ระบบควบคุมหุ่นยนต์แขนกลอัตโนมัติโดยการประมวลผลภาพ

AUTOMATIC ARM ROBOT CONTROL SYSTEM VIA COMPUTER VISION

นายภัทรพล ศิวปรียานุพงษ์ รหัส 48370969

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ - 2 ก.ค. 2556
เลขทะเบียน 1628080
เลขเรียกหนังสือ ส
มหาวิทยาลัยนเรศวร ส 368 ร 2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	**ระบบควบคุมหุ่นยนต์แขนกลอัตโนมัติโดยการประมวลผลภาพ**
ผู้ดำเนินโครงงาน	**นายภัทรพล ศิวปรียานุพงษ์ รหัส 48370969**
ที่ปรึกษาโครงงาน	อาจารย์เศรษฐา ตั้งค้าวานิช
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงงาน
(อาจารย์เศรษฐา ตั้งค้าวานิช)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ริยะมงคล)

.....กรรมการ
(อาจารย์สิรภพ คชรัตน์)

ชื่อหัวข้อโครงงาน	ระบบควบคุมหุ่นยนต์แขนกลอัตโนมัติโดยการประมวลผลภาพ
ผู้ดำเนินโครงงาน	นายภัทรพล ศิวปรียานุพงษ์ รหัส 48370969
ที่ปรึกษาโครงงาน	อ. เศรษฐา ตั้งค้าวานิช
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

.....

บทคัดย่อ

โครงงานนี้นำเสนอการศึกษาและพัฒนาระบบควบคุมหุ่นยนต์แขนกลอัตโนมัติโดยการประมวลผลภาพเพื่อเพิ่มขีดความสามารถในการสั่งการควบคุมแขนกลให้สามารถจับวัตถุเองไม่ต้องทำการควบคุมการจับโดยที่มีกล้องเว็บแคม 2 ตัวคอยจับภาพวัตถุเพื่อส่งภาพที่จับได้ไปประมวลผลที่เครื่องคอมพิวเตอร์เพื่อคำนวณหาตำแหน่งและระยะห่างจากจุดตรงกลางของวัตถุมาเทียบกับฐานของแขนกล โดยเฉพาะการประมวลผลภาพนั้นประกอบด้วยการนำภาพที่ได้จากกล้องเว็บแคมมากรองค่าสีเพื่อจะได้วัตถุภาพสีที่ต้องการกับพื้นหลังจากนั้นแปลงจากภาพสีที่ได้เป็นภาพระดับเทา และทำให้เป็นภาพขาวดำในขั้นตอนต่อไป เพื่อแยกวัตถุกับส่วนที่เป็นพื้นหลังออกจากกัน ในส่วนของการเข้าไปจับวัตถุจะมีการกำหนดเส้นทางล่วงหน้าให้จดจำและแขนกลจะเดินตามเป็นเส้นตรงไปจับวัตถุ จากการทดลองที่ไม่มีการควบคุมแสงสว่างให้ดีจะทำให้กล้องจับภาพวัตถุไม่ได้หรือหาจุดตรงกลางวัตถุไม่ตรงและแขนกลเข้ามาบังกล้องตัวบนจะส่งผลให้แขนกลไม่สามารถจับวัตถุได้ ซึ่งระบบที่ออกแบบขึ้นนี้ถูกพัฒนาขึ้นบนภาษาวิชวลสตูดิโอ c# 2008 โดยใช้การประมวลผลภาพ เข้ามาช่วยในการประมวลผล

แขนกลอัตโนมัติจะถูกควบคุมโดยไมโครคอนโทรเลอร์เพียงแค่หนึ่งตัว ซึ่งใช้ไมโครคอนโทรเลอร์ในการประมวลผลคำสั่งที่รับมาจากเครื่องคอมพิวเตอร์ซึ่งจะมีเมนูสำหรับให้ผู้ใช้คอยควบคุมแขนกลที่เครื่องคอมพิวเตอร์โดยค่าที่กำหนดให้จะนำไปใช้ควบคุมตำแหน่งของเซอร์โวมอเตอร์ในข้อต่อต่างๆของแขนกลซึ่งสามารถนำไปประยุกต์ใช้ในงานที่อันตรายต่อมนุษย์ที่ไม่สามารถสัมผัสวัตถุเองได้ เช่นในโรงงานผสมสารเคมี หรือโรงงานบรรจุสารกัมมันตภาพรังสีได้

Project title	Automatic Arm Robot Control System Via Computer Vision
Name	Mr. Patarapol Siwaprecyanupong ID. 48370969
Project advisor	Mr. Settha Thangkawanit
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2010

.....

Abstract

This project provides the studying and development of an automatic arm robot control system via computer vision in order to increase capacity of the robot arm control. The objective itself cannot make control the catch two cameras by webcam at the object to capture images, and sent the image captured to a computer processor. To calculate the position and distance from the center of the object relative to the base of the robot arm. Image processing consists of webcam, which color filtering to the object and background of them, then conversion color image to gray scale image and black to white image in the next step. The processing to separate the object that the background apart in access will have the path planning and remember that Robot's arms will follow a straight line to the object. The experiments with on control the lighting, the camera will not capture the object or it does not match in the middle. And robot ram to shield on the camera, the robot ram camera cannot touch the object. The system therefore was designed to develop the Visual studio c# 2008 by using the image processing to assist in the process.

The automatic robot arm were controlled by single processor, which used microcontroller processing in command received from the computer, which users control menu robot arms from computer. The value assigned will be used to control the position of serve motor in various joints of the robot arm. It can be applied in danger work. People cannot touch objects, such as in mixed chemicals factory, or plant containing radioactive substances.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากการได้รับความช่วยเหลืออนุเคราะห์ของหลายๆท่านด้วยกัน โดยเฉพาะอย่างยิ่ง อ. เศรษฐา ตั้งค้าวานิช ที่ได้สละเวลาให้คำปรึกษาและเสนอแนะแนวทางสำหรับการแก้ไขปัญหาที่เกิดขึ้นขณะการจัดทำโครงงาน พร้อมด้วยคณะกรรมการที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ริยะมงคล และ อ. สิรภพ คชรัตน์ ที่ให้คำแนะนำในการเขียนรูปเล่มปริญญานิพนธ์ และขอบคุณศูนย์วิจัยร่วมเฉพาะทางด้านการผลิตชั้นสูงในอุตสาหกรรมฮาร์ดดิสก์ไดร์ฟ (I/U CRC in HDD Advanced Manufacturing) สถาบันวิทยาการหุ่นยนต์ภาคสนาม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีกับบริษัทอินโนเวกซ์ (ประเทศไทย) จำกัด

ผู้จัดทำโครงงานจะขอขอบพระคุณทุกๆท่านที่มีส่วนร่วมในการทำโครงงานนี้ตลอดจนผู้เขียน ผู้คิดค้นทฤษฎีต่างๆและขอกราบขอบคุณบิดา มารดา ที่คอยสนับสนุนในด้านการเงินจนโครงงานบรรลุเป้าหมายในที่สุด

คณะผู้จัดทำโครงงานวิศวกรรม
นายภัทรพล ศิวปรียานุพงษ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตการทำโครงงาน 1

1.4 ขั้นตอนการดำเนินงานของโครงงาน 2

1.5 แผนการดำเนินงาน 2

1.6 ผลที่คาดว่าจะได้รับ 3

1.7 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎี 4

2.1 หุ่นยนต์ (Robot) 4

2.2 แขนกล (Arm Robot) 8

2.3 ไมโครคอนโทรลเลอร์ตระกูล AVR 10

2.4 มาตรฐาน RS-232 18

2.5 เซอร์โวมอเตอร์ (Servo Motor) 20

2.6 การประมวลผลภาพดิจิตอล (Digital Image Processing) 20

2.7 การแยกบริเวณ (Segmentation) 28

2.8 การวางแผนเส้นทาง (Path Planning) 29

2.9 ทฤษฎีการประสมแสง (The) 30

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินโครงงาน 32

3.1 ภาพรวมระบบของแขนกล (Robot Arm System Overview) 32

3.2 การพัฒนาโครงสร้างของแขนกลอัตโนมัติ 33

3.3 กล้องเว็บแคม (Web Camera) 40

บทที่ 4 ผลการทดลอง 42

4.1 จุดประสงค์ของการทดลอง 42

4.2 ขั้นตอนการทดลอง 42

4.3 การทดสอบโปรแกรมที่พัฒนาขึ้นมา 43

4.4 การทดสอบการเคลื่อนที่ของแขนกลที่พัฒนาขึ้น 44

บทที่ 5 บทสรุปและวิเคราะห์โครงงาน 61

5.1 สรุปผลการทดลอง 61

5.2 ปัญหาและแนวทางในการแก้ไข 61

5.2 แนวทางในการพัฒนาต่อ 62

เอกสารอ้างอิง 63

ภาคผนวก ก 65

ภาคผนวก ข 70

ภาคผนวก ค 75

ประวัติผู้เขียนโครงงาน 78

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงาน 2

2.2 แสดงเวลา Bit time ใน Baud rate 20

4.1 ผลการทดสอบการหยิบจับวัตถุสีแดงตรงกลาง 47

4.2 ผลการทดสอบการหยิบจับวัตถุสีเหลืองซ้ายสุด 50

4.3 ผลการทดสอบการหยิบจับวัตถุสีน้ำเงินขวาสุด 53

4.4 ผลการทดสอบการหยิบจับวัตถุสีแดงซ้ายสุด 54

4.5 ผลการทดสอบการหยิบจับวัตถุสีน้ำเงินตรงกลางสองชั้น 56

4.6 ผลการทดสอบการหยิบจับวัตถุสีแดงขวาสุดสามชั้น 57

4.7 ผลการทดสอบการหยิบจับวัตถุสีเหลืองตรงกลางสามชั้น 68

สารบัญรูป

รูปที่ หน้า

2.1 ไดอะแกรมการทำงานของ Robot 4

2.2 หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (Fixed Robot) 5

2.3 หุ่นยนต์คาร์ทีเซียน (Cartesian Coordinated Robot) 5

2.4 หุ่นยนต์ทรงกระบอก (Cylindrical Coordinated Robot) 6

2.5 หุ่นยนต์ทรงกลม (Spherical Coordinated Robot) 6

2.6 หุ่นยนต์ข้อต่อ (Join-Arm Coordinated Robot) 7

2.7 หุ่นยนต์สการ่า (Scara Robot) 7

2.8 หุ่นยนต์ชนิดที่เคลื่อนที่ได้ (Mobile Robot) 8

2.9 แสดงข้อต่อแบบต่างๆ ที่นิยมใช้ 9

2.10 ส่วนประกอบต่างๆของแขนกลเมื่อเปรียบเทียบกับแขนมนุษย์ 10

2.11 โครงสร้างภายในของไมโครคอนโทรลเลอร์ 11

2.12 โครงสร้างอธิบายส่วนประกอบของชิพ Atmega 1280 12

2.13 ขาต่างๆของชิพไมโครคอนโทรลเลอร์ Atmega 1280 16

2.14 โครงสร้างของ AVR 17

2.15 แสดงสถานะของการทำงานด้วยกัน 8 บิต 17

2.16 การใช้มาตรฐาน RS-232 เชื่อมต่ออุปกรณ์ 18

2.17 รูปแบบการสื่อสารอนุกรม 1 เฟรม 20

2.18 ส่วนประกอบของเซอร์โวมอเตอร์ 20

2.19 สัญญาณที่สามารถป้อนให้กับเซอร์โวมอเตอร์ 21

2.20 ตำแหน่งของเซอร์โวมอเตอร์เมื่อป้อนความกว้าง 21

2.21 กระบวนการทำงานแยกวัตถุออกจากพื้นหลัง 23

2.22 ภาพขยายของรูปภาพ 24

2.23 โคออร์ดิเนตของภาพ 2 มิติ 24

2.24 แสดงภาพค่า RGB ในแต่ละพิกเซล 25

2.25 ความสัมพันธ์ระหว่าง picture elements ของภาพกับพิกเซล 25

2.26 ภาพเชิงดิจิตอลประเภท Intensity Image 26

2.27 ตัวอย่างภาพสเกลสีเทา 27

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.28 ภาพเชิงดิจิตอลประเภท Binary Image 27
2.29 ตัวอย่าง Binary Image 28
2.30 ตัวอย่างค่าของภาพที่ใช้ในการคำนวณ 28
2.31 แสดงการเคลื่อนที่ของแขนกลมองจากกล้องด้านข้าง 28
2.32 แสดงการเคลื่อนที่ของแขนกลมองจากกล้องด้านบนลงมา 29
2.33 แสดงการผสมสี 30
2.34 แบบจำลองระบบสี RGB 31
3.1 ไดอะแกรมแสดงการทำงานของระบบควบคุมแขนกล 32
3.2 รูปร่างและขนาดของเซอร์โวมอเตอร์ Futaba S3003 33
3.3 รูปร่างและขนาดของเซอร์โวมอเตอร์ Tower Pro 9805MG 33
3.4 รูปร่างและขนาดของเซอร์โวมอเตอร์ Tower Pro MG996R 34
3.5 แสดงตำแหน่งการวางอุปกรณ์มอเตอร์ขับเคลื่อนของแขนกล 34
3.6 ส่วนลำตัวของแขนกล 35
3.7 ท่อนแขนส่วนที่ต่อจากหัวไหล่ 35
3.8 ท่อนแขนส่วนที่ต่อจากข้อศอก 36
3.9 ท่อนแขนส่วนที่ต่อจากข้อมือ 36
3.10 ภาพรวมของการใช้บอร์ดไมโครคอนโทรลเลอร์ควบคุมแขนกล 37
3.11 ภาพรวมแสดงการเชื่อมต่อทั้งหมดของแขนกลอัตโนมัติ 37
3.12 ภาพรวมการทำงานของส่วนโปรแกรมทั้งหมด 38
3.13 หน้าตาโปรแกรมส่วนของคอมพิวเตอร์ 39
3.14 กล้องเว็บแคม 40
4.1 โปรแกรมที่เครื่องคอมพิวเตอร์ 43
4.2 โปรแกรมที่เครื่องคอมพิวเตอร์เมื่อกล้องเว็บแคมเริ่มจับภาพ 44
4.3 แสดงจุดอ้างอิงต่างๆของแขนกลอัตโนมัติ 45
4.4 วัตถุสีเหลี่ยมผืนผ้าสีแดงที่ใช้ในการทดสอบ 45
4.5 โปรแกรมแสดงวัตถุสีแดงอยู่ตรงกลางหนึ่งชั้น 46
4.6 ผลการทดสอบหยิบจับวัตถุสีแดงตรงกลาง 46
4.7 ภาพการนำวัตถุสีแดงไปวางไว้ตรงจุดที่กำหนด 47

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.8 วัตถุสี่เหลี่ยมผืนผ้าที่ใช้ในการทดสอบ 48

4.9 โปรแกรมแสดงการวางวัตถุสีเหลืองซ้ายสุดหนึ่งชั้น 49

4.10 โปรแกรมแสดงขณะทำการทดสอบหยิบจับวัตถุสีเหลืองซ้ายสุด 49

4.11 ภาพการนำวัตถุสีเหลืองไปวางไว้จุดที่กำหนด 50

4.12 วัตถุสี่เหลี่ยมผืนผ้าสีน้ำเงินที่ใช้ในการทดสอบ 51

4.13 โปรแกรมแสดงการวางวัตถุสีน้ำเงินไว้ขวาสุดหนึ่งชั้น 52

4.14 โปรแกรมแสดงขณะหยิบจับวัตถุสีน้ำเงินขวาสุดหนึ่งชั้น 52

4.15 โปรแกรมแสดงวัตถุสีแดงขวาสุดสองชั้น 54

4.16 โปรแกรมแสดงวัตถุสีน้ำเงินตรงกลางสองชั้น 55

4.17 โปรแกรมแสดงวัตถุสีแดงขวาสุดสามชั้น 57

4.18 โปรแกรมแสดงวัตถุสีเหลืองตรงกลางสามชั้น 58

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

เนื่องจากในปัจจุบันเทคโนโลยีในด้านต่างๆมีการวิวัฒนาการก้าวหน้าเป็นอย่างมากไม่ว่าจะเป็นด้านเกษตรกรรม ด้านอุตสาหกรรม ที่เกี่ยวข้องกับการผลิตสินค้าต่างๆ ลักษณะงานบางอย่างมีความเหมาะสมที่จะใช้หุ่นยนต์ที่ควบคุมโดยระบบอัตโนมัติด้วยปัจจัยหลายประการ เช่น ความซ้ำซากจำเจของงาน ความรวดเร็วในการทำงาน ความปลอดภัยของมนุษย์ในการเข้าถึงกระบวนการตลอดจน ความได้เปรียบทางกายภาพของหุ่นยนต์ เป็นต้น

คณะผู้จัดทำจึงมีแนวคิดในการออกแบบและสร้างแขนกลอัตโนมัติหยิบจับสิ่งของโดยจะมุ่งเน้นแนวทางในการนำไปใช้ในโรงงานอุตสาหกรรม ซึ่งเป็นอุปกรณ์ที่สามารถทำงานแทนมนุษย์ในพื้นที่ที่เสี่ยงอันตรายซึ่งสามารถนำไปพัฒนาให้สามารถทำงานที่ซับซ้อนและมีประสิทธิภาพมากขึ้นต่อไปในอนาคตได้

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการทำงานของแขนกลที่มีในโรงงานอุตสาหกรรม
2. เพื่อศึกษาการใช้ไมโครคอนโทลเลอร์ในการควบคุมอุปกรณ์ต่างๆ
3. เพื่อศึกษาระบบควบคุมที่สามารถควบคุมตำแหน่งภารหมุนของมอเตอร์ไฟฟ้ากระแสตรงได้
4. เพื่อศึกษาการรับส่งข้อมูลระหว่างไมโครคอมพิวเตอร์ผ่านระบบประมวลผลภาพ
5. เพื่อนำแขนกลอัตโนมัติที่สร้างขึ้นมาใช้ในการควบคุมโดยการประมวลผลภาพ
6. เพื่อนำแขนกลที่สร้างขึ้นมาใช้เป็นแนวทางในการพัฒนาทางด้านการศึกษาและการวิจัยต่อไปในอนาคต

1.3 ขอบเขตของโครงงาน

1. สามารถควบคุมแขนกลอัตโนมัติจากไมโครคอนโทรลเลอร์ 1 ตัว
2. แขนกลอัตโนมัติสามารถหยิบจับวัตถุที่เป็น สีแดง สีเหลือง และสีน้ำเงิน
3. ใช้กล้องในการรับสัญญาณภาพ 2 ตัว เพื่อจับภาพของวัตถุ
4. ตัวแขนกลอัตโนมัติเป็นสีดำพื้นเป็นสีดำ
5. มีการกำหนดเส้นทางล่วงหน้าไว้สำหรับการเคลื่อนที่

1.4 ขั้นตอนการดำเนินงานของโครงงาน

1. ศึกษาทฤษฎีพื้นฐานของไมโครคอนโทรลเลอร์และการนำไปใช้งาน
2. ศึกษาทฤษฎีการควบคุมเซอร์โวมอเตอร์ และสเต็ปมอเตอร์
3. ศึกษาโครงสร้างและการทำงานของแขนกลอัตโนมัติ
4. ศึกษาทฤษฎีการประมวลผลภาพ
5. ศึกษาทฤษฎีและทดลองวางแผนเส้นทางล่วงหน้าให้แขนกลเคลื่อนที่
6. ศึกษาลักษณะการเคลื่อนที่ของแขนกลอัตโนมัติในรูปแบบต่างๆ
7. ทำการออกแบบลักษณะโครงสร้างและเงื่อนไขการทำงานของแขนกล
8. วิเคราะห์การทำงานและลักษณะการเคลื่อนที่ของแขนกลอัตโนมัติ
9. ทดลองการทำงานของแขนกลอัตโนมัติ
10. จัดการปรับปรุงการทำงานของแขนกลอัตโนมัติ
11. วิเคราะห์และสรุปผลการทำงานที่ได้

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	ปี 2553						ปี 2554	
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาการใช้ไมโครคอน-โทลเลอร์ควบคุมอุปกรณ์	←	→						
2. ศึกษาหลักการทำงานของแขนกลอัตโนมัติ		←	→					
3. ศึกษาการส่งข้อมูลเพื่อทำการประมวลผลภาพ			←	→				
4. เขียนโปรแกรมเพื่อควบคุมแขนกลอัตโนมัติ				←	→			
5. ปรับปรุงและแก้ไข					←	→		
6. วิเคราะห์ผลการทำงาน						←	→	
7. สรุปผลการทำงาน							←	→
8. จัดทำเป็นรูปเล่มโครงงาน								← →

1.6 ผลที่คาดว่าจะได้รับ

1. ได้รับต้นแบบแขนกลอัตโนมัติ ที่สามารถนำไปประยุกต์ใช้งานอุตสาหกรรมในที่อันตรายต่อมนุษย์
2. ได้แขนกลอัตโนมัติที่สามารถหยิบจับวัตถุทรงต่างๆ ได้
3. ได้หลักการใช้งานเรื่องการประมวลผลภาพของวัตถุ
4. สามารถทำการควบคุมแขนกลอัตโนมัติที่ถูกต้องแม่นยำ
5. ได้ต้นแบบแขนกลอัตโนมัติที่สามารถนำไปใช้เป็นแนวทางในการพัฒนาการด้านการศึกษาและการวิจัยระบบควบคุมหุ่นยนต์ต่อไปในอนาคต

1.7 งบประมาณ

1. ค่าวัสดุและอุปกรณ์	700	บาท	
2. ค่าจัดทำรูปเล่มรายงาน	300	บาท	
รวมเป็นเงินทั้งสิ้น	1,000	บาท	(หนึ่งพันบาทถ้วน)

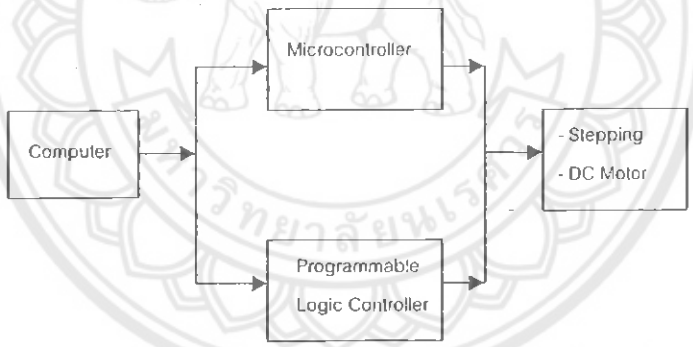
*ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่2
หลักการและทฤษฎี

ในบทนี้จะกล่าวถึงทฤษฎีที่ใช้ในการจัดทำโครงงาน ซึ่งประกอบไปด้วย ทฤษฎีพื้นฐานของหุ่นยนต์ประเภทต่างๆ ทฤษฎีพื้นฐานของไมโครคอนโทรลเลอร์และการนำไปใช้งาน ทฤษฎีการควบคุมเซอร์โวมอเตอร์ และทฤษฎีในการประมวลผลภาพ

2.1 หุ่นยนต์ (Robot) [1]

หุ่นยนต์ หรือ โรบอต (Robot) คือเครื่องจักรกลชนิดหนึ่ง มีลักษณะโครงสร้างและรูปร่างแตกต่างกัน หุ่นยนต์ในแต่ละประเภทจะมีหน้าที่การทำงานในด้านต่างๆ ตามการควบคุมโดยตรงของมนุษย์ มีลักษณะการทำงานแบบอัตโนมัติ (Automatics Machine) หรือกึ่งอัตโนมัติ (Semi automatics Machine) และสามารถเขียนโปรแกรมให้ทำงานอย่างใดอย่างหนึ่ง หรือหลายอย่างได้ ซึ่งมีไดอะแกรมการทำงานดังรูปที่ 2.1

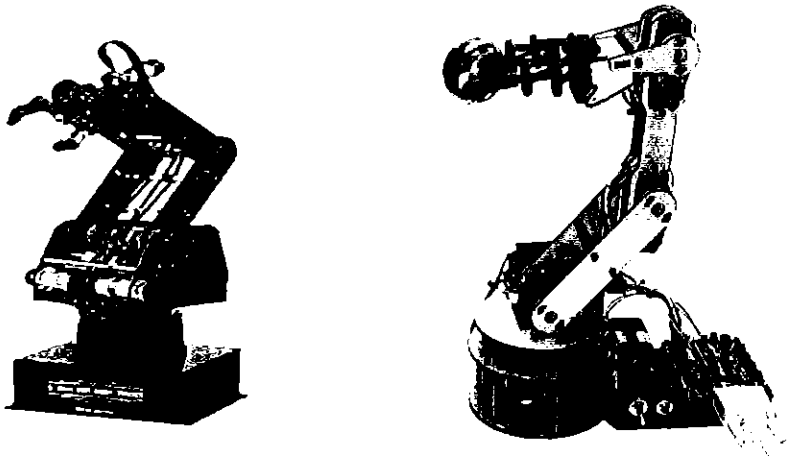


รูปที่ 2.1 ไดอะแกรมการทำงานของ Robot [1]

หุ่นยนต์ถูกแบ่งออกเป็น 2 ประเภทตามลักษณะการใช้งาน คือ

2.1.1 หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (Fixed Robot) [1]

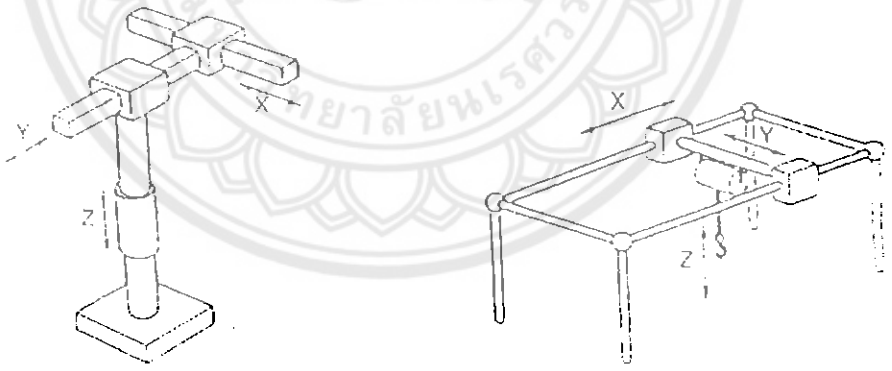
หุ่นยนต์ชนิดนี้ไม่สามารถเคลื่อนที่ไปไหนได้ด้วยตัวเอง หุ่นยนต์ประเภทนี้จะมีลักษณะเป็นแขนกล สามารถเคลื่อนไหวได้เฉพาะแต่ละข้อต่อ ภายในตัวเองเท่านั้น ส่วนมากมักถูกนำไปใช้ในโรงงานอุตสาหกรรม ดังตัวอย่างในรูปที่ 2.2 ซึ่งในปัจจุบันสามารถแบ่งออกตามลักษณะการทำงานได้ 5 ชนิด ดังนี้



รูปที่ 2.2 หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (Fixed Robot) [1]

1. หุ่นยนต์คาร์ทีเซียน (Cartesian Coordinated Robot) [1]

หุ่นยนต์ชนิดนี้ จะมีการเคลื่อนเป็นแนวเส้นตรงทั้ง 3 แกน (Three linear axes) คือ เคลื่อนที่ตามแนวแกน X, Y, Z ดังแสดงในรูปที่ 2.3 ตัวอย่างหุ่นยนต์ชนิดนี้คือ เครนในโรงงานอุตสาหกรรม เครื่องจักร CNC หรือ Inspection เป็นต้น

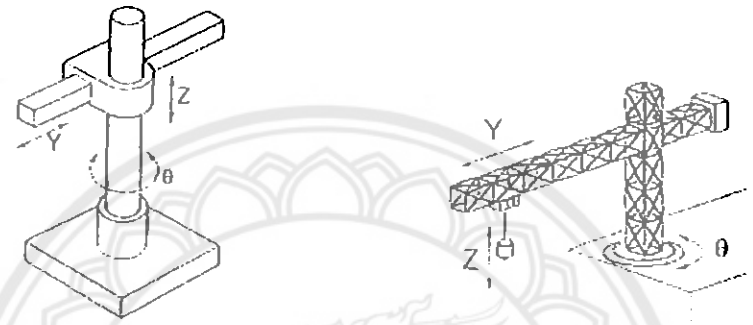


ก) ทิศทางการเคลื่อนที่ ข) ลักษณะการทำงาน

รูปที่ 2.3 หุ่นยนต์คาร์ทีเซียน (Cartesian Coordinated Robot) [1]

2. หุ่นยนต์ทรงกระบอก (Cylindrical Coordinated Robot) [1]

หุ่นยนต์ชนิดนี้ จะมีการเคลื่อนเป็นแนวเส้นตรง 2 แกน (Two linear axes) คือ แกน Y, Z ส่วนฐานจะหมุนรอบเป็นวงกลม (Rotation material handling) ดังแสดงในรูปที่ 2.4 ตัวอย่างหุ่นยนต์ชนิดนี้คือ เครนสำหรับสร้างตึกสูง (Tower Cane) หรืออุปกรณ์ขนถ่ายวัสดุทั่วไป (General material handling) เป็นต้น



ก) ทิศทางการเคลื่อนที่ ข) ลักษณะการทำงาน

รูปที่ 2.4 หุ่นยนต์ทรงกระบอก (Cylindrical Coordinated Robot) [1]

3. หุ่นยนต์ทรงกลม (Spherical Coordinated Robot) [1]

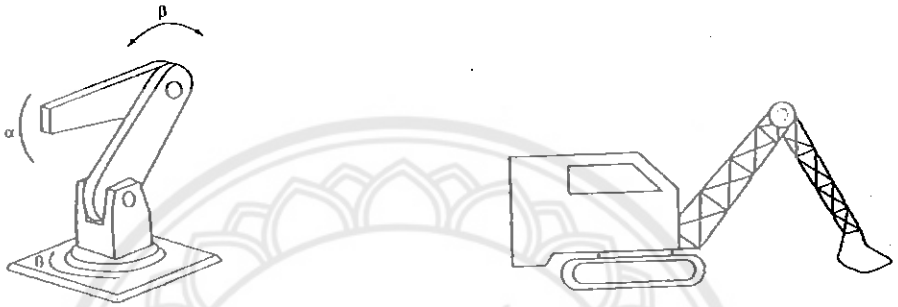
หุ่นยนต์ชนิดนี้จะมีการเคลื่อนที่เป็นแนวเส้นตรง 1 แกน (One linear axes) และเคลื่อนที่แบบหมุน 2 แกน (Two rotating axes) ดังแสดงในรูปที่ 2.5 ตัวอย่างหุ่นยนต์ชนิดนี้ คือ บันไดของรถดับเพลิง, Material transfer หรือ Parts cleaning เป็นต้น

ก) ทิศทางการเคลื่อนที่ ข) ลักษณะการทำงาน

รูปที่ 2.5 หุ่นยนต์ทรงกลม (Spherical Coordinated Robot) [1]

4. หุ่นยนต์ข้อต่อ (Join-Arm Coordinated Robot) [1]

หุ่นยนต์ชนิดนี้จะมีการเคลื่อนที่แบบหมุน 3 แกน (Three rotating axes) โดยการเคลื่อนที่จะมีลักษณะคล้ายกับแขนของมนุษย์ หรือเรียกว่า Revolute coordinates ดังแสดงในรูปที่ 2.6 ตัวอย่างหุ่นยนต์ชนิดนี้คือ งานประกอบ (Assambly), งานเชื่อม (Welding), งานขนถ่ายวัสดุ (Material Handling) เป็นต้น



ก) ทิศทางการเคลื่อนที่ ข) ลักษณะการทำงาน

รูปที่ 2.6 หุ่นยนต์ข้อต่อ (Join-Arm Coordinated Robot) [1]

5. หุ่นยนต์สการ่า (Scara Robot) [1]

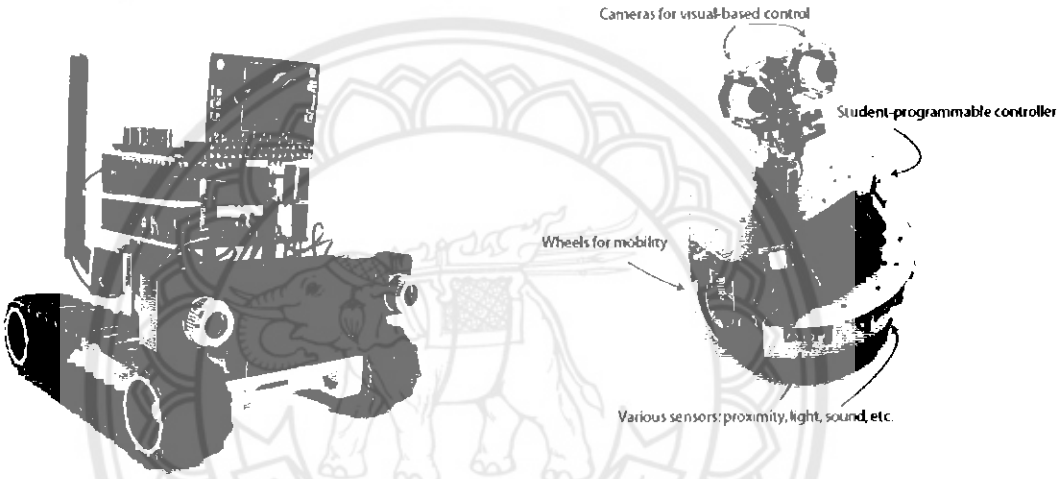
หุ่นยนต์ชนิดนี้จะมีการเคลื่อนที่เป็นแนวเส้นตรง 1 แกน (One linear axes) และเคลื่อนที่แบบหมุนได้ 2 แกน (Two rotating axes) ลักษณะการทำงานจะคล้ายกับหุ่นยนต์ทรงกระบอก ดังแสดงในรูปที่ 2.7 ตัวอย่างหุ่นยนต์ชนิดนี้คือ Automatic assembly, Die casting หรือ Welding เป็นต้น

ก) ทิศทางการเคลื่อนที่ ข) ลักษณะการทำงาน

รูปที่ 2.7 หุ่นยนต์สการ่า (Scara Robot) [1]

2.1.2 หุ่นยนต์ชนิดที่เคลื่อนที่ได้ (Mobile Robot) [1]

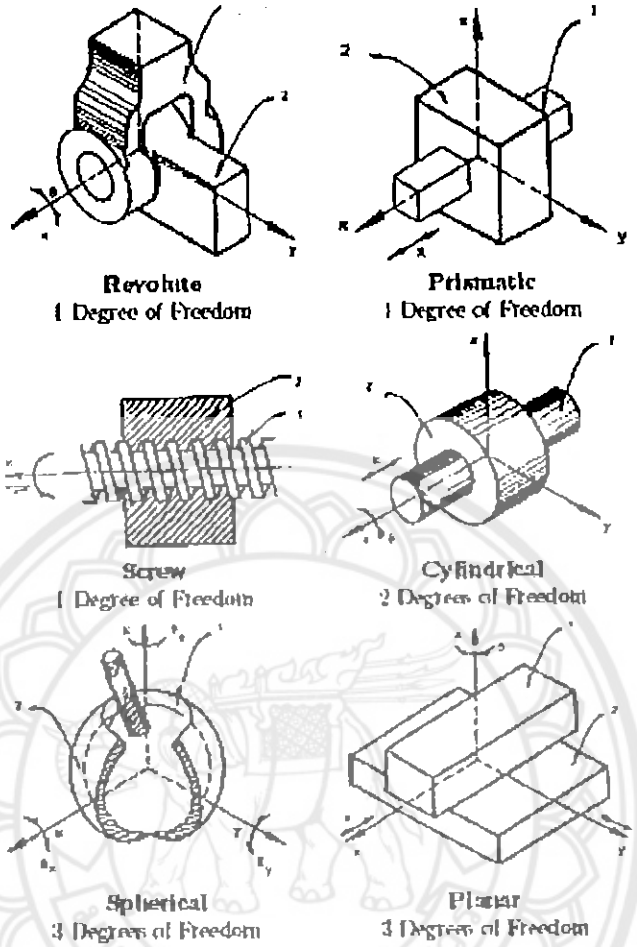
หุ่นยนต์ประเภทนี้จะสามารเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง บ้างก็เคลื่อนที่โดยการใช้ล้อ หรือบางแบบก็เคลื่อนที่โดยการใช้ขา ซึ่งหุ่นยนต์ประเภทนี้ส่วนใหญ่ยังเป็นงานวิจัยที่อยู่ในห้องทดลอง เพื่อพัฒนาออกมาใช้งานในรูปแบบต่างๆ เช่นหุ่นยนต์สำรวจดาวอังคารขององค์การนาซ่า แต่ปัจจุบันก็ได้มีการพัฒนาให้มีลักษณะเป็นสัตว์เลี้ยงอย่างสุนัข เพื่อให้มาเป็นเพื่อนเล่นกับคน หรือแม้กระทั่งมีการพัฒนาหุ่นยนต์ให้สามารถเคลื่อนที่แบบสองขาได้อย่างมนุษย์ เพื่ออนาคตจะสามารถนำไปใช้ในงานที่มีความเสี่ยงต่ออันตรายแทนมนุษย์ ดังตัวอย่างในรูปที่ 2.8



รูปที่ 2.8 หุ่นยนต์ชนิดที่เคลื่อนที่ได้ (Mobile Robot) [1]

2.2 แขนกล (Robot Arm) [2]

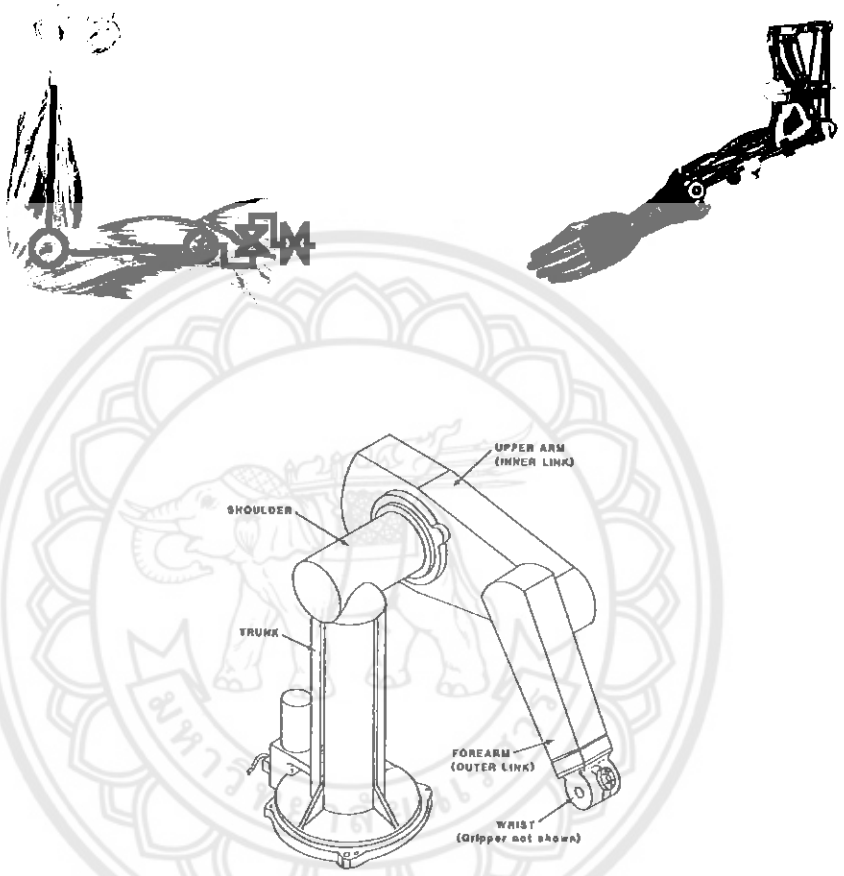
โครงสร้างโดยทั่วไปของหุ่นยนต์แขนกลจะประกอบไปด้วยท่อนแขน (1ink) ที่นำมาประกอบด้วยข้อต่อ (joint) ซึ่งมีหลายแบบ ดังแสดงในรูปที่ 2.9 โดยที่แต่ละแบบก็จะอนุญาตให้เกิดการเคลื่อนที่ของท่อนแขนที่แตกต่างกันไป



รูปที่ 2.9 แสดงข้อต่อแบบต่างๆ ที่นิยมใช้ [2]

ในการสร้างแขนกลโดยทั่วไปข้อต่อที่นิยมใช้มากที่สุดคือ ข้อต่อแบบหมุน (revolute joint) และข้อต่อแบบเลื่อน (prismatic joint) สำหรับข้อต่อแบบหมุน ท่อนแขนสองท่อนถูกยึดติดกันที่จุดหมุนซึ่งอยู่บนท่อนแขน โดยแต่ละท่อนสามารถหมุนได้รอบจุดหมุนนี้ เราสามารถบอกตำแหน่งของสองท่อนแขนที่สัมพันธ์กันด้วยมุมที่ท่อนแขนหมุนไป ส่วนข้อต่อแบบเลื่อนนั้นท่อนแขนสองท่อนติดอยู่ด้วยกันในลักษณะเดียวกันกับอากาศวิทยุรถยนต์ที่ยืดหดได้ โดยท่อนแขนแต่ละท่อนสามารถเลื่อนเข้าออกได้ในหนึ่งทิศทาง เราสามารถระบุตำแหน่งที่สัมพันธ์กันของสองท่อนแขนได้จากระยะเลื่อนเข้าออก ข้อต่อทั้งสองแบบนี้ถูกใช้มากที่สุดในการสร้างแขนกล โดยแขนกลที่มีมุมอิสระ (Degee of Freedom) สูง สามารถสร้างขึ้นได้โดยการประกอบท่อนแขนหลายท่อนด้วยข้อต่อสองแบบนี้ แขนกลทำงานด้วยการเคลื่อนที่ของท่อนแขนที่สัมพันธ์กันเพื่อให้ปลายแขน (end effector) ไปอยู่ในตำแหน่งและทิศทางที่เหมาะสม เพื่อเครื่องมือที่ติดอยู่ที่ปลายแขนจะได้ทำงานที่ต้องการได้โดยสะดวกและมีประสิทธิภาพ

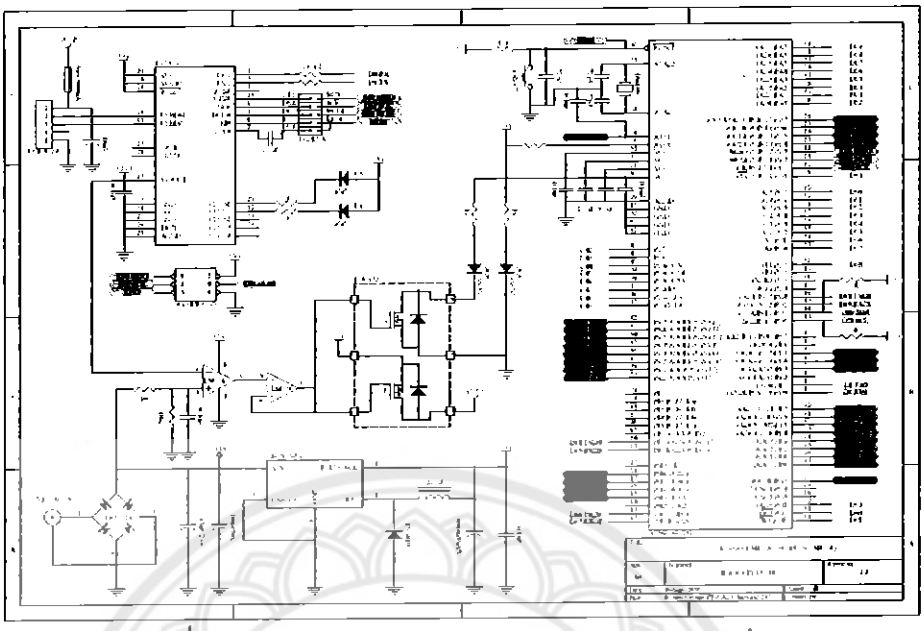
สำหรับท่อนแขนที่เกิดจากการเรียงต่อกันไป เรานิยมเรียกท่อนแขนที่อยู่นิ่งยึดติดกับพื้นว่า ฐาน (base) และเรียกท่อนถัดมาตามชื่อส่วนของแขนว่า ไหล่ (shoulder) ข้อศอก (elbow) แขนท่อนบน(foream) และข้อมือ (wrist) ดังแสดงในรูปที่ 2.10



รูปที่ 2.10 ส่วนประกอบต่างๆของแขนกลเมื่อเปรียบเทียบกับแขนมนุษย์ [2]

2.3 ไมโครคอนโทรลเลอร์ตระกูลของ AVR [3]

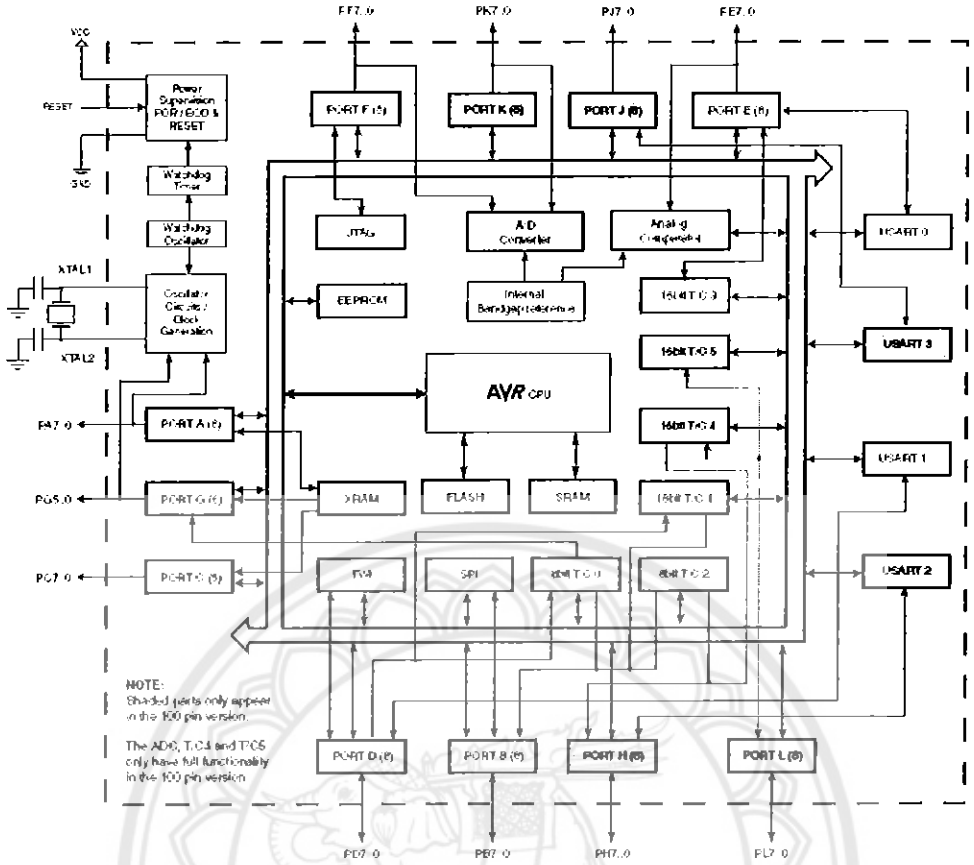
ไมโครคอนโทรลเลอร์ AVR เป็นชิพไมโครคอนโทรลเลอร์ของบริษัท Atmel มีสถาปัตยกรรมภายในเป็นแบบ RISC (reduced instruction set computer) โดยใช้สัญญาณนาฬิกาเพียง 1 ลูกในการปฏิบัติงานใน 1 คำสั่ง โดยจะประกอบด้วยหน่วยความจำโปรแกรมภายในที่เป็นแบบแฟลช์ โปรแกรมข้อมูลได้แบบ In-System programmable และในบางเบอร์ยังสามารถมีการกำหนดตำแหน่งของหน่วยความจำที่สร้างเป็นบูตโหลดเดอร์ (เขียนโปรแกรมเพื่อติดต่อกับ PC หรือไอซีตัวอื่นๆ และยังสามารถโปรแกรมให้กับตัวเองได้) ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวชิพเดียวกัน ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 โครงสร้างภายในของไมโครคอนโทรลเลอร์[3]

2.3 ไมโครคอนโทรลเลอร์ ATmega1280 [3]

ไมโครคอนโทลเลอร์ คือ ชิพไมโครคอนโทรลเลอร์ ATmega1280 ที่เป็นชิพตระกูล AVR ของบริษัท Atmel รองรับการเขียนโปรแกรมภาษาซีของ Arduino ได้ทันทีมาเป็นหน่วยประมวลผลหลัก โดยชิพรุ่นนี้มีหน่วยความจาแฟลชสาหรับเก็บเขียนโปรแกรม 128 กิโลไบต์ มีหน่วยความจาแรม 8 กิโลไบต์ มี EEPROM อีก 4 กิโลไบต์สาหรับใช้เป็นที่เก็บข้อมูลถาวรได้เมื่อไม่มีไฟเลี้ยง และยังมีขาน าเข้า/ส่งออก หรือไอ/โอ (I/O) สาหรับต่อใช้งานทั่วไป 86 ขา มี PWM (สาหรับควบคุมอัตราการหมุนของมอเตอร์) ที่กำหนดความละเอียดได้ระดับ 16 บิตให้ใช้งานถึง 12 ช่องสัญญาณ มีช่องสื่อสารแบบอนุกรม 4 พอร์ต และสามารถแปลงสัญญาณแอนาล็อกเป็นดิจิทัล (ADC) อีก 16 ช่องสัญญาณ ซึ่งจะเห็นว่ามีความสามารถพื้นฐานที่มากพอสาหรับงานควบคุมที่หลากหลาย สามารถทำงานได้โดยใช้ชิปเดียว มีคำสั่งที่สนับสนุนการเขียนโปรแกรมควบคุมและสามารถเข้าถึงข้อมูลระดับบิตได้



รูปที่ 2.12 โครงสร้างอธิบายส่วนประกอบของชิพ ATMEGA1280 [3]

2.3.1 หน้าที่ของพอร์ต A ถึง L มีดังนี้ [3]

Port A (PA7..PA0)

พอร์ต A เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง (bi-directional I/O port) ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port B (PB7..PB0)

พอร์ต B เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) สามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส จุดเด่นของพอร์ตนี้ก็คือ เป็นพอร์ตที่มีความสามารถในการขับกระแสได้ดีกว่าพอร์ตอื่นๆ

Port C (PC7..PC0)

พอร์ต C เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port D (PD7..PD0)

พอร์ต D เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port E (PE7..PE0)

พอร์ต E เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port F (PF7..PF0)

พอร์ต F รองรับการนำเข้าเพื่อท าการแปลงสัญญาณแอนาล็อกมาเป็นดิจิทัล

พอร์ต F เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

พอร์ต F รองรับหน้าที่การเชื่อมประสานกับ JTAG และถ้ามีการเปิดการท างานการเชื่อมประสานกับ JTAG ตัวพูลอัพของขา PF7(TDI), PF5(TMS), และ PF4(TCK)จะทำงานจนกว่าจะเกิดการรีเซ็ต

Port G (PG5..PG0)

พอร์ต G เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port H (PH7..PH0)

พอร์ต H เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port J (PJ7..PJ0)

พอร์ต J เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port K (PK7..PK0)

พอร์ต K รองรับการนำเข้าเพื่อทำการแปลงสัญญาณแอนาล็อกมาเป็นดิจิตอล

พอร์ต K เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

Port L (PL7..PL0)

พอร์ต L เป็นไอ/โอพอร์ตแบบทำงาน 2 ทิศทาง ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ส

2.3.2 คุณสมบัติของไมโครคอนโทรลเลอร์ ATmega1280 [3]

– เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตประสิทธิภาพสูงแต่ใช้พลังงานต่างในตระกูล AVR สถาปัตยกรรมแบบ RISC

–มีชุดคาสั่ง 135 คาสั่ง และส่วนใหญ่คาสั่งเหล่านี้จะใช้เพียง 1 สัญญาณนาฬิกาในการประมวลผลคำสั่ง

– มีเรจิสเตอร์สาหรับใช้งานทั่วไปขนาด 8 บิต จานวน 32 ตัว

– ทางานได้สูงสุดที่ 16 ล้านคาสั่งต่อวินาที (MIPS) เมื่อใช้สัญญาณนาฬิกา 16 เมกะเฮิร์ซ (MHz)

–หน่วยความจำ

– หน่วยความจาแฟลชสาหรับโปรแกรมขนาด 128 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง

– หน่วยความจาแบบ EEPROM ขนาด 4 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง

– หน่วยความจาแรมชนิดเอสแรม (SRAM) ขนาด 8 กิโลไบต์

– เก็บข้อมูลได้กว่า 20 ปีที่อุณหภูมิ 85?C และกว่า 100 ปีที่อุณหภูมิ 25?C

– มีระบบโปรแกรมตัวเองอยู่ในตัวชิพ

–สามารถทาการอ่านขณะเขียนได้จริงโดยสามารถล็อกการทางานได้เพื่อความปลอดภัยของซอฟต์แวร์

– มีการเชื่อมประสานกับ JTAG (IEEE std. 1149.1 compliant)

2.3.3 คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก

– มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จานวน 2 ตัว ที่สามารถแยกโหมดการทางานจากกันได้ 2 โหมดคือ Prescalar และ Compare

– มีตัวตั้งเวลาและตัวนับขนาด 16 บิต จานวน 4 ตัว ที่แยกโหมดการทางานได้ 3 โหมด คือ Prescaler,Compare- และ Capture

– มีตัวนับแบบเวลาจริง (Real Time Counter) ที่แยกวงจรกำหนดความถี่ได้

– มี PWM จานวน 12 ช่องสัญญาณที่สามารถกำหนดความละเอียดได้ 16 บิต

– มีตัวปรับผลการเปรียบเทียบของเอาต์พุต

– มีตัวแปลงสัญญาณแอนาล็อกให้เป็นดิจิทัลขนาด 10 บิต จานวน 16 ช่องสัญญาณ

– มีพอร์ตสื่อสารอนุกรมที่สามารถกำหนดอัตราการรับ/ส่งได้จานวน 4 พอร์ต

– เชื่อมประสานอนุกรมแบบ SPI ได้ทั้งการเป็นมาสเตอร์และสเลฟ (Master/Slave)

– มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้นแบบ ส่งข้อมูลแบบเรียงไบต์ (Byte Oriented)

–มีตัวตั้งเวลาแบบวอตช์ด็อกที่สามารถกำหนดการทางานได้โดยสามารถแยกสัญญาณนาฬิกาได้จากตัวชิพ

– มีตัวเปรียบเทียบสัญญาณแบบแอนาล็อกอยู่ในตัว

– มีการรองรับการขัดจังหวะและการเวก-อัพ (Wake-up) เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ

2.3.4 คุณสมบัติพิเศษ [3]

– มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการเกิดบราวน์เอาท์ (Brown-out) ที่สามารถกำหนดการท างานได้

– มีตัวตรวจหาความเที่ยงตรงของออสซิเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)

– มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)

– มีโหมดการทำงานสลีป 6 แบบ คือ : Idle, ADC Noise Reduction, Power-save, Power-down,Standby, และ Extended Standby

– มีขาของไอ/โอที่สามารถกำหนดการท างานได้ 86 ขา

– ตัวถังแบบ TQFP ชนิด 100 ขา

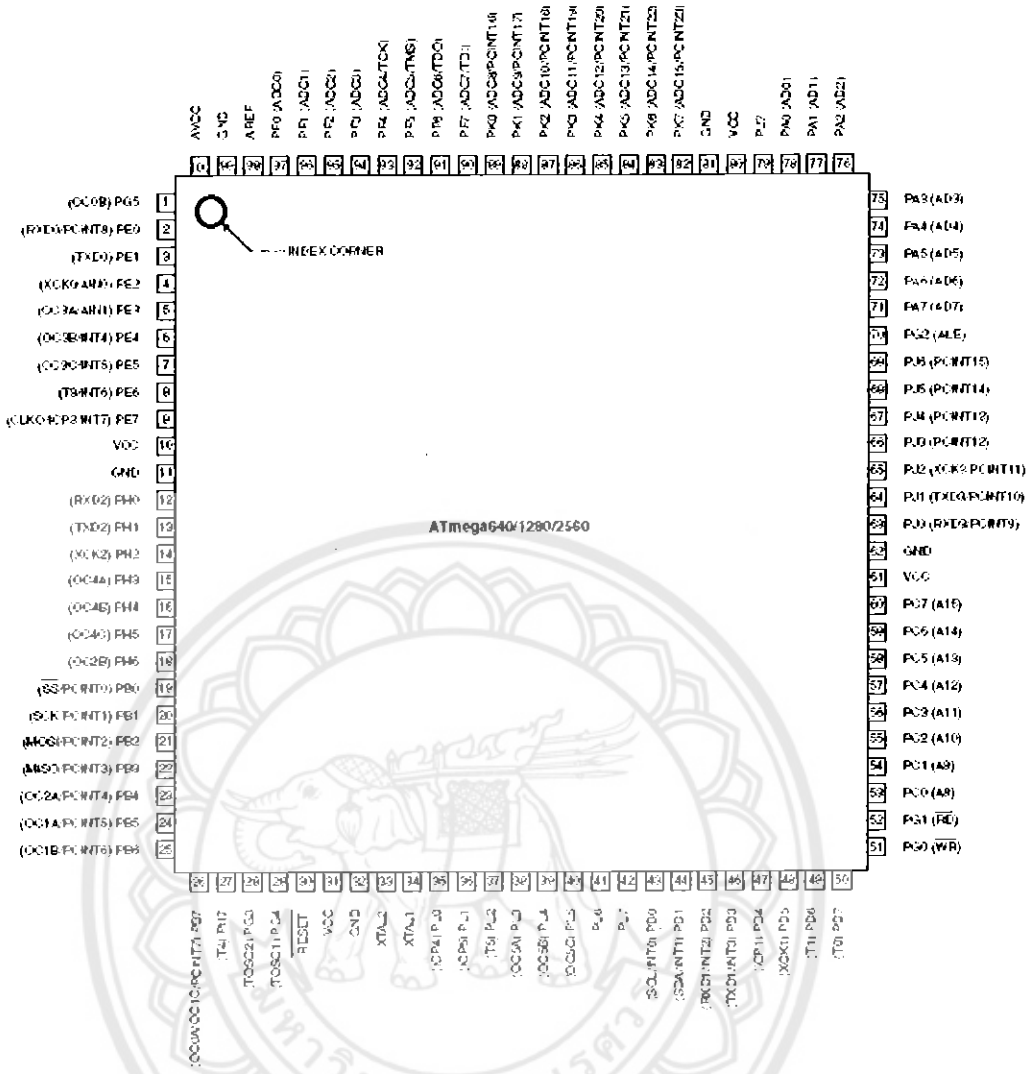
– ช่วงอุณหภูมิที่ชิพทำงานได้ -40°C ถึง 85°C

– โหมดการทำงาน: ที่ 1 MHz ต้องการแรงดัน 1.8V กระแส 500 μA

– โหมดเพาเวอร์ดาวน์ (Power-down) ต้องการกระแสเพียง 0.1 μA ที่แรงดัน 1.8V

2.3.5 ขาต่างๆ ของ ชิพไมโครคอนโทรลเลอร์ ATMEGA1280 [3]

ไมโครคอนโทรลเลอร์ ATMEGA1280 โครงสร้างของ ชิพเป็นแบบ RISC ขนาด 100 ขา โดยขาต่างๆ จะทำหน้าที่เป็นขาควบคุม ขาพอร์ตอินพุต/ พอร์ตเอาต์พุต ที่สามารถกำหนดการทำงานได้ 86 ขา และ ขารีเซต ดังแสดงในรูปที่ 2.12



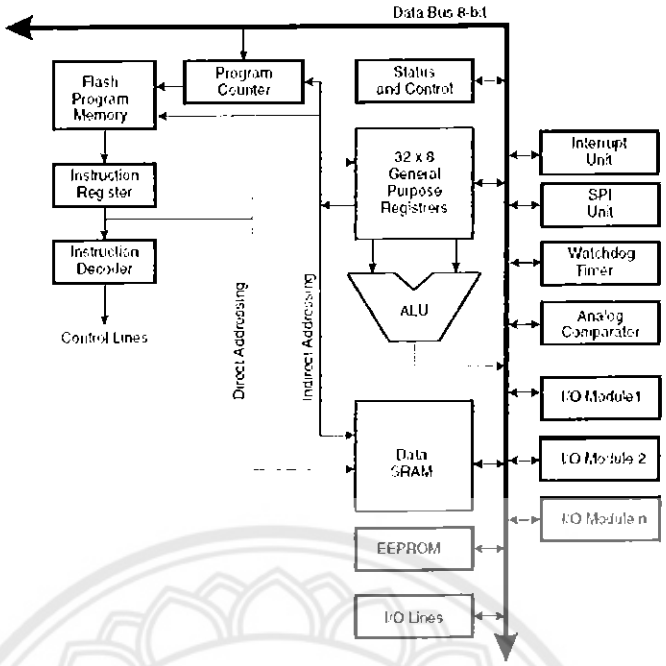
รูปที่ 2.13 ขาต่างๆ ของ ชิพไมโครคอนโทรลเลอร์ ATMEGA1280 [3]

2.3.6 ผังสถาปัตยกรรมของ AVR [3]

จากผังสถาปัตยกรรมของ AVR รีจิสเตอร์เก็บสถานะจัดเก็บสารสนเทศเกี่ยวกับผลของการทำงานที่เกิดจากการประมวลผลชุดสั่งเกี่ยวกับการคำนวณครั้งหลังสุด ซึ่งข้อมูลนี้มีประโยชน์ต่อการนำไปใช้เป็นเงื่อนไขสำหรับการทำงานของคำสั่งถัดไป

รีจิสเตอร์ประเภทนี้จะถูกปรับปรุงค่าหลังจากมีการดำเนินงานด้วยหน่วยประมวลผลทางคณิตศาสตร์และตรรกะหรือเอแอลยู (ALU: Arithmetic and Logic Unit)

รีจิสเตอร์เก็บสถานะจะไม่ถูกจัดเก็บโดยอัตโนมัติเมื่อเข้าสู่การเรียกโปรแกรมย่อยที่ตอบสนองการขัดจังหวะและไม่ถูกนำกลับคืนหลังจากที่การขัดจังหวะนั้นเสร็จสิ้นแล้ว



รูปที่ 2.14 โครงสร้างของ AVR [3]

รูปที่ 2.15 แสดงสถานะของการทำงานด้วยกัน 8 บิต [3]

– Bit 7 – I: Global Interrupt Enable

บิตการเปิดการทำงานของการขัดจังหวะแบบโกบอลจะถูกกำหนดเป็น 1 หรือถูกเซ็ต (set) เพื่อยอมให้มีการทำการขัดจังหวะได้ แต่ถ้าบิตนี้เป็น 0 หรือถูกล้างค่า (cleared) จะหมายความว่าไม่มีการเกิดการขัดจังหวะ ณ ขณะเวลานั้น นั่นหมายความว่า บิต I จะถูกล้างค่าโดยฮาร์ดแวร์เองเมื่อมีเกิดการขัดจังหวะไปเป็นที่เรียบร้อยแล้ว และจะถูกเซ็ตค่าด้วยคำสั่ง RETI เพื่อทำการขัดจังหวะครั้งต่อไป นอกจากนี้เราสามารถเซ็ตและล้างค่าบิต I ได้ด้วยคำสั่ง SEI และ CLI

– Bit 6 – T: Bit Copy Storage

คำสั่งคัดลอกบิตที่ชื่อว่า BLD (Bit LoaD) และ BST (Bit STore) จะใช้บิต T เป็นแหล่งข้อมูลบิตหรือที่เก็บบิตระหว่างที่ทำงาน นั่นหมายความว่า เมื่อเราสั่ง BST จะมีการนำบิตจากรีจิสเตอร์มาเก็บเอาไว้ที่บิต T และเมื่อเราใช้คำสั่ง BLD ก็จะนำบิตจากบิต T ไปเก็บในรีจิสเตอร์

– Bit 5 – H: Half Carry Flag

บิต H จะใช้กับการคำนวณเกี่ยวกับ BCD

– Bit 4 – S: Sign Bit

บิต S เป็นที่เก็บผลของเครื่องหมายของตัวเลข โดยคำนวณมาจาก S = N * V

– Bit 3 – V: Two's Complement Overflow Flag

บิต V ใช้ในการสนับสนุนการทำ 2 คอมพลีเม็นต์ (Twocomplement) ของตัวเลข

– Bit 2 – N: Negative Flag

บิต N เป็นตัวบอกว่าจากการคำนวณหรือดำเนินการทางตรรกศาสตร์นั้นก่อให้เป็นผลลัพธ์ค่าลบหรือไม่

– Bit 1 – Z: Zero Flag

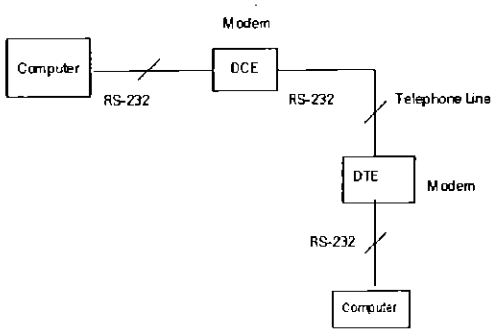
บิต Z เป็นตัวชี้ให้รู้ว่าจากการคำนวณดำเนินการทางตรรกศาสตร์แล้วเกิดค่าเป็น 0

– Bit 0 – C: Carry Flag

บิต C เป็นที่เก็บผลว่าจากการคำนวณหรือดำเนินการทางตรรกศาสตร์แล้วเกิดการยืมบิตกัน

2.4 มาตรฐาน RS-232C [4]

มาตรฐาน RS-232C ได้จัดพิมพ์ขึ้นเมื่อ ค.ศ. 1969 RS ย่อมาจาก Recommended Standard ส่วน 232 คือหมายเลขบ่งบอกมาตรฐานตัวนี้ และ C เป็นหมายเลขฉบับสุดท้ายของมาตรฐานตัวนี้ จุดประสงค์ของมาตรฐาน RS-232 ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูล (DCE : Data Communication Equipment) กับอุปกรณ์สื่อสารข้อมูล (DCE : Data Communication Equipment) สำหรับผู้ใช้คอมพิวเตอร์ทั่วไป DTE ก็หมายถึง ตัวไมโครคอมพิวเตอร์ ส่วน DCE หมายถึง โมเด็ม (modem) และอุปกรณ์อื่นๆ เช่นเครื่องพิมพ์ที่รับสัญญาณแบบอนุกรม อาจจะเป็นไปได้ทั้ง DTE และ DCE ซึ่งจะขึ้นอยู่กับผู้ผลิตสำหรับข้อแตกต่างของ DTE และ DCE ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 การใช้มาตรฐาน RS-232เชื่อมต่ออุปกรณ์ [4]

มาตรฐานการสื่อสารอนุกรม RS-232 นั้น จะมีข้อกำหนดพารามิเตอร์อยู่ 4 ค่าคือ

1. ค่าอัตราความเร็วในการรับส่งข้อมูล (Baud Rate) คือ ค่าอัตราความเร็วในการรับ-ส่งข้อมูลข่าวสารต่อ 1 วินาที หน่วยเป็นบิตต่อวินาที (Bit per second) ซึ่งอัตราความเร็วในการสื่อสารข้อมูลจะมีค่าตั้งแต่ 110 ถึง 76,800 เช่น ถ้าอัตราความเร็วในการรับส่งข้อมูล 9600 ก็หมายถึง การรับ-ส่งข้อมูล 9600 บิตต่อ 1 วินาที ดังแสดงในตารางที่ 2.2

2. ค่าความกว้างข้อมูล (Data width) คือ ข้อมูลที่รับส่งข้อมูลเป็นกลุ่มโดยมีขนาด 7 บิต หรือ 8 บิตขึ้นอยู่กับการสื่อสารรับส่งข้อมูลว่าจะเลือกขนาดใดในการสื่อสารระหว่างกัน

3. ค่าพาริตี้บิต (Parity Bit) พาริตี้บิต เป็นบิตสำหรับตรวจสอบความถูกต้องของข้อมูลในการรับ-ส่งข้อมูล โดยการนับจานวนบิตที่เป็น "1" ในข้อมูลเป็นจำนวนเลขคู่หรือจำนวนเลขคี่ การกำหนดพาริตี้บิตในการสื่อสารข้อมูลมีรูปแบบการกำหนด เช่น พาริตี้คู่ (Even parity) พาริตี้คี่ (odd parlty) หรือไม่มีพาริตี้ (None)

4. ค่าบิตจบ (Stop Bit) เป็นบิตสำหรับปิดท้ายข้อมูล โดยอาจมี 1 บิตหรือ 2 บิต

2.4.1 รูปแบบการสื่อสารรับส่งข้อมูลแบบอนุกรม [4]

ในการสื่อสารอนุกรม จะมีรูปแบบการสื่อสารข้อมูลเป็นกลุ่มบิตซึ่งเรียกว่า เฟรม ใน 1 เฟรมนั้นจะประกอบด้วย

1. บิตเริ่มต้น (Start Bit) มีขนาด 1 บิต เป็นบิตเริ่มต้นที่ทำหน้าที่บอกอุปกรณ์ภาครับข้อมูลว่าข้อมูลกำลังจะมาถึงมีขนาด 1 บิต

2. บิตข้อมูล (Data Bit) มีขนาด 7-8 บิต เป็นกลุ่มบิตที่เป็นข้อมูลในการสื่อสารระหว่างไมโครคอนโทรลเลอร์ กับไมโครคอมพิวเตอร์

3. บิตพาริตี้ (Parity Bit) มีขนาด 1 บิต เป็นบิตในการตรวจสอบความถูกต้องของข้อมูล

4. บิตจบการสื่อสาร (Stop Bit) มีขนาด 1-2 คิดเป็นบิตที่บอกว่าสิ้นสุดข้อมูลในการสื่อสารอนุกรมใน 1 เฟรม จะประกอบด้วย บิตเริ่มต้น 1 บิต บิตข้อมูล 8 บิต และบิตจบหรือบิตสิ้นสุดข้อมูล ส่วนพาริตี้บิตไม่มี ตัวอย่างส่งข้อมูลตัวอักษร A ซึ่งตัวอักษร A มีรหัสแอสกี้คือ 4I H หรือ 01000001 สังเกตจากรูปแบบการส่งข้อมูล ดังแสดงในรูปที่ 2.17

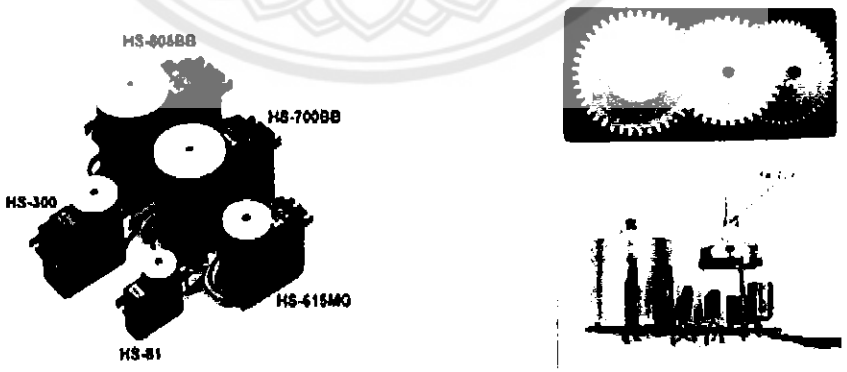
ตารางที่ 2.2 แสดงเวลา Bit Time ใน Baud Rate

Baud rate	Bit time
300	3.33 mS
600	1.66 mS
1200	833 μS
2400	416 μS
4800	208 μS
9600	104 μS
19200	52 μS

รูปที่ 2.17 รูปแบบการสื่อสารอนุกรม 1 เฟรม [4]

2.5 เซอร์โวมอเตอร์ (Servo Motor)[5]

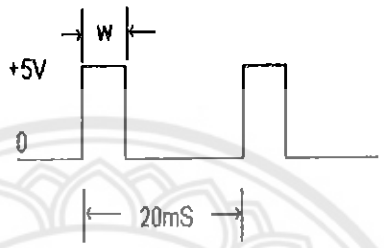
เซอร์โวมอเตอร์ (Servo Motor) คือ มอเตอร์ไฟฟ้ากระแสตรง DC MOTOR ที่ถูกประกอบร่วมด้วยชุดเกียร์และส่วนควบคุมต่างๆ ไว้ในโมดูลเดียวกัน ดังแสดงในรูปที่ 2.17 โดยจะมีสัญญาณใช้งาน 1 เส้น และอีก2 เส้นเป็น VCC และ GND เท่านั้น ซึ่งสามารถควบคุมให้ตัวเซอร์โวมอเตอร์หมุนซ้าย หรือ ขวาได้ +90 องศา - 90 องศา (180 องศา)



รูปที่ 2.18 ส่วนประกอบของเซอร์โวมอเตอร์[5]

2.5.1 หลักการทำงานของเซอร์โวมอเตอร์ [5]

เซอร์โวมอเตอร์สามารถควบคุมการทำงานได้โดยป้อนสัญญาณความกว้าง pulse ให้กับเซอร์โวมอเตอร์ ซึ่งความกว้างของสัญญาณ pulse (w) นี้จะไปกำหนดตำแหน่งของเซอร์โวมอเตอร์ ว่าจะหมุนไปอยู่ตำแหน่งใด ซึ่งเราสามารถป้อนความกว้าง pulse (w) ได้ตั้งแต่ 1.0 ms – 2.0 ms โดยที่คาบเวลา (Period) ในการส่งสัญญาณความกว้าง pulse เท่ากับ 20 ms ดังแสดงในรูปที่ 2.18



รูปที่ 2.19 สัญญาณที่สามารถป้อนให้กับเซอร์โวมอเตอร์ [5]

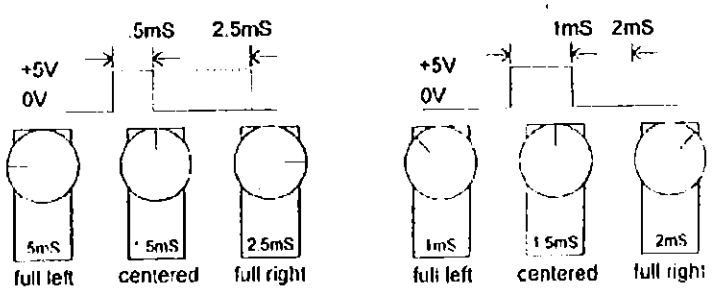
โดยจะมีจุดที่อ้างอิงตำแหน่ง 3 จุดด้วยกันคือ

1. ตำแหน่งซ้ายสุด เราสามารถกำหนดให้เซอร์โวมอเตอร์ เคลื่อนที่ไปยังตำแหน่งซ้ายสุด (-90 องศา) ได้โดยการป้อนความกว้าง pulse (w) เท่ากับ 0.5 ms

2. ตำแหน่งตรงกลาง เราสามารถกำหนดให้เซอร์โวมอเตอร์ เคลื่อนที่ไปยังตำแหน่งตรงกลาง (0 องศา) ได้โดยการป้อนความกว้าง pulse (w) เท่ากับ 1.5 ms

3. ตำแหน่งขวาสุด เราสามารถกำหนดให้เซอร์โวมอเตอร์ เคลื่อนที่ไปยังตำแหน่งซ้ายสุด (90 องศา) ได้โดยการป้อนความกว้าง pulse (w) เท่ากับ 2.5 ms

จากสุดอ้างอิงตำแหน่งทั้ง 3 ทำให้สามารถคำนวณตำแหน่งที่ต้องการให้หมุนไปได้ เช่นถ้าต้องการเซอร์โวมอเตอร์หมุนไปยังตำแหน่ง -45 องศา ก็ทำการป้อน (0.5+1.5)/2 = 1.0 ms ให้กับเซอร์โวมอเตอร์ ดังแสดงในรูปที่ 2.20



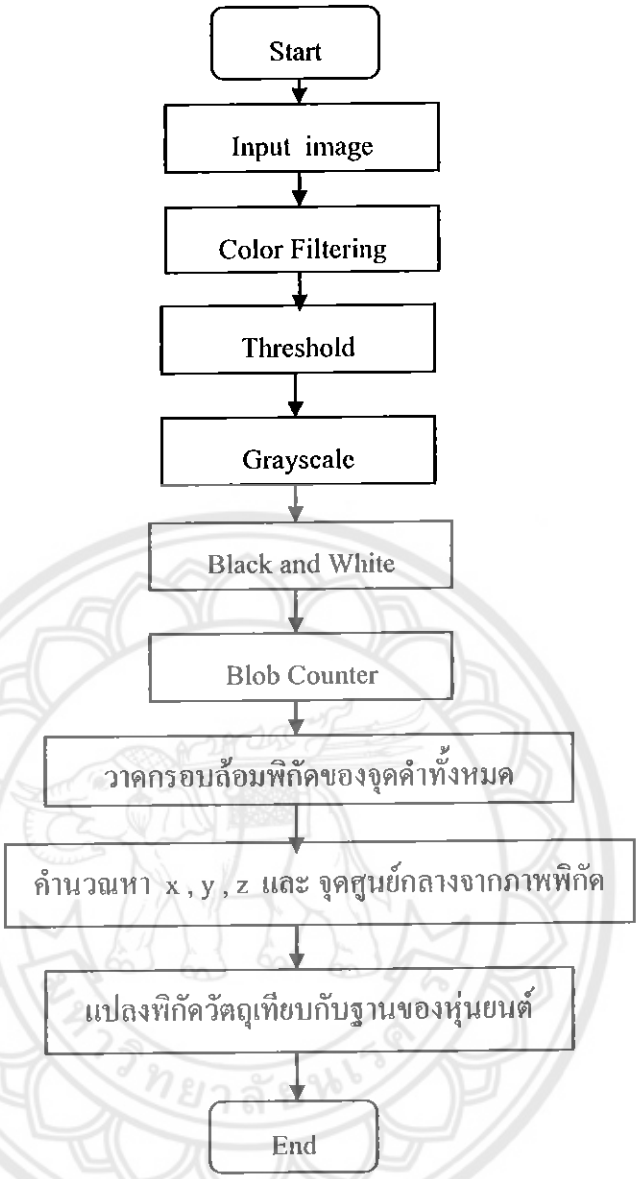
รูปที่ 2.20 ตำแหน่งของเซอร์โวมอเตอร์เมื่อป้อนความกว้าง pulse [5]

2.6 การประมวลผลภาพดิจิตอล (Digital Image Processing) [6]

กระบวนการประมวลผลภาพ หมายถึงการนำเอารูปภาพมาวิเคราะห์ หรือประมวลผลโดยวิธีการทางคณิตศาสตร์เพื่อให้ได้ผลลัพธ์ที่ต้องการ เช่น การปรับปรุงคุณภาพของภาพให้ดีขึ้น การหาตำแหน่งของวัตถุต่างๆ ในภาพ การแปลความหมายจากภาพให้เป็นข้อความด้วยเครื่องอ่านอักขระด้วยแสง(Optical Character Recognition) การตกแต่งภาพ เป็นต้นกระบวนการประมวลผลภาพเริ่มจากการแปลงภาพที่มนุษย์มองเห็นให้เป็นภาพที่คอมพิวเตอร์เข้าใจ ซึ่งปกติแล้วสายตามนุษย์จะมองเห็นภาพเป็นแบบอนาล็อก สามารถอธิบายได้ด้วยคณิตศาสตร์ที่มีตัวแปรแบบนับได้อย่างต่อเนื่อง แต่เครื่องคอมพิวเตอร์จะใช้เลขฐานสองเป็นหลักในการคำนวณ จึงจำเป็นต้องนำภาพอนาล็อก มาแปลงเป็นภาพดิจิตอลก่อนเข้าสู่เครื่องคอมพิวเตอร์

2.6.1 การทำกระบวนการประมวลผลภาพ [6]

การทำกระบวนการประมวลผลภาพ คือ การนำภาพดิจิตอลมาทำกระบวนการทางคณิตศาสตร์โดยการใส่อัลกอริทึมต่างๆเข้าไป จะได้ผลลัพธ์ออกมา ดังแสดงในรูปที่ 2.20 หัวใจสำคัญของกระบวนการประมวลผลภาพคือ การวิเคราะห์ภาพด้วยกระบวนการคณิตศาสตร์ ซึ่งมีขั้นตอนดังต่อไปนี้ เริ่มจากการรับภาพวัตถุจากกล้องเว็บแคมเข้ามา เพื่อแปลงให้อยู่ในรูปแบบของเอกสารภาพเชิงดิจิตอล จากนั้นจึงเข้าสู่ขั้นตอนการประมวลผลภาพ โดยมีขั้นตอนต่าง ๆ ดังนี้ เริ่มแรกรับภาพวัตถุสีจากกล้องเว็บแคมก่อนแล้วทำการกรองค่าสีที่ต้องการเพื่อให้เหลือเพียงวัตถุที่เป็นภาพสีกับพื้นหลังจากนั้นแปลงภาพสีให้เป็นภาพระดับเทาและก็แปลงภาพระดับเทาให้เป็นภาพขาวดำและก็จำแนกวัตถุออกจากพื้นหลัง ผลลัพธ์ที่ได้ก็คือ วัตถุที่ต้องการให้แขนกลอัตโนมัติเข้าไปจับ

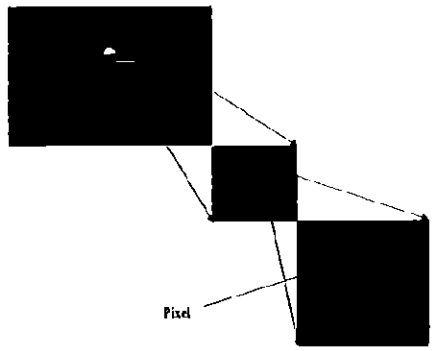


รูปที่ 2.21 กระบวนการทำงานแยกวัตถุออกจากพื้นหลัง [6]

2.6.2 ภาพดิจิตอล (Digital Image) [6]

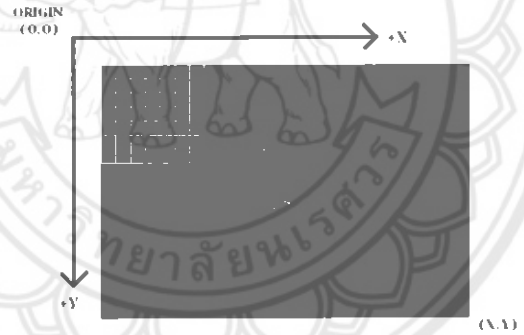
ภาพดิจิตอลเป็นผลมาจากการสุ่มค่าในปริภูมิ (Space) และทำการแปลงสัญญาณจากอนาล็อกให้เป็นดิจิตอล ของค่าความส่องสว่าง (Brightness) ปริภูมินี้ใช้ได้กับการแสดงภาพดิจิตอล ซึ่งมีความกว้างและความสูงของภาพแสดงในแกน X และY ตามลำดับ ส่วนจุดใดๆ ที่วางบนระนาบ XY จะเป็นฟังก์ชั่น f(x, y) เรียกว่า จุดภาพ (Pixel) ภาพที่เก็บในคอมพิวเตอร์จะถูกจัดเก็บในรูปของพิกเซล หมายถึง ส่วนประกอบมูลฐานของภาพ ซึ่งจะเก็บค่าของจุดสีหรือความเข้มของสีในรูป RGB เมื่อซูมภาพเข้าไปจะเป็นจุดสีหรือช่องสี่เหลี่ยมสีเดียว ดังรูปที่ 2.22 แต่ละจุดสีนั้น คือ พิกเซล นั่นเองหากภาพที่มีความละเอียด

มากจะมีจำนวนพิกเซล มากตามไปด้วยและจะใช้พื้นที่ในการเก็บข้อมูลและประมวลผลภาพมากขึ้น



รูปที่ 2.22 ภาพขยายของรูปภาพ [6]

ภาพ 2 มิติที่เก็บในคอมพิวเตอร์จะมีการบันทึกตำแหน่งของแต่ละพิกเซล ในรูปของโคออร์ดิเนต (X , Y) ซึ่ง X , Y จะมีจุดกำเนิดอยู่ที่มุมบนซ้ายของภาพและมีค่าไปในทิศทางบวกไปหามุมล่างขวาของภาพนั้นๆ



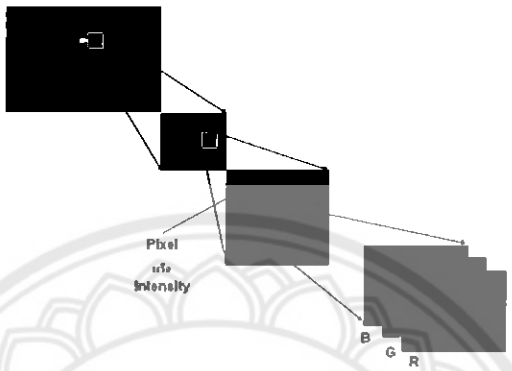
รูปที่ 2.23 โคออร์ดิเนตของภาพ 2 มิติ [6]

ลักษณะของการเก็บข้อมูลในรูปสีต่างๆโดยพิจารณาจากขนาดของข้อมูลมีดังนี้

1. ขาวดำ จะเก็บค่าตำแหน่ง (X, Y) เป็น (0.0) จนถึงค่าสูงสุด ให้แต่ละพิกเซล มีค่าเป็น 0 หรือ 1 เมื่ออ่านค่าในแต่ละแถวก็จะให้ข้อมูลแบบดิจิตอลออกมา
2. Gray Scale จะเก็บค่าสีในแต่ละพิกเซล ขนาด 8 บิตหรือ 1 ไบต์ช่องที่มีค่าเป็น 0 ก็คือสีดำไปจนถึงช่องที่มีค่าเป็น 255 นั่นคือสีขาวส่วนค่าระหว่างนั้นจะออกเป็นสีเทา
3. RGB (8 บิต) จะเก็บค่าสีในแต่ละพิกเซลเป็นค่าสีแดง เขียว และน้ำเงิน (R , G , B) ขนาด 8 บิต คือ R 8 บิต, G 8 บิต, B 8 บิตดังนั้น RGB Image 1 พิกเซลจะประกอบ ไปด้วยจำนวนบิตทั้งหมด 24 บิต ทำให้ RGB Image มีจำนวนสีที่เป็นไปได้ทั้งหมด 2^{24} สี ตัวอย่าง ดังรูปที่ 2.25 ซึ่งแต่ละสีจะมีค่าตั้งแต่ 0 - 255 โดยจะ

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

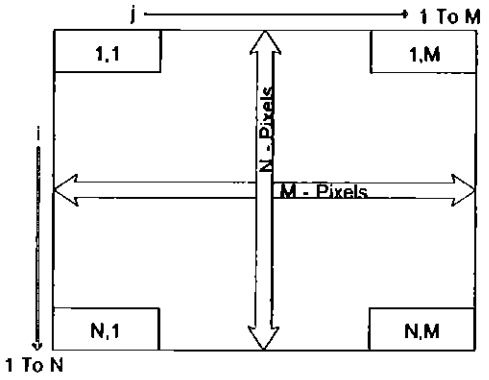
เรียงลำดับในรูป (R,G,B) เมื่อนำค่ามารวมกันก็จะทำให้ในแต่ละพิกเซล มีค่าสีต่างๆ ดังตัวอย่าง เช่น (0 , 0 , 0) เป็นสีดำ (255 , 255, 255) เป็นสีขาว (255 , 0 , 0) สีแดง และ (200, 200, 200) เป็นสีเทา เป็นต้น และในรูปที่ 2.25 จะเห็นว่าค่า RGB มีค่าเป็น (99 , 65, 10) ซึ่งจะออกเป็นสีแดง ผลของการผสมแสง RGB



จากรูป 2.24 แสดงภาพค่า RGB ในแต่ละพิกเซล [6]

2.6.3 พิกเซล (Pixel) [6]

ภาพสามารถอธิบายได้ด้วยค่าของพิกเซลที่อยู่ในรูปของเมตริกซ์ขนาด N×M แต่ละตำแหน่งของพิกเซลจะแทนด้วย P(x,y) ซึ่งจะแสดงถึงความเข้มของแสงบนรูปภาพที่ตำแหน่ง (X,Y) ดังแสดงในรูปที่ 2.24 ความสัมพันธ์ระหว่าง Picture element ของภาพกับพิกเซลของเมตริกซ์ของจุดเริ่มต้นจะอยู่คนละตำแหน่งกัน ภาพจะมีจุดเริ่มต้นที่มุมล่างด้านซ้าย ส่วนเมตริกซ์มี พิกเซลเริ่มต้นอยู่มุมบนด้านซ้ายค่าจำนวน (Numerical Value) หรือขนาด (Magnitude) ของพิกเซลแสดงด้วยค่าเฉลี่ยความเข้มของแสงบนพื้นที่ Pixel element ซึ่งแทนด้วย P(x,y) มีช่วง 0 ถึง 1



รูปที่ 2.25 ความสัมพันธ์ระหว่าง picture elements ของภาพกับพิกเซล [6]

2.6.4 Color filtering [7]

เป็นการประมวลผลภาพแบบกรองค่าพิกเซลที่อยู่ภายในหรือภายนอกช่วงที่กำหนดของสี RGB ซึ่งการกรองค่าพิกเซลจะเป็นการคัดแยกวัตถุที่เป็นสีกับพื้นหลังให้ออกจากกันโดยจะกำหนดค่าตัวเลขของ RGB ระหว่าง 0–255 ซึ่งเป็นค่าของสีที่ต้องการ

2.6.5 Blob Counter [7]

Blob counter มีประโยชน์มากสำหรับการประมวลผลภาพ คือ สามารถนับจำนวนวัตถุที่อยู่บนพื้นหลังในรูปแบบภาพแบบไบนารีหรือดึงภาพที่เล็กใหญ่ออกมาจากพื้นหลัง ซึ่งมีความคิดมาจาก ภาพที่มีการเชื่อมต่อกันและแยกวัตถุแต่ละสีที่มีความแตกต่างกันได้

2.6.6 จุดภาพแบบสเกลสีเทา (Grayscale Image) [6]

ค่าในแต่ละพิกเซลของ Gray Image คือค่าความเข้มของแสง ณ แต่ละตำแหน่งของพิกเซล ซึ่งจะอยู่ในรูปของ Gray Scale ดังรูปที่ 2.25 ขั้นตอนการแปลงภาพสีให้เป็นภาพระดับเทา มีค่าตั้งแต่ 0 (สีดำ)จนถึง 255 (สีขาว) ทำได้โดยแยกระดับสีแต่ละพิกเซล(Pixel) ออกจากกันในรูปแบบสี RGB จากนั้นนำค่าสี RGB มาเข้าสูสมการเพื่อคำนวณหาค่าสีเทา และนำค่าที่ได้ไปแทนที่จุดพิกเซลเดิมโดยคำนวณได้จากสมการดังต่อไปนี้

$$G' = 0.3R + 0.59G + 0.11B \quad \text{หรือ} \quad G' = \frac{R+G+B}{3}$$

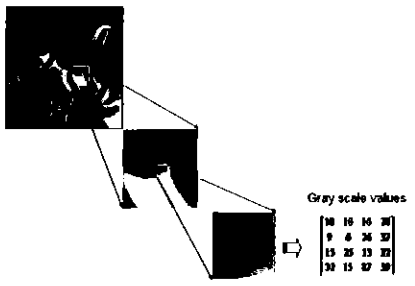
โดยกำหนดให้

G' คือค่าระดับสีเทา

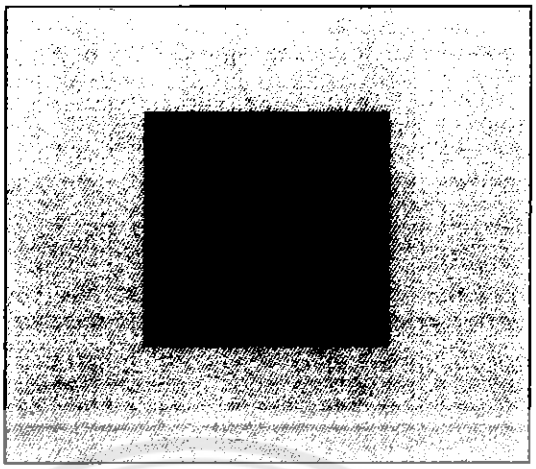
R คือค่าระดับสีแดง

G คือค่าระดับสีเขียว

B คือค่าระดับสีน้ำเงิน



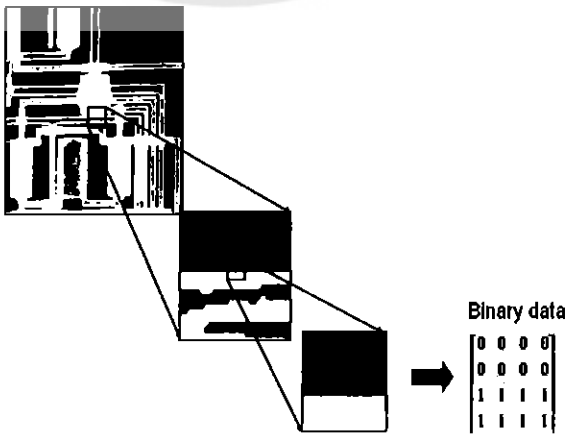
รูปที่ 2.26 ภาพเชิงดิจิตอลประเภท Intensity Image [6]



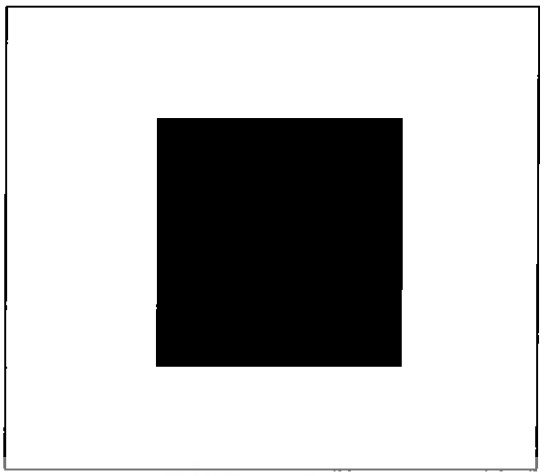
รูปที่ 2.27 ตัวอย่างภาพสเกลสีเทา [6]

2.6.7 ภาพขาวดำ (Black and White Image) [6]

ค่าในแต่ละพิกเซล ของ Black and White Image จะใช้แค่ 1 บิต ซึ่งจะมีค่าที่เป็นไปได้คือ 0 (สีดำ) และ 1 (สีขาว) เท่านั้น ดังรูปที่ 2.27 ขั้นตอนการแปลงภาพสีเทา ให้กลายเป็นภาพขาวดำ จะทำให้สามารถแยกวัตถุ ออกจากพื้นหลังได้ โดยอาศัยวิธีการทำเธรดโซลด์ (Threshold) เธรดโซลด์เป็นวิธีที่ใช้ในการแปลงภาพสีต่าง ๆ ให้อยู่ในรูปแบบของภาพระดับขาวดำ โดยใช้ค่าเธรดโซลด์ในการจำแนกวัตถุ และพื้นหลังออกจากกัน หากจุดใดในภาพนั้นมีค่าความเข้มน้อยกว่าค่าเธรดโซลด์จุดภาพนั้นก็จะถูกปรับให้เป็นสีขาว แต่ถ้าจุดใดในภาพมีค่าความเข้มมากกว่าค่าเธรดโซลด์ ก็จะถูกปรับให้เป็นสีดำ เป็นต้น



รูปที่ 2.28 ภาพเชิงดิจิตอลประเภท Binary Image [6]



รูปที่ 2.29 ตัวอย่าง Binary Image [6]

2.7 การแยกบริเวณ(Segmentation) [9]

กระบวนการนี้เป็นการแยกบริเวณของภาพที่มีลักษณะร่วมกันออกเป็นส่วนๆ ซึ่งมุ่งเน้นว่าจะใช้วิธีใดหรือเงื่อนไขใดในการแยกวัตถุที่สนใจออกจากฉากหลัง หรือจะใช้เงื่อนไขใดในการพิจารณาว่า พิกเซลที่กำลังพิจารณาอยู่นั้น จัดเป็นวัตถุ(Objects) ใดที่อยู่ในภาพ หรือบริเวณใดที่จัดเป็นฉากหลัง (background)

2.7.1 การแยกบริเวณโดยการใช้ Threshold [9]

ค่า Threshold เป็นค่าที่เป็นจำนวนเต็มที่มีค่าระหว่าง 0 ถึง 255 เช่นเดียวกับความเข้มแสงของพิกเซลที่อยู่ในภาพสีเทา (Gray scale) สำหรับการแยกบริเวณโดยการใช้ค่า Threshold นั้น จะเป็นการแปลงภาพแบบสเกลสีเทาให้เปลี่ยนเป็นภาพที่มีเพียง 2 ระดับ (binary image) โดยการใช้เงื่อนไขว่าถ้าค่าความเข้มแสงที่พิกเซลตำแหน่งใดมีค่าต่ำกว่าหรือเท่ากับค่า Threshold ให้ค่าพิกเซลตำแหน่งนั้นมีค่าเป็น 0 หรือเป็นสีดำ และถ้าพิกเซลใดมีค่าสูงกว่าค่า Threshold แล้วให้พิกเซลนั้นมีค่าเป็น 1 หรือ 255 ให้เปลี่ยนเป็นสีขาว

2.7.2 เทคนิคการหารตำแหน่งวัตถุ [6]

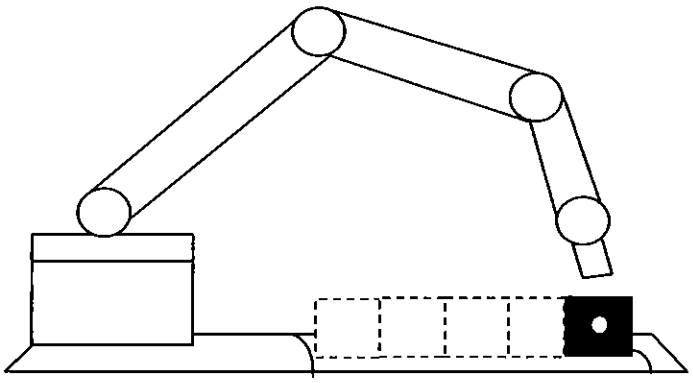
เมื่อภาพได้ถูกผ่านการแยก/กรองระดับสี จะมีค่าของของสีแต่ละจุดภาพอยู่สองค่า คือ 0 กับ 255 จะแทนค่านี้ให้อยู่ในรูปเลขไบนารี่คือ ค่า 0 ลอจิก 0 ค่า 255 ลอจิก 1

	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
2	1	1	0	0	0	0	1	1	1
3	1	1	0	0	0	0	1	1	1
4	1	1	0	0	0	0	1	1	1
5	1	1	0	0	0	0	1	1	1
6	1	1	0	0	0	0	1	1	1
7	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1

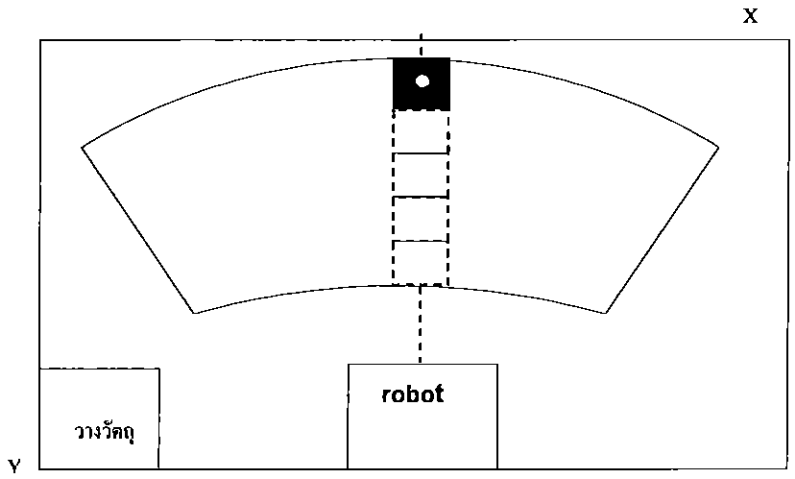
รูปที่ 2.30 ตัวอย่างค่าของภาพที่ใช้ในการคำนวณ [6]

2.8 การวางแผนเส้นทาง (Path Planning) [8]

งานที่แขนกลต้องทำคือให้เคลื่อนที่ไปตามเส้นทางที่กำหนดไว้ล่วงหน้า ในที่นี้ให้แขนกลใช้เส้นทางเดินแบบเส้นตรงซึ่งแสดงไว้ในรูปที่ 2.31 คือแสดงการเคลื่อนที่ของแขนกลมองจากกล้องด้านข้าง และ รูปที่ 2.32 คือแสดงการภาพแขนกลมองจากกล้องด้านบน โดยมีจุดเริ่มต้นที่ตัวฐานแขนกลที่พิกัด (320) และจุดสุดท้ายจะต้องเคลื่อนที่ไปยังจุดตรงกลางของวัตถุเพื่อที่จะจับวัตถุ ซึ่งการทำงานของแขนกลจะเป็นการควบคุมปุ่มให้แขนกลเข้าไปจับวัตถุแล้วให้จดจำค่าองศาของเซอร์โวแต่ละตัวแล้วนำค่าองศาที่ได้ไปเขียนโปรแกรมเพิ่มในส่วนของไทม์เมอร์เพื่อควบคุมให้แขนกลไปจับวัตถุเองโดยอัตโนมัติ การให้แขนกลจดจำค่าองศานั้นในขอบเขตที่ใช้ในการวางวัตถุหนึ่งชั้นจะมีการเซตวัตถุทั้งหมด 5 ครั้ง เริ่มจากไกลสุดลงมาใกล้ๆที่ตัวฐานของแขนกล ซึ่งจะมีวางวัตถุเป็นสองชั้นและสามชั้นการกระทำดังกล่าวรวมกันจะทำทั้งหมด 15 ครั้ง และการเข้าไปจับวัตถุนั้นจะเดินเป็นเส้นตรง รวมทั้งการเคลื่อนที่ไปจับวัตถุจะอยู่ในช่วงขอบเขตที่กำหนดเท่านั้น



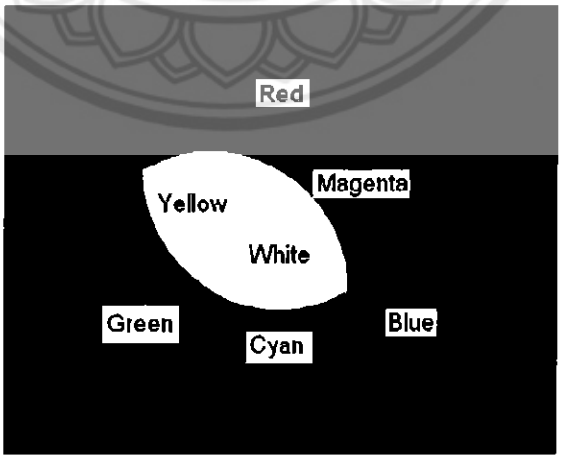
รูปที่ 2.31 แสดงการเคลื่อนที่ของแขนกลมองจากกล้องด้านข้าง [8]



รูปที่ 2.32 แสดงการเคลื่อนที่ของแขนกลมองจากกล้องด้านบนลงมา [8]

2.9 ทฤษฎีการประสมแสง [13]

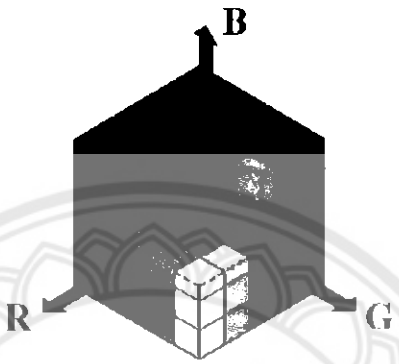
แสงจริงๆที่เรามองเห็นประกอบด้วยสี 3 สีคือ Red Green และ Blue (RGB) ดังรูปที่ 2.32 เมื่อทั้ง 3 สีรวมกันเราจะมองเห็นเป็นสีขาว ทั้งนี้จะขึ้นกับความเข้มของแต่ละสีด้วย เช่นถ้าเราเอา Red ผสมกับ Blue ที่ความเข้มสูงสุดจะกลายเป็นสี Magenta ดังภาพ ส่วนสีดำนั้นเกิดจากการที่ไม่มีแสงสีใดเลย และหากความเข้มของสีใดสีหนึ่งเปลี่ยนแปลง ก็จะทำให้ผลของสีอื่นเด่นชัดขึ้น เช่นเดียวกับการเก็บข้อมูลภาพ สีต่างๆที่เราเห็นในภาพนั้นจะประกอบด้วยค่า RGB ซึ่งสามารถนำมาวิเคราะห์ค่าสีได้



จากรูป 2.33 แสดงการผสมสี [13]

2.9.1 แบบจำลองระบบสี RGB [13]

แบบจำลอง RGB นั้นจะมีโมเดลเป็นทรงสีเหลี่ยมลูกบาศก์โดยมีค่า R, G และ B เป็นแกนดังรูปที่ 2.33 ซึ่งระบบสีแบบนี้จะมีข้อเสียคือหากค่าแสงมีการเปลี่ยนแปลงจะทำให้ค่า RGB ของวัตถุเปลี่ยนแปลงไปด้วย



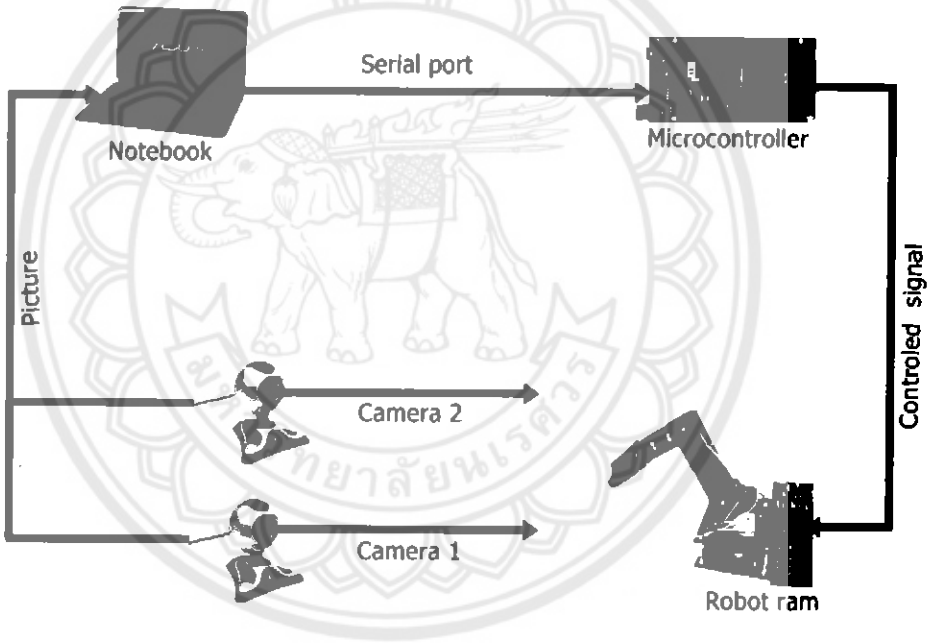
รูปที่ 2.34 แบบจำลองระบบสี RGB [13]

จากการศึกษาทฤษฎีในบทนี้ จะเป็นพื้นฐานซึ่งนำไปสู่การพัฒนาระบบควบคุมแขนกลอัตโนมัติโดยการประมวลผลภาพ การสร้างบอร์ดที่ใช้ขับมอเตอร์ที่อยู่ในส่วนข้อต่อต่างๆของแขนกล และการเขียนโปรแกรมเพื่อควบคุมอุปกรณ์ต่างๆ ซึ่งจะกล่าวถึงในบทต่อไป

บทที่ 3
วิธีการดำเนินโครงงาน

จากการศึกษาหลักการและทฤษฎีต่างๆในบทที่ 2 จึงได้ทำการพัฒนาระบบควบคุมแขนกลอัตโนมัติ ดังนี้

3.1 ภาพรวมระบบของแขนกล (Robot Arm System Overview)



รูปที่ 3.1 ไดอะแกรมแสดงการทำงานของระบบควบคุมแขนกล

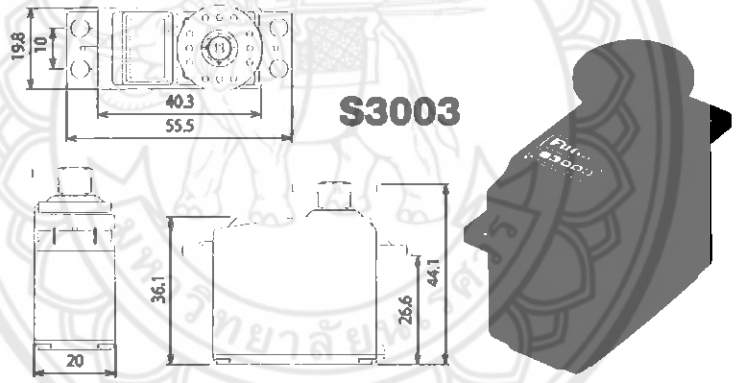
การทำงานเบื้องต้นของแขนกลอัตโนมัติ สามารถอธิบายได้ดังรูป 3.1 เริ่มต้นเมื่อมีการเปิดพอร์ตและเลือกการเชื่อมต่อกล้องเว็บแคม 2 ตัว ทำการเลือกวัตถุสีที่จะจับกล้องตัวที่ 1 จะจับวัตถุสีที่เลือกจากนั้นจึงนำมาประมวลผลภาพคำนวณหามุมกับความยาวและกล้องตัวที่ 2 จะคำนวณความสูงและส่งค่าให้กับไมโครคอนโทรลเลอร์ผ่านพอร์ตอนุกรมเพื่อควบคุมแขนกลหยิบจับวัตถุแล้วนำไปวางไว้ในตำแหน่งที่กำหนดเอาไว้

3.2 การพัฒนาโครงสร้างของแขนกลอัตโนมัติ

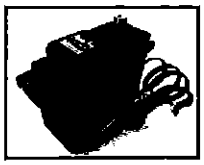
3.2.1 ส่วนประกอบของแขนกลอันโนมัติ

การพัฒนาโครงสร้างของแขนกล ได้มีการใช้วัสดุดังนี้

- เซอร์โวมอเตอร์ Futaba S3003 1 ตัว
- เซอร์โวมอเตอร์ Tower Pro MG996R 2 ตัว
- เซอร์โวมอเตอร์ Tower Pro 9805MG 1 ตัว
- เซอร์โวมอเตอร์ GWS S03T 1 ตัว
- กริปเปอร์ 1 ชุด
- แผ่นอะคริลิคขนาดความหนา 3 มิลลิเมตร
- น็อตตัวผู้
- น็อตตัวเมีย



รูปที่ 3.2 รูปร่างและขนาดของเซอร์โวมอเตอร์ Futaba S3003 [10]



9805MG

9805MG servo, Digital , Double ball bearing,metal gear set

Weight: 180g
Dimension: 67.9*30.2*56mm
Stall torque: 22kg/cm(4.8V)/25kg/cm(6V)
Operating speed: 0.20sec/60degree(4.8v), 0.16sec/60degree(6v)
Operating voltage: 4.8-6V
Dead band width: 5us

รูปที่ 3.3 รูปร่างและขนาดของเซอร์โวมอเตอร์ Tower Pro 9805MG [11]

MG996R

MG996R weight- 55.0g

MG996R weight- 55.0g
Dimension 40.7*19.7*42.9mm
Stall torque 9.4kg/cm(4.8V),11kg/cm(6V)
Operating speed 0.17sec/60degree(4.8v), 0.14sec/60degree(6v)
Operating voltage 4.8-7.2V
Temperature range 0°C_ 55°C
Dead band width 5us

รูปที่ 3.4 รูปร่างและขนาดของเซอร์โวมอเตอร์ Tower Pro MG996R [11]

สำหรับการจัดทำโครงสร้างของแขนกลจะใช้แผ่นอะครีลิคในการทำเป็นลำตัวและส่วนแขนของแขนกลอัตโนมัติโดยใช้แผ่นอะครีลิคที่ได้ทำการตัดให้เป็นไปตามรูปที่ได้ทำการออกแบบเอาไว้แล้วจึงทำการประกอบโครงสร้างส่วนต่างๆเข้าด้วยกันโดยใช้น้ำยาประสานอะครีลิคและน็อตเป็นตัวยึด ทำการติดมอเตอร์เข้ากับส่วนต่าง ๆ ของแขนกล

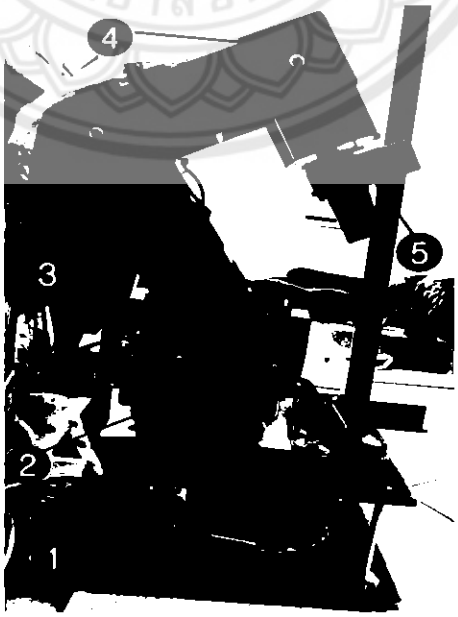
จุดที่ 1 คือลำตัวของหุ่นยนต์แขนกลใช้เซอร์โว GWS มอเตอร์ในการขับเคลื่อน

จุดที่ 2 คือข้อต่อส่วนหัวไหล่ใช้เซอร์โวมอเตอร์ Tower Pro 9805MG ในการขับเคลื่อน

จุดที่ 3 คือข้อต่อส่วนข้อศอก

จุดที่ 4 คือข้อต่อส่วนข้อมือจะใช้เซอร์โวมอเตอร์ Tower Pro MG996R ในการขับเคลื่อน

จุดที่ 5 คือส่วนมือจับจะใช้กริปเปอร์ทำงานร่วมกับเซอร์โวมอเตอร์ Futaba S3003 ในการขับเคลื่อน



รูปที่ 3.5 แสดงตำแหน่งการวางอุปกรณ์มอเตอร์ขับเคลื่อนของแขนกล

สำหรับส่วนประกอบต่างๆของแขนกลมีขนาดดังต่อไปนี้

รูปที่ 3.6 ส่วนลำตัวของแขนกล

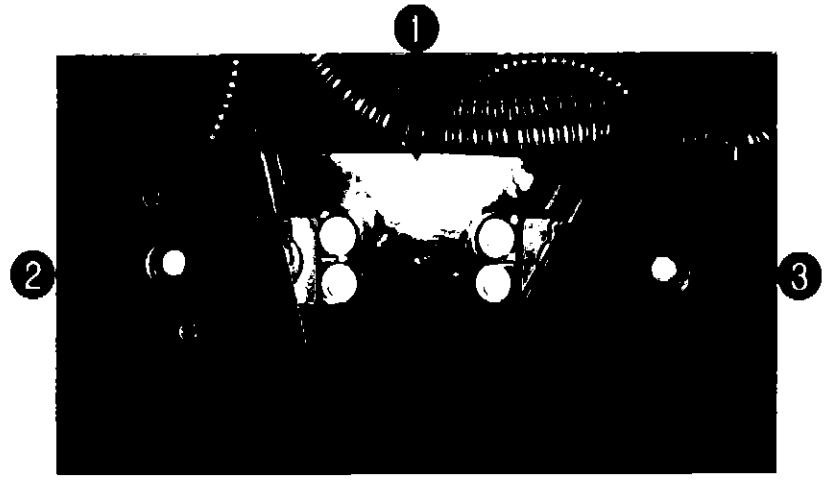
จากรูปที่ 3.6 ส่วนลำตัวของแขนกลใช้เซอร์โวมอเตอร์ในการขับเคลื่อน จุดที่ 1 และ จุดที่ 2 มีขนาดกว้าง 15.2 เซนติเมตร ยาว 15.2 เซนติเมตร

รูปที่ 3.7 ท่อนแขนส่วนที่ต่อจากหัวไหล่

จุดที่ 1 คือท่อนแขนส่วนที่ต่อจากหัวไหล่มีขนาด กว้าง 16.9 เซนติเมตร ยาว 20.1 เซนติเมตร สูง 14.4 เซนติเมตร

จุดที่ 2 คือ ฐานที่ใช้ยึดเซอร์โวมอเตอร์และใช้ยึดให้ท่อนแขนติดกับข้อต่อส่วนหัวไหล่มีขนาด ยาว 18.8 เซนติเมตร สูง 7 เซนติเมตร

จุดที่ 3 คือฐานที่ใช้ยึดข้อต่อส่วนหัวไหล่ให้ติดกับลำตัวมีขนาด กว้าง 17.6 เซนติเมตร ยาว 15.2 เซนติเมตร



รูปที่ 3.8 ท่อนแขนส่วนที่ต่อจากข้อศอก

จุดที่ 1 คือท่อนแขนส่วนที่ต่อจากข้อศอกมีขนาด กว้าง 4.7 เซนติเมตร ยาว 15.3 เซนติเมตร สูง 4.5 เซนติเมตร

จุดที่ 2 คือส่วนที่ใช้ยึดท่อนแขนให้ติดกับเซอร์โวมอเตอร์ส่วนข้อต่อข้อมือ

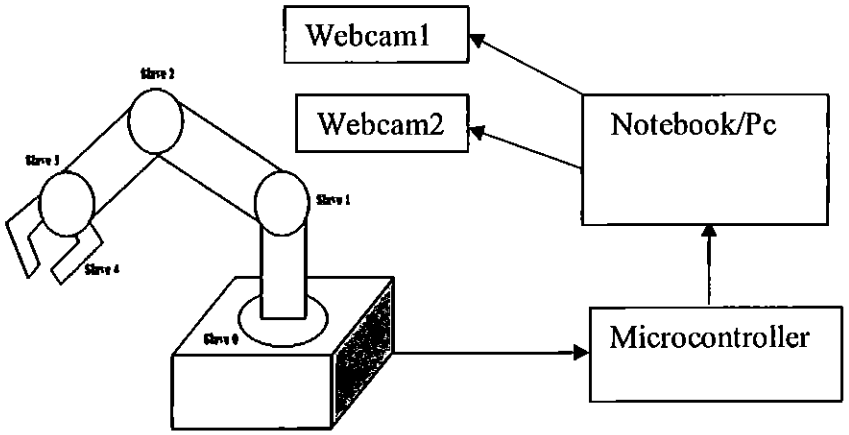
จุดที่ 3 คือส่วนที่ใช้ยึดท่อนแขนให้ติดกับเซอร์โวมอเตอร์ส่วนข้อต่อข้อศอก

รูปที่ 3.9 ท่อนแขนส่วนที่ต่อจากข้อมือ

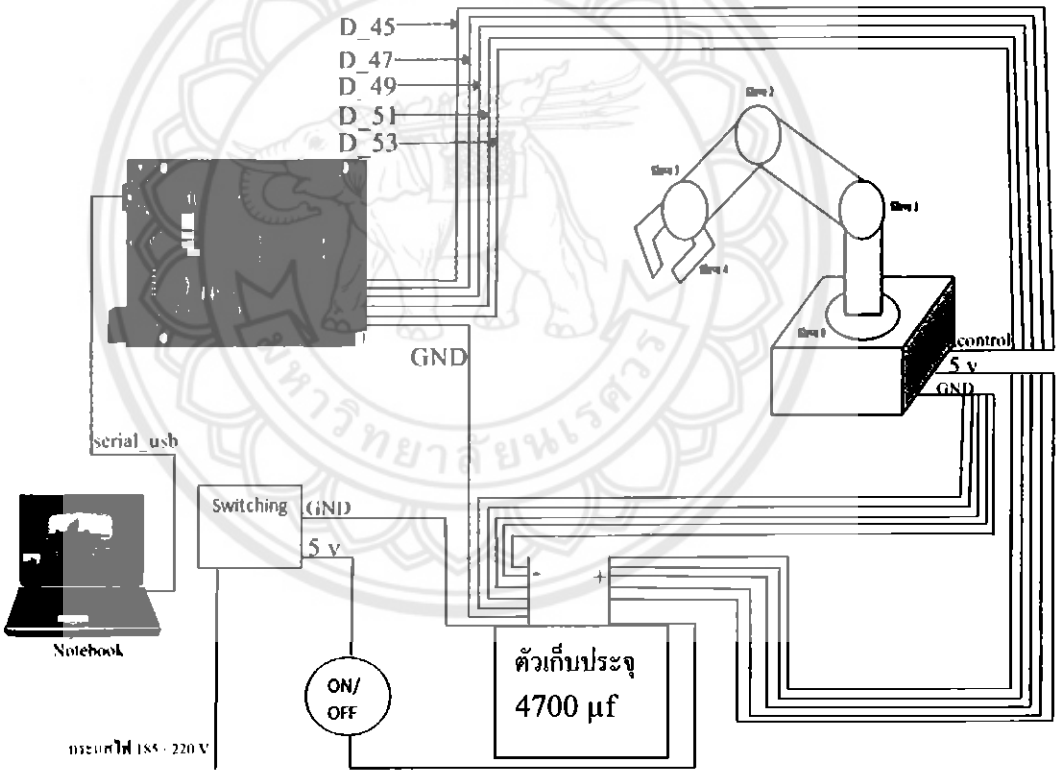
จุดที่ 1 คือท่อนแขนส่วนที่ต่อจากข้อมือมีขนาด กว้าง7.2 เซนติเมตร ยาว 8 เซนติเมตร สูง 4.5 เซนติเมตร

จุดที่ 2 คือส่วนที่ใช้ยึดท่อนแขนให้ติดกับมือจับ

จุดที่ 3 คือส่วนที่ใช้ยึดท่อนแขนให้ติดกับเซอร์โวมอเตอร์ส่วนข้อต่อข้อมือ

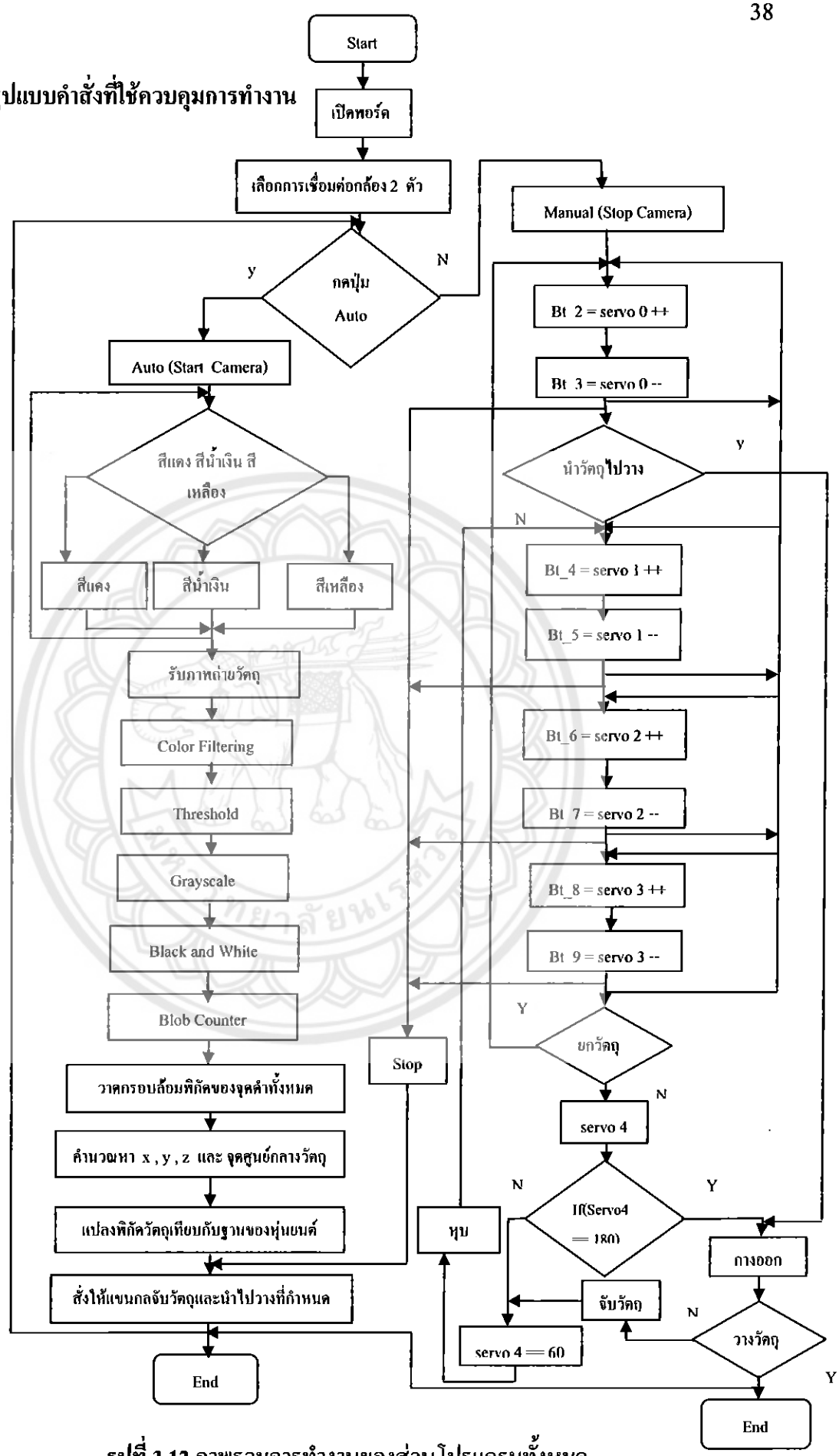


รูปที่ 3.10 ภาพรวมของการใช้บอร์ดไมโครคอนโทรลเลอร์ควบคุมแขนกล

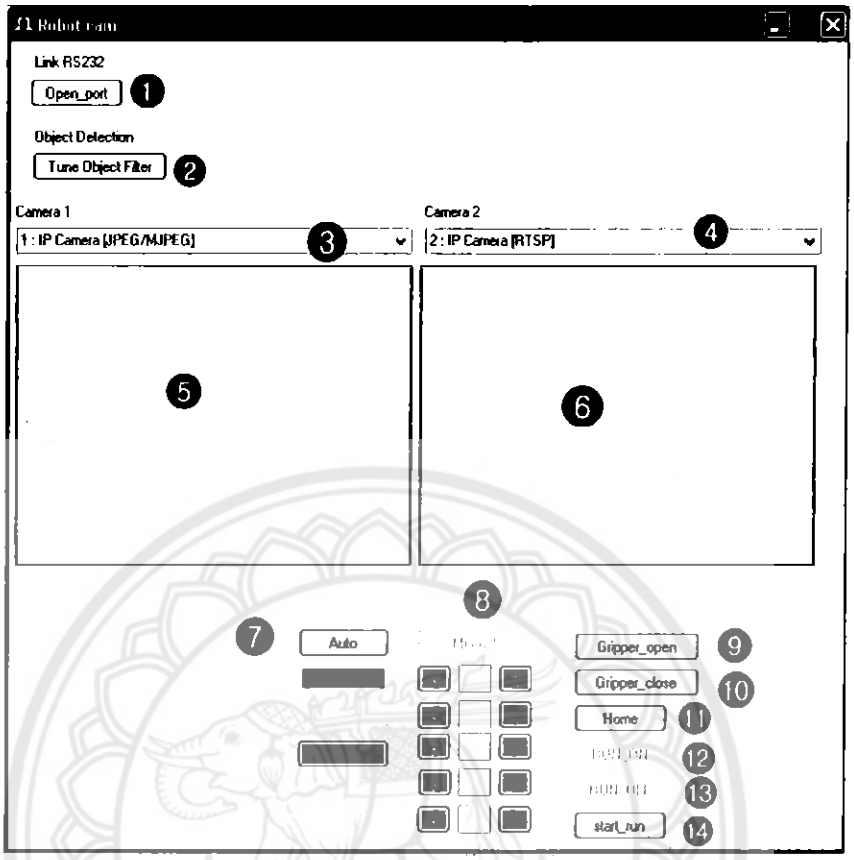


รูปที่ 3.11 ภาพรวมแสดงการเชื่อมต่อทั้งหมดของแขนกลอัตโนมัติ

3.2.2 รูปแบบคำสั่งที่ใช้ควบคุมการทำงาน



รูปที่ 3.12 ภาพรวมการทำงานของส่วนโปรแกรมทั้งหมด



รูปที่ 3.13 หน้าตาโปรแกรมส่วนของคอมพิวเตอร์

อธิบายการทำงานของโปรแกรมส่วนต่างๆได้ดังนี้

ส่วนที่ 1 เป็นส่วนที่ทำหน้าที่แสดงสถานะของการเปิดพอร์ต RS323

ส่วนที่ 2 เป็นส่วนที่ใช้สำหรับเซตค่าสีหรือ การผสมสี (Color)

ส่วนที่ 3 , 4 เป็นการเลือกการเชื่อมต่อกล้อง webcam 2 ตัวผ่านพอร์ต usb

ส่วนที่ 5 , 6 Video Displays เป็นส่วนแสดงภาพการทำงานของแขนกลที่ได้รับมาจากกล้องเว็บแคม ซึ่งใช้ในการจับภาพวัตถุโดยจะเริ่มแสดงภาพเมื่อกดปุ่ม Auto และ จะหยุดการแสดงภาพเมื่อปิดโปรแกรมลง

ส่วนที่ 7 Auto เมื่อกดจะแสดงการเชื่อมต่อกล้องเว็บแคม แล้วทำการเลือกสี แดง สีเหลือง สีน้ำเงิน

ส่วนที่ 8 Munnal เป็นการเซตค่าเซอร์โวมอเตอร์ทั้ง 5 ตัว โดยจะมีปุ่ม button 2 ถึง button 11 ในควบคุมให้เซอร์โวมอเตอร์ของแต่ละข้อต่อต่างๆ ซึ่งในส่วนนี้ก็สามารถควบคุมให้จับวัตถุได้ด้วยแต่จะเป็นกดควบคุมไม่ใช่ Auto

ส่วนที่ 9 Gipper open เป็นส่วนที่สั่งให้ gipper กางออก

ส่วนที่ 10 Gipper close เป็นการสั่งให้ gipper หุบหรือจับวัตถุ

ส่วนที่ 11 Home จะมีการตั้งค่าส่วนต่างๆ ของตัวแขนกลให้อยู่ในรูปแบบพร้อมที่จะไปจับวัตถุ

ส่วนที่ 12 Run_on เป็นปุ่มที่เมื่อทำการเลือกสีแล้วก็ทำการกดปุ่ม Run_on ตัวแขนกลก็จะหมุนไปตามวัตถุสีที่เราเลือกเอาไว้

ส่วนที่ 13 Run_off เมื่อกดปุ่มตัวแขนกลก็จะยกเลิกการหมุนตามวัตถุเพื่อรอรับคำสั่งใหม่จากผู้ควบคุมแขนกลอีกทีเพื่อให้แขนกลทำงานต่อไป

ส่วนที่ 14 Start_run เป็นปุ่มเมื่อกด Run_on แขนกลจะหมุนตามสีที่เลือกแล้วก็กด Start_run ตัวแขนกลก็จะไปหยิบวัตถุที่เป็นสีที่เลือกเอาไว้แล้วเอาไปวางไว้ในตำแหน่งที่กำหนด

3.3 กล้องเว็บแคม (Web Camera)

ในการเขียนโปรแกรมแสดงภาพการทำงานของหุ่นยนต์แขนกลอัตโนมัติจากกล้องเว็บแคม ซึ่งจะใช้ควบคุมในวิชวลสตูดิโอ c# 2008 ในการติดต่อระหว่างคอมพิวเตอร์กับกล้องเว็บแคม โดยภาพที่ได้จะถูกส่งไปประมวลผลภาพเพื่อให้ตัวแขนกลอัตโนมัติจับวัตถุตามที่ระบุเป็นสีต่างๆ กล้องเว็บแคมที่ใช้แสดงในรูปที่ 3.14

รูปที่ 3.14 กล้องเว็บแคม

รายละเอียดของ webcam Oker 177

- กล้องเว็บแคม Webcam Oker OE-177
- ความละเอียด 16 ล้าน pixels
- Resolution : 2.0 M pixels up to 16.0M by software
- ใช้เลนส์แก้วคุณภาพดี ให้ภาพคมชัดสูงมากระดับ Hi – End

- Frame rate : 60 fps แสดงผลภาพที่ไวมาก ไม่กระตุก
- มีระบบปรับความสมดุลของภาพเพื่อให้ได้สีเป็นธรรมชาติสมจริง
- มีไมโครโฟนในตัว (USB Microphone)
- Body ทำด้วย โลหะทนทานดูทันสมัย
- หมุนปรับโฟกัสที่หน้าเลนส์ได้
- สามารถปรับขาให้ใช้ได้กับจอ หรือตั้งวางได้
- ปรับแสงเองอัตโนมัติ ลดปัญหาเรื่องแสงไม่เพียงพอ และย้อนแสง
- มี Software สำหรับใส่ Effect ต่างๆ
- สามารถอัดวีดีโอ และถ่ายภาพนิ่งได้
- สามารถนำไปทำเป็นกล้องวงจรปิด CCTV ได้ดีมาก
- สามารถติดตั้งข้างานกับโน๊ตบุ๊ค หรือคอมพิวเตอร์ที่มีกล้องอยู่แล้วได้
- รองรับโปรแกรมแชททุกโปรแกรม
- Info noise rate : 48 dB
- Focus range : 30 mm – infinitive
- Plug and Play (ใช้งานได้ทันที่โดยไม่ต้องลงไดรเวอร์)
- Window NT/2000/XP/Vista/Windows 7(32/64 bit)/Mac OS X

จากการพัฒนาระบบควบคุมหุ่นยนต์แขนกลอัตโนมัติโดยการประมวลผลภาพ การสร้างหุ่นยนต์การเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์และส่วนประกอบต่างๆเข้าด้วยกันเพื่อใช้ในการควบคุมในส่วนข้อต่อต่างๆของแขนกล ในบทต่อไปจะกล่าวถึงการทำงาน และผลที่ได้จากการควบคุมการทำงานของแขนกล

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบการทำงานของโปรแกรมและแขนกลอัตโนมัติที่พัฒนาต่อมาว่าสามารถทำการเชื่อมต่อกับคอมพิวเตอร์โดยผ่านพอร์ต usb และสามารถสั่งการทำงานไปควบคุมตัวแขนกลอัตโนมัติที่ต่อกับเครื่องคอมพิวเตอร์ได้หรือไม่ จากนั้นจึงทำการทดสอบการจับวัตถุที่เป็นสีแดงสีน้ำเงินสีเหลืองเพื่อหาข้อผิดพลาดของโปรแกรมและแขนกลอัตโนมัติที่พัฒนาขึ้น เพื่อนำข้อผิดพลาดมาทำการวิเคราะห์ถึงปัญหา การปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น และแนวทางในการพัฒนาต่อไป

4.1 จุดประสงค์ของการทดลอง

4.1.1 เพื่อทดสอบโปรแกรมที่พัฒนาขึ้นมาว่าสามารถทำงาน ได้หรือไม่

4.1.2 เพื่อทดสอบการสั่งงานจากเครื่องคอมพิวเตอร์ว่ากล้องประมวลผลภาพและตัวแขนกลอัตโนมัติจะสามารถจับวัตถุเป็นไปตามคำสั่งที่เขียนโปรแกรมหรือไม่

4.1.3 เพื่อทดสอบการเคลื่อนที่ของแขนกลในส่วนข้อต่อต่างๆว่าสามารถทำงานได้ตามคำสั่งของผู้ใช้ได้หรือไม่

4.1.4 เพื่อทดสอบการจับวัตถุ สีแดง สีน้ำเงิน สีเหลือง ของแขนกลอัตโนมัติแล้วไปวางในที่ที่กำหนดตามที่ต้องการของผู้ใช้ได้หรือไม่

4.1.5 เพื่อทดสอบหาข้อผิดพลาดที่เกิดขึ้นกับโปรแกรมที่พัฒนาขึ้นและแขนกล พร้อมทั้งนำข้อผิดพลาดมาทำการวิเคราะห์ถึงปัญหา การปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น และแนวทางในการพัฒนาต่อไป

4.2 ขั้นตอนการทดลอง

4.2.1 จัดเตรียมการทดลองโดยนำโปรแกรมที่พัฒนาไปติดตั้งที่เครื่องคอมพิวเตอร์

4.2.2 นำอุปกรณ์ที่ใช้ในการทดสอบมาต่อเข้ากับเครื่องคอมพิวเตอร์ ดังนี้

4.2.2.1 นำกล้องเว็บแคม (Webcam) มาต่อเข้ากับคอมพิวเตอร์

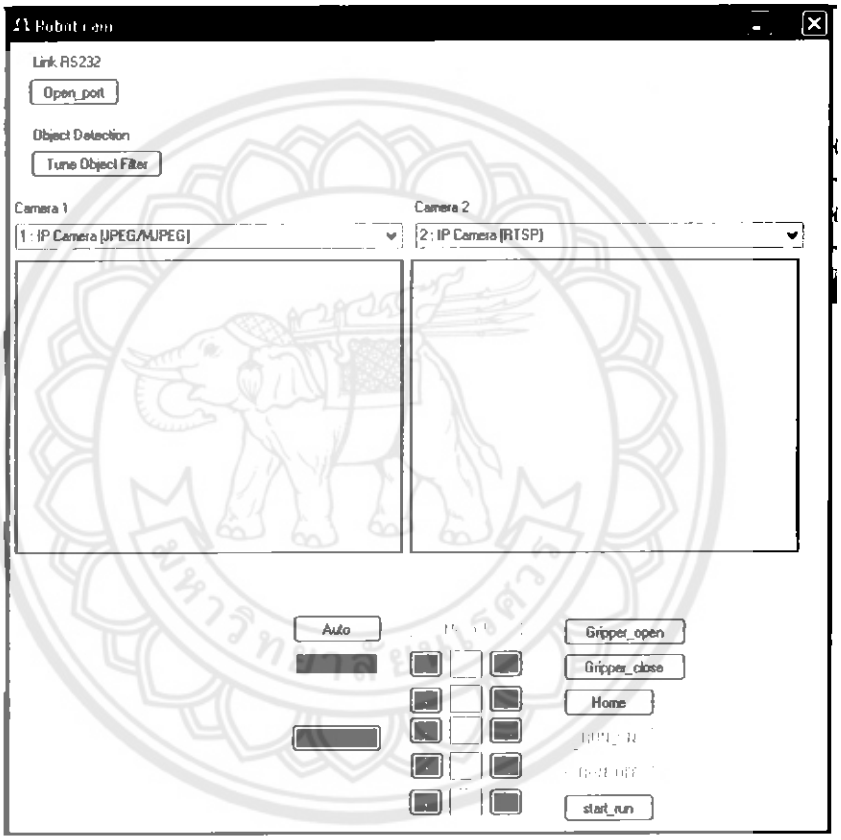
4.2.2.2 นำแขนกลอัตโนมัติมาต่อเข้ากับเครื่องคอมพิวเตอร์โดยเชื่อมต่อผ่านพอร์ต usb

4.2.3 ทำการเชื่อมต่อกระแสไฟโดยใช้สวิชชิ่ง เป็นตัวแปลงกระแสไฟจาก 220 V มาเป็นกระแสไฟ 5 v เข้ากับตัวไมโครคอนโทรลเลอร์มีสวิชเปิดปิดกระแสไฟ ทดลองส่งคำสั่งไปควบคุมการทำงานและสังเกตผลการทำงานที่ได้

4.2.4 ตรวจสอบข้อผิดพลาดที่เกิดขึ้นกับโปรแกรมที่พัฒนาขึ้นและแขนกลอัตโนมัติขณะทำการทดสอบ จากนั้นนำข้อผิดพลาดมาทำการวิเคราะห์ถึงปัญหา การปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น และแนวทางในการพัฒนาต่อไป

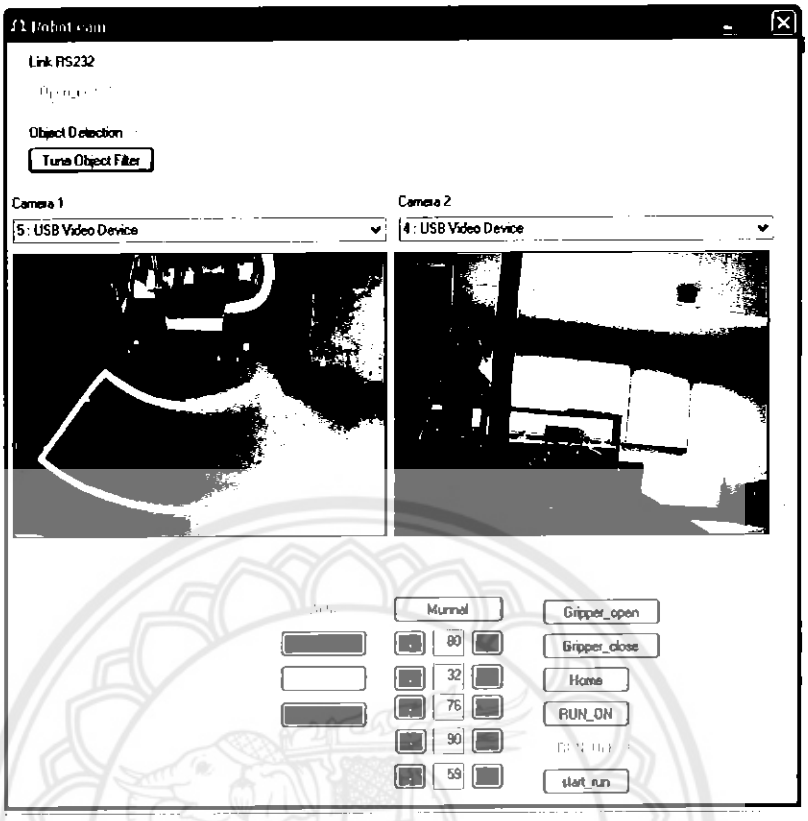
4.3 การทดสอบโปรแกรมที่พัฒนาขึ้นมา

เมื่อทำการเปิดโปรแกรมขึ้นมาโปรแกรมจะแสดง ดังในรูปที่ 4.1



รูปที่ 4.1 โปรแกรมที่เครื่องคอมพิวเตอร์

เมื่อทำการเปิดพอร์ตอนุกรมและเลือกการเชื่อมต่อกล้อง 2 ตัว เพื่อเชื่อมต่อเครื่องคอมพิวเตอร์กับแขนกลอัตโนมัติแล้วก็กดปุ่ม Auto ซึ่ง camera 1 จะเป็นกล้องที่มองภาพจากด้านบนลงมาและใช้เป็นตัวจับวัตถุสีต่างๆ ส่วน camera 2 จะมองภาพจากด้านข้างของตัวแขนกลอัตโนมัติ ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 โปรแกรมที่เครื่องคอมพิวเตอร์เมื่อกล้องเว็บแคมเริ่มจับภาพ

4.4 การทดสอบการทำงานของแขนกลที่พัฒนาขึ้น

4.4.1 ส่วนประกอบต่างๆของตัวแขนกลอัตโนมัติ

การทดสอบจับเวลาการเคลื่อนที่ของแขนกลอัตโนมัติเพื่อหาข้อผิดพลาดและนำเวลาที่ได้จากการทดลองมาหาค่าเฉลี่ยของการทำงานของแขนกลอัตโนมัติที่สามารถทำได้ โดยได้ทำการกำหนดพื้นที่ที่ตัวแขนกลอัตโนมัติจะสามารถหยิบจับวัตถุที่เป็นสีแดง สีน้ำเงิน และ สีเหลือง

จุดที่ 1 เป็นความกว้างของพื้นไม้อัดกว้าง 60.5 เซนติเมตร

จุดที่ 2 เป็นความยาวของพื้นไม้อัดยาว 40 เซนติเมตร

จุดที่ 3 คืออลูมิเนียมใช้เป็นความสูงของขาตั้งกล้องตัวบนสูง 40 เซนติเมตร

จุดที่ 4 คือกล้องตัวบนที่ใช้เพื่อจับภาพวัตถุและอลูมิเนียมที่ใช้ยึดกล้องตัวบนให้อยู่ตรงกลางของพื้นไม้อัดยาวเท่ากับ 60.5 เซนติเมตร

จุดที่ 5 คือกล้องตัวด้านข้างเพื่อมองเห็นมุมภาพจากด้านข้าง

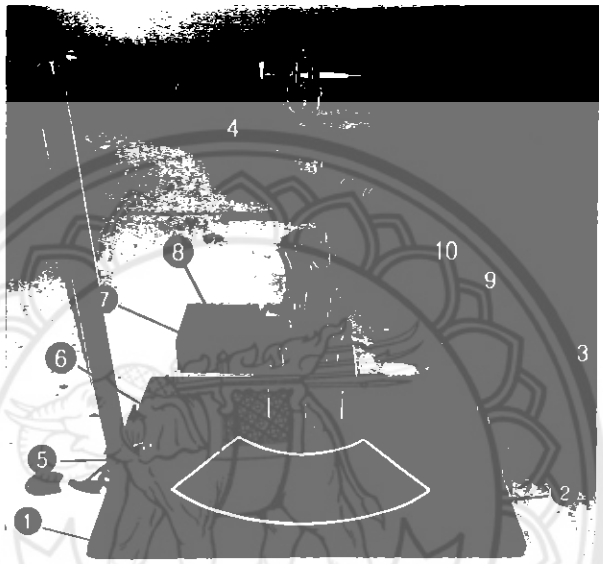
จุดที่ 6 คือพื้นที่ที่กำหนดให้วางวัตถุที่เป็นสีเพื่อให้แขนกลอัตโนมัติสามารถไปจับวัตถุได้มีขนาด มีขนาด 105 องศา

จุดที่ 7 คือตัวแขนกลอัตโนมัติ

จุดที่ 8 คือกล่องสีดำข้างในประกอบด้วยไมโครคอนโทลเลอร์และตัวเก็บประจุ 4700 μf

จุดที่ 9 คือตำแหน่งที่กำหนดให้แขนกลนำวัตถุที่เป็นสีไปวาง

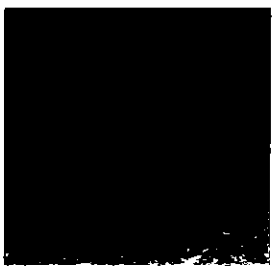
จุดที่ 10 คือ switching ที่แปลงกระแสไฟ 185 – 220 โวลต์ เป็นกระแสไฟ 5 โวลต์เพื่อจ่ายไฟไปเลี้ยงเซอร์โวทั้ง 5 ตัว ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 แสดงจุดอ้างอิงต่างๆของแขนกลอัตโนมัติ

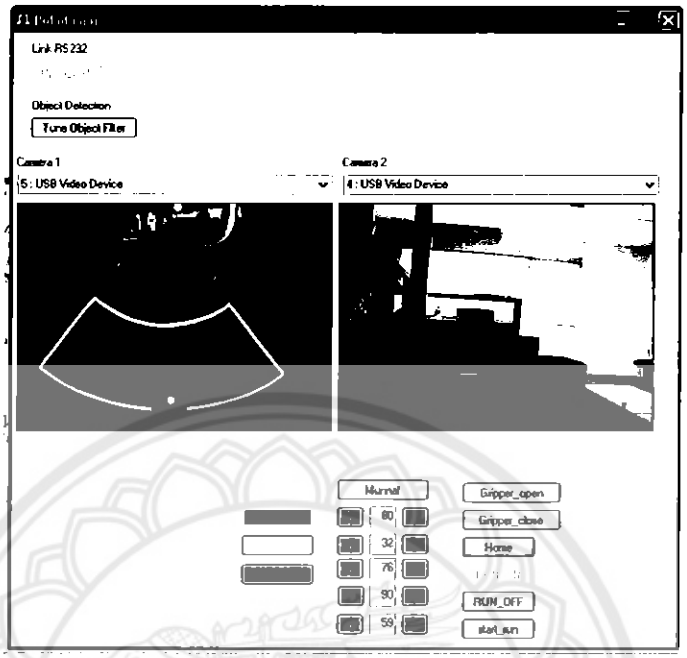
4.4.2 การทดสอบการหยิบจับวัตถุสีแดงที่วางหนึ่งชั้น

การทดสอบการหยิบวัตถุสี่เหลี่ยมผืนผ้าที่เป็นสีแดง ดังแสดงในรูปที่ 4.4 ซึ่งมีขนาดความกว้าง 4.5 เซนติเมตร ยาว 4 เซนติเมตร สูง 2.5 เซนติเมตร



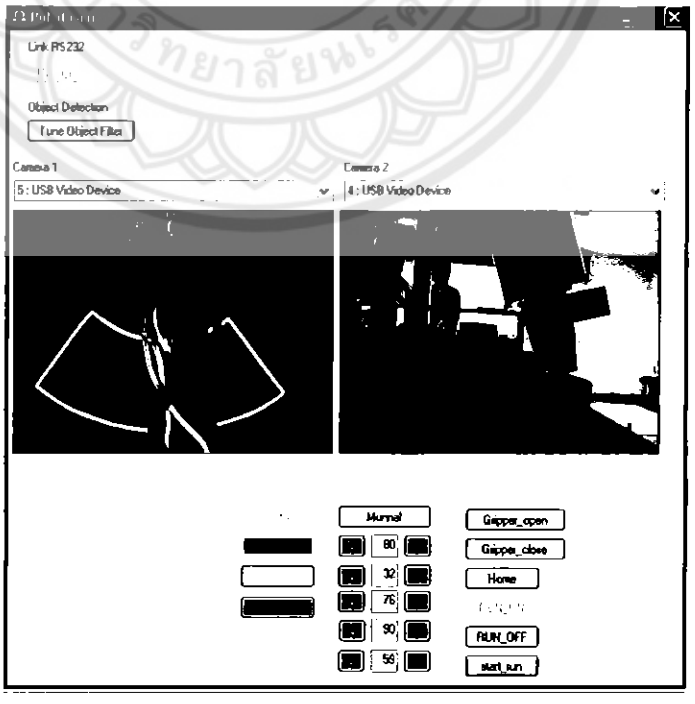
รูปที่ 4.4 วัตถุสี่เหลี่ยมผืนผ้าสีแดงที่ใช้ในการทดสอบ

จากรูปที่ 4.4 วัตถุทรงสี่เหลี่ยมพื้นผ้าซึ่งเป็นฟองน้ำเนื่องจากป้องกันความเสียหายอาจเกิดจากเซอร์โวมอเตอร์เฟืองเสีย



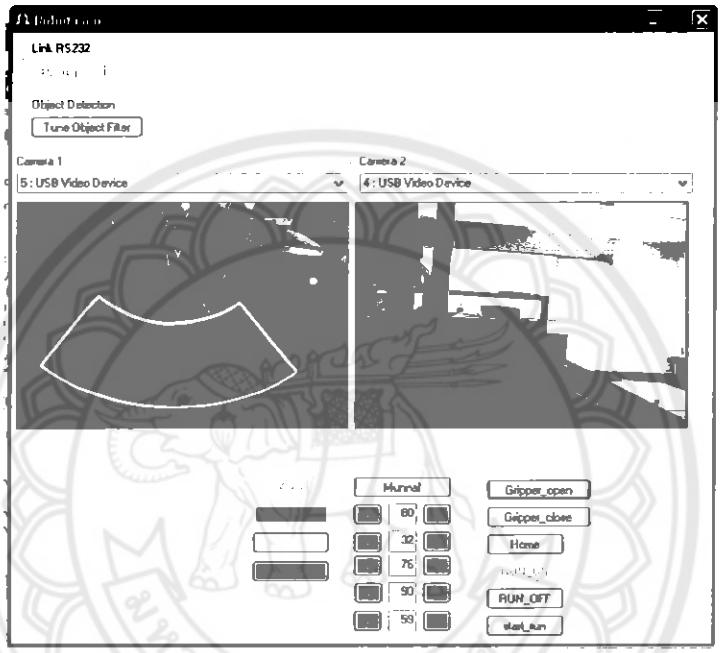
จากรูปที่ 4.5 โปรแกรมแสดงวัตถุสีแดงอยู่ตรงกลางหนึ่งชั้น

จากรูปที่ 4.5 จะเป็นการวางวัตถุหนึ่งชั้นอยู่ตรงกลางเพื่อรอคำสั่งให้แขนกลไปจับ



ตารางที่ 4.6 ผลการทดสอบหยิบจับวัตถุสีแดงตรงกลาง

จากรูปที่ 4.6 จะเห็นว่า เมื่อมีการกดที่สีแดงแล้วก็กด RUN_ON เพื่อให้ตัวแขนกลอัตโนมัติหมุนไปตามวัตถุที่เป็นสีแดงและก็กด Start_run แขนกลอัตโนมัติก็จะสามารถไปหยิบจับวัตถุที่เป็นสีแดงเองและนำไปวางไว้ที่กำหนดเอาไว้ ดังแสดงในรูปที่ 4.7 จากการทดสอบจับวัตถุสีแดงที่อยู่ตรงกลางหนึ่งชั้นทั้งหมด 20 ครั้งซึ่งค่าที่ได้จะบันทึกไว้ดังตารางที่ 4.1



รูปที่ 4.7 ภาพการนำวัตถุสีแดงไปวางไว้ตรงจุดที่กำหนด

ตารางที่ 4.1 ผลการทดสอบการหยิบจับวัตถุสีแดงตรงกลาง

ครั้ง	จับได้	จับไม่ได้	หมายเหตุ
1		/	-
2	/		ไม่วางวัตถุเอง
3	/		-
4	/		ไม่วางวัตถุเอง
5		/	-
6		/	-
7	/		-
8	/		ไม่วางวัตถุเอง

9		/	-
10	/		-
11	/		-
12	/		-
13		/	-
14		/	-
15	/		ไม่วางวัตถุเอง
16		/	-
17	/		-
18		/	-
19	/		ไม่วางวัตถุเอง
20	/		-

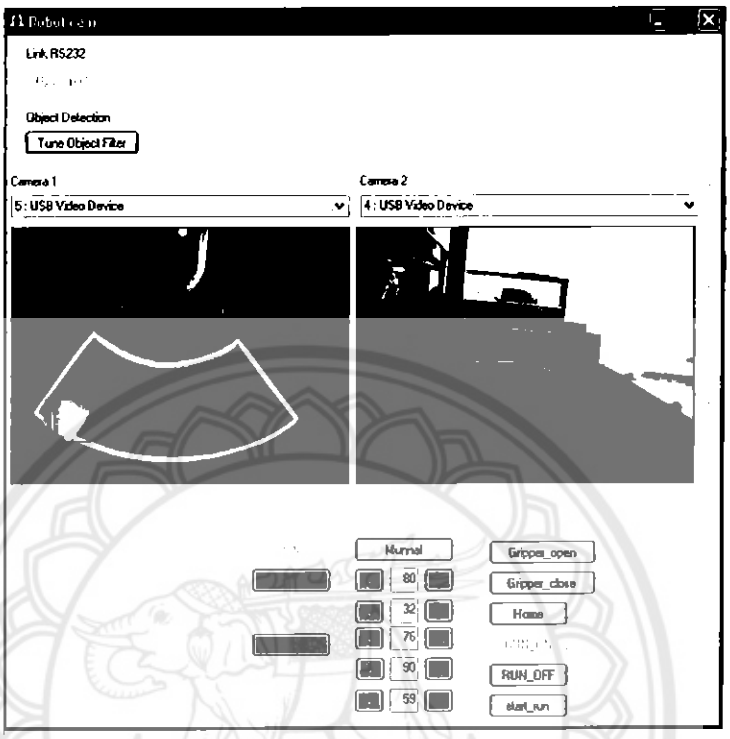
สรุปจากการทดลองตารางที่ 4.1 จากผลการทดสอบการจับวัตถุสีแดงที่อยู่ตรงกลางหนึ่งชั้นซึ่งได้ทำการทดสอบทั้งหมด 20 ครั้ง เฉลี่ยแล้วสามารถจับได้ร้อยละ 60

4.4.3 การทดสอบการหยิบจับวัตถุสีเหลืองที่วางหนึ่งชั้น

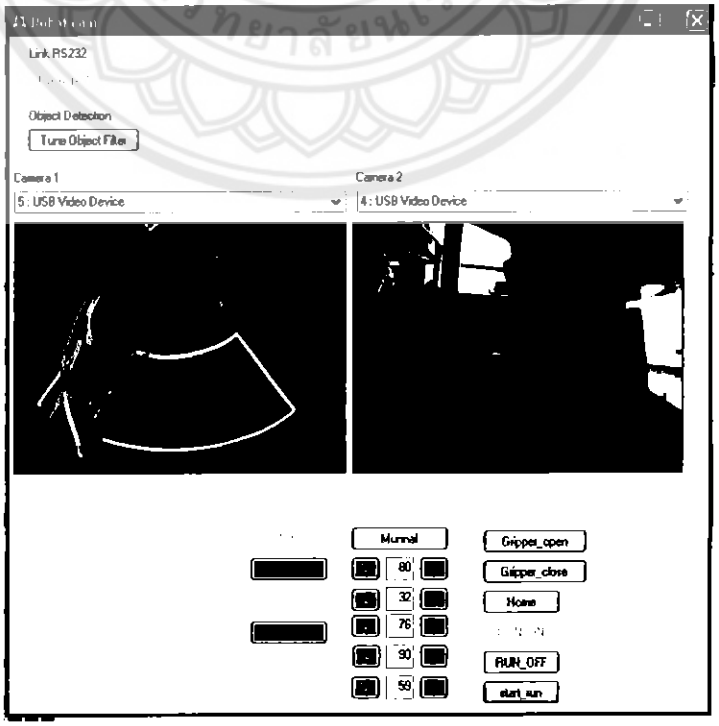
การทดสอบการหยิบวัตถุสี่เหลี่ยมผืนผ้าที่เป็นสีเหลือง ดังแสดงในรูปที่ 4.6 ซึ่งมีขนาดความกว้าง 4.5 เซ็นติเมตร ยาว 4 เซนติเมตร สูง 2.5 เซนติเมตร

รูปที่ 4.8 วัตถุสี่เหลี่ยมผืนผ้าสีเหลืองที่ใช้ในการทดสอบ

จากรูปที่ 4.6 วัตถุทรงสี่เหลี่ยมพื้นผ้าซึ่งเป็นฟองน้ำเนื่องจากป้องกันความเสียหาย อาจเกิดจากเซอร์โวมอเตอร์เฟืองเสีย

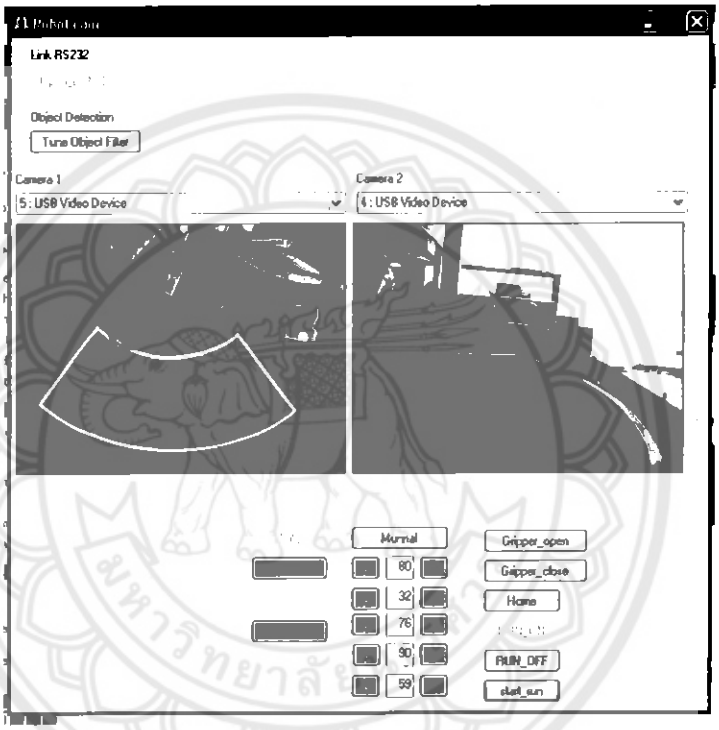


รูปที่ 4.9 โปรแกรมแสดงการวางวัตถุสีเหลืองซ้ายสุดหนึ่งชั้น



รูปที่ 4.10 โปรแกรมแสดงขณะทำการทดสอบหยิบจับวัตถุสีเหลืองซ้ายสุด

จากรูปที่ 4.8 จะเห็นว่า เมื่อมีการกดที่สีเหลืองแล้วก็กด RUN_ON เพื่อให้ตัวแขนกลอัตโนมัติหมุนไปตามวัตถุที่เป็นสีเหลืองและก็กด Start_run แขนกลอัตโนมัติก็จะสามารถไปหยิบจับวัตถุที่เป็นสีเหลืองเองและจะนำไปวางไว้ที่กำหนดเอาไว้ ดังแสดงในรูปที่ 4.9 จากการทดสอบจับวัตถุสีเหลืองที่ซ้ายสุดหนึ่งชั้นทั้งหมด 20 ครั้งซึ่งค่าที่ได้จะบันทึกไว้ดังตารางที่ 4.2

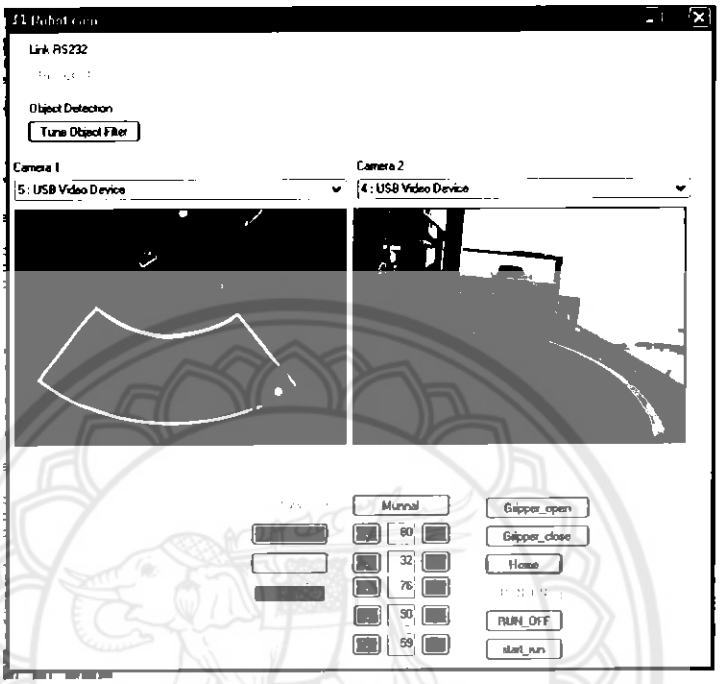


รูปที่ 4.11 ภาพการนำวัตถุสีเหลืองไปวางไว้จุดที่กำหนด

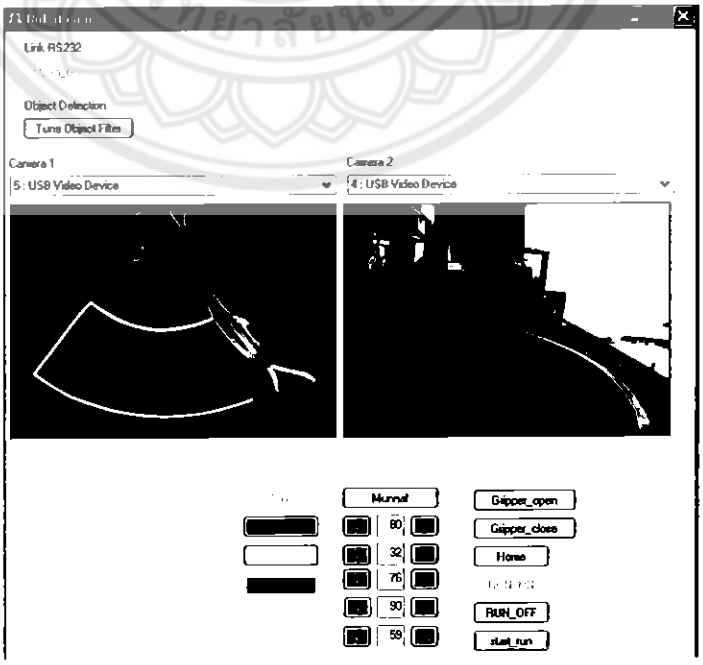
ตารางที่ 4.2 ผลการทดสอบการหยิบจับวัตถุสีเหลืองซ้ายสุด

ครั้ง	จับได้	จับไม่ได้	หมายเหตุ
1		/	-
2		/	-
3	/		-
4		/	-
5	/		-
6		/	-
7	/		-

จากรูปที่ 4.10 วัตถุทรงสี่เหลี่ยมพื้นผ้าซึ่งเป็นฟองน้ำเนื่องจากป้องกันความเสียหายอาจเกิดจากเซอร์โวมอเตอร์เฟืองเสีย



รูปที่ 4.13 โปรแกรมแสดงวางวัตถุสีน้ำเงินขวาสุดหนึ่งชั้น



รูปที่ 4.14 โปรแกรมแสดงขณะหยิบจับวัตถุสีน้ำเงินขาวสุด

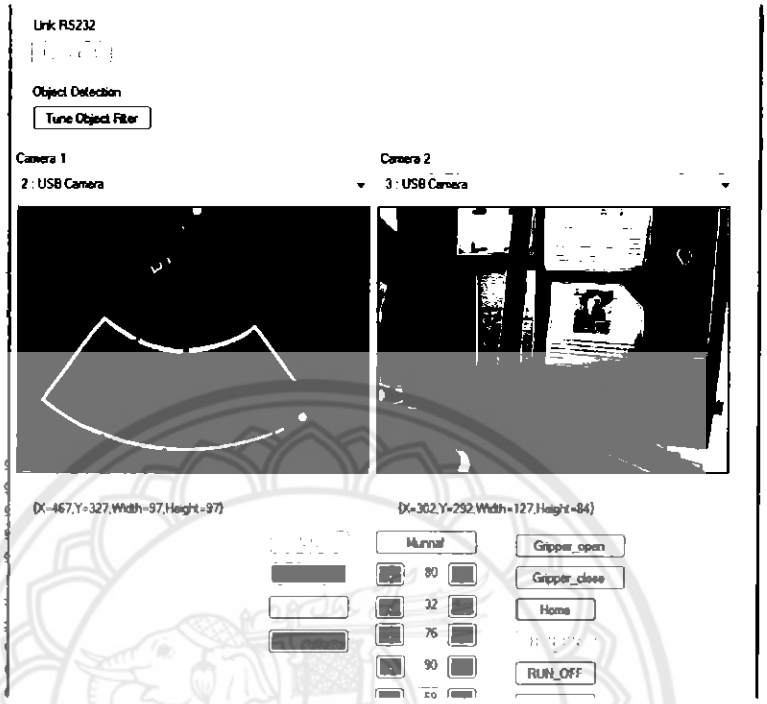
จากรูปที่ 4.12 จะเห็นว่า เมื่อมีการกดที่สีน้ำเงินแล้วก็กด RUN_ON เพื่อให้ตัวแขนกลอัตโนมัติหมุนไปตามวัตถุที่เป็นสีน้ำเงินและกด Start_run แขนกลอัตโนมัติก็จะสามารถไปหยิบวัตถุที่เป็นสีน้ำเงินเองและจะนำไปวางไว้ที่กำหนดเอาไว้ จากการทดสอบจับวัตถุสีน้ำเงินที่อยู่ขวาหนึ่งชั้นทั้งหมด 20 ครั้งซึ่งค่าที่ได้จะบันทึกไว้ดังตารางที่ 4.3

ตารางที่ 4.3 ผลการทดสอบการหยิบจับวัตถุสีน้ำเงินขวาสุด

ครั้ง	จับได้	จับไม่ได้	หมายเหตุ
1	/		ไม่วางวัตถุเอง
2	/		-
3	/		-
4		/	-
5	/		-
6	/		ไม่วางวัตถุเอง
7	/		-
8	/		-
9		/	-
10	/		-
11		/	-
12	/		-
13	/		-
14	/		-
15	/		-
16	/		ไม่วางวัตถุเอง
17		/	-
18		/	-
19	/		ไม่วางวัตถุเอง
20	/		-

สรุปจากการทดลองตารางที่ 4.3 จากผลการทดสอบการจับวัตถุสีน้ำเงินที่อยู่ขวาสุดหนึ่งชั้นซึ่งได้ทำการทดสอบทั้งหมด 20 ครั้ง เฉลี่ยแล้วสามารถจับได้ร้อยละ 75

4.4.5 การทดสอบหยิบจับวัตถุสีแดงที่วางเป็นสองชั้น



รูปที่ 4.15 โปรแกรมแสดงวัตถุสีแดงขวาสุดสองชั้น

จากรูปที่ 4.13 จะเห็นว่า เมื่อมีการกดที่สีแดงแล้วก็กด RUN_ON เพื่อให้ตัวแขนกลอัตโนมัติหมุนไปตามวัตถุที่เป็นสีแดงและกด Start_run แขนกลอัตโนมัติก็จะสามารถไปหยิบวัตถุที่เป็นสีแดงเองและจะนำไปวางไว้ที่กำหนดเอาไว้ จากการทดสอบจับวัตถุสีแดงที่อยู่ขวาสองชั้นทั้งหมด 20 ครั้งซึ่งค่าที่ได้จะบันทึกไว้ดังตารางที่ 4.4

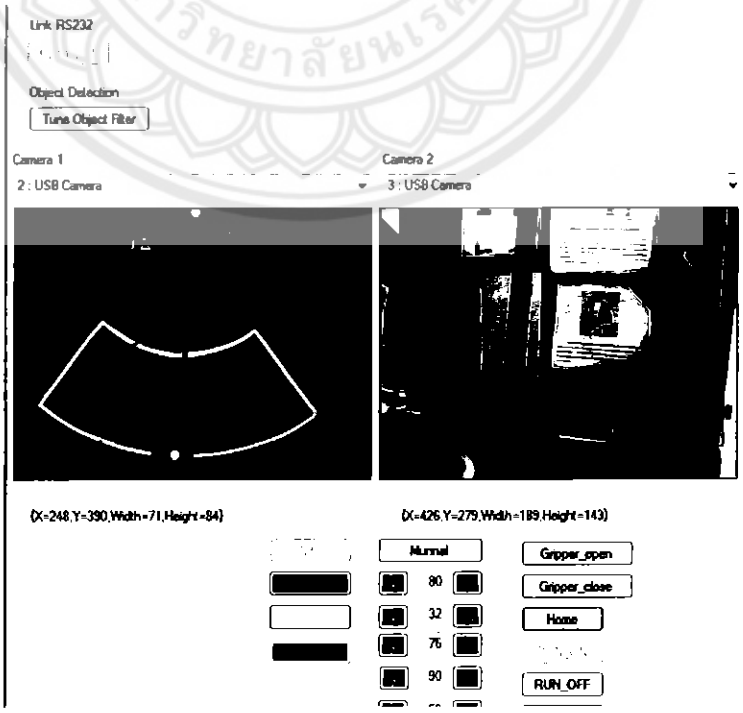
ตารางที่ 4.4 ผลการทดสอบการหยิบจับวัตถุสีแดงซ้ายสุด

ครั้ง	จับได้	จับไม่ได้	หมายเหตุ
1	/		-
2	/		ไม่วางวัตถุเอง
3	/		-
4	/		ไม่วางวัตถุเอง
5	/		-
6	/		-
7	/		ไม่วางวัตถุเอง
8	/		-

9	/		ไม่วางวัตถุเอง
10	/		-
11	/		-
12	/		-
13	/		-
14	/		ไม่วางวัตถุเอง
15		/	-
16	/		-
17		/	-
18	/		-
19	/		-
20	/		-

สรุปจากการทดลองตารางที่ 4.4 จากผลการทดสอบการจับวัตถุสีแดงที่อยู่ขวาสุดสองชั้นซึ่งได้ทำการทดสอบทั้งหมด 20 ครั้ง เฉลี่ยแล้วสามารถจับได้ร้อยละ 90

4.4.6 การทดสอบหยิบจับวัตถุสีน้ำเงินที่วางสองชั้น



รูปที่ 4.16 โปรแกรมแสดงวัตถุสีน้ำเงินตรงกลางสองชั้น

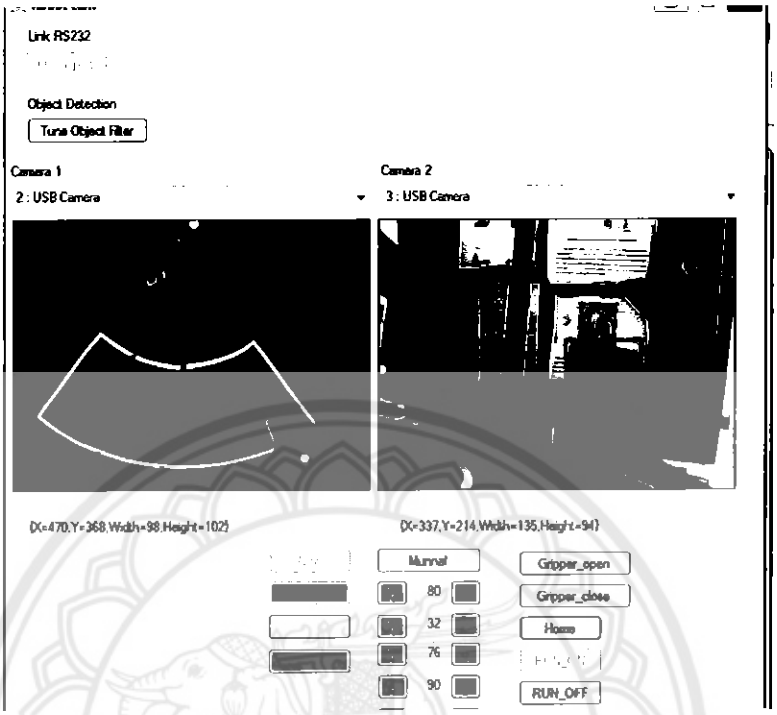
จากรูปที่ 4.14 จะเห็นว่า เมื่อมีการกดที่สีน้ำเงินแล้วก็กด RUN_ON เพื่อให้ตัวแขนกลอัตโนมัติหมุนไปตามวัตถุที่เป็นสีน้ำเงินและกด Start_run แขนกลอัตโนมัติก็จะสามารถไปหยิบวัตถุที่เป็นสีน้ำเงินเองและจะนำไปวางไว้ที่กำหนดเอาไว้ จากการทดสอบจับวัตถุสีน้ำเงินที่อยู่ขวาสองชั้นทั้งหมด 20 ครั้งซึ่งค่าที่ได้จะบันทึกไว้ดังตารางที่ 4.5

ตารางที่ 4.5 ผลการทดสอบการหยิบจับวัตถุสีน้ำเงินตรงกลางสองชั้น

ครั้ง	จับได้	จับไม่ได้	หมายเหตุ
1		/	-
2	/		-
3		/	-
4		/	-
5	/		-
6	/		ไม่วางวัตถุเอง
7		/	-
8		/	-
9	/		-
10		/	-
11		/	-
12		/	-
13	/		-
14	/		ไม่วางวัตถุเอง
15		/	-
16	/		-
17		/	-
18	/		-
19		/	-
20	/		-

สรุปจากการทดลองตารางที่ 4.5 จากผลการทดสอบการจับวัตถุสีน้ำเงินที่อยู่ตรงกลางสองชั้นซึ่งได้ทำการทดสอบทั้งหมด 20 ครั้ง เฉลี่ยแล้วสามารถจับได้ร้อยละ 45

4.4.7 การทดสอบหยิบวัตถุสีแดงเป็นสามชั้น



รูปที่ 4.17 โปรแกรมแสดงวัตถุสีแดงขวาสุดสามชั้น

จากรูปที่ 4.15 จะเห็นว่า เมื่อมีการกดที่สีแดงแล้วก็กด RUN_ON เพื่อให้ตัวแขนกลอัตโนมัติหมุนไปตามวัตถุที่เป็นสีแดงและกด Start_run แขนกลอัตโนมัติก็จะสามารถไปหยิบวัตถุที่เป็นสีแดงเองและจะนำไปวางไว้ที่กำหนดเอาไว้ จากการทดสอบจับวัตถุสีแดงที่วางเป็นสามชั้นทั้งหมด 20 ครั้งซึ่งค่าที่ได้จะบันทึกไว้ดังตารางที่ 4.6

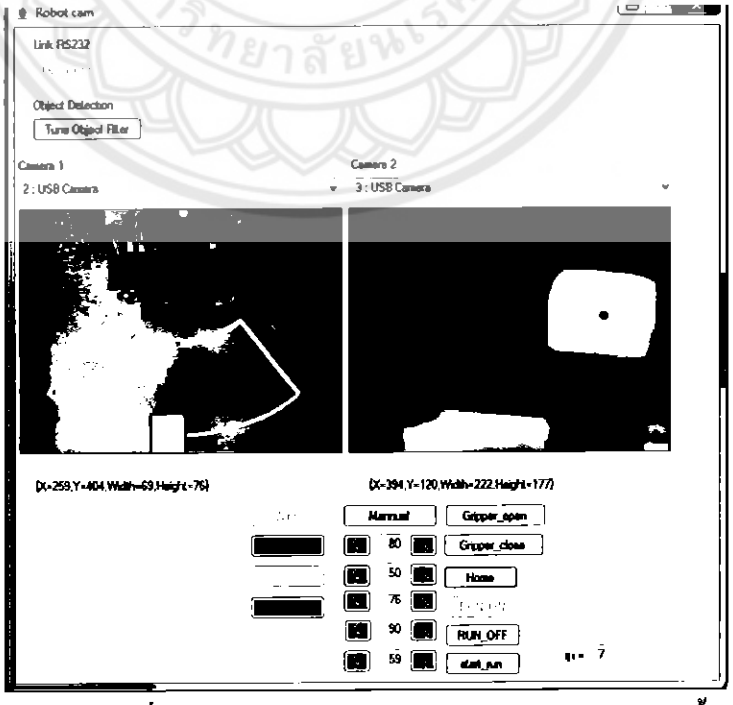
ตารางที่ 4.6 ผลการทดสอบการหยิบจับวัตถุสีแดงขวาสุดสามชั้น

ครั้ง	จับได้	จับไม่ได้	หมายเหตุ
1	/		-
2	/		ไม่วางวัตถุเอง
3		/	-
4	/		-
5	/		ไม่วางวัตถุเอง
6		/	-
7	/		-
8	/		-

9	/		-
10	/		-
11		/	-
12	/		ไม่วางวัตถุเอง
13	/		-
14		/	-
15	/		-
16	/		-
17	/		ไม่วางวัตถุเอง
18		/	-
19	/		-
20		/	-

สรุปจากการทดลองตารางที่ 4.6 จากผลการทดสอบการหยิบจับวัตถุสีแดงขวาสุดสามชั้นซึ่งได้ทำการทดสอบทั้งหมด 20 ครั้ง เฉลี่ยแล้วสามารถจับได้ร้อยละ 70

4.4.8 การทดสอบหยิบจับวัตถุสีเหลืองสามชั้นตรงกลาง



รูปที่ 4.18 โปรแกรมแสดงวัตถุสีเหลืองตรงกลางสามชั้น

จากรูปที่ 4.16 จะเห็นว่า เมื่อมีการกดที่สีเหลืองแล้วก็กด RUN_ON เพื่อให้ตัวแขนกลอัตโนมัติหมุนไปตามวัตถุที่เป็นสีเหลืองและกด Start_run แขนกลอัตโนมัติก็จะสามารถไปหยิบวัตถุที่เป็นสีเหลืองเองและจะนำไปวางไว้ที่กำหนดเอาไว้ จากการทดสอบจับวัตถุสีเหลืองที่วางเป็นสามชั้นทั้งหมด 20 ครั้งซึ่งค่าที่ได้จะบันทึกไว้ดังตารางที่ 4.7

ตารางที่ 4.7 ผลการทดสอบการหยิบจับวัตถุสีเหลืองตรงกลางสามชั้น

ครั้ง	จับได้	จับไม่ได้	หมายเหตุ
1		/	-
2		/	-
3	/		-
4		/	-
5	/		ไม่วางวัตถุเอง
6	/		-
7		/	-
8	/		-
9		/	-
10		/	-
11	/		ไม่วางวัตถุเอง
12		/	-
13	/		-
14		/	-
15		/	-
16	/		ไม่วางวัตถุเอง
17		/	-
18	/		-
19		/	-
20		/	-

สรุปจากการทดลองตารางที่ 4.7 จากผลการทดสอบการหยิบจับวัตถุสีเหลืองตรงกลางสามชั้นซึ่งได้ทำการทดสอบทั้งหมด 20 ครั้ง เฉลี่ยแล้วสามารถจับได้ร้อยละ 40

จากที่ได้ทำการทดลองระบบที่ได้พัฒนาขึ้นมาและเก็บข้อมูลต่างๆ บทต่อไปจะกล่าวถึงปัญหาที่พบ การแก้ปัญหา และแนวทางในการพัฒนาต่อไป

บทที่ 5
บทสรุปและวิเคราะห์โครงงาน

5.1 สรุปผลการทดลอง

จากการจัดทำโครงงานระบบควบคุมหุ่นยนต์แขนกลอัตโนมัติโดยการประมวลผลภาพ ผลที่ได้รับจากโครงงานนี้ก็คือ สามารถเขียนโปรแกรมที่ใช้ในการควบคุมแขนกลอัตโนมัติโดยการประมวลผลภาพได้ โดยที่โปรแกรมที่พัฒนาขึ้นนี้ ส่วนของแขนกลอัตโนมัติและส่วนของการประมวลผลภาพจากกล้องซึ่งโปรแกรมทั้งสองส่วนสามารถเชื่อมต่อกันเพื่อควบคุมแขนกลอัตโนมัติได้ จากผลการทดลองพบว่าถ้าไม่มีการควบคุมแสงสว่างให้พอดีหรือประมวลผลภาพแล้วคำนวณหาจุดตรงกลางวัตถุไม่ตรงจะทำให้แขนกลไม่สามารถจับวัตถุได้ ซึ่งจะมีการกำหนดเส้นทางล่วงหน้าไว้สำหรับให้แขนกลเดินตามเป็นเส้นตรงเข้าไปจับวัตถุและเมื่อแขนกลเข้าไปจับวัตถุที่วางอยู่ตรงกลางตัวแขนกลจะบังกล้องตัวด้านบนจะทำให้แขนกลไม่สามารถจับวัตถุได้และเมื่อวางวัตถุเป็นสามชั้นที่ตำแหน่งด้านซ้ายสุดกล้องตัวด้านข้างจะมองไม่เห็นวัตถุส่งผลให้กล้องไม่สามารถจับภาพเพื่อคำนวณค่าความสูงของวัตถุทำให้แขนกลไม่สามารถจับวัตถุได้ที่แขนกลสามารถจับวัตถุที่วางสามชั้นได้จะอยู่ระหว่างกลางไปทางขวาสุดบางครั้งจับวัตถุได้แต่ไม่วางวัตถุเองอาจเกิดจากโปรแกรมค้างและมีข้อจำกัดที่ว่าถ้าวางวัตถุเป็นสองชั้นหรือสามชั้นแขนจะหยิบจับวัตถุที่อยู่ชั้นบนสุดเท่านั้น และการส่งคำสั่งเพื่อไปควบคุมแขนกลอัตโนมัติขณะที่เข้าไปจับวัตถุทำให้แขนกลมีอาการกระตุกเนื่องจากโครงสร้างของตัวแขนกลไม่แข็งแรงพอหรือตัวเซอร์โวมอเตอร์ที่ส่วนหัวไหล่ไม่นิ่งพอ

จากการทดลองแขนกลจะสามารถจับวัตถุที่วางอยู่ขาวสุดโดยส่วนมากเนื่องจากกล้องสามารถจับภาพวัตถุได้ดีและเวลาแขนกลเข้าไปจับวัตถุตัวแขนกลจะไม่บังกล้องตัวด้านบนหรือกล้องตัวด้านข้างสามารถจับภาพวัตถุได้ดีจึงสามารถสรุปได้ว่าโปรแกรมที่พัฒนาขึ้นสามารถนำไปใช้ควบคุมการทำงานของแขนกลอัตโนมัติในการหยิบจับวัตถุที่เป็นสีแดง สีเหลือง สีน้ำเงิน ได้

5.2 ปัญหาและแนวทางในการแก้ไข

5.2.1 การเข้าไปจับวัตถุสีต่างๆ จะมีปัญหาเรื่อง แสงสว่างไม่เพียงพอ หรือ ถ้าแสงสว่างมากเกินไปก็จะทำให้กล้องมองเห็นวัตถุเป็นสีอื่น ซึ่งจะทำให้ตัวแขนกลไม่สามารถเข้าไปจับสีนั้นได้ ทางที่ดีควรที่จะใช้สีแบบ HSV เข้ามาช่วย

5.2.2 ระหว่างที่ตัวแขนกลอัตโนมัติเข้าไปจับวัตถุแขนกลจะมีอาการสั่นเนื่องจากโครงสร้างของแขนกลทำด้วยอะครีลิคทำให้ไม่แข็งแรงหรือตัวเซอร์โวไม่นิ่งพอ จึงควรออกแบบด้วยวัสดุที่แข็งแรงมากกว่านี้

5.2.3 ปัญหาของการผสมสีจะใช้การเลื่อนให้เจอค่าสีที่ต้องการและมีการกรอกค่าตัวเลขซึ่งเป็นการประมาณเอาทำให้ได้ค่าสีที่ไม่ค่อยตรงมาก ควรเขียนโปรแกรมโดยใช้คลิ๊กเม้าส์วัตถุที่เป็นสีหรือเขียนโปรแกรมโดยใช้ OpenCV เข้ามาช่วย

5.2.4 เซอร์โวมอเตอร์ที่ใช้มีความละเอียดไม่มากพอ จึงทำให้การควบคุมแขนกลไปยังตำแหน่งต่างๆทำได้ไม่ละเอียดมากนัก ควรเปลี่ยนเซอร์โวมอเตอร์ที่มีความละเอียดมากกว่านี้

5.2.5 ไม่สามารถควบคุมการหมุนเซอร์โวมอเตอร์ได้อย่างละเอียด เนื่องจากการออกแบบแขนกลที่ไม่ดี ควรออกแบบแขนกลใหม่

5.2.6 มีข้อผิดพลาดของโปรแกรมอยู่บ้างในการเซตค่าตำแหน่งของเซอร์โวมอเตอร์ให้จำค่าองศาไว้ล่วงหน้าในการเข้าไปจับวัตถุ ควรใช้หลักการของกลศาสตร์การเคลื่อนไหวไปข้างหน้าเข้ามาช่วยเพื่อความแม่นยำในการเข้าไปรับวัตถุ

5.2.7 เมื่อวางวัตถุเป็นสองชั้นและสามชั้น แขนกลจะสามารถหยิบจับวัตถุสีที่วางอยู่ชั้นบนสุดเท่านั้น

5.3 แนวทางในการพัฒนาต่อ

5.3.1 ในการควบคุมแขนกลอัตโนมัติสิ่งที่คำนึงถึง คือผลตอบสนองของเวลา (response time)และความแม่นยำในการจับวัตถุ หลังจากได้ทำการสั่งงานแขนกล ซึ่งต้องการระบบ Real time เพื่อให้ได้การทำงานที่ถูกต้องตามที่ต้องการทันที

5.3.2 เพิ่มความแข็งแรงของแขนกลโดยใช้วัสดุอื่นในการทำโครงสร้างของแขนกล

5.3.3 เพิ่มความละเอียดในการควบคุมแขนกลโดยใช้เซอร์โวมอเตอร์หรือสเต็ปมอเตอร์ที่มีความละเอียดมากขึ้นหรือใช้วิธีการเขียนโปรแกรมเพื่อควบคุมแขนกลแบบอื่นๆ

5.3.4 เพื่อการทำงานที่แม่นยำควรที่จะใช้หลักการของกลศาสตร์การเคลื่อนไหวไปข้างหน้าหรือ กลศาสตร์การเคลื่อนไหวผกผัน เข้ามาช่วยเพื่อให้แขนกลคำนวณและไปจับเองโดยไม่ต้องให้จดจำค่าไว้ล่วงหน้า

5.3.5 ควรเขียนโปรแกรมโดยใช้ OpenCV เพื่อแก้ปัญหาเรื่องแสงสว่าง

5.3.6 สามารถนำหลักการไปประยุกต์ในการพัฒนาระบบควบคุมแขนกลในการใช้งานจริงได้

เอกสารอ้างอิง

[1] ภควัต รักศรี. (6 ตุลาคม 2553). หุ่นยนต์. สืบค้นเมื่อ 24 มกราคม 2554, จาก http://www.coe.or.th/e_engineers/knc_detail.php?id=86.

[2] อรรถวิทย์ สุดแสง. (มิถุนายน 2547). แขนกล. สืบค้นเมื่อ 28 สิงหาคม 2553, จาก http://www.cp.eng.chula.ac.th/~attawith/class/mani.pdf.

[3] จารุต บุศราทิจ. (18 สิงหาคม 2552). บทความแนะนำ ATmega1280. ไมโครคอนโทรลเลอร์ตระกูล AVR. สืบค้นเมื่อ 15 ธันวาคม 2553, จาก http://www.etteam.com/product2009/ET-AVR/ET-EASY_MEGA1280.html

[4] ศิริพงษ์ รังแสง. (12 สิงหาคม 2553). การติดต่อสื่อสารผ่านพอร์ต RS-232. สืบค้นเมื่อ 8 ธันวาคม 2553, จาก http://www2.m108.com/jma/index.php?view=article&id=60%3Avisual-basic--rs-232&option=com_content&Itemid=56

[5] วัชรินทร์ เดารพ. (24 มกราคม 2546). คู่มือการใช้งาน Servo Motor. สืบค้นเมื่อ 17 พฤศจิกายน 2553, จาก http:// www.ett.co.th/download/23Robot/ET-ROBOT.../ET-ROBOT-877.pdf

[6] ธนธรรม สองสี, ธวัชชัย บุญเหลือง, ธีระ เจียศิริพงษ์กุล, เริงวุฒิ ชูเมือง, วัชรินทร์ โพธิ์เงิน. (7 กันยายน 2553). การประชุมวิชาการเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทย ครั้งที่ 23 (4 – 7 พฤศจิกายน 2552). จังหวัดเชียงใหม่. ใน การตรวจสอบการวางลวดทองแดงของเครื่องประกอบลวดทองแดงแบบอัตโนมัติโดยการประมวลผลภาพ. สืบค้นเมื่อ 16 กันยายน 2553, จาก http://www.tsme.org/ME_NETT/ME_NETT23/topic/file/CST-009342.pdf

[7] Andrew, Kirillov. (November 15, 2009). **Programming C#.** Retrieved 15, 2011, from https://code.google.com/p/aforgc/issues/attachmentText?id=148&aid=-8127960866513521308&name=MainForm.cs&token=a44f7ea13a1659ae20f3760280f59

[8] อรรถวิทย์ สุดแสง. (มิถุนายน 2547). การวางแผนการเคลื่อนที่. สืบค้นเมื่อ 11 พฤศจิกายน 2554, จาก http://www.cp.eng.chula.ac.th/~attawith/class/motion.pdf

[9] Jubchay blogs. (4 มกราคม 2554). การแยกบริเวณรูปภาพ. สืบค้นเมื่อ 25 สิงหาคม 2554, จาก http://www.bloggang.com/viewblog.php?id=hin-kmitnb&date=19-01-2011&group=1&gblog=6

เอกสารอ้างอิง (ต่อ)

[10] บริษัท อีทีที. (13 กุมภาพันธ์ 2547). **Servo Futaba S3003**. สืบค้นเมื่อ 2 ธันวาคม 2553, จาก http://www.ett.co.th/product/1602.html

[11] บริษัท อีทีที. (13 กุมภาพันธ์ 2547). **Servo MG series**. สืบค้นเมื่อ 2 ธันวาคม 2553, จาก http://www.ett.co.th/product/1603.html

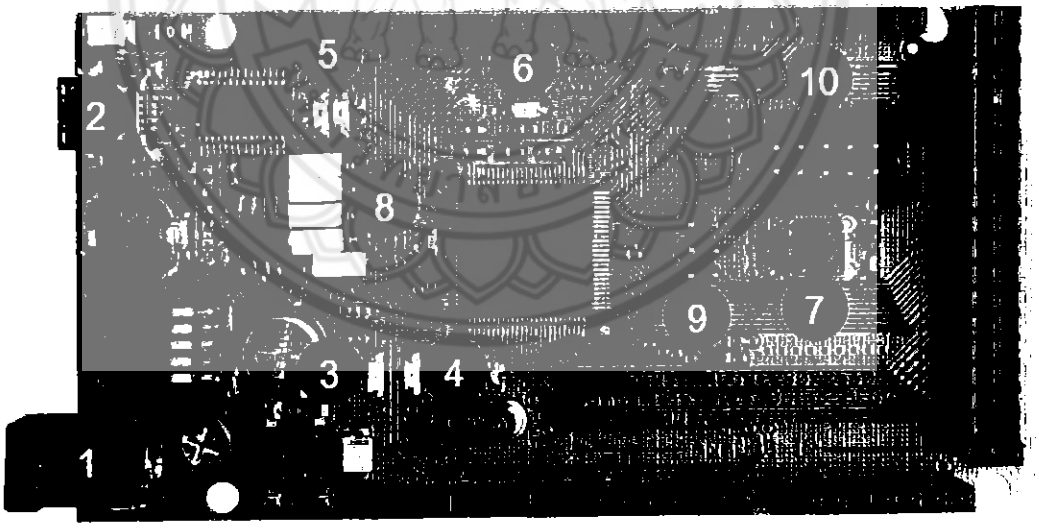
[12] เอกชัย มะการ. (2552). **เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทลเลอร์ ตระกูล AVR ด้วย Arduino**. (1). กรุงเทพฯ: บริษัท อีทีที จำกัด 1112/96-98 ถนนสุขุมวิท แขวงพระโขนง เขตคลองเตย 10110.

[13] วิรุณ ตั้งเจริญ. (กันยายน 2535). ทฤษฎีสี เพื่อการสร้างสรรค์ศิลปะ. **ทฤษฎีสี**. สืบค้นเมื่อ 20 กันยายน 2554, จาก http://www.prc.ac.th/newart/webart/colour01.html

ภาคผนวก ก

บอร์ดไมโครคอนโทรลเลอร์

การควบคุมการเคลื่อนไหวของแขนกลได้ใช้บอร์ดไมโครคอนโทรลเลอร์เป็นตัวควบคุมตำแหน่งต่าง ๆ ของมอเตอร์ โดยได้เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล AVR ATmega1280 ของของบริษัท Atmel รองรับการเขียนโปรแกรมภาษาซีของ Arduino ได้ทันทีมาเป็นหน่วยประมวลผลหลัก โดยชิพรุ่นนี้มีหน่วยความจำแฟลชสำหรับเก็บเขียนโปรแกรม 128 กิโลไบต์ มีหน่วยความจำแรม 8 กิโลไบต์ มี EEPROM อีก 4 กิโลไบต์สาหรับใช้เป็นที่เก็บข้อมูลถาวรได้เมื่อไม่มีไฟเลี้ยง และยังมีขานาเข้า/ส่งออก หรือไอ/โอ (I/O) สาหรับต่อใช้งานทั่วไป 86 ขา มี PWM (สาหรับควบคุมอัตราการหมุนของมอเตอร์) ที่กำหนดความละเอียดได้ระดับ 16 บิตให้ใช้งานถึง 12 ช่องสัญญาณ มีช่องสื่อสารแบบอนุกรม 4 พอร์ต และสามารถแปลงสัญญาณแอนาล็อกเป็นดิจิทัล (ADC) อีก 16 ช่องสัญญาณ และสามารถรองรับการดาวน์โหลดโปรแกรมได้ภายในบอร์ด ซึ่งจะเห็นว่ามีความสามารถพื้นฐานที่มากพอสาหรับงานควบคุมที่หลากหลาย สามารถทำงานได้โดยใช้ชิปเดียว มีคำสั่งที่สนับสนุนการเขียนโปรแกรมควบคุม จึงทำให้สามารถใช้งานได้อย่างสะดวก



หมายเลข 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงจากภายนอก สามารถใช้ได้กับแหล่งจ่ายทั้งแบบ AC และDC พร้อมวงจร Bridge Rectifier และ Switching ช่วยลดความร้อนของ IC Regulate เมื่อมีการดึงกระแสมากๆได้เป็นอย่างดี สามารถใช้กับแรงดัน Input 7-20V

หมายเลข 2 เป็นขั้วต่อ USB สำหรับติดต่อสื่อสารกับคอมพิวเตอร์ PC โดยใช้ FT232RL เป็นUSB Bridge ในการเชื่อมต่อระหว่างคอมพิวเตอร์ PC และ MCU ในบอร์ด และยัง

สามารถใช้ไฟจาก พอร์ต USB เป็นแหล่งจ่ายให้กับบอร์ดได้ด้วย โดยจะมี Poly Fuse ขนาด 500mA สำหรับป้องกันการดึงกระแสเกินจากพอร์ต USB ด้วย และที่พิเศษคือมีวงจรสำหรับตรวจสอบแหล่งจ่ายเพื่อสลับการใช้งานแหล่งจ่ายจาก USB ไปเป็น External Supply ได้เอง โดยอัตโนมัติ โดยเมื่อไม่ได้ต่อ External Supply บอร์ดจะใช้ไฟจากพอร์ต USB เป็นแหล่งจ่ายในการทำงาน แต่เมื่อ มีการต่อ External Supply วงจรจะสลับไปใช้แหล่งจ่ายจาก External Supply เองโดยอัตโนมัติ

- LED +VCC ใช้แสดงสถานะเมื่อมีการจ่ายไฟให้กับบอร์ด
- LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟจาก External Supply

หมายเลข 3 เป็น LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟเลี้ยงจาก External Supply

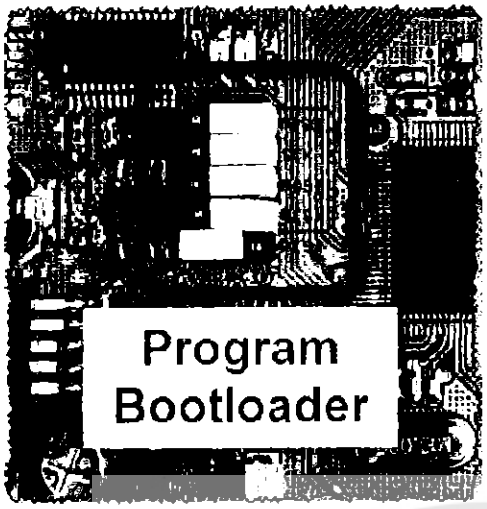
หมายเลข 4 เป็น LED +VCC ใช้แสดงสถานะของแหล่งจ่ายไฟเลี้ยง (+VCC) ของบอร์ด โดยเมื่อบอร์ดใช้แหล่งจ่ายจาก External Supply จะแสดงสถานะโดยการให้ LED VEXT และ LED +VCCติดสว่างพร้อมกันทั้งคู่ แต่ถ้าบอร์ดใช้แหล่งจ่ายจากพอร์ต USB จะแสดงสถานะโดยการให้ LED +VCC ติดสว่างเพียงดวงเดียว

หมายเลข 5 เป็น LED แสดงสถานะของ RX และ TX ใช้สำหรับแสดงการรับส่งข้อมูลระหว่างบอร์ด ET-EASY MEGA1280 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB

หมายเลข 6 เป็น LED D13 ใช้สำหรับทดแสดงการทำงานของ Bootloader และ ใช้ทดสอบการทำงานของบอร์ดจากการควบคุมของ Pin Digital-13 ทำงานด้วย Logic "1" และ หยุดทำงานด้วยLogic "0"

หมายเลข 7 เป็นสวิตช์ Reset ใช้สำหรับสั่ง Reset การทำงานของบอร์ด

หมายเลข 8 เป็นชุด Jumper สำหรับเลือก การ Program Bootloader ผ่าน USB Port และการใช้งานตามปรกติ

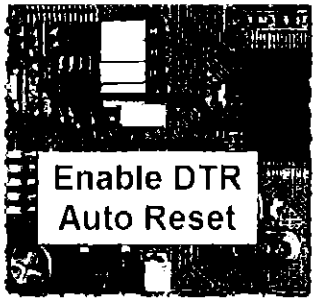


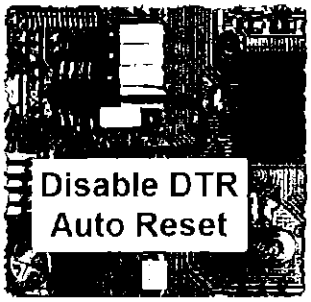
หมายเลข 9 เป็นขั้วต่อ AVRISP ใช้สำหรับ Download Code ให้กับ MCU โดยขั้วต่อ AVRISP นี้จะสามารถใช้งานได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ ATMEGA1280 และใช้ขั้วต่อตรงตามมาตรฐาน AVRISP

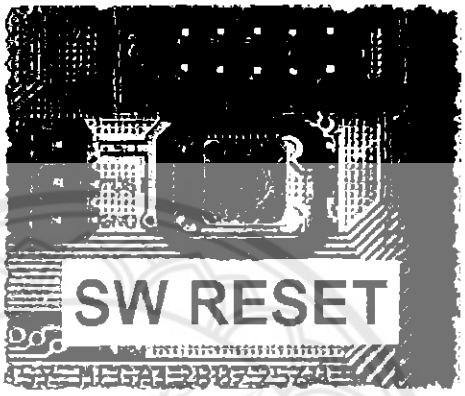
หมายเลข 10 เป็นขั้วต่อสัญญาณจาก D[22..29] สำหรับเชื่อมต่อกับบอร์ด I/O ของ อีทีที รวมทั้งจอแสดงผล LCD โดยใช้ร่วมกับ 10PIN LCD หรือ ET-CONV SPI TO LCD

3.3 คุณสมบัติของสัญญาณต่างๆของบอร์ด ET-EASY MEGA1280

RESET เป็นสัญญาณ Input Logic Reset ของ MCU เมื่อเป็น Logic Low จะทำให้ MCU อยู่ในสภาวะรีเซ็ต เมื่อเป็น Logic High จะทำให้ MCU อยู่ในสภาวะทำงานตามปรกติ โดยสัญญาณ RESET นี้จะถูกควบคุมจาก 2 แหล่ง คือ จาก สวิตช์ RESET ภายในบอร์ด และ จากสัญญาณ DTR ของ FT232RL ถ้ามีการเลือก Enable Jumper ของ Auto Reset จาก DTR ไว้ ดัง (รูปที่ 3)







+3V3 เป็นแหล่งจ่ายไฟขนาด +3.3V ที่ได้จากวงจร Regulate ภายในของ FT232RL สามารถจ่ายกระแสได้สูงสุด 50mA ซึ่งเมื่อต้องการนำแหล่งจ่าย +3.3V นี้ไปใช้งานเป็นแหล่งจ่ายให้กับอุปกรณ์ต่างๆ ต้องระมัดระวังเรื่องการดึงกระแสของโหลดด้วย ถ้าโหลดมีการดึงกระแสมากกว่า 50mAอาจทำให้ FT232RL เกิดความเสียหายได้

+VIN เป็นไฟ DC ที่รับมาจาก Jack VIN(External Supply) แต่ผ่านการ Rectifier และ Filter เป็นDC แล้ว มีขนาดแรงดันเฉลี่ยตามขนาดแรงดันที่ป้อนให้กับบอร์ดทาง Jack VIN

+5V เป็นจุดต่อแหล่งจ่ายไฟของบอร์ดออกไปใช้งาน ซึ่งมาจากแหล่งกำเนิด 2 แหล่ง คือ จากพอร์ตUSB และจาก External Supply ซึ่งถ้าต่อแหล่งจ่ายให้บอร์ดจาก External Supply ผ่านทาง JackVIN แหล่งจ่าย +5V นี้จะมาจาก Switching Regulate (LM2575-5V) สามารถจ่ายกระแสได้สูงสุดถึง 1A แต่ถ้าใช้แหล่งจ่ายจากพอร์ต USB แหล่งจ่าย +5V นี้จะมาจากพอร์ต USB โดยตรงโดยจะมีฟิวส์ แบบ Poly ขนาด 500mA ต่อป้องกันการดึงกระแสเกินเพื่อป้องกันความเสียหายของพอร์ตUSB โดยจะจ่ายกระแสได้สูงสุดไม่เกิน 500mA ขึ้นอยู่กับคุณสมบัติในการจ่ายกระแสของพอร์ตUSB และการ Configure ค่าให้กับ FT232RL ด้วย

A0-A15 เป็นขาสัญญาณ Analog Input แบบ ADC มีขนาดความละเอียด 10บิต มี 16 Pin สามารถรับแรงดัน Analog Input ได้ 0-5VDC

D0-D53 เป็นขาสัญญาณ Digital Input/Output แบบ TTL มีทั้งหมด 54 Pin สามารถใช้ทำหน้าที่เป็น Input หรือ Output ตามการกำหนดจากโปรแกรม โดยมีบาง Pin สามารถกำหนดหน้าที่ใช้งานเป็นฟังก์ชั่นพิเศษต่างๆเพิ่มเติมได้อีก

D0-D1 ถูกสงวนไว้ใช้ทำหน้าที่เป็นพอร์ตสื่อสารอนุกรม RS232 (UART0) โดยได้ทำการเชื่อมต่อกับ USB Bridge ของ FT232RL เพื่อใช้ Upload Code ให้กับบอร์ด และยังสามารถใช้ทดลองติดต่อสื่อสารรับส่งข้อมูลระหว่างบอร์ดกับคอมพิวเตอร์ PC ได้ด้วย

D2-D13 สามารถ โปรแกรมหน้าที่เป็น PWM ขนาด 8 บิต มี 14 Pin ได้

D14 สามารถ โปรแกรมหน้าที่เป็น TX3 สำหรับ ส่งข้อมูลของ UART3 ได้ด้วย

D15 สามารถ โปรแกรมหน้าที่เป็น RX3 สำหรับ รับข้อมูลให้กับ UART3 ได้ด้วย

D16 สามารถ โปรแกรมหน้าที่เป็น TX2 สำหรับ ส่งข้อมูลของ UART2 ได้ด้วย

D17 สามารถ โปรแกรมหน้าที่เป็น RX2 สำหรับ รับข้อมูลให้กับ UART2 ได้ด้วย

D18 สามารถ โปรแกรมหน้าที่เป็น TX1 สำหรับ ส่งข้อมูลของ UART1 ได้ด้วย

D19 สามารถ โปรแกรมหน้าที่เป็น RX1 สำหรับ รับข้อมูลให้กับ UART1 ได้ด้วย

D20 สามารถ โปรแกรมหน้าที่เป็น SDA ของ I2C Bus สำหรับใช้สื่อสารกับ I2C ได้ด้วย

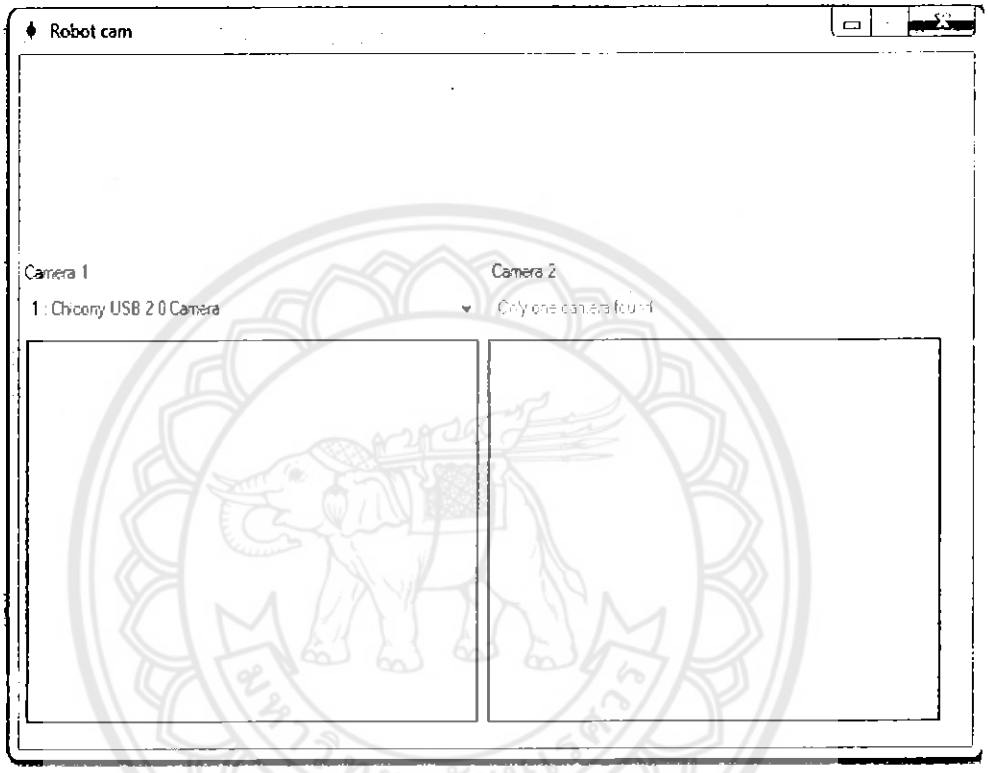
D21 สามารถ โปรแกรมหน้าที่เป็น SCL ของ I2C Bus สำหรับใช้สื่อสารกับ I2C ได้ด้วย

AREF เป็นสัญญาณ Analog Reference จากภายนอกที่ต้องการป้อนให้กับ MCU ซึ่งตามปรกติแล้ว ATMEGA1280 สามารถโปรแกรมให้เลือกใช้แรงดันอ้างอิงจากภายในได้อยู่แล้วโดยสามารถเลือกเป็น 1.1V หรือ 2.56V หรือ AVCC(+5V) โดยไม่จำเป็นต้องป้อนแรงดันอ้างอิงจากภายนอกให้กับบอร์ดอีก แต่ถ้าต้องการแรงดันอ้างอิงที่มีความแตกต่างจากที่กล่าวมาแล้วก็สามารถป้อนแรงดันอ้างอิงจากภายนอกผ่านทางขา AREF นี้เข้าไปเองได้ระหว่าง 0-5V

ภาคผนวก ข

การเขียนโปรแกรมเชื่อมต่อกล้องสองตัว

โดยจะต้องทำการออกแบบหน้าตาส่วนของ Application ดังข้างล่างนี้



รูปที่ ข-1 หน้าตาโปรแกรมกล้องสองตัว

โค้ดตัวอย่างการเขียนโปรแกรมเชื่อมต่อกล้องสองตัว

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Windows.Forms;
//
using System.Threading;
using System.IO.Ports;
using System.IO;
using System.Drawing.Text;
using System.Drawing.Drawing2D;
//
using AForge;
using AForge.Imaging;
```

```
using AForge.Imaging.Filters;
using AForge.Video;
using AForge.Video.DirectShow;
//
namespace TwoCamerasVision
{

    public partial class MainForm : Form
    {
        //list of video devices
        FilterInfoCollection videoDevices;
        // form to show detected objects
        DetectedObjectsForm detectedObjectsForm;
        // form to tune object detection filter
        TuneObjectFilterForm tuneObjectFilterForm;

        ColorFiltering colorFilter = new ColorFiltering( );
        GrayscaleBT709 grayFilter  = new GrayscaleBT709( );
              //use two blob counters,sothe could run in parallel
      BlobCounter blobCounter1 = new BlobCounter( );
        BlobCounter blobCounter2 = new BlobCounter( );

        private AutoResetEvent camera1Acquired = null;
        private AutoResetEvent camera2Acquired = null;
        private Thread trackingThread = null;
        // object coordinates in both cameras
        private float x1, y1, x2, y2;
        int originy, originx;

        public MainForm( )
        {
            InitializeComponent( );
            //กำหนดค่า  BaudRate and   serrialport
            serialPort1.BaudRate = 19200;
            serialPort1.Parity = Parity.None;
            serialPort1.DataBits = 8;

            timer2.Interval = 70;//กำหนดเวลา
            timer2.Start();
            button13.Enabled = false;
            button14.Enabled = false;
            button15.Enabled = false;
            button17.Enabled = false;
            button18.Enabled = false;


            servo[0] = 80;
            servo[1] = 32;
            servo[2] = 76;
            servo[3] = 90;
            servo[4] = 59;

            if (button13.Enabled == false)
            {
                sw_munnul = 1;
            }
            // show device list

            try
            {
                // enumerate video devices
```

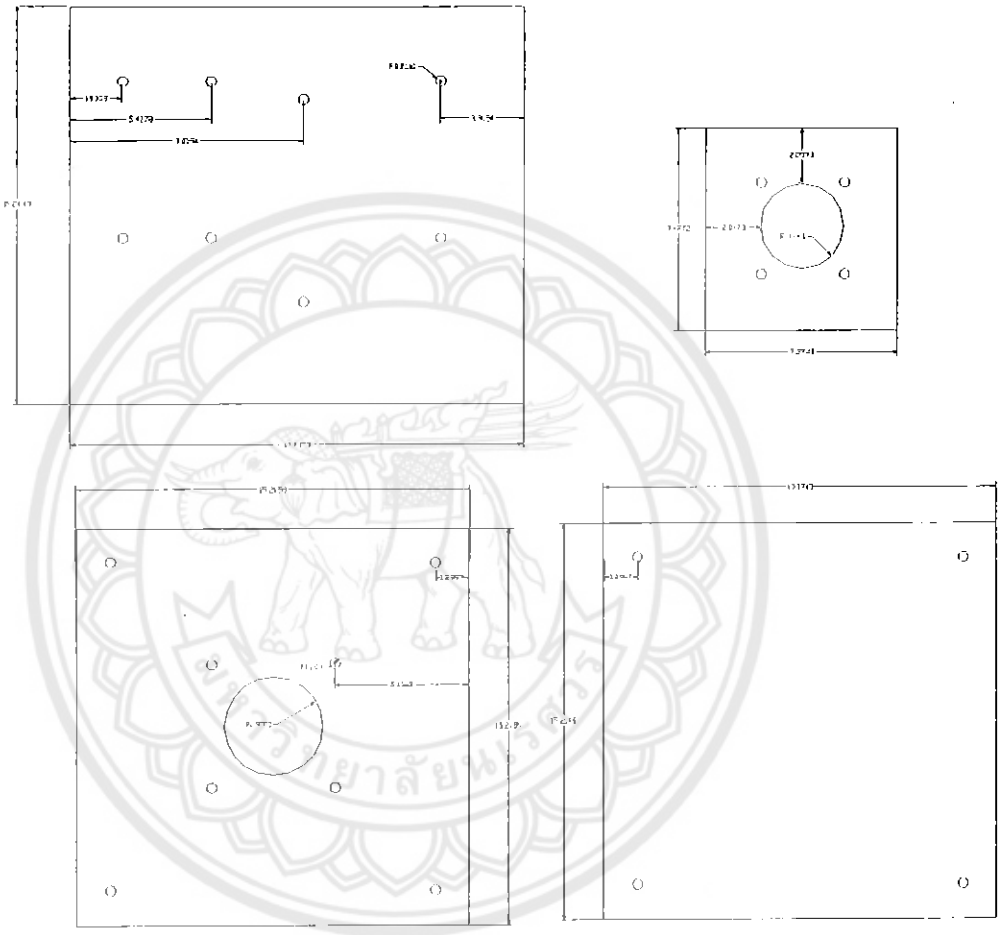
```
            videoDevices = new FilterInfoCollection(
FilterCategory.VideoInputDevice );

            if ( videoDevices.Count == 0 )
            {
                throw new Exception( );
            }
            //
            for (int i = 1, n = videoDevices.Count; i <= n; i++)
            {
             string cameraName = i+":" +videoDevices[i - 1].Name;

                camera1Combo.Items.Add(cameraName);
                camera2Combo.Items.Add(cameraName);
            }

            // check cameras count
            if ( videoDevices.Count == 1 )
            {
                camera2Combo.Items.Clear( );
                camera2Combo.Items.Add( "Only one camerafound" );
                camera2Combo.SelectedIndex = 0;
                camera2Combo.Enabled = false;
            }
            else
            {
                camera2Combo.SelectedIndex = 1;
            }
            camera1Combo.SelectedIndex = 0;
        }
        catch
        {
            camera1Combo.Items.Add( "No cameras found" );
            camera2Combo.Items.Add( "No cameras found" );

            camera1Combo.SelectedIndex = 0;
            camera2Combo.SelectedIndex = 0;

            camera1Combo.Enabled = false;
            camera2Combo.Enabled = false;
        }

        // configure blob counters
        //กล้องที่ 1
        blobCounter1.MinWidth  = 25;//
        blobCounter1.MinHeight = 25;
        blobCounter1.FilterBlobs = true;
        blobCounter1.ObjectsOrder = ObjectsOrder.Size;//ขน

        //กล้องที่ 2
        blobCounter2.MinWidth = 25;
        blobCounter2.MinHeight = 25;
        blobCounter2.FilterBlobs = true;
        blobCounter2.ObjectsOrder = ObjectsOrder.Size;
    }



    // Main form closing - stop cameras
```

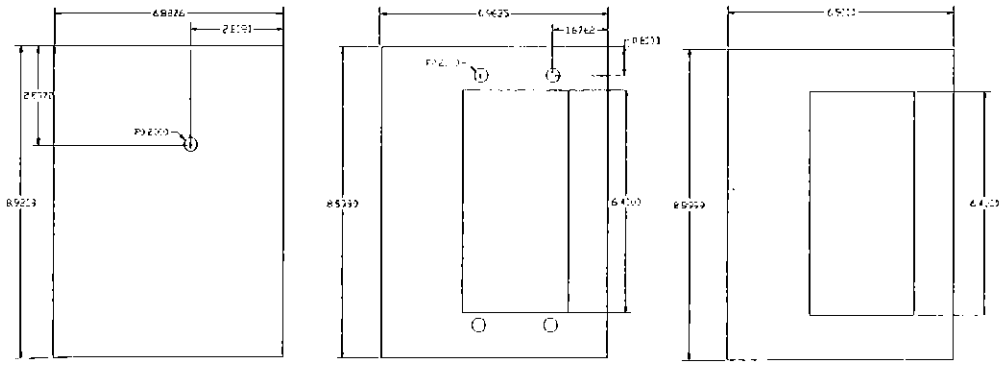
```
  private void MainForm_FormClosing( object sender,
FormClosingEventArgs e )
  {
      StopCameras( );
  }
  // On "Start" button click - start cameras

  // On "Stop" button click - stop cameras
  private void stopButton_Click( object sender, EventArgs e )
  {
      StopCameras( );
      sw_munnul = 0;
  }

  // Start cameras
  private void StartCameras( )
  {
      // create first video source
            VideoCaptureDevice videoSource1 = new
VideoCaptureDevice(
videoDevices[camera1Combo.SelectedIndex].MonikerString );
      videoSource1.DesiredFrameRate = 10;
      videoSourcePlayer1.VideoSource = videoSource1;
      videoSourcePlayer1.Start( );
      // create second video source
      if ( camera2Combo.Enabled == true )
      {
          System.Threading.Thread.Sleep( 500 );

          VideoCaptureDevice videoSource2 = new
VideoCaptureDevice(
videoDevices[camera2Combo.SelectedIndex].MonikerString );
          videoSource2.DesiredFrameRate = 10;

          videoSourcePlayer2.VideoSource = videoSource2;
          videoSourcePlayer2.Start( );
      }

      camera1Acquired = new AutoResetEvent( false );
      camera2Acquired = new AutoResetEvent( false );
      // start tracking thread
     trackingThread = new Thread( new ThreadStart(
TrackingThread ) );
      trackingThread.Start( );
  }

  // Stop cameras
  private void StopCameras( )
  {
      videoSourcePlayer1.SignalToStop();
      videoSourcePlayer2.SignalToStop();
      videoSourcePlayer1.WaitForStop();
      videoSourcePlayer2.WaitForStop();
      if ( detectedObjectsForm != null )
      {
          detectedObjectsForm.UpdateObjectPicture( 0, null );
          detectedObjectsForm.UpdateObjectPicture( 1, null );
      }

      if ( trackingThread != null )
      {
```

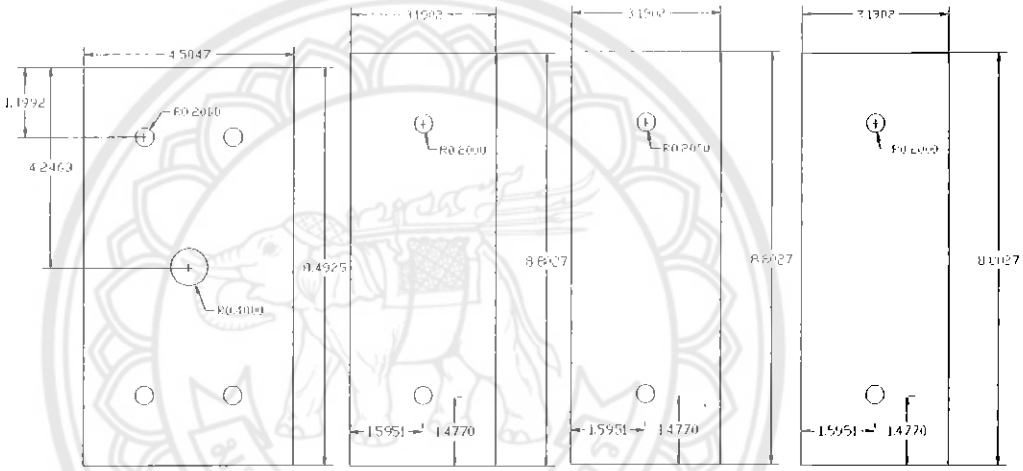
```
                    //signal tracking thread to stop
                    x1 = y1 = x2 = y2 = -1;
                    camera1Acquired.Set( );
                    camera2Acquired.Set( );
                    trackingThread.Join( );
                }
            }
        }
```

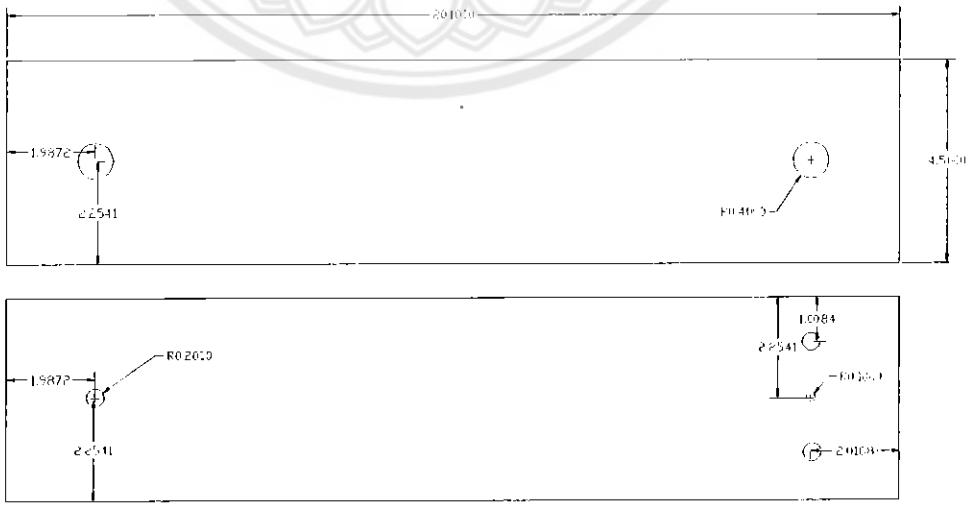
ภาคผนวก ค

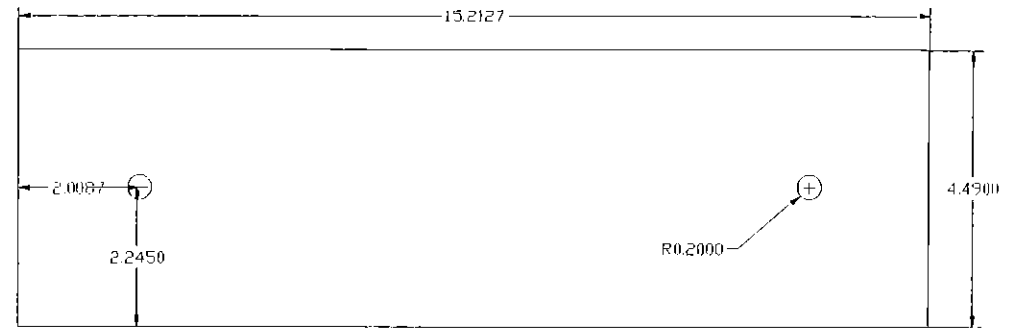
แบบร่างของแขนกลที่ใช้ในโครงงาน

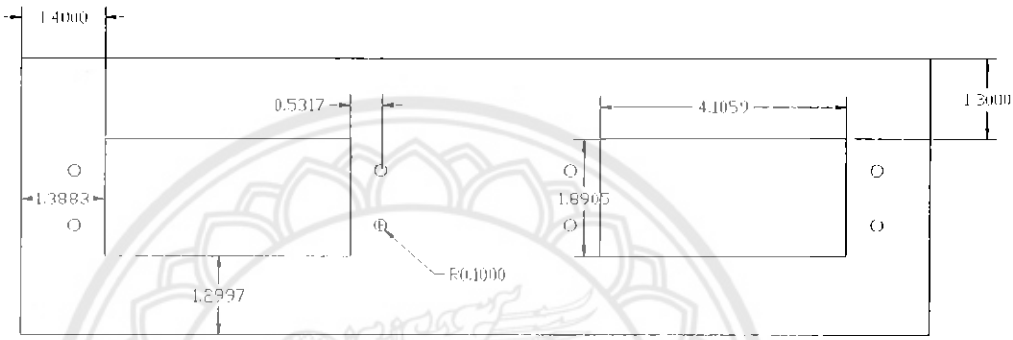


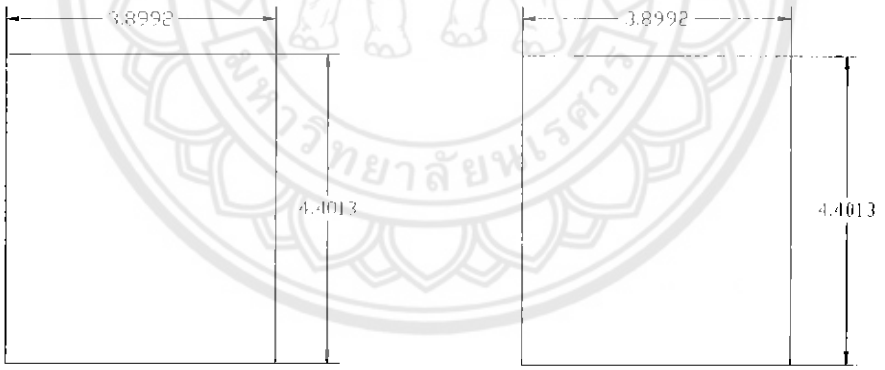


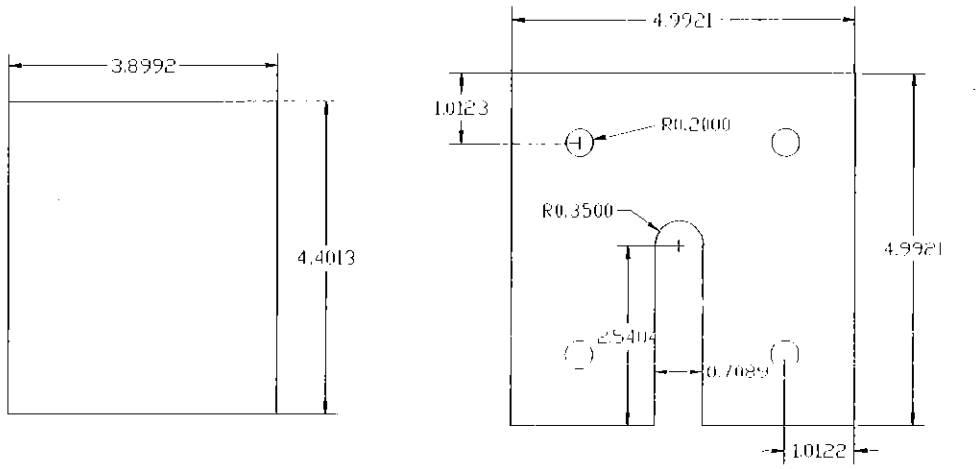





15.2127
2.0087
2.2450
R0.2000
4.4900


1.4000
0.5317
4.1059
1.3000
1.3883
1.8905
R0.1000
1.8997


3.8992
4.4013
3.8992
4.4013


3.8992
4.4013
4.9921
1.0123
R0.2000
R0.3500
4.9921
2.5404
0.7089
1.0122

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายภัทรพล ศิวปรียานุพงษ์

ภูมิลำเนา 172 หมู่ 15 ต.นาบัว อ.นครไทย จ. พิษณุโลก

ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนต้นจากโรงเรียนนาบัววิทยาคม
- จบระดับประกาศนียบัตรวิชาชีพ (ปวช.) จากโรงเรียนบริหารธุรกิจและเทคโนโลยีพิษณุโลก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 7 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: sakookl@hotmail.com