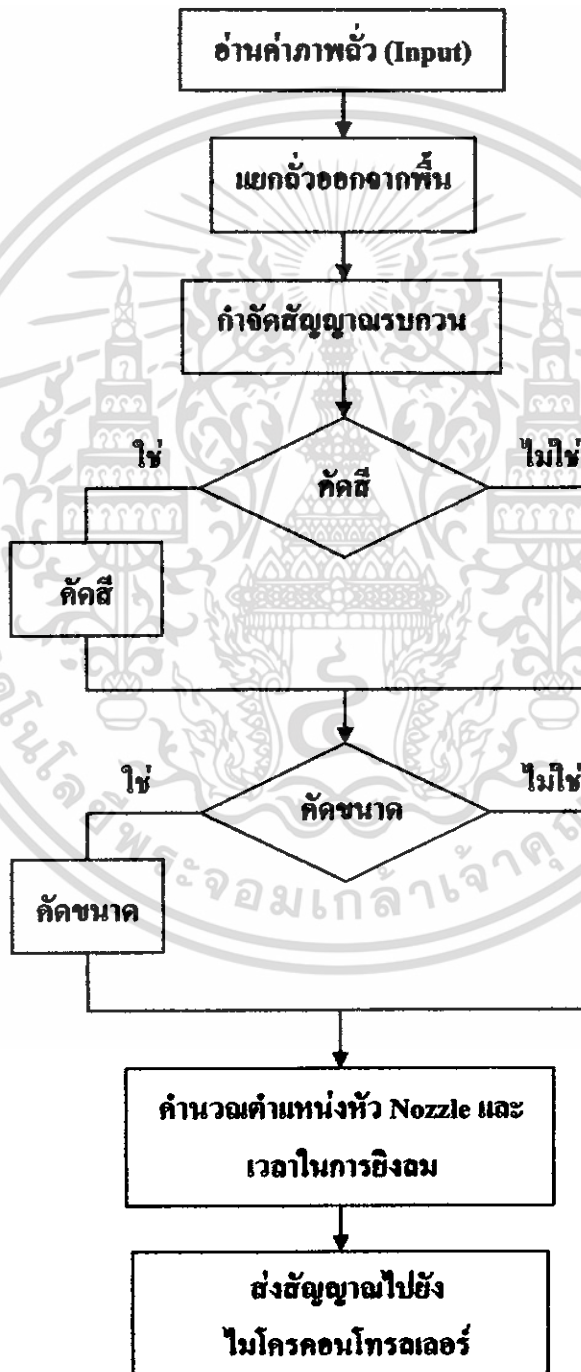


รูปที่ 3.1 แผนผังขั้นตอนการทำงานของเครื่องจักรขนาดและคุณภาพสี่ของตัวระฆังญี่ปุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการเขียนโปรแกรม

โปรแกรมที่เขียนแบ่งออกเป็น 3 โปรแกรม ได้แก่ โปรแกรม GBA (ภาคผนวก ก) ใช้ประมวลผลภาพเพื่อตัดขนาดและคุณภาพสีของตัวระเบิดู๊นและประมวลผลสั่งการ ไมโครคอนโทรลเลอร์ โปรแกรมรับภาพจากกล้อง (Camera) ใช้ดึงภาพจากกล้อง CCD progressive scan (ภาคผนวก ข) และโปรแกรม Finding Color Been (ภาคผนวก ค) ใช้คำนวณค่าเฉลี่ยและค่าเบี่ยงมาตรฐานของสีตัวเพื่อใช้เป็นค่าเทรชโฮลด์ โดยมี Flow Chart การทำงานของโปรแกรมหลัก GBA ดังรูปที่ 3.2



รูปที่ 3.2 Flow chart การทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 ตัวอย่างภาพถั่วแระญี่ปุ่นจากกล้องถ่ายภาพ CCD progressive scan

ภาพจากกล้องถ่ายภาพ CCD progressive scan ภาพถั่วแระญี่ปุ่นบนสายพานสีขาว ถ่ายรูปโดยใช้ มุมภาพค้ำบน ซึ่งภาพจะเก็บอยู่ในแบบสี RGB และเป็นไฟล์ BMP Image ดังรูปที่ 3.3



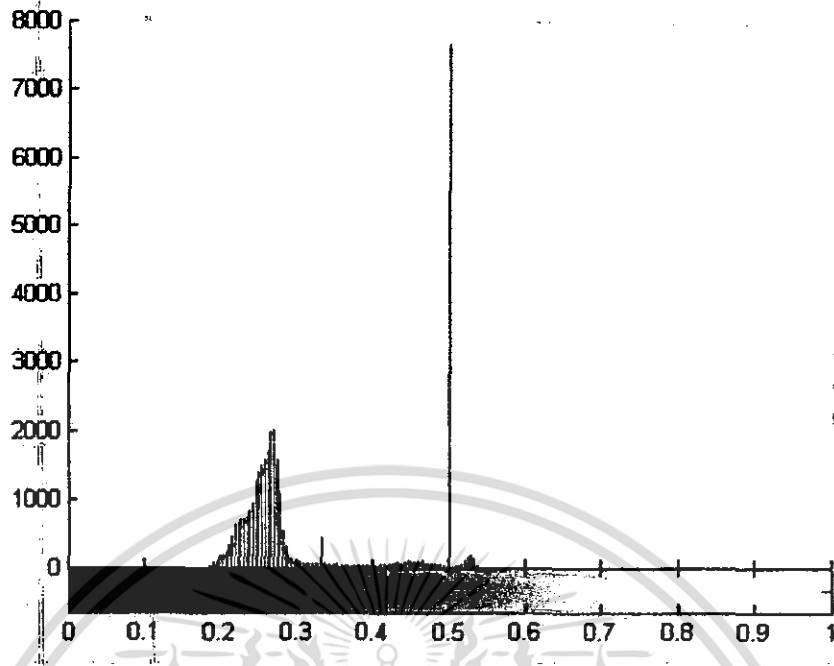
รูปที่ 3.3 ภาพถั่วแระญี่ปุ่นบนสายพานสีขาว

3.2.2 การแยกส่วนของถั่วแระญี่ปุ่นออกจากพื้นหลัง (Segmentation)

ในการวิเคราะห์ลักษณะทางกายภาพของถั่วแระญี่ปุ่นจะวิเคราะห์กับภาพถ่ายดังแสดงในรูปที่ 3.3 ซึ่งภาพนั้นเป็นภาพของถั่วแระญี่ปุ่นหลายฝัก กับพื้นหลังที่ประกอบไปด้วยหลายสี ดังนั้นก่อนที่จะสามารถวิเคราะห์ลักษณะทางกายภาพของถั่วแระญี่ปุ่นได้ จึงจะต้องแยกส่วนของถั่วแระญี่ปุ่นออกจากพื้นหลังเสียก่อน ซึ่งก็มีวิธีในการดำเนินการต่อไปนี้

1. การสร้างแบบจำลองค่า Hue ของถั่วแระญี่ปุ่น

จะเริ่มต้นโดยการนำภาพถ่ายถั่วแระญี่ปุ่นบนสายพานในมุมมองค้ำบน ดังรูปที่ 3.3 ซึ่งเป็นภาพที่อยู่ในรูปแบบของสี RGB จากนั้นจะทำการแปลงค่าสีของภาพจาก RGB เป็นค่า HSV เพื่อนำมาสร้างแบบจำลองค่า Hue เฉพาะในส่วนของถั่วแระญี่ปุ่น เมื่อนำมาทำเป็นฮิสโตแกรมของค่า Hue ของ โดยในฮิสโตแกรมประกอบด้วยค่า Hue ของสีถั่วกับพื้นหลังจะได้ดังรูปที่ 3.4 ดังนั้นจึงสามารถนำแบบจำลองค่า Hue ของถั่วแระญี่ปุ่นที่สร้างขึ้นมาใช้ในการพิจารณากับภาพเพื่อแยกส่วนของถั่วแระญี่ปุ่นออกจากพื้นหลัง



รูปที่ 3.4 ฮิสโตแกรมค่า Hue รูปถั่วแระญี่ปุ่นกับพื้นหลังสีขาว มี 2 ลูป ลูปที่ 1 เป็นค่า Hue ของถั่วแระญี่ปุ่น ส่วนลูปที่ 2 เป็นลูปของพื้นหลัง

2. การหาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน

นำภาพถั่วแระญี่ปุ่น ซึ่งเป็นภาพ RGB มาตัดเอาแต่เฉพาะส่วนที่เป็นสีของถั่วแระญี่ปุ่นดังรูปที่ 3.5 นำไปแปลงเป็นค่าสี Hue แล้วนำมาค่าสี Hue ที่ได้มาทำการหาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน โดยใช้โปรแกรม Finding Color Bean

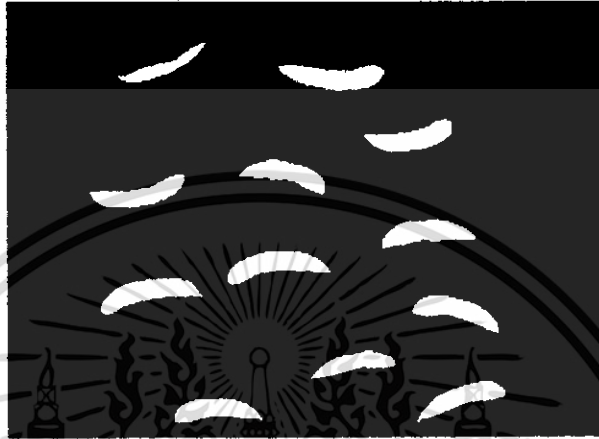


รูปที่ 3.5 ภาพถั่วแระญี่ปุ่นที่ตัดบางส่วนเพื่อนำมาวิเคราะห์สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การแยกส่วนของตัวแระฉี่ปุ่นออกจากพื้นหลัง

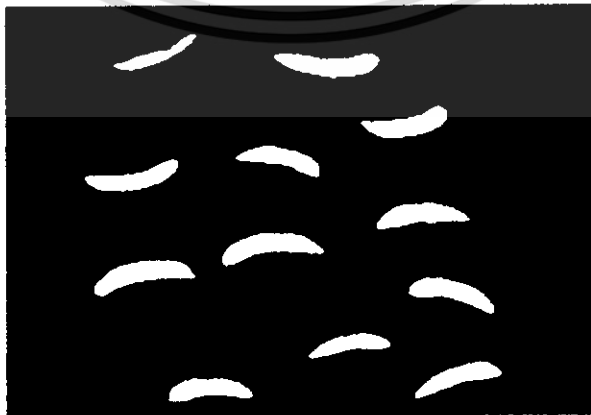
ภาพของตัวแระฉี่ปุ่นตัวอย่างที่มีรูปแบบสี RGB ทั้งหมดจะถูกแปลงมาเป็นค่า Hue จากนั้นจะนำค่า Hue ที่ได้มาพิจารณากับแบบจำลองค่าสี Hue ของภาพของตัวแระฉี่ปุ่นที่สร้างขึ้น โดยที่จะกำหนดว่าพิกเซลใดที่มีค่า Hue อยู่ในขอบเขตที่กำหนด จะถือว่าพิกเซลนั้นเป็นส่วนของตัวแระฉี่ปุ่น ทำให้ได้ภาพของตัวแระฉี่ปุ่นที่ถูกแยกออกจากพื้นหลัง ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 ภาพของตัวแระฉี่ปุ่นที่ถูกแยกออกจากพื้นหลัง

3.2.3 การกำจัดสัญญาณรบกวน

ภาพของวัตถุที่ได้จากการถ่ายภาพของกล้องถ่ายภาพ CCD progressive scan เพื่อนำมาประมวลผลภาพด้วยคอมพิวเตอร์ ภาพที่ได้อาจมีสัญญาณรบกวนเกิดขึ้น ทำให้คุณภาพของภาพนั้นลดลงไป ดังนั้นก่อนที่จะประมวลผลภาพต่อไปจึงต้องมีการวิเคราะห์เพื่อกำจัดสัญญาณรบกวนที่เกิดขึ้นนั้นออกไป โดยใช้วิธีมอร์โฟโลยีทางคณิตศาสตร์ ซึ่งในโปรแกรม Visual Basic C/C++ นี้ได้ใช้วิธีการทำ Opening จากรูปที่ 3.6 มีสัญญาณรบกวนเกิดขึ้น เมื่อผ่านการทำ Opening แล้วจะได้รูปภาพที่ไม่มีสัญญาณรบกวนดังรูปที่ 3.7



รูปที่ 3.7 รูปภาพที่ถูกกำจัดสัญญาณรบกวนแล้ว โดยใช้วิธี Opening

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การหาเรคโธลด์สีของถั่วแระญี่ปุ่น

ถั่วแระญี่ปุ่นที่จะส่งออกต้องมีสีเขียวสด ไม่มีสีเหลือง หรือสีน้ำตาลคล้ำปะปนอยู่ด้วย จึงจำเป็นต้องมีการคัดแยกเอาถั่วแระญี่ปุ่นที่มีสีอื่นปะปนอยู่ด้วยออกก่อนที่จะนำส่งออกต่างประเทศ ซึ่งในการคัดแยกคุณภาพสี มีขั้นตอนดังนี้คือ หาค่าเฉลี่ย Hue สีของถั่วแระญี่ปุ่นแต่ละฝัก โดยถ้าค่าเฉลี่ย Hue ของฝักใดมีค่ามากกว่าค่า Color Threshold ที่กำหนดไว้ ก็แสดงว่าฝักนั้นก็ยังมีสีเขียวสดตามต้องการ แต่ถ้าค่าเฉลี่ย Hue มีค่าน้อยกว่า Color Threshold ก็แสดงว่าฝักนั้นมีสีเหลืองและสีน้ำตาลคล้ำปะปนอยู่ด้วย ก็จะถูกคัดออกดังแสดงในรูปที่ 3.8 และการหาค่าเฉลี่ย Hue ของถั่วแระญี่ปุ่น หาได้จากโปรแกรม Finding Color bean (ภาคผนวก ก)



(a)



(b)

รูปที่ 3.8 (a) ถั่วแระญี่ปุ่นที่มีหลายสีปะปนกันอยู่

(b) ถั่วเหลืองที่มีสี Hue น้อยกว่าค่าสีเรคโธลด์ถูกคัดออก

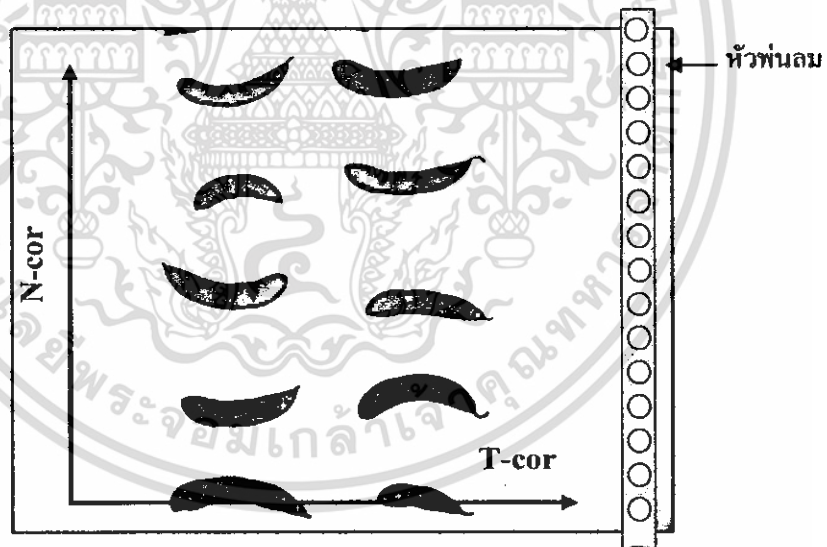
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การหาขนาดของถั่วระอูญี่ปุ่น

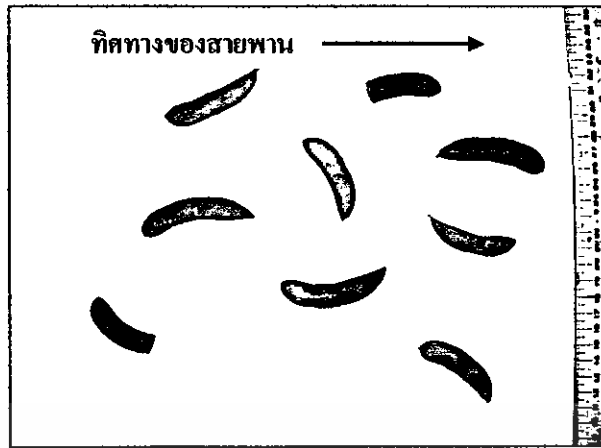
การหาขนาดของถั่วระอูญี่ปุ่น โดยนำภาพถั่วระอูญี่ปุ่นที่ถูกกำจัดสัญญาณรบกวนแล้วมาทำการวิเคราะห์องค์ประกอบที่เชื่อมต่อกันของภาพ แล้วนับจำนวนพิกเซลของแต่ละฝัก จากนั้นนำมาเปรียบเทียบกับอัตราส่วนของจำนวนพิกเซลต่อพื้นที่หน่วยตารางเซนติเมตร

การหาอัตราส่วนของจำนวนพิกเซลต่อพื้นที่หน่วยตารางเซนติเมตร ทำโดยใช้กล้องถ่ายภาพถั่วทั้งสองแกน ได้แก่แกน T-cor คือ แกนที่ใช้คำนวณเวลาในการส่งไมโครคอนโทรลเลอร์ มีลักษณะตามการเคลื่อนที่ของสายพาน และแกน N-cor คือ แกนที่ใช้คำนวณตำแหน่งของหัวพ่นลม มีลักษณะขวางการเคลื่อนที่ของสายพาน ดังรูปที่ 3.9 พร้อมกับไม้บรรทัดสเกลเป็นเซนติเมตร ภาพที่ได้จากกล้องจะมีขนาด 640 x 480 พิกเซล จากนั้นนำมาเทียบกับขนาดของภาพถั่วที่วัดได้จากไม้บรรทัดในรูปคือ 37.6 x 28 ซม. (รูปที่ 3.10) แล้วคำนวณจำนวนพิกเซลใน 1 ตร.ซม. ซึ่งขนาดของถั่วระอูญี่ปุ่นหาได้ตามสมการ

$$\text{ขนาดถั่ว (ตร.ซม.)} = (\text{จำนวนพิกเซลของถั่วระอูญี่ปุ่น}) / (\text{จำนวนพิกเซลใน 1 ตร.ซม.})$$



รูปที่ 3.9 ลักษณะแกนของสายพาน มีแกนตามแนวการเคลื่อนที่ของสายพาน (T-cor) และแกนขวางการเคลื่อนที่ของสายพาน (N-cor)



(a)



(b)

รูปที่ 3.10 (a) คือรูปภาพถั่วแระญี่ปุ่นในแนวแกน N-coordinate

(b) คือรูปภาพถั่วแระญี่ปุ่นในแนวแกน T-coordinate

3.2.6 การคำนวณหาจุด Centroid ของฝักถั่ว

ภาพที่ได้จากกล้อง CCD progressive scan มีสเกลเป็นพิกเซล โดยที่แต่ละพิกเซลถูกระบุค่าสองค่าด้วยกันคือ ค่าจากแกนเคลื่อนที่ตามสายพาน และค่าจากแกนขวางการเคลื่อนที่ของสายพาน เป็นคู่ลำดับ ดังแสดงในรูปที่ 3.11

จุด centroid ของฝักถั่วประกอบด้วยสองค่าด้วยกัน ได้แก่ ค่าในแกน T-cor และแกน N-cor การคำนวณหาจุด centroid ในแนวแกน T-cor โดยนำผลรวมค่าในแนวแกน T-cor ทั้งหมดภายในฝักถั่วมาหารด้วยพื้นที่ของถั่วเป็นพิกเซล จากรูปที่ 3.11 ค่าในแนวแกน T-cor ของถั่วคือ 6×9 และ 7×9 โดยที่ถั่วมีพื้นที่ 18 พิกเซล ค่าจุด centroid ในแนวแกน T-cor เท่ากับ $[(6 \times 9) + (7 \times 9)] / 18$ ได้ 6.5 ในทำนองเดียวกัน ค่าจุด centroid ในแนวแกน N-cor เท่ากับ $(2+2+3+3+4+4+5+5+6+6+7+7+8+8+9+9+10+10) / 18$ ได้ 6 ดังนั้นรูปถั่ว (รูปที่ 3.11) จะมีจุด centroid อยู่พิกเซลที่ (6.5, 6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N - cor

(4,1)	(5,1)	(6,1)	(7,1)	(8,1)	(9,1)
(4,2)	(5,2)	(6,2)	(7,2)	(8,2)	(9,2)
(4,3)	(5,3)	(6,3)	(7,3)	(8,3)	(9,3)
(4,4)	(5,4)	(6,4)	(7,4)	(8,4)	(9,4)
(4,5)	(5,5)	(6,5)	(7,5)	(8,5)	(9,5)
(4,6)	(5,6)	(6,6)	(7,6)	(8,6)	(9,6)
(4,7)	(5,7)	(6,7)	(7,7)	(8,7)	(9,7)
(4,8)	(5,8)	(6,8)	(7,8)	(8,8)	(9,8)
(4,9)	(5,9)	(6,9)	(7,9)	(8,9)	(9,9)
(4,10)	(5,10)	(6,10)	(7,10)	(8,10)	(9,10)
(4,11)	(5,11)	(6,11)	(7,11)	(8,11)	(9,11)

T - cor

รูปที่ 3.11 จาลองภาพถ่ายรูปถั่วมีสเกลเป็นพิกเซล

3.2.7 การคำนวณตำแหน่งหัว nozzle และเวลาในการยิงของหัว nozzle

นำค่าจุด centroid ของฝักถั่วที่ถูกคัดทิ้งมาแปลงค่าจากพิกเซลเป็นเซนติเมตร ค่าจุด centroid ในแนวแกน N-cor ใช้คำนวณหาตำแหน่งของหัว nozzle ที่จะยิงฝักนั้น ส่วนค่าจุด centroid ในแนวแกน T-cor ใช้คำนวณหาเวลาในการยิงหัว nozzle ที่จะยิงฝักที่ถูกคัดทิ้ง สามารถคำนวณได้จากสมการ

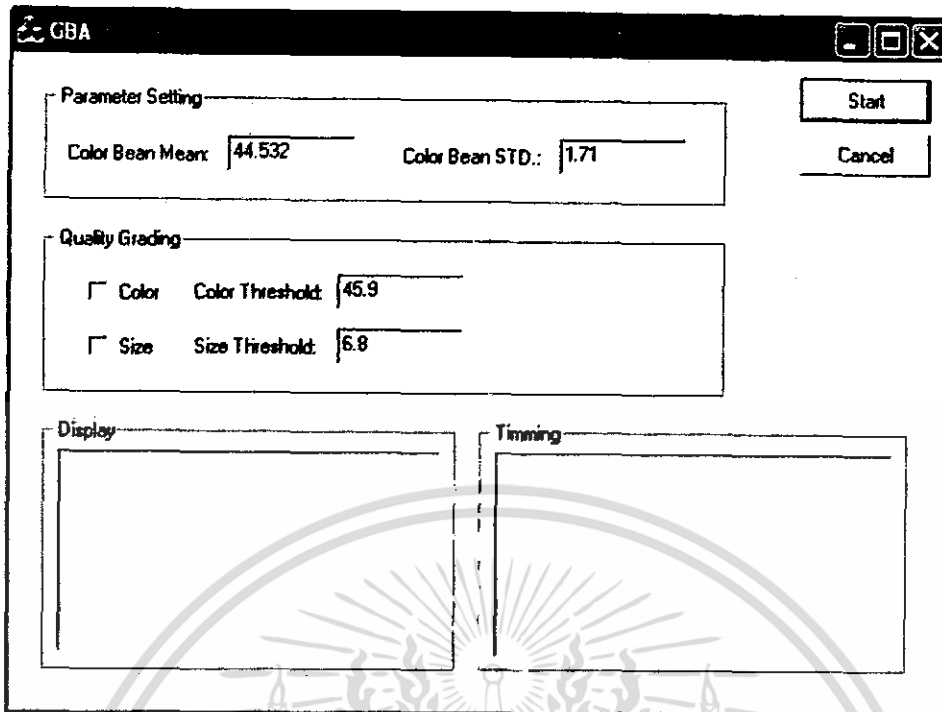
$$t = \frac{s}{v}$$

- โดย
- t คือ เวลาที่หัว nozzle จะยิงฝักถั่ว (วินาที)
 - s คือ ระยะจากจุด centroid ในแนวแกน T-cor ถึงระยะสุดสายพาน (เมตร)
 - v คือ ความเร็วของสายพาน (เมตร/วินาที)

3.2.8 ลักษณะของโปรแกรม GBA

โปรแกรม GBA (Green Soybean Analysis) ที่ใช้วิเคราะห์ถั่วจะแบ่งเป็น 3 ส่วน ดังรูปที่ 3.12 โดยส่วนแรก Parameter Setting ประกอบด้วยค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของสีถั่ว ซึ่งคำนวณได้จากโปรแกรม Mat lab เพื่อทำการแยกถั่วออกจากพื้นหลัง ส่วนที่สองจะเป็นกระบวนการคัดแยก โดยแบ่งเป็นสองกระบวนการ ได้แก่ กระบวนการคัดแยกสี และคัดแยกขนาด โดยที่แต่ละกระบวนการจะมีค่าเทรดโซลด์ เพื่อใช้คัดถั่วที่ไม่ต้องการออก ถั่วฝักไหนมีค่าสีหรือขนาดน้อยกว่าค่าเทรดโซลด์ จะถูกคัดออก ซึ่งในส่วนนี้สามารถเลือกทำการกระบวนการใดกระบวนการหนึ่ง หรือทำทั้งสองกระบวนการพร้อมกันก็ได้ ส่วนที่สามคือส่วนที่แสดงผลตำแหน่งของฝักถั่วที่ไม่ต้องการและตำแหน่งของหัวพ่นลมที่จะยิง จากนั้นโปรแกรมจะส่งสัญญาณไปยังชุดอุปกรณ์ไมโครคอนโทรลเลอร์ เพื่อสั่งหัวพ่นลมให้ยิงถั่วได้ตามตำแหน่งที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ลักษณะของโปรแกรม GBA

3.3 วงจรการควบคุมสัญญาณ

ในส่วนของการควบคุมสัญญาณนี้จะเป็นส่วนที่รับข้อมูลมาจากในส่วนแรก คือโปรแกรม GBA โดยหลักการแล้วพิจารณาเฉพาะค่าที่บอกตำแหน่งของหัวพ่นลม ที่มีอยู่ 20 ตำแหน่ง จากนั้นก็จะส่งค่าออกจาก Serial port เพื่อให้ Microcontroller รับค่าที่ส่งมา และส่งสัญญาณต่อไปให้อุปกรณ์ Relay switch ทำงานดังรูปที่ 3.13 เมื่อไฟครบวงจรการทำงานแล้ว อุปกรณ์ที่พ่นลมในที่นี้ใช้ Nozzle ในการพ่นลมทำงานได้ตรงตามตำแหน่งที่ต้องการเพื่อกำจัดฝักที่ไม่ต้องการทิ้งไป



รูปที่ 3.13 วงจรการควบคุมสัญญาณ

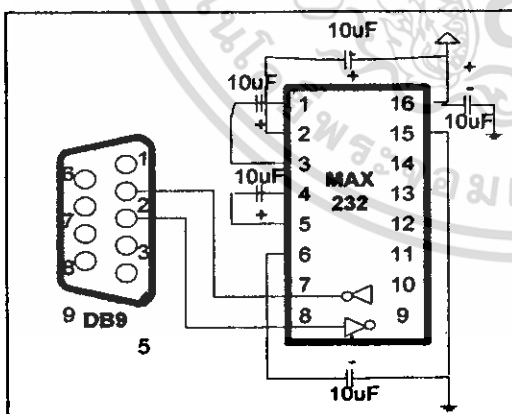
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 วงจรส่งค่าผ่าน serial port

ในส่วนการส่งค่าผ่าน Serial port นั้น หลังจากมีการประมวลผลและได้ฟังก์ชันที่ต้องคัดทิ้งออกและจุด coordinate ที่ฟังก์ชันนั้น ๆ มาแล้ว ที่ส่วนภาคการส่งค่าจะสนใจเฉพาะตำแหน่งที่ต้องการปล่อยออก ซึ่งในระบบ มี 20 ตำแหน่ง ดังตารางที่ 3.1 เรียงตามหน้ากว้างสายพจน

ตารางที่ 3.1 ค่าที่รับตามตำแหน่งหัวพ่นลมที่ต้องการ

ตำแหน่งที่	ค่าที่รับ	ตำแหน่งที่	ค่าที่รับ
1	A	11	K
2	B	12	L
3	C	13	M
4	D	14	N
5	E	15	O
6	F	16	P
7	G	17	Q
8	H	18	R
9	I	19	S
10	J	20	T



(a)



(b)

รูปที่ 3.14 (a) วงจรส่งค่าผ่าน Serial port

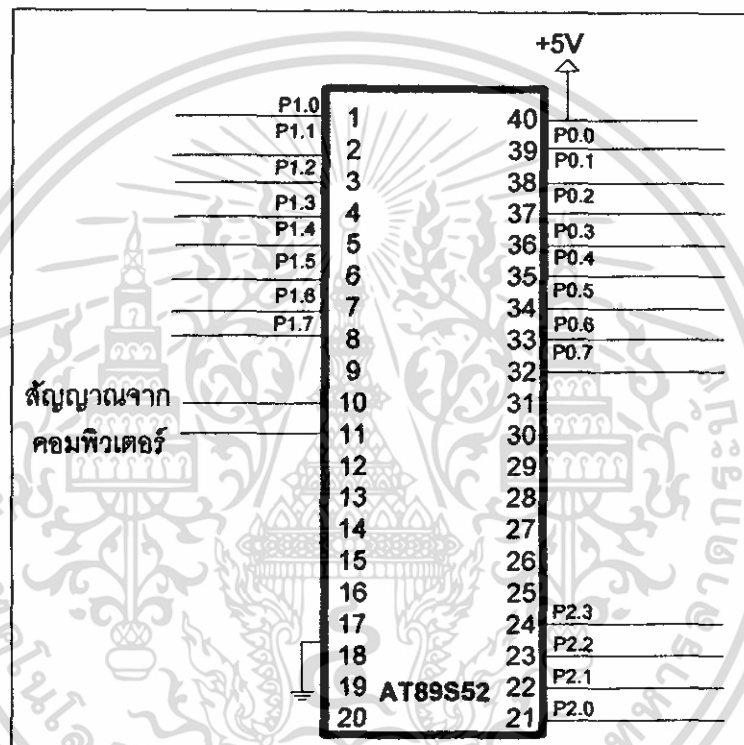
(b) สาย Serial Port (RS232)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.14 จะเห็นว่า DB9 เป็น Port ที่เชื่อมต่อมาจากคอมพิวเตอร์ จากนั้นผ่าน Max 232 เป็นไอซีที่ใช้แปลงระดับแรงดัน 0 โวลต์ และ 5 โวลต์ เป็น +12 โวลต์ และ -12 โวลต์ ซึ่งทำให้ไมโครคอนโทรลเลอร์สามารถเชื่อมต่อแบบอนุกรมกับคอมพิวเตอร์ได้ตามมาตรฐาน RS-232 [7]

3.3.2 วงจรภาครับสัญญาณ

ในส่วนนี้เป็นอุปกรณ์ที่เรียกว่า Microcontroller เลือกใช้ตัว AT89S52 เป็นตัวส่งสัญญาณต่อไปยังอุปกรณ์พ่นลมตามตำแหน่งที่ต้องการ โดยต้องมีการเขียนโปรแกรมเพื่อควบคุมการทำงาน Microcontroller ดังแสดงใน ภาคผนวก ง



รูปที่ 3.15 อุปกรณ์ Microcontroller

จากนั้นนำ Code โปรแกรมที่ได้ burn ลงไปใน Microcontroller มีลักษณะดังรูปที่ 3.15 โดยในการส่งสัญญาณเพื่อให้อุปกรณ์พ่นลมทำงานในแต่ละครั้ง จะส่งทั้งหมด 2 สัญญาณด้วยกัน โดยสัญญาณแรกส่งเพื่อให้อุปกรณ์พ่นลมทำงาน และอีก 1 วินาทีถัดมาจะส่งสัญญาณที่ 2 เพื่อให้อุปกรณ์พ่นลมหยุดทำงาน

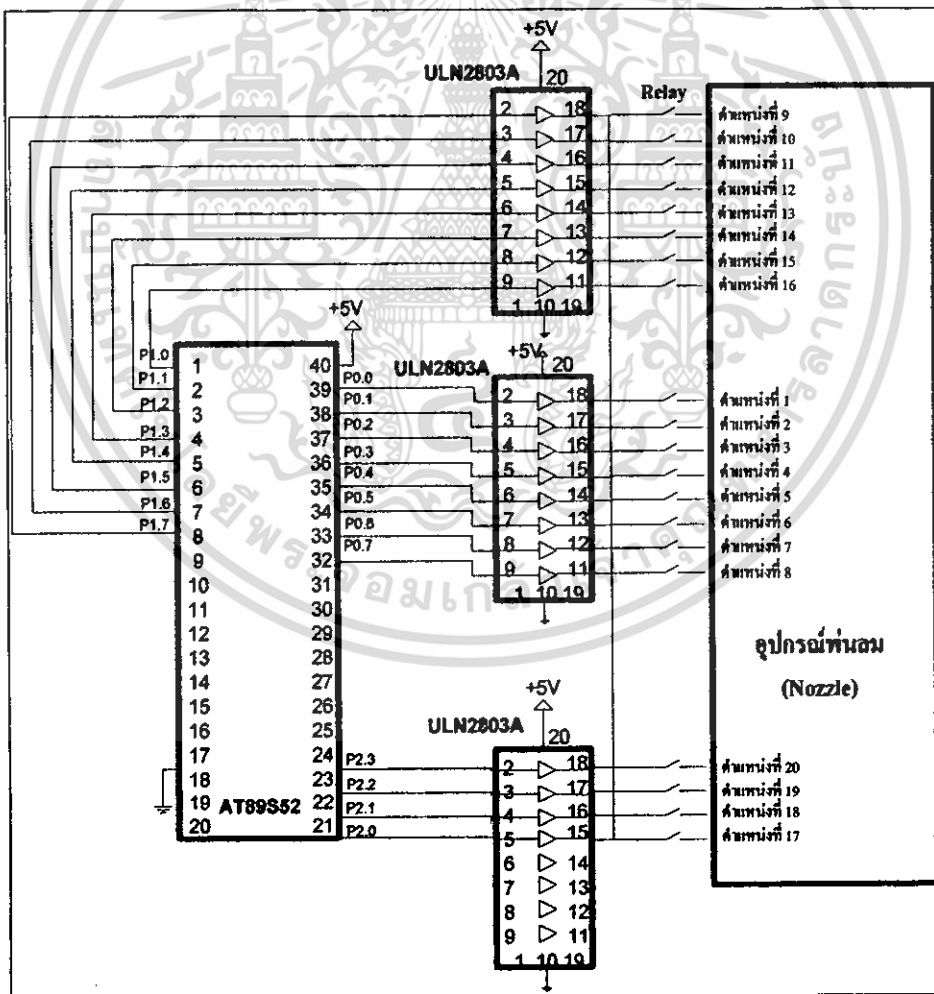
3.3.3 วงจรควบคุมภาคส่งสัญญาณและเชื่อมต่อกับอุปกรณ์พ่นลม

ในส่วนภาคส่งสัญญาณ (รูปที่ 3.16 (b)) โดยหลักการแล้วจะส่งสัญญาณออกตามขา Microcontroller โดยผ่าน ULN2803A ซึ่งมีหน้าที่เป็น driver arrays ที่รับสัญญาณลอจิกต่ำ (5 โวลต์) ที่ขาสัญญาณที่ 1-8 แล้วขับสัญญาณลอจิก (5 – 30 โวลต์) ตามกระแสที่ป้อนให้จากขาสัญญาณที่ 10 ออกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากขาสัญญาณที่ 11-18 ตามกำหนด ดังรูปที่ 3.17 ทั้ง Microcontroller (AT89S52) และ ULN2803A จะทำงานได้เมื่อมีการจ่ายไฟเลี้ยงจากหม้อแปลงตัวเดียวกัน (รูปที่ 3.16 (a))



รูปที่ 3.16 (a) หม้อแปลง
(b) วงจรควบคุมภาคส่งสัญญาณ



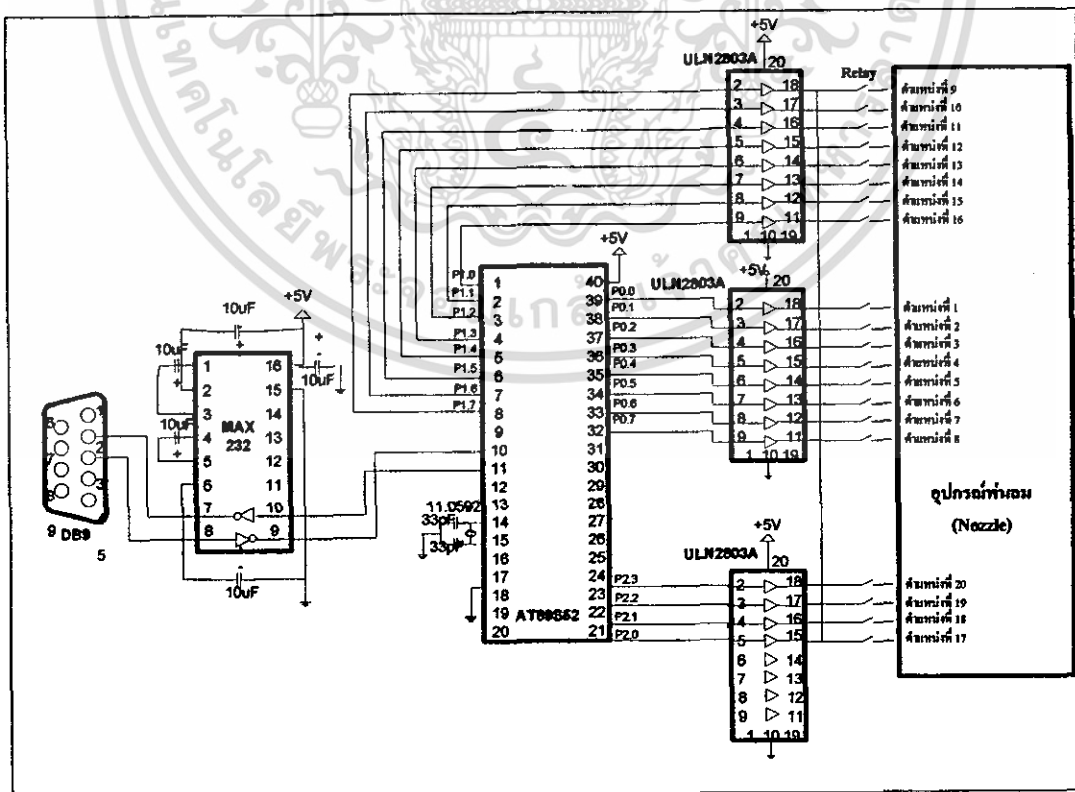
รูปที่ 3.17 วงจรควบคุมภาคส่งสัญญาณและเชื่อมต่อกับอุปกรณ์พ่นลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยขา 1 ขาจะมีความสัมพันธ์กับตำแหน่งหัวพ่นลมในแต่ละตำแหน่งดังตารางที่ 3.2 และแสดงวงจรการควบคุมสัญญาณรวมดังรูปที่ 3.18

ตารางที่ 3.2 ตำแหน่งขาที่ส่งสัญญาณตามตำแหน่งหัวพ่นลมที่ต้องการ

ตำแหน่งที่	ขาที่ส่งสัญญาณ	ตำแหน่งที่	ขาที่ส่งสัญญาณ
1	39	11	3
2	38	12	4
3	37	13	5
4	36	14	6
5	35	15	7
6	34	16	8
7	33	17	21
8	32	18	22
9	1	19	23
10	2	20	24



รูปที่ 3.18 วงจรการควบคุมสัญญาณรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบอุปกรณ์และการสร้างเครื่องกัณฑ์ขนาดและคุณภาพถั่วและถั่วป่น

การออกแบบและสร้างเครื่องกัณฑ์ขนาดและคุณภาพถั่วและถั่วป่นเพื่อการส่งออกโดยวิธีประมวลผลภาพในระดับการศึกษาเบื้องต้น ในส่วนของการออกแบบและสร้างเครื่องกัณฑ์ขนาดและคุณภาพถั่วและถั่วป่นเพื่อการส่งออกโดยวิธีประมวลผลภาพ แบ่งออกเป็นส่วนใหญ่ๆ 3 ส่วน คือ 1. ระบบสายพานลำเลียง 2. ระบบนิวเมติกส์ (Pneumatics) 3. ระบบการประมวลผลภาพ ซึ่งในแต่ละส่วนจะมีอุปกรณ์ส่วนประกอบต่างๆ ที่ใช้ในการออกแบบและสร้างเครื่องกัณฑ์ขนาดและคุณภาพถั่วและถั่วป่นเพื่อการส่งออกโดยวิธีประมวลผลภาพ และการประกอบรวมอุปกรณ์เข้าด้วยกัน เพื่อให้ได้เครื่องกัณฑ์ขนาดและคุณภาพถั่วและถั่วป่นเพื่อการส่งออกโดยวิธีประมวลผลภาพ

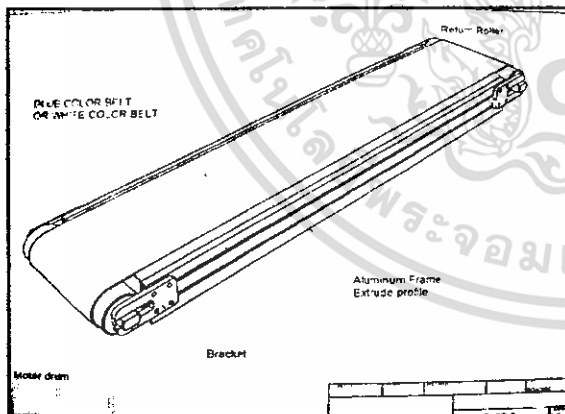
3.4.1 อุปกรณ์ที่ใช้ในการสร้างเครื่องกัณฑ์แยกถั่วและถั่วป่น

อุปกรณ์ทั้งหมดของระบบการทำงานแบ่งออกเป็น 3 ส่วนดังนี้คือ

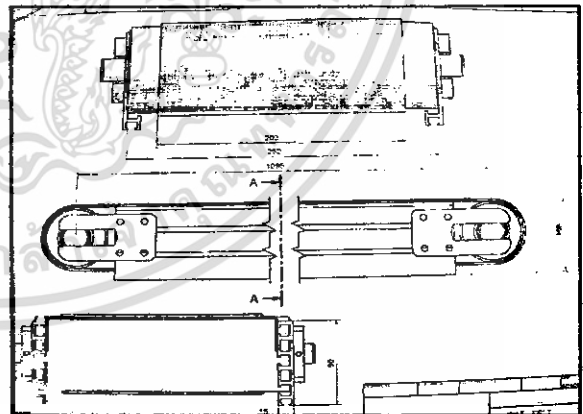
1. ชุดอุปกรณ์ระบบสายพานลำเลียง (Belt conveyer)

จำนวน 1 ชุด ซึ่งมีแบบดังรูปที่ 3.19 (a) และ (b) ซึ่งอุปกรณ์ที่ใช้ในระบบสายพานลำเลียง มีดังนี้

ก) ชุดสายพานลำเลียง (Belt conveyer) 1 ชุด ตัวสายพานที่ใช้เป็นสายพานสีขาว ขนาดหน้ากว้างสายพาน 35 เซนติเมตร ยาว 1 เมตร 35 เซนติเมตร มอเตอร์ขับเคลื่อนสายพานเป็นชนิดมอเตอร์ที่อยู่ภายในตัวล้อขับ ขนาด 0.085 กิโลวัตต์ ตัวโครงแบบสายพานทำมาจากอลูมิเนียม ความเร็วแล่นสูงสุด 0.63 เมตรต่อวินาที



(a)



(b)

รูปที่ 3.19 (a) แบบชุดสายพานลำเลียง

(b) ชุดสายพานลำเลียง

ข) อุปกรณ์ปรับความเร็วรอบ (Inverter) 1 เครื่องใช้ในการปรับความเร็วรอบของชุดสายพาน ใช้ไฟ 220 โวลต์ ซึ่งการออกแบบกำหนดให้สายพานวิ่งด้วยความเร็วสูงสุด 0.63 เมตรต่อวินาที ที่ความถี่ 50 เฮิร์ตซ์ ดังแสดงในรูปที่ 3.20

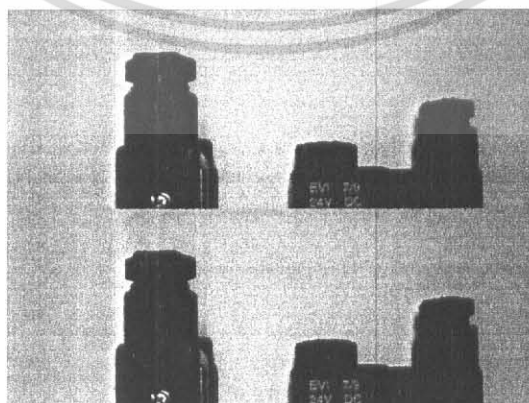


รูปที่ 3.20 อุปกรณ์ปรับความเร็วรอบ

2. ชุดอุปกรณ์ระบบนิวแมติกส์

อุปกรณ์ที่ทำหน้าที่ในส่วนของระบบนิวแมติกส์ ประกอบด้วยอุปกรณ์หลายชนิดเข้าด้วยกัน ระบบนี้จะเป็นระบบที่ทำหน้าที่รับสัญญาณมาจากชุดไมโครคอนโทรลเลอร์ สั่งการให้ nozzle ทำการปล่อยลมออกไปยังชุดหัวพ่นลมที่จะทำการยิงด้วยแรงดันอีกทีหนึ่ง ซึ่งระบบนี้จะมีอุปกรณ์ต่าง ๆ ดังนี้

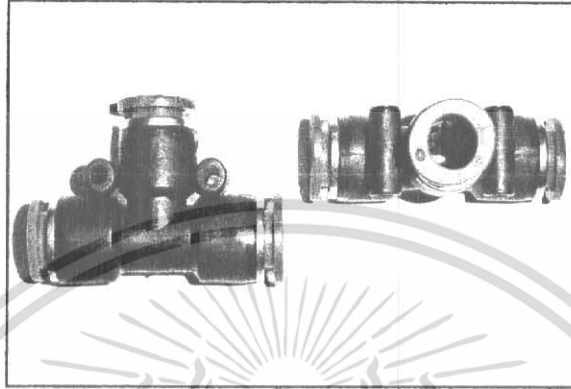
ก) หัวพ่นลม (nozzle) เป็นอุปกรณ์ที่ทำงานโดยใช้ไฟฟ้าจากเครื่อง Power Supply กระแสตรง ขนาด 24 โวลต์ ดังแสดงในรูปที่ 3.21 โดยรับสัญญาณมาจากชุดไมโครคอนโทรลเลอร์ สั่งการให้ลมถูกปล่อยออกไปโดยลมที่พ่นออกจากอุปกรณ์นี้จะต้องส่งผ่านไปยัง อุปกรณ์ชุดหัวพ่นลมก่อน เพื่อที่จะใช้ทำการยิงด้วยแรงดัน ในระบบทั้งหมดได้ใช้อุปกรณ์ชนิดนี้ทั้งหมดจำนวน 20 ตัว ซึ่งเป็นอุปกรณ์หัวพ่นลมขนาด 8 มิลลิเมตร



รูปที่ 3.21 หัวพ่นลม

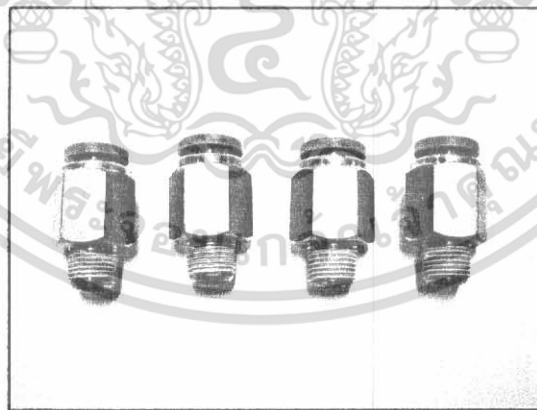
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข) **ข้อต่อ 3 ทาง** มีหน้าที่ในการแบ่งลมจากต้นทางคือถังอัดอากาศ ผ่านมายังตัว reducing valve ส่งต่อมายังชุด regulator ผ่านข้อต่อ 3 ทาง เพื่อแยกลมไปยังอุปกรณ์ชุดหัวพ่นลม จำนวน 20 ตัว ดังรูปที่ 3.22 ซึ่งในระบบได้ใช้อุปกรณ์ชนิดนี้จำนวน 20 ตัว โดยใช้อุปกรณ์เชื่อมต่อคือสายท่อลม ขนาด 8 มิลลิเมตร



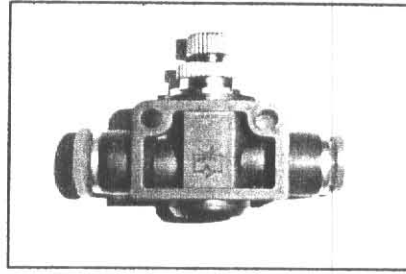
รูปที่ 3.22 ข้อต่อ 3 ทาง

ค) **อุปกรณ์ต่อตรง** มีหน้าที่ในการนำไปติดตั้งกับชุด Nozzle จำนวน 40 ตัว และติดตั้งกับชุดหัวพ่นลมอีก 20 ตัว เพื่อให้สามารถทำการเชื่อมต่อของระบบโดยใช้สายท่อลมเป็นตัวเชื่อมต่อกันของระบบ อุปกรณ์นี้เป็นอุปกรณ์ต่อตรงขนาด 8 มิลลิเมตร ใช้ต่อออกจากอุปกรณ์ชุดหัวพ่นลม และชุดหัวพ่นลม รวมจำนวนอุปกรณ์ต่อตรงที่ใช้ทั้งสิ้นจำนวน 60 ตัว ดังรูปที่ 3.23



รูปที่ 3.23 อุปกรณ์ต่อตรง

ง) **Reducing valve** มีหน้าที่ในการปรับลดแรงดันลมที่ออกจากถังอัดอากาศ ให้มีแรงดันที่เหมาะสม แต่จะเป็นการปรับอย่างหยาบเท่านั้นเพราะไม่สามารถอ่านค่าแรงดันลมได้ เพราะฉะนั้นจะต้องส่งผ่านลมนี้ไปยังชุด regulator เพื่อปรับแรงดันลมให้ละเอียด ตามที่ได้คำนวณไว้คือ 7.48 bar (หัวข้อ 3.4.4) ซึ่งในระบบนี้จะใช้อุปกรณ์ชนิดนี้จำนวน 1 ตัว เท่านั้น ดังรูปที่ 3.24 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



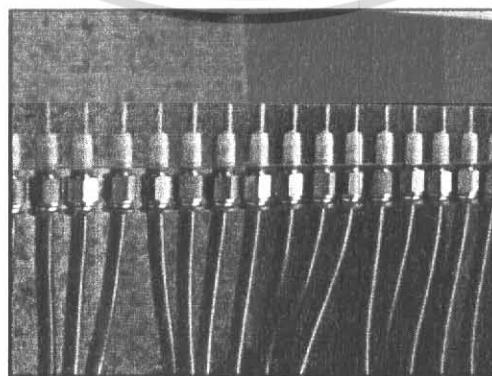
รูปที่ 3.24 Reducing valve

จ) สายท่อลม มีหน้าที่ในการเชื่อมต่ออุปกรณ์ส่วนต่าง ๆ เข้าด้วยกัน ดังนี้คือ ชุดหัวพ่นลม ชุดหัวพ่นลม ข้อต่อ 3 ทาง เครื่องอัดอากาศ regulator และ reducing valve ตามลำดับ ซึ่งจะใช้สายท่อลมที่มีขนาด 8 มิลลิเมตร แสดงดังรูปที่ 3.25



รูปที่ 3.25 สายท่อลม

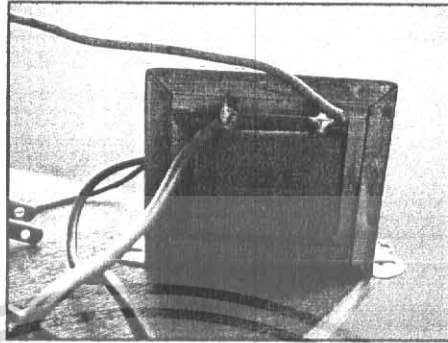
ฉ) ชุดหัวพ่นลม มีหน้าที่ในการยิงลวดและญี่ปุ่น โดยทำการรับลมทันทีที่ nozzle รับสัญญาณมาจากชุดไมโครคอนโทรลเลอร์ ลมก็จะออกไปยังชุดหัวพ่นลมทันทีที่ได้รับสัญญาณ ชุดหัวพ่นลมซึ่งทำจากแผ่นอะคริลิกใสขนาดกว้าง 3 เซนติเมตร ยาว 50 เซนติเมตร หนา 3 เซนติเมตร เจาะรูทั้งด้านบนและด้านล่าง จำนวน 20 รู เท่ากับจำนวน nozzle ซึ่งรูด้านล่างมีขนาดเส้นผ่านศูนย์กลาง 1.25 มิลลิเมตร ส่วนด้านบนทำการเจาะรูเพื่อทำการติดตั้งร่วมกับอุปกรณ์ต่อตรง ซึ่งแต่ละรูจะมีระยะห่างระหว่างรู 1.7 เซนติเมตร ซึ่งชุดหัวพ่นลมที่ได้ทำออกมามีลักษณะดังรูปที่ 3.26



รูปที่ 3.26 ชุดหัวพ่นลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข) **Power Supply** จำนวน 1 เครื่อง เป็นอุปกรณ์แปลงกระแสไฟฟ้าสลับให้เป็นกระแสตรง โดยแปลงไฟฟ้ากระแสสลับ 220 โวลต์ เป็นไฟฟ้ากระแสตรง 24 โวลต์ เพื่อทำการจ่ายกระแสไฟให้กับ nozzle จำนวน 20 ตัว ดังรูปที่ 3.27



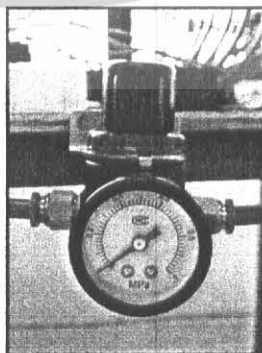
รูปที่ 3.27 Power supply

ข) **เครื่องอัดอากาศ** 1 เครื่อง ขนาด 0.5 hp แรงดันลมสูงสุด 8 bar ซึ่งนำมาใช้งานในการจ่ายลมให้กับ nozzle จำนวน 20 ตัว ผ่านชุดปรับแรงดันลม ดังรูปที่ 3.28



รูปที่ 3.28 เครื่องอัดอากาศ

ค) **ชุด Regulator** ทำหน้าที่ปรับแรงดันลมให้มีแรงดันที่เหมาะสมในการใช้งาน ซึ่งจากการคำนวณจะต้องใช้แรงดันลมที่มีขนาด 7.48 Bar (หัวข้อ 3.4.4) ดังรูปที่ 3.29



รูปที่ 3.29 Regulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ชุดอุปกรณ์ระบบการประมวลผลภาพ

ชุดอุปกรณ์ระบบการประมวลผลภาพ เป็นการนำกล้องถ่ายภาพที่มีคุณสมบัติที่มีความสามารถในการถ่ายภาพเคลื่อนที่ และอัตราการถ่ายภาพที่สูง รวมถึงความละเอียดความคมชัดของภาพ ที่จะนำไปประมวลผล อุปกรณ์ในส่วนของการประมวลผลภาพมีดังนี้คือ

ก) กล้องถ่ายภาพ CCD progressive scan จำนวน 1 ชุด กล้องถ่ายภาพที่นำมาใช้ในระบบประมวลผลภาพนี้ เป็นกล้องของบริษัท HITACHI รุ่น KP- FD32F ดังรูปที่ 3.30 ซึ่งมีคุณสมบัติเป็นกล้อง single CCD รูปแบบสีเป็นแบบ RGB และเป็นกล้องแบบ CCD progressive scan ซึ่งกล้องจะต่อกับคอมพิวเตอร์ โดยใช้ IEEE1394.b cable และ IEEE1394.b camera control software เวอร์ชัน 1.02 ซึ่งที่คอมพิวเตอร์จะติดตั้ง capture card รุ่น PCI 800 ด้วย

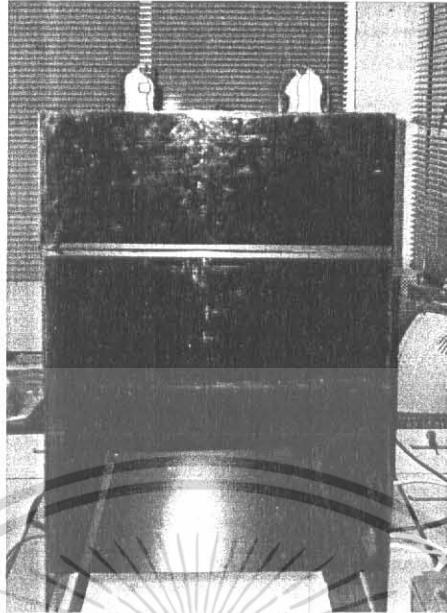


รูปที่ 3.30 กล้องถ่ายภาพ CCD progressive scan

ข) คอมพิวเตอร์ 1 เครื่อง คอมพิวเตอร์เป็นส่วนที่สำคัญในการรับภาพจากกล้องถ่ายภาพที่ถ่ายภาพ ส่งผ่านมาที่ capture card PCI 800 จากนั้นส่งต่อมาประมวลผลที่โปรแกรมที่ถูกเขียนไว้ในคอมพิวเตอร์ คุณลักษณะเครื่อง คือ ใช้ระบบปฏิบัติการ Window XP, CPU Pentium 4 2.40GHz, NVIDIA Geforce2 MX/MX 400 64MB และ RAM 768 MB

ค) Light Box อุปกรณ์นี้มีลักษณะเป็นกล่องสองชั้นโดยชั้นบนติดตั้งหลอดไฟตรงกลางจะเป็นแผ่นอะคริลิกสีขาวขุ่น ส่วนชั้นล่างติดตั้งกล้องในตำแหน่งตรงกลาง ซึ่ง Light Box จะมีลักษณะคล้ายกับอุโมงค์ ทำมาจากแผ่นอะคริลิกสีดำ หนาประมาณ 3 มิลลิเมตร มีขนาด กว้าง 50 เซนติเมตร ยาว 40 เซนติเมตร สูง 76 เซนติเมตร ซึ่งจะวางไว้บนสายพาน ให้ถั่วระอู้นุ่น ลอดผ่านไปเพื่อที่จะทำการถ่ายภาพเวลาที่สายพานเคลื่อนที่ ภายในกล่องนี้จะประกอบไปด้วยการติดตั้งกล้องไว้ตรงกลาง และทำการติดหลอดไฟเป็นหลอดตะเกียบ ขนาด 14 W จำนวน 4 หลอด เพื่อเพิ่มความสว่างให้แก่กล้องเวลาทำการถ่ายภาพ ดังแสดงในรูป 3.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



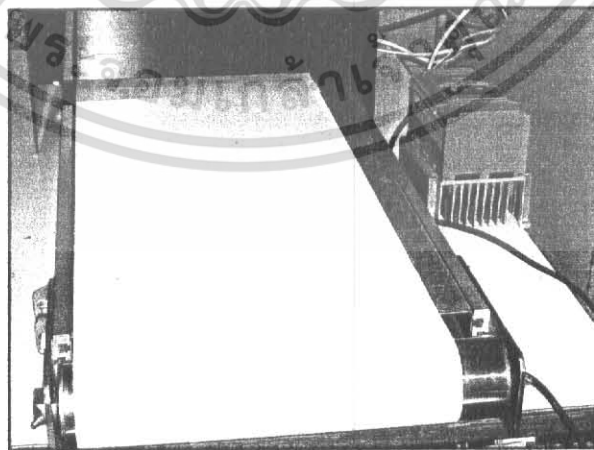
รูปที่ 3.31 Light Box

3.4.2 การติดตั้งอุปกรณ์

เมื่อเตรียมอุปกรณ์ของระบบทั้งหมดแล้ว การติดตั้งอุปกรณ์แบ่งออกเป็น 3 ส่วนดังนี้คือ 1. การติดตั้งอุปกรณ์ในส่วนของชุดอุปกรณ์สายพานลำเลียง 2. การติดตั้งอุปกรณ์ในส่วนของระบบนิวแมติกส์ 3. การติดตั้งอุปกรณ์ในส่วนของระบบการประมวลผลภาพ

1. การติดตั้งอุปกรณ์ในส่วนของชุดอุปกรณ์สายพานลำเลียง

ติดตั้งชุดอุปกรณ์สายพานลำเลียงบน โต๊ะ จากนั้นต่อสายไฟเข้ากับตัว Inverter ซึ่งจะทำได้ทำให้สามารถปรับตั้งความเร็วของสายพานได้ตามที่ต้องการ ซึ่งความเร็วสูงสุดของสายพานที่ได้คำนวณไว้มีค่าเท่ากับ 0.63 เมตรต่อวินาที (หัวข้อ 3.4.3) ดังแสดงในรูป 3.32

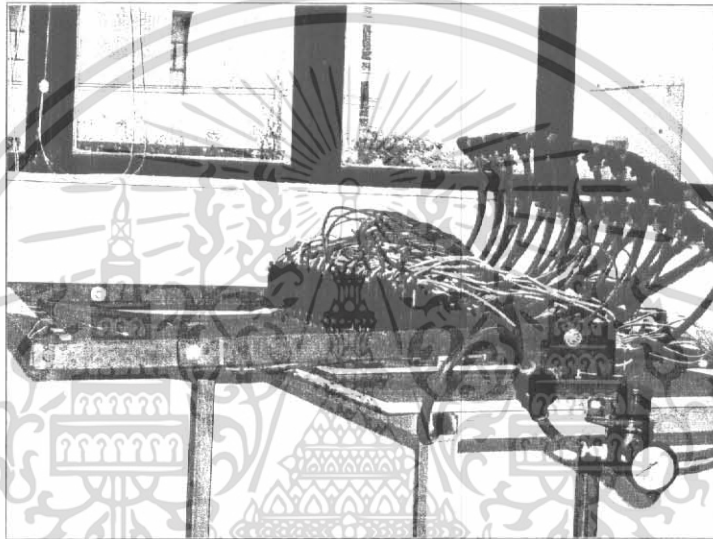


รูปที่ 3.32 ชุดอุปกรณ์สายพานลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การติดตั้งอุปกรณ์ในส่วนของระบบนิวแมติกส์

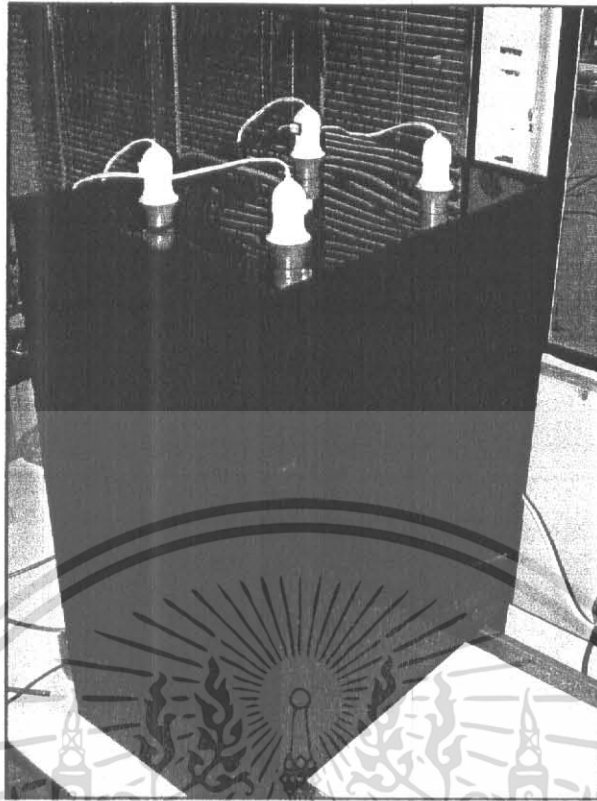
เมื่อได้อุปกรณ์ในส่วนของระบบนิวแมติกส์มาทั้งหมดแล้ว จากนั้นก็ดำเนินการประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน เริ่มตั้งแต่ ชุดหัวฟันลมนำอุปกรณ์ต่อตรงจำนวน 20 ตัว มาติดตั้งร่วมกับชุดหัวฟันลมต่อไปคือนำสายท่อลมมาเชื่อมต่อชุดหัวฟันลมเข้ากับ nozzle จำนวน 20 ตัว จากนั้นต่อสายท่อลมออกไปยังอุปกรณ์ข้อต่อ 3 ทาง จำนวน 20 ตัว เพื่อรวมสายท่อลมเพื่อรวมเป็นแหล่งจ่ายลมเดียวกัน จากนั้นต่อสายท่อลมเข้ากับตัว regulator แล้วต่อออกไปยัง reducing valve เพื่อต่อเข้ากับถังอัดอากาศ สำหรับ power supply จะทำการจ่ายไฟกระแสตรง ขนาด 24 โวลต์ ให้กับ nozzle เพื่อทำการปล่อยลมไปยังชุดหัวฟันลม การติดตั้งอุปกรณ์ในส่วนของระบบนิวแมติกส์นี้แสดงดังในรูป 3.33



รูปที่ 3.33 ชุดอุปกรณ์ระบบนิวแมติกส์

3. การติดตั้งอุปกรณ์ในส่วนของระบบการประมวลผลภาพ

เมื่อได้อุปกรณ์ในส่วนของระบบการประมวลผลภาพแล้ว จากนั้นติดตั้งอุปกรณ์ต่าง ๆ เข้าด้วยกันดังนี้ ติดตั้งกล้องถ่ายภาพไว้ภายใน light box ตรงกลาง ติดตั้งหลอดไฟจำนวน 4 หลอด ไว้ด้านบน โดยใช้แผ่นอะคริลิก สีขาวขุ่นเพื่อทำการกรองแสงให้มีความสม่ำเสมอทั่ว light box จากนั้นนำสาย IEEE1394.b cable มาเชื่อมต่อกล้องถ่ายภาพเข้ากับ capture card PCI 800 ที่ติดตั้งภายในคอมพิวเตอร์ เมื่อติดตั้งอุปกรณ์ในส่วนของระบบการประมวลผลภาพออกมาได้ดังรูป 3.34



รูปที่ 3.34 ชุดอุปกรณ์ประมวลผลภาพ

3.4.3 การคำนวณจุดถ่ายเทวัสดุออกจากหัวสายพาน [8]

สำหรับตัวระบุญี่ปุ่น การคำนวณจุดถ่ายเทวัสดุออกจากหัวสายพาน ซึ่งการถ่ายเทออกจากหัวสายพาน วัสดุจะถ่ายเทเป็นลักษณะพาราโบลา ซึ่งสามารถคำนวณหาตำแหน่ง (x, y) ต่างๆ ได้จากสมการดังนี้คือ

$$x = v_i t$$

$$y = 0.5gt^2$$

เมื่อ v_i คือ ความเร็วของวัสดุ (เมตร/วินาที)

t คือ เวลาในการเคลื่อนที่ (วินาที)

g คือ ค่านิจโน้มถ่วงของโลก (9.81 เมตร/วินาที²)

ค่าความเร็วของวัสดุสามารถคำนวณหาได้จากสมการ

$$v = R\omega$$

$$\omega = 2\pi n$$

เมื่อ R คือ รัศมีของหัวเครื่องวัดความเร็วรอบ (1.57 เซนติเมตร)

n คือ ความเร็วรอบของล้อขับสายพาน (382.5 รอบต่อนาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

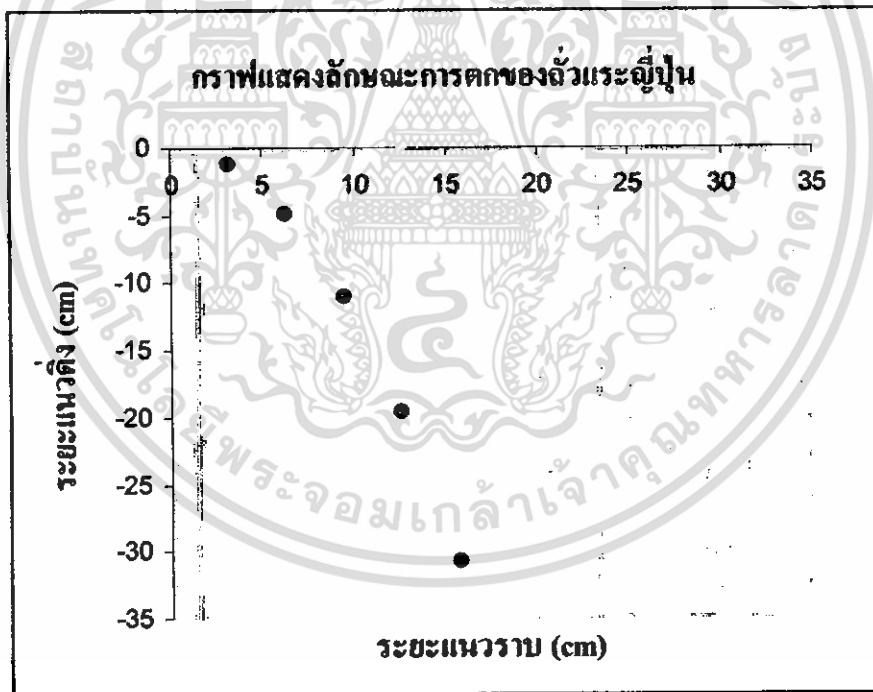
$$v = 1.57 \times 10^{-3} \times 2 \times \pi \times 382.5$$

$$v = 0.63 \text{ เมตร/วินาที}$$

ดังนั้น ที่ระยะตามแนวแกน x และระยะตามแนวแกน y ณ เวลาต่างๆทุกเวลา 0.05 วินาที สามารถคำนวณได้ดังตารางที่ 3.3 จากนั้นนำข้อมูลที่ได้นี้มาเขียนกราฟแสดงลักษณะการตกของถั่วระอู๋ปุ่น ได้ดังรูปที่ 3.35

ตารางที่ 3.3 ระยะตามแนวแกน x และระยะตามแนวแกน y ที่ตำแหน่งต่างๆ

ตำแหน่งที่	ระยะแกน x (cm)	ระยะแกน y (cm)
1	3.15	-1.23
2	6.3	-4.91
3	9.45	-11.04
4	12.6	-19.62
5	15.75	-30.66



รูปที่ 3.35 กราฟแสดงลักษณะการตกของถั่วระอู๋ปุ่น

การพิจารณาจุดที่วัตถุแยกตัวออกจากสายพาน พิจารณาได้จากระยะชั่ว (hp) และรัศมี (r) ซึ่งระยะชั่วสามารถคำนวณได้จากสูตร

$$hp = \frac{gr^2}{v^2}$$

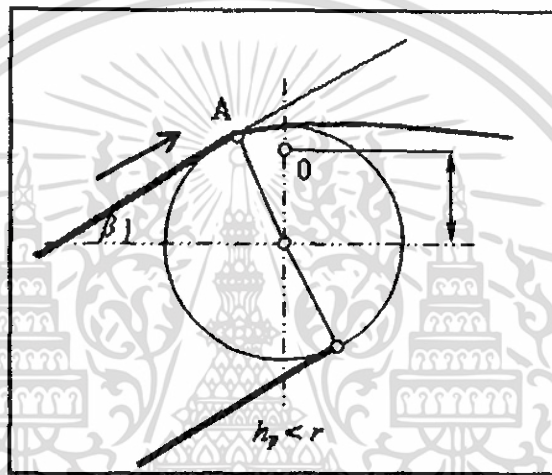
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อ g คือ ค่านิจโน้มถ่วงของโลก (9.81 เมตร/วินาที²)
 r คือ รัศมีของล้อขับ (0.04 เมตร)
 v คือ ความเร็วของสายพาน (0.63 เมตรต่อวินาที)

$$hp = \frac{9.81 \times 0.04^2}{0.63^2}$$

$$hp = 0.025$$

แต่ $r = 0.04$ ดังนั้น hp น้อยกว่า r แสดงว่าฝักถั่วแยกตัวออกจากสายพาน ใน Quadrant ที่ 2 ซึ่งมุม β ของสายพานค้ำแยกทำมุมกับแนวราบเท่ากับ 0 องศา (ดังรูปที่ 3.36)



รูปที่ 3.36 ลักษณะการเคลื่อนที่ของวัสดุบนสายพาน

3.4.4 การคำนวณแรงค้ำลมที่จะใช้ในการค้ำแยกถั่วระอู้น

ก่อนที่จะสามารถคำนวณหาแรงค้ำลมที่จะใช้ในการค้ำแยกถั่วระอู้นได้นั้นจะต้องคำนวณแรงค้ำในแนวตั้งของถั่วระอู้นก่อน จากนั้นจะต้องออกแบบมุมหรือทิศทางการยิง โดยออกแบบให้ถั่วระอู้นเมื่อถูกยิงแล้วมีแรงลัพท์ทำมุม 45 องศากับแนวราบ สามารถคำนวณหาแรงที่ถั่วระอู้นตกในแนวตั้งค้ำสมการ

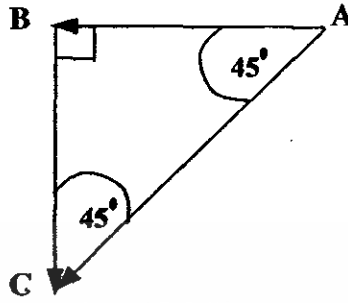
$$\vec{F} = m\vec{g} = \vec{AB}$$

โดยที่ m คือ น้ำหนักเฉลี่ยของถั่วระอู้นเท่ากับ 0.004 กิโลกรัม
 g คือ ค่านิจโน้มถ่วงของโลก (9.81 เมตร/วินาที²)

$$F = 0.004 \times 9.81 = 0.039N$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น เมื่อนำมาเขียนอัตราส่วนแรงโดยการแตกแรงให้มีแรงลัพธ์ทำมุม 45 องศา กับแนวราบ จะได้แรงที่หัวพ่นลมทำมุมยั้งคั้งฉาก 90 องศา ออกมาเท่ากับ 0.039 ดังรูปที่ 3.37



รูปที่ 3.37 การออกแบบทิศทางการยิงของหัวกระสุนเมื่อถูกยิง

และคำนวณหาแรงดันลมได้จาก

$$P = F/A$$

โดยที่ F เท่ากับ 0.039 N

A คือ พื้นที่วงกลมที่ขนาดเส้นผ่านศูนย์กลาง (D) 1.25 มิลลิเมตร

$$P = \frac{0.039}{\frac{\pi D^2}{4}}$$

$$P = 24960 \text{ Pascal}$$

$$P = 0.24960 \text{ Bar}$$

จากการคำนวณที่ได้คือความดันลมที่ใช้ต่อ 1 หัวพ่นลม แต่ทั้งระบบมีจำนวนมีจำนวนหัวพ่นลมทั้งหมด 20 หัว ดังนั้นต้องใช้แรงดันลมทั้งหมดโดยคิดค่า safety factor เท่ากับ 50% จะเท่ากับ 7.48 Bar

3.4.5 การคำนวณอัตราเร็วในการถ่ายภาพ

เนื่องจากการออกแบบต้องการให้กล้องสามารถถ่ายภาพได้ครอบคลุมถึงด้านข้างของสายพาน ซึ่งจะต้องติดตั้งกล้องที่ระยะความสูง 76.5 เซนติเมตรจากผิวสายพาน และจะสามารถถ่ายภาพได้ในระยะทาง 0.48 เมตร ดังนั้น สามารถคำนวณหาความเร็วของสายพานเพื่อให้ตรงกับอัตราในการถ่ายภาพได้ คือ ที่อัตราการถ่ายภาพ 1.875 frame/s

1.875	frame	ใช้เวลา	1	วินาที
1	frame	ใช้เวลา	$(1 \times 1) / 1.875$	= 0.533 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาเทียบกับความเร็วสายพานจะได้

1 วินาที ได้ระยะทางสายพาน 0.63 เมตร

0.533 วินาที ได้ระยะทาง $(0.533 \times 0.63) / 1 = 0.336$ เมตร

ดังนั้นควรติดตั้งกล้อง CCD progressive scan ไว้ที่ระดับความสูง 76.5 เซนติเมตร เพื่อให้ได้ระยะในการถ่ายภาพ 0.336 เมตร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดสอบและผลการทดสอบ

4.1 สีของถั่วระอุญี่ปุ่น

ใช้โปรแกรม Finding Color Bean หาค่าเฉลี่ยของสีถั่วระอุญี่ปุ่นที่ต้องการ จากนั้นนำค่าสี hue เฉลี่ยมาเปรียบเทียบกับหาค่าสีเรด โคลด์ ได้ผลดังตารางที่ 4.1

ตารางที่ 4.1 ค่าสี Hue เฉลี่ยของถั่วระอุญี่ปุ่นทั้ง 30 ฝัก

ช่วงของสีถั่ว	ถั่วระอุญี่ปุ่นฝักที่	ค่าสี Hue เฉลี่ย
สีถั่วที่ไม่ต้องการ	1	0.1971
	2	0.2003
	3	0.2048
	4	0.2050
	5	0.2107
	6	0.2143
	7	0.2223
	8	0.2236
	9	0.2257
	10	0.2268
	11	0.2313
	12	0.2319
	13	0.2321
	14	0.2324
	15	0.2380
สีถั่วที่ต้องการ	1	0.2550
	2	0.2673
	3	0.2677
	4	0.2684

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 (ต่อ)

ช่วงของสีข้าว	ตัวระบุสีข้าว	ค่าสี Hue เฉลี่ย
	5	0.2694
	6	0.2696
	7	0.2705
	8	0.2717
	9	0.2722
	10	0.2727
	11	0.2747
	12	0.2761
	13	0.2781
	14	0.2786
	15	0.2851

จากตารางจะเห็นว่าค่าสี Hue เฉลี่ยของสีที่ต้องการอยู่ในช่วงตั้งแต่ 0.2550 ขึ้นไป แต่เนื่องจากค่า Hue ที่ได้จากโปรแกรม Finding Color Bean จะนำมาประมวลผลในโปรแกรมได้นั้นต้องนำมาคูณ 180* เสียก่อน จะได้ $0.2550 \times 180 = 45.9$ ฉะนั้นการทดลองนี้จึงตั้งค่าสีเรดโซลด์ (Color Threshold) ไว้ที่ 45.9 ดังนั้นตัวระบุสีข้าวใดที่มีค่าสี Hue เฉลี่ยต่ำกว่าค่าสีเรดโซลด์ ก็จะถูกคัดแยกออกไป

4.2 ขนาดของตัวระบุสีข้าว

หาขนาดพื้นที่ของตัวระบุสีข้าวที่ต้องการจากโปรแกรม GBA จากนั้นนำค่าขนาดที่ได้มาเปรียบเทียบกับหาขนาดเรดโซลด์ (Size Threshold) ได้ผลดังตารางที่ 4.2

จากตารางจะเห็นว่าขนาดของตัวระบุสีข้าวที่ต้องการอยู่ในช่วง 6.8 ตร.ซม. ขึ้นไป ค่าขนาดเรดโซลด์จึงมีค่าเท่ากับ 6.8 ตร.ซม. ดังนั้นตัวระบุสีข้าวใดที่มีขนาดน้อยกว่าค่าขนาดเรดโซลด์ ก็จะถูกคัดแยกออกไป

หมายเหตุ * ค่าสี Hue จากโปรแกรม Mat lab มีสเกลแตกต่างจากค่าสี Hue ที่ได้จากโปรแกรมภาษา C++ ถ้านำค่าสี Hue จากโปรแกรม Mat lab มาคิดในโปรแกรมภาษา C++ ต้องนำค่านั้นมาคูณด้วย 180 จึงเป็นสเกลเดียวกับโปรแกรมภาษา C++

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ขนาดของถั่วระอู๋ปุ่นโดยใช้โปรแกรมทั้ง 30 ฝึก

ช่วงของขนาดถั่ว	ถั่วระอู๋ปุ่นฝึกที่	ขนาดของถั่วระอู๋ปุ่น (ตร.จม.)
ขนาดถั่วที่ไม่ต้องการ	1	4.8756
	2	5.0332
	3	5.2496
	4	5.3354
	5	5.4332
	6	5.5266
	7	5.559
	8	5.7681
	9	6.3354
	10	6.3565
	11	6.4982
	12	6.5474
	13	6.6002
	14	6.655
	15	6.7252
ขนาดถั่วที่ต้องการ	1	6.9326
	2	7.2930
	3	7.7044
	4	7.3644
	5	7.0448
	6	7.4426
	7	6.9734
	8	7.6605
	9	7.2374
	10	7.8032
	11	7.5331
	12	7.0958

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ)

ช่วงของขนาดถั่ว	ถั่วระอู๋ปุ่นฝักที่	ขนาดของถั่วระอู๋ปุ่น (กร.ซม.)
	13	6.9848
	14	7.3502
	15	7.4448

4.3 สมรรถนะของเครื่องคัด

4.3.1 การทำงาน

การทำงานของเครื่องคัดขนาดและคุณภาพสีถั่วระอู๋ปุ่น เริ่มต้นโดยรับภาพจากกล้อง CCD progressive scan มาวิเคราะห์ผลที่โปรแกรม GBA ซึ่งภายในโปรแกรมประกอบด้วย 2 ฟังก์ชัน ได้แก่ ฟังก์ชันคัดสี และฟังก์ชันคัดขนาด ซึ่งสามารถเลือกทำงานฟังก์ชันใดฟังก์ชันหนึ่ง หรือสามารถทำทั้งสอง ฟังก์ชันพร้อมกัน โปรแกรมวิเคราะห์ผลภาพถั่วฝักที่ไม่ต้องการและตำแหน่งของหัวพ่นลมที่ใช้ยิงฝักเสีย นั้น ๆ จากนั้นจะส่งสัญญาณสั่งไมโครคอนโทรลเลอร์ ซึ่งไมโครคอนโทรลเลอร์จะสั่งให้หัวพ่นลมยิงฝักที่ไม่ต้องการนั้นออกลงสู่ถังเสีย

4.3.2 ความสามารถ

เครื่องคัดขนาดและคุณภาพสีถั่วระอู๋ปุ่น มีความสามารถในการรับภาพจากกล้อง CCD progressive scan มาประมวลผลภาพแล้วส่งสัญญาณไปที่ชุดไมโครคอนโทรลเลอร์ เพื่อสั่งให้ชุดอุปกรณ์ หัวพ่นลมทำงาน โดยยิงฝักเสียที่ไม่ต้องการ ซึ่งในการคำนวณความสามารถของเครื่องคัดขนาดและคุณภาพสีของถั่วระอู๋ปุ่น สามารถหาได้โดยพิจารณาจากความสามารถในการถ่ายภาพ 1 ภาพมีจำนวนฝักถั่วสูงสุด 17 ฝัก และเวลาที่ใช้ในการถ่ายภาพ 1 ภาพใช้เวลาประมาณ 0.533 วินาที ดังนั้นความสามารถของเครื่องคัดขนาดและคุณภาพสีของถั่วระอู๋ปุ่น สามารถคัดได้ 11,482 ฝักต่อชั่วโมง

4.3.3 ประสิทธิภาพ

การทำงานของระบบเครื่องคัดแยกขนาดและคุณภาพสีของถั่วระอู๋ปุ่น 1) ประสิทธิภาพการคัดแยกโดยใช้การประมวลผลภาพโดยใช้โปรแกรม GBA สามารถคัดฝักถั่วที่ไม่ต้องการออกได้ถูกต้อง 100 เปอร์เซ็นต์ การคำนวณตำแหน่งของหัวพ่นลม คำนวณได้ถูกต้อง 100 เปอร์เซ็นต์ ซึ่งมีความเร็วในการประมวลผลเฉลี่ย 0.68 วินาที ส่วน Microcontroller ทำงานตามที่โปรแกรมสั่งถูกต้อง 100 เปอร์เซ็นต์ 2) ประสิทธิภาพการทำงานของเครื่องคัดขนาดและคุณภาพสีถั่วระอู๋ปุ่น สามารถหาลักษณะการวางตัวของฝักถั่วบนสายพานลำเลียงได้ โดยทำการทดสอบ 2 ลักษณะคือ ตามแนวการเคลื่อนที่ของสายพานและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขวางแนวการเคลื่อนที่ของสายพาน พบว่า ฝักขนาดเล็กและขนาดกลางควรวางฝักถั่วตามแนวการเคลื่อนที่ของสายพาน จึงจะทำให้มีประสิทธิภาพในการคัดแยกได้ถูกต้อง 90% และ 87.5% ตามลำดับ ส่วนฝักขนาดใหญ่ควรวางฝักถั่วขวางแนวการเคลื่อนที่ของสายพาน จึงจะทำให้มีประสิทธิภาพในการคัดแยกได้ถูกต้อง 75 % แสดงดังตารางที่ 4.3 และ 4.4

ตารางที่ 4.3 ประสิทธิภาพในการคัดแยกสูงสุด

ประสิทธิภาพในการคัดแยกสูงสุด (ใช้ความดันลมสูงสุด)					
ลักษณะการปล่อยเมล็ด	ขนาดฝักถั่ว	พื้นที่ภาพฉายเฉลี่ย (ซม) ²	จำนวนฝัก (ฝัก)	จำนวนฝักของถั่วเสีย (%)	
				ช่องทางออกฝักดี	ช่องทางออกฝักเสีย
ตามแนวการเคลื่อนที่ของสายพาน	เล็ก	4.62	30	10	90
	กลาง	5.51	24	12.5	87.5
	ใหญ่	6.68	20	85	15
ตามแนวขวางการเคลื่อนที่ของสายพาน	เล็ก	4.62	30	20	80
	กลาง	5.51	24	50	50
	ใหญ่	6.68	20	25	75

ตารางที่ 4.4 ประสิทธิภาพในการคัดแยกต่ำสุด

ประสิทธิภาพในการคัดแยกสูงสุด (ใช้ความดันลมสูงสุด)					
ลักษณะการปล่อยเมล็ด	ขนาดฝักถั่ว	พื้นที่ภาพฉายเฉลี่ย (ซม) ²	จำนวนฝัก (ฝัก)	จำนวนฝักของถั่วเสีย (%)	
				ช่องทางออกฝักดี	ช่องทางออกฝักเสีย
ตามแนวการเคลื่อนที่ของสายพาน	เล็ก	4.62	30	93	7
	กลาง	5.51	24	95	5
	ใหญ่	6.68	20	75	25
ตามแนวขวางการเคลื่อนที่ของสายพาน	เล็ก	4.62	30	80	20
	กลาง	5.51	24	91	9
	ใหญ่	6.68	20	70	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการคำนวณประสิทธิภาพในการคัดแยก

ตัวอย่างการคำนวณประสิทธิภาพในการคัดแยกสูงสุดสำหรับฝักถั่วขนาดเล็กและป้อนถั่วในลักษณะตามแนวการเคลื่อนที่ของสายพาน สามารถคำนวณหาเปอร์เซ็นต์การคัดได้ดังนี้

$$\begin{aligned} \text{เปอร์เซ็นต์การคัดถูกต้อง} &= \frac{\text{จำนวนถั่วเหลืองฝักสดที่คัดได้ตามต้องการ}}{\text{จำนวนถั่วเหลืองฝักสดทั้งหมด}} \times 100 \\ &= \frac{27}{30} \times 100 \\ &= 90 \text{ เปอร์เซ็นต์} \end{aligned}$$

$$\begin{aligned} \text{เปอร์เซ็นต์การคัดผิดพลาด} &= \frac{\text{จำนวนถั่วเหลืองฝักสดที่คัดผิดพลาด}}{\text{จำนวนถั่วเหลืองฝักสดทั้งหมด}} \times 100 \\ &= \frac{3}{30} \times 100 \\ &= 10 \text{ เปอร์เซ็นต์} \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 บทสรุป

การออกแบบและสร้างเครื่องคัดขนาดและคุณภาพของถั่วระเฉีปุ่นเพื่อการส่งออกโดยใช้โปรแกรม GBA ในการวิเคราะห์ภาพ ได้ทำการเขียนและพัฒนาโปรแกรมที่ใช้ในการคัดแยกขนาดและสีของถั่วระเฉีปุ่น รวมถึงการออกแบบและสร้างเครื่องคัดขนาดและคุณภาพสีของถั่วระเฉีปุ่น สามารถสรุปผลการทดสอบที่ได้ออกมาดังนี้คือ

ประสิทธิภาพการคัดแยกโดยประมวลผลภาพ(โปรแกรม GBA) สามารถคัดฝักถั่วที่ไม่ต้องการออกได้ถูกต้อง 100 เปอร์เซ็นต์ อีกทั้งยังคำนวณตำแหน่งของหัวพ่นลม คำนวณได้ถูกต้อง 100 เปอร์เซ็นต์ และส่งสัญญาณให้ Microcontroller ทำงานตามที่โปรแกรมสั่งได้ถูกต้อง 100 เปอร์เซ็นต์

ประสิทธิภาพของอุปกรณ์ที่ใช้ในการคัดแยก ซึ่งได้แบ่งแนวทดสอบการเคลื่อนที่ของฝักถั่วออกเป็น 2 แนว คือ ตามแนวการเคลื่อนที่และตามขวางการเคลื่อนที่สายพาน และแบ่งขนาดของเมล็ดถั่วออกเป็น 3 ขนาดคือ เล็ก กลาง ใหญ่ จะเห็นได้ว่าขนาดฝักเล็กและขนาดกลางควรวีอนในตามแนวการเคลื่อนที่สายพานเพื่อให้มีประสิทธิภาพการคัดแยกสูงสุดคือ คัดถูกต้อง 90 เปอร์เซ็นต์ 87.5 เปอร์เซ็นต์ ตามลำดับ ส่วนขนาดใหญ่วางในลักษณะขวางการเคลื่อนที่ของสายพานเพื่อให้มีประสิทธิภาพการคัดแยกสูงสุดคือ คัดถูกต้อง 75 เปอร์เซ็นต์

5.2 วิจารณ์

1. โปรแกรม GBA ไม่สามารถคำนวณเวลาในการสั่งหัวพ่นลมได้อย่างแน่นอน เนื่องจากการรันโปรแกรมบน ระบบปฏิบัติการ Window XP ซึ่งระบบปฏิบัติการนี้มีกระบวนการอื่นที่ไม่เกี่ยวข้องกับโปรแกรม GBA รันอยู่ตลอดเวลา ทำให้เวลาในการประมวลผลของโปรแกรมเปลี่ยนแปลงตลอดเวลา จึงไม่สามารถคำนวณเวลาในการสั่งหัวพ่นลมได้อย่างแน่นอน วิธีการแก้ไข คือติดตั้งเซ็นเซอร์ ในระยะก่อนที่ถั่วจะถูกยิงโดยหัวพ่นลม เมื่อถั่วเคลื่อนที่มาถึงเซ็นเซอร์ ให้เซ็นเซอร์สั่งให้ชุดอุปกรณ์ไมโครคอนโทรลเลอร์ทำงานสั่งหัวพ่นลมให้ยิงฝักถั่วที่ไม่ต้องการ ซึ่งในขณะที่นั้นชุดอุปกรณ์ไมโครคอนโทรลเลอร์ต้องได้รับคำสั่งสัญญาณตำแหน่งของหัวพ่นลมที่จะใช้ยิงฝักถั่วจากโปรแกรม GBA เสียก่อน

2. โปรแกรม camara สามารถดึงภาพจากกล้อง CCD progressive scan ได้แต่สีของภาพที่ได้มีสีไม่ตรงกับภาพจริง ดังนั้นจำเป็นต้องพัฒนาโปรแกรมต่อไป ส่วนการทดลองจริงทั้งหมดในปริญญา นีพนธ์เล่มนี้ได้ใช้โปรแกรมของกล้อง CCD progressive scan ดึงภาพถั่วมาวิเคราะห์แทน

3. ส่วนของ Microcontroller จะสามารถส่งสัญญาณให้ชุดอุปกรณ์พ่นลมทำได้ทีละ 1 ครั้ง โดยหัวพ่นลมจะปล่อยลมนาน 1 วินาที ถึงจะทำงานในหัวถัดไปได้ ในบางครั้งถ้าต้องทำงานพร้อมกัน 2 ตำแหน่งจะไม่สามารถทำงานได้ ในส่วนตรงนี้ต้องมีการแก้ไขในส่วนของ code โปรแกรมต่อไป

4. ความดันลมต่ำเกินกว่าที่กำหนด เนื่องจากมีการสูญเสียในส่วนต่างๆของอุปกรณ์ และสายต่อเพื่อปล่อยลมนั้นอยู่ห่างมากเกินไป เพราะฉะนั้นในส่วนนี้จะมีการแก้ไขขนาดท่อเมนที่ต่อกับถังลมให้มีขนาดใหญ่ขึ้น



เอกสารอ้างอิง

1. Customs Department (2006) Import statistic, frozen Edamame Japanese Ministry of Finance, http://www.customs.go.jp/toukei/info/index_e.htm
http://www.moac-info.net/ict/strategy/strategy_detail.php?id=0003
2. ณัฐพล แสงจันทร์ และดาวพระศุกร์ นุ่มเอี่ยม. 2548. “การศึกษาความเป็นไปได้ในการคัดแยกถั่วเหลืองโดยใช้กระบวนการประมวลผลภาพ.” ปรียญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเกษตร, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
3. รศ.สมยศ จุณณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์, ภาควิชาโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4. Chalidabhong, T., Sirisomboon, P. and Yimyam, P., 2006, “2D/3D Vision-Based Mango’s Feature Extraction and Sorting”, 9th International Conference on Control, Automation, Robotics and Vision. (ICARCV 2006), 5-8 December 2006 Grand Hyatt Singapore.
5. พูนพัฒน์ พูนน้อย และ อัมพวัน ต้นสกุล. 2548. “เครื่องคัดมะม่วงโดยใช้ระบบ Machine Vision.”, วารสารวิจัยและพัฒนา มจร. . 28(1) : 43-58.
6. Aleixos, N., Blasco, J., Navarron, F., Molto, E.. 2001. “Multispectral inspection of citrus in real-time using machine vision and digital signal processors.”
7. IRPUS, (23 กุมภาพันธ์ 2550), Max 232 : ชื่อโครงการ “ระบบติดตามยานพาหนะจีพีเอสบนโครงข่ายจีเอสเอ็ม”, http://www.irpus.org/project_file2547_2006-08-22_FE0151-47.pdf
8. Spivakovsky, A. and V. Dyachlov, 1985. Conveying machines, Volume I. Mir publishers, Moscow.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ก ส่วนของโปรแกรม GBA

ข ส่วนของโปรแกรม camera

ค ส่วนของโปรแกรม Finding Color Bean

ง ส่วนของโปรแกรม ควบคุม Microcontroller

จ ตารางแสดงผลการทำงานของโปรแกรม GBA

ฉ ตารางพื้นที่ภาพของถั่วแต่ละขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
//{{AFX_DATA_INIT(CAboutDlg)
//}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
CDialog::DoDataExchange(pDX);
//{{AFX_DATA_MAP(CAboutDlg)
//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
//{{AFX_MSG_MAP(CAboutDlg)
// No message handlers
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CGBADlg dialog

CGBADlg::CGBADlg(CWnd* pParent /*=NULL*/)
: CDialog(CGBADlg::IDD, pParent)
{
//{{AFX_DATA_INIT(CGBADlg)
m_mean = 44.532f;
m_std = 1.71f;
m_colorTh = 45.9f;
m_sizeTh = 6.8f;
m_checkColor = FALSE;
m_checkSize = FALSE;
//}}AFX_DATA_INIT
// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CGBADlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CGBADlg)
    DDX_Control(pDX, IDC_LIST2, m_listTime);
    DDX_Control(pDX, IDC_LIST1, m_listCen);
    DDX_Text(pDX, IDC_EDIT1_MEAN, m_mean);
    DDX_Text(pDX, IDC_EDIT2_STD, m_std);
    DDX_Text(pDX, IDC_EDIT3_COLOR_TH, m_colorTh);
    DDX_Text(pDX, IDC_EDIT4_SIZE_TH, m_sizeTh);
    DDX_Check(pDX, IDC_CHECK1_COLOR, m_checkColor);
    DDX_Check(pDX, IDC_CHECK2_SIZE, m_checkSize);
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CGBADlg, CDialog)
   //{{AFX_MSG_MAP(CGBADlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGON()
    ON_BN_CLICKED(IDCOK, OnStrat)
    ON_WM_TIMER()
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CGBADlg message handlers

BOOL CGBADlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);           // Set big icon
    SetIcon(m_hIcon, FALSE);        // Set small icon
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// TODO: Add extra initialization here

return TRUE; // return TRUE unless you set the focus to a control
}

void CGBADlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CGBADlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (LPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CGBADlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CGBADlg::OnStrat()
{
    IplImage *p_img;
    IplImage *p_outcolor;
    IplImage *p_outsize;
    int *xC[MAX],*yC[MAX],*xS[MAX], *yS[MAX];
    int *noz[MAX],*ncheck[MAX];
    int *tC[MAX],*tS[MAX];
    int i;

    for(i=0;i<=MAX;i++)
    {
        xC[i] =0;
        yC[i] =0;
        xS[i] =0;
        yS[i] =0;
        tC[i] =0;
        tS[i] =0;
        noz[i]=0;
    }

    // Read image
    p_img = cvLoadImage("D:\\Test_bean\\9.bmp",1);

    //Time elapsee in microsecond
    //////////////////////////////////////
    double dTick = cvGetTickFrequency();
    double dBegin = cvGetTickCount();

    // Background subtraction
    //////////////////////////////////////
    IplImage *p_hue,*p_seg,*p_mor,*p_spare,*p_pre;

    int width =640;
    int height = 480;
    int x,y,chan1,chan2;
    unsigned char p;
    p_hue = cvCreateImage(cvSize(width,height), 8, 3);
    cvCvtColor( p_img, p_hue, CV_BGR2HSV);

    p_outcolor = cvCreateImage(cvSize(width,height), 8, 1);
    p_outsize = cvCreateImage(cvSize(width,height), 8, 1);

    //Converts color
    //////////////////////////////////////
    p_seg = cvCreateImage(cvSize(width,height), 8, 1);
    UpdateData(TRUE);
    float mean = m_mean;

```

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float std = m_std;

for(x=0;x<p_hue->width;x++)
    for(y=0;y<p_hue->height;y++)
    {
        chan1 = ((p_hue->width)*y+x)*p_hue->nChannels;
        chan2 = ((p_seg->width)*y+x)*p_seg->nChannels;
        p = (unsigned char)p_hue->imageData[chan1];

        if((abs(p - mean)) <= (K*std))
        {
            p_seg->imageData[chan2] = 255;
            p_seg->imageData[chan2+1] = 255;
            p_seg->imageData[chan2+2] = 255;
        }
        else
        {
            p_seg->imageData[chan2] = 0;
            p_seg->imageData[chan2+1] = 0;
            p_seg->imageData[chan2+2] = 0;
        }
    }

// Clean noise (Morphology)
//////////
p_mor = cvCreateImage(cvSize(width,height), 8, 1);
p_spare = cvCreateImage(cvSize(width,height), 8, 1);

IplConvKernel *kernel;
kernel= cvCreateStructuringElementEx(5,5,2,2,CV_SHAPE_ELLIPSE,NULL);
cvMorphologyEx( p_seg, p_mor, p_spare, kernel, CV_MOP_OPEN,1);

//Connected component
//////////
CvConnectedComp comp;
CvScalar newval;
CvScalar lo_dif = cvScalar(0);
CvScalar up_dif = cvScalar(0);
newval.val[0]= 0;

for(x=0;x<p_mor->width;x++)
    for(y=0;y<p_mor->height;y++)
    {
        chan1 = ((p_mor->width)*y+x)*p_mor->nChannels;
        p = p_mor->imageData[chan1];

        if( p == 255)
        {
            CvPoint seed = cvPoint(x,y);
            cvFloodFill(p_mor,seed,newval,lo_dif,up_dif,&comp,8);
            newval.val[0] += 1;
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

int ncon;
int count[MAX];

for (i=0;i<MAX;i++)
    count[i] = 0;

for(x=0;x<p_mor->width;x++)
    for(y=0;y<p_mor->height;y++)
    {
        chan1 = ((p_mor->width)*y+x)*p_mor->nChannels;
        p = p_mor->imageData[chan1];

        for (ncon = 0; ncon < MAX; ncon++)
        {
            if (p == ncon)
                count[ncon] ++;
        }
    }

float size[MAX];
for (i=0;i<MAX;i++)
    size[i] = count[i]*PIX;
for (i=0;i<MAX;i++)
{
    if (size[i] < 0.5) //get rid of un-wanted thing
    {
        for(x=0;x<p_mor->width;x++)
            for(y=0;y<p_mor->height;y++)
            {
                chan1 = ((p_mor->width)*y+x)*p_mor-
>nChannels;
                p = p_mor->imageData[chan1];

                if (p == i)
                    p_mor->imageData[chan1] = 0;
            }
        }
    }

for(x=0;x<p_mor->width;x++)
    for(y=0;y<p_mor->height;y++)
    {
        chan1 = ((p_mor->width)*y+x)*p_mor-
>nChannels;
        p = p_mor->imageData[chan1];

        if (p != 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
p_mor->imageData[chan1] = 255;

// Show image before grading
////////////////////
// cvNamedWindow ("Image", 1);
// cvShowImage ("Image", p_mor);

// Color grading
////////////////////
ColorGrading(p_img, p_mor, p_hue, p_outcolor, xC, yC) ;

// Size grading
////////////////////
SizeGrading(p_outcolor, p_outsize, xS, yS) ;

//Calculate Nozzle
////////////////////
CalculateNozzle(xC, yC, xS, yS, tC, tS, noz, ncheck);

//SendSignal to Nozzle
////////////////////
SendSignaltoNozzle(tC, tS, noz, dBegin, dTick, ncheck);
}

void CGBADlg::ColorGrading(IplImage *p_in, IplImage *p_mor, IplImage *p_hue, IplImage
*p_out, int *centroid_x[], int *centroid_y[])
{
UpdateData(TRUE);
if (m_checkColor == FALSE)
{
p_out = p_mor;
//AfxMessageBox("No Color grading");
}
else
{
int width = 640;
int height = 480;
int x,y,chan1,chan2,chan3,i;
unsigned char p, q, r;
int count[MAX];

CvConnectedComp comp;
CvScalar newval;
CvScalar lo_dif = cvScalar(0);
CvScalar up_dif = cvScalar(0);
newval.val[0]= 0;

```

```

IplImage *c_pro = cvCreateImage(cvSize(width,height), 8, 1);
IplImage *c_out = cvCreateImage(cvSize(width,height), 8, 1);

cvCopy(p_mor, c_pro, NULL);
cvCopy(p_mor, c_out, NULL);

for(x=0;x<c_pro->width;x++)
    for(y=0;y<c_pro->height;y++)
    {
        chan1 = ((c_pro->width)*y+x)*c_pro->nChannels;
        p = c_pro->imageData[chan1];

        if (p == 255)
        {
            CvPoint seed = cvPoint(x,y);
            cvFloodFill(c_pro,seed,newval,lo_dif,up_dif,&comp,8);
            newval.val[0] += 1;
        }
    }

//*****//
AfxMessageBox("Color grading is processing");
//*****//

float hue[MAX];
for (i=0;i<MAX;i++)
{
    count[i] = 0;
    hue[i] = 0;
}

for(x=0;x<c_pro->width;x++)
    for(y=0;y<c_pro->height;y++)
    {
        chan1 = ((c_pro->width)*y+x)*c_pro-
>nChannels;
        chan2 = ((p_hue->width)*y+x)*p_hue-
>nChannels;

        p = c_pro->imageData[chan1];
        q = p_hue->imageData[chan2];

        for (int ncon = 0; ncon < MAX; ncon++)
        {
            if (p == ncon)
            {
                count[ncon] ++;
                hue[ncon] += q;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

float avghue[MAX];

UpdateData(TRUE);
for (i=0;i<MAX;i++)
    avghue[i] = hue[i]/count[i];

for (i=0;i<MAX;i++)
{
    if (avghue[i] > m_colorTh)
    {
        for(x=0;x<c_pro->width;x++)
            for(y=0;y<c_pro->height;y++)
            {
                chan1 = ((c_pro->width)*y+x)*c_pro-
>nChannels;
                p = c_pro->imageData[chan1];
                if (p == i)
                    c_pro->imageData[chan1] = 0;
            }
    }
}

for(x=0;x<c_pro->width;x++)
    for(y=0;y<c_pro->height;y++)
    {
        chan1 = ((c_pro->width)*y+x)*c_pro->nChannels;
        chan2 = ((c_out->width)*y+x)*c_out->nChannels;
        p = c_pro->imageData[chan1];
        q = c_out->imageData[chan2];

        if (p != 0)
        {
            c_pro->imageData[chan1] = 255;
            c_out->imageData[chan2] = 0;
        }
    }

for(x=0;x<c_pro->width;x++)
    for(y=0;y<c_pro->height;y++)
    {
        chan1 = ((c_pro->width)*y+x)*c_pro-
>nChannels;
        p = c_pro->imageData[chan1];

        if (p == 255)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            CvPoint seed = cvPoint(x,y);
            cvFloodFill(c_pro,seed,newval,lo_dif,up_dif,&comp,8);
            newval.val[0] += 1;
        }
    }

    int pix_x[MAX];
    int pix_y[MAX];

    for (i=0;i<MAX;i++)
    {
        count[i] = 0;
        pix_x[i] = 0;
        pix_y[i] = 0;
    }

    for(x=0;x<c_pro->width;x++)
        for(y=0;y<c_pro->height;y++)
        {
            chan1 = ((c_pro->width)*y+x)*c_pro-
                >nChannels;
            p = c_pro->imageData[chan1];
            for (int ncon = 0; ncon < MAX; ncon++)
            {
                if (p == ncon)
                {
                    pix_x[ncon] += x;
                    pix_y[ncon] += y;
                    count[ncon] ++;
                }
            }
        }

    //int centroid_x[MAX];
    //int centroid_y[MAX];

    for(i=0;i<MAX;i++)
    {
        if (count[i] != 0)
        {
            centroid_x[i] = (int *) (pix_x[i]/count[i]);
            centroid_y[i] = (int *) (pix_y[i]/count[i]);

            // Show centroid of color grading
            //////////////////////////////////////

            CString msg;
            msg.Format("C%d: %d, %d\n", i, centroid_x[i], centroid_y[i] );
            m_listCen.AddString(msg);
            UpdateData(FALSE);
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

// Show image after color grading
////////////////////
/* cvNamedWindow ("Colorc_out", 1);
   cvShowImage ("Colorc_out", c_out);
   cvNamedWindow ("Colorc_pro", 1);
   cvShowImage ("Colorc_pro", c_pro);*/

   cvCopy(c_out,p_out,NULL);

}

}

void CGBADlg::SizeGrading(IplImage *s_in, IplImage *s_out,int *centroid_x[], int *centroid_y[])
{
   UpdateData(TRUE);
   if (m_checkSize == FALSE)
   {
      s_out = s_in;
      //AfxMessageBox("No Size grading");
   }
   else
   {
      int width = 640;
      int height = 480;
      int x,y,chan1,chan2,chan3,i;
      unsigned char p, q, r;
      int count[MAX];

      CvConnectedComp comp;
      CvScalar newval;
      CvScalar lo_dif = cvScalar(0);
      CvScalar up_dif = cvScalar(0);
      newval.val[0] = 0;

      IplImage *s_pro = cvCreateImage(cvSize(width,height), 8, 1);
      IplImage *s_out = cvCreateImage(cvSize(width,height), 8, 1);

      cvCopy(s_in,s_pro,NULL);
      cvCopy(s_in,s_out,NULL);

      for(x=0;x<s_pro->width;x++)
         for(y=0;y<s_pro->height;y++)
         {
            chan1 = ((s_pro->width)*y+x)*s_pro->nChannels;
            p = s_pro->imageData[chan1];

            if (p == 255)
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CvPoint seed = cvPoint(x,y);

cvFloodFill(s_pro,seed,newval,lo_dif,up_dif,&comp,8);
    newval.val[0] += 1;
}

}

//*****//
    AfxMessageBox("Size grading is processing");
//*****//

int ncon;

for (i=0;i<MAX;i++)
    count[i] = 0;

for(x=0;x<s_pro->width;x++)
    for(y=0;y<s_pro->height;y++)
    {
        chan1 = ((s_pro->width)*y+x)*s_pro->nChannels;
        p = s_pro->imageData[chan1];
        for (ncon = 0; ncon < MAX; ncon++)
        {
            if (p == ncon)
                count[ncon] ++;
        }
    }

float size[MAX];

for (i=0;i<MAX;i++)
    size[i] = count[i]*PIX;

for (i=0;i<MAX;i++)
{
    UpdateData(TRUE);
    if (size[i] >= m_sizeTh)
    {
        for(x=0;x<s_pro->width;x++)
            for(y=0;y<s_pro->height;y++)
            {
                chan1 = ((s_pro->width)*y+x)*s_pro-
>nChannels;
                p = s_pro->imageData[chan1];

                if (p == i)
                    s_pro->imageData[chan1] = 0;
            }
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(x=0;x<s_pro->width;x++)
    for(y=0;y<s_pro->height;y++)
    {
        chan1 = ((s_pro->width)*y+x)*s_pro-
>nChannels;
        chan2 = ((s_out->width)*y+x)*s_out-
>nChannels;
        p = s_pro->imageData[chan1];
        q = s_out->imageData[chan2];
        if (p != 0)
            s_pro->imageData[chan1] = 255;
            s_out->imageData[chan2] = 0;
    }
for(x=0;x<s_pro->width;x++)
    for(y=0;y<s_pro->height;y++)
    {
        chan1 = ((s_pro->width)*y+x)*s_pro-
>nChannels;
        p = s_pro->imageData[chan1];
        if (p == 255)
        {
            CvPoint seed = cvPoint(x,y);
            cvFloodFill(s_pro,seed,newval,lo_dif,up_dif,&comp,8);
            newval.val[0] += 1;
        }
    }
int pix_x[MAX];
int pix_y[MAX];
for (i=0;i<MAX;i++)
{
    count[i] = 0;
    pix_x[i] = 0;
    pix_y[i] = 0;
}
for(x=0;x<s_pro->width;x++)
    for(y=0;y<s_pro->height;y++)
    {
        chan1 = ((s_pro->width)*y+x)*s_pro-
>nChannels;
        p = s_pro->imageData[chan1];
        for (ncon = 0; ncon < MAX; ncon++)
        {
            if (p == ncon)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        pix_x[ncon] += x;
        pix_y[ncon] += y;
        count[ncon] ++;
    }
}

//int centroid_x[MAX];
//int centroid_y[MAX];

for(i=0;i<MAX;i++)
{
    if (count[i] != 0)
    {
        centroid_x[i] = (int *) (pix_x[i]/count[i]);
        centroid_y[i] = (int *) (pix_y[i]/count[i]);

        /// Show centroid of sizing
        //////////////////////////////////////

        CString msg;
        msg.Format("S%d: %d, %d\n", i, centroid_x[i], centroid_y[i] );
        m_listCen.AddString(msg);
        UpdateData(FALSE);
    }
}

// Show image after size grading
////////////////////////////////////

/*
cvNamedWindow ("Sizes_out", 1);
cvShowImage ("Sizes_out", s_out);
cvNamedWindow ("Sizes_pro", 1);
cvShowImage ("Sizes_pro", s_pro); */
}
}

```

```

void CGBADlg::CalculateNozzle(int *xC[],int *yC[],int *xS[],int *yS[],int* nTimeC[],int*
nTimeS[],int *noz[],int *ncheck[])
{
    float dXC[MAX],dYC[MAX],dXS[MAX],dYS[MAX];
    int i;

    // Calibrate picture
    //////////////////////////////////////
    for (i=1;i<MAX;i++)
    {
        dXC[i] = 0;
        dYC[i] = 0;
        dXS[i] = 0;
        dYS[i] = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

for (i=1; i<MAX; i++)
{
    int a= (int) xC[i];
    dYC[i] = a*SC_N;

    a = (int) xS[i];
    dYS[i] = a*SC_N;

    a = (int) yC[i];
    dXC[i] = a*SC_T;

    a = (int) yS[i];
    dXS[i] = a*SC_T;

    if( (dXC[i] != 0) || (dXS[i] !=0) )
    {
        CString msg;
        msg.Format("C_5%d: %d,%d\n", i, dXC[i],dXS[i]);
        m_listCen.AddString(msg);
        UpdateData(FALSE);
    }
}

/// Find Nozzle
//////////

for (i=1; i<MAX; i++)
{
    noz[i] = 0;
    ncheck[i] = 0;
}

for (i=1; i<MAX; i++)
{
    if (dXC[i] <= 0.75)
    {
        noz[i] = (int *) 1;
        ncheck[i] = (int *) 1;
    }
    else
    {
        if(dXS[i] <= 0.75)
        {
            noz[i] = (int *) 1;
            ncheck[i] = (int *) 3;
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((dXC[i]>0.75) && (dXC[i]<=2.35))
{
    noz[i] = (int *) 2;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>0.75) && (dXS[i]<=2.35))
    {
        noz[i] = (int *) 2;
        ncheck[i] = (int *) 3;
    }
}

if((dXC[i]>2.35) && (dXC[i]<=4.05))
{
    noz[i] = (int *) 3;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>2.35) && (dXS[i]<=4.05))
    {
        noz[i] = (int *) 3;
        ncheck[i] = (int *) 3;
    }
}

if((dXC[i]>4.05) && (dXC[i]<=5.85))
{
    noz[i] = (int *) 4;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>4.05) && (dXS[i]<=5.85))
    {
        noz[i] = (int *) 4;
        ncheck[i] = (int *) 3;
    }
}

if((dXC[i]>5.85) && (dXC[i]<=7.65))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        noz[i] = (int *) 5;
        ncheck[i] = (int *) 1;
    }
    else
    {
        if((dXS[i]>5.85) && (dXS[i]<=7.65))
        {
            noz[i] = (int *) 5;
            ncheck[i] = (int *) 3;
        }
    }
}

if((dXC[i]>7.65) && (dXC[i]<=9.35))
{
    noz[i] = (int *) 6;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>7.65) && (dXS[i]<=9.35))
    {
        noz[i] = (int *) 6;
        ncheck[i] = (int *) 3;
    }
}

if((dXC[i]>9.35) && (dXC[i]<=11))
{
    noz[i] = (int *) 7;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>9.35) && (dXS[i]<=11))
    {
        noz[i] = (int *) 7;
        ncheck[i] = (int *) 3;
    }
}

}

if((dXC[i]>11) && (dXC[i]<=12.75))
{
    noz[i] = (int *) 8;
    ncheck[i] = (int *) 1;
}
else
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((dXS[i]>11) && (dXS[i]<=12.75))
{
    noz[i] = (int *) 8;
    ncheck[i] = (int *) 3;
}
}

if((dXC[i]>12.75) && (dXC[i]<=14.55))
{
    noz[i] = (int *) 9;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>12.75) && (dXS[i]<=14.55))
    {
        noz[i] = (int *) 9;
        ncheck[i] = (int *) 3;
    }
    if((dXC[i]>14.55) && (dXC[i]<=16.25))
    {
        noz[i] = (int *) 10;
        ncheck[i] = (int *) 1;
    }
    else
    {
        if((dXS[i]>14.55) && (dXS[i]<=16.25))
        {
            noz[i] = (int *) 10;
            ncheck[i] = (int *) 3;
        }
    }
}

if((dXC[i]>16.25) && (dXC[i]<=18))
{
    noz[i] = (int *) 11;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>16.25) && (dXS[i]<=18))
    {
        noz[i] = (int *) 11;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ncheck[j] = (int *) 3;
}
}

if((dXC[i]>18) && (dXC[i]<=19.8))
{
    noz[i] = (int *) 12;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>18) && (dXS[i]<=19.8))
    {
        noz[i] = (int *) 12;
        ncheck[i] = (int *) 3;
    }
}

if((dXC[i]>19.8) && (dXC[i]<=21.6))
{
    noz[i] = (int *) 13;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>19.8) && (dXS[i]<=21.6))
    {
        noz[i] = (int *) 13;
        ncheck[i] = (int *) 3;
    }
}

if((dXC[i]>21.6) && (dXC[i]<=23.35))
{
    noz[i] = (int *) 14;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>21.6) && (dXS[i]<=23.35))
    {
        noz[i] = (int *) 14;
        ncheck[i] = (int *) 3;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if((dXC[i]>23.35) && (dXC[i]<=25.1))
{
    noz[i] = (int *) 15;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>21.35) && (dXS[i]<=25.1))
    {
        noz[i] = (int *) 15;
        ncheck[i] = (int *) 3;
    }
}
}

if((dXC[i]>25.1) && (dXC[i]<=26.9))
{
    noz[i] = (int *) 16;
    ncheck[i] = (int *) 1;
}
else
{
    if((dXS[i]>25.1) && (dXS[i]<=26.9))
    {
        noz[i] = (int *) 16;
        ncheck[i] = (int *) 3;
    }
}

if(dXC[i]>26.9)
{
    noz[i] = (int *) 17;
    ncheck[i] = (int *) 1;
}
else
{
    if(dXS[i]>26.9)
    {
        noz[i] = (int *) 17;
        ncheck[i] = (int *) 3;
    }
}
}

if(noz[i] != (int *) 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CString msg;
        msg.Format("noz[%d]: %d \n ", i, noz[i]);
        m_listTime.AddString(msg);
        UpdateData(FALSE);
    }
}

/// Find Time to shoot beans
////////////////////////////////////
int sC[MAX],sS[MAX];

for (i=1;i<MAX;i++)
{
    sC[i] = 0;
    sS[i] = 0;
    nTimeC[i] = 0;
    nTimeS[i] = 0;
}
for (i=1;i<MAX;i++)
{
    if((dYC[i] !=0) || (dYS[i] != 0))
    {
        sC[i] = (37.6 - dYC[i]) + 118;
        sS[i] = (37.6 - dYS[i]) + 118;

        //nTimeC[i] = (float *) (sC[i]*V); // v = 63 cm/s
        //nTimeS[i] = (float *) (sS[i]*V); // v = 63 cm/s

        nTimeC[i] = (int *) (sC[i]/V); // v = 63 cm/s
        nTimeS[i] = (int *) (sS[i]/V); // v = 63 cm/s

        int a=(int) nTimeC[i];

        CString msg;
        msg.Format("TimeC[%d]: %d \n TimeS[%d]: %d \n", i,
nTimeC[i], i , nTimeS[i]);

        m_listTime.AddString(msg);
        UpdateData(FALSE);
    }
}

}

HANDLE hComm;
DCB dcb;
char *chCommPort ="COM3";
int valReturn;
byte chBuffer;
unsigned long nReadPort;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned long nWritePort;
```

```
void CGBADlg::SendSignaltoNozzle(int *tC[],int *tS[],int *noz[], double dBegin, double dTick,int *ncheck[])
```

```
{
    // TODO: Add your message handler code here and/or call default
    hComm =
    CreateFile(chCommPort,GENERIC_READ|GENERIC_WRITE,0,NULL,OPEN_EXISTING,0,NULL);
    if(hComm == INVALID_HANDLE_VALUE)
    {
        // m_listTime = "Can't CreateFile";
        // UpdateData(FALSE);
        //exit(1);
    }

    valReturn = GetCommState(hComm,&dcb);

    if(!valReturn)
    {
        // m_listTime = "Check State Com Port Error";
        //UpdateData(FALSE);
        //exit(0);
    }

    dcb.BaudRate = CBR_9600;
    dcb.ByteSize = 8;
    dcb.Parity = NOPARITY;
    dcb.StopBits = ONESTOPBIT;

    valReturn = SetCommState(hComm,&dcb);

    if(!valReturn)
    {
        // m_listTime = "Initial Com Port Error";
        // UpdateData(FALSE);
        //exit(3);
    }

    byte AsciiSet[] = {0x41,0x42,0x43,0x44,0x45,0x46,0x47,0x48,0x49,0x4A,
                       0x4B,0x4C,0x4D,0x4E,0x4F,0x50,0x51,0x52,0x53,0x54};
```

```
UpdateData(TRUE);
```

```
int i;
double dEnd[MAX],dTime[MAX];
int dDelay[MAX];
```

```
for(i=0;i<MAX;i++)
{
    dEnd[i] = 0;
    dTime[i] = 0;
    dDelay[i] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i=1;i<MAX;i++)
{
    int test = (int ) (noz[i]);

    if(test != 1)
    {

        int b = (int ) (ts[i]);
        int a = (int ) (tc[i]);
        int c = (int ) (ncheck[i]);

        dEnd[i] = cvGetTickCount();
        dTime[i] = ((dEnd[i] - dBegin)/dTICK);

        CString msg;
        msg.Format("TimeC[%d]: %f \n", i, dTime[i]);
        m_listTime.AddString(msg);
        UpdateData(FALSE);

        if (c = 1)
            dDelay[i] = 10000; //(int ) ((a*1000) - dTime[i]);

        if(c = 3)
            dDelay[i] = 60000; //(int ) ((b*1000) - dTime[i]);

        // SetTimer(test,dDelay[i],NULL);

        for(int j=0;j<700000000;j++)
            for(int h=0;h<700000000;h++)
                for(int r=0;r<700000000;r++)
                    for(int e=0;e<700000000;e++)
                    {
                        int c = j++;
                        int v = h++;
                        // int w = r++;
                        // int n = e++;
                    }

        switch(test)
        {
            case 1 : WriteByte(AsciiSet[0]);
                    break;
            case 2 : WriteByte(AsciiSet[1]);
                    break;
            case 3 : WriteByte(AsciiSet[2]);
                    break;
            case 4 : WriteByte(AsciiSet[3]);
                    break;
            case 5 : WriteByte(AsciiSet[4]);
                    break;
            case 6 : WriteByte(AsciiSet[5]);
                    break;
        }
    }
}

```

```

        break;
    case 7 : WriteByte(AsciiSet[6]);
        break;
    case 8 : WriteByte(AsciiSet[7]);
        break;
    case 9 : WriteByte(AsciiSet[8]);
        break;
    case 10 : WriteByte(AsciiSet[9]);
        break;
    case 11 : WriteByte(AsciiSet[10]);
        break;
    case 12 : WriteByte(AsciiSet[11]);
        break;
    case 13 : WriteByte(AsciiSet[12]);
        break;
    case 14 : WriteByte(AsciiSet[13]);
        break;
    case 15 : WriteByte(AsciiSet[14]);
        break;
    case 16 : WriteByte(AsciiSet[15]);
        break;
    case 17 : WriteByte(AsciiSet[16]);
        break;
    case 18 : WriteByte(AsciiSet[17]);
        break;
    case 19 : WriteByte(AsciiSet[18]);
        break;
    case 20 : WriteByte(AsciiSet[19]);
        break;
    }
}

CloseHandle(hComm);
}

BOOL CGBADlg::WriteByte(byte byteb)
{
    //iBytesWritten=0;
    if(WriteFile(hComm,&byteb,1,&nWritePort,NULL)==0)
        return FALSE;
    else
        return TRUE;
}

void CGBADlg::OnTimer(UINT nIDEvent)
{
    // TODO: Add your message handler code here and/or call default

    CDialog::OnTimer(nIDEvent);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ส่วนของโปรแกรม camara

```
// This program is developed by Phatcharee KHUMHOM (AJ)

// CameraDlg.cpp : implementation file

#include "stdafx.h"
#include "Camera.h"
#include "CameraDlg.h"
#include "1394Camera.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

#include "cv.h"
#include "highgui.h"
#include "cxcore.h"
#include "cvcam.h"

#pragma comment (lib,"cv.lib")
#pragma comment (lib,"highgui.lib")
#pragma comment (lib,"cxcore.lib")
#pragma comment (lib,"cvcam.lib")
#pragma comment (lib,"1394Camera.lib")

// adjust parameters to your needs
#define CAM_RES_WIDTH 640
#define CAM_RES_HEIGHT 480
#define CAM_RES_CHANNELS 3

//buffer for temorary storage of frame data
unsigned char buf[CAM_RES_WIDTH*CAM_RES_HEIGHT*CAM_RES_CHANNELS];

/////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}

//camera object of the 1394CMU driver
C1394Camera Camera;

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CCameraDlg dialog

CCameraDlg::CCameraDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CCameraDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CCameraDlg)
    // NOTE: the ClassWizard will add member initialization here
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CCameraDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//{{AFX_DATA_MAP(CCameraDlg)
    // NOTE: the ClassWizard will add DDX and DDV calls here
//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CCameraDlg, CDialog)
//{{AFX_MSG_MAP(CCameraDlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDOK, OnCamera)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CCameraDlg message handlers

BOOL CCameraDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);           // Set big icon
    SetIcon(m_hIcon, FALSE);        // Set small icon

    // TODO: Add extra initialization here

    return TRUE; // return TRUE unless you set the focus to a control
}

void CCameraDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

```

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

```

void CCameraDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

```

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.

```

HCURSOR CCameraDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

```

```

void CCameraDlg::OnCamera()
{
    if(Camera.CheckLink() != CAM_SUCCESS)
    {
        AfxMessageBox("\nNo Link.");
    }

    if(Camera.InitCamera(TRUE) != CAM_SUCCESS)
    {
        AfxMessageBox("\nInitialization failed.");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if(Camera.SetVideoFormat(0) != CAM_SUCCESS)
{
    AfxMessageBox("\nCould not set video format.");
}

if(Camera.SetVideoMode(4) != CAM_SUCCESS)
{
    AfxMessageBox("\nCould not set video mode.");
}

if(Camera.SetVideoFrameRate(3) != CAM_SUCCESS)
{
    AfxMessageBox("\nCould not set frame rate.");
}

if(Camera.StartImageAcquisition() != CAM_SUCCESS)
{
    AfxMessageBox("\nCould not start image acquisition.");
}

if(Camera.AcquireImage() != CAM_SUCCESS)
{
    AfxMessageBox("\nCould not acquisition image.");
}

cvNamedWindow( "Source",1);

IplImage *Image = cvCreateImage( cvSize(CAM_RES_WIDTH, CAM_RES_HEIGHT), 8,
CAM_RES_CHANNELS);

int Img_nL = Image->height;
int Img_nC = Image->width*Image->nChannels;

Camera.getRGB( (unsigned char*)Image->imageData,Img_nL*Img_nC);

cvShowImage("Source", Image);
cvWaitKey(0);

Camera.StopImageCapture();
cvReleaseImage(&Image);
cvDestroyWindow("Source");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ส่วนของโปรแกรม Finding Color Bean

```
// This program is developed by Phatcharee KHUMHOM (AJ)

% Find Mean Hue
I = imread ('D:\AJ\C1.jpg');           // อ่านภาพ Input

imshow (I);                           // แสดงรูปภาพที่อ่าน

out = rgb2hsv(I);
H=out(:,:,1);
meanH = mean(H);
meanH = mean (meanH);
meanH                                     // คำนวณค่าเฉลี่ยสี Hue ของถั่ว

% Find Histogramme
I = imread ('D:\AJ\A1.jpg');
out = rgb2hsv(I);
H=out(:,:,1);
imhist(H)                                // หาฮิสโตแกรม

% Find std
I = imread ('D:\AJ\C1.jpg');
out = rgb2hsv(I);
H=out(:,:,1);
st = std (H);
mean (st)                                // หาค่าเบี่ยงเบนมาตรฐานของสีถั่ว
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

ส่วนของโปรแกรมควบคุม Microcontroller

// This program is developed by Penthida KOOHASANEH (Pui)

```
#include<at89x52.h>
```

```
sbit Output1 = P0^0;
sbit Output2 = P0^1;
sbit Output3 = P0^2;
sbit Output4 = P0^3;
sbit Output5 = P0^4;
sbit Output6 = P0^5;
sbit Output7 = P0^6;
sbit Output8 = P0^7;
sbit Output9 = P1^0;
sbit Output10 = P1^1;
sbit Output11 = P1^2;
sbit Output12 = P1^3;
sbit Output13 = P1^4;
sbit Output14 = P1^5;
sbit Output15 = P1^6;
sbit Output16 = P1^7;
sbit Output17 = P2^0;
sbit Output18 = P2^1;
sbit Output19 = P2^2;
sbit Output20 = P2^3;
```

```
sbit Chk1S = P3^2;
unsigned int DelLop;
unsigned int delay = 3000;
unsigned char dat;
```

```
//*****
```

```
void serial(void) interrupt 4
```

```
{
    if(RI)
    {
        RI = 0;
        dat = SBUF;
    }

```

```
    if(TI)
    {
        TI = 0;
    }
}
```

```
//*****
```

```
void Delay_200uS(unsigned int DelLop)
```

```
{
    TR0 = 1;
    while(DelLop--)
    {
        while(!TF0);
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    TFO = 0;
}

TRO = 0;
dat = 0;

}

//*****

void main(void)
{
    P0 = 0x00;
    P1 = 0x00;
    P2 = 0x00;

    SCON = 0x50;
    TMOD = 0x22;
    TH1 = 0xFD;
    TL1 = 0xFD;
    TI = 0;
    RI = 0;
    EA = 1;
    ES = 1;
    TR1 = 1;

    TH0 = 0x48;
    TL0 = 0x48;

    while(1)
    {
        // Delay_200uS(5000);
        // Chk1S = ~Chk1S;

        switch(dat)
        {
            case 'A':{ Output1 = 1; Delay_200uS(delay);Output1 = 0; break;}
            case 'B':{ Output2 = 1; Delay_200uS(delay);Output2 = 0; break;}
            case 'C':{ Output3 = 1; Delay_200uS(delay);Output3 = 0; break;}
            case 'D':{ Output4 = 1; Delay_200uS(delay);Output4 = 0; break;}
            case 'E':{ Output5 = 1; Delay_200uS(delay);Output5 = 0; break;}
            case 'F':{ Output6 = 1; Delay_200uS(delay);Output6 = 0; break;}
            case 'G':{ Output7 = 1; Delay_200uS(delay);Output7 = 0; break;}
            case 'H':{ Output8 = 1; Delay_200uS(delay);Output8 = 0; break;}
            case 'I':{ Output9 = 1; Delay_200uS(delay);Output9 = 0; break;}
            case 'J':{ Output10 = 1; Delay_200uS(delay);Output10 = 0; break;}
            case 'K':{ Output11 = 1; Delay_200uS(delay);Output11 = 0; break;}
            case 'L':{ Output12 = 1; Delay_200uS(delay);Output12 = 0; break;}
            case 'M':{ Output13 = 1; Delay_200uS(delay);Output13 = 0; break;}
            case 'N':{ Output14 = 1; Delay_200uS(delay);Output14 = 0; break;}
            case 'O':{ Output15 = 1; Delay_200uS(delay);Output15 = 0; break;}
            case 'P':{ Output16 = 1; Delay_200uS(delay);Output16 = 0; break;}
            case 'Q':{ Output17 = 1; Delay_200uS(delay);Output17 = 0; break;}
            case 'R':{ Output18 = 1; Delay_200uS(delay);Output18 = 0; break;}
            case 'S':{ Output19 = 1; Delay_200uS(delay);Output19 = 0; break;}
            case 'T':{ Output20 = 1; Delay_200uS(delay);Output20 = 0; break;}

            default : break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ตารางแสดงผลการทำงานของโปรแกรม GBA

ตารางที่ ข.1 ผลการคัดเลือของตัวระบุที่ขึ้นจากโปรแกรม GBA

ตัวระบุที่ขึ้น รูปที่	จำนวนสักถั่ว	คัดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของ ไมโครคอนโทรลเลอร์	การทำงาน ของหัวพ่นลม
		จำนวนสักถั่วเสีย	ตำแหน่งหัวพ่นลม	จำนวนสักถั่วเสีย	ตำแหน่งหัวพ่นลม		
1	10	1	9	1	9	1	9
2	9	1	13	1	13	1	13
3	8	1	10	1	10	1	10
4	10	1	9	1	9	1	9
5	9	1	3	1	3	1	3
6	10	1	3	1	3	1	3
7	7	1	6	1	6	1	6
8	11	1	8	1	8	1	8
9	10	2	4	2	4	2	4
			4		4		4
10	11	2	14	2	14	2	14
			9		9		9

ตารางที่ จ.1 (ต่อ)

ตัวระบุผู้ป็น รูปที่	จำนวนสัปดาห์	คัดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของ ไมโครคอนโทรลเลอร์	การทำงานของ ของหัวพิน
		จำนวนสัปดาห์	ตำแหน่งหัวพิน	จำนวนสัปดาห์	ตำแหน่งหัวพิน		
11	8	3	5	3	5	3	5
			8		8		8
			9		9		9
12	10	3	3	3	3	3	3
			9		9		9
			7		7		7
13	10	2	10	2	10	2	10
			12		12		12
14	10	2	15	2	15	2	15
			3		3		3
15	10	1	9	1	9	1	9
16	10	2	16	2	16	2	16
			13		13		13
17	10	2	8	2	8	2	8
			5		5		5

ตารางที่ จ. 1 (ต่อ)

ตัวระบุผู้ป้อน วัสดุ	จำนวนสัปดาห์	คัดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของ ไมโครคอนโทรลเลอร์	การทำงาน ของหัวพ่นลม
		จำนวนสัปดาห์	ตำแหน่งหัวพ่นลม	จำนวนสัปดาห์	ตำแหน่งหัวพ่นลม		
18	10	2	9	2	9	2	9
			3		3		3
19	10	2	10	2	10	2	10
			3		3		3
20	10	2	16	2	16	2	16
			6		6		6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ. 2 ผลการศึกษาค้นคว้าวิจัยจากโปรแกรม GBA

ตัวระบุชิ้น รูปที่	จำนวนผู้วิจัย	คัดแยกโดยสาขา		ประเมินผลจากโปรแกรม GBA		การทำงานของ ไมโครคอนโทรลเลอร์	การทำงาน ของหัวพ่นลม
		จำนวนผู้วิจัย	ตำแหน่งหัวพ่นลม	จำนวนผู้วิจัย	ตำแหน่งหัวพ่นลม		
1	10	1	4	1	4	4	4
2	9	1	16	1	16	16	16
3	8	1	13	1	13	13	13
4	10	1	15	1	15	15	15
5	9	2	15	2	15	15	15
			7		7	7	7
6	10	2	3	2	3	3	3
			15		15	15	15
7	7	2	6	2	6	6	6
			13		13	13	13
8	11	2	16	2	16	16	16
			4		4	4	4
9	10	1	12		12	12	12
10	11	3	17	3	17	17	17
			4		4	4	4

ตารางที่ จ. 2 (ต่อ)

ตัวระบุปีงบประมาณ ปีที่	จำนวนสังกัด	คัดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของไมโครคอมพิวเตอร์	การทำงาน
		จำนวนสังกัดด้วย	ตำแหน่งหัวหน้า	จำนวนสังกัดด้วย	ตำแหน่งหัวหน้า		
			6		6	6	6
11	8	1	6	1	6	6	6
12	10	1	6	1	6	6	6
13	10	2	10	2	10	10	10
			13		13	13	13
14	10	1	9	1	9	9	9
15	10	2	12	2	12	12	12
			9		9	9	9
16	10	2	6	2	6	6	6
			5		3	3	3
17	10	2	15	2	15	15	15
			13		13	16	13
18	8	2	16		16	16	16
			6		6	6	6

วารสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ. 2 (ต่อ)

ตัวระบุรูปแบบ	จำนวนสีแก้ว	กัณฑ์โดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของไมโครคอนโทรลเลอร์	การทำงานของหัวพินลม
		จำนวนสีแก้ว	ตำแหน่งหัวพินลม	จำนวนสีแก้ว	ตำแหน่งหัวพินลม		
19	10	2	16	2	16	16	16
			6		6	6	6
20	10	2	10	2	10	10	10
			3		3	3	3

กรณีเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3. ผลการกีดสีและกีดขนาดของถ้วยระดูปีนจากโปรแกรม GBA

ถ้วยระดูปีน รูปที่	จำนวนสักถ้วย	กีดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของ ไมโครคอมพิวเตอร์	การทำงาน ของหัวพนม
		จำนวนสักถ้วย	ตำแหน่งหัวพนม	จำนวนสักถ้วย	ตำแหน่งหัวพนม		
1	10	2	9	2	9	9	9
			4		4	4	4
2	9	2	13	2	13	13	13
			16		16	16	16
3	8	2	10	2	10	10	10
			13		13	13	13
4	10	2	9	2	9	9	9
			15		15	15	15
5	9	3	3	3	3	3	3
			15		15	15	15
			7		7	7	7
6	10	3	3	3	3	3	3
			3		3	3	3
			15		15	15	15

ถ้ามีเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ. 3 (ต่อ)

ตัวระบุผู้พิมพ์ รูปที่	จำนวนผู้ถือ	คัดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของไมโครคอมพิวเตอร์	การทำงาน ของหัวพิมพ์
		จำนวนผู้ถือ	ตำแหน่งหัวพิมพ์	จำนวนผู้ถือ	ตำแหน่งหัวพิมพ์		
7	7	3	6	3	6	6	6
			6		6		6
			13		13		13
8	11	3	8	3	8	8	8
			16		16		16
			4		4		4
9	10	3	4	3	4	4	4
			4		4		4
			12		12		12
10	11	5	14	5	14	14	14
			9		9		9
			17		17		17
			4		4		4
			6		6		6

มหาวิทยาลัยสงขลานครินทร์ ขอสงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ. 3 (ต่อ)

ตัวระบุชิ้น รูปที่	จำนวนสักตัว	คัดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของ ไมโครคอนโทรลเลอร์	การทำงาน ของหัวเข็ม
		จำนวนสักตัวเสีย	ตำแหน่งหัวพินลม	จำนวนสักตัวเสีย	ตำแหน่งหัวพินลม		
11	8	4	5	4	5	5	5
			8		8	8	8
			6		6	6	6
			9		9	9	9
12	10	4	3	4	3	3	3
			9		9	9	9
			7		7	7	7
			6		6	6	6
13	10	4	10	4	10	10	10
			13		13	13	13
			13		13	13	13
			10		10	10	10
14	10	3	9		9	9	9
			15		15	15	15

นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ. 3 (ต่อ)

ตัวระบุรูปที่	จำนวนปีกตัว	คัดแยกโดยสายตา		ประมาณผลจากโปรแกรม GBA		การทำงานของไมโครคอนโทรลเลอร์	การทำงานของหัวพิน
		จำนวนปีกตัวเสีย	ตำแหน่งหัวพิน	จำนวนปีกตัวเสีย	ตำแหน่งหัวพิน		
			3		3	3	3
15	10	3	12	3	12	12	12
			9		9	9	9
			9		9	9	9
16	10	4	5	4	5	5	5
			6		6	6	6
			16		16	16	16
			13		13	13	13
17	10	4	8	4	8	8	8
			5		5	5	5
			15		15	15	15
			13		13	13	13
18	10	4	9	4	9	9	9
			3		3	3	3

ตารางที่ จ. 3 (ต่อ)

ตัวระบุ รูปที่	จำนวนสักตัว	คัดแยกโดยสายตา		ประมวลผลจากโปรแกรม GBA		การทำงานของ ไมโครคอนโทรลเลอร์	การทำงาน ของหัวพิมพ์
		จำนวนสักตัวเสีย	ตำแหน่งหัวพิมพ์	จำนวนสักตัวเสีย	ตำแหน่งหัวพิมพ์		
			16		16	16	16
			6		6	6	6
19	10	4	10	4	10	10	10
			3		3	3	3
			16		16	16	16
			6		6	6	6
20	10	4	10	4	10	10	10
			3		3	3	3
			16		16	16	16
			6		6	6	6

นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

ตารางพื้นที่ภาพถ่ายของถั่วแต่ละขนาด

ตารางที่ ฉ. 1 พื้นที่ภาพถ่ายของถั่วแต่ละขนาด

ถั่วฝักที่	พื้นที่ภาพถ่าย (ตร.ซม.)		
	ขนาดเล็ก	ขนาดกลาง	ขนาดใหญ่
1	5.548	6.08	5.8
2	5.79	5.25	6.98
3	4.92	5.17	4.87
4	5.64	6.07	6.42
5	5.19	5.5	6.71
6	4.17	5.27	7.33
7	4.03	5.11	6.19
8	4.94	5.52	7.31
9	4.01	4.9	6.64
10	5.04	6.38	9.09
11	4.35	5.91	5.88
12	4.38	5.05	7.54
13	3.79	5.15	7.09
14	5.08	5.81	7.11
15	4.86	6.16	6.34
16	4.35	5.97	7.19
17	4.19	4.96	6.07
18	4.24	6.11	6.28
19	4.1	4.82	6.25
20	4.88	5.08	6.47
21	4.36	5.37	
22	4.95	5.16	
23	4.72	5.97	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ฉ. 1 (ต่อ)

ตัวฝึกที่	พื้นที่ภาพฉาย (ตร.ซม.)		
	ขนาดเล็ก	ขนาดกลาง	ขนาดใหญ่
24	4.58	5.4	
25	4.41		
26	4.01		
27	4.38		
28	4.48		
29	5.01		
30	4.06		
เฉลี่ย	4.42	5.40	6.68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้