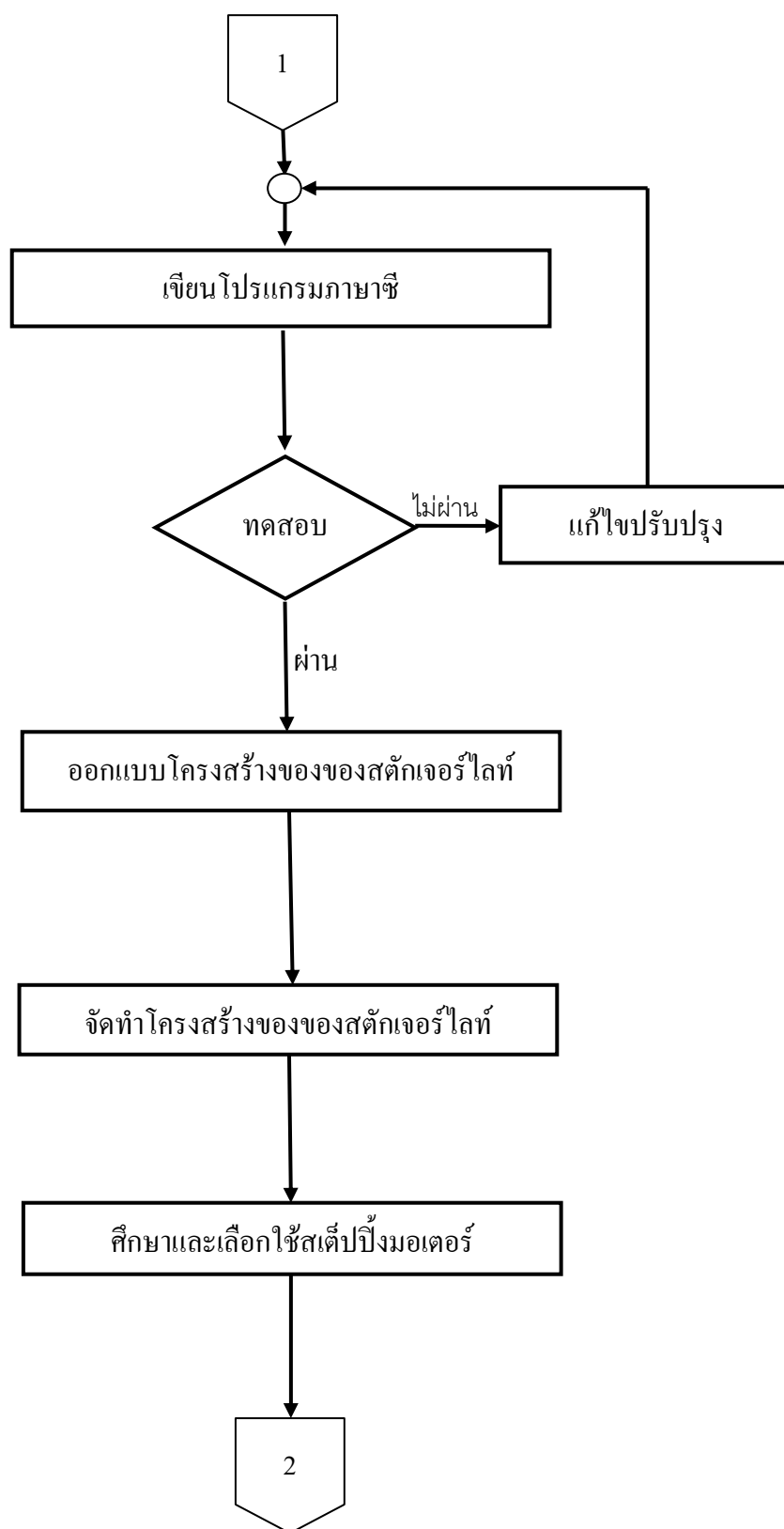
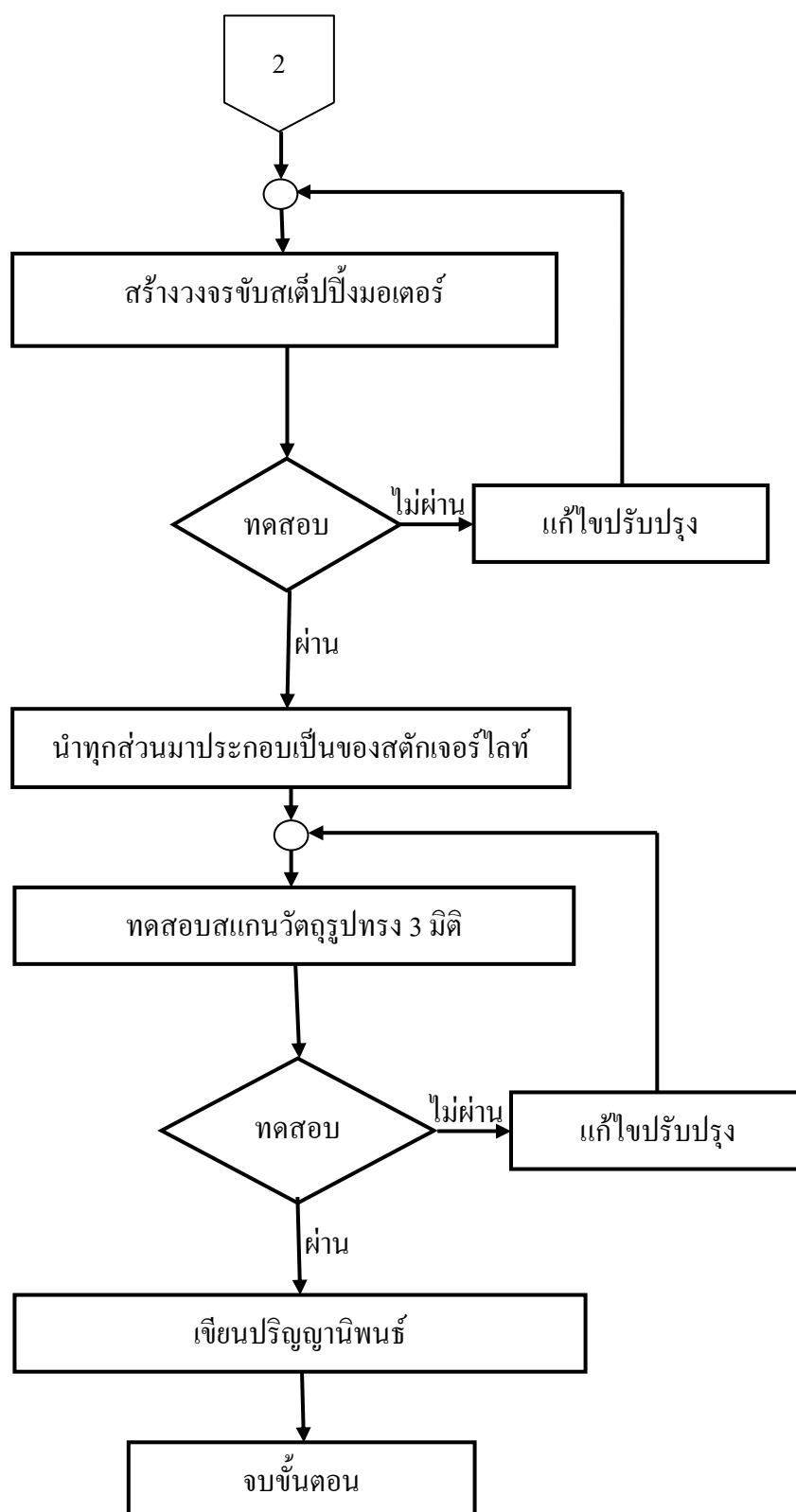


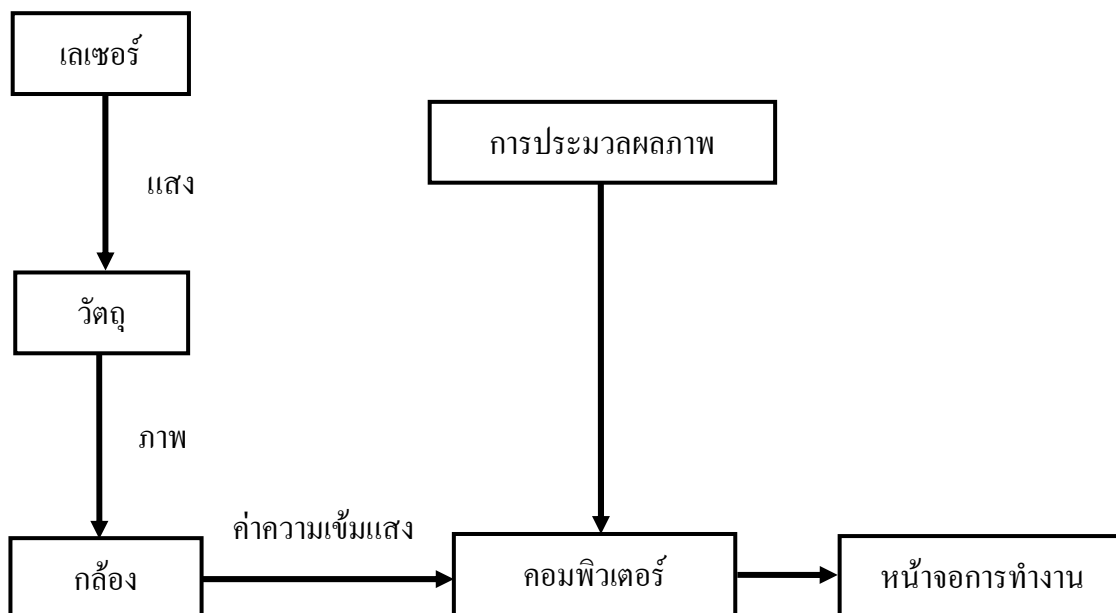


ภาพที่ 3.1 แผนผังการดำเนินงาน (ต่อ)



ภาพที่ 3.1 แผนผังการดำเนินงาน (ต่อ)

3.2 บล็อกไดอะแกรมแสดงหลักการทำงานของสตักเจอร์ไลท์



ภาพที่ 3.2 ระบบการทำงานของสตักเจอร์ไลท์

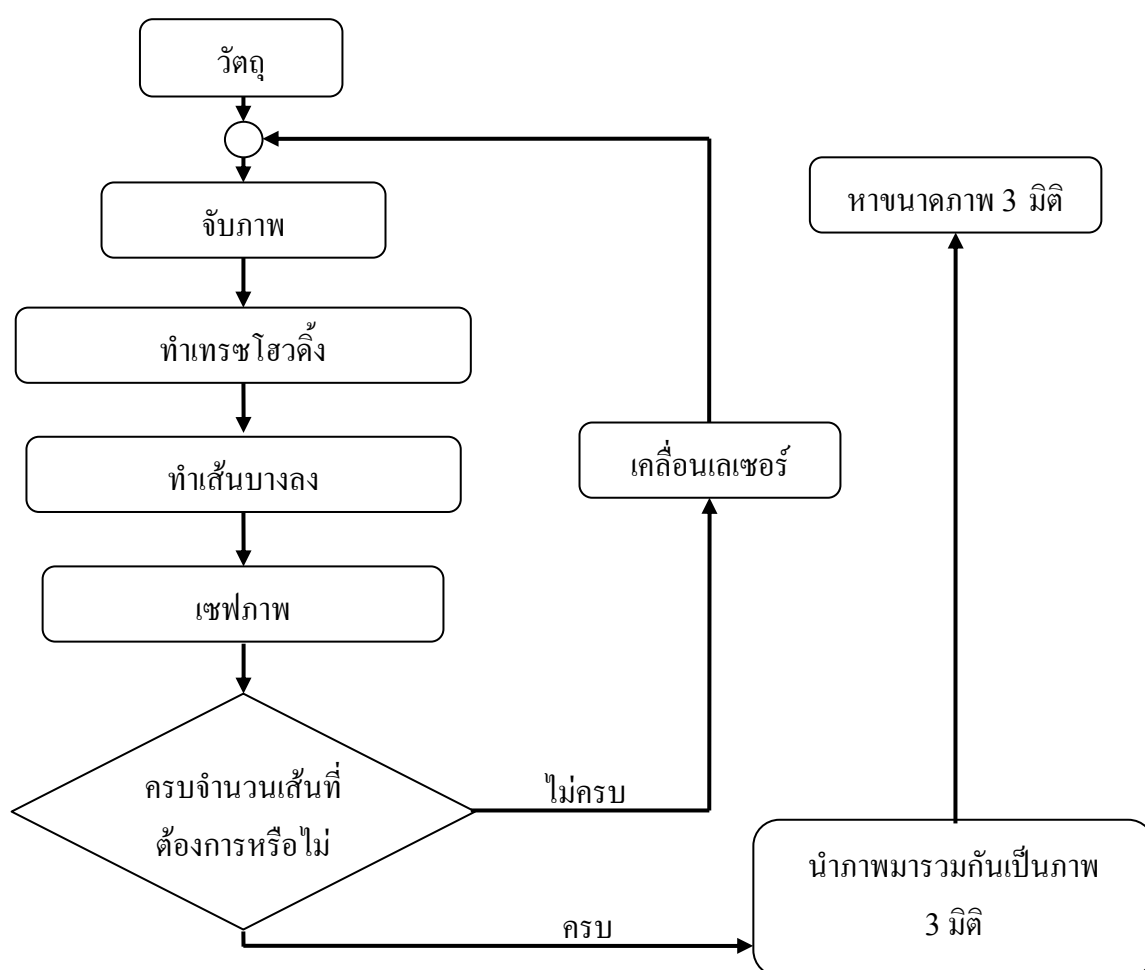
3.2.1 ส่วนที่ถ่ายภาพวัตถุจะเป็นกล้องเว็บแคม (Webcam) ซึ่งเป็นกล้องของบริษัท KER รุ่น OE – 017 ที่ติดต่อกับเครื่องคอมพิวเตอร์โดยผ่านพอร์ตยูเอสบี (USB 1.1 or 2.0) แต่ในกรณีที่จะถ่ายภาพขนาด 1 ล้านพิกเซล (Megapixel) จะต้องใช้ยูเอสบี 2.0 ไฮสปีด (USB 2.0 Hi-Speed) โดยความละเอียดจริงอยู่ที่ 640×480 พิกเซลและ สามารถจับภาพได้ 30 เฟรมต่อวินาที และสามารถเข้ากับระบบปฏิบัติการวินโดวส์เอ็กซ์พี ได้



ภาพที่ 3.3 เว็บแคม KER รุ่น OE - 017

3.2.2 ส่วนที่เป็นคอมพิวเตอร์ คือ ตัวควบคุมที่สั่งให้ระบบทำงานหรือประมวลผลค่าที่ต้องการซึ่งจะเป็น ซอฟต์แวร์ที่เขียนรหัสโปรแกรมเพื่อนำไปสั่งงานสแต็ปมอเตอร์ และซอฟต์แวร์ที่เกี่ยวกับการประมวลผลภาพ ที่ใช้ในการประมวลผลสตักเจอร์ไลท์ จะใช้ไลบรารีของโอเพ่นซีวี ที่ใช้ร่วมกับภาษาซีในการประมวลผลภาพ

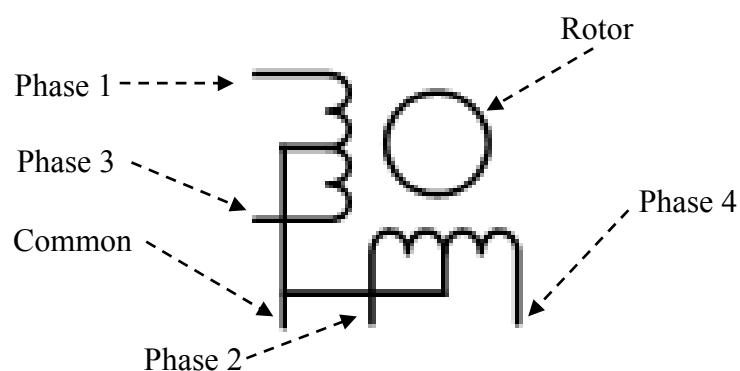
การทำงานของโปรแกรมในการประมวลผลภาพและอุปกรณ์ต่างๆ นั้นในการใช้งานระบบจะมีการทำงานร่วมกันระหว่างส่วนของการถ่ายภาพวัตถุ และส่วนของการเคลื่อนที่ของแสง จากนั้นจึงจะนำภาพถ่ายที่ได้ไปทำการประมวลผลเพื่อให้ได้เป็นโครงสร้าง 3 มิติ คือเมื่อได้ภาพของวัตถุแล้วจะนำไปทำการ เทรชโฮลดิ้ง (Thresholding) เพื่อหาเส้นขอบของวัตถุ จากนั้นก็ทำให้เส้นบางลง (Thinning) แล้วทำการเก็บค่าความเข้มแสงของแต่ละจุดไว้ในอาร์เรย์ซึ่งเป็นค่า 0 กับ 255 หรือ ดำ กับ ขาว เท่านั้น (ซึ่งผ่านการทำเทรชโฮลดิ้งแล้ว) จนครบรอบที่ต้องการเก็บจากนั้นจึงนำภาพที่ได้ไปรวมเพื่อสร้างเป็นตัวแบบ 3 มิติ และหาขนาดโดยกระบวนการทั้งหมดได้แสดงดังภาพที่ 3.4



ภาพที่ 3.4 แสดงระบบการทำงานของสตักเจอร์ไลท์

3.2.3 ส่วนที่เคลื่อนวัตถุ จะใช้สเต็ปปีงมอเตอร์เป็นตัวส่งกำลังให้บอลสกรูเคลื่อนที่ซึ่งเป็นการเคลื่อนเลเซอร์โดยมีตัววงจรขับเคลื่อนสเต็ปปีงมอเตอร์ และ คอมพิวเตอร์เป็นตัวควบคุมระยะในการเคลื่อนที่ของเลเซอร์

3.2.3.1 สเต็ปปีงมอเตอร์ จากข้อมูลในบทที่ 2 กล่าวถึงสเต็ปปีงมอเตอร์ ว่าสเต็ปปีงมอเตอร์ ที่ใช้ในโครงงานนั้นเป็นชนิด ยูนิโพลาร์ (Uni-Polar) 4 เฟส 5 สาย ซึ่งมีลักษณะการพันขดลวด ดังภาพที่ 3.5 โครงสร้างภายนอกของสเต็ปปีงมอเตอร์ เป็นดังภาพที่ 3.6 และ ค่าพารามิเตอร์ของสเต็ปปีงมอเตอร์ที่ใช้ในโครงงานแสดงดังตารางที่ 3.1



ภาพที่ 3.5 สเต็ปปีงมอเตอร์แบบ ยูนิโพลาร์



ภาพที่ 3.6 สเต็ปปีงมอเตอร์ ที่เลือกใช้ขับเคลื่อนบอลสกรู

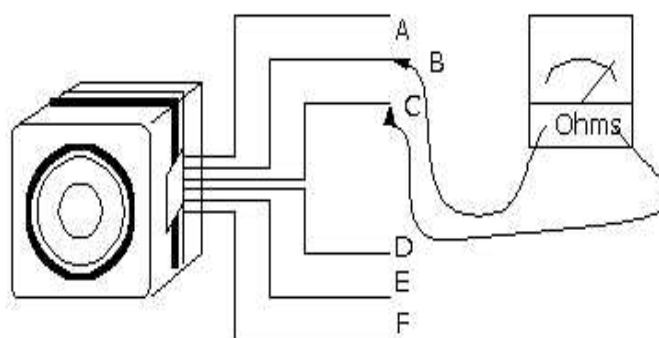
ตารางที่ 3.1 ค่าพารามิเตอร์ของสเต็ปปีงมอเตอร์ที่ใช้ในโครงการ

ลำดับที่	รายการ	ค่าพารามิเตอร์	หน่วย
1	มุมต่อ 1 เฟส (Step Angle)	1.8	องศา
2	ชนิดสเต็ปปีงมอเตอร์ (Drive Sequence)	ยูนิโพลาร์	-
3	กระแส (Reted Current)	0.96	แอมแปร์
4	ความต้านทาน (Resistance)	24	โอห์ม
5	แรงดัน (Voltage)	8.5	โวลต์
6	แรงบิด (Torque)	-	นิวตันเมตร
7	มวล (Mass)	-	กรัม

การตรวจสอบหาเฟสของขดลวด สเต็ปปีงมอเตอร์ มีอยู่ 3 ขั้นตอน

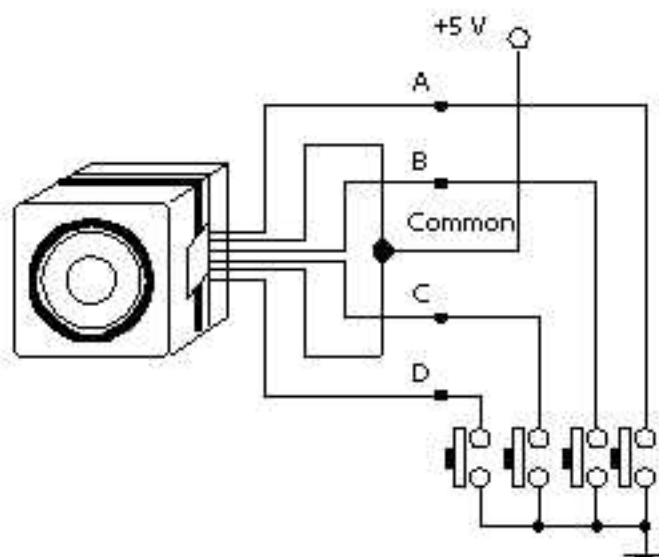
ขั้นตอนที่ 1 ให้สังเกตว่า สเต็ปปีงมอเตอร์ที่นำมาทดลองที่เป็นแบบยูนิโพลาร์จะมีจำนวนสาย 5 เส้น หรือ 6 เส้น

ขั้นตอนที่ 2 ใช้มิเตอร์วัดค่าความต้านทานของเส้นลวดในแต่ละขด ดังภาพที่ 3.7 ขั้นตอนการวัดให้หาสายที่ต่อเป็นจุดร่วมเสียก่อน (Common) โดยให้ใช้ มัลติมิเตอร์ ตั้งค่าไว้สำหรับการวัดค่าความต้านทานแต่ละเส้น สังเกตที่ค่าความต้านทานถ้าหากไม่ได้วัดระหว่างจุดต่อร่วมกับสายแต่ละเส้น ค่าความต้านทานจะมีค่าเป็น 2 เท่าของการวัดระหว่างจุดต่อร่วมกับสายที่ใช้งาน ตัวอย่างเช่นถ้าให้จุด B เป็นจุดร่วม หากวัดระหว่างที่จุด A กับจุด B จะมีค่าเท่ากับ 24 โอห์ม แต่ถ้าวัดระหว่างที่จุด A และจุด C ซึ่งไม่ใช่จุดร่วมก็จะได้ค่าเท่ากับ 48 โอห์ม หากเป็นแบบที่มีสาย 6 เส้นก็จะมีจุดร่วมสองจุด เพราะมีขดลวดคนละชุดกัน และสายที่เป็นจุดร่วมส่วนใหญ่จะมีสีเหมือนกัน และถ้าเป็นแบบ 5 สายเส้นก็จะมีจุดร่วมเพียงจุดเดียวเท่านั้น



ภาพที่ 3.7 แสดงการใช้มิเตอร์วัดค่าความต้านทานของเส้นลวดในแต่ละขด

ขั้นตอนที่ 3 หากเป็นแบบที่มีสาย 6 เส้นก็ให้ทำการต่อจุดร่วมเข้าด้วยกันจะได้เป็น 5 เส้น แล้วต่อวงจรตาม ภาพที่ 3.8 หลังจากนั้นทดลองกดสวิตช์ ที่ต่อเข้ากับแต่ละจุดโดยเริ่มที่จุด A จุด B จุด C และ จุด D แล้วให้สังเกตการณ์หมุนของสเต็ปปีงมอเตอร์ว่าหมุนได้ต่อเนื่องหรือไม่ หากมีการกระโดดข้ามสเต็ปก็ให้ทดลองโดยเรียงลำดับการกดสวิตช์ใหม่ จนหาลำดับของสเต็ปปีงมอเตอร์ได้



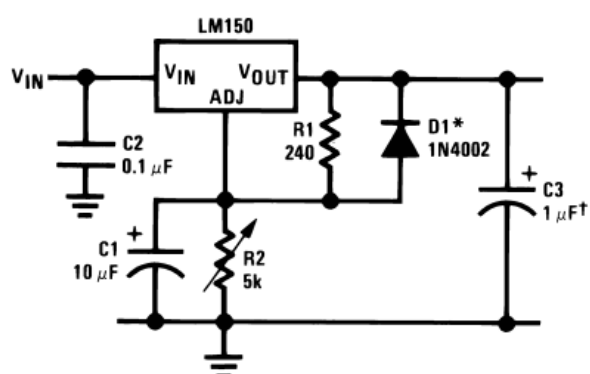
ภาพที่ 3.8 แสดงการต่อวงจรการใช้งานสเต็ปปีงมอเตอร์

3.2.3.2 การออกแบบวงจรขับ สเต็ปปีงมอเตอร์ จะแบ่งออกเป็น 2 ส่วน คือส่วนที่ใช้ไอซี กับ วงจรที่ใช้เพิ่มกระแสโดยใช้ทรานซิสเตอร์ชนิดพีเอ็นพี เพื่อให้สามารถจ่ายกระแสไฟฟ้าเพียงพอต่อความต้องการของ สเต็ปปีงมอเตอร์ และสามารถใช้เป็นตัวควบคุมการจ่ายไฟเข้าสเต็ปปีงมอเตอร์ โดยอธิบายไว้ในบทที่ 2 ซึ่งได้แสดงไว้ในภาพที่ 3.9 และมีชุดเรกติไฟเออร์ (Rectifier) และ ชุดเรกกูเลเตอร์ (Regulator) ในการแปลงสัญญาณจากแรงดันกระแสไฟฟ้ากระแสสลับ (AC) เป็นแรงดันกระแสไฟฟ้ากระแสตรง (DC) ต่อไปด้วยในบอร์ดเดียวกันกับวงจรขับสเต็ปปีงมอเตอร์

วงจรรีگูเลเตอร์ ซึ่งสามารถปรับแรงดันได้ตั้งแต่ 3 โวลต์ ถึง 35 โวลต์ และจ่ายกระแส 3 แอมแปร์ สามารถคำนวณหาค่าแรงดันเอาต์พุต (V_{out}) ได้ดังต่อไปนี้

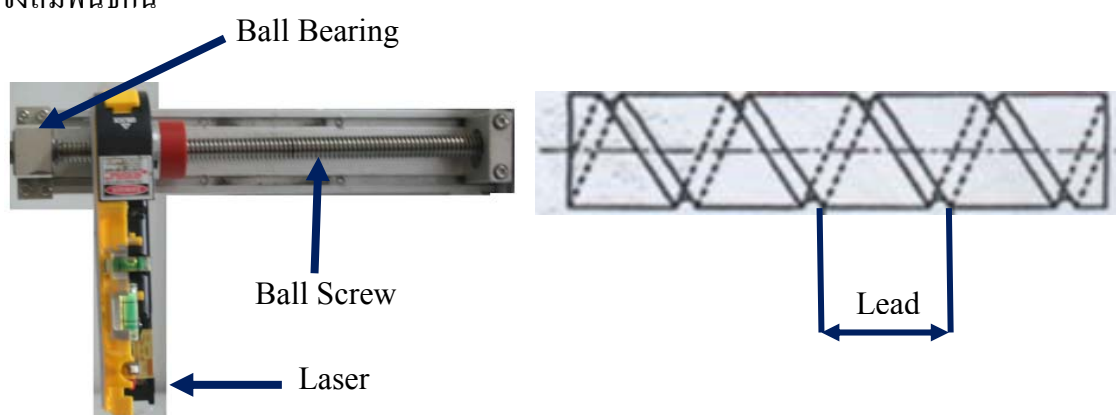
$$V_{out} = 1.25V \left(1 + \left(\frac{R2}{R1}\right)\right) + I_{ADJ}(R2) \quad (3-1)$$

สามารถหาค่า $R1$ $R2$ ได้จากภาพที่ 3.10

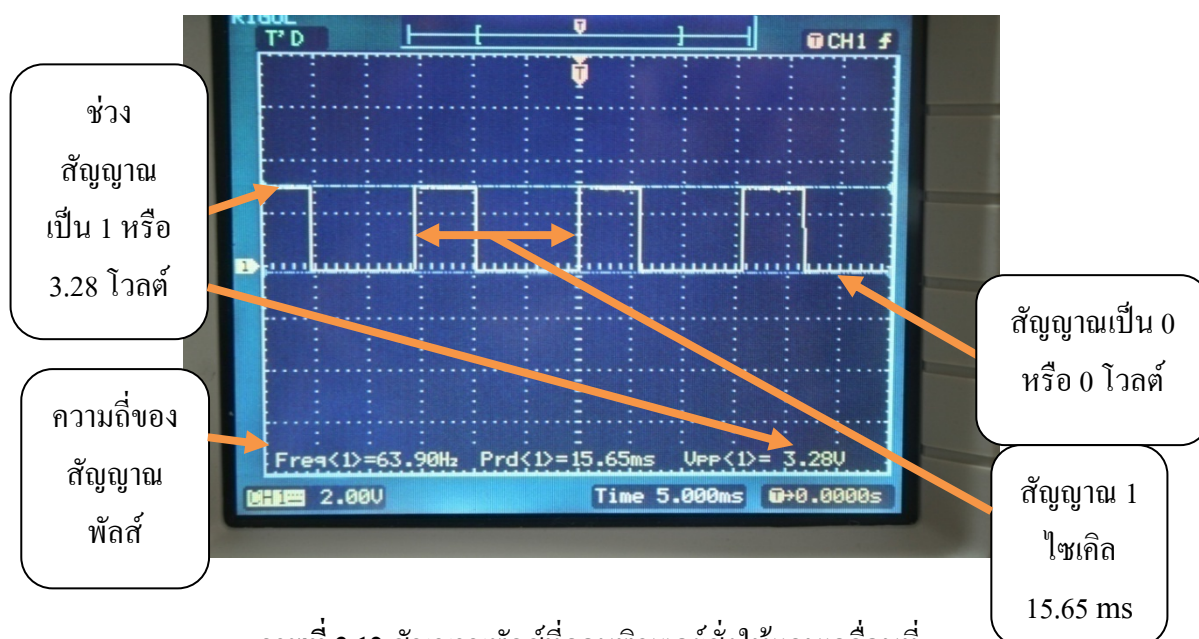


ภาพที่ 3.10 วงจรรีกูเลเตอร์

3.2.3.3 การคำนวณหาความเร็วของการเคลื่อนที่ของบอลสกรู นั้นเป็นความเร็วเชิงเส้น ซึ่งถูกควบคุมโดยคอมพิวเตอร์ ดังนั้นความเร็วในการเคลื่อนที่ของ บอลสกรูกับสัญญาณคอมพิวเตอร์ จึงสัมพันธ์กัน



ภาพที่ 3.11 แสดงลักษณะของบอลสกรู



ภาพที่ 3.12 สัญญาณพัลส์ที่คอมพิวเตอร์สั่งให้แกนเคลื่อนที่

จากภาพที่ 3.12 เป็นสัญญาณคอมพิวเตอร์ที่สั่งให้บอลสกรูเคลื่อนที่ซึ่งใน 1 ช่วงพัลส์ต่อ นาที มีความเร็วเท่ากับ 15.65 วินาที

เมื่อสเต็ปปิ้งมอเตอร์ทำงานแบบฮาร์ฟสเต็ปหมุน 1 รอบ ต่อ 400 สเต็ป

$$\begin{aligned}
 \text{ดังนั้นความเร็วรอบของมอเตอร์} &= \frac{1 \text{ rev}}{400 \text{ step}} \times \frac{1 \text{ step}}{15.65 \text{ ms}} \times \frac{100 \text{ ms}}{1 \text{ s}} \times \frac{60 \text{ s}}{1 \text{ min}} \\
 &= 9.585 \text{ รอบต่อนาที } \left(\frac{\text{rev}}{\text{min}} \right)
 \end{aligned} \tag{3-2}$$

เมื่อใส่เฟืองทดขนาด 40 ต่อ 49 จะได้สมการดังนี้

เมื่อ X คือ ความเร็วรอบ (รอบต่อนาที)

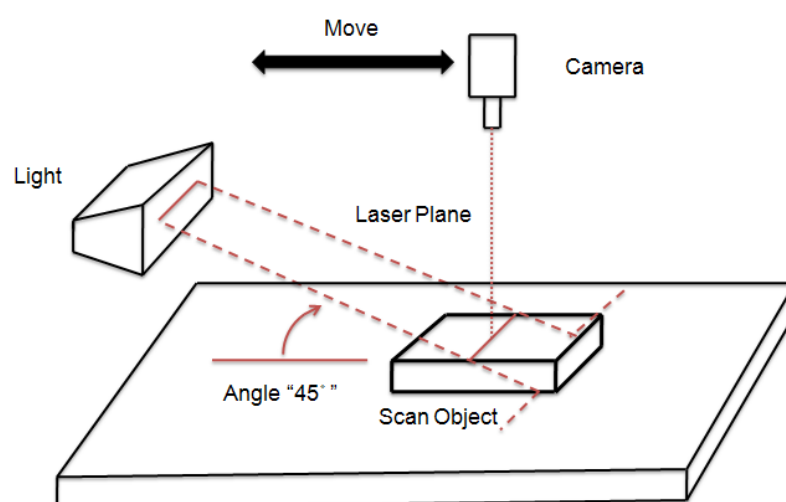
$$\begin{aligned}
 \frac{9.585}{40} &= \frac{X}{49} \\
 X &= \frac{9.585}{40} \times 49 \\
 &= 11.742 \text{ รอบต่อนาที } \left(\frac{\text{rev}}{\text{min}} \right)
 \end{aligned} \tag{3-3}$$

$$\text{เมื่อบอลสกรูมีระยะ (Lead)} = 2.5 \text{ มิลลิเมตร (mm)}$$

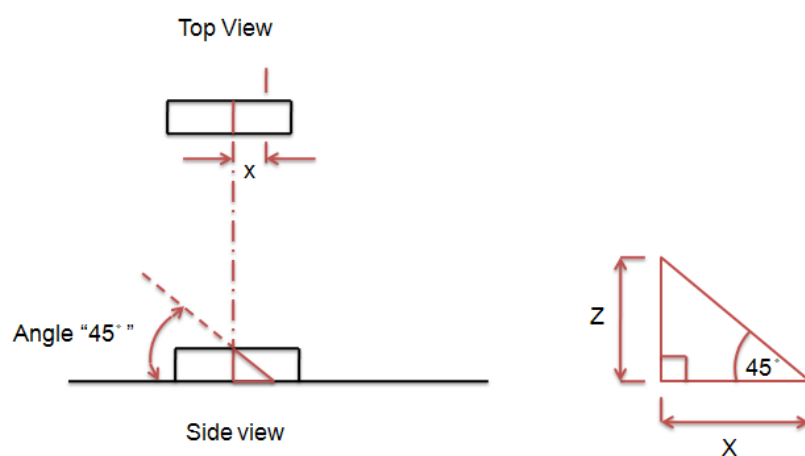
$$\text{การเคลื่อนที่ของบอลสกรูมีระยะการเคลื่อนที่} = \frac{2.5 \text{ mm}}{1 \text{ rev}}$$

$$\begin{aligned} \text{ดังนั้นบอลสกรูมีความเร็วในการเคลื่อนที่} &= \frac{2.5 \text{ mm}}{1 \text{ rev}} \times \frac{11.742 \text{ rev}}{1 \text{ min}} \\ &= 29.355 \text{ มิลลิเมตรต่อนาที } \left(\frac{\text{mm}}{\text{min}} \right) \end{aligned}$$

3.2.4 การคำนวณหาขนาดของภาพที่สแกน ซึ่งการหาขนาดนั้นจะใช้กฎของสามเหลี่ยม ที่ทำมุม 45 องศา ที่แสดงในภาพ 3.13 และ ภาพที่ 3.14



ภาพที่ 3.13 แสดงการทำงานของสติกเกอร์ไลท์



ภาพที่ 3.14 แสดงการทำมุม 45 องศา ของชิ้นงานกับแสงเลเซอร์

เมื่อทำมุม 45 องศา จะสามารถเข้าสู่สูตรได้โดยการคำนวณดังนี้

Z หมายถึงค่าความสูงของชิ้นงาน (มิลลิเมตร)

$$\tan 45^\circ = \frac{Z}{X} \quad (3-4)$$

จากสมการ 3-4 นั้น $\tan 45^\circ$ จะเท่ากับ 1 ดังนั้น ค่า X จะเท่ากับ Z ถ้าหาค่า X ได้ก็จะสามารถหาค่า Z ได้ แต่ก่อนจะหาค่า Z เราต้องรู้ค่า X ก่อนโดยค่า X จะบอกเป็นพิกเซล ซึ่งค่า X จะบอกได้ในการประมวลผลภาพดังนั้นเราก็จะรู้จำนวนพิกเซลของ X ไม่สามารถรู้ได้ว่า X ที่เป็นหน่วยพิกเซลนั้นมีค่าเท่ากับความสูงของขนาดจริงหรือเปล่าตามสมการที่ 3-4 จึงต้องเปรียบเทียบระหว่างพิกเซล กับขนาดที่มีหน่วยเป็นมิลลิเมตร แต่เนื่องจากขนาดของภาพมีความกว้าง 640 พิกเซล ความยาว 480 พิกเซลนั้นโดยอาจจะวัดด้วยไม้บรรทัดหรือเวอร์เนียจากการวัดค่านั้นจะแทนตัวแปรเป็น A จะได้สูตรดังนี้

X หมายถึงจำนวนพิกเซลที่วัดได้จากโปรแกรม

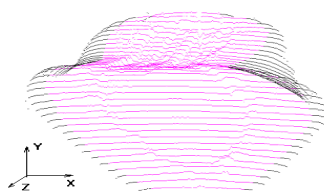
A หมายถึงขนาดความยาว 480 พิกเซล ที่วัดจากไม้บรรทัดหรือสายวัด

X_m หมายถึงค่าขนาดจริงของวัตถุที่ต้องการทราบเป็นมิลลิเมตร

$$\frac{480 \text{ pixel}}{A \text{ mm}} = \frac{X \text{ pixel}}{X_m}$$

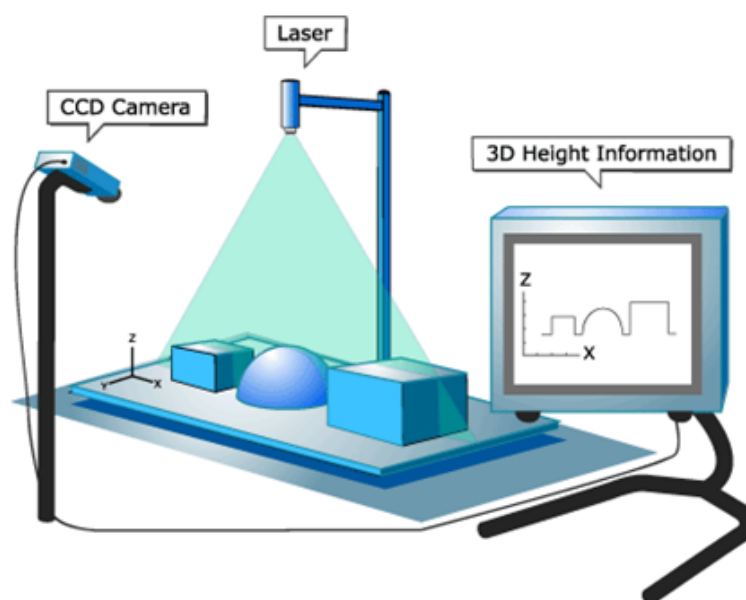
$$X_m = \frac{X \times A}{480} \quad (3-5)$$

จากสมการที่ 3-5 สามารถหาค่าขนาดจริงของวัตถุ (X_m) ที่มีค่าเป็นมิลลิเมตรได้แล้วและนำมาเปรียบเทียบกับวัตถุว่าขนาดที่หาได้จากการคำนวณมีขนาดความสูงเท่ากับขนาดของวัตถุที่วัดได้จากเครื่องมือวัดหรือไม่ ถ้าได้เท่ากับวัตถุจริงก็สามารถรู้ได้ว่าภาพที่นำมาทำเป็นภาพ 3 มิตินั้นเป็นขนาดจริงแต่อยู่ในรูปของพิกเซล



ภาพที่ 3.15 การรวมเส้นของแสงเลเซอร์

3.2.5 วัตถุรูปทรงเรขาคณิต ซึ่งในโครงการนี้จะใช้วัตถุที่เป็นรูปทรงเรขาคณิตอย่างง่ายคือ ไม่มีความซับซ้อนมาก ดังภาพที่ 3.16



ภาพที่ 3.16 การสแกนรูปทรงเรขาคณิตอย่างง่าย

3.2.6 เลเซอร์ เป็นแสงที่นำมาตกกระทบกับวัตถุเพื่อให้เกิดเป็นรูปภาพเหมือนกับการสแกนรูปภาพหรือการทำสำเนาวัตถุนั้นๆ ซึ่งจะประหยัดเวลาในการเขียนแบบงานหรือการตรวจสอบขนาดคุณภาพ ความผิดพลาด และการคัดแยกผลผลิต ซึ่งเลเซอร์ที่ใช้นั้นจะแสดงดังภาพที่ 3.17



ภาพที่ 3.17 เลเซอร์ที่ใช้ในโครงการ

3.3 การทำงานของโปรแกรม

จะเป็นการทำงานแบบอัตโนมัติซึ่งเมื่อเราเซตค่าต่างๆแล้วเราก็กดปุ่มอัตโนมัติเครื่องก็จะทำงานเองแต่ในการอธิบายโปรแกรมนั้นจะเป็นไปทีละขั้นตอนดังนี้

3.3.1 การประกาศไลบรารีโอเพ่นซีวี พอร์ตขนานและประกาศจำนวนเต็มดังนี้

```
#include "windows.h"                                     ประกาศไลบรารีใช้ พอร์ตพรีนเตอร์
#define portprint 0x378
```

```
.....
#include <iostream.h>                                     ประกาศไลบรารีโอเพ่นซีวี
#include <cv.h>
#include <cvaux.h>
#include <highgui.h>
#include <cvcam.h>
.....
```

3.3.2 การฉายภาพออกกล้องเว็บแคม ประกาศพอยต์เตอร์ (Pointer) สร้างหน้าต่างเพื่อรับภาพจากกล้องและ แปลงภาพสี เป็นโทนสีเทา

```
CvCapture*capture=cvCaptureFromCAM(0);                 ประกาศพอยต์เตอร์สำหรับ
IplImage *img10=0,*gray10=0;                           กล้องและภาพ
```

```
.....
while (1){
    cvGrabFrame(capture);                                เก็บภาพไว้ในตัวแปรพอยต์เตอร์
    img10 = cvRetrieveFrame(capture);
    gray10 = cvCloneImage( img10 );
    gray10=cvCreateImage(cvSize(img10->width,img10->height),IPL_DEPTH_8U,1);
    .....
    gray10->origin = img10->origin;                       กำหนดให้ภาพอยู่ตำแหน่งเริ่มต้น
    .....
    cvCvtColor (img10,gray10,CV_BGR2GRAY);               แปลงภาพสีเป็นภาพโทนสีเทา
    .....
    cvNamedWindow("3D",CV_WINDOW_AUTOSIZE);
    cvShowImage("3D",gray10);
    .....
}                                                         ทำการสร้างหน้าต่างสำหรับแสดงภาพ
.....
```

```
cvDestroyWindow("3D");                                   ทำการคืนค่าทรัพยากรต่างๆ ให้แก่ระบบโดยสั่งปิดหน้าต่าง
cvReleaseCapture(&capture);                             โปรแกรมและสั่งคืนข้อมูลในตัวแปร capture เรียกมาใช้งาน
.....
```

3.3.3 การแยกวัตถุออกจากพื้นหลัง (Thresholding) เพื่อทำให้เห็นวัตถุเด่นชัดขึ้นมา

for(i=0;i<img10->height;i++)	วนลูปค่าความกว้างของภาพและความยาวของภาพ
{	
for(j=0;j<img10->width;j++)	
{	
.....	
CvScalar s;	ทำหน้าที่เก็บค่าความสว่างของจุดสี
s = cvGet2D(gray10,i,j);	ณ ตำแหน่ง (i,j) นั้นๆ ไว้
.....	
if (s.val[0]<80)	ถ้าค่าความสว่างของจุดสีที่รับมามีค่าน้อยกว่า
{	80 ฟังก์ชัน cvSet2D จะทำหน้าที่เปลี่ยนค่า
s.val[0]=0;	ความสว่างของจุดสี ณ ตำแหน่ง (i,j) นั้นๆ
cvSet2D(gray10,i,j,s);	เป็น 0 (หรือสีดำ) แต่ถ้ามากกว่าให้เป็นค่า 255
}	(หรือสีขาว)
else	
{	
s.val[0]=255;	
cvSet2D(gray10,i,j,s);	
}	
}	
}	
.....	

3.3.4 การทำให้เส้นบางลง (Thinning) วิธีการทำให้เส้นบางลงนั้นสามารถทำได้โดยการนำหน้ากาก (Karnel) ที่มีขนาด 3x3 ไปทำการวน (Convolution) ให้ทั่วทั้งภาพโดยในขณะที่กำลังวนในภาพไปตามจุดพิกเซล (Pixel) ต่างๆ ก็จะทำการพิจารณาพิกเซลบริเวณขอบภาพว่าพิกเซลไหนที่สามารถลบได้แต่จะต้องไม่ได้ลบออกจริงๆ โดยจะตรวจสอบก่อนว่าพิกเซล นั้นเป็นจุดปลายของเส้นขอบหรือไม่ และถ้าลบพิกเซลนั้นออกไปแล้วจะต้องไม่ทำให้คุณลักษณะเดิมของภาพเสียไป เมื่อทำให้เส้นบางลงแล้วก็ทำการเก็บภาพไว้ในอาร์เรย์จนหมดเส้นที่ต้องการ แล้วรวมเส้นและเซฟภาพ

for(i=2;i<(480-2);i++)	โปรแกรมเข้าสู่การทำงานวนลูปในการทำให้เส้นของแสงเลเซอร์
{	ที่ตกกระทบวัตถุบางลงโดยจะประมวลผลทางแนวนอน
for(j=2;j<(640-2);j++)	
{	
CvScalar s,s1,s2,s3,s4;	ประกาศตัวแปร s, s1, s2, s3 และ s4 ไว้รับค่าพิกเซลอ้างอิง
.....	

```

s = cvGet2D(gray,i-1,j);      ฟังก์ชัน cvGet2D ทำหน้าที่เก็บค่าความสว่างของจุดสี ณ
s1 = cvGet2D(gray,i-2,j);     ตำแหน่ง (i,j) อ้างอิงนั้นๆ ไปยังตัวแปรที่กำหนดไว้
s2 = cvGet2D(gray,i,j);
s3 = cvGet2D(gray,i+1,j);
s4 = cvGet2D(gray,i+2,j);

.....

F = (s1.val[0]+s4.val[0])+(2*(s.val[0]+s3.val[0]))+(3*s2.val[0]);
rray[i][j]=F;
}}                               คำนวณค่าตามตำแหน่งหน้าภาพ แล้วนำผลลัพธ์ที่ได้มาเก็บ
                                ไว้ที่ตัวแปร F จากนั้นก็เอาไปเก็บไว้ที่ตัวแปรอารีย์

.....

for(i=2;i<(480);i++)
{
for(j=0;j<(640-2);j++)
{
    A = rray[i-1][j];
    B = rray[i-2][j];
    C = rray[i][j];
    D = rray[i+1][j];
    E = rray[i+2][j];
    if(C>A && C>B && C<D && C<E)           เปรียบเทียบเพื่อหาขอบภาพของวัตถุ
                                                บริเวณฝั่งซ้าย
    {
        CvScalar s;
        s.val[0]=0;                          กัดฟิสิกเซลบริเวณขอบวัตถุฝั่งซ้าย
        cvSet2D(gray,i,j,s);
    }

.....

    if(C<A && C<B && C>D && C>E)           เปรียบเทียบเพื่อหาขอบภาพของวัตถุ
                                                บริเวณฝั่งขวา
    {
        CvScalar s;
        s.val[0]=0;                          กัดฟิสิกเซลบริเวณขอบวัตถุฝั่งขวา
        cvSet2D(gray,i,j,s);
    }
}}

.....

for(i=2;i<(480-2);i++)          โปรแกรมเข้าสู่การทำงานวนลูปในการทำให้เส้นของแสงเลเซอร์
{                                ที่ตกกระทบวัตถุบางลงโดยจะประมวลผลทางแนวตั้ง
for(j=2;j<(640-2);j++)
{
    CvScalar s,s1,s2,s3,s4;      ประกาศตัวแปร s, s1, s2, s3 และ s4 ไว้รับค่าฟิสิกเซลอ้างอิง
}
}

```


s = cvGet2D(gray,i,j-1);	ฟังก์ชัน cvGet2D ทำหน้าที่เก็บค่าความสว่างของจุดสี ณ
s1 = cvGet2D(gray,i,j-2);	ตำแหน่ง (i,j) อ้างอิงนั้นๆ ไปยังตัวแปรที่กำหนดไว้
s2 = cvGet2D(gray,i,j);	
s3 = cvGet2D(gray,i,j+1);	
s4 = cvGet2D(gray,i,j+2);	
G = (s1.val[0]+s4.val[0])+(2*(s.val[0]+s3.val[0]))+(3*s2.val[0]);	
tarray[i][j] = G;	
}}	คำนวณค่าตามตำแหน่งหน้าภาพ แล้วนำผลลัพธ์ที่ได้มาเก็บ
	ไว้ในตัวแปร G จากนั้นก็เอาไปเก็บไว้ในตัวแปรอาเรย์
for(i=0;i<(480);i++)	
{	
for(j=2;j<(640-2);j++)	
{	
A = rray[i][j-1];	
B = rray[i][j-2];	
C = rray[i][j];	
D = rray[i][j+1];	
E = rray[i][j+2];	
if(C>A && C>B && C<D && C<E)	เปรียบเทียบเพื่อหาขอบภาพของวัตถุ
{	บริเวณด้านบน
CvScalar s;	
s.val[0]=0;	กัศพิกเซลบริเวณขอบวัตถุด้านบน
cvSet2D(gray,i,j,s);	
}	
if(C<A && C<B && C>D && C>E)	เปรียบเทียบเพื่อหาขอบภาพของวัตถุ
{	บริเวณด้านล่าง
CvScalar s;	
s.val[0]=0;	กัศพิกเซลบริเวณขอบวัตถุด้านล่าง
cvSet2D(gray,i,j,s);	
}}}	

3.3.5 การเขียนโปรแกรมเพื่อกระตุ้นเฟสแบบฮาล์ฟสเต็ป สั่งให้สเต็ปปิ้งมอเตอร์หมุนและเคลื่อนที่บอลสกรู ซึ่งการสั่งจะเป็นแบบ 8 บิต คือ 0000 0000 ถ้าจะให้สเต็ปปิ้งมอเตอร์เคลื่อนที่ โดยจะสั่ง 0x01 0x03 0x02 จนถึง 0x09 จะเป็นแบบ ฮาล์ฟสเต็ป อ่านคำอธิบายในบทที่ 2 หรือจะสั่งย้อนกลับ ก็เขียนโปรแกรมย้อนกลับคือ 0x09 0x08 0x0C จนถึง 0x01 ส่วนการเคลื่อนที่ไปที่ระยะเท่าไรหาก็มีลิเมตรนั้นจะใช้ for ในการบอกว่าจะเคลื่อนที่ไปเท่าไร

```
int a;
for (a = 0; a < 150; a++)
{
```

โปรแกรมเข้าสู่การทำงานวนลูปในการทำการเคลื่อนที่ของแสงเลเซอร์โดยโปรแกรมในที่นี่สั่งให้เคลื่อนที่ไป 6 มิลลิเมตร

```
.....
UpdateData (TRUE);
    _outp (0x378,0x09);
        Sleep(5);
    _outp (0x378,0x08);
        Sleep(5);
    _outp (0x378,0x0C);
        Sleep(5);
    _outp (0x378,0x04);
        Sleep(5);
    _outp (0x378,0x06);
        Sleep(5);
    _outp (0x378,0x02);
        Sleep(5);
    outp (0x378,0x03);
        Sleep(5);
    _outp (0x378,0x01);
        Sleep(5);
    UpdateData (FALSE);
}
```

เขียนโปรแกรมสั่งให้ทำงานแบบฮาล์ฟสเต็ปคู่ได้จากตารางที่ 2.3

3.3.6 การรวมภาพจะทำการเก็บค่าในตัวแปรอาเรย์ในแต่ละเส้นแล้วนำมาบวกกันถ้ามีค่าน้อยกว่า 255 ก็ให้เท่ากับ 0 แต่ ถ้ามากกว่า 255 ให้มีค่าเป็น 255 และเก็บไปจนครบจำนวนเส้นที่ต้องการก็จะเซฟภาพเพื่อเก็บไว้

```
for(i=0;i<480;i++) {
for(j=0;j<640;j++) {
```

โปรแกรมเข้าสู่การทำงานวนลูปในภาพเพื่อทำการรวมภาพให้เกิดเป็นภาพ 3 มิติ

<pre> CvScalar s; s = cvGet2D(gray01,i,j); A1 = rray1[i][j]; B1 = rray2[i][j]; C1 = A1 + B1; </pre>	นำภาพแรกที่ทำกรประมวลผลภาพมาเก็บไว้ในตัวแปรอาร์เรย์ rray1 และนำภาพที่สองที่ทำกรประมวลผลภาพมาเก็บในตัวแปรอาร์เรย์ rray2 และนำอาร์เรย์ทั้งสองมาบวกกัน
<pre> if(C1>=255) { s.val[0] = 255; A1 = s.val[0]; rray1[i][j] = A1; cvSet2D(gray01,i,j,s); } </pre>	นำค่าอาร์เรย์ที่บวกกันมาทำการเปรียบเทียบถ้าค่ามากกว่า 255 ก็จะให้เท่ากับ 255 (สีขาว)
<pre> if(C1<255) { s.val[0] = 0; A1 = s.val[0]; rray1[i][j] = A1; cvSet2D(gray01,i,j,s); }}} </pre>	นำค่าอาร์เรย์ที่บวกกันมาทำการเปรียบเทียบถ้าค่าน้อยกว่า 255 ก็จะให้เท่ากับ 0 (สีดำ)
<pre> cvSaveImage("Z.bmp",gray01); </pre>	ทำการเซฟภาพเมื่อครบจำนวนเส้นที่ต้องการและเป็นภาพ 3 มิติแล้ว