

อภิธานนาการ



สำนักหอสมุด

การพัฒนาระบบควบคุมกล้องติดตามวัตถุ  
Development of an object tracking camera control system

สำนักหอสมุด มหาวิทยาลัยธนเรศวร
รับลงทะเบียน.....
เลขทะเบียน..... 17196429
เลขเรียกหนังสือ.....

นายสรารุติ จิตต์ใจฉ่ำ รหัส 51361445

ร/ร  
ร 359 ก  
2557

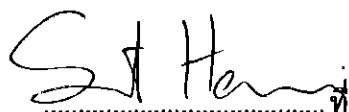
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล  
คณะวิศวกรรมศาสตร์มหาวิทยาลัยธนเรศวร  
ปีการศึกษา 2557

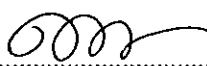


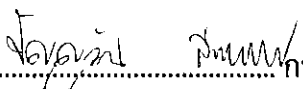
## ใบรับรองโครงการงาน

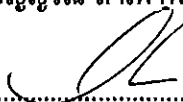
หัวข้อโครงการงาน	การพัฒนาระบบควบคุมกล้องติดตามวัตถุ Development of an object tracking camera control system
ผู้ดำเนินโครงการงาน	สรารุณี จิตดีใจฉ่ำ รหัสนิต 51361445
อาจารย์ที่ปรึกษา	ดร.สุเมธ เหมาะวัฒนะชัย
ที่ปรึกษาร่วม	ผศ.ดร.พรพิศุทธิ์ วรจิรันตนัน
สาขาวิชา	วิศวกรรมเครื่องกล
ภาควิชา	วิศวกรรมเครื่องกล
ปีการศึกษา	2557

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏบรจรัม อนุมัติให้โครงการงานวิจัยฉบับนี้เป็นส่วนหนึ่งของ  
การศึกษาดำเนินหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล

  
..... ที่ปรึกษาโครงการงาน  
(ดร.สุเมธ เหมาะวัฒนะชัย)

  
..... ที่ปรึกษาร่วม  
(ผศ.ดร.พรพิศุทธิ์ วรจิรันตนัน)

  
..... กรรมการ  
(ดร.ปัญญาวัฒน์ ลำเพาพงศ์)

  
..... กรรมการ  
(ผศ.ดร.ขวัญชัย ไกรทอง)

หัวข้อโครงการ : การพัฒนาระบบควบคุมกล้องติดตามวัตถุ  
ผู้ดำเนินโครงการ : สราวุฒิจิตตใจฉ่ำ รหัสสนิสิต 51361445

ที่ปรึกษาโครงการ : ดร.สุเมธ เหมะวัฒนะชัย  
ผู้ช่วยที่ปรึกษาโครงการ : ผศ.ดร.พรพิศุทธิ์ วรจิรันตน์  
ภาควิชา : วิศวกรรมเครื่องกล  
ปีการศึกษา : 2557

---

### บทคัดย่อ

งานวิจัยนี้เป็นการศึกษาและพัฒนาระบบตรวจจับตำแหน่งและติดตามวัตถุด้วยเทคโนโลยีภาพ ในราคาที่ประหยัด โดยใช้กล้องเว็บแคม (Webcam) จำนวนหนึ่งตัว ซึ่งมีราคาไม่แพงมาใช้ในการตรวจจับตำแหน่งของวัตถุ โดยมีการพัฒนาซอฟต์แวร์บนโปรแกรม LabVIEW ในการเลือกสีของวัตถุที่ต้องการตรวจจับ เมื่อระบบคอมพิวเตอร์ประมวลตำแหน่งวัตถุแล้ว จะส่งสัญญาณไปควบคุมสเตปมอเตอร์ เพื่อหมุนกล้องให้หันหน้าตรงไปตามตำแหน่งที่วัตถุเคลื่อนที่ไปได้ ซึ่งวิธีการนี้เป็นการช่วยเพิ่มพื้นที่การตรวจจับของกล้อง โดยตำแหน่งของวัตถุที่แม่นยำสามารถคำนวณได้จากมุมที่หันไปของกล้องร่วมกับพิกัดวัตถุในจอภาพที่กล้องจับได้

ในการทดลองเพื่อทดสอบความถูกต้องของระบบทำโดยการตรวจจับวัตถุบนระนาบอ้างอิงที่มีระยะห่างในแนวตั้งฉาก ไปถึงหน้ากล้องประมาณ 100 cm โดยมีการทดลองด้วยการเคลื่อน marker ไปตามจุดต่างๆบนระนาบ แล้วเปรียบเทียบตำแหน่ง marker ที่คำนวณได้กับค่าตำแหน่งจริงบนระนาบอ้างอิง ซึ่งผลการทดลองพบว่าระบบที่พัฒนาขึ้นสามารถหันกล้องติดตามวัตถุได้อย่างแม่นยำ โดยได้หลากหลายสีตามที่ตั้งค่าไว้ เช่น สีแดง เหลือง และม่วง เป็นต้น โดยมีค่าความผิดพลาดของระบบตรวจจับตำแหน่งบนระนาบอ้างอิงไม่เกิน 1 cm ในแนวแกน x และน้อยกว่า 0.5 ในแนวแกน y โดยกล้องทำมุมกับแนวแกนตั้งฉาก ในช่วงตั้งแต่ -16.2 ถึง 16.2 องศา

Project Title : Development of an object tracking camera control system  
Name : Mr.Sravut Jitjaicham  
Project Advisor : Dr. Sumet Heamawatanachai  
ASST. Project advisor : Asst. Prof. Dr. Ponpisut Worrajiran  
Academic Year : 2014

---

### Abstract

This research is a study and development of a low-cost object tracking camera control system. A web camera is used for image acquisition. A software was developed on LabVIEW program to select color of object to be tracked. After analyzing the position of the object, the software will send a command to a stepping motor to rotate the camera to the direction of the object. This method also increases the area of detection of the camera. The coordinate position of the object can be calculated from angular direction of the camera and the information of the object (pixels) in the captured image.

To demonstrate the accuracy of the system. A reference plane was set at the distance of 100 cm from the camera. A marker was place and moving slowly on the reference plane, then the system calculated the position of the marker and compare with the actual position. The experiment results reveal that the developed system was able to rotate the camera to follow the motion of the object correctly. There were many color of object that can be set for tracking such as red, yellow and purple. The experiment also show that the calculated tracking position errors were not over than 1 cm for x axis and less than 0.5 cm for the y-axis over the camera's angular position between -16.2 to 16.2 degree.

## กิตติกรรมประกาศ

โครงการวิศวกรรมเครื่องกลฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ทางคณะผู้ดำเนินงาน ต้องขอขอบพระคุณ ดร.สุเมธ เทมะวัฒน์ชัย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการ และ ผศ.ดร.พรพิศุทธิ์ วรจิรันตน์ อาจารย์ที่ปรึกษาโครงการร่วม ที่กรุณาให้คำปรึกษาและชี้แนะแนวทางการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นในระหว่างการดำเนินโครงการ ตลอดจนติดตามประเมินผลการดำเนินโครงการมาโดยตลอด ทางคณะผู้จัดทำขอขอบพระคุณท่านอาจารย์เป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณท่านอาจารย์ทุกท่าน ที่อบรมสั่งสอน และให้ความรู้แก่ผู้ดำเนินงาน

ขอขอบพระคุณฝ่ายเลขานุการ ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ช่วยอำนวยความสะดวกในการดำเนินโครงการ

ขอขอบคุณหน่วยวิจัยเทคโนโลยีด้านวิศวกรรมความเที่ยงตรงและการแพทย์ (PMET) ที่ได้ให้ใช้สถานที่ในการทำโครงการ และให้อุปกรณ์อำนวยความสะดวกต่างๆ

ขอขอบพระคุณบิดามารดา ที่ให้การอุปการะเลี้ยงดูและสั่งสอนจนกระทั่งสามารถเติบโตมาจนถึงปัจจุบัน ตลอดจนช่วยอุปการะทางการเงินและคอยให้กำลังใจ จนกระทั่งโครงการนี้เสร็จสมบูรณ์

สุดท้ายนี้ ผู้ดำเนินงานขอมอบคุณงามความดีที่เกิดขึ้นจากโครงการนี้ แต่ผู้มีพระคุณทุกท่านที่มีส่วนร่วมในการทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี และถ้าเกิดข้อผิดพลาดประการใดจากโครงการนี้ ผู้ดำเนินงานต้องกราบขออภัยไว้ ณ ที่นี้ด้วย

ผู้ดำเนินโครงการวิศวกรรมศาสตร์

นาย สราวุฒิ จิตต์ใจฉ่ำ

กรกฎาคม 2559

## สารบัญ

	หน้า
ปกใน .....	ก
ใบรับรองปริญญาโท .....	ข
บทคัดย่อ .....	ค
Abstract.....	ง
กิตติกรรมประกาศ .....	จ
สารบัญ .....	ช
สารบัญตาราง .....	ฉ
สารบัญรูป .....	ญ
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ .....	2
1.4 ขอบเขตของโครงการ .....	2
1.5 ขั้นตอนการดำเนินงาน .....	2
1.6 แผนการดำเนินงาน .....	3
1.7 รายละเอียดงบประมาณตลอดโครงการ .....	3
บทที่ 2 ทฤษฎี .....	4
2.1 ความหมายของกล้องดิจิทัล .....	4
2.2 กระบวนการถ่ายภาพด้วยระบบดิจิทัล .....	5
2.3 เซ็นเซอร์รับภาพในกล้องดิจิทัล .....	6
2.4 ความละเอียด (resolution) .....	7
2.5 กล้องเว็บแคม (Webcam) .....	7
2.6 ระบบสี .....	8
2.7 หลักการหาค่าต่างระดับโดยวิธีตรีโกณมิติ .....	12
2.8 หลักการทำงานของระบบและการระบุพิกัด .....	13
2.9 สเตปปีงมอเตอร์ (Stepping Motor) .....	17
2.10 ไมโครคอนโทรลเลอร์ .....	22

## สารบัญ (ต่อ)

หน้า

บทที่ 2 (ต่อ)	
2.11 ไมโครคอนโทรลเลอร์ Arduino .....	23
2.12 ส่วนประกอบต่างๆ ใน LabVIEW .....	27
บทที่ 3 วิธีการดำเนินงาน .....	30
3.1 ศึกษาและออกแบบระบบตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่าย.....	30
3.2 การพัฒนาโปรแกรมตรวจจับวัตถุ ด้วยโปรแกรม LabVIEW .....	33
3.3 วิธีทำการทดลอง.....	38
3.4 เครื่องมือและอุปกรณ์ที่ใช้.....	46
บทที่ 4	
ผลการทดลอง และวิเคราะห์ผลการทดลอง .....	50
4.1 การศึกษาและรวบรวมข้อมูลจากการทดสอบแยกแยะสีจากกลุ่มตัวอย่าง ....	50
4.2 ทำการทดลองเพื่อสอบเทียบหาค่า Pixels per Radian .....	54
4.3 ทดสอบการจับตำแหน่งวัตถุหยุดนิ่งบนระนาบ .....	56
4.4 ทดสอบการกลิ้งเคลื่อนที่ติดตามวัตถุเป้าหมาย .....	61
บทที่ 5	
สรุปโครงการ และข้อเสนอแนะ .....	66
5.1 สรุปผล .....	66
5.2 ข้อเสนอแนะ .....	66
อ้างอิง.....	67
ประวัติผู้จัดทำ .....	68

## สารบัญตาราง

	หน้า
ตารางที่ 1.1 ระยะเวลาและแผนปฏิบัติงาน.....	2
ตารางที่ 2.1 ค่าความละเอียด Pixels.....	7
ตารางที่ 2.2 Single-Phase Driver .....	19
ตารางที่ 2.3 Two-Phase Driver .....	19
ตารางที่ 2. 4 Half Step Motor .....	20
ตารางที่ 3.1 เครื่องมือและอุปกรณ์ที่ใช้ .....	46
ตารางที่ 4.1 ผลการทดลองค่า HSL ที่วัดได้ของ Marker .....	52
ตารางที่ 4.2 สรุปค่า HSL ที่วัดได้ของ Marker .....	53
ตารางที่ 4.3 ค่า PPR เฉลี่ยจากการทำการทดลอง .....	55
ตารางที่ 4.4 ตัวอย่างผลต่างระหว่างค่า X, Y และ Xc, Yc ที่ได้จากการหาค่า error ในทดลอง ...	57



## สารบัญรูป

	หน้า
รูปที่ 2.1 การทำงานของกล้องดิจิทัล .....	4
รูปที่ 2.2 กล้องWeb Camera รูปแบบต่างๆ .....	5
รูปที่ 2.3 การทำงาน Sensor CCD และ CMOS.....	6
รูปที่ 2.4 ตัวอย่างการใช้กล้อง Web Camera .....	7
รูปที่ 2.5 ระบบสีแบบ RGB .....	8
รูปที่ 2.6ระบบสี CMYK .....	9
รูปที่ 2.7ระบบสีแบบ HSB หรือ HSL .....	10
รูปที่ 2.8 ระบบสี Lab .....	11
รูปที่ 2.9 การวัดความสูงโดยวิธีตรีโกณมิติ.....	12
รูปที่ 2.10 ภาพรวมระบบติดตามวัตถุและการระบุพิกัด .....	13
รูปที่ 2.11 มุมมอง top view ของ Object Plane เมื่อทำการหันกล้อง .....	14
รูปที่ 2.12 มุมมอง side view ของกล้องตั้งฉากกับ Object Plane .....	15
รูปที่ 2.13 หน้าจอก่อนปรับตำแหน่งวัตถุที่จุด (2,2) .....	16
รูปที่ 2.14 หน้าจอหลังปรับเข้าสู่ (0,2) .....	16
รูปที่ 2.15 หน้าจอตำแหน่งวัตถุซึ่งเยื้องจากแนวกลางเล็กน้อย .....	16
รูปที่ 2.16 Unipolar Step Motor .....	17
รูปที่ 2.17 สเตปปีงมอเตอร์ .....	18
รูปที่ 2.18 สเตปมอเตอร์ 4 เฟส แบบยูนิโพลาร์ .....	18
รูปที่ 2.19 การใช้มิเตอร์วัดค่าความต้านทาน Step Motor .....	21
รูปที่ 2.20 การต่อวงจรเพื่อทดสอบโดยการสวิตช์เพื่อหาลำดับสาย .....	21
รูปที่ 2.21 รูปการต่อ Arduino กับบอร์ดคอนเนกประสงค์ และ ต่อแบบ Shield .....	23

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 2.22 รายละเอียดตัวบอร์ด .....	24
รูปที่ 2.23 การสื่อสาร Arduino ผ่านสาย USB .....	25
รูปที่ 2.24 เลือกบอร์ด Arduino ที่ต้องการ upload .....	25
รูปที่ 2.25 เลือกหมายเลข Comport ของบอร์ด .....	26
รูปที่ 2.26 การตรวจสอบความถูกต้องของโปรแกรม และการอัปเดตลงบอร์ด .....	26
รูปที่ 2.27 Front Panel ของ LabVIEW .....	27
รูปที่ 2.28 Object ที่อยู่บน Front Panel ของ LabVIEW .....	28
รูปที่ 2.29 Controls Palette ที่ใช้ในการออกแบบ Front Panel .....	28
รูปที่ 2.30 Tools Palette ที่ใช้ในการออกแบบ Front Panel .....	29
รูปที่ 2.31 ตัวอย่าง Block Diagram .....	29
รูปที่ 3.1 แผนภาพการทำงานของกระบวนการตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่าย .....	30
รูปที่ 3.2 รูป Object Plan และ Image Plane .....	31
รูปที่ 3.3 แบบเค้าโครงในการทดลอง และการวางแนวแกน X,Y,Z .....	31
รูปที่ 3.4 บอร์ดชุดควบคุมการทำงาน .....	32
รูปที่ 3.5 การวางชุดควบคุมบนฐานยึด Step motor .....	32
รูปที่ 3.6 การต่อที่จับดอกสว่านกับ Step motor และกลิ้ง.....	33
รูปที่ 3.7 ภาพรวมของ Diagram ที่ใช้ในการทดลอง .....	34
รูปที่ 3.8 ภาพรวมของโปรแกรมที่ใช้ในการทดลอง .....	34
รูปที่ 3.9 การควบคุมการตัดแยก Marker ด้วย ระบบสีและ Pixel.....	35
รูปที่ 3.10 ผลการตรวจจับวัตถุเป้าหมายด้วยโปรแกรม .....	36

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.11 ส่วนส่งคำสั่งและแสดงผลการควบคุม Step Motor .....	37
รูปที่ 3.12 รายละเอียดในการใส่ค่าในการทดลองและ ตัวเลขผลการทดลอง .....	37
รูปที่ 3.13 การควบคุมสเตปมอเตอร์ผ่าน Arduino IDE .....	38
รูปที่ 3.14 การทดลองการคัดกรองโดยการปรับค่า HSL ของ Yellow Marker .....	39
รูปที่ 3.15 การกำหนดค่าในระนาบแกน X, Y ของ Image Plane .....	40
รูปที่ 3.16 รายละเอียดต่างของโปรแกรมในการบันทึกค่า ในการคำนวณหา Pixels per Radian ..	41
รูปที่ 3.17 ภาพรวมการบันทึกค่า Yellow Marker ครั้งที่ 1 .....	41
รูปที่ 3.18 การทดสอบบอกพิกัดระยะจากการคำนวณค่าที่ได้จากกล้อง Web Camera .....	43
รูปที่ 3.19 การติดตามวัตถุเคลื่อนที่ตาม Yellow Marker .....	44
รูปที่ 3.20 การออกแบบการเคลื่อนที่เพื่อทดสอบบอกพิกัดติดตามวัตถุเคลื่อนที่ .....	44
รูปที่ 3.21 รูปกราฟที่ได้จากค่าการทดลองติดตามการเคลื่อนที่ .....	45
รูปที่ 3.22 ไมโครคอนโทรเลอร์ Arduino .....	47
รูปที่ 3.23 บอร์ดขับเคลื่อนมอเตอร์ YFROBOT Motor Shield .....	47
รูปที่ 3.24 Step Motor Minebea 5V 2A .....	47
รูปที่ 3.25 ฐานรองการทดลอง ยึดมอเตอร์ และวางระบบควบคุม .....	48
รูปที่ 3.26 ภาพของสายไฟที่ใช้ต่อวงจรภายในระบบ .....	48
รูปที่ 3.27 ภาพสาย USB ที่ใช้ต่อบอร์ด Arduino กับคอมพิวเตอร์ .....	48
รูปที่ 3.28 กล้อง Web Camera ยี่ห้อ OKER 5 ล้าน Pixel .....	49
รูปที่ 3.29 Marker เป้ากระดาษสีทรงกลม .....	49

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.30 ที่จับตอกสว่าน MACOH รุ่นเกลียว ½ “ .....	49
รูปที่ 4.1 การทำงานทั้งหมดของระบบตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่าย .....	50
รูปที่ 4.2 หน้าจอขณะทำงานทดลอง .....	51
รูปที่ 4.3 รายละเอียดหน้าจอขณะทำงานทดลอง ( zoom รูปที่ 4.2 ) .....	51
รูปที่ 4.4 ลำดับการระบุตำแหน่ง Marker .....	54
รูปที่ 4.5 รายละเอียดหน้าจอขณะทำการทดลอง .....	57
รูปที่ 4.6 ตัวอย่างการทดลองที่ P เท่ากับ 8 .....	57
รูปที่ 4.7 ตัวอย่างการทดลองที่ P เท่ากับ -3 .....	58
รูปที่ 4.8 ตัวอย่างการทดลองที่ P เท่ากับ -7 .....	58
รูปที่ 4.9 กราฟค่า Error ในแนวแกน X .....	59
รูปที่ 4.10 กราฟค่า Error ในแนวแกน Y .....	60
รูปที่ 4.11 กระดาษติด Marker ที่ถอดสีเหลืองออกและ Marker ติดด้ามโลหะ .....	61
รูปที่ 4.12 ตัวอย่างการติดตามวัตถุ (1) .....	62
รูปที่ 4.10 ตัวอย่างการติดตามวัตถุ (2) .....	62
รูปที่ 4.11 ตัวอย่างการติดตามวัตถุ (3) .....	63
รูปที่ 4.15 การเลื่อนที่ในลักษณะสี่เหลี่ยม .....	64
รูปที่ 4.16 การเคลื่อนที่ในลักษณะฟันปลา .....	65
รูปที่ 4.17 การเลื่อนที่ในลักษณะฟันปลาทรงเหลี่ยม .....	65

## บทที่ 1

### บทนำ

#### 1. ความเป็นมา และความสำคัญของปัญหา

การตรวจวัดตำแหน่งของวัตถุได้อย่างถูกต้องนั้น มีความสำคัญต่อการวิเคราะห์พฤติกรรม การเคลื่อนที่ของวัตถุ โดยตามหลักกลศาสตร์นั้น พิกัดตำแหน่งของวัตถุที่เวลาต่างๆ สามารถนำมาใช้ในการคำนวณหาความเร็วและความเร่งของการเคลื่อนที่ของวัตถุนั้นๆ ตลอดจนการประมาณแรงที่กระทำกับวัตถุได้ ซึ่งมีประโยชน์ในหลายลักษณะงาน เช่นในด้านวิศวกรรมหุ่นยนต์ การใช้แขนกลช่วยหยิบจับวัตถุนสายพานการผลิต นั้นจำเป็นต้องมีระบบตรวจวัดตำแหน่งและความเร็วของวัตถุนสายพานลำเลียง เพื่อเป็นข้อมูลให้ระบบควบคุมแขนกลให้เคลื่อนที่ด้วยตำแหน่งและความเร็วที่ถูกต้องเหมาะสมในการหยิบจับในด้านวิศวกรรมการแพทย์มีการนำเทคโนโลยีภาพถ่าย ( Machine Vision Technology ) มาใช้ โดยการใช้ระบบกล้องความเร็วสูงมาช่วยในการตรวจจับการเคลื่อนที่ของข้อต่อเพื่อช่วยในการประเมินความผิดปกติในการเดินของผู้ป่วย เป็นต้น

ระบบตรวจจับการเดินของผู้ป่วยที่กล่าวมานั้น มีราคาแพงหลายล้านบาท เนื่องจากต้องนำเข้าจากต่างประเทศ โดยจะมีระบบให้แสงในย่านอินฟราเรด ไปสะท้อนกับมาร์กเกอร์ ( Marker ) ซึ่งต้องทำการติดอยู่กับข้อต่อส่วนขาของผู้ป่วยระหว่างการทดสอบ ประกอบกับต้องใช้กล้องที่มีความเร็วในการจับภาพสูง ( 100-1000 เฟรมต่อวินาที ) มาประมวลผลพร้อมกันหลายตัวเพื่อให้ได้มุมมองที่กว้างขึ้นครอบคลุมพื้นที่ทดสอบ ซึ่งแม้ว่าระบบดังกล่าวจะให้ผลการตรวจวัดที่แม่นยำ และสามารถวัดตำแหน่งมาร์กเกอร์ได้พร้อมกันหลายตัว แต่ก็ยังมีข้อจำกัดในด้านวัตถุที่ตรวจจับได้ โดยทั่วไปต้องเป็นMarker จากบริษัทผู้ผลิตเท่านั้น ดังนั้นงานวิจัยนี้จึง ได้พัฒนาแนวคิดเพื่อสร้างระบบตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่ายในราคาที่เหมาะสม โดยการใช้กล้องเว็บแคม ( Webcam ) จำนวนหนึ่งตัว ซึ่งมีราคาไม่แพงและมีความเร็วการจับภาพประมาณ 25 เฟรมต่อวินาที มาใช้ในการตรวจจับตำแหน่งของวัตถุ โดยจะมีการพัฒนาซอฟต์แวร์ในการเลือกสีของวัตถุที่ต้องการตรวจจับ เมื่อระบบคอมพิวเตอร์ประมวลตำแหน่งแล้ว จะส่งสัญญาณไปควบคุมสเตปมอเตอร์ เพื่อหมุนกล้องให้หันหน้าตรงไปตามตำแหน่งที่วัตถุเคลื่อนที่ไปได้ ซึ่งวิธีการนี้เป็นการช่วยเพิ่มพื้นที่การตรวจจับของกล้อง โดยตำแหน่งของวัตถุที่แม่นยำจะสามารถคำนวณได้จากมุมที่หันไปของกล้องร่วมกับพิกัดวัตถุในจอภาพที่กล้องจับได้

## 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 พัฒนาระบบควบคุมการหันกล้องด้วยสแตปมอเตอร์เพื่อติดตามตำแหน่งวัตถุ
- 1.2.2 พัฒนาโปรแกรมวิเคราะห์ตำแหน่งวัตถุจากภาพถ่าย
- 1.2.3 การทดสอบความถูกต้องระบบจับภาพและวัดตำแหน่ง
- 1.2.4 ทดลองตรวจจับและวิเคราะห์ตำแหน่งของวัตถุ

## 1.3 ขอบเขตงานวิจัย

- 1.3.1 พัฒนาระบบควบคุมการหันกล้อง 1 แกนหมุนเพื่อติดตามวัตถุ
- 1.3.2 ทดลองตรวจจับตำแหน่งของวัตถุใน 1 ระนาบ (บนกระดาน) โดยวิเคราะห์ตำแหน่งเมื่อวัตถุหยุดนิ่ง

## 1.4 ขั้นตอนการดำเนินงาน

- 1.4.1 ศึกษาข้อมูลการดำเนินงาน
- 1.4.2 นำเสนอโครงร่างโครงการ
- 1.4.3 ออกแบบอุปกรณ์ และวางแผนการทดลอง
- 1.4.4 ดำเนินการสร้างอุปกรณ์ และชุดทดลอง
- 1.4.5 เก็บข้อมูลการทดลอง
- 1.4.6 สรุปผลการทดลอง และจัดทำรายงาน

## 1.5 ระยะเวลาและแผนปฏิบัติงาน

ตาราง 1.1 ระยะเวลาและแผนปฏิบัติงาน

การดำเนินงาน	เดือน/ปี 2557									
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	
1. ศึกษาข้อมูล	■									
2. เสนอโครงร่าง				■						
3. ออกแบบอุปกรณ์					■					
4. ดำเนินการสร้าง						■				
5. เก็บข้อมูลการทดลอง							■			
6. สรุปผลและทำรายงาน								■		

## 1.6 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1.6.1 เครื่องมือสามารถติดตามวัตถุเป้าหมาย และสามารถแยกสีแยกขนาดแล้วหันตามคำสั่งจากการประมวลผลโดยคำสั่งที่กำหนดไว้ได้

## 1.7 งบประมาณที่ใช้

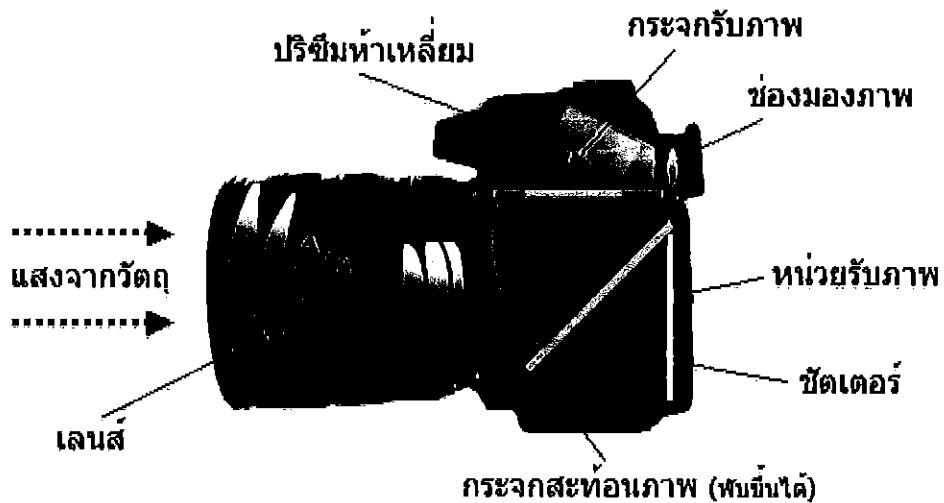
1.7.1 มอเตอร์และอุปกรณ์ทางไฟฟ้า	1000 บาท
1.7.2 ฐานควบคุมกลไกขับเคลื่อนล้อ	500 บาท
1.7.3 ฐานการสร้างรูปแบบตัวอย่างการเคลื่อนที่ของวัตถุ	500 บาท
1.7.4 ค่าเอกสาร และค่าจัดทำรูปเล่ม	1000 บาท
รวมทั้งสิ้น	3000 บาท

## บทที่ 2

### หลักการและทฤษฎี

#### 2.1 ความหมายของกล้องดิจิทัล [1]

กล้องดิจิทัล คือกล้องถ่ายภาพที่ไม่ต้องใช้ฟิล์ม แต่ใช้อุปกรณ์อิเล็กทรอนิกส์ที่เรียกว่า เซ็นเซอร์เป็นตัวรับแสงผ่านการประมวลผลได้ภาพดิจิทัล โดยมีการบันทึกภาพไว้ในรูปแบบไฟล์ใน ส่วนบันทึกภาพภายในกล้อง ซึ่งสามารถส่งต่อไปยังคอมพิวเตอร์ เพื่อตกแต่งภาพด้วยโปรแกรมต่าง ๆ แล้วพิมพ์ออกมาเป็นภาพหรือส่งผ่านอินเทอร์เน็ต และใช้งานในลักษณะอื่น ๆ ได้อย่างหลากหลาย บนพื้นผิวของเซ็นเซอร์รับภาพจะแบ่งออกเป็นส่วนเล็ก ๆ ที่เรียกกันว่า พิกเซล ( Pixel ) ซึ่งแต่ละ พิกเซลจะบันทึกข้อมูล 1 จุดภาพเมื่อรวมกันหลาย ๆ จุดอัดกันแน่นจะกลายเป็นภาพที่สามารถ มองเห็นได้ว่าเป็นภาพอะไร โดยมีภาพรวมการทำงานตามรูปที่ 2.1



รูปที่ 2.1 การทำงานของกล้องดิจิทัล [1]



## 2.2 กระบวนการถ่ายภาพด้วยระบบดิจิทัล

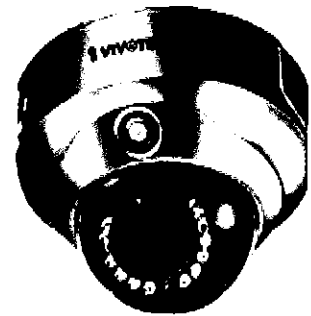
กล้องถ่ายภาพแบบดิจิทัล เมื่อถ่ายภาพแสงจะเข้าไปในกล้องถ่ายภาพ และเข้าไปยังหน่วยประมวลผล ( image sensor ) ก่อน จากนั้นข้อมูลภาพจะถูกแปลงเป็นสัญญาณดิจิทัลไปเก็บที่หน่วยความจำของกล้องถ่ายภาพ เช่น memory card และ hard disc โดยกล้องถ่ายภาพแบบดิจิทัล ตัวอย่างเช่น web camera นี้มีรูปร่างที่หลากหลายแบบดังรูปที่ 2.2



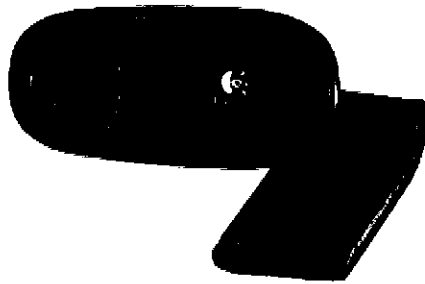
(a)



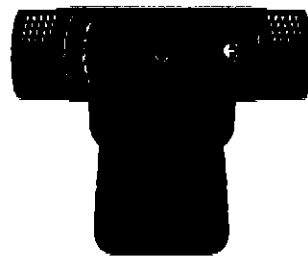
(b)



(c)



(d)



(e)

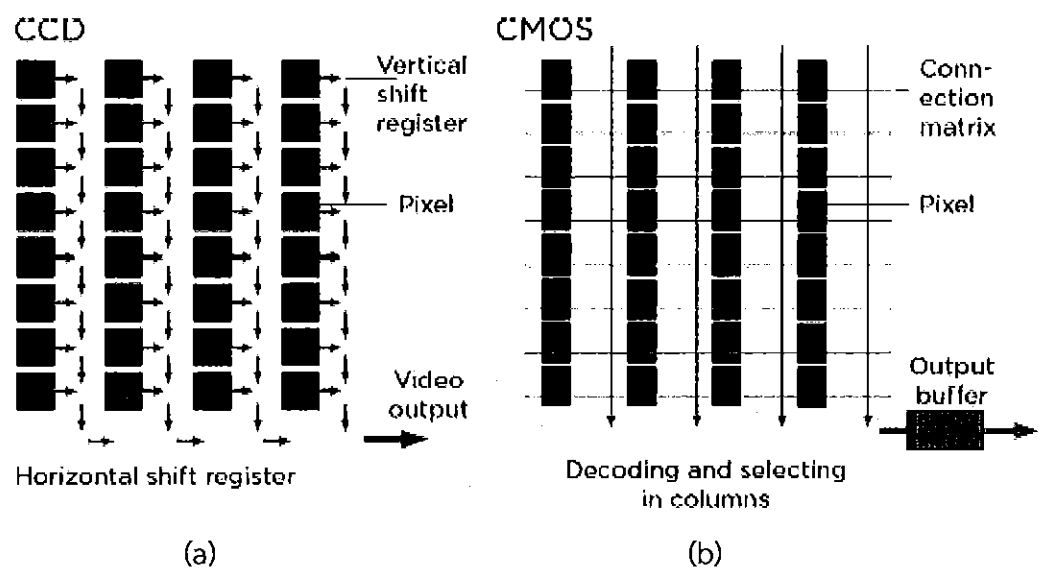


(f)

รูปที่ 2.2 กล้องWeb Camera รูปแบบต่างๆ [2]

## 2.3 เซ็นเซอร์รับภาพในกล้องดิจิทัล [2]

หน่วยประมวลผลของกล้องถ่ายภาพดิจิทัลหรือเรียกว่า “เซนเซอร์รับภาพ” ซึ่งส่วนใหญ่ กล้องดิจิทัลจะมีขนาดเซนเซอร์ที่เล็กกว่าขนาดกล้อง โดยสามารถแบ่งเซนเซอร์รับภาพออกได้ 2 ประเภท คือ CCD (charge couple device) และ CMOS ( complementary metal oxide semiconductor ) ซึ่งมีลักษณะการรับภาพและส่งข้อมูลภาพที่แตกต่างกันดังรูปที่ 2.3



รูปที่ 2.3 การทำงาน Sensor CCD และ CMOS [2]

2.3.1 CCD ( Charge Couple Device ) จากรูปที่ 3.2 (a) เป็น image sensor ที่ทำงานโดยจับโฟตอนของแสงแล้ว แปลงออกมาเป็นจุดพิกเซลของภาพด้วยการทำงานของโฟโตไดโอดที่อยู่บน CCD มีลักษณะเป็นตารางที่เรียงตามหลักการผสมสี มีการส่งผ่านสัญญาณสมำเสมอในรูปแบบอนาล็อก ทำให้คุณภาพของภาพดีแต่ช้า ส่วนใหญ่ CCD มีหลายขนาด เพื่อใช้ในการแปลงข้อมูลภาพ ปัจจุบันมีการพัฒนา CCD เพื่อปรับสีสันของภาพให้ดูสมจริงอย่างไรก็ตาม CCD จะกินไฟมากกว่า แต่ส่วนใหญ่ก็นิยมใช้มากกว่า CMOS

2.3.2 CMOS ( Complementary Metal Oxide Semiconductor ) จากรูปที่ 3.2 (b) ใช้เทคโนโลยีเดียวกับการทำวงจรรีเลย์ทรอนิกส์ในซีพียูของคอมพิวเตอร์ แต่ละส่วนจะเป็นตัวรับแสงและมีส่วนใช้แปลงประจุไฟฟ้าเป็นสัญญาณอนาล็อกด้วย ทำให้พื้นที่รับแสงน้อยลง ความสมำเสมอของสัญญาณไม่ค่อยดีนัก ปัจจุบันมีการพัฒนา CMOS ให้มีคุณภาพดีขึ้น กล้องดิจิทัล บางรุ่นนำ CMOS มาใช้งานเนื่องจากกินไฟน้อยกว่า CCD และราคาประหยัดกว่ารวมถึงการประมวลผลเพื่อการบินที่อีกด้วย

## 2.4 ความละเอียด ( resolution ) [2]

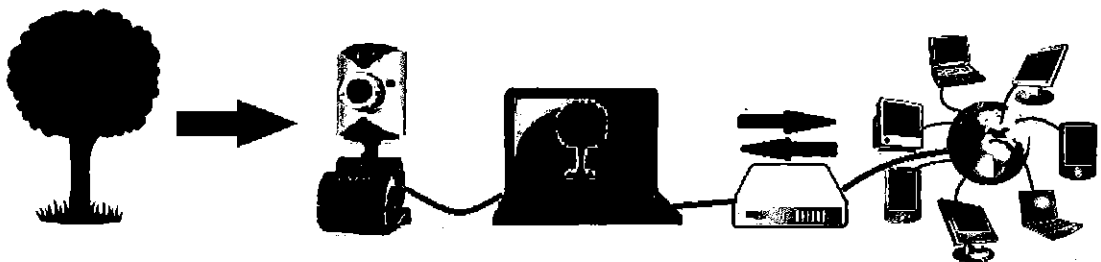
กล้องดิจิทัลที่มีความละเอียดเท่าใดที่เหมาะสมกับการใช้งาน ส่วนใหญ่กล้องดิจิทัลที่จำหน่ายนั้นจะบอกคุณสมบัติความละเอียดมากที่สุดของกล้องนั้นๆ เช่น 7 ล้านพิกเซล 8 ล้านพิกเซล ซึ่งคำนวณจากจำนวน Pixel ของภาพในแนวตั้งและแนวนอนคูณกันเป็นจำนวนพิกเซลทั้งหมด ได้เท่าไรและนำ 1 ล้านพิกเซลมาหาร (1 ล้านพิกเซลเท่ากับ 1 เมกกะพิกเซล) จะได้ความละเอียดของภาพเป็นหน่วยล้านพิกเซล ตามตารางที่ 2.1

ตารางที่ 2.1 ค่าความละเอียด Pixels [2]

Image recording Quality		Pixels	Mega pixels	Print Size
Large	Fine	3504x2336	8.2	A3 or Larger
Medium	Fine	2544x1696	4.3	A5-A4
Small	Fine	1728x1152	2	A5-smaller

## 2.5 กล้องเว็บแคม ( Webcam )

เว็บแคม ( Webcam ) หรือเรียกเต็มๆ ว่า Web Camera แต่ในบางครั้งก็มีคนเรียกว่า Video Camera หรือ Video Conference เป็นอุปกรณ์อินพุตที่สามารถจับภาพเคลื่อนไหวของวัตถุไปปรากฏในหน้าจอคอมพิวเตอร์ และสามารถส่งภาพเคลื่อนไหวนี้ผ่านระบบเครือข่ายเพื่อให้คนอื่นอีกฟากหนึ่งสามารถเห็นตัวเคลื่อนไหวได้เหมือนอยู่ต่อหน้า ถือว่าเป็นอุปกรณ์ที่มีประโยชน์อีกตัวหนึ่ง และเริ่มมีความจำเป็นมากขึ้นเรื่อย ๆ ในปัจจุบันใช้ เทคโนโลยี CMOS ในการผลิตเซ็นเซอร์รับภาพในกล้อง เนื่องจากความคุ้มค่าในการผลิตและคุณภาพที่ดีจากความก้าวหน้าในเทคโนโลยีการผลิต ในการใช้เพื่อการสื่อสารมีภาพรวมการทำงาน ตามรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างการใช้กล้อง Web Camera

## 2.6 ระบบสี [3]

โดยทั่วไปสีในธรรมชาติและสีที่สร้างขึ้น จะมีรูปแบบการมองเห็นของสีที่แตกต่างกัน ซึ่งรูปแบบการมองเห็นสี ที่ใช้ในงานด้านกราฟิกทั่วไปนั้นมียู่ด้วยกัน 4 ระบบคือ

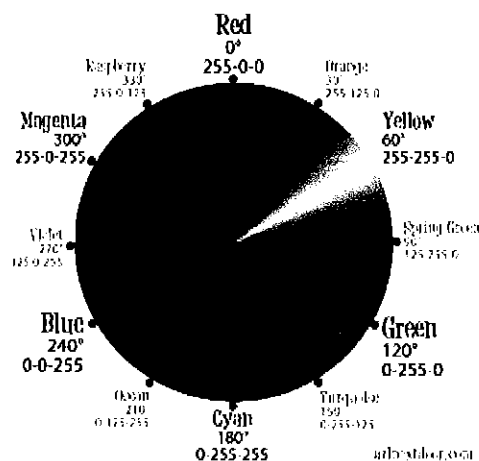
- 2.6.1 ระบบสีแบบ RGB ตามหลักการแสดงสีของเครื่องคอมพิวเตอร์
- 2.6.2 ระบบสีแบบ CMYK ตามหลักการแสดงสีของเครื่องพิมพ์
- 2.6.3 ระบบสีแบบ HSB หรือ HSL ตามหลักการมองเห็นสีของสายตามนุษย์
- 2.6.4 ระบบสีแบบ Lab ตามมาตรฐานของ CIE ซึ่งไม่ขึ้นอยู่กับอุปกรณ์ใดๆ

### 2.6.1 ระบบสีแบบ RGB

เป็นระบบสีที่ประกอบด้วยแม่สี 3 สีคือ แดง ( Red ), เขียว ( Green ) น้ำเงิน ( Blue) ในสัดส่วนความเข้มข้นที่แตกต่างกันเมื่อนำมาผสมกันทำให้เกิดสีต่างๆ บนจอคอมพิวเตอร์ได้มากถึง 16.7 ล้านสี ซึ่งใกล้เคียงกับสีที่ตามองเห็นได้โดยปกติ และจุดที่สีทั้งสามสีรวมกันจะกลายเป็นสีขาว นิยมเรียกการผสมสีแบบนี้ว่าแบบ “Additive” หรือการผสมสีแบบบวก ซึ่งเป็นการผสมสีขั้นที่ 1 หรือ ถ้านำเอา Red Green Blue มาผสมครั้งละ 2 สี ก็จะทำให้เกิดสีใหม่ เช่น

Blue + Green = Cyan  
 Red + Blue = Magenta  
 Red + Green = Yellow

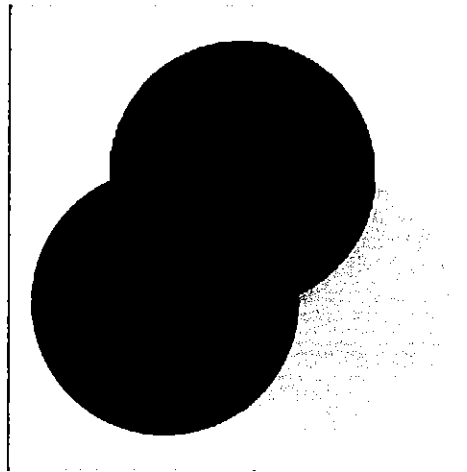
แสงสี RGB มักจะถูกใช้สำหรับการส่องสว่างทั้งบนจอทีวีและจอคอมพิวเตอร์ ซึ่งสร้างจากการให้กำเนิดแสงสีแดง สีเขียว และสีน้ำเงิน ทำให้สีดูสว่างกว่าความเป็นจริงมีระบบสีตามรูปที่ 2.5



รูปที่ 2.5 ระบบสีแบบ RGB [3]

### 2.6.2 ระบบสีแบบ CMYK [3]

เป็นระบบสีที่ใช้กับเครื่องพิมพ์ที่พิมพ์ออกทางกระดาษ ซึ่งประกอบด้วยสีพื้นฐาน คือ สีฟ้า (Cyan), สีม่วงแดง (Magenta), สีเหลือง (Yellow), และเมื่อนำสีทั้ง 3 สีมาผสมกันจะเกิดสีเป็น สีดำ (Black) แต่จะไม่ดำสนิทเนื่องจากหมึกพิมพ์มีความไม่บริสุทธิ์ โดยเรียกการผสมสีทั้ง 3 สีข้างต้นว่า “Subtractive Color” หรือการผสมสีแบบลบ หลักการเกิดสีของระบบนี้คือ หมึกสีหนึ่ง จะดูดกลืนสีจากสีหนึ่งแล้วสะท้อนกลับออกมาเป็นสีต่างๆ เช่น สีฟ้าดูดกลืนสีม่วงแล้วสะท้อนออกมาเป็นสีน้ำเงิน ซึ่งจะสังเกตได้ว่าสีที่สะท้อนออกมาจะเป็นสีหลักของระบบ RGB การเกิดสีนี้ในระบบนี้จึงตรงข้ามกับการเกิดสีในระบบ RGB มีระบบสีตามรูปที่ 2.6



รูปที่ 2.6ระบบสี CMYK [3]

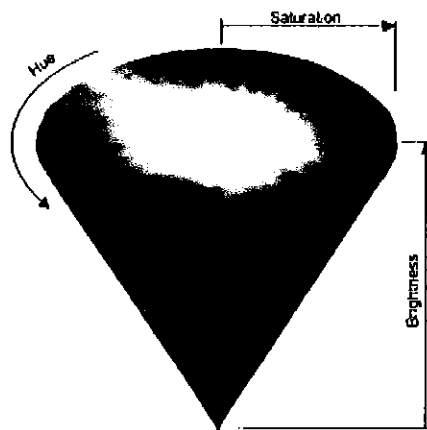
### 2.6.3 ระบบสีแบบ HSB หรือ HSL [3]

เป็นระบบสีพื้นฐานในการมองเห็นสีด้วยสายตาของมนุษย์มีระบบสีตามรูปที่ 2.7 ประกอบด้วยลักษณะของสี 3 ลักษณะ คือ

1. Hue คือ สีต่างๆ ที่สะท้อนออกมาจากวัตถุเข้ามายังตาของ ทำให้สามารถมองเห็นวัตถุเป็นสีต่างๆ ได้ ซึ่งแต่ละสีจะแตกต่างกันตามความยาวของคลื่นแสงที่มากกระทบวัตถุและสะท้อนกลับ ที่ตาของ Hue ถูกวัดโดยตำแหน่งการแสดงสีบน Standard Color Wheel ซึ่งถูกแทนด้วยองศา 0 ถึง 360 องศา แต่โดยทั่วๆ ไปแล้วมักจะเรียกการแสดงสีนั้นๆ เป็นชื่อของสีเลย เช่น สีแดง สีม่วง สีเหลือง

2. Saturation คือ ความสดของสี โดยค่าความสดของสีจะเริ่มที่ 0 ถึง 100 ถ้ากำหนด Saturation ที่ 0 สีจะมีความสดน้อย แต่ถ้ากำหนดที่ 100 สีจะมีความสดมาก ถ้าถูกวัดโดยตำแหน่งบน Standard Color Wheel ค่า Saturation จะเพิ่มขึ้นจากจุดกึ่งกลางจนถึงเส้นขอบ โดยค่าที่เส้นขอบจะมีสีที่ชัดเจนและอิ่มตัวที่สุด

3. Brightness หรือ Luminance คือ ระดับความสว่างและความมืดของสี โดยค่าความสว่างของสีจะเริ่มที่ 0 ถึง 100 ถ้ากำหนดที่ 0 ความสว่างจะน้อยซึ่งจะเป็นสีดำ แต่ถ้ากำหนดที่ 100 สีจะมีความสว่างมากที่สุด ยังมีค่า Brightness มากจะทำให้สีนั้นสว่างมากขึ้น



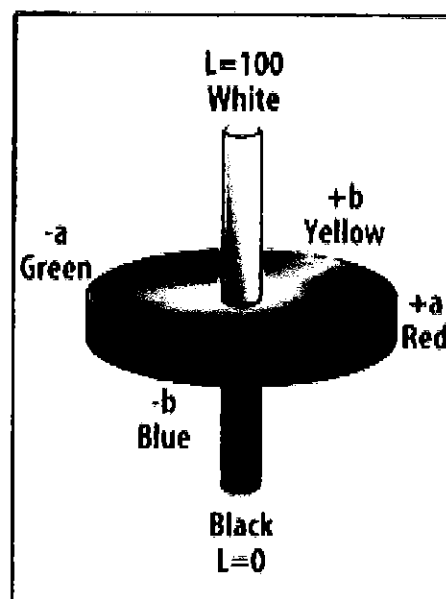
รูปที่ 2.7 ระบบสีแบบ HSB หรือ HSL [3]

### 2.6.4 ระบบสีแบบ Lab [3]

ระบบสีแบบ Lab เป็นค่าสีที่ถูกกำหนดขึ้นโดย CIE ( Commission International d' Eclairage ) จากความพยายามในการให้เป็นสีมาตรฐานกลางของการวัดสีทุกรูปแบบ ครอบคลุมทุกสีใน RGB และ CMYK และใช้ได้กับสีที่เกิดจากอุปกรณ์ทุกอย่างไม่ว่าจะเป็นจอคอมพิวเตอร์ เครื่องพิมพ์ เครื่องสแกนและอื่นๆ ส่วนประกอบของโหมดสีนี้ได้แก่ L หรือ Luminance เป็นการกำหนดความสว่างซึ่งมีค่าตั้งแต่ 0 ถึง 100 ถ้ากำหนดที่ 0 จะกลายเป็นสีดำ แต่ถ้ากำหนดที่ 100 จะกลายเป็นสีขาว

A เป็นค่าของสีที่ไล่จากสีเขียวไปสีแดง

B เป็นค่าของสีที่ไล่จากสีน้ำเงินไปสีเหลือง

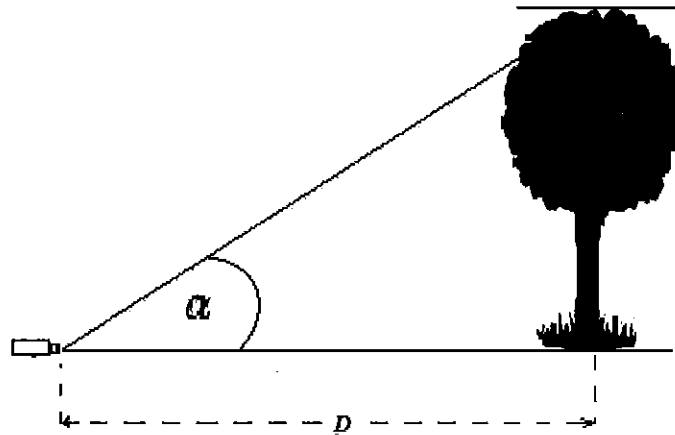


*Lab model*

รูปที่ 2.8 ระบบสี Lab [3]

## 2.7 หลักการหาค่าต่างระดับโดยวิธีตรีโกณมิติ

งานวัดขนาดความสูงวัตถุด้วยตรีโกณมิติ เป็นการวัดระดับโดยอาศัยการวัดมุมตั้งและระยะทางราบแล้วคำนวณหาค่าต่างระดับด้วยสูตรทางตรีโกณมิติเครื่องมือที่สำคัญคือกล้องที่สามารถวัดมุมและเครื่องมือวัดระยะทาง ดังแสดงในรูปที่ 2.9 ซึ่งความสูงของต้นไม้จะเป็นฟังก์ชันของระยะทางและมุมดังแสดงในสมการที่ (1)



รูปที่ 2.9 การวัดความสูงโดยวิธีตรีโกณมิติ

$$\text{ความสูงต้นไม้} = D \tan(\alpha) \quad (1)$$

เมื่อ  $D$  คือระยะห่างจากกล้องและต้นไม้

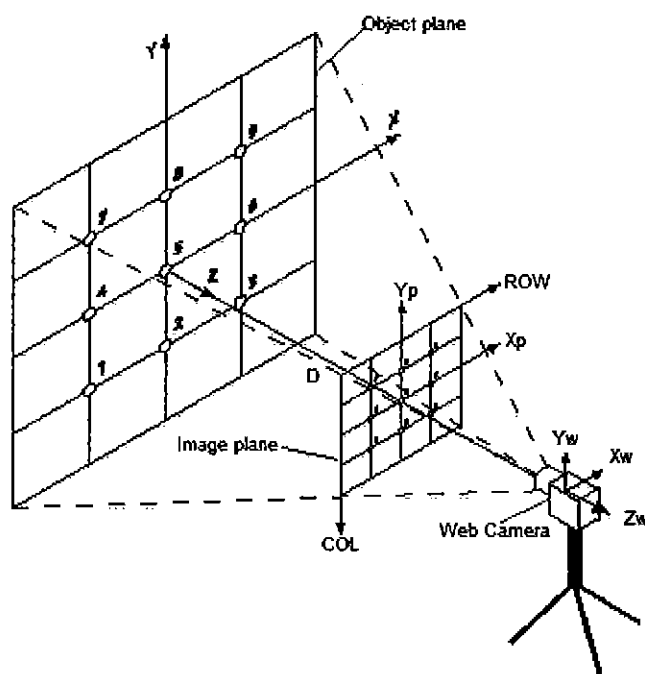
$\alpha$  คือมุมที่ทำกับเส้นขนานกับพื้น และเส้นจากกล้องถึงความสูงของต้นไม้

จากสมการที่ (1) จะเห็นว่าความสูงของวัตถุสามารถคำนวณจากมุม ( $\alpha$ ) ที่ทำกับระนาบอ้างอิง ( $D$ )



## 2.8 หลักการทำงานของระบบและการระบุพิกัด

ระบบติดตามวัตถุและการระบุพิกัดในงานวิจัยแสดงภาพรวมในรูปที่ 2.10 ประกอบด้วย กล้อง Web cam ซึ่งติดตั้งอยู่บนฐานวาง โดยจะมีระบบหมุนทิศทางของกล้องด้วย Step motor จำนวน 1 แกนหมุนรอบแกน  $Y_w$  เพื่อให้กล้องหันหน้าไปที่วัตถุที่สนใจที่มุมต่างๆได้ จะเห็นได้ว่า วัตถุจะอยู่บนระนาบวัตถุ ( Object plane ) ซึ่งถูกกำหนดด้วยพิกัด XYZ โดยแกน Z ของวัตถุจะอยู่ในแนวเดียวกับแกน  $Z_w$  ของ พิกัดกล้อง ภาพถ่ายที่กล้องตรวจจับได้จะถูกแสดงผลบนภาพด้วยพิกัดแบบพิกเซล ( ROW & COLUMN ) โดยในที่นี้กล้องที่ใช้มีความละเอียด 640 ( ROW ) x 480 ( COL ) ซึ่งในงานวิจัยนี้มีการกำหนดพิกัดแบบพิกเซลเพิ่มเติมคือ  $X_p, Y_p$  ซึ่งเป็นพิกัดที่อ้างอิงจุดกำเนิดที่กลางภาพ โดยเมื่อกล้องมีการหมุนรอบแกน  $Y_w$  จะทำให้แกน Z และ  $Z_w$  ทำมุมระหว่างกันเป็นมุม  $\theta_m$  โดยเมื่อมองรูปที่ 2.10 จากมุมมองด้านบน จะแสดงดังรูปที่ 2.11 ซึ่งแสดงพิกัดระบบในระนาบ X-Z จะทำให้สามารถคำนวณตำแหน่งวัตถุ ในแนวแกน X ได้ สำหรับ ตำแหน่งวัตถุในแนวแกน Y นั้น การคำนวณจะต้องอาศัยมุมมองในระนาบ Y-Z ดังแสดงในรูปที่ 2.12



รูปที่ 2.10 ภาพรวมระบบติดตามวัตถุและการระบุพิกัด

โดย  $D$  คือ ค่าระยะห่างจาก Web Camera ตั้งฉากกับ Object Plane

$X$  คือ ค่าพิกัดในแนวแกน X ของ Object Plane

$Y$  คือ ค่าพิกัดในแนวแกน X ของ Object Plane

$Z$  คือ ค่าพิกัดในแนวแกน X ของ Object Plane

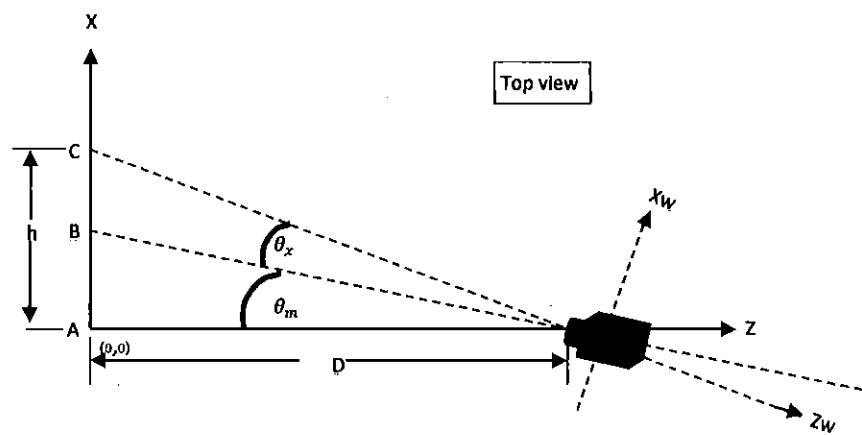
$X_p$  คือ ค่าพิกัดในแนวแกน X ของ Image Plane

$Y_p$  คือ ค่าพิกัดในแนวแกน Y ของ Image Plane

$X_w$  คือ ค่าพิกัดในแนวแกน X ของ Web Camera

$Y_w$  คือ ค่าพิกัดในแนวแกน Y ของ Web Camera

$Z_w$  คือ ค่าพิกัดในแนวแกน Z ของ Web Camera



รูปที่ 2.11 มุมมอง top view ของ Object Plane เมื่อทำการหันกล้อง

จากรูปที่ 2.11 แสดงให้เห็นว่าเมื่อกำลังหมุนหันหน้าไปจากจุดตั้งฉากกับ Object Plane หรือจุด A ไปอยู่ตรงแนวกับจุด B จะทำให้ จุดศูนย์กลางของภาพใน Image Plane เปลี่ยนแปลงด้วย โดยหากต้องการหาพิกัดตำแหน่งของจุด C จะสามารถหาได้จากค่าระยะ  $h$  ซึ่งคือระยะห่างจาก A ถึง C ดังแสดงในสมการที่ (2)

$$h = D \tan(\theta_m + \theta_x) \quad (2)$$

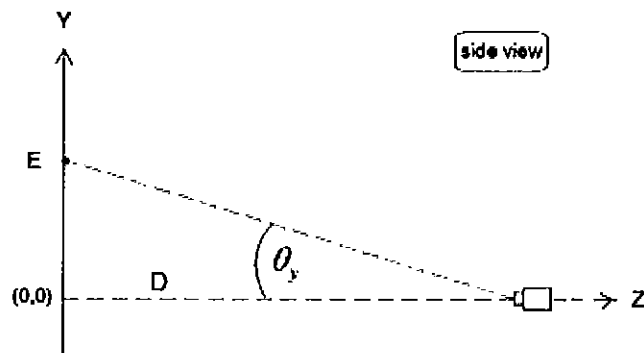
โดย

$\theta_m$  คือ มุมที่กล้องหมุนไปจากแกนอ้างอิงไปแล้ว

$\theta_x$  คือ มุมจากแนวกลางของกล้องถึงเป้าหมายที่จุด C

ได้ ระยะห่างระหว่างกล้องกับ B หาได้จาก  $\sqrt{(D \tan \theta_m)^2 + (D)^2}$

และ ระยะห่างระหว่างกล้องกับ C หาได้จาก  $\sqrt{(D)^2 + (h)^2}$

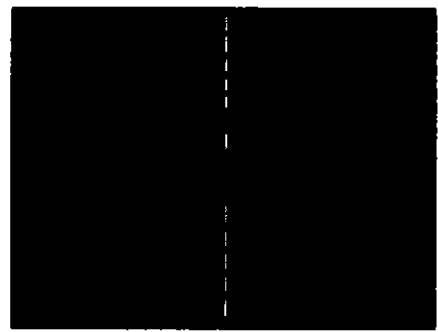
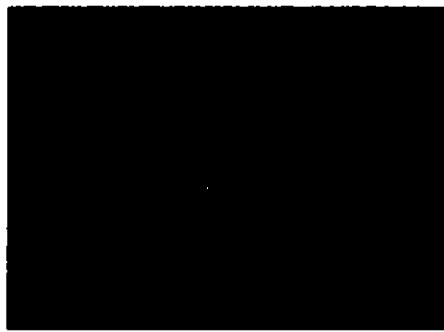


รูปที่ 2.12 มุมมอง side view ของกล้องตั้งฉากกับ Object Plane

จากรูปที่ 2.12 เป็นกรณีที่ไม่มีกั้นกล้อง หรือแนวกลางกล้องตรงกับแนวแกน Z จึงสามารถคำนวณระยะห่าง E จากจุด (0,0) ในระนาบแกน Y ดังนี้

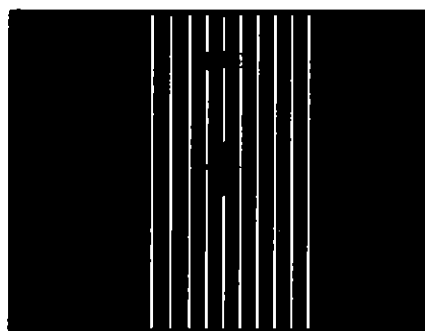
$$\text{ระยะห่างจาก (0,0) ถึง E} = D \tan \theta_y$$

สำหรับหลักการเขียนโปรแกรมเพื่อให้กล้องเคลื่อนที่ตามวัตถุนั้นทำได้โดยการนำภาพของวัตถุมาคำนวณเพื่อหาตำแหน่งที่ห่างจาก Center of Picture ซึ่งคือจุดกากบาท (+) ดังแสดงในรูปที่ 2.13 และ 2.14 ซึ่งแสดงภาพที่ถูกตรวจจับบน Image Plane ซึ่งโดยหลักการแล้วเมื่อระบบตรวจจับได้ว่าวัตถุอยู่เยื้องไปจากแนวแกนกลางดังรูปที่ 2.13 แล้ว ระบบจะสั่งการให้ Step Motor ทำการหมุนกล้องให้หันเข้าหาวัตถุเป็นผลให้ จุดวัตถุในภาพเคลื่อนเข้าสู่ระนาบแกน  $x = 0$  โดยนำระยะห่างคาดเคลื่อนจากจุดกลางจอ “+” (กึ่งกลางของกล้อง) แสดงในรูปที่ 2.13 มาป้อนกลับ เพื่อทำการสั่งปรับการหมุนมอเตอร์จนได้ ดังแสดงในรูปที่ 2.14



รูปที่ 2.13 หน้าจอก่อนปรับตำแหน่งวัตถุที่จุด (2,2)      รูปที่ 2.14 หน้าจอหลังปรับเข้าสู่ (0,2)

แต่การทำงานโดยใช้สเตปมอเตอร์ในงานจริงจะไม่สามารถหมุนให้ภาพวัตถุอยู่ตรงแนวกลางของกล้องพอดีได้ ตามรูปที่ 2.15 ทั้งนี้เนื่องจากข้อจำกัดด้านความละเอียดของสเตปมอเตอร์ที่ใช้ ซึ่งมีความละเอียดสเตปละ 1.8 องศา เพื่อให้การทำงานของระบบมีความนิ่ง ไม่สั่นไหวเกินไปในการติดตามวัตถุ ในการโปรแกรมจะกำหนดขอบเขตที่จะหยุดการหมุนกล้อง เมื่อวัตถุอยู่ในขอบเขตที่เหมาะสม เช่นห่างกับแนวแกนกลางไม่เกิน กี่พิทเชล ซึ่งค่าขอบเขตที่เหมาะสมก็ขึ้นกับความเร็วที่มอเตอร์หมุนด้วยเช่นกัน โดยรูปที่ 2.15 จะมีเส้นสีขาวแสดงตัวอย่างแนวกลางของภาพเมื่อมีการหมุนสเตปมอเตอร์ไปตำแหน่งต่างๆ ซึ่งระยะห่างระหว่างเส้นขาวแสดงความต่างของมุมหมุนของสเตปมอเตอร์ 1 สเตป



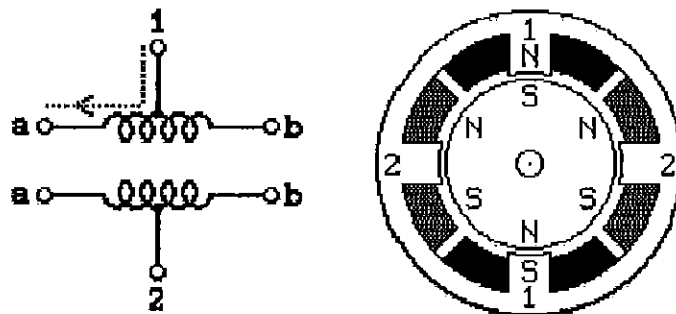
รูปที่ 2.15 หน้าจอตำแหน่งวัตถุซึ่งเยื้องจากแนวกลางเล็กน้อย

## 2.9 สเต็ปป์มอเตอร์ ( Stepping Motor ) [4]

สเต็ปป์มอเตอร์หรือ สเต็ปมอเตอร์เป็นอุปกรณ์เอาต์พุตอย่างหนึ่ง ซึ่งสามารถนำไอซี ไมโครคอนโทรลเลอร์ มาทำการควบคุมได้สะดวก และเป็นมอเตอร์ที่เหมาะสมสำหรับใช้ในงาน ควบคุมการหมุน ที่ต้องการตำแหน่ง และทิศทางที่แน่นอน การทำงานของ สเต็ปป์มอเตอร์จะ ขับเคลื่อนทีละขั้นๆ ละ ( 4 Step ) 0.9, 1.8, 5, 7.5, 15 หรือ 50 องศา ซึ่งขึ้นอยู่กับคุณสมบัติแต่ละ ชนิดของ สเต็ปป์มอเตอร์ตัวนั้นๆ สเต็ปป์มอเตอร์จะแตกต่างจากมอเตอร์กระแสตรงทั่วไป ( DC MOTOR ) โดยการทำงานของมอเตอร์กระแสตรงจะหมุนไปแบบต่อเนื่อง ไม่สามารถหมุนเป็นแบบ สเต็ปๆ ได้ดังนั้นในการนำไปกำหนดตำแหน่งจึงควบคุมได้ยากกว่า แต่ในส่วนใหญ่จะใช้ สเต็ปป์ มอเตอร์มาทำการควบคุมโดยใช้วิธีในระบบดิจิทัล เช่น พรินเตอร์ ( Printer ) พล็อตเตอร์ ( X-Y Plotter ) ดิสก์ไดรฟ์ ( Disk drive )

### 2.9.1 Unipolar Step Motor

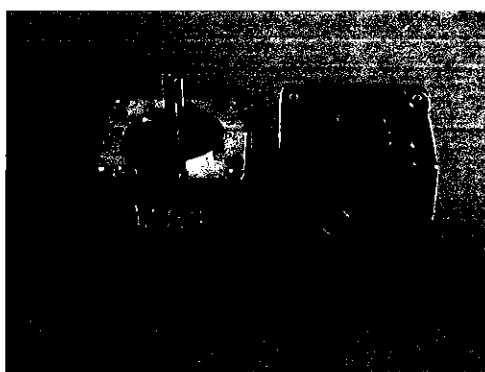
Unipolar สเต็ปมอเตอร์ มีทั้งแบบสเต็ปมอเตอร์แบบแม่เหล็กถาวรและสเต็ปมอเตอร์แบบผสม มอเตอร์ แบบนี้มีสายไฟออกมา 5 หรือ 6 เส้น ดังเช่น มอเตอร์แบบที่มี center tap ของขดลวดจะต่อกับไฟชั่วคราว และปลายทั้ง 2 ด้านของขดลวดแต่ละขดจะถูก ต่อลง ground เพื่อใช้กลับทิศทางของการหมุน และในกรณีที่ต้องการให้ step การหมุนของ มอเตอร์มีความละเอียดขึ้น จะต้องเลือกใช้โรเตอร์ที่มีขั้วมากขึ้น สเต็ปมอเตอร์ส่วนใหญ่จะ หมุนด้วยมุม 30 องศาต่อขั้น แต่มีสเต็ปมอเตอร์บางรุ่นสามารถหมุนด้วยมุม 15 หรือ 7.5 องศาต่อขั้น สำหรับสเต็ปมอเตอร์แบบแม่เหล็กถาวร จะมีความละเอียดของมุมที่ดีที่สุดที่ 1.8 องศา ต่อขั้น และสเต็ปมอเตอร์แบบผสมส่วนใหญ่จะมีความละเอียดของขั้น การหมุนเป็น 3.6 และ 1.8 องศาต่อขั้นและมีความละเอียดในการหมุนสูงสุดเท่าที่มีการผลิตขึ้นมาเป็น 0.72 องศาต่อขั้น มีโครงสร้างดังรูป 2.16



รูปที่ 2.16 Unipolar Step Motor [4]

## 2.9.2 ข้อดีของสเตปป์มอเตอร์เมื่อเปรียบเทียบกับมอเตอร์กระแสตรง

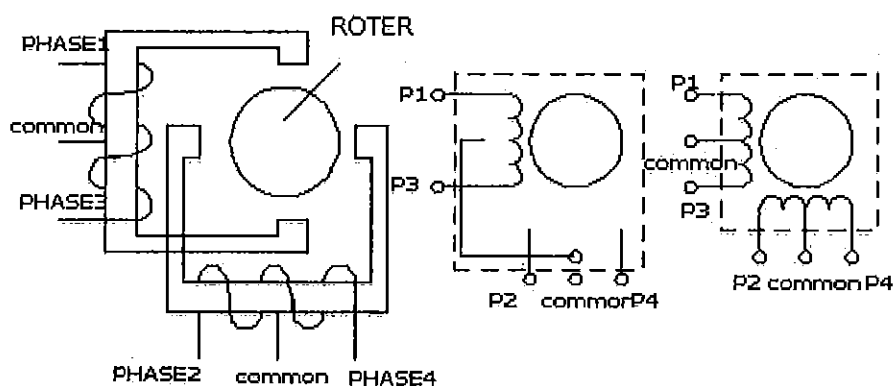
1. สามารถใช้ในงานควบคุมตำแหน่งในลักษณะวงจรถวลแบบเปิด ( Open Loop Control ) ได้ โดยที่ไม่ต้องการสัญญาณป้อนกลับ ( Feed Back signal ) แต่อาศัยการนับจำนวนของพัลส์ ( Pulse ) ที่ส่งไปควบคุมการหมุนแทน
2. ไม่มีส่วนของแปรงถ่านที่จะสึกหรอและไม่เกิดการสปาร์คที่แปรงถ่านซึ่งอาจก่อให้เกิดสัญญาณรบกวน ตัวอย่าง Step Motor ตามรูปที่ 2.17



รูปที่ 2.17 สเตปมอเตอร์

สเตปป์มอเตอร์ที่จะนำมาใช้ในการทดลองนี้ จะใช้สเตปป์แบบยูนิโพลาร์ ( Uni-Polar stepper motor ) ซึ่งโครงสร้างของสเตปป์มอเตอร์แบบนี้จะมี ส่วนประกอบที่สำคัญ 2 ส่วน ด้วยกันคือ

- 1 ส่วนที่ทำการหมุน ( Rotor ) จะเป็นแม่เหล็กถาวรหรืออื่นๆ ตามรูปที่ 2.18
- 2 ส่วนที่อยู่กับที่ ( Stator ) เป็นขดลวดที่พันไว้จำนวนหลายๆขด



รูปที่ 2.18 สเตปมอเตอร์ 4 เฟส แบบยูนิโพลาร์ [4]

### 2.9.3 วิธีการขับเคลื่อนมอเตอร์ให้หมุนโดยการกระตุ้นเฟส [4]

ในการควบคุมสเตปมอเตอร์เพื่อที่จะให้ทำการหมุน มีวิธีการควบคุมกระแสไฟที่จ่ายให้กับขดลวดสเตเตอร์ ( Stator ) ในแต่ละเฟสของสเตปมอเตอร์ อย่างเป็นลำดับที่แน่นอน โดยถ้าหากต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆเป็นสถานะลอจิก "1" และในการกระตุ้นเฟสของสเตปมออยู๋ด้วยกัน 2 แบบคือ

ก.การกระตุ้นเฟส แบบฟูลสเตปมอเตอร์ ( Full Step Motor ) ยังสามารถแบ่งการกระตุ้นเฟสออกได้เป็นอีก 2 วิธีด้วยกันคือ

#### 1. การกระตุ้นเฟสแบบฟูลสเตปแบบ 1 เฟส ( Single-Phase Driver )

ตามตารางที่ 2.2 จะเป็นการป้อนกระแสไฟให้กับขดลวด ของสเตปมอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไป ดังนั้นกระแส ที่ไหลในขดลวด จะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเตปมอเตอร์มีน้อย

ตารางที่ 2.2 Single-Phase Driver

สเตปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

#### 2. การกระตุ้นเฟสแบบฟูลสเตปแบบ 2 เฟส ( Two-Phase Driver )

แสดงดังตารางที่ 2.3 เป็นการป้อนกระแสให้กับขดลวด 2 ขด ของสเตปมอเตอร์พร้อมๆกันไป และจะกระตุ้นเรียงถัดกันไปเช่นเดียวกับแบบหนึ่งเฟส ดังนั้นการกระตุ้นแบบนี้จึงต้องใช้กำลังไฟมากขึ้น และจะทำให้มีแรงบิดของมอเตอร์มากกว่าการกระตุ้นแบบ 1 เฟส

ตารางที่ 2.3 Two-Phase Driver

สเตปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

ข. การกระตุ้นเฟส แบบฮาล์ฟสเตป ( Half Step Motor ) หรือ one-two phase Driver คือการกระตุ้นเฟสแบบ ฟูลสเตป 1 เฟส และ 2 เฟส เรียงลำดับกันไป แสดงดังตารางที่ 2.4 แรงบิดที่ได้จากการกระตุ้นเฟสแบบนี้จะมีเพิ่มมากขึ้น เพราะช่วงของสเตปมีระยะสั้นลง ในการกระตุ้นแบบนี้จะต้องมีการกระตุ้นที่เฟสถึง 2 ครั้ง จึงจะได้ระยะของ สเตป เท่ากับการกระตุ้นเพียงครั้งเดียวของแบบฟูลสเตป 2 แบบแรก ความละเอียดของการหมุน ตำแหน่งองศาต่อ สเตปก็เป็นสองเท่าของแบบแรก ความถูกต้องของตำแหน่งที่กำหนดจึงมีมากขึ้น

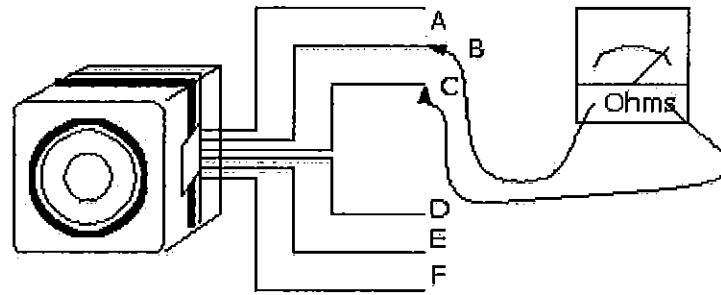
ตารางที่ 2.4 Half Step Motor

สเตปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

#### 2.9.4 วิธีการตรวจสอบหาเฟสของขดลวดสเตปป์มอเตอร์

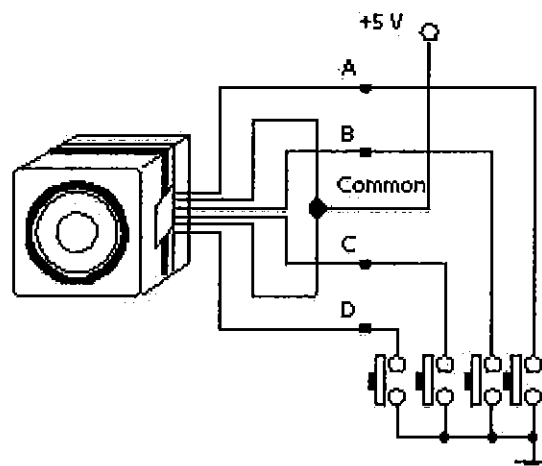
1. ให้สังเกตว่าสเตปป์มอเตอร์ที่นำมาทดลองที่เป็นแบบยูนิโพลาร์ ( Uni.-Polar stepper motor ) จะมีจำนวนสาย 5 เส้นหรือ 6 เส้น
2. ใช้มิเตอร์วัดค่าความต้านทานของเส้นลวดในแต่ละขดดังรูปที่ 2.19 ขั้นตอนการวัด ให้หาสายที่ต่อเป็นจุดร่วมเสียก่อน ( common ) โดยให้ใช้ มัลติเตอร์ตั้งค่าไว้สำหรับการวัดค่าความต้านทานแต่ละเส้นสังเกตที่ค่าความต้านทานถ้าหากไม่ได้ วัดระหว่าง จุดต่อร่วม ( common ) กับสายแต่ละเส้น ค่าความต้านทานจะมีค่าเป็น 2 เท่าของการวัดระหว่างจุดต่อร่วมกับสายที่ใช้งาน ตัวอย่างเช่น ถ้าให้จุด B เป็นจุดร่วม หากวัดระหว่างที่จุด A กับจุด B จะมีค่าเท่ากับ 60 Ohm แต่ถ้าวัดระหว่างที่จุด A และจุด C ซึ่งไม่ใช่จุดร่วมก็จะได้ค่าเท่ากับ 120 Ohm หากเป็นแบบที่มีสาย 6 เส้นก็จะมีจุดร่วมสองจุด เพราะมีขดลวดคนละชุดกัน และสายที่เป็นจุดร่วมส่วนใหญ่จะมีสีเหมือนกัน ทำนองเดียวกันหากเป็นแบบที่มีสาย 5 เส้นก็จะมีจุดร่วมเพียงจุดเดียวเท่านั้น





รูปที่ 2.19 การใช้มิเตอร์วัดค่าความต้านทาน Step motor [4]

3. หากเป็นแบบที่มีสาย 6 เส้นก็ให้ทำการต่อจุดรวมเข้าด้วยกันจะได้เป็น 5 เส้น แล้วต่อวงจรตาม รูปที่ 2.20 หลังจากนั้นให้ทดสอบกดสวิตช์ ที่ต่อเข้ากับแต่ละจุดโดยเริ่มที่จุด A จุด B จุด C และจุด D แล้วให้สังเกตการหมุนของสเตปป์มอเตอร์ว่าหมุนได้ต่อเนื่องหรือไม่ หากมีการกระโดด ช้ามสเตปก็ให้ทดสอบโดยเรียงลำดับการกดสวิตช์ใหม่จนหาลำดับของสายได้ถูกต้องคือมอเตอร์เดินตามที่ละสเตป อย่างเป็นลำดับ



รูปที่ 2.20 การต่อวงจรเพื่อทดสอบโดยการสวิตช์เพื่อหาลำดับสาย [4]

## 2.10 ไมโครคอนโทรลเลอร์ ( Microcontroller ) [5]

เป็นอุปกรณ์ไอซี ( IC: Integrated Circuit ) ที่สามารถโปรแกรมการทำงานได้ซับซ้อน สามารถรับข้อมูลในรูปแบบสัญญาณดิจิทัลเข้าไปทำการประมวลผล แล้วส่งผลลัพธ์ข้อมูลดิจิทัลออกมา เพื่อนำไปใช้งานตามที่ต้องการได้ ไมโครคอนโทรลเลอร์ภายในชิพจะมีหน่วยความจำ, Port อยู่ในชิพ เพียงตัวเดียวซึ่งอาจจะเรียกได้ว่าเป็นคอมพิวเตอร์ชิพเดี่ยว หรือไมโครคอนโทรลเลอร์ ซึ่งเป็น ไมโครโปรเซสเซอร์ชนิดหนึ่ง เช่นเดียวกับหน่วยประมวลผลกลาง ( CPU: Central Processing Unit ) ที่ใช้ในคอมพิวเตอร์ แต่ได้รับการพัฒนาแยกออกมาภายหลังเพื่อนำไปใช้ในวงจรทางด้านงานควบคุม คือ แทนที่ในการใช้งานจะต้องต่อวงจรภายนอกต่าง ๆ เพิ่มเติมเช่นเดียวกับไมโครโปรเซสเซอร์ ก็จะทำให้การรวมวงจรที่จำเป็น เช่น หน่วยความจำ, ส่วนอินพุท/เอาต์พุท บางส่วนเข้าไปในตัว ไอซีเดียวกัน และเพิ่มวงจรบางอย่างเข้าไปด้วยเพื่อให้มีความสามารถเหมาะสมกับการใช้ในงานควบคุม เช่น วงจรตั้งเวลา, วงจรการสื่อสารอนุกรม วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล เป็นต้น สรุปคือ

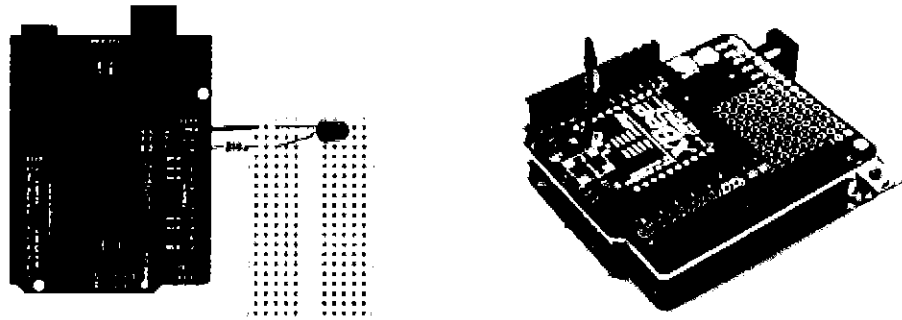
$$\text{Microcontroller} = \text{Microprocessor} + \text{Memory} + \text{I/O}$$

เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้งานควบคุมหลายประการ เช่น

1. ชิพไอซีและระบบที่ได้มีขนาดเล็ก
2. ระบบที่ได้มีราคาถูกกว่าการใช้ชิพไมโครโปรเซสเซอร์
3. วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในการต่อวงจร
4. มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุมโดยเฉพาะซึ่งใช้งานได้ง่าย
5. ช่วยลดระยะเวลาในการพัฒนาระบบได้

## 2.11 ไมโครคอนโทรลเลอร์ Arduino [5]

Arduino อ่านว่า (อา-ดู-อี-โน้ หรือ อาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรรีเลย์ทรานซิสเตอร์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ ( ดูตัวอย่างรูปที่ 2.21 ) เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย



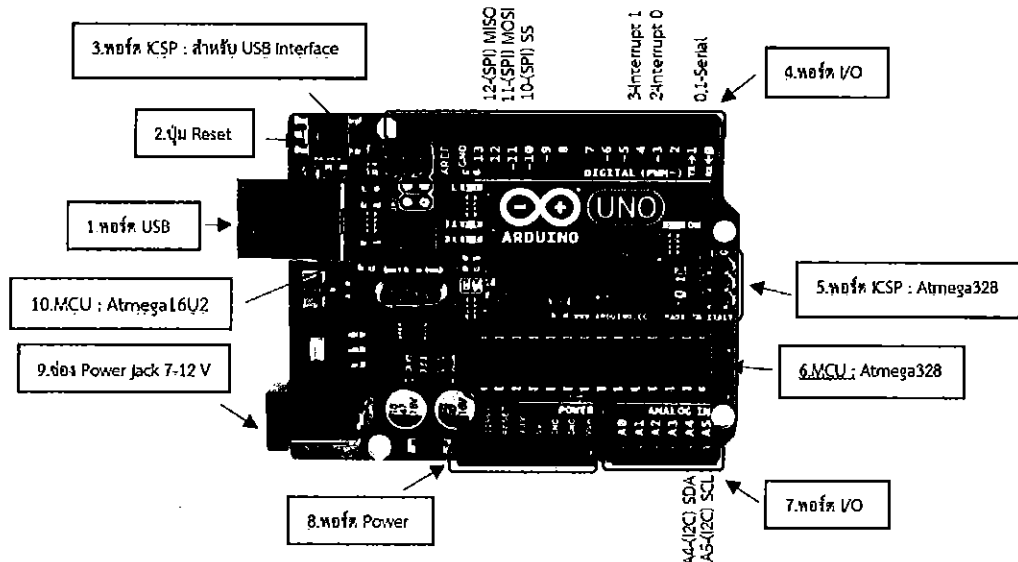
รูปที่ 2.21 รูปการต่อ Arduino กับบอร์ดต่อเนกประสงค์ และ ต่อแบบShield [5]

### 2.11.1 จุดเด่นของบอร์ด Arduino

1. ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
2. มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
3. Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
4. ราคาไม่แพง
5. Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

### 2.11.2 Layout & Pin out Arduino Board ( Model : Arduino UNO R3 )

โดยบอร์ดมีรายละเอียดตามรูปที่ 2.22



รูปที่ 2.22 รายละเอียดตัวบอร์ด [5]

- 1) USBPort: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
- 2) Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
- 3) ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
- 4) I/OPort: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆเพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
- 5) ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
- 6) MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
7. I/OPort: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
- 8) Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอกประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND,  $V_{in}$
- 9) Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
- 10) MCU Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

19196429

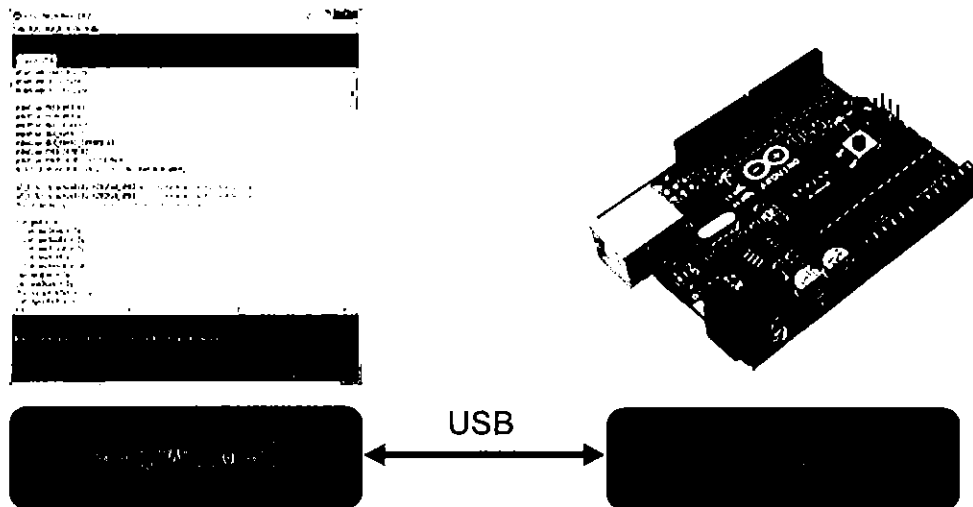
- 5 11 2550



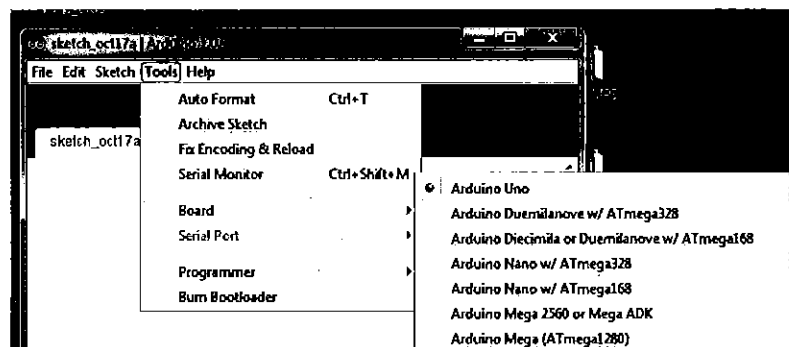
สำนักหอสมุด

### 2.11.3 รูปแบบการเขียนโปรแกรมบน Arduino

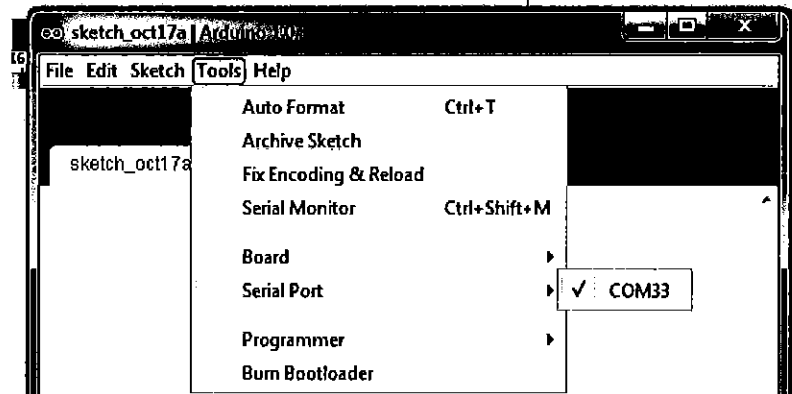
- 1) เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก [Arduino.cc/en/main/software](http://Arduino.cc/en/main/software)
- 2) ทำการสื่อสารคอมพิวเตอร์และบอร์ดผ่านสาย USB ดังรูปที่ 2.23 หลังจากเขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com port ตามรูปที่ 2.24, 2.25



รูปที่ 2.23 การสื่อสาร Arduino ผ่านสาย USB [5]

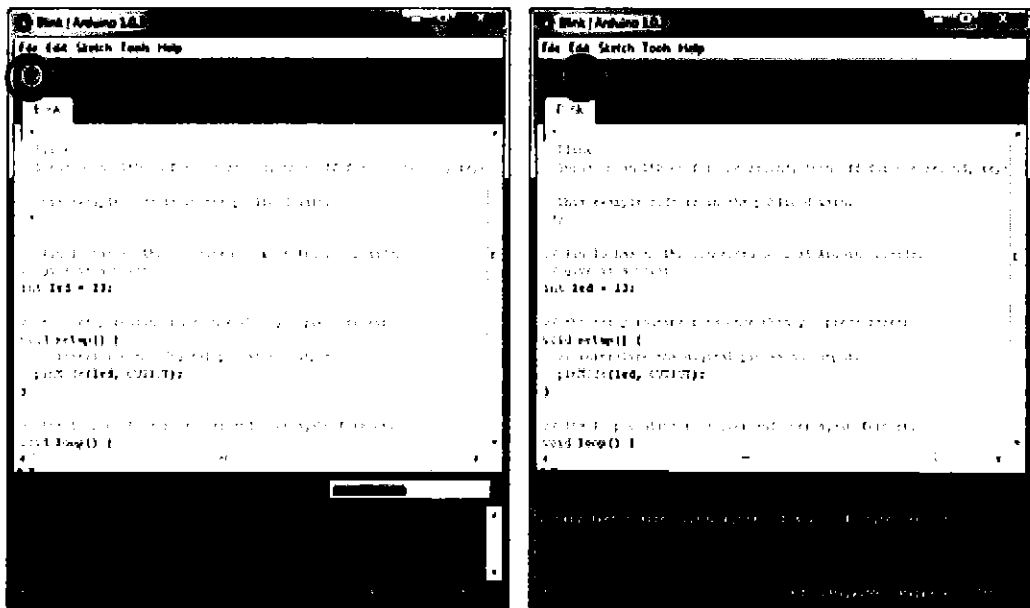


รูปที่ 2.24 เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload [5]



รูปที่ 2.25 เลือกหมายเลข Comport ของบอร์ด [5]

3) กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้น กดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที ตามรูปที่ 2.26



รูปที่ 2.26 การตรวจสอบความถูกต้องของโปรแกรม และการอัปโหลดลงบอร์ด [5]

## 2.12 ส่วนประกอบต่างๆ ใน LabVIEW [6]

โปรแกรมที่เขียนขึ้นมาโดย LabVIEW จะเรียกว่า Virtual Instrument ( VI ) เพราะลักษณะที่ปรากฏทางจอภาพเมื่อผู้ใช้ใช้งานจะเหมือนกับเครื่องมือหรืออุปกรณ์ทางวิศวกรรม ในขณะที่เดียวกันหลังฉากของอุปกรณ์เสมือนจริงเหล่านั้นจะเป็นการทำงานของฟังก์ชัน, Subroutines และโปรแกรมหลักเหมือนกับภาษาทั่วไป สำหรับ VI หนึ่งๆ จะประกอบด้วยส่วนประกอบ 3 ส่วนคือ

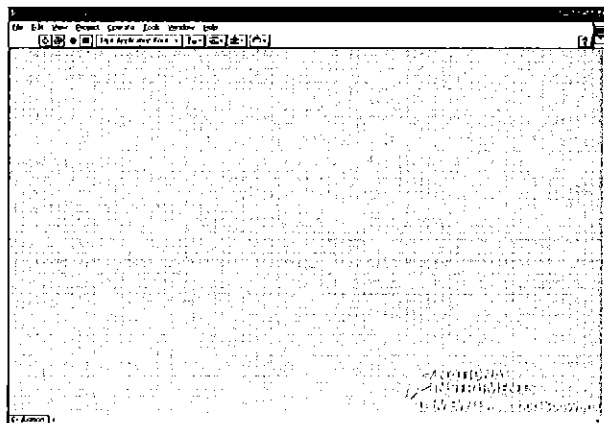
### 2.12.1 Front Panel

### 2.12.2 Block Diagram

### 2.12.3 Icon และ Connector

ทั้งสามส่วนนี้จะประกอบกันขึ้นมาเป็นอุปกรณ์เสมือนจริง ลักษณะและหน้าที่ของส่วนประกอบทั้งสามมีดังต่อไปนี้

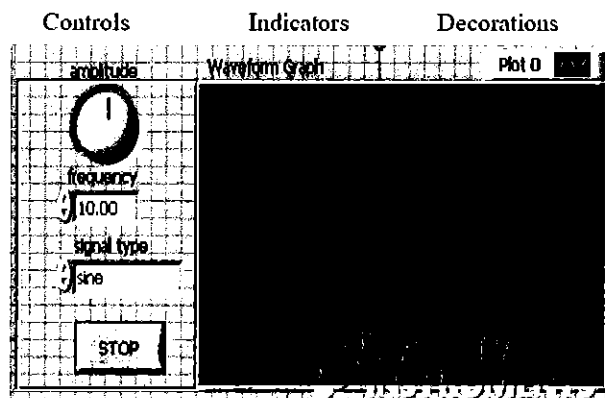
2.12.1 Front Panel หรือหน้าแสดงผล จะเป็นส่วนที่ใช้สื่อความกันระหว่างผู้ใช้กับโปรแกรม (หรือที่นิยมเรียก User Interface ) โดยทั่วไปจะมีลักษณะเหมือนกับหน้าแสดงผลของของเครื่องมือหรืออุปกรณ์ที่ใช้งานด้านการวัดต่างๆ ไป โดยทั่วไปจะประกอบด้วย สวิตช์ ปิดเปิด, ปุ่มบิด, ปุ่มกด จอแสดงผลหรือแม้แต่ค่าที่ผู้ใช้สามารถกำหนด ดังรูป 2.27



รูปที่ 2.27 Front Panel ของ LabVIEW [6]

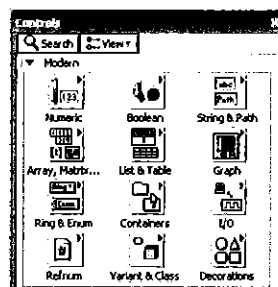
Object ที่อยู่บน Front Panel จะมีอยู่สามประเภท คือ

- 1) Control คือประเภทที่รับค่าจากผู้ใช้ ( Input ) ซึ่งผู้ใช้สามารถพิมพ์ค่าลงไป หรือใช้เมาส์คลิกเพื่อเปลี่ยนแปลงค่าได้ เช่น ปุ่มหมุน ปุ่มเลื่อน สวิตช์ เป็นต้น
- 2) Indicators คือประเภทที่ใช้แสดงค่าต่างๆ เท่านั้น ( Output ) ผู้ใช้ไม่สามารถแก้ไขได้ เช่น กราฟ มิเตอร์ LED
- 3) Decorations เป็น Object ที่ไม่เกี่ยวข้องกับโปรแกรมและcode บน Block Diagram เลยแต่มีไว้เพื่อความสวยงามเป็นระเบียบของ Front panel เท่านั้นนั่นเอง ลักษณะของ Front Panel แสดงดังรูป 2.8



รูปที่ 2.28 Object ที่อยู่บน Front Panel ของ LabVIEW [6]

เครื่องมือที่ใช้ในการออกแบบ Front Panel จะประกอบไปด้วย Control Palette และ Tools Palette ซึ่ง LabVIEW มี Controls Palette ที่ใช้ในการออกแบบ Front Panel แสดงดังรูป 2.28 ซึ่งเป็นส่วนที่ติดต่อกับผู้ใช้งาน ( User Interface ) โดยจะจัดเป็นกลุ่มต่างๆ เช่น กลุ่มของตัวเลข ( Numeric ) ซึ่งภายในกลุ่มจะมี Control และ Indicator ต่างๆ ที่เกี่ยวกับตัวเลขตามรูปที่ 2.29



รูปที่ 2.29 Controls Palette ที่ใช้ในการออกแบบ Front Panel



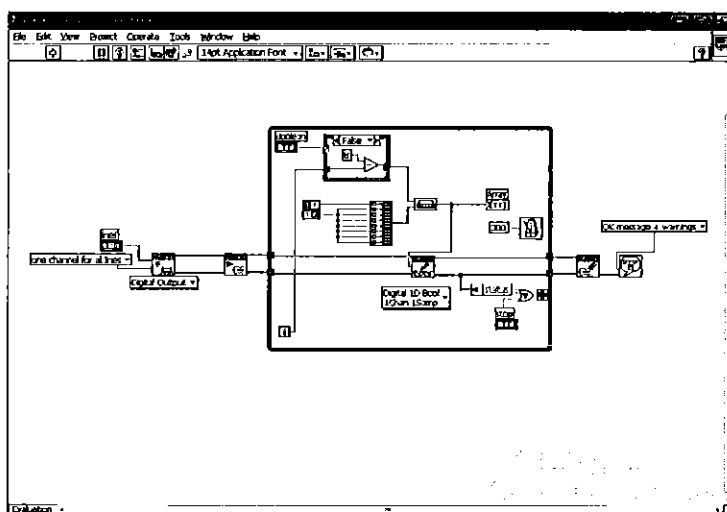
Tools Palette คือ เครื่องมือที่ใช้ในการพัฒนาโปรแกรม ซึ่งจะใช้ทั้งการออกแบบ Front Panel และ Block Diagram ตามรูปที่ 2.29



รูปที่ 2.30 Tools Palette ที่ใช้ในการออกแบบ Front Panel

### 2.12.2 Block Diagram

เพื่อให้เกิดความเข้าใจง่ายขึ้น อาจมอง Block Diagram นี้เป็นเสมือนกับ Source Code หรือโปรแกรมของ LabVIEW ซึ่งปรากฏว่าอยู่ในรูปของภาษา G ส่วนประกอบภายใน Block Diagram นี้จะประกอบด้วย ฟังก์ชัน ค่าคงที่ โปรแกรมควบคุมการทำงานหรือโครงสร้าง จากนั้นในแต่ละส่วนเหล่านี้ ซึ่งจะปรากฏในรูปของ Block จะได้รับการต่อสาย ( Wire ) สำหรับ Block ที่เหมาะสมเข้าด้วยกัน เพื่อกำหนดลักษณะการไหลของข้อมูลระหว่าง Block เหล่านั้น ทำให้ข้อมูลได้รับการประมวลผลตามที่ต้องการและแสดงผลออกมาให้แก่ผู้ใช้ต่อไป ตามรูปที่ 2.31



รูปที่ 2.31 ตัวอย่าง Block Diagram [6]

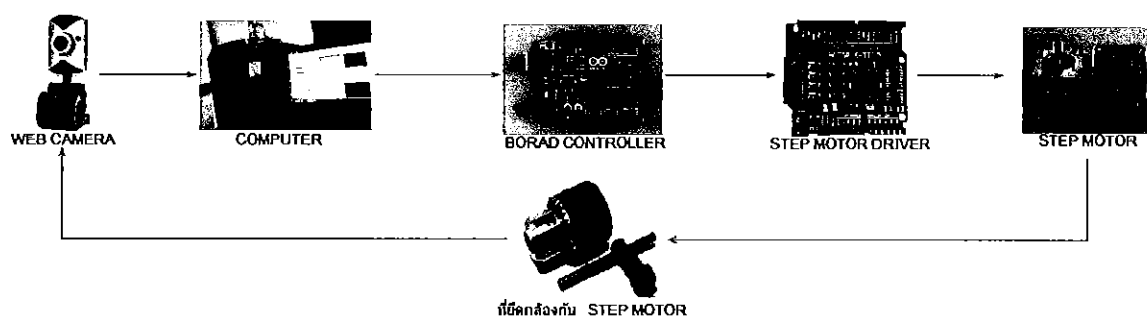
## บทที่ 3

### วิธีการดำเนินงาน

ในการทดลองโครงงานนี้ เป็นการทดลองจากการพัฒนาแนวคิด เพื่อสร้างระบบตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่าย เริ่มจากการซอฟต์แวร์ การออกแบบระบบควบคุมการเคลื่อนที่ของกล้องในการหมุนรอบแนวแกน Y และออกแบบชุดควบคุมการทำงาน ชุดที่จับตอกส่วานที่จะมาใช้ยี่ห้อ Step Motor กับ Web Camera เป็นต้น ทำการทดลอง วิเคราะห์ผลการทดลอง และสรุปผลการทดลอง

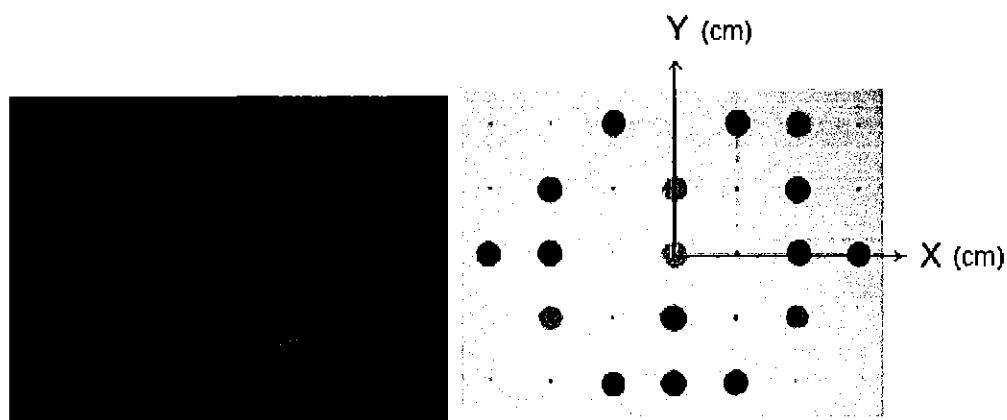
#### 3.1 ศึกษาและออกแบบระบบตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่าย

ทำการทดลองโดยการเชื่อมต่อส่วนการทำงานต่างอาศัยอุปกรณ์ ดังรูป 3.1 เพื่อให้การทดลองมีประสิทธิภาพมากขึ้นจึงนำจำเป็นต้องออกแบบฐานวางอุปกรณ์เพื่อให้ step motor หนึ่ง โดยมีลำดับการทำงานของอุปกรณ์ทั้งหมดในการทดลองคือ กล้อง Web Camara ส่งข้อมูลภาพไปที่คอมพิวเตอร์ประมวลผลด้วยโปรแกรม LabVIEW สื่อสารกับ Micro Controller Arduino ส่งคำสั่งที่ได้รับจากการประมวลผลไปยัง Step motor driver แล้วทำการขับ Step Motor ให้หมุนตามคำสั่ง

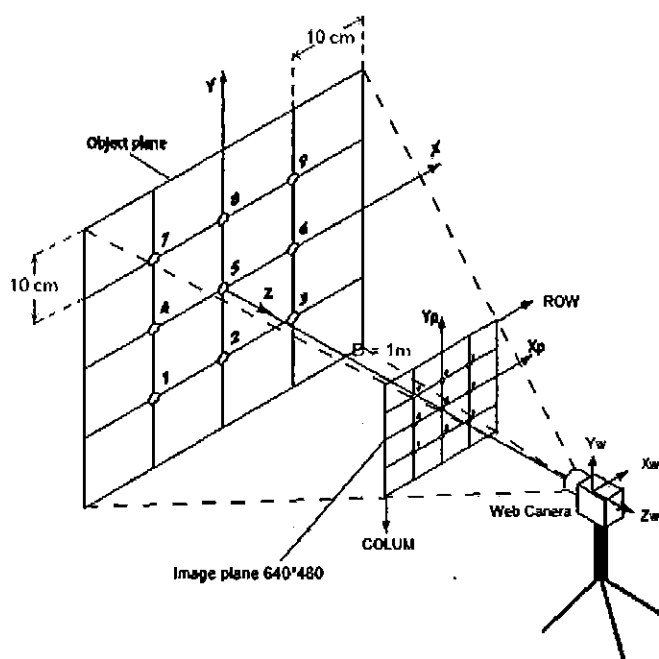


รูปที่ 3.1 แผนภาพการทำงานของกระบวนการตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่าย

ออกแบบการทดลอง สร้างโดยการติด Marker บนกระดานตีเส้นตารางขนาด 10x10 cm บนกระดาน 90x70 cm ( Object Plane ) แล้วทำการติด Marker สีต่างลงไปพร้อมทั้งตั้งกล้อง ที่ระยะห่างจากกล้อง 1 m แล้วเปิดดูภาพที่ได้จากกล้อง ที่มีความละเอียด 640 x 480 pixel ( Image Plane ) แสดงในรูปที่ 3.2 และมีแบบเค้าโครงแสดงในรูปที่ 3.3

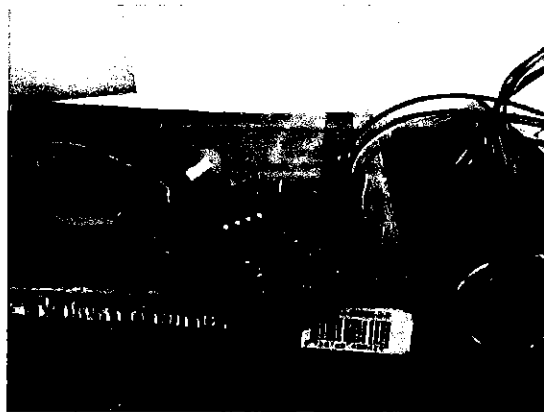


รูปที่ 3.2 รูป Object Plan และ Image Plane



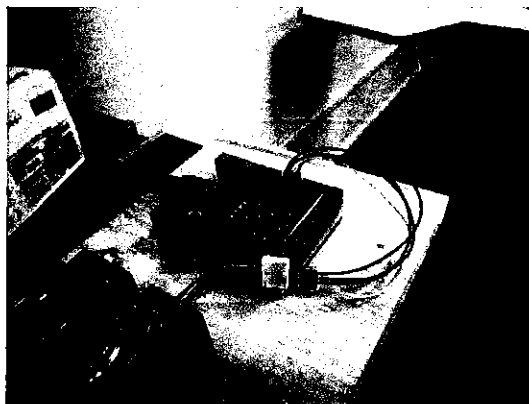
รูป 3.3 แบบเค้าโครงในการทดลอง และการวางแนวแกน X,Y,Z

ประกอบและทดสอบ Step Motor ระบบควบคุมประกอบด้วย บอร์ด Arduino UNO R3 บอร์ด Step Motor driver ( Arduino Shield ) ประกอบเป็นบอร์ดซึ่งมีหน้าที่ควบคุมการหมุนรอบ แกน Y ตามรูปที่ 3.4



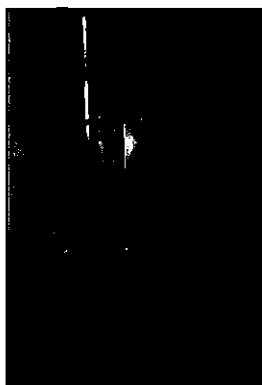
รูปที่ 3.4 บอร์ดชุดควบคุมการทำงาน

การทำงานของ Step Motor มีความจำเป็นที่ต้องยึดเพื่อให้นิ่งและผลการทดลองไม่คลาดเคลื่อนจากการสั่นในการทดลอง พร้อมทั้งทำที่วางชุดควบคุมให้เป็นระเบียบด้วยจึงสร้างฐานการทดลองด้วยไม้แผ่นหนา ช่องวางชุดควบคุม และฐานยึด Step Motor ดังรูปที่ 3.5



รูปที่ 3.5 การวางชุดควบคุมบนฐานยึด Step Motor

การเพิ่มสูงของกล้องเนื่องจากทำการทดลองได้ดีขึ้น เนื่องด้วยทำให้การทดลองได้ความสูงตรงความต้องการหรือตรงกับกึ่งกลางเป้า ( Object Plane ) ใช้ที่จับดอกสว่านในการจับแกน Step Motor เนื่องจากยึดได้มั่นคงและสามารถเพิ่มระยะก้านที่ต่อกับกล้องได้สะดวก ตามรูปที่ 3.6



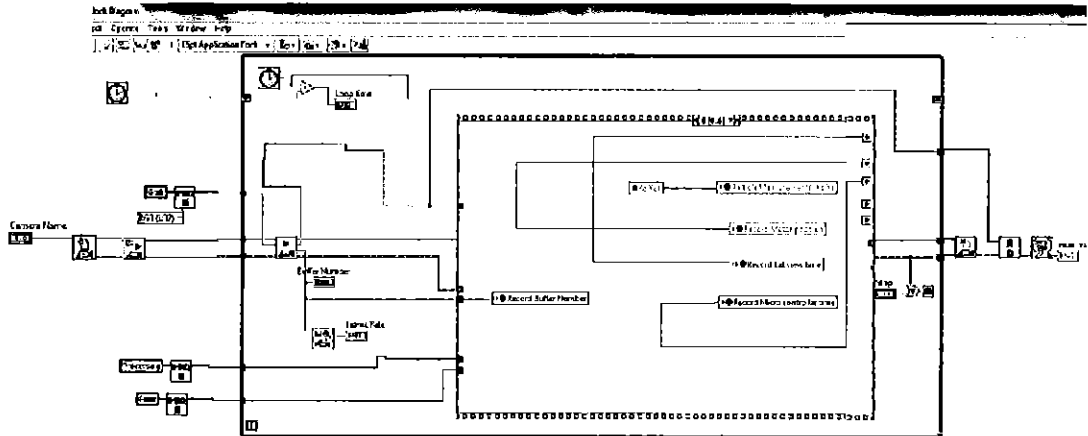
รูปที่ 3.6 การต่อที่จับดอกสว่านกับ Step motor และกล้อง

### 3.2 การพัฒนาโปรแกรมตรวจจับวัตถุ ด้วยโปรแกรม LabVIEW

LabVIEW ซึ่งเป็นโปรแกรมที่ใช้รูปภาพ หรือสัญลักษณ์แทนการเขียนด้วยตัวอักษร ซึ่งข้อดีข้อแรกก็คือการลดความผิดพลาดด้านการสะกดผิดหรือพิมพ์ผิดออกไป เป็นการเขียนโดยใช้หลักการของ Data Flow ซึ่งเมื่อเริ่มส่งข้อมูลเข้าสู่โปรแกรม จะต้องกำหนดทิศทางไหลของข้อมูลว่าจะไปที่การประเมินผลและคำนวณ และแสดงผล ซึ่งลักษณะการเขียนหรือ Data Flow นี้จะมีลักษณะเหมือนกับการเขียน Block Diagram ซึ่งทำให้ผู้เขียนโปรแกรมสามารถให้ความสนใจกับการเคลื่อนที่และเปลี่ยนแปลงข้อมูลได้โดยไม่ต้องจดจำรูปแบบคำสั่งที่ยุ่งยาก โดยมีโปรแกรมโดยรวมดังรูปที่ 3.7 และหน้าจอบทควบคุมโดยรวม ตามรูปที่ 3.8

มีการทำงานหลักอยู่ 4 ส่วน

1. ส่วนแรกจะทำงานโดยรับภาพจากกล้องดิจิทัลส่งมาเป็นข้อมูลให้กับตัวโปรแกรม
2. รับภาพมาประมวลผลเช่นคัดแยกด้วยระบบสีจำนวน Pixels และคัดแยกขนาด รวมทั้งคำนวณค่าพิกัดของวัตถุที่จับได้
3. ส่วนควบคุม Step motor ทำหน้าที่ส่งคำสั่งการควบคุมไปยัง Micro controller
4. ส่วนบันทึกผล ทำการบันทึกการทดลอง เวลา, คำสั่ง, พิกัดเป้าหมาย ในไฟล์ .txt



รูปที่ 3.7 ภาพรวมของ Diagram ที่ใช้ในการทดลอง

COM1

```

1FF7480701104561
1FF528070110440004
1FF1680701107E556
1FF5280701107E547
    
```

File/Read: 953296  
 File/Read: 948432  
 L (cm): 110  
 P (atop): 0

Selection Values

Measurement Parameter	Value
Area	15.35
Range Lower Value	10000
Range Upper Value	10000
Range	OFF
Measurement Type	OFF

Particle Measurements (Pixels)

Area	Perim	CentX	CentY	Area	Perim	CentX	CentY
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35
15.35	15.35	15.35	15.35	15.35	15.35	15.35	15.35

Selected Image

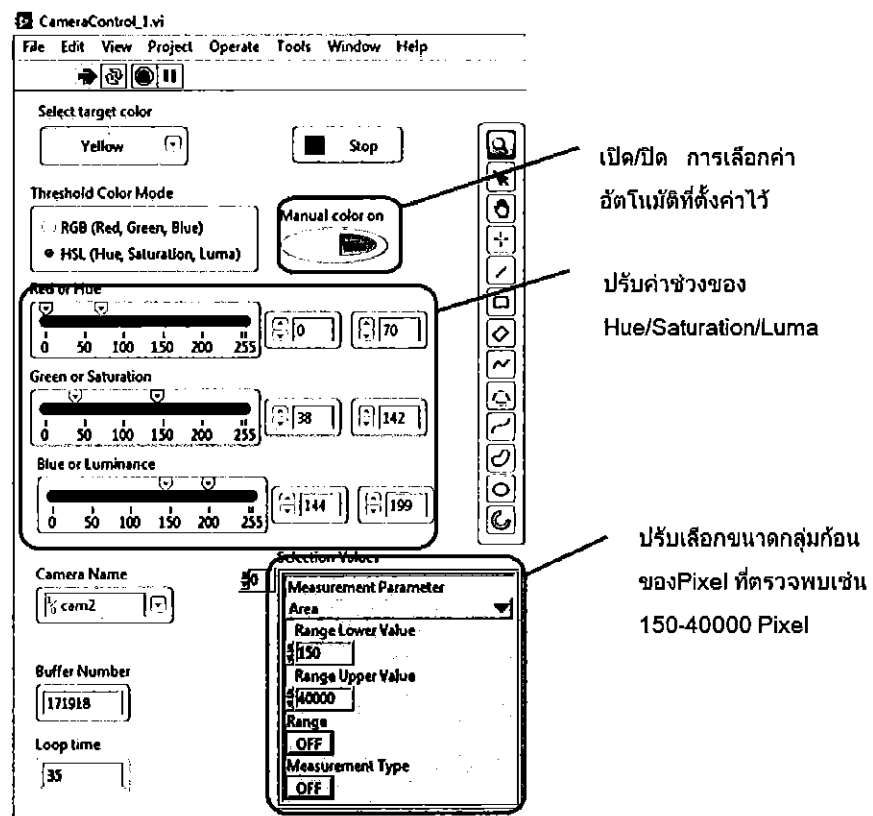
Area: 15.35

COM1

Xc	Yc
15.35	15.35
15.35	15.35
15.35	15.35
15.35	15.35
15.35	15.35
15.35	15.35
15.35	15.35
15.35	15.35
15.35	15.35
15.35	15.35

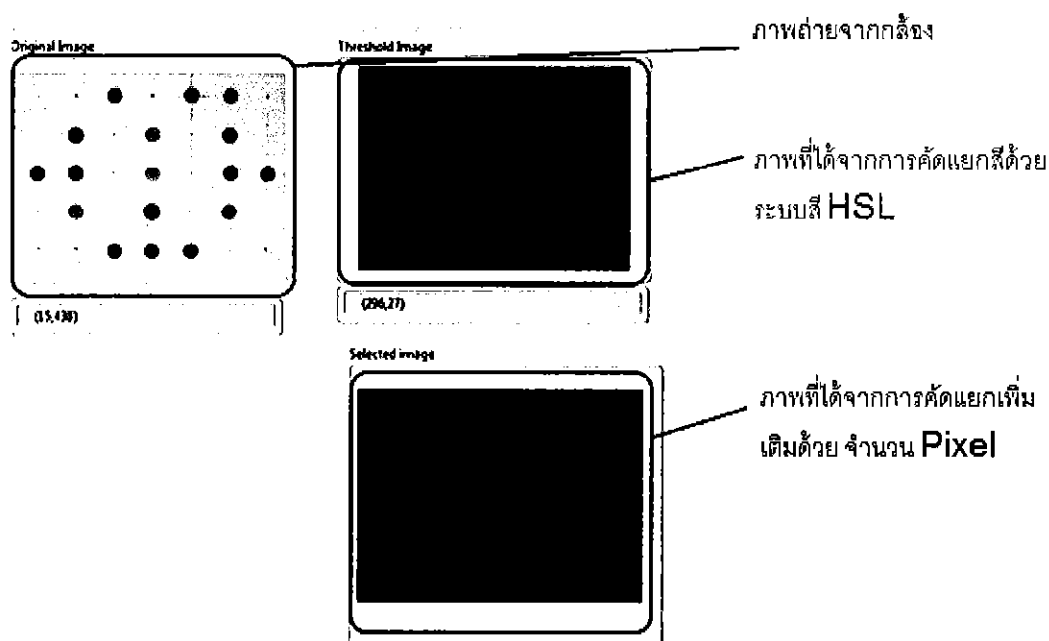
รูปที่ 3.8 ภาพรวมของโปรแกรมที่ใช้ในการทดลอง

ส่วนควบคุมการคัดแยก Marker ด้วยระบบสี HSL ทำหน้าที่คัดแยกวัตถุด้วยระบบสี มีส่วนสำคัญได้แก่ ปุ่มเปิดปิดระบบ Manual ถ้าปิดโปรแกรมจะใช้ค่าที่ได้ตั้งไว้ ส่วนปรับช่วงค่าสีในระบบ HSL ใช้ได้ทั้งการลากเมาส์ และพิมพ์ตัวเลขระบุจำนวน Pixels ที่เกาะกลุ่มกัน โดยระบุช่วงไว้เช่น 150 - 40000 Pixels ตามรูปที่ 3.9



รูปที่ 3.9 การควบคุมการคัดแยก Marker ด้วย ระบบสีและ Pixel

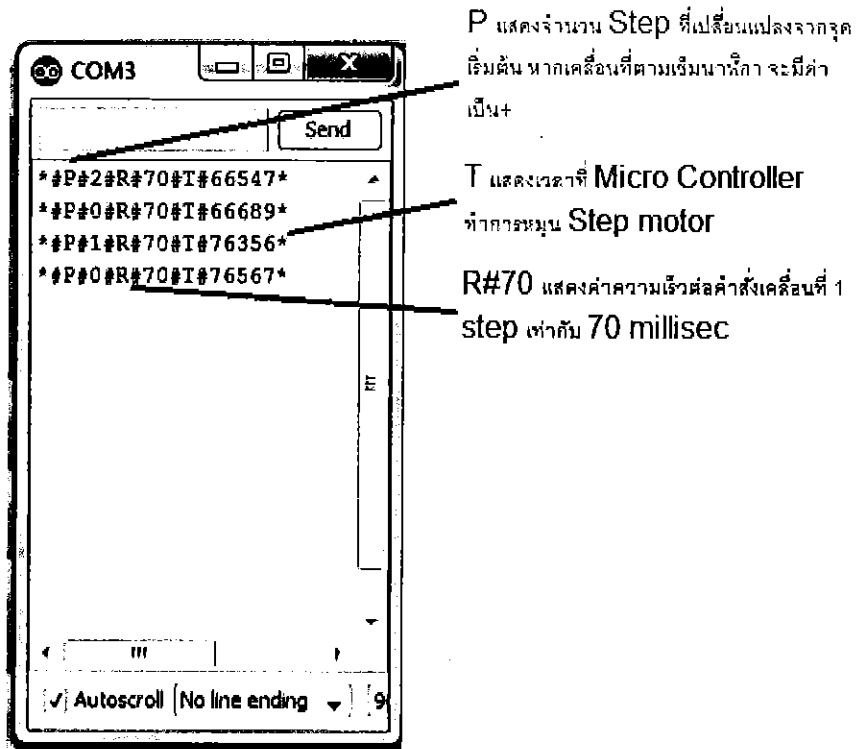
หน้าจอแสดงผลแบ่งเป็น 3 หน้าจอหน้าจอแรกที่วงสี่เหลี่ยมเป็นภาพที่ได้รับจากกล้องโดยสามารถ Zoom เข้า-ออกได้ หน้าจอส่วนวงสี่เหลี่ยมเป็นหน้าจอแสดงการตัดแยกวัตถุด้วยระบบสี HSL หน้าจอส่วนสุดท้ายเป็นหน้าจอที่นำภาพจากจอที่สอง มาประมวลผลซ้ำโดยใช้การระบบช่วงจำนวน Pixels ในการตัดแยก ตามรูปที่ 3.10



รูปที่ 3.10 ผลการตรวจจับวัตถุเป้าหมายด้วยโปรแกรม

ส่วนส่งคำสั่งและแสดงผลการควบคุม Step motor สามารถทำหน้าที่ในการป้อนคำสั่งและแสดงสถานะของ Step motor โดยแบ่งเป็น P คือจำนวน Step ที่ Step motor ที่เคลื่อนไปจากจุดเริ่มต้นโดยมีค่าเป็นบวกเมื่อเคลื่อนที่ตามเข็มนาฬิกา T แสดงเวลาที่ Micro controller ได้รับคำสั่ง และ R คือค่าแสดงความเร็วของสเตปมอเตอร์ที่ได้ทำการตั้งไว้ มีหน่วยเป็น millisecond ตามรูปที่ 3.11





รูปที่ 3.11 ส่วนส่งคำสั่งและแสดงผลการควบคุม Step Motor

ส่วนแสดงผลและใส่ค่าต่างๆในการทดลอง ตามรูปที่ 3.12

Particle Measurements (Pixels)					
รู0	321.88	147.28	885.00	0.00	0.00
รู0	505.41	331.23	892.00	0.00	0.00
	138.03	331.66	861.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00

Pi/rad	953.046
Pj/rad	948.632
L (cm)	100

P (step)	0
----------	---

Xc Yc	
รู0	0.197286 9.80531
รู0	19.7042 -9.83259
	-19.329 -9.87213
	0 0
	0 0

แสดงค่าพิกัดที่วัดได้ที่ Image plane และจำนวน Pixel ของกลุ่มก้อนที่พบ

ที่ใส่ค่าในการทดลอง

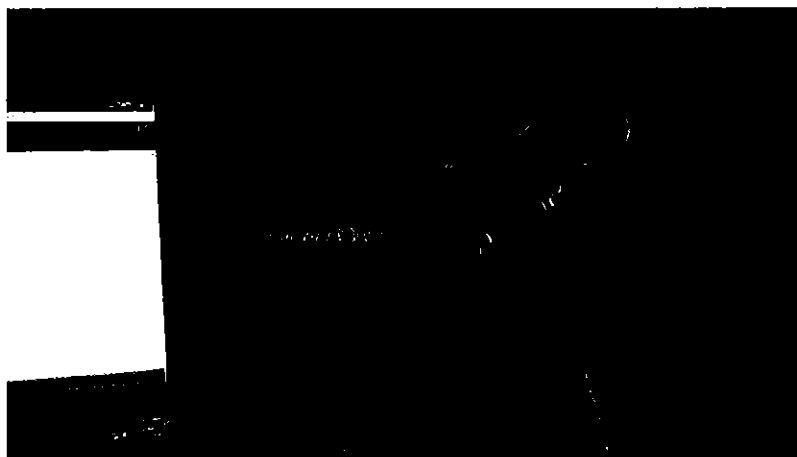
แสดงจำนวนการเคลื่อนที่ P

แสดงผลการคำนวณค่าของพิกัดใน Object plane ด้วยโปรแกรม

รูปที่ 3.12 รายละเอียดในการใส่ค่าในการทดลองและ ตัวเลขผลการทดลอง

### 3.3 วิธีทำการทดลอง

ในเบื้องต้นทำการทดลองควบคุมสเตปมอเตอร์ ให้เคลื่อนที่ตามเป้าหมายที่กำหนด ควบคุมความเร็วรอบด้วยโปรแกรม Arduino IDE ผ่านชุดขยายสัญญาณไฟฟ้าโดยลักษณะการทดลองแสดงดังรูปที่ 3.13



รูปที่ 3.13 การควบคุมสเตปมอเตอร์ผ่าน Arduino IDE

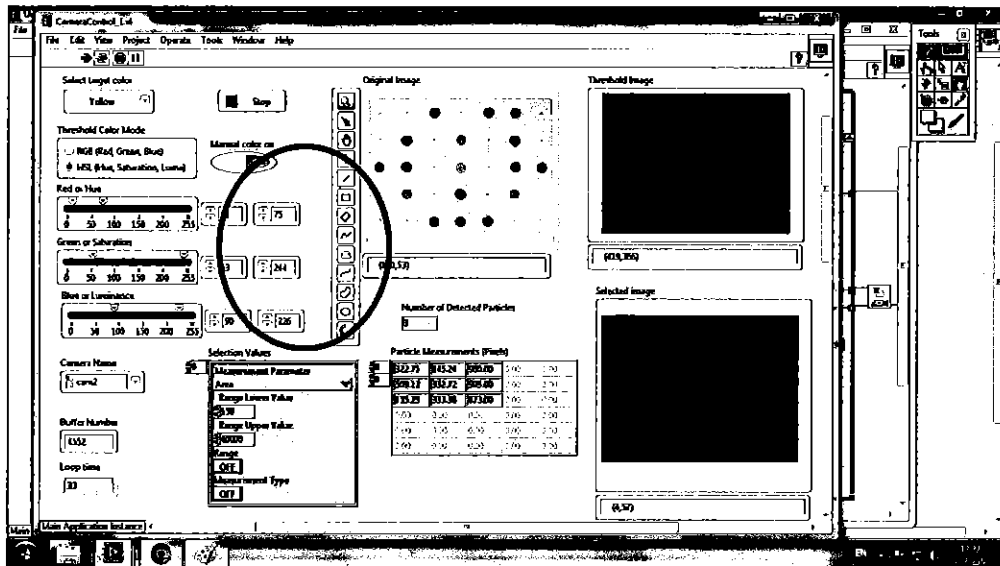
การใช้คำสั่งอื่นในการขับ step motor เช่นการขับแบบ half step (  $\frac{1}{2}$  ) จากการทดลองพบว่า ใช้งานได้แต่ผลที่ได้คือ การขยับต่อ step ไม่สม่ำเสมอจึงไม่นำมาใช้ในการทดลองหากต้องใช้งานที่ ละเอียดยหรือ รอบมากกว่า 200 step/รอบ จึงควรใช้ชุดเกียร์มากกว่า แต่ในการทดลองนี้พบว่า การหมุนโดย Full step โดยไม่ใส่ชุดเกียร์ ก็เพียงพอต่อการทดลอง จึงไม่ใส่ชุดเกียร์เพิ่มลงไปในงานทดลองนี้

แบ่งการทดลองหลักเป็น 4การทดลองดังต่อไปนี้

- 3.3.1 การทดสอบแยกแยะสีจากกลุ่มตัวอย่าง ( Marker )
- 3.3.2 ทำการทดลองเพื่อสอบเทียบหาค่า Pixels per Radian
- 3.3.3 ทดสอบการจับตำแหน่งวัตถุหยุดนิ่งบนระนาบ
- 3.3.4 การทดสอบกล้องเคลื่อนติดตามวัตถุเป้าหมาย

### 3.3.1 การศึกษาและรวบรวมข้อมูลจากการทดสอบแยกแยะสีจากกลุ่มตัวอย่าง ( Marker )

จากการศึกษาเบื้องต้น เลือกค่าสีแบบ HSL เพราะสามารถแยกสีวัตถุดีกว่าได้ RGB และทำการทดลองโดยปรับค่า จนกว่าจะเหลือแต่สีเป้าหมายโดยดูค่าความสว่างจากวงกลมสีแดงในรูปที่ 3.14



รูปที่ 3.14 การทดลองการคัดกรองโดยการปรับค่า HSL ของ Yellow Marker

โดยมีขั้นตอนดังต่อไปนี้

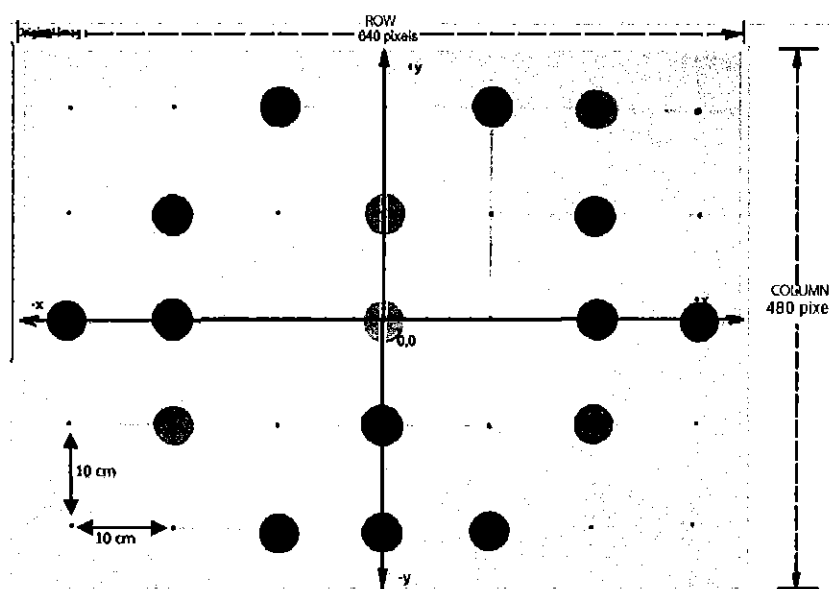
1. รับภาพจากกล้อง กดปุ่มสีเหลืองในรูป 3.14 เพื่อเริ่มการปรับค่าแบบ Manual
2. เลือก HSL ปรับค่า Hue แถบสีแดงโดยการเลื่อนแถบหรือกรอกค่าเช่น 0 - 255 เป็นต้น ปรับค่ามากที่สุด น้อยสุดที่ที่ยังทำให้ Threshold Image ยังเห็นเป้าหมายที่เลือกอยู่โดยเทียบกับรูปที่กล้องรับเข้ามา
3. ปรับค่า Saturation แถบสีเขียวด้วยวิธีเดียวกับขั้นตอนที่ 2
4. ปรับค่า Luminance แถบสีเขียวย ด้วยวิธีเดียวกับขั้นตอนที่ 2 จะได้รูป Threshold Image ตามรูป 3.14 ( เลือก Yellow Marker )
5. บันทึกค่า แล้วทำการทดลองกับ Marker สีอื่นต่อไปด้วยขั้นตอนเดียวกัน

### 3.3.2 ทำการทดลองเพื่อสอบเทียบหาค่า Pixels per Radian

ในการทดลองนี้ทำการตีตารางขนาด 10 cm และทำการติด Marker แล้วทำการทดสอบค่าตำแหน่งที่จอภาพที่กล้อง Web Camera ถ่ายซึ่งมีขนาดแสดงผลเท่ากับ 640 x 480 Pixels ตามรูปที่ 3.15 กำหนดค่าในแนวแกน X, Y ของรูปโดยอาศัยค่าจาก ROW และ COLUMN อ้างอิงจาก

ค่า ROW มีค่าน้อยไปหามากจากซ้ายไปขวามีค่า 0 - 640 Pixels

ค่า COLUMN มีค่าน้อยไปหามากจากล่างขึ้นไปที่บนมีค่า 0 ~ 480 Pixels



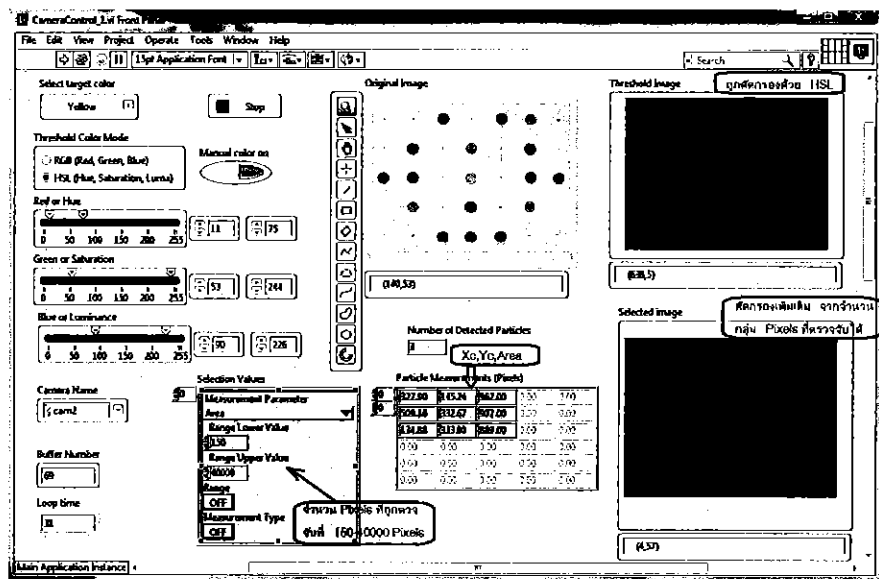
รูปที่ 3.15 การกำหนดค่าในระนาบแกน X, Y ของ Image Plane

จึงกำหนดค่าพิกัดจาก

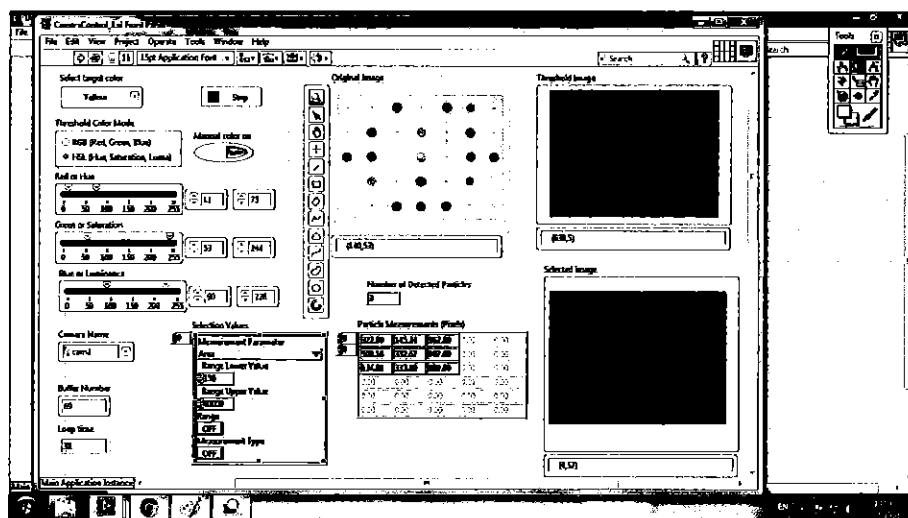
$$\text{ค่าในแกน } y \text{ ของ Image plane} = Y_p = 240 - \text{COLUMN}$$

$$\text{ค่าในแกน } x \text{ ของ Image plane} = X_p = \text{ROW} - 320$$

จากการทำงานเบื้องต้นของโปรแกรม สามารถแยกสีและ บอกรจำนวนPixelที่ตรวจจับได้ ได้ทำการเพิ่มค่าความกว้างช่วงของกลุ่มก้อน Pixels ที่ตรวจจับได้พร้อมทั้งระบุจุดศูนย์กลางของกลุ่มก้อนที่ถูกคัดกรองแล้วเพื่อระบุพิกัดค่าในหน่วย Xp, Yp ก่อนจะทำการทำการถ่ายภาพของแต่ละจุด Marker ละ 5 ครั้ง เพื่อบันทึกค่าที่ได้ ตัวอย่างเช่น Yellow Marker มี 3 ตำแหน่งจึงทำการบันทึกค่า yellow1, yellow2 , yellow3 เป็นต้น ก่อนจะทำการคำนวณเพื่อหาค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และคำนวณหาค่า Pixels per radian จากค่าทำการบันทึกในลักษณะดังกล่าวของทุก Marker มีรายละเอียดหน้าจอบทคอมพิวเตอร์ที่สำคัญต่อการทดลองตาม รูปที่ 3.16 และแสดงตัวอย่างการทดลอง รูปที่ 3.17



รูปที่ 3.16 รายละเอียดต่างของโปรแกรมในการบันทึกค่า ในการคำนวณหา Pixels per Radian



รูปที่ 3.17 ภาพรวมการบันทึกค่า Yellow Marker ครั้งที่ 1

### 3.3.3 ทดสอบการจับตำแหน่งวัตถุหยุดนิ่งบนระนาบ

เป็นการทดสอบโดยทำการหมุนกล้องผ่านสเตปมอเตอร์ในตำแหน่งตั้งแต่ -9 ถึง 9 step แล้วทำการระบุพิกัดตำแหน่งของ Yellow Marker ทั้งสามจุด ที่อยู่บน Object plane เพื่อดูว่าการคำนวณพิกัดของระบบว่ามีความถูกต้องหรือไม่ เมื่อกล้องทำมุมที่แตกต่างกันกับระนาบอ้างอิง โดยการคำนวณพิกัดอาศัยค่าที่ได้จากการทดลองที่ 3.3.2 หรือค่า  $Pixel_x/Radian$  และค่า  $Pixel_y/Radian$  ค่าที่ Image plane หรือค่า  $P_x, P_y$  ที่ได้จากการหาค่าการคำนวณด้วยโปรแกรมในช่อง Particle Measurement (Pixels) ของหน้าจอเสร็จแล้วโปรแกรมจะรับค่าไปคำนวณด้วยสมการที่ (3), (4)

$$X_c = L \tan \left( \left( \frac{1.8 \times P \times \pi}{180} \right) - \frac{X_p}{\left( \frac{Pixel_x}{rad} \right)} \right) \quad (3)$$

$$Y_c = L \tan \left( \frac{Y_p}{\left( \frac{Pixel_y}{rad} \right)} \right) \quad (4)$$

จะได้ค่า  $X_c, Y_c$  ออกมาตามรูปที่ 3.18 ทำการตรวจสอบความคลาดเคลื่อนกับ  $X, Y$  ที่ Object plane หรือ ตำแหน่ง Marker ที่ติดอยู่บนระนาบทดลอง แล้วทำการบันทึกผล

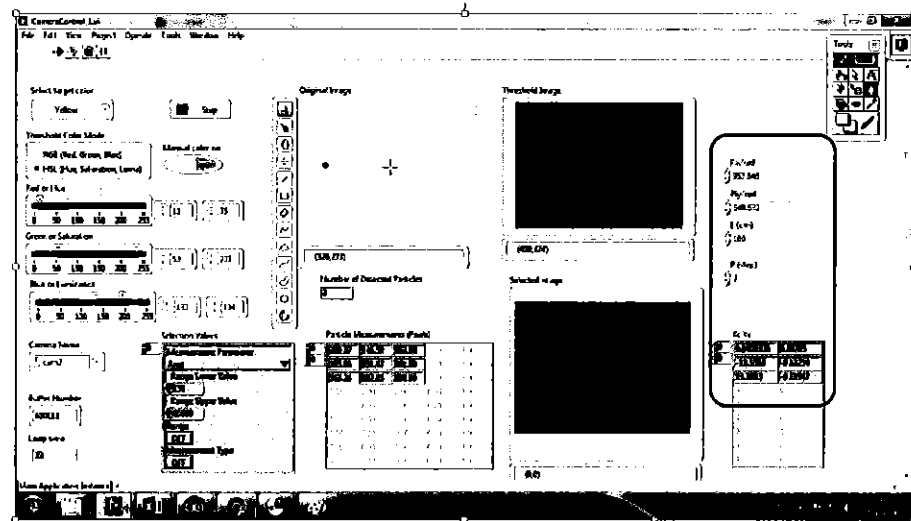
โดยที่  $L$  = ระยะห่างตั้งฉากจากกล้องถึง Object Plane ( ในการทดลองใช้ 100 cm )

$P$  = จำนวนที่สเตปมอเตอร์ทำการหมุนโดยใช้ค่าอ้างอิงโดยให้

ตามเข็มนาฬิกาเป็นบวก ทวนเข็มนาฬิกาเป็นลบ

$X_c$  = ค่าในแนวแกน X ของ Object Plane ที่ได้จากการคำนวณ

$Y_c$  = ค่าในแนวแกน Y ของ Object Plane ที่ได้จากการคำนวณ



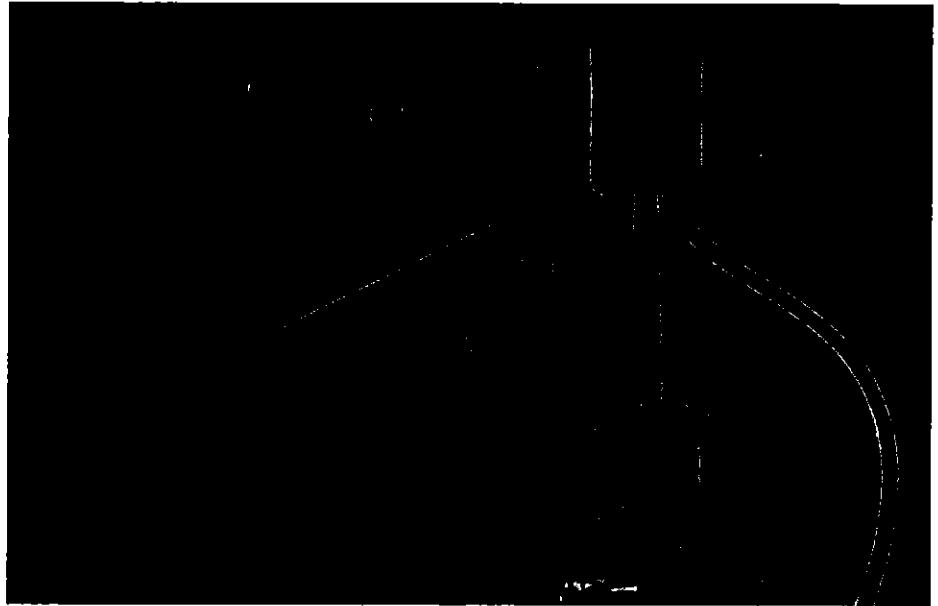
รูปที่ 3.18 การทดสอบบอกพิกัดระยะจากการคำนวณค่าที่ได้จากกล้อง Web Camera

### 3.3.4 การทดสอบการเคลื่อนที่ติดตามเป้าหมายของกล้อง

เริ่มต้นด้วยการนำเอา Yellow Marker ทั้งหมดออกจากกระดาน Object Plane ก่อนการทดลองเพื่อไม่ให้ระบบสับสนในการระบุตำแหน่ง เนื่องจากการทดลองการติดตามวัตถุเคลื่อนที่นี้จะใช้ Marker สีเหลืองเช่นกัน โดยการนำ Yellow Marker ติดด้านโลหะเพื่อสะดวกต่อการเคลื่อนที่ไปตำแหน่งต่างๆบน Object Plane และ เปิดการทำงานการติดตามวัตถุของโปรแกรม เพื่อให้ระบบพร้อมเคลื่อนที่ติดตามวัตถุตามที่ออกแบบไว้ จากนั้นทำการบันทึกค่าระหว่างการเคลื่อนที่ดังต่อไปนี้ ด้วยโปรแกรม

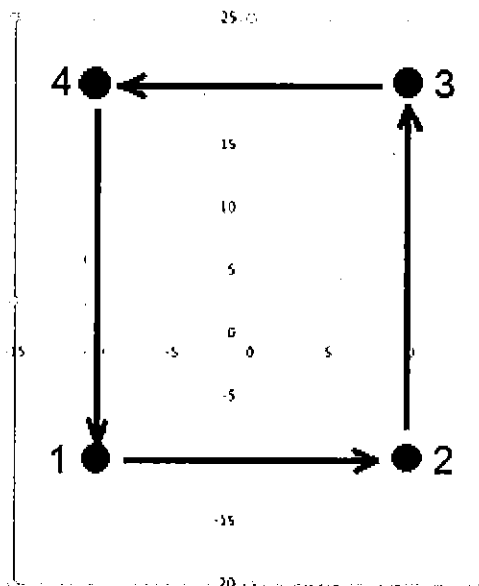
1. เวลาอ้างอิงของไมโครคอนโทรลเลอร์ในหน่วย millisecond
2. ค่า  $X_c$ ,  $Y_c$  ณ เวลาเดียวกับที่ไมโครคอนโทรลเลอร์ได้รับคำสั่ง
3. ค่า  $Y_c$ ,  $X_c$  ณ เวลาเดียวกับที่ไมโครคอนโทรลเลอร์ได้รับคำสั่ง

การทดลองนี้เป็นการทดสอบความสามารถของระบบ ในการเคลื่อนที่ติดตามวัตถุ (Yellow Marker) ซึ่งเคลื่อนตำแหน่งบนระนาบวัตถุ Object Plane โดยมีการตั้งค่าความเร็วการหมุนมอเตอร์ที่มีระยะเวลาในการเปลี่ยนตำแหน่งที่ 70 millisecond/step ดังแสดงการทดลองตามรูปที่ 3.19



รูปที่ 3.19 การติดตามวัตถุเคลื่อนที่ตาม Yellow Marker

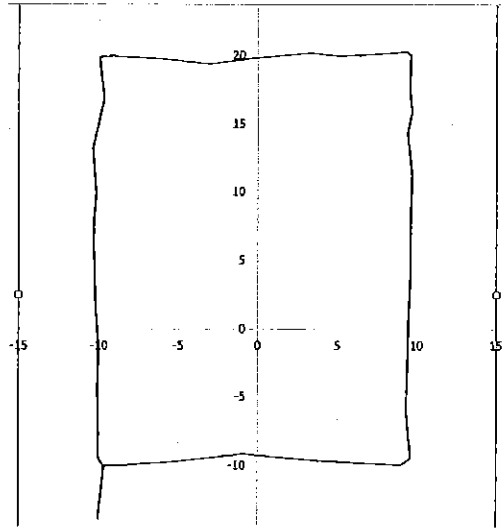
การออกแบบการเคลื่อนที่ติดตามวัตถุ โดยเริ่มเคลื่อนที่จากจุด  $(-10,-10) \gg (10,-10) \gg (10,20) \gg (-10,20) \gg (-10,-10)$  ตามรูปที่ 3.20 แล้วนำค่าที่ทำการบันทึกมาแสดงในรูปกราฟตามรูปที่ 3.21



รูปที่ 3.20 การออกแบบการเคลื่อนที่เพื่อทดสอบบอกรหัสติดตามวัตถุเคลื่อนที่

โดยกล้อง Web Camera ร่วมกับอุปกรณ์ที่จัดทำ



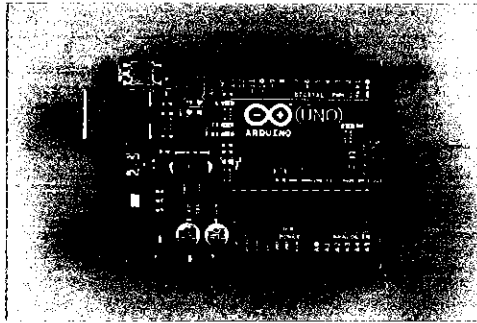


รูปที่ 3.21 กราฟที่ได้จากค่าการทดลองติดตามการเคลื่อนที่ (ตามรูปแบบในรูปที่3.20)

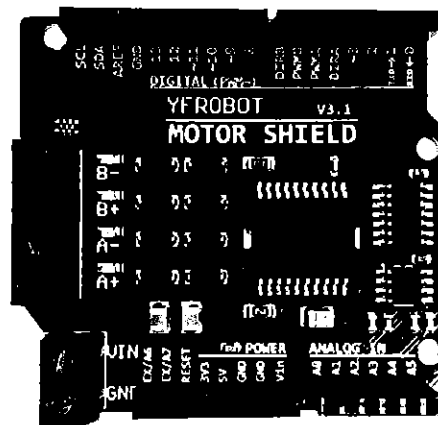
### 3.4 เครื่องมือและอุปกรณ์ที่ใช้

ตารางที่ 3.1 เครื่องมือและอุปกรณ์ที่ใช้

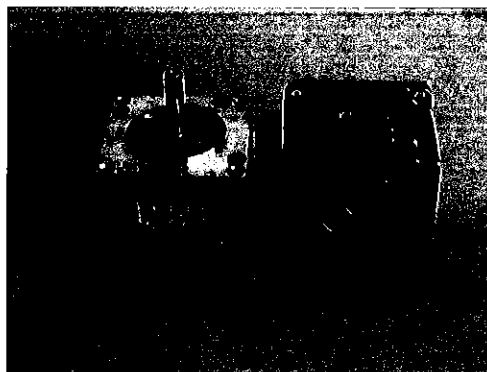
รูปที่	เครื่องมือ/อุปกรณ์	ยี่ห้อ/รุ่น	คุณสมบัติ
3.22	ระบบควบคุม	Arduino UNO R3	บอร์ด ไมโครคอนโทรลเลอร์สำเร็จรูป
3.23	ตัวขยายสัญญาณ ขับมอเตอร์	YFROBOT Motor Shield	ขยายสัญญาณขับ step motor ได้สองตัว
3.24	Step motor	Minebea 5V 2A 1.8 degree/step	6สาย การหมุน1รอบมี 200 step แรงบิดสูง
3.25	แท่นรองการทดลอง	-	ยึดฐานมอเตอร์ และวางชุดระบบควบคุม
3.26	มัลติมิเตอร์	UT30D	วัดความต้านทานไฟฟ้า มอเตอร์
3.27	สายไฟจัมป์เปอร์	-	ต่อฟ่วงอุปกรณ์ได้แน่น
3.28	สาย USB 1.5m	Arduino	ต่อระบบควบคุม กับคอมพิวเตอร์
3.29	กล้อง Web Camera	OKER	5 ล้าน Pixel
3.30	เป้ากระดาษสีทรงกลม Marker	-	มี 5 สี ได้แก่ เหลือง,เขียว,ฟ้า,แดง,ม่วง
3.31	ที่จับดอกสว่าน	MACOH รุ่นเกลียว	ต่อมอเตอร์กับกล้อง Web Camera



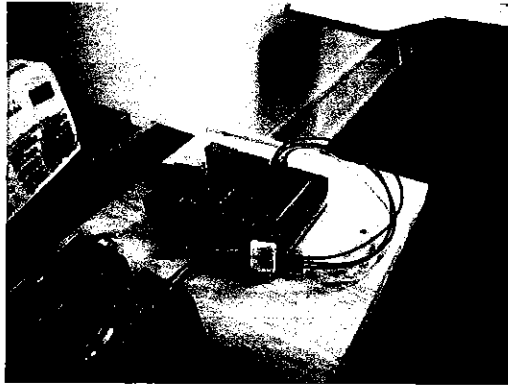
รูปที่ 3.22 ไมโครคอนโทรลเลอร์ Arduino



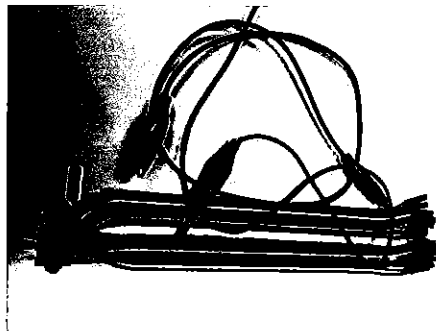
รูปที่ 3.23 บอร์ดขับมอเตอร์ YFROBOT Motor Shield



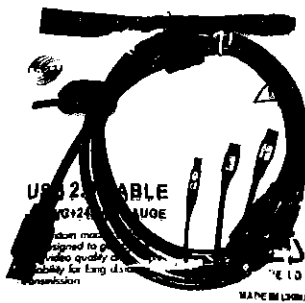
รูปที่ 3.24 Step Motor Minebea 5V 2A



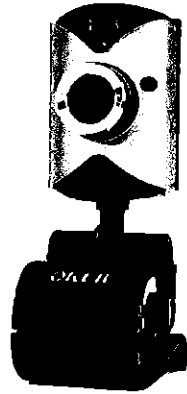
รูปที่ 3.25 ฐานรองการทดลอง ยึดมอเตอร์ และวางระบบควบคุม



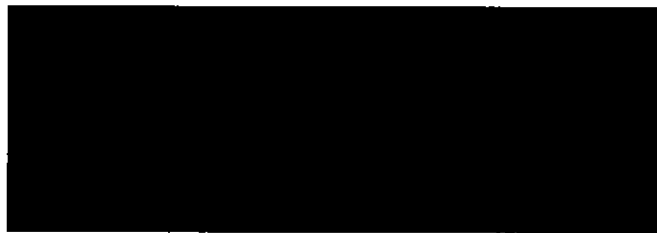
รูปที่ 3.26 ภาพของสายไฟที่ใช้ต่อวงจรภายในระบบ



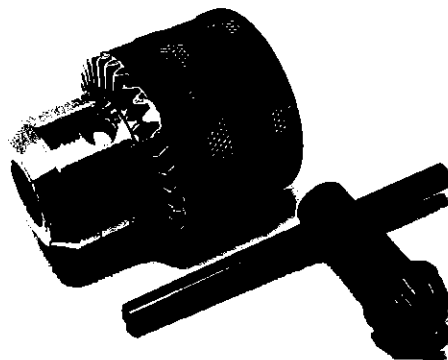
รูปที่ 3.27 ภาพสาย USB ที่ใช้ต่อบอร์ด Arduino กับคอมพิวเตอร์



รูปที่ 3.28 กล้อง Web Camera ยี่ห้อ OKER 5 ล้าน Pixel



รูปที่ 3.29 Marker เป่ากระดาษสีทรงกลม



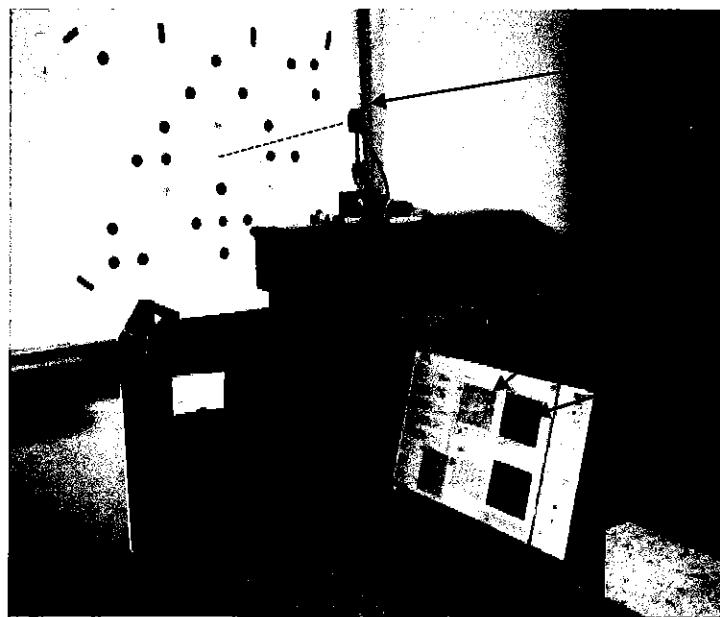
รูปที่ 3.30 ที่จับดอกสว่าน MACOH รุ่นเกลียว 1/2 "

## บทที่ 4

### ผลการทดลองและวิเคราะห์

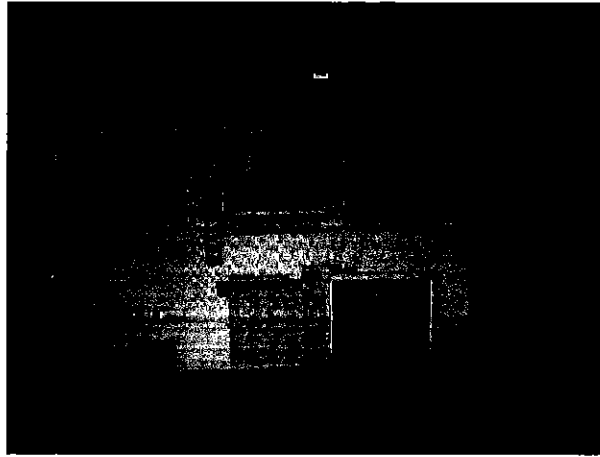
#### 4.1 การศึกษาและรวบรวมข้อมูลจากการทดสอบแยกแยะสีจากกลุ่มตัวอย่าง ( Marker )

ในการทดลองนี้ เป็นการทดสอบความสามารถในแยกแยะสีของระบบติดตามวัตถุที่พัฒนาขึ้น โดยระบบสีที่ใช้ในการเลือกวัตถุเป้าหมาย คือระบบสี HSL โดยเริ่มจากการติดตั้งกล้องห่างจากระนาบอ้างอิง (ติดบนกระดาน) เป็นระยะ 1 เมตร โดยจัดวางชุดทดสอบตามรูปที่ 4.1 โดยหน้าจอแสดงผลของโปรแกรม แสดงในรูปที่ 4.2 และรูปที่ 4.3



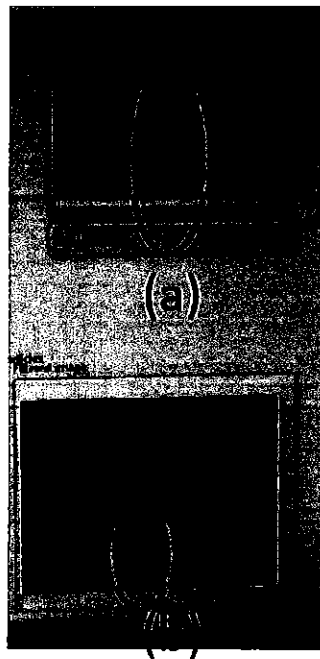
กล้องตั้งอยู่กับที่ ณ จุดกึ่งกลางกระดานObject  
แสดงภาพที่กล้องตรวจจับได้  
จอแสดงผลการคัดแยกด้วยระบบสี

รูปที่ 4.1 การทำงานทั้งหมดของระบบตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่าย



รูปที่ 4.2 หน้าจอขณะทำงานทดลอง

เนื่องด้วยการทดลองนี้จะดูผลที่จอ (a) ในรูปที่ 4.3 เป็นหลัก และยังไม่ใช้การทำงานในหน้าจอ (b) เมื่อทำการทดลองจึงทำการเก็บค่าเพียงแค่ค่าสี HSL ที่เหมาะสมกับวัตถุสีต่างๆ จากนั้นทำการบันทึกการทดลองตามตารางที่ 4.1

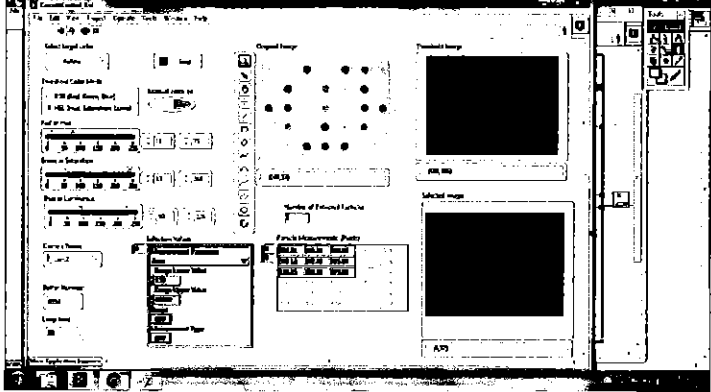
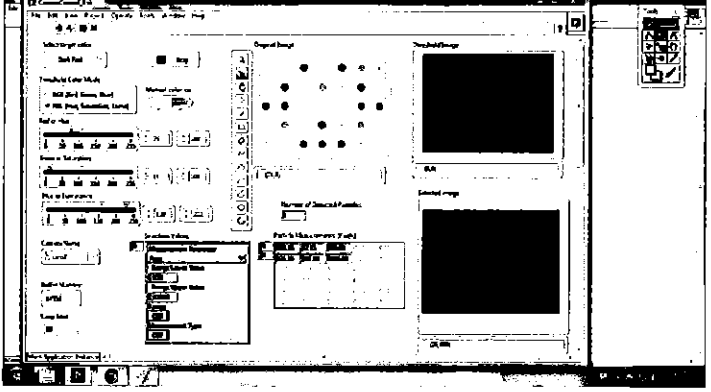
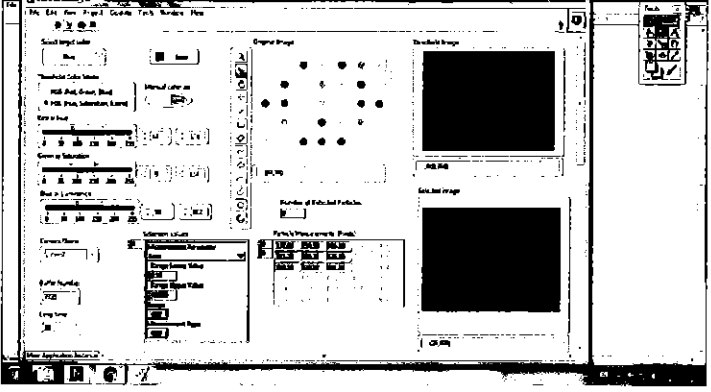


จาก (a) ไป (b) เราจะเห็นว่า  
หลังการคัดแยกด้วยขนาด  
Pixel เพื่อกำจัดจุดไม่พึง  
ประสงค์เช่น วัตถุที่ไม่ใช่  
เป้าหมาย และ จุดที่การ  
ประมวลผลผิดพลาดจากตัว  
Sensor, ตัวโปรแกรม เป็นต้น

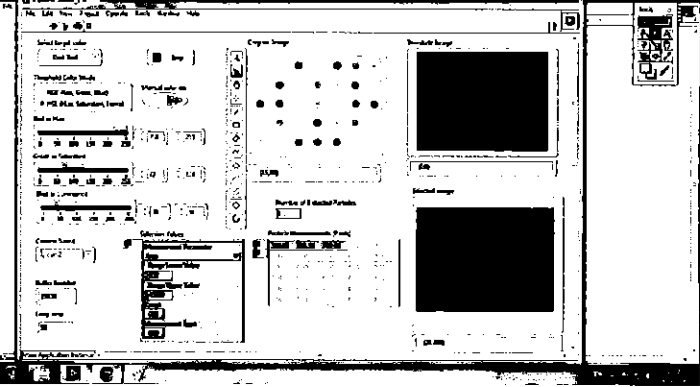
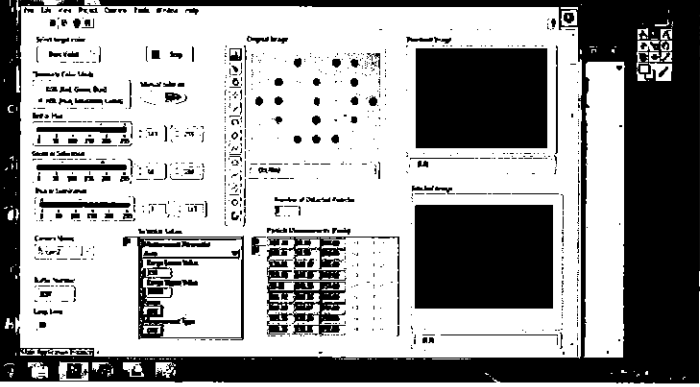
รูปที่ 4.3 รายละเอียดหน้าจอขณะทำงานทดลอง ( zoom รูปที่ 4.2 )

ตารางที่ 4.1 ผลการทดลองค่า HSL ที่วัดได้ของ Marker

จาก HSL คือ Hue, Saturation, Luminance

Marker color	ผลการทดลอง
เหลือง	 <p>ตรวจพบ 3 Marker แยก HSL ได้ 11 - 75, 53 - 244, 90 - 266</p>
เขียว	 <p>ตรวจพบ 2 Marker แยก HSL ได้ 79 - 108, 11 - 108, 159 - 232</p>
ฟ้า	 <p>ตรวจพบ 3 Marker แยก HSL ได้ 84 - 174, 79 - 154, 90 - 162</p>



Marker color	ผลการทดลอง
แดง	 <p>ตรวจพบ 1 Marker แยก HSL ได้ 218 - 255, 72 - 124, 38 - 96</p>
ม่วง	 <p>ตรวจพบ 9 Marker แยก HSL ได้ 183 - 255, 69 - 189, 0 - 113</p>

จากการบันทึกผลการทดลองจากตารางที่ 4.1 สามารถสรุปได้ตามช่วงค่า HSL ที่เหมาะสม ระหว่างการทดลอง ของแต่ละสี Marker ตามตารางที่ 4.2

ตารางที่ 4.2 สรุปค่า HSL ที่วัดได้ของ Marker

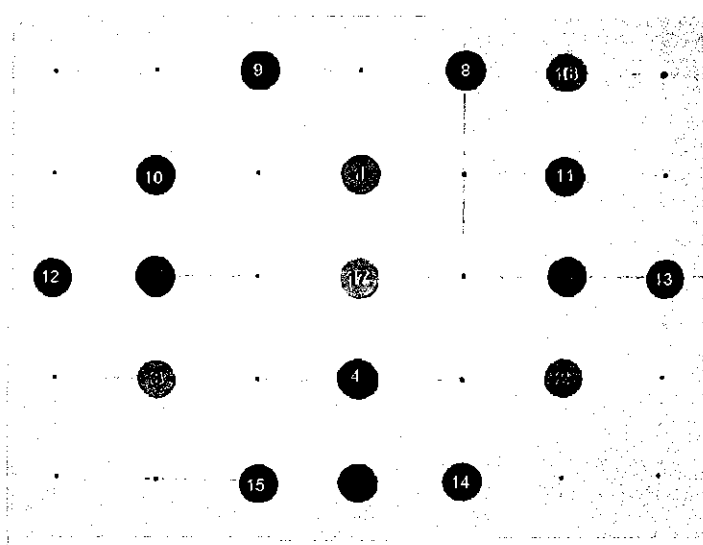
Marker color	Hue	Saturation	Brightness
yellow	11 - 75	53 - 244	90 - 266
green	79 - 108	11 - 108	159 - 232
blue	84 - 174	79 - 154	90 - 162
red	218 - 255	72 - 124	38 - 96

## 4.2 ทำการทดลองเพื่อสอบเทียบหาค่า Pixels per Radian

ค่า Pixels per Radian เป็นค่าที่แสดงถึงความละเอียดของกล้องอย่างหนึ่ง โดยเป็นการแสดงว่าภาพวัตถุจะเอียงไปจากจุดกลางภาพเป็นกี่พิกเซลเมื่อวัตถุอยู่ในตำแหน่งที่ทำมุมกับแนวแกนกลางของกล้องเป็นมุม 1 radian ซึ่งในการทดลองนี้ อ้างอิง จากรูปที่ 3.3 โดยในเบื้องต้นต้องทำให้สามารถหาค่าจากพิกัดค่า  $P_x$ ,  $P_y$  จากโปรแกรมเมื่อได้ค่ามาแล้วจึงหาค่ากลางเพื่อใช้ในการคำนวณ Pixels per Radian ( PPR ) เพราะเมื่อได้ค่า PPR ในแนวแกน X, Y ของ Image plane (เป้าหมายจริง) จะสามารถหาค่า  $X_p = \text{ROW} - 320$  และ  $Y_p = 240 - \text{COL}$  จากนั้นจะสามารถหาค่ามุมที่กระทำบนระนาบได้  $\theta_x = \frac{X_p}{PPR_x}$ ,  $\theta_y = \frac{Y_p}{PPR_y}$  ในการทดลองคล้ายกับการทดลอง 4.1 เนื่องด้วยการตรวจวัดด้วยกล้องค่าที่แสดงบนหน้าจอจึงไม่มี ค่าที่แสดงบนหน้าจอจะเปลี่ยนแปลงตลอดเช่น 0.1-0.3 เป็นต้น ในการทดลองจึงทำการบันทึกรูปหน้าจอขณะทำการทดลองและทำซ้ำ 5 ครั้งเพื่อหาค่าเฉลี่ยเพื่อบันทึกค่าที่นิ่ง ณ ช่วงเวลาที่ทำการทดลองเพื่อใช้ในการคำนวณ ก่อนทำการทดลองจึงกำหนดตำแหน่ง Marker โดยจัดจากบนไปล่างและไล่สีตามลำดับเหลือง>แดง>ฟ้า>ม่วง>เขียว โดยใช้รูปที่ 4.4 ซึ่งแสดงตำแหน่ง สี และหมายเลขเพื่ออ้างอิง โดยผลการทดลองแสดงในตารางที่ 4.3

$$\text{กำหนดให้ } \frac{\text{Pixel}_x}{\text{rad}} \text{ คือ } PPR_x = \frac{[(\text{Row}) - 320]}{\theta_x} \quad (5)$$

$$\frac{\text{Pixel}_y}{\text{rad}} \text{ คือ } PPR_y = \frac{[240 - (\text{Column})]}{\theta_y} \quad (6)$$



รูปที่ 4.4 ลำดับการระบุตำแหน่ง Marker

ตารางที่ 4.3 ค่า PPR เฉลี่ยจากการทำการทดลอง

เลขที่	สี,ลำดับ	พิกัด Object plane		พิกัด Image Plane		$\theta_x$	$\theta_y$	PPR <sub>x</sub>	PPR <sub>y</sub>	
		X	Y	Row	Column					
1	Yellow1	0	10	322.752	145.23	-	0.09967	-	950.85	
2	Yellow2	20	-10	509.164	332.688	0.19740	-0.09967	958.30	929.96	
3	Yellow3	-20	-10	135.288	333.422	-0.19740	-0.09967	935.75	937.33	
4	[Redacted]	0	-10	319.64	333.924	-	-0.09967	-	942.36	
5		0	-20	319.932	429.748	-	-0.19740	-	961.26	
6		20	0	511.326	239.284	0.19740	-	969.25	-	
7		-20	0	132.664	239.5	-0.19740	-	949.04	-	
8		10	20	417.234	49.906	0.09967	0.19740	975.57	963.01	
9		-10	20	228.444	50.066	-0.09967	0.19740	918.60	962.20	
10		-20	10	132.954	146.544	-0.19740	0.09967	947.57	937.67	
11		20	10	508.902	147.314	0.19740	0.09967	956.97	929.94	
12		-30	0	39.128	240.562	-0.29146	-	963.68	-	
13		30	0	601.888	241.378	0.29146	-	967.17	-	
14		10	-20	415.724	428.224	0.09967	-0.19740	960.42	953.54	
15		-10	-20	228.39	430.288	-0.09967	-0.19740	919.15	963.99	
16		20	20	511.104	52.182	0.19740	0.19740	968.13	951.48	
17		0	0	322.038	240.6	-	-	-	-	
		ค่าเฉลี่ย							953.05	948.63

จากตารางที่ 4.3 จะเห็นได้ว่าในกรณีที่ตั้งนั้นที่พิกัด X, Y ( Object plane ) ที่มีค่า = 0 จะไม่สามารถคำนวณค่า PPR ได้เนื่องจากมุมที่ทำกับแกนกลางเป็น 0 โดยกรณีที่จุดนั้นมีค่า ตำแหน่งพิกัด X, Y ที่ไม่เท่ากับศูนย์จะสามารถหาความสัมพันธ์ดังต่อไปนี้

$$\text{เนื่องจากการหาค่า } \theta_x = \tan^{-1}\left(\frac{X}{100 \text{ cm}}\right) \quad (7)$$

$$\text{เนื่องจากการหาค่า } \theta_y = \tan^{-1}\left(\frac{Y}{100 \text{ cm}}\right) \quad (8)$$

จากการทดลองนี้สามารถนำค่า PPR ไปใช้ในการทดลองหรือคำนวณในโปรแกรมได้โดย  $X_p, Y_p$  มาหารด้วย  $PPR_x, PPR_y$  ทำให้หาค่ามุมในหน่วย radian ได้ทำการคำนวณของโปรแกรมได้ มีประสิทธิภาพมากขึ้นสามารถคำนวณพิกัด Marker ที่ Object Plane ได้โดยเพียงทราบค่า  $X_p, Y_p$

### 4.3 ทดสอบการจับตำแหน่งวัตถุหยุดนิ่งบนระนาบ

การทดลองนี้มีจุดประสงค์เพื่อหาว่าเมื่อกล้องมีการหมุนไปด้วยมุมต่างๆ จะได้ผลการคำนวณตำแหน่งวัตถุที่ถูกต้องหรือไม่ โดยการทดลองนี้ใช้ Yellow Marker ในการทำการทดลองแล้วใช้ค่า PPR มาใส่ในสมการของตัวโปรแกรม ทำการสั่งคำสั่งหมุน Step Motor -9 ถึง 9 step ( หมุนตามเข็มนาฬิกาเป็นบวก ) แล้วทำการคำนวณค่าพิกัดของ Yellow Marker ที่แต่ละ P ที่ทำการทดลองแต่ละ step ทำการถ่ายภาพการนำจอมผลการทดลอง 4 ครั้งเพื่อนำไปหาค่าเฉลี่ยพิกัดที่คำนวณได้ในตำแหน่งที่เคลื่อนที่ไปนั้น มีรายละเอียดตามรูปที่ 4.5 ตัวอย่างภาพการทดลอง รูปที่ 4.6, 4.7, 4.8 แล้วทำการบันทึกผล

โดย P คือตำแหน่งที่ทำการหมุน Step Motor ไปโดยมีหน่วยเป็น step

$X_c$  = พิกัด X บน Object Plane ที่ได้จากการคำนวณ

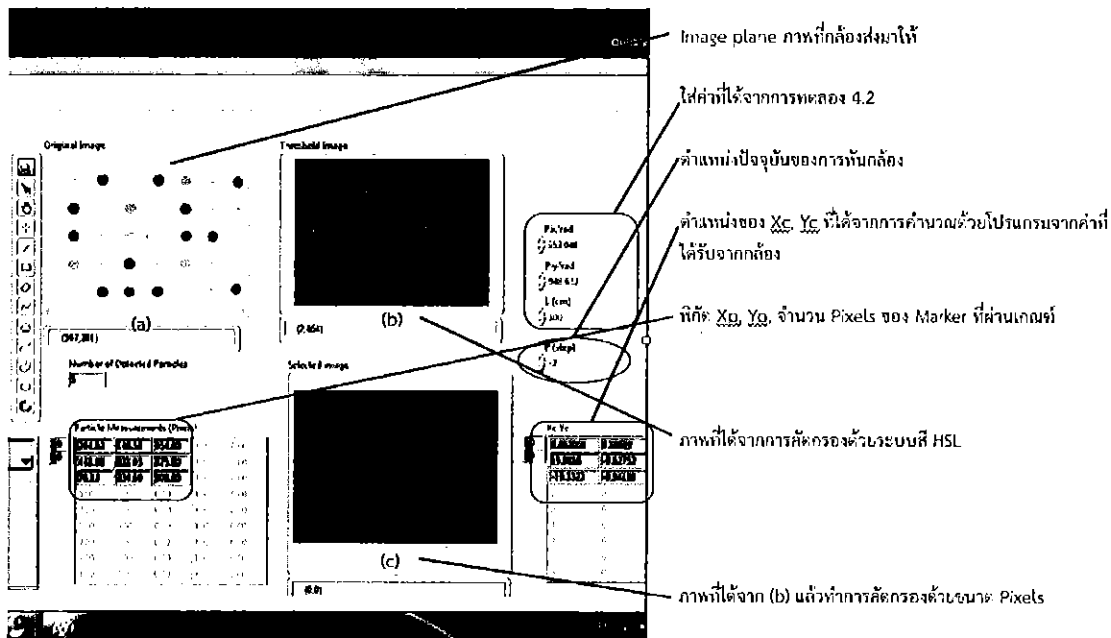
$Y_c$  = พิกัด Y บน Object Plane ที่ได้จากการคำนวณ

L = ระยะห่างจากกล้อง กับ Object Plane ใช้ค่าที่ 100 cm

X = พิกัด X บน Object Plane ที่ได้จากการวัด

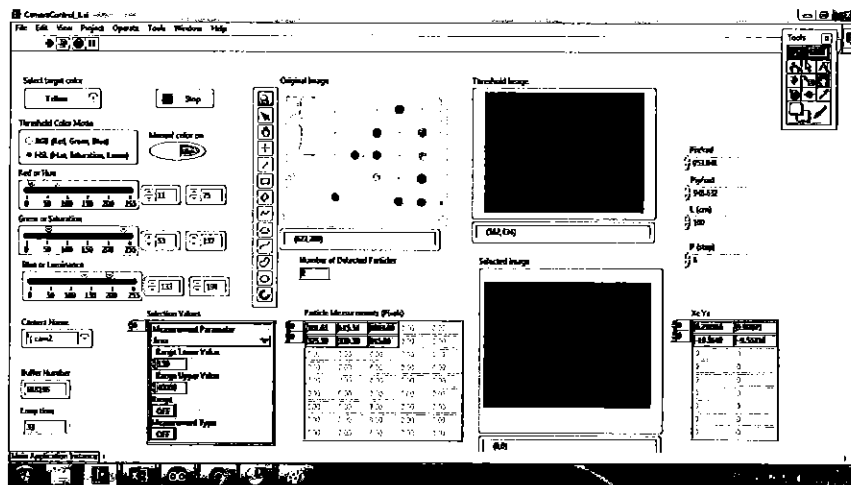
Y = พิกัด Y บน Object Plane ที่ได้จากการวัด

รายละเอียดโดยรวมของส่วนแสดงผลและ ส่วนเพิ่มเติมคำสั่งในการทดลองตามรูปที่ 4.5

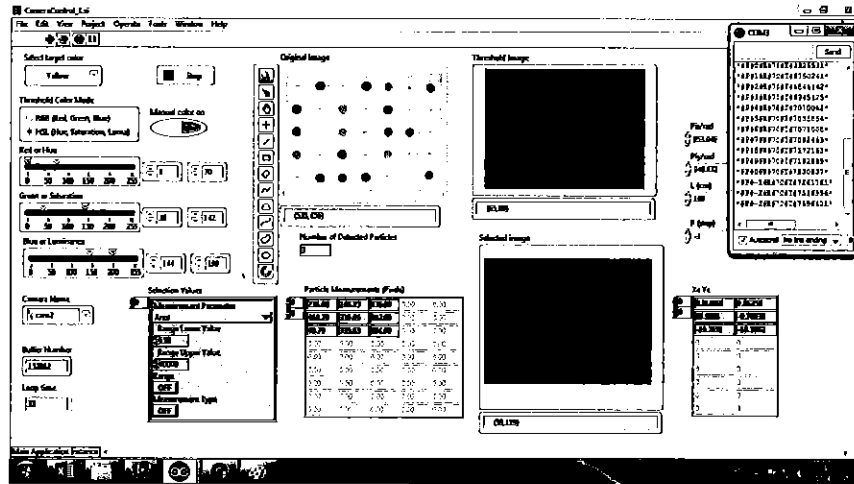


รูปที่ 4.5 รายละเอียดหน้าจอขณะทำการทดลอง

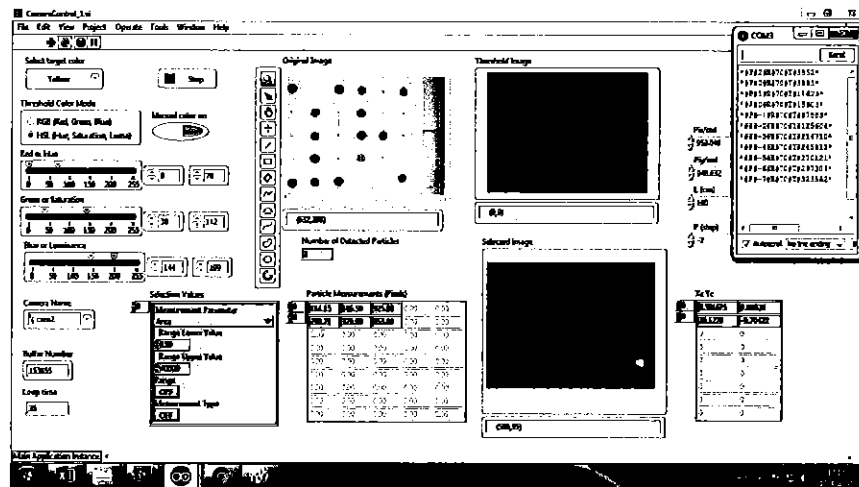
เนื่องด้วยค่าที่หน้าจอจะวิ่งอยู่ตลอดเวลาเพราะเป็นการประมวลผลแบบ Real-Time หรือประมวลผลผันแปรตามข้อมูลในตอนนั้นจึงจำเป็นต้องถ่ายภาพนิ่งเพื่อเก็บค่าข้อมูลที่นิ่งแล้วตามรูปที่ 4.6, 4.7 และรูปที่ 4.8 โดยตัวอย่างแสดงการคำนวณตำแหน่งวัตถุและความผิดพลาดจากตำแหน่งจริง ที่ตำแหน่งของมอเตอร์และกล้องต่างๆคือ -1, 0 และ 1 สเต็ป แสดงดังตารางที่ 4.4



รูปที่ 4.6 ตัวอย่างการทดลองที่ P เท่ากับ 8



รูปที่ 4.7 ตัวอย่างการทดลองที่ P เท่ากับ -3



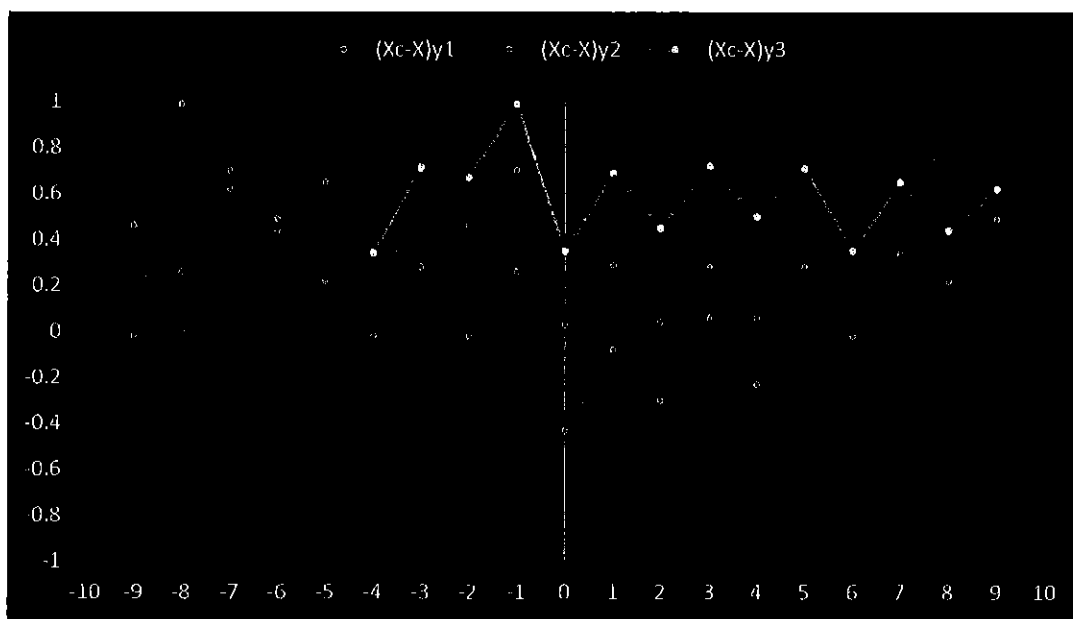
รูปที่ 4.8 ตัวอย่างการทดลองที่ P เท่ากับ -7

ตารางที่ 4.4 ตัวอย่างผลต่างระหว่างค่า X,Y และ Xc,Yc ที่ได้จากการหาค่า error ในทดลอง

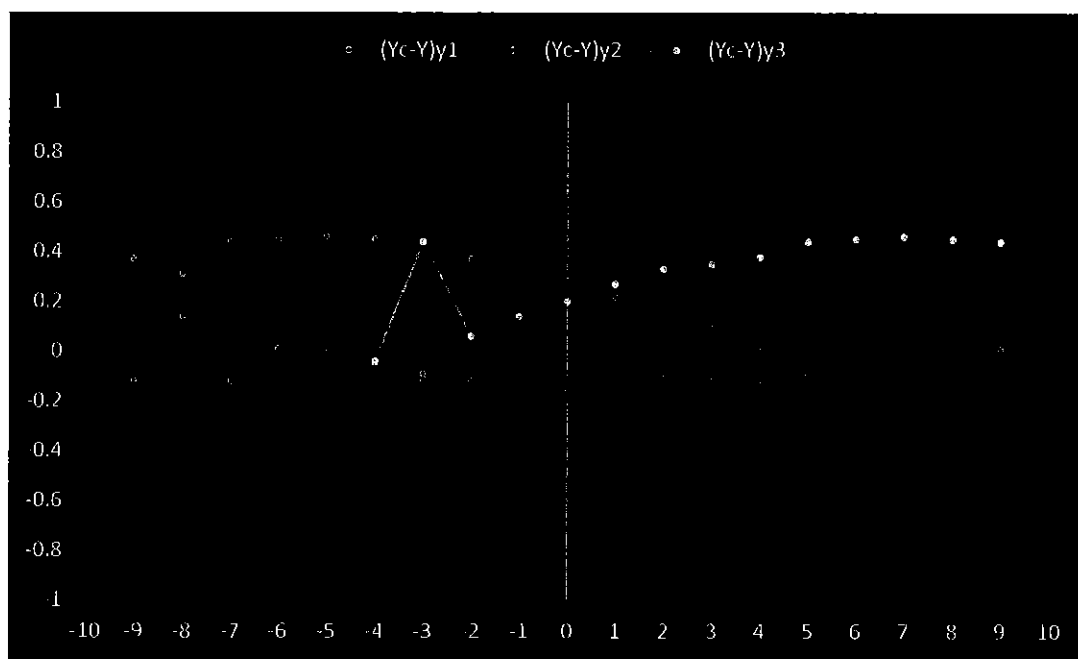
ตำแหน่ง	P	X (cm)	Y (cm)	Xc (cm)	Yc (cm)	$X_{error}$ (cm)	$Y_{error}$ (cm)
Yellow1	-1	0	10	0.70	9.88	0.70	-0.12
Yellow2	-1	20	-10	20.26	-9.69	0.26	0.31
Yellow3	-1	-20	-10	-19.01	-9.86	0.99	0.14
Yellow1	0	0	10	0.03	9.85	0.03	-0.15
Yellow2	0	20	-10	19.57	-9.72	-0.43	0.28
Yellow3	0	-20	-10	-19.64	-9.80	0.35	0.20
Yellow1	1	0	10	0.29	9.88	0.29	-0.12
Yellow2	1	20	-10	19.92	-9.78	-0.08	0.22
Yellow3	1	-20	-10	-19.31	-9.73	0.69	0.27

กราฟแสดงค่า Error ที่ได้จากการทดลองตามรูปกราฟที่ 4.9 และ 4.10

เมื่อ Yellow1, Yellow2, Yellow3 แทนสัญลักษณ์ด้วย  $y_1, y_2, y_3$  อ้างอิงตามรูปที่ 4.5



รูปที่ 4.9 กราฟค่า Error ในแนวแกน X



รูปที่ 4.10 กราฟค่า Error ในแนวแกน Y

โดยจะพบว่าค่าความคลาดเคลื่อนที่ได้จากการคำนวณ มีค่าที่แตกต่างจากค่าจริงที่ Object plane ดังนี้

จากกราฟที่ 4.9 ที่แกน X จากตำแหน่งการเคลื่อนที่ของ Step Motor ( P ) ที่ -9 ถึง 9 คือ

y1 มีค่าโดยอยู่ใน Error เฉลี่ยอยู่ช่วง -0.02 cm ถึง 0.71 cm

y2 มีค่าโดยอยู่ใน Error เฉลี่ยอยู่ช่วง -0.43 cm ถึง 0.99 cm

y3 มีค่าโดยอยู่ใน Error เฉลี่ยอยู่ช่วง 0.35 cm ถึง 0.99 cm

จากกราฟที่ 4.10 ที่แกน Y จากตำแหน่งการเคลื่อนที่ของ Step Motor ( P ) ที่ -9 ถึง 9 คือ

y1 มีค่าโดยอยู่ใน Error เฉลี่ยอยู่ช่วง -0.15 cm ถึง 0.46 cm

y2 มีค่าโดยอยู่ใน Error เฉลี่ยอยู่ช่วง -0.12 cm ถึง 0.45 cm

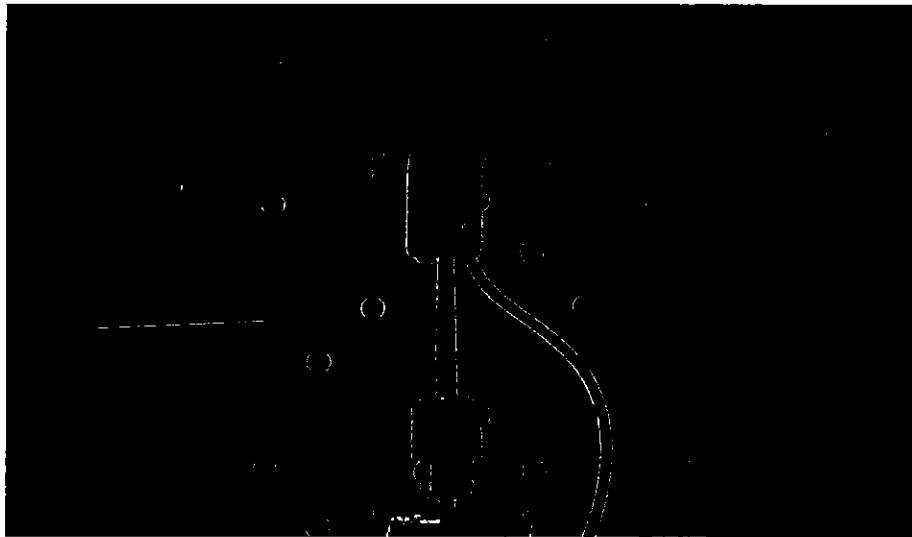
y3 มีค่าโดยอยู่ใน Error เฉลี่ยอยู่ช่วง -0.04 cm ถึง 0.46 cm



#### 4.4 การทดสอบกล่องเคลื่อนติดตามวัตถุเป้าหมาย

การทดลองติดตามวัตถุเป้าหมายเป็นการทดลองเพื่อทดสอบการติดตามเป้าหมายของโปรแกรมว่าสามารถทำงานได้จริง และสามารถบ่งชี้ค่าระหว่างทำการทดลองได้ และระบบสามารถแยกแยะสีเป้าหมายขณะทำการทดลองได้ ทั้งนี้เพื่อเป็นแนวทางในการนำไปต่อยอดเพื่อพัฒนาต่อไป

การทดลองนี้เริ่มต้นด้วยการถอด Yellow Marker ออกจากกระดานที่ทำการทดสอบแล้วทำการนำไปติดแท่งโลหะเพื่อใช้เป็นเป้าเคลื่อนที่ ตามรูปที่ 4.11



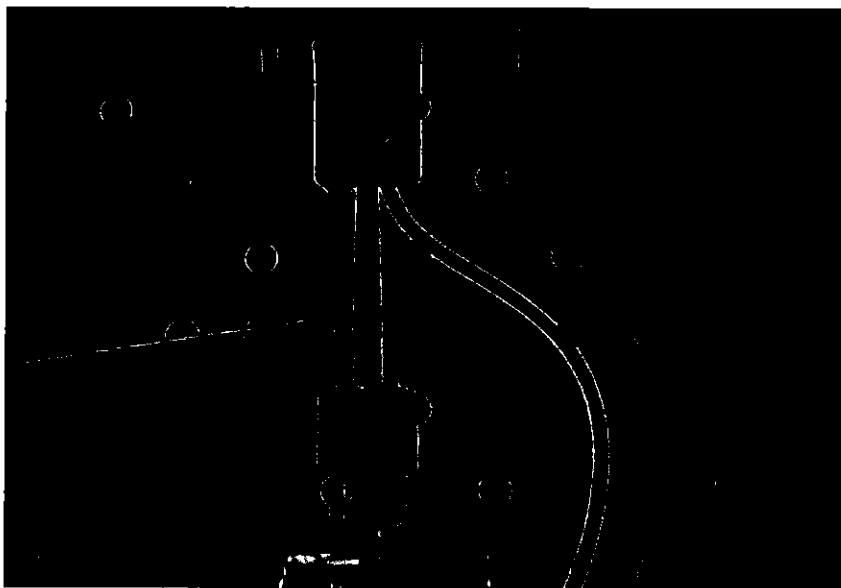
รูปที่ 4.11 กระดาษติด Marker ที่ถอดสีเหลืองออกและ Marker ติดด้ามโลหะ

ทำการสั่งการให้โปรแกรมทำงานติดตามวัตถุด้วยการหันกล้องแบบอัตโนมัติ จากนั้นทำการเคลื่อนที่ Yellow Marker ไปตามจุดต่างๆ บนระนาบ พร้อมทั้งทำการบันทึกค่าอัตโนมัติตามตัว Yellow Marker

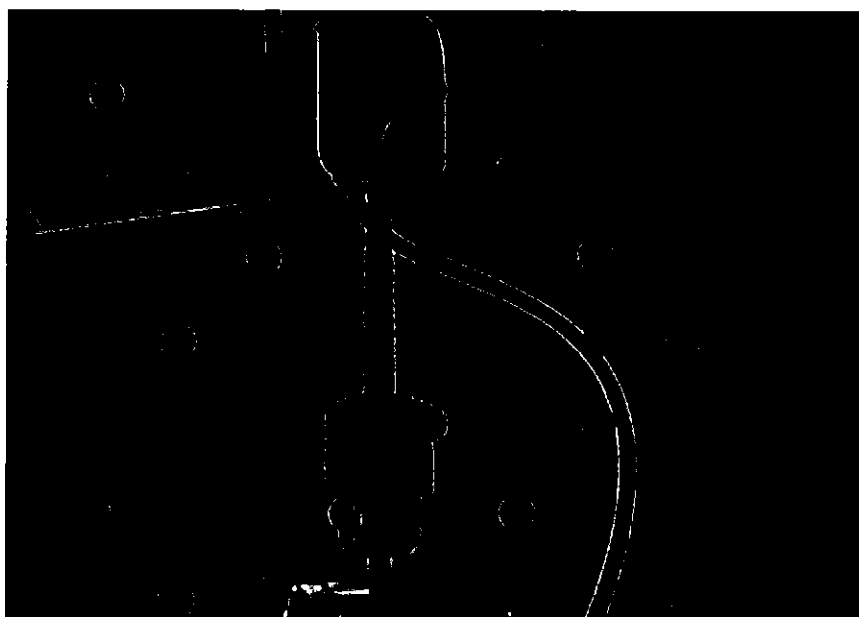
โดย เมื่อเคลื่อนไปที่กลางกระดานทดสอบ กล้องจะเคลื่อนที่ตามดังรูปที่ 4.12

เมื่อเคลื่อนไปทางขวากระดานทดสอบ กล้องจะเคลื่อนที่ตามดังรูปที่ 4.13

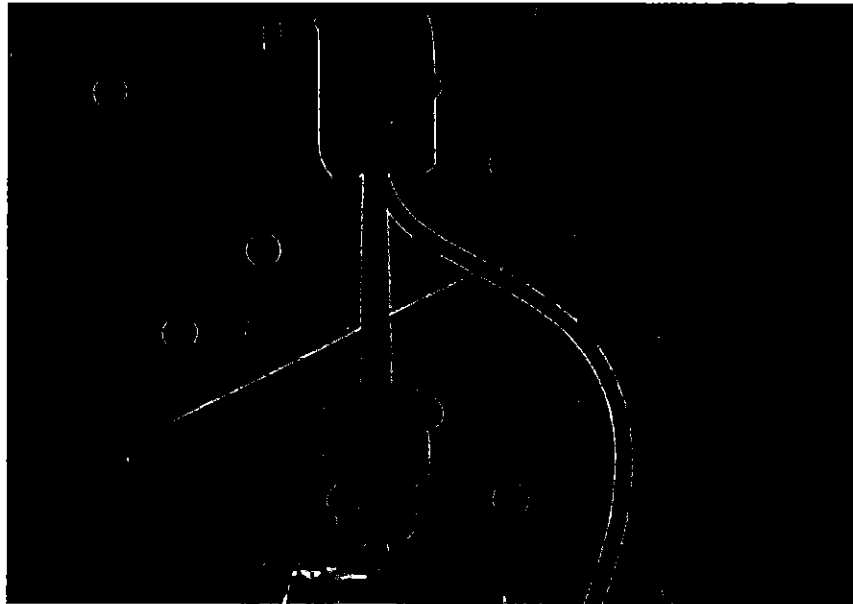
เมื่อเคลื่อนไปทางขวามากขึ้นเทียบกระดานทดสอบ กล้องจะเคลื่อนที่ตามดังรูปที่ 4.14



รูปที่ 4.12 ตัวอย่างการติดตามวัตถุ (1)



รูปที่ 4.13 ตัวอย่างการติดตามวัตถุ (2)



รูปที่ 4.14 ตัวอย่างการติดตามวัตถุ (3)

จากผลการทดลอง จะเห็นได้ว่าระบบกล้องสามารถหันติดตามวัตถุเป้าหมายได้ โดยในการทดลองมีการบันทึกค่าตำแหน่งพิกัดวัตถุ และค่าที่เกี่ยวข้องได้แก่

1. ค่า  $X_p, Y_p$  บน image plane
2. ค่า  $X_c, Y_c$  คือ ค่าพิกัด  $X, Y$  ของเป้าหมายที่ได้จากการคำนวณ
3. ค่า Microcontroller time หรือค่าเวลาอ้างอิงที่ Microcontroller Arduino ควบคุม Step Motor โดยได้รับคำสั่งจากการประมวลผลให้ทำการเคลื่อนที่ติดตาม Yellow Marker

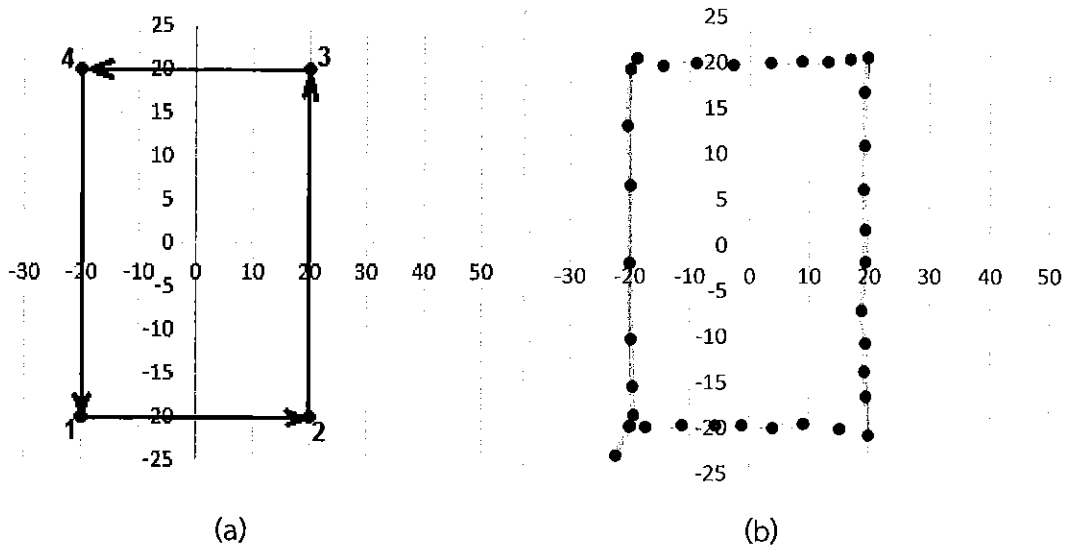
ในการทดลองทุกครั้งทีโปรแกรมประมวลผลติดตามวัตถุเพื่อเคลื่อนที่ติดตามจะนำค่าทั้งสามบันทึกเป็นไฟล์ .txt จะทำให้ได้ข้อมูล 3 ชุดคือ  $X_p, Y_p / X_c, Y_c / Mctime$  ( Microcontroller ) เมื่อทำการสร้างกราฟแสดงความสัมพันธ์ จะใช้ข้อมูลของ  $X_c, Y_c$  มาใช้ในการสร้างกราฟนำชุดข้อมูลที่ได้อ่านเรียงจากจุดแรกไปจุดสุดท้ายโดยเทียบเคียงกับ Mctime เพื่อตรวจสอบตำแหน่งวัตถุที่เวลาต่างๆ เพื่อวิเคราะห์เส้นทางในการเคลื่อนที่ของวัตถุที่บันทึกมาตามลำดับเวลา

#### 4.4.1 ผลการเคลื่อนที่ตามวัตถุเป้าหมาย

จากการอธิบายในเบื้องต้นในหัวข้อ 3.3.4 ทำการทดลองโดยการเคลื่อนที่ Marker ไปตามตำแหน่งต่างๆบนระนาบจริง (กระดานทดสอบ) โดยใช้กล้องติดตามวัตถุที่ระบบติดตามอัตโนมัติ นำข้อมูลที่ทำการบันทึกมาทำการสร้างกราฟแสดงผล แล้วเปรียบเทียบกับแนวการเคลื่อนของวัตถุตามที่ได้ ออกแบบไว้ โดยผลการทดลองพบว่าระบบสามารถตรวจจับและติดตามการเคลื่อนที่ของวัตถุได้ในหลายรูปแบบของการเคลื่อนที่ ดังแสดงในรูปที่ 4.15 - 4.17

โดย (a) คือ รูปแบบการเคลื่อนที่ที่ได้ออกแบบไว้ในการทดลอง

(b) คือ กราฟแสดงผลของพิกัดที่บันทึกได้จริงจากระบบ



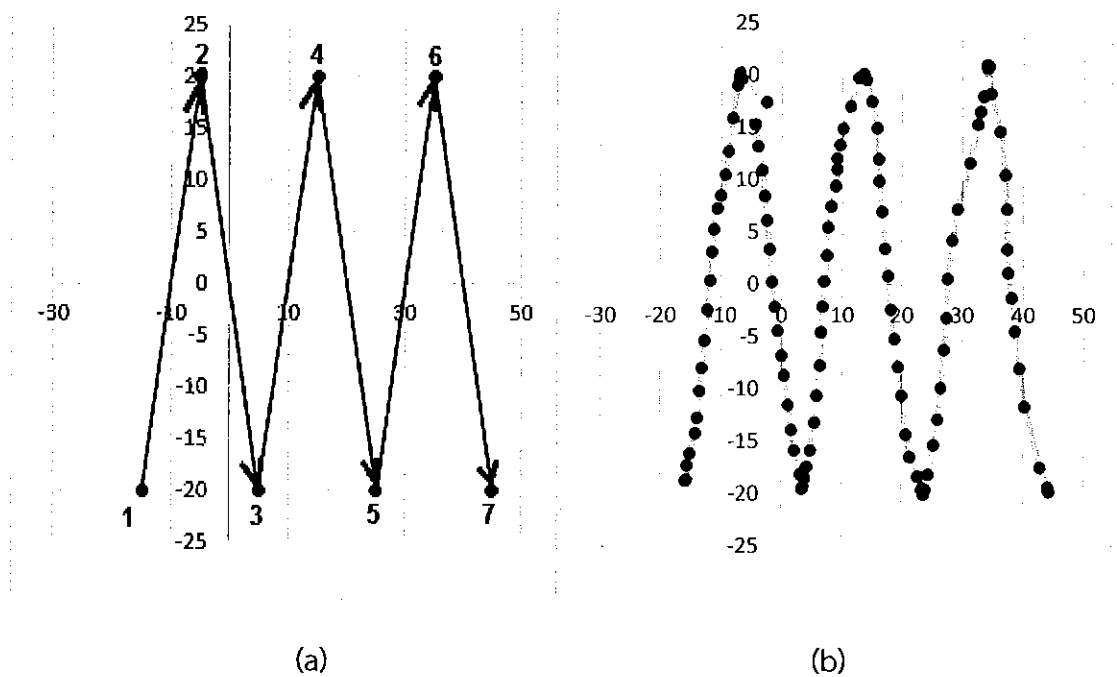
รูปที่ 4.15 การเคลื่อนที่ในลักษณะสี่เหลี่ยม

จากการทดลองพบว่า การบันทึกผลจะจุดจะใกล้หรือห่างกันขึ้นกับ ความเร็วของการเคลื่อนที่ Marker โดยถ้ามีความเร็วในการเคลื่อนที่มาก ระยะต่อจุดข้อมูลแต่ละจุดจะห่างมากขึ้น โดย ความคลาดเคลื่อนหรือความผิดพลาดของการตรวจจับตำแหน่งเกิดขึ้นบ้าง ซึ่งเป็นผลมาจากหลายสาเหตุคือ

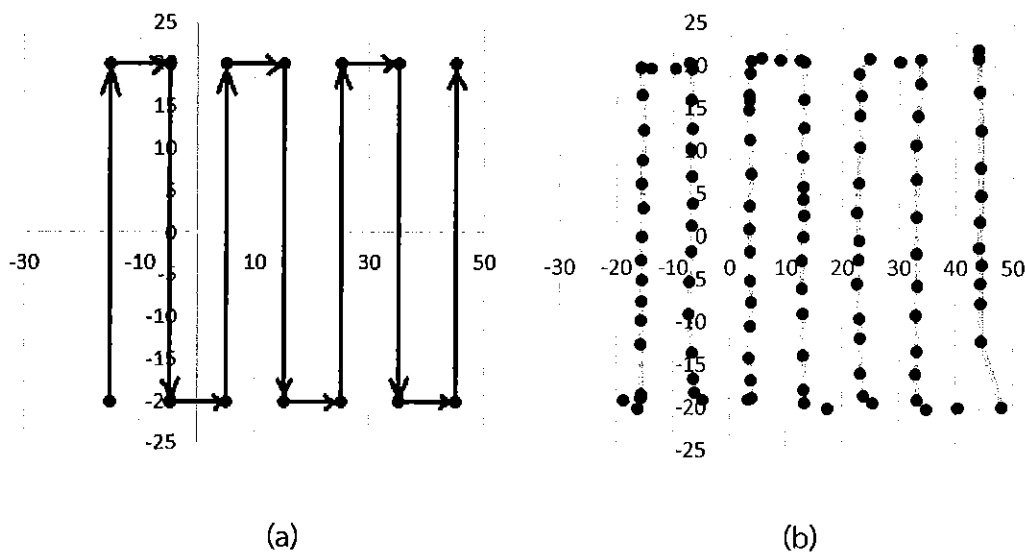
1. เวลาในการจับภาพ และตำแหน่ง การหมุน Step Motor เหลื่อมกัน

2. ความผิดพลาดเชิงระบบ ซึ่งเป็นผลจากความละเอียดกล้อง และแปรผันของสิ่งแวดล้อม เช่น ความสว่าง

3. ตัวผู้ทำการทดลอง ในการเคลื่อนตำแหน่ง Marker ไปตามจุดต่างๆ อาจคลาดเคลื่อนจากที่ ออกแบบการทดลองไว้



รูปที่ 4.16 การเคลื่อนที่ในลักษณะฟันปลา



รูปที่ 4.17 การเลื่อนที่ในลักษณะพื้นปลาทรงเหลี่ยม

จากการทดลองพบว่าค่าที่ได้จากการวัดในการทดลองมีรูปแบบที่ใกล้เคียงกับ การออกแบบ การทดลองที่เตรียมไว้และมีการผิดพลาดเล็กน้อยและในบางจุดจะพบการผิดพลาดที่ชัดเจนที่เกิดเช่น บริเวณ  $(-2.4, 17.4)$  ในรูปที่ 4.16 และ  $(48, -19.8)$  ในรูปที่ 4.17 ซึ่งอาจเกิดจากความผิดพลาดในการวางตำแหน่ง Marker ของผู้ทำการทดลองเอง

## บทที่ 5 สรุปโครงการ และข้อเสนอแนะ

### 5.1 สรุปผล

ในการพัฒนาระบบตรวจจับตำแหน่งวัตถุด้วยเทคโนโลยีภาพถ่ายในราคาที่เหมาะสมนี้ ทำให้ประสบความสำเร็จ โดยการใช้กล้องเว็บแคม ( Webcam ) จำนวนหนึ่งตัว ซึ่งมีราคาไม่แพงและมีความเร็วการจับภาพประมาณ 25 เฟรมต่อวินาที มาใช้ในการตรวจจับตำแหน่งของวัตถุ และมีการควบคุมทิศทางของกล้องด้วย Step Motor ประกอบกับการควบคุมด้วยบอร์ดไมโครคอนโทรลเลอร์ ซึ่งสั่งการผ่านโปรแกรมที่พัฒนาขึ้นด้วยภาษา LabVIEW

ในการศึกษาปัจจัยที่มีผลต่อตรวจจับวัตถุด้วยระบบ Image Plane นั้นในงานวิจัยนี้ ใช้การแยกแยะด้วยระบบสี HSL ซึ่งระบบที่พัฒนาขึ้นนี้สามารถหมุนทิศทางของให้หันไปตามวัตถุได้จริงตามวัตถุประสงค์ที่ตั้งไว้ และสามารถระบุตำแหน่งของวัตถุเป้าหมายได้โดยการคำนวณ อย่างถูกต้อง โดยความผิดพลาดไม่เกิน 1 cm ที่ระยะนำวัตถุห่างจากกล้อง 100 cm

### 5.2 ข้อเสนอแนะ

5.2.1 ในการทดลองควรออกแบบให้มีการตรวจจับวัตถุด้วยกล้องในมุมมองต่างๆจำนวนมากขึ้น เพื่อให้สามารถจับวัตถุในพิกัดแบบสามมิติได้

5.2.2 ในการทดสอบควรพัฒนาโปรแกรมที่มีประสิทธิภาพในการเลือกวัตถุให้มีความหลากหลายขึ้นเช่นการจับวัตถุรูปแบบวงกลม สีเหลี่ยม หรือตรวจจับใบหน้า เป็นต้น

5.2.3 ควรมีการเพิ่มความละเอียด step การหมุนโดยการเพิ่มชุดเกียร์ชุด ลงไปใน Step Motor เพื่อเพิ่มความละเอียดในการเข้าสู่เป้าหมายมากขึ้น

5.2.4 ในการทดลองพบว่าคุณภาพของกล้องเป็นตัวหลักในการได้ผลออกมาดีมาน้อยระดับใด ดังนั้นหากต้องการได้ชุดควบคุมที่ดีขึ้นจึงควรเพิ่มคุณภาพกล้องด้วย ทั้งนี้ควรเลือกกล้องที่มีอัตราการส่งถ่ายข้อมูลเร็วขึ้นด้วย เช่น 100 เฟรมต่อวินาที

## อ้างอิง

- [1] ณัฐพงษ์ ฐิติมานะกุล และรัชศักดิ์ สิทธิวิไล. (2550). กล้องดิจิทัล. กรุงเทพฯ: เสริมวิทย์ อินโฟเมชั่น.
- [2] ศักดิ์เศรษฐ์ ประกอบผล. (2553) การถ่ายภาพด้วยระบบดิจิทัล. กรุงเทพมหานคร: อีแมจโฟกัส
- [3] ผกา มาศ ผจญแก้ว. (2542). เทคโนโลยีเพื่อการผลิตภาพสี, การสัมมนาการผลิตภาพสี. นนทบุรี. ศูนย์ฝึกอบรมเทคโนโลยีการพิมพ์แห่งชาติ
- [4] จงรัก สามารถ. (6 ธันวาคม พ.ศ. 2548). Steping Motor. สืบค้นเมื่อ 11 ตุลาคม 2557, จาก <http://www.adisak51.com/home.html>
- [5] Alan G. Smith. (2011) Introduction to Arduino: New York. Coppyrighted Material
- [6] ทีมงานสมาร์ทเลิร์นนิ่ง. (2554) เริ่มต้นการใช้งาน LabVIEW. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง



### ประวัติผู้จัดทำ

ชื่อ - สกุล : นาย สราวุฒิ จิตต์ใจฉ่ำ

วัน เดือน ปี เกิด : 30 ตุลาคม พ.ศ. 2531  
ที่อยู่ปัจจุบัน : 25/2 หมู่ที่ ตำบล ไผ่ล้อม อำเภอ ลับแล จังหวัด อุตรดิตถ์ 53210

ประวัติการศึกษา

2559 : วศ.บ. (วิศวกรรมเครื่องกล) มหาวิทยาลัยนเรศวร

2551 : มัธยมศึกษา โรงเรียน จุฬารณราชวิทยาลัยพิษณุโลก