อภินันทนาการ

สำนักหอสมุด

การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็ก ด้วยตัวรับภาพแบบฟิล์มสกรีน

THE DEVELOPMENT OF COMPUTED TOMOGRAPHY SYSTEM FOR SMALL ANIMAL IMAGING USING FILM-SCREEN DETECTOR

นางสาวพรเพ็ญ บุญแสวง รหัส 52362038

สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน 17 ต.ค. 2560
เลขทะเบียน
เลขเรียกหนังสือ

ป
พ 352ก
2557

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2557

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน
ผู้ดำเนินโครงงาน	นางสาวพรเพ็ญ บุญแสวง รหัส 52362038
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร.สุชาติแย้มเม่น
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2557

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

.....ที่ปรึกษาร่วมโครงงาน
(ผู้ช่วยศาสตราจารย์ ดร.นันทวัฒน์ อู่ดี)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุวิทย์ กิระวิทยา)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พรพิศุทธิ์ วรจิรันตน์)

ชื่อหัวข้อโครงงาน	การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน
ผู้ดำเนินโครงงาน	นางสาวพรเพ็ญ บุญแสวง รหัส52362038
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2557

.....

บทคัดย่อ

โครงงานนี้ได้ศึกษาและออกแบบระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนเพื่อใช้การตรวจวินิจฉัยและติดตามผลการศึกษาในสัตว์ทดลองขนาดเล็กซึ่งมีการประมวลผลการทำงานของระบบผ่านไมโครคอนโทรลเลอร์ 16F886 โดยการควบคุมการทำงานของสเตปปิ้งมอเตอร์ ซึ่งมีหน้าที่ในการหมุนฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา และควบคุมการทำงานของดีซีมอเตอร์ซึ่งทำหน้าที่ในการเลื่อนแผ่นฟิล์มสกรีนขึ้นลงตามแนวดิ่ง โดยมีเอนโคดเดอร์ทำหน้าที่ในการตรวจวัดระยะการเลื่อนขึ้นลงตามแนวดิ่งของแผ่นฟิล์มสกรีน โดยระบบนี้สามารถรองรับสัตว์ทดลองที่มีขนาดเส้นผ่านศูนย์กลางไม่เกิน 3 เซนติเมตร และความสูงของสัตว์ทดลองไม่เกิน 32 เซนติเมตร จากผลการทดลองการทำงานของระบบพบว่าค่าความผิดพลาดเฉลี่ยในการเลื่อนแผ่นฟิล์มสกรีนขึ้นลงตามแนวดิ่งมีค่าอยู่ที่0.2 เซนติเมตร โดยที่ความคลาดเคลื่อนที่เกิดขึ้นกับระบบนั้นยอมรับค่าได้ และไม่มีผลต่อการทำงาน

Project title	The Development of Computed Tomography System for Small Animal Imaging Using Film-Screen Detector.
Name	Ms. Pornpen Boonsawaeng ID. 52362038
Project advisor	Assistant Professor. Suchart Yammen , Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2014

.....

Abstract

In this project, a computer x-ray system was studied and designed for small animal tomography with a film screening to diagnose and monitor small animal studies, where the stepper motor was operated by a microcontroller of 16F886 to rotate a vertical axis of small animal model from 0.9 degrees to 360 degrees, and the film screen was moved in the vertical direction by using the direct current motor including with the encoder to check the distance. The developed system can be used for an experimental animal whose diameter is less than 3 centimeters, and whose height is less than 32 centimeters.

From the experimental results of the developed system, it has been found that the average error in moving vertically the film screen (up and down) is 0.2 cm. This occurred error value is accepted, and has no effect on operation.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาเป็นอย่างยิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น เป็นอาจารย์ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ ดร.นันทวัฒน์ อู่ดีซึ่งเป็นอาจารย์ที่ปรึกษาร่วม และให้ความกรุณาในการตรวจทานปริญญานิพนธ์ คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุวิทย์ กิระวิทยา และ ผู้ช่วยศาสตราจารย์ ดร.พรพิศุทธิ์ วรจิรันตน์ ซึ่งเป็นคณะกรรมการในการสอบโครงงานที่ให้คำแนะนำ ชี้แนะแนวทางและข้อคิดเห็นต่างๆที่เป็นประโยชน์ในโครงงานนี้ ทำให้โครงงานนี้ออกมาสมบูรณ์แบบยิ่งขึ้น

ขอขอบพระคุณ เพื่อนคณะสหเวชศาสตร์ สาขาวิชารังสีเทคนิคที่ให้คำแนะนำเกี่ยวกับเรื่องเครื่องเอกซเรย์คอมพิวเตอร์ขนาดเล็กสำหรับนำมาใช้กับสัตว์ทดลองขนาดเล็ก ที่ช่วยให้คำแนะนำในการออกแบบโปรแกรมสำหรับควบคุมระบบที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆตลอดระยะเวลาของการศึกษาเล่าเรียน ซึ่งเป็นความรู้ที่สามารถนำไปใช้ในการทำโครงงานนี้และยังสามารถนำไปใช้ในการประกอบอาชีพในอนาคต

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณคุณพ่อ คุณแม่ ผู้มอบความรักความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอขอบคุณทุกๆคนในครอบครัวของ คณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นางสาว พรเพ็ญ บุญแสวง

สารบัญ

หน้า

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

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญ 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของงาน 2
1.4 ขั้นตอนการดำเนินงาน 2
1.5 ผลที่คาดว่าจะได้รับ 3
1.6 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎี 4
2.1 เอกซเรย์คอมพิวเตอร์ 4
2.2 ไมโครคอนโทรลเลอร์ 5
2.3 ภาษาซี 6
2.3.1 Function Heading 7
2.3.2 Variable Declaration 7
2.3.3 Compound Statements 7
2.3.4 decision statements 10
2.4 สเตปปิ้งมอเตอร์ 11
2.4.1 การกระตุ้นเฟสแบบฟลูสเตป 1 เฟส 12
2.4.2 การกระตุ้นเฟสแบบฟลูสเตป 2 เฟส 12
2.4.3 การกระตุ้นเฟสแบบฟลูสเตป 1 เฟส และ 2 เฟส 13

สารบัญ (ต่อ)

หน้า

2.5 มอเตอร์ไฟฟ้ากระแสตรง 14
2.5.1 การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง 15
2.5.2 การมอดูเลชั่นทางความกว้างของพัลส์ 15
2.5.3 การขับและกลับทิศทางของมอเตอร์กระแสตรง 16
2.6 เอนโคดเดอร์ 16
2.7 แอลซีดี 17
2.7.1 โพสซิทีฟเมทริกต์ 18
2.7.2 แอกซิทีฟเมทริกต์ 18

บทที่ 3 ขั้นตอนการดำเนินงาน 19
3.1 ทบทวนวรรณกรรม 19
3.2 ศึกษาและค้นคว้าข้อมูลของไมโครคอนโทรลเลอร์และภาษาซี 19
3.3 ศึกษาและค้นคว้าข้อมูลของสเตปปิ้งมอเตอร์ 20
3.4 ศึกษาและค้นคว้าข้อมูลของมอเตอร์ไฟฟ้ากระแสตรงและเอนโคดเดอร์ 20
3.5 ศึกษาและค้นคว้าข้อมูลของจอแอลซีดี 20
3.6 ออกแบบและพัฒนาโครงสร้างระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัด ขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน 20
3.6.1 ออกแบบส่วนโครงรับการทำงานสเตปปิ้งมอเตอร์ 20
3.6.2 ออกแบบส่วนโครงรับการทำงานมอเตอร์กระแสตรง 22
3.6.3 ออกแบบระบบการพัฒนาเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมทำการทดสอบ 23
3.7 ออกแบบระบบควบคุมการทำงานของสเตปปิ้งมอเตอร์ และมอเตอร์ไฟฟ้ากระแสตรง 27
3.7.1 ควบคุมการหมุนสเตปปิ้งมอเตอร์ และการหมุนมอเตอร์ไฟฟ้ากระแสตรงพร้อมกันทั้ง 2 ตัว 27
3.7.2 ควบคุมการหมุนสเตปปิ้งมอเตอร์ตัวเดียว 28
3.7.3 ควบคุมการหมุนกลับทิศมอเตอร์ไฟฟ้ากระแสตรง 28
3.8 ออกแบบลายปริ้นวงจรและกล่องควบคุมชุดไมโครคอนโทรลเลอร์ PIC16F886 29

สารบัญ (ต่อ)

หน้า

3.8.1 การออกแบบลายปริ้นวงจร 29

3.8.2 การออกแบบกล่องควบคุมไมโครคอนโทรเลอร์ PIC 16F886 30

3.9 ทดสอบระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน 31

3.10 วิเคราะห์ผลการทดสอบ สรุปผลการทดสอบ ปัญหาและข้อเสนอแนะ 31

3.11 จัดทำรูปเล่มปริญญานิพนธ์ 31

บทที่ 4 ผลการทดสอบและการวิเคราะห์ผลการทดสอบ 32

4.1 ผลการทดสอบการออกแบบระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน 32

4.1.1 โครงรับการทำงานของสเตปปิ้งมอเตอร์ 33

4.1.2 โครงรับการทำงานมอเตอร์กระแสตรง 33

4.1.3 โครงสร้างของอุปกรณ์การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมใช้งานได้ 34

4.2 ผลการทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่อง เอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการหมุนฐานหุ่นจำลองสัตว์ทดสองขนาดเล็กรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา 35

4.3 ผลการทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร 38

4.3.1 การทดสอบการหมุนของมอเตอร์ไฟฟ้ากระแสตรงจากการประมวลผลของไมโครคอนโทรลเลอ์ PIC16F886 38

4.3.2 การทดสอบนี้เป็นการทดสอบการหมุนของมอเตอร์ไฟฟ้ากระแสตรง ซึ่งการทดสอบนี้พ่วงกับอุปกรณ์การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน 38

4.4 สรุปผลการทดสอบ 40

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลและข้อเสนอแนะ 41
5.1 สรุปผลการดำเนินโครงงาน 41
5.2 ปัญหาและข้อเสนอแนะ 42
5.3 แนวทางการพัฒนาโครงงาน 42

เอกสารอ้างอิง 43
ภาคผนวก ก 44
ภาคผนวก ข 47
ภาคผนวก ค 66
ภาคผนวก ง 70

ประวัติผู้ดำเนินโครงงาน 102

สารบัญตาราง

ตารางที่ หน้า

2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ PIC 16F886 6

2.2 แสดงตัวดำเนินการทางคณิตศาสตร์ 7

2.3 แสดงตัวดำเนินการเปรียบเทียบ 8

2.4 แสดงตัวดำเนินการทางตรรกะ 8

2.5 แสดงตัวดำเนินการกำหนดค่า 9

2.6 แสดงลำดับการทำงานของตัวดำเนินการ 10

2.7 การกระตุ้นเฟสแบบฟลูสเตป 1 เฟส หรือ แบบเวฟ 12

2.8 การกระตุ้นเฟสแบบฟลูสเตป 2 เฟส 13

2.9 การกระตุ้นเฟสแบบฟลูสเตป 1 เฟสและ 2 เฟส 13

4.1 ตารางการหมุนองศา เริ่มต้น 0 องศา ถึง 360 องศา 35

4.2 ตารางการปรับระดับ เริ่มต้น 20 เซนติเมตร วัดจากพื้น ถึง 32 เซนติเมตร 39

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างไมโครคอนโทรลเลอร์ PIC 16F886 6
2.2 สเตปปิ้งมอเตอร์ 7
2.3 โครงสร้างภายในของสเตปปิ้งมอเตอร์ 8
2.4 มอเตอร์ไฟฟ้ากระแสตรง 8
2.5 โครงสร้างภายในมอเตอร์ไฟฟ้ากระแสตรง 9
2.6 การทำงานของเอนโคดเดอร์ 10
2.7 กราฟการทำงานของเอนโคดเดอร์ 12
2.8 กราฟสัญญาณพัลส์เอนโคดเดอร์ 13
2.9 แอลซีดี 16 ตัวอักษร 2 บรรทัด 13
3.1a ด้านข้างบนของฐาน 21
3.1b ด้านบนของฐาน 21
3.1c ด้านข้างฐานกับสเตปปิ้งมอเตอร์ 21
3.2a ด้านบนสเตปปิ้งมอเตอร์ 22
3.2b ด้านข้างสเตปปิ้งมอเตอร์ 22
3.3a ด้านบนของมอเตอร์ไฟฟ้ากรแสตรง 22
3.3b ด้านข้างของมอเตอร์ไฟฟ้ากรแสตรง 22
3.4a แสดงด้านบนของอุปกรณ์ 23
3.4b แสดงด้านข้างของอุปกรณ์ 24
3.5a ด้านข้างของตำแหน่งสเตปปิ้งมอเตอร์ 24
3.5b ด้านหน้าของตำแหน่งสเตปปิ้งมอเตอร์ 24
3.6a ด้านหลังของกล่องใส่ฟิล์ม 25
3.6b ด้านหน้าของกล่องใส่ฟิล์ม 25
3.7 แสดงด้านข้างของตำแหน่งและกลไกการเลื่อนระดับขึ้นลงของแผ่นฟิล์ม 25
3.8 ระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน 26
3.9 แผนภาพแสดงการทำงานของสเตปปิ้งมอเตอร์และมอเตอร์ไฟฟ้ากระแสตรง 27
3.10 แผนภาพแสดงการทำงานของสเตปปิ้งมอเตอร์ 28

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.11 แผนภาพแสดงการหมุนกลับทิศของมอเตอร์ไฟฟ้ากระแสตรง 28

3.12 รูปแบบลายปริ้นที่ออกแบบ 29

3.13 ออกแบบกล่องไมโครคอนโทรลเลอร์ 30

4.1 โครงรับการทำงานของสเตปปิ้งมอเตอร์ 33

4.2 โครงรับการทำงานมอเตอร์กระแสตรง 33

4.3 โครงสร้างการพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมใช้งานได้ 34

มหาวิทยาลัยนเรศวร

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เครื่องเอกซเรย์คอมพิวเตอร์ เป็นการถ่ายภาพทางรังสีโดยใช้คอมพิวเตอร์สร้างภาพในแนวตัดขวางของร่างกาย และนำภาพที่ถ่ายด้วยรังสีเอกซเรย์มาประกอบกันเพื่อนำมาประมวลผล ซึ่งมีการพัฒนาอย่างต่อเนื่อง โดยที่การพัฒนาระบบนี้มีต้นทุนเชิงพาณิชย์สูง และยังมีการใช้ในวงจำกัด ทั้งนี้ผู้จัดทำโครงงานจึงได้รับมอบหมายจากคณะสหเวชศาสตร์ สาขารังสีเทคนิค โดยมี ผู้ช่วยศาสตราจารย์ ดร.นันทวัฒน์ อู่ดี เป็นที่ปรึกษางานวิจัยเรื่องการสร้างระบบเอ็กซเรย์คอมพิวเตอร์ที่มีรายละเอียดของฟิล์มสูง ใช้ตัวรับภาพระบบฟิล์มสกรีน ให้ทำการออกแบบและพัฒนาระบบนี้เพื่อหาแนวทางลดต้นทุนการผลิต และลดการนำเข้าอุปกรณ์

การพัฒนาระบบเอกซเรย์คอมพิวเตอร์นั้นมีมาอย่างต่อเนื่องรวมทั้งยังมีการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์ขนาดเล็กสำหรับนำมาใช้กับสัตว์ทดลองขนาดเล็ก มีรายงานการศึกษาถึงความแตกต่างของผลภาพระหว่างระบบเอกซเรย์คอมพิวเตอร์ขนาดเล็กและระบบเอกซเรย์คอมพิวเตอร์ทั่วไปสำหรับถ่ายภาพสัตว์ขนาดเล็ก พบว่าระบบเอกซเรย์คอมพิวเตอร์ขนาดเล็กใช้เวลาในการประมวลผลภาพนานกว่าระบบเอกซเรย์คอมพิวเตอร์ทั่วไปเล็กน้อย แต่ทำให้ภาพที่ออกมานั้นมีรายละเอียดสูง

อย่างไรก็ตามในปัจจุบันการพัฒนาระบบเอกซเรย์คอมพิวเตอร์ที่มีรายละเอียดของฟิล์มสูงโดยใช้ตัวรับภาพระบบฟิล์ม-สกรีนมีต้นทุนเชิงพาณิชย์ของระบบเอกซเรย์คอมพิวเตอร์ขนาดเล็กที่สูงและยังมีการใช้ในวงจำกัด ดังนั้นผู้จัดทำจึงต้องการศึกษาและพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับสร้างภาพตัดขวางในสัตว์ทดลองขนาดเล็กขึ้นโดยการใช้เครื่องเอกซเรย์ทั่วไปและลดการนำเข้าเพื่อการตรวจวินิจฉัยและติดตามผลการศึกษาในสัตว์ทดลองขนาดเล็ก

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อพัฒนาระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนที่สามารถปรับการหมุนของฐานรอบแนวแกนดิ่งโดยสามารถปรับมุมการหมุนของฐานหุ่นจำลองสัตว์ขนาดเล็กจากมุม 0 องศาไปทีละ 0.9 องศาจนถึง 360 องศา

1.2.2 เพื่อพัฒนาระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนที่สามารถเลื่อนระดับแผ่นฟิล์มขึ้นลงตามแนวดิ่ง

1.2.3 อุปกรณ์นี้สามารถรองรับหุ่นจำลองสัตว์ขนาดเล็กมีขนาดเส้นผ่านศูนย์กลาง 3 เซนติเมตร และมีความสูงไม่เกิน 32 เซนติเมตร

1.3 ขอบเขตของโครงงาน

1.3.1 ตัวควบคุมการทำงานใช้ไมโครคอลโทรลเลอร์ PIC 16F886

1.3.2 การปรับองศาการหมุนฐานรอบแนวแกนดิ่งของหุ่นจำลองสัตว์ทดลองขนาดเล็กใช้สเตปปิ้งมอเตอร์กระแสตรงชนิด 103H703-0456 ขนาดกระแส 2 แอมแปร์ ขนาดแหล่งจ่ายไฟ 3.5 โวลต์ ควบคุมแบบยูนิโพล่าร์ (Unipolar) สาย 6 เส้น

1.3.3 การปรับระดับขึ้นลงของแผ่นฟิล์มสามารถปรับระดับขึ้นในแนวดิ่งใช้มอเตอร์ไฟฟ้ากระแสตรง 6000 รอบต่อนาที ZYTD-38SRZ-101023 ขนาดแหล่งจ่ายไฟ 12 ถึง 24 โวลต์

1.3.4 แสดงผลจำนวนระยะทางของมอเตอร์ไฟฟ้ากระแสตรง แต่ละระดับออกทางจอแอลซีดี ที่มีขนาด 16 ตัวอักษร 2 บรรทัด

1.4 ขั้นตอนและแผนการดำเนินงาน

กิจกรรม	ปี 2558-2559					
	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1.ทบทวนวรรณกรรม	←	→				
2.ศึกษาและค้นคว้าทฤษฎีที่เกี่ยวข้อง	←	→				
3.ออกแบบและพัฒนาโครงสร้างระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน		←	→			
4.ออกแบบระบบควบคุมการทำงานของสเตปปิ้งมอเตอร์และมอเตอร์ไฟฟ้ากระแสตรง		←	→			
5.ออกแบบลายปริ้นวงจรและกล่องควบคุมชุดไมโครคอนโทรลเลอร์ PIC 16F886			←	→		
6.ทดสอบและปรับปรุงแก้ไข					←	→
7.วิเคราะห์ผลการทดสอบ					←	→
8.สรุปผลการทดสอบ					←	→
9.จัดทำรูปเล่มปริญญานิพนธ์						← →

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 ได้รับอุปกรณ์พัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

1.5.2 ต่อยอดความรู้การสร้างภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

1.5.3 มอบโอกาสให้กับหน่วยงานเกี่ยวข้องที่ขาดอุปกรณ์การสร้างภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

1.6 งบประมาณ

1.6.1 ค่าอุปกรณ์อิเล็กทรอนิกส์	500	บาท
1.6.2 ค่าถ่ายเอกสารและค่าจัดทำรูปเล่ม	500	บาท
รวมเป็นเงิน (หนึ่งพันบาทถ้วน)	1,000	บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีเบื้องต้น

โครงงานนี้ได้มีการศึกษาหลักการและทฤษฎีของการนำไปใช้ในการประมวลผลผ่านไมโครคอนโทรลเลอร์ เพื่อใช้ในการออกแบบและพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน โดยมีส่วนประกอบของอุปกรณ์ได้แก่ สเตปปิ้งมอเตอร์ มอเตอร์ไฟฟ้ากระแสตรง เอนโคดเดอร์ และจอแสดงผลแอลซีดี โดยมีรายละเอียดของการศึกษาและค้นคว้าดังนี้

2.1 เครื่องเอกซเรย์คอมพิวเตอร์ (Computed tomography หรือ CT Scan)

โครงงานนี้มีความจำเป็นต้องใช้เครื่องเอกซเรย์คอมพิวเตอร์ต่อพ่วงกับอุปกรณ์ที่พัฒนาขึ้น ดังนั้นจึงต้องศึกษาหลักการและทฤษฎีเพื่อนำไปใช้ฉายภาพของหุ่นจำลองสัตว์ทดลองขนาดเล็ก

เอกซเรย์คอมพิวเตอร์ (Computed tomography หรือ CT Scan) เป็นการถ่ายภาพรังสีส่วนตัดโดยอาศัยคอมพิวเตอร์ในการประมวลผลภาพเครื่องเอกซเรย์คอมพิวเตอร์ซึ่งเครื่องแรกเกิดขึ้นโดยการพัฒนาทฤษฎีในการคำนวณค่าการกระจายสัมประสิทธิ์การลดลงของตัวกลางที่ดูดกลืนรังสีแกมมา ต่อมาได้ศึกษาโดยวิธีการคำนวณจากข้อมูลความเข้มรังสีที่วัดได้จากการที่รังสีทะลุผ่านร่างกายของผู้ป่วยออกมา จนสามารถสร้างเครื่องเอกซเรย์คอมพิวเตอร์ซึ่งสามารถใช้ถ่ายภาพตัดขวางของผู้ป่วยได้สำเร็จ หลังจากนั้นผู้ผลิตเครื่องเอกซเรย์คอมพิวเตอร์หลายบริษัทได้พัฒนาเครื่องเอกซเรย์คอมพิวเตอร์ให้สามารถสแกนผู้ป่วยได้ตลอดทั้งตัว และพัฒนาต่อมาเรื่อยๆจนถึงปัจจุบัน และภาพที่ตัดเป็นแผ่น (Tomography) หมายถึงภาพที่ตัดเป็นแผ่นโดยภาพที่ถูกตัดเป็นแผ่นนั้นใช้กับวิธีสร้างภาพใหม่จากโครงสร้างภายในของวัตถุที่มีความเสถียรภาพจากการวัดโครงสร้างภายนอก ในทางเอกซเรย์ภาพที่ตัดเป็นแผ่น (Tomography) หมายถึงการสร้างภาพวัตถุสามมิติจากภาพถ่ายวัตถุสองมิติ วัตถุของสนใจในการเอกซเรย์ที่รูปภาพถูกแสดงค่าโดยฟังก์ชันการประเมินค่าจริง ที่เรียกว่าสัมประสิทธิ์การลดทอน (Attenuation Coefficient) ซึ่งสัมประสิทธิ์การลดทอนหาได้จากจำนวนแนวโน้มของวัตถุที่จะดูดซับ หรือกระจายรังสีของพลังงานที่ให้ โดยปกติค่านี้ได้มาจากตำแหน่งต่างๆภายในวัตถุ และที่ถูกใช้เพื่อหาส่วนที่กระจายหายไปภายนอก ค่าสัมประสิทธิ์การลดทอนเหมือนค่าความหนาแน่นซึ่งมีประโยชน์สำหรับภาพทางการแพทย์เพราะว่าโครงสร้างทางกายวิภาควิทยามีความแตกต่างของค่าสัมประสิทธิ์การลดทอน ค่าที่มีความแตกต่างสำคัญมากต่อการวินิจฉัยระหว่างเนื้อเยื่อปกติและมะเร็ง

ส่วนของการประมวลผลผ่านไมโครคอนโทรลเลอร์นั้นเพื่อจะนำไปควบคุมการทำงานของสเตปปิ้งมอเตอร์ และมอเตอร์ไฟฟ้ากระแสตรง ให้แสดงผลออกทางจอแอลซีดี

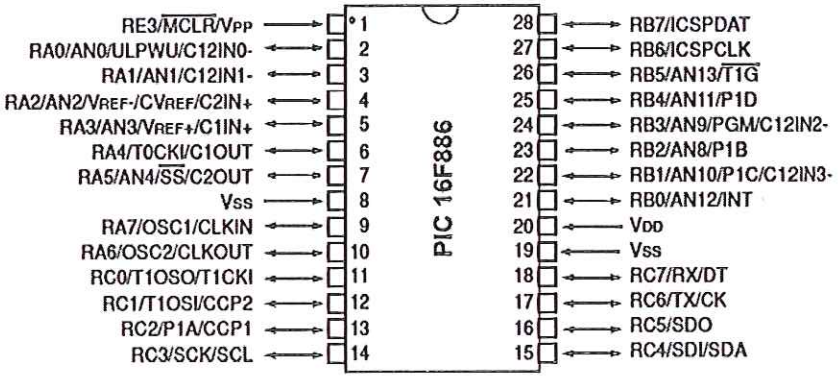
เนื่องโครงงานนี้ใช้ไมโครคอนโทรลเลอร์ PIC 16F886 จึงได้ศึกษาหลักการและทฤษฎีไปออกแบบและพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน โดยมีรายละเอียดดังนี้

2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

โครงงานนี้เลือกใช้ไมโครคอนโทรลเลอร์ PIC 16F886PIC เนื่องจากอุปกรณ์การประมวลผลตัวนี้มีคุณสมบัติสามารถควบคุมการทำงานของสเตปปิ้งมอเตอร์ และมอเตอร์ไฟฟ้ากระแสตรงให้มีการแสดงผลออกทางจอแอลซีดีได้ และราคาไม่สูง เพราะต้องการพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนขึ้นเพื่อให้มีราคาต้นทุนต่ำ จึงได้ศึกษาหลักการและทฤษฎีของไมโครคอนโทรเลอร์โดยมีรายละเอียดดังนี้

ไมโครคอนโทรลเลอร์ (Microcontroller) คืออุปกรณ์ประเภทสารกึ่งตัวนำที่รวบรวมฟังก์ชั่นการทำงานต่างๆไว้ภายในตัวของมันเองโดยมีโครงสร้างใกล้เคียงกับคอมพิวเตอร์ซึ่งภายในประกอบด้วยหน่วยรับข้อมูล และโปรแกรมหน่วยประมวลผล หน่วยความจำ หน่วยแสดงผล โดยส่วนประกอบเหล่านี้มีความสมบูรณ์ในตัวของมันเอง ทำให้มีขนาดเล็ก และสามารถเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ต่างๆที่เชื่อมกับตัวมันและง่ายต่อการนำไปใช้งาน

ไมโครคอนโทรลเลอร์ที่ใช้งานในปัจจุบันมีอยู่ด้วยกันหลายรุ่น หลายรูปแบบ หลายโครงสร้าง ในแต่ละโครงสร้างอันได้แก่หน่วยความจำภายใน จำนวนขา จำนวนพอร์ต ที่แตกต่างกัน ดังนั้นการเลือกไมโครคอนโทรลเลอร์ไปใช้งาน จึงขึ้นอยู่กับความต้องการของผู้ใช้ หรือความเหมาะสมของงาน ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานอย่างกว้างขวาง แต่โดยมากจะเป็นการนำไปใช้ฝังในระบบของอุปกรณ์อื่นๆ (Embeded Systems) เพื่อใช้ควบคุมการทำงานบางอย่าง เช่น ใช้ในระบบยานยนต์ ระบบขับเคลื่อน ระบบการทำงานอัตโนมัติ เป็นต้น เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้ในงานควบคุมหลายประการ และโครงงานนี้ได้เลือกใช้ไมโครคอนโทรเลอร์ PIC 16F886 โดยมีรายละเอียดดังในแสดงรูปที่ 2.1



รูปที่ 2.1 โครงสร้างไมโครคอนโทรลเลอร์ PIC 16F886

คุณสมบัติของไมโครคอนโทรลเลอร์ PIC 16F886 ที่เลือกมาใช้งานซึ่งได้แก่ แหล่งจ่ายแรงดันมากกว่า 2 โวลต์ ควบคุมการรีเซต ปรับความเร็วมอเตอร์ได้ ความถี่ระดับกลาง สำรองการออกแบบโปรแกรมหน่วงเวลา แปลงสัญญาณอนาล็อกเป็นดิจิตอล ซึ่งคุณสมบัติของไมโครคอนโทรลเลอร์ PIC 16F886 ได้แสดงตามตารางที่ 2.1

ตารางที่ 2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ PIC 16F886

คุณสมบัติที่ต้องการใช้	ไมโครคอนโทรเลอร์ PIC 16F886
แหล่งจ่ายแรงดันมากกว่า 2 โวลต์	2 ถึง 5.5 โวลต์
ควบคุมการรีเซ็ต	Power-on Reset
ปรับความเร็วรอบมอเตอร์ได้	Two-Speed Start-Up mode
ความถี่ระดับกลาง	Software selectable frequency range of 8 MHz to 32 kHz
โปรแกรมหน่วงเวลา	Clock mode switching during operation
แปลงสัญญาณอนาล็อกเป็นดิจิตอล	0-bit 11 channel Analog-to-Digital (A/D) Converter

ในส่วนของไมโครคอนโทรเลอร์ PIC 16F886 ซึ่งมีหน้าที่ในการประมวลผลของสเตปปิ้งมอเตอร์ให้หมุนตามรอบแกนแนวดิ่ง และมอเตอร์ไฟฟ้ากระแสตรงให้หมุนเลื่อนระดับขึ้นลงตามแนวดิ่งสามารถเขียนคำสั่งด้วยภาษาซี

2.3 ภาษาซี (C Programming Language)

ภาษาซี (C Programming Language) คือ ภาษาคอมพิวเตอร์ใช้สำหรับพัฒนาโปรแกรมทั่วไป ถูกพัฒนาครั้งแรกเพื่อใช้เป็นภาษาสำหรับพัฒนาระบบปฏิบัติการยูนิกซ์ (Unix Opearating System) แทนภาษาแอสเซมบลี ซึ่งเป็นภาษาระดับต่ำที่สามารถกระทำในระบบฮาร์ดแวร์ได้ด้วยความรวดเร็วแต่จุดอ่อนของภาษาแอซเซมบลีก็คือความยุ่งยากในการโปรแกรมความเป็นเฉพาะตัว และความแตกต่างกันไปในแต่ละเครื่อง เดนนิส ริตชี (Dennis Ritchie) จึงได้คิดค้นพัฒนาภาษาใหม่นี้ขึ้นมาโดยการรวบรวมเอาจุดเด่นของแต่ละภาษาระดับสูงผนวกเข้ากับภาษาระดับต่ำ เรียกชื่อว่าภาษาซี เมื่อได้รับความนิยมมากขึ้น จึงมีผู้ผลิตได้รวบรวมภาษาซีออกมาแข่งขันกัน ทำให้เริ่มมีการใส่ลูกเล่นต่างๆเพื่อดึงดูดใจผู้ซื้อทาง American National Standard Institute (ANSI) จึงกำหนดมาตรฐานของภาษาซีขึ้นเรียกว่า ANSI C เพื่อคงมาตรฐานของภาษาไว้ไม่ให้เปลี่ยนแปลงไป

โครงสร้างของโปรแกรมภาษาซี

โปรแกรมในภาษาซีทุกโปรแกรมจะประกอบด้วยฟังก์ชันอย่างน้อยหนึ่งฟังก์ชันคือฟังก์ชันหลักโดยโปรแกรมภาษาซีจะเริ่มทำงานที่ฟังก์ชันหลักก่อนในแต่ละฟังก์ชันจะประกอบด้วย

2.3.1 Function Heading ประกอบด้วยชื่อฟังก์ชัน และอาจมีรายการของ argument (บางคนเรียก parameter) อยู่ในวงเล็บ

2.3.2 Variable Declaration ส่วนประกาศตัวแปร สำหรับภาษาซี ตัวแปรหรือค่าคงที่ทุกตัว ที่ใช้ในโปรแกรมจะต้องมีการประกาศก่อนว่าจะใช้งานอย่างไร จะเก็บค่าในรูปแบบใดเช่น interger หรือ real number

2.2.3 Compound Statements ส่วนของประโยคคำสั่งต่างๆ ซึ่งแบ่งเป็นประโยคเชิงซ้อน(compound statement) กับ ประโยคนิพจน์ (expression statment) โดยประโยคเชิงซ้อนจะอยู่ภายในวงเล็บปีกกาคู่หนึ่ง { และ } โดยในหนึ่งประโยคเชิงซ้อน จะมีประโยคนิพจน์ที่แยกจากกันด้วยเครื่องหมาย semicolon (;) หลายๆ ประโยครวมกัน และ อาจมีวงเล็บปีกกาใส่ประโยคเชิงซ้อนย่อยเข้าไปอีกได้

2.3.3.1 ตัวดำเนินการ (Operator)

การกำหนดตัวดำเนินการในภาษาซีนั้นมีหลายประเภทด้วยกัน ซึ่งแต่ละประเภทจะทำหน้าที่แตกต่างกันออกไป ดังต่อไปนี้

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

ตัวดำเนินการทางคณิตศาสตร์ใช้เป็นตัวเชื่อมในการเขียนโปรแกรม เพื่อหาผลลัพธ์จากการคำนวณซึ่งสามารถกระทำกับข้อมูลได้หลายรูปแบบ แสดงในตารางที่ 2.2

ตารางที่ 2.2 แสดงตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
+	บวก (Addition)	A + b
-	ลบ (Subtraction)	A – b
*	คูณ (Multiplication)	A * b
/	หาร (Division)	A / b
%	หารเอาผลลัพธ์เฉพาะเศษ (Modulus)	A % b
++	เพิ่มค่าขึ้น 1 ค่า (Increment)	A++
--	ลดค่าลง 1 ค่า (Decrement)	A- -

- ตัวดำเนินการเปรียบเทียบ (Comparative Operator)

ตัวดำเนินการเปรียบเทียบ คือ เครื่องหมายที่ใช้ในการเปรียบเทียบหรือตรวจสอบในทางคณิตศาสตร์ โดยจะแสดงตารางที่ 2.3 ซึ่งผลลัพธ์ที่จะได้จะมี 2 กรณีคือ ถ้าถูกต้องหรือเป็นจริงจะมีค่าเป็น 1 ถ้าผิดหรือเป็นเท็จจะมีค่าเป็น 0 ผลลัพธ์ของการเปรียบเทียบ มีชื่อเรียกอีกอย่างหนึ่งว่า ค่าคงที่บูลีน (Boolean Constant) ดังนี้

8 > 4 ผลลัพธ์เป็นจริง ค่าคงที่บุลีนเป็น 1

6 < = 2 ผลลัพธ์เป็นเท็จ ค่าคงที่บูลีนเป็น 0

-2 > -6 ผลลัพธ์เป็นจริง ค่าคงที่บูลีนเป็น 1

A > a ผลลัพธ์เป็นจริง ค่าคงที่บูลีนเป็น 1 (เปรียบเทียบค่าตามรหัสแอสกี)

ตารางที่ 2.3 แสดงตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
>	มากกว่า (Greater Than)	a > b
<	น้อยกว่า (Less Than)	a < b
> =	มากกว่าหรือเท่ากับ (Greater Than or Equal)	a > = b
< =	น้อยกว่าหรือเท่ากับ (Less Than or Equal)	a < = b
= =	เท่ากับ (Equal)	a = = b
! =	ไม่เท่ากับ (Not Equal)	a ! = b

- ตัวดำเนินการทางตรรกะ (Logical Operator)

ตัวดำเนินการทางตรรกะคือ เครื่องหมายที่ใช้เชื่อมเงื่อนไข 2 เงื่อนไขหรือมากกว่า เพื่อให้การเปรียบเทียบมีความละเอียดมากขึ้น แสดงตารางที่ 2.4

ตารางที่ 2.4 แสดงตัวดำเนินการทางตรรกะ

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
& &	และ (and)	Income>=5000&&income<=10000
||	หรือ (or)	Hour<O||hour>24
!	ไม่ใช่ (not)	! a &&! b

ตัวดำเนินการ && จะให้ผลลัพธ์เป็นจริงเมื่อเงื่อนไขทั้งสองเป็นจริง แต่ถ้าเงื่อนไขใดเป็นเท็จหรือทั้งสองเงื่อนไขเป็นเท็จจะทำให้ผลลัพธ์เป็นเท็จ

ตัวดำเนินการ || จะใช้ผลลัพธ์เป็นจริงเมื่อเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง หรือเป็นจริงทั้งสองเงื่อนไข แต่ถ้าเป็นเท็จทั้งสองเงื่อนไขจะทำให้ผลลัพธ์เป็นเท็จ

ตัวดำเนินการ ! จะให้ผลลัพธ์เป็นจริงเมื่อไขสันหลัง not เป็นเท็จ แต่ถ้าเงื่อนไขหลัง not เป็นจริงจะทำให้ผลลัพธ์เป็นเท็จ

- ตัวดำเนินการกำหนดค่า (Assignment Operator)

ตัวเนินการกำหนดค่าใช้ในการกำหนดค่าให้กับตัวแปรที่มีการคำนวณและนำค่าที่ได้ไปเก็บไว้ยังตัวแปรอื่น แสดงตารางที่ 2.5

ตารางที่ 2.5 แสดงตัวดำเนินการกำหนดค่า

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
=	การกำหนดค่าให้เท่ากับ (Assignment)	a = b
+ =	การเพิ่มค่า (Addition)	A += b มาจาก a = a + b
- =	การลบค่า (Subtraction)	A -= b มาจาก a = a – b
* =	การคูณ (Multiply)	A *= b มาจาก a = a * b
/ =	การหารได้ผลลัพธ์จำนวนเต็ม (Devide)	A /= b มาจาก a = a / b
% =	การหารได้ผลลัพธ์เศษ (Devide)	A %= b มาจาก a = a % b
& =	ดำเนินการ bitwise and	A &= b มาจาก a = a & b
l =	ดำเนินการ inclusive or	A l= b มาจาก a = a l b
^ =	ดำเนินการ exclusive or	A ^= b มาจาก a = a ^ b
<< =	การเลื่อนบิตไปทางซ้าย (Left Shift)	A <<= b มาจาก a = a << b
>> =	การเลื่อนบิตไปทาขวา (Right Shift)	A >>= b มาจาก a = a >> b

2.3.3.2 นิพจน์ (Expression) คือการนำเอาค่าคงที่ตัวแปรหรือฟังก์ชันมากระทำอย่างใดอย่างหนึ่งโดยมีการใช้ตัวดำเนินการ (Operator) มากำหนดค่า เปรียบเทียบและบ่งชี้การกระทำนั้น ลำดับการประมวลผลนิพจน์ การกำหนดนิพจน์และใช้โอเปอเรเตอร์หรือตัวดำเนินการหลายๆตัวพร้อมๆกันภายในนิพจน์เดียวกัน ถ้าผู้เขียนไม่จัดลำดับการทำงานก่อนหลังของโอเปอเรเตอร์ให้ถูกต้องก่อนจะทำให้ได้ผลลัพธ์ที่คาดเคลื่อนจากความเป็นจริง โดยการประเมินค่าตามลำดับของโอเปอเรเตอร์ของค่าคงที่จำนวนมากกว่าสองตัวสามารถทำได้โดยใช้กฎ Precedence ซึ่งข้อกำหนดสำหรับลำดับการประมวลผลนิพจน์เป็นดังแสดงในตารางที่ 2.6

ตารางที่ 2.6 แสดงลำดับการทำงานของตัวดำเนินการ

ลำดับที่	ตัวดำเนินการ	ทิศทางการประมวลผล
1	( )	ซ้ายไปขวา
2	++, - -	ขวาไปซ้าย
3	*, / , %	ซ้ายไปขวา
4	+, -	ซ้ายไปขวา
5	=	ขวาไปซ้าย

2.3.4 คำสั่งเงื่อนไขเพื่อการตัดสินใจ (decision statements) เป็นคำสั่งที่มีการทดสอบเงื่อนไขก่อนที่จะทำงานตามคำสั่งที่กำหนดไว้ ซึ่งได้แก่คำสั่ง if, if else และโครงสร้าง else if

2.3.4.1 คำสั่ง if เป็นคำสั่งที่ให้มีการทดสอบเงื่อนไขก่อนที่จะนำไปทำงานตามคำสั่งที่กำหนดไว้

2.3.4.2 คำสั่ง if else เป็นคำสั่งที่ต่างจากคำสั่ง if แบบง่ายที่ผ่านมาตรงที่คำสั่ง if แบบง่ายนั้น มีทางเลือกที่จะให้ทำอยู่เพียงทางเลือกเดียวคือ ถ้าเป็นจริงก็ทำทางเลือกนั้นแต่ถ้าเป็นเท็จก็ไม่ทำ แต่คำสั่ง if - else นี้มีทางเลือกที่ให้ทำอยู่สองกรณี คือ

กรณีที่ 1 ถ้าเงื่อนไขเป็นจริงให้ทำกับสายงานหนึ่ง คือ ทำตามชุดคำสั่งที่ตามหลังเงื่อนไข (Condition) ที่ใช้ตรวจสอบ

กรณีที่ 2 ถ้าเงื่อนไขเป็นเท็จก็ให้ทำกับอีกสายงานหนึ่ง คือ ทำตามชุดคำสั่งส่วนของ else

รูปแบบของคำสั่งจะเห็นว่าจะมีลักษณะการเขียนคล้ายกับคำสั่ง if แต่จะมีการเพิ่มเติมคำว่า else เข้าไปด้วย ซึ่งการทำงานของคำสั่งแบบนี้นั้นจะทำการตรวจสอบเงื่อนไขที่ if ก่อนว่าเป็นจริงหรือเท็จ โดยถ้าเป็นจริงจะไปทำงานในกลุ่มคำสั่งที่เรากำหนดไว้ใน if แต่ถ้าเป็นเท็จก็จะไปทำงานในกลุ่มคำสั่งที่อยู่ในส่วนของ else

ส่วนการทำงานของระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนซึ่งจะประกอบด้วยการทำงานของสเตปปิ้งมอเตอร์ให้สามารถหมุนฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กตามรอบแกนแนวดิ่ง มอเตอร์ไฟฟ้ากระแสตรงให้สามารถปรับระดับแผ่นฟิล์มขึ้นลงตามแนวดิ่งได้โดยการประมวลผลของไมโครคอนโทรลเลอร์ PIC 16F886 และสามารถแสดงผลของระยะทางที่มอเตอร์ดำเนินการออกทางจอแอลซีดี ซึ่งระบบการทำงานนี้จึงได้ทำการศึกษาและค้นคว้าหลักการและทฤษฎีเบื้องต้นของสเตปปิ้งมอเตอร์ มอเตอร์ไฟฟ้ากระแสตรง และจอแอลซีดี โดยมีรายละเอียดดังนี้

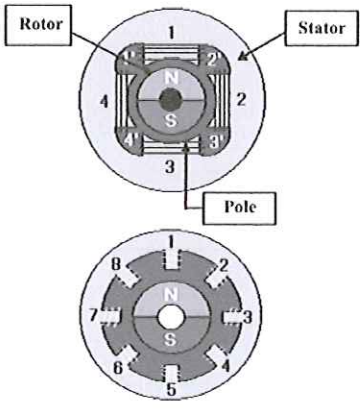
2.4 สเตปปิ้งมอเตอร์ (Stepping Motor)

โครงงานนี้นำสเตปปิ้งมอเตอร์มาทำหน้าที่ในปรับการหมุนของฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กตามรอบแกนแนวดิ่ง โดยลักษณะของสเตปปิ้งมอเตอร์ที่นำมาใช้งานแสดงดังรูปที่ 2.2 ดังนั้นจึงได้ทำการศึกษาและค้นคว้าหลักการและทฤษฎีของสเตปปิ้งมอเตอร์ โดยมีรายละเอียดดังนี้

รูปที่ 2.2 สเตปปิ้งมอเตอร์

สเตปปิ้งมอเตอร์ (Stepping Motor) เป็นมอเตอร์ที่มีลักษณะเมื่อป้อนไฟฟ้าให้กับมอเตอร์ทำให้หมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด ซึ่งสามารถกำหนดตำแหน่งของการหมุนด้วยตัวเลขของศาหรือระยะทางได้อย่างละเอียดโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวกำหนดและจัดเก็บตัวเลขโดยสเตปปิ้งมอเตอร์

โครงสร้างภายในมอเตอร์จะมีขั้วแม่เหล็กบนสเตเตอร์ (Stator) ทำมาจากแผ่นเหล็กวงแหวนที่มีขี่ยื่นออกมาประกอบกันเป็นชั้นๆ โดยที่แต่ละขี่นั้นจะมีขดลวด (Coil) พันสวมอยู่เมื่อมีการป้อนกระแสผ่านขดลวดทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic) ในที่นี้ถ้าเพิ่มจำนวนรอบของขั้วแม่เหล็กมากขึ้นจะเพิ่มจำนวนสเตปต่อรอบมากขึ้นดังแสดงในรูปที่ 2.3 ซึ่งโครงสร้างภายในและจำนวนรอบที่มากขึ้นของขั้วแม่เหล็ก



รูปที่ 2.3 โครงสร้างภายในของสเตปปิ้งมอเตอร์

การนำไปใช้งานใช้ระบบเปิด (Open Loop system) สามารถทำงานได้โดยไม่ต้องมีการป้อนค่าพารามิเตอร์ย้อนกลับแต่ทุกวิธีที่ต้องการกำหนดตำแหน่งที่แน่นอนจะต้องการป้อนกลับไปยังระบบและตัวบอกตำแหน่งว่าถูกต้องหรือผิดพลาดได้ ตัววงจรไมโครคอนโทรลเลอร์เองจะมีจุดอ้างอิง ไว้ให้เริ่มต้นการทำงานและอ้างอิงตำแหน่งได้ถูกต้อง โดยมีแนวทางของสเต็ปปิ้งมอเตอร์เป็นอุปกรณ์จำพวกเชิงกลทางไฟฟ้ามีกลุ่มของไบนารี่ (Binary) แรงดัน (Voltage) เป็นอินพุต (Input) และการเคลื่อนที่แบบเชิงมุมเป็นเอาท์พุต (Output) หรือ ว่าหมุนไปทีละสเตปซึ่งอยู่ระหว่าง 0.1 - 30 องศา ขึ้นอยู่ที่โครงสร้างของสเตปปิ้งมอเตอร์ตามสัญญาณพัลส์ที่จ่ายให้กับขดสเตเตอร์ทำให้เกิดแรงผลักแก่โรเตอร์ (Rotor) หมุนไป และมีขดลวดหลายชุดในที่นี้เรียกว่า เฟส (Phase) ดังนั้นสัญญาณที่ต่อเนื่องเป็นลำดับ ลักษณะของไบนารี่จะต้องไปผ่านวงจรขับมอเตอร์ (Driver) ก็จะทำให้โรเตอร์หมุนไป อย่างต่อเนื่อง การสั่งงานควบคุมการหมุนของสเต็ปมอเตอร์ทำงานเพื่อที่จะทำการหมุนมีวิธีการควบคุมกระแสไฟที่จ่ายให้กับขดลวดสเตเตอร์ในแต่ละเฟสของสเตปปิ้งมอเตอร์อย่างเป็นลำดับที่แน่นอน ถ้าต้องการให้กระแสไหลในเฟสใดๆก็จะทำให้สถานะของเฟสนั้นๆเป็นสถานะลอจิก "1" และในการกระตุ้นเฟส แบ่งการสั่งงานควบคุมการหมุนของสเตปปิ้งมอเตอร์ได้เป็น 3 รูปแบบ มีรายละเอียดดังนี้

2.4.1 การกระตุ้นเฟสแบบฟลูสเตป 1 เฟส (Single-Phase Driver) หรือ แบบเวฟ (Wave) ซึ่งแสดงดังตารางที่ 2.7 จะเป็นการป้อนกระแสไฟให้กับขดลวดของสเตปปิ้งมอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไปดังนั้นกระแส ที่ไหลในขดลวด จะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเต็ปปิ้งมอเตอร์มีน้อย

ตารางที่ 2.7 การกระตุ้นเฟสแบบฟลูสเตป 1 เฟส หรือ แบบเวฟ

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON

2.4.2 การกระตุ้นเฟสแบบฟลูสเตป 2 เฟส (Two-Phase Driver) ซึ่งแสดงดังตาราง 2.8 เป็นการป้อนกระแสให้กับขดลวด 2 ขด ของสเตปปิ้งมอเตอร์พร้อมๆกันไป และจะกระตุ้นเรียงถัดกับไปเช่นเดียวกับแบบหนึ่งเฟส ดังนั้นการกระตุ้นแบบนี้จึงต้องใช้กำลังไฟมากขึ้น และจะทำให้มีแรงบิดของมอเตอร์มากกว่าการกระตุ้นแบบ 1 เฟส

ตารางที่ 2.8 การกระตุ้นเฟสแบบฟลูสเตป 2 เฟส (Two-Phase Driver)

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 3
1	ON	ON	OFF	OFF
2	OFF	ON	ON	OFF
3	OFF	OFF	ON	ON
4	ON	OFF	OFF	ON

2.4.3 การกระต้นเฟสแบบฟลสเตป 1 เฟส และ 2 เฟส (One-Two Phase Driver) หรือ แบบฮาลฟ์สเตป (Half Step Motor)

การกระตุ้นเฟสแบบฟลูสเตป 1 เฟสและ 2 เฟส (One-Two Phase Driver) หรือ แบบฮาลฟ์สเตป (Half Step Motor) เรียงลำดับกันไปแสดงดังตารางที่ 2.9 แรงบิดที่ได้จากการกระตุ้นเฟสแบบนี้จะมีเพิ่มมากขึ้นเพราะช่วงของสเตปมีระยะสั้นลงในการกระตุ้นแบบนี้จะต้องมีการกระตุ้นที่เฟสถึง 2 ครั้งจึงจะได้ระยะของสเตปเท่ากับการกระตุ้นเพียงครั้งเดียวของแบบฟูลสเตป 2 แบบแรก ความละเอียดของการหมุนตำแหน่งองศาต่อสเตป ก็เป็นสองเท่าของแบบแรก ความถูกต้องของตำแหน่งที่กำหนดจึงมีมากขึ้น

ตารางที่ 2.9 การกระตุ้นเฟสแบบฟลูสเตป 1 เฟสและ 2 เฟส (One-Two Phase Driver)

สเตป	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON

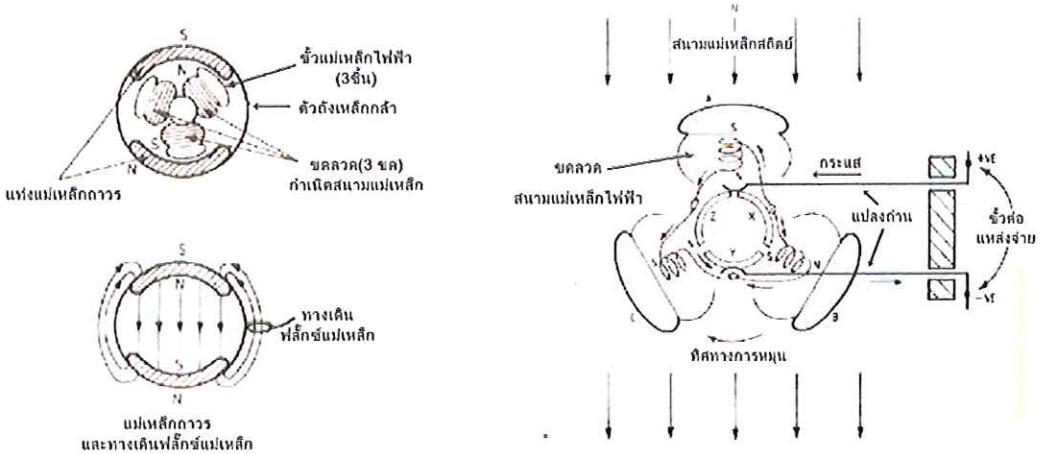
2.5 มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor หรือ DC Motor)

โครงงานนี้ได้นำมอเตอร์ไฟฟ้ากระแสตรงมาใช้งานการปรับระดับขึ้นลงของแผ่นฟิล์มตามแนวดิ่ง ดังนั้นจึงได้ทำการศึกษาและค้นคว้าหลักการและทฤษฎีเบื้องต้น โดยมอเตอร์ไฟฟ้ากระแสตรงมีลักษณะดังแสดงในรูปที่ 2.4 มีรายละเอียดดังนี้

รูปที่ 2.4 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor หรือ DC Motor) เป็นเครื่องกลไฟฟ้าชนิดหนึ่งที่เปลี่ยนแปลงพลังงานไฟฟ้ามาเป็นพลังงานกลซึ่งเป็นอุปกรณ์ที่นิยมใช้กันอย่างแพร่หลายในโรงงานต่างๆที่ใช้ควบคุมเครื่องจักรกลในโรงงานอุตสาหกรรมต่างๆและมีหลายชนิดให้เลือกใช้ให้เหมาะสมกับประเภทของงาน

หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) หลักการทำงานโดยวิธีการผ่านกระแสให้กับขดลวดในสนามแม่เหล็กซึ่งทำให้เกิดแรงแม่เหล็ก ส่วนแรกนี้จะขึ้นอยู่กับกระแสและกำลังของสนามแม่เหล็ก โครงสร้างภายในทั่วไปของมอเตอร์ไฟฟ้ากระแสตรงดังแสดงในรูปที่ 2.5



รูปที่ 2.5 โครงสร้างภายในมอเตอร์ไฟฟ้ากระแสตรง

เมื่อเป็นแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะแปรงถ่านผ่านคอมมิว-เตเตอร์เข้าไปในขดลวดอาร์เมเจอร์สร้าง สนามแม่เหล็กขึ้นและกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field Coil) สร้างขั้วเหนือ-ใต้ขึ้นจะเกิดสนามแม่เหล็ก 2 สนามในขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดกัน ทิศทางตรงข้ามจะหักล้างกันและทิศทางเดียวจะเสริมแรงกันทำให้เกิดแรงบิดในตัวอาร์เมเจอร์ซึ่งวางแกนเพลาและแกนเพลานี้สวมอยู่กับตลับลูกปืนของมอเตอร์ทำให้อาร์เมเจอร์นี้หมุนได้ขณะที่ตัวอาร์เมเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร (Rotor) ซึ่งหมายความว่า ตัวหมุนการที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกิริยาต่อกันทำให้ขดลวดอาเมเจอร์หรือโรเตอรหมุนไปนั้นเป็นไปตามกฎมือซ้ายของเฟลมมิ่ง (Fleming left hand rule)

2.5.1 การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง

การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไปเช่นการควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นเราจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชั่นทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

2.5.2 วิธีการมอดูเลชั่นทางความกว้างของพัลส์ (Pulse Width Modulation)

การมอดูเลชั่นทางความกว้างของพัลส์ (Pulse Width Modulation) เป็นการปรับเปลี่ยนที่สัดส่วนและความกว้างของสัญญาณพัลส์โดยความถี่ของสัญญาณพัลส์จะไม่มีการเปลี่ยนแปลงหรือเป็นการเปลี่ยนแปลงค่าของช่วงความกว้างของพัลส์ที่สถานะลอจิกสูง (Duty Cycle) โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด เช่น ถ้าค่าดิวตี้ไซเคิลมีค่าเท่ากับ 50% ก็หมายถึงใน 1 รูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่งและสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง และในทำนองเดียวกันถ้าค่าดิวตี้ไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น ถ้าค่าดิวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่าจะไม่มีสถานะลอจิกต่ำ หาได้จากค่าความสัมพันธ์ สัดส่วนระหว่างช่วงของสัญญาณพัลส์กับคาบเวลาทั้งหมดของสัญญาณคิดเป็น 100 % ของค่าดิวตี้ไซเคิล ซึ่งแสดงสมการการหาค่าดิวตี้ไซเคิล (สมการที่ 1) ได้ดังนี้

$$\text{ค่าดิวตี้ไซเคิล} = \frac{\text{ช่วงของสัญญาณพัลส์}}{\text{คาบเวลาทั้งหมดของสัญญาณ}} \times 100\% \qquad \text{(สมการที่ 1)}$$

2.5.3 การขับและกลับทิศทางของมอเตอร์กระแสตรง

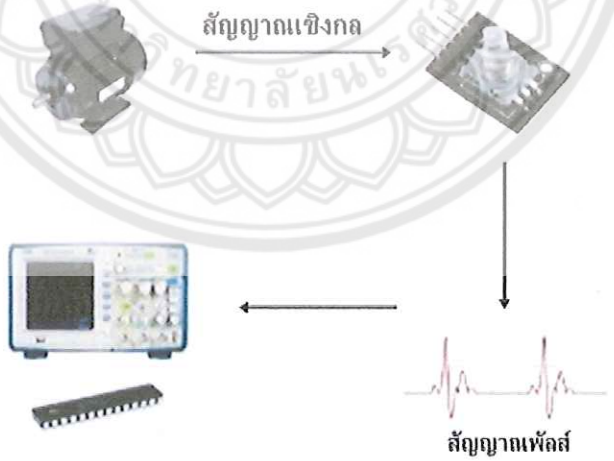
การขับและกลับทิศทางของมอเตอร์กระแสตรงโดยการใช้ไมโครคอนโทรเลอร์เป็นตัวควบคุมการหมุนของมอเตอร์และทิศทางของมอเตอร์จะต้องมีส่วนของวงจรที่เรียกว่าวงจรขับมอเตอร์ (Driver) ในส่วนของวงจรกลับทิศทางสามารถใช้รีเลย์ต่อวงจรสวิตซ์เพื่อกลับทิศทางของขั้วไฟกระแสตรง หรืออาจใช้อุปกรณ์สารกึ่งตัวนำที่เป็นวงจรขับกำลังเช่น ทรานซิสเตอร์ มอสเฟต ขึ้นอยู่ที่การใช้งานของอุปกรณ์นั้นๆ

ในส่วนของมอเตอร์ไฟฟ้ากระแสตรงซึ่งทำหน้าที่เลื่อนระดับขึ้นลงของแผ่นฟิล์มตามแนวดิ่งโดยมีเอนโคดเดอร์ทำการตรวจจับระยะทางในการขึ้นลงตามแนวดิ่งของแผ่นฟิล์ม

2.6 เอนโคดเดอร์ (Encoder)

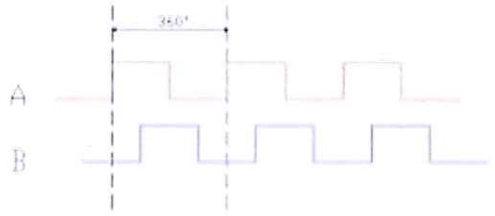
เอนโคดเดอร์เป็นเซนเซอร์ตรวจสอบการหมุนมีหน้าที่เกี่ยวข้องกับการหมุนทั้งหมดดังนี้ ตรวจวัดความเร็วรอบ ทิศทางการหมุนของมอเตอร์ มุมของการหมุน จำนวนรอบของการหมุน โดยสามารถตรวจสอบได้ 2 แบบ คือ เชิงเส้น (Linear) และเชิงมุม (Angular) โดยทั่วไปการนำมาใช้งานนิยมแบบเชิงมุม

หลักการทำงานของเอนโคดเดอร์ คือจะรับสัญญาณเชิงกลจากการหมุนของมอเตอร์ไฟฟ้ากระแสตรง มาแปลให้อยู่ในรูปแบบของคำสั่ง ซึ่งจะสั่งสัญญาณพัลส์และประมวลผลผ่านไมโครคอนโทรลเลอร์นำไปใช้ตรวจสอบระยะทางมีลักษณะการทำงานตามที่แสดงในรูปที่ 2.6



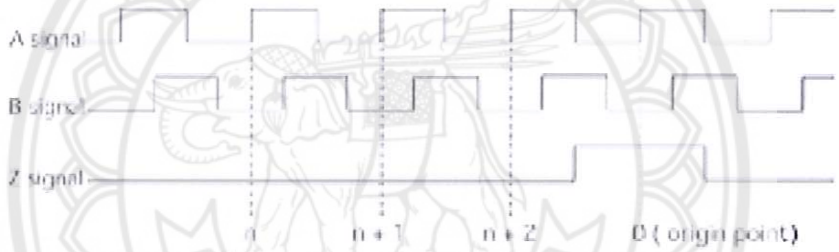
รูปที่ 2.6 การทำงานของเอนโคดเดอร์

โดยมีจากหลักการทำงานของเอนโคดเดอร์คือจะวัดการนำ และตามของเฟสคลื่นในช่วง 360 องศาดังที่แสดงในรูปกราฟที่ 2.7



รูปที่ 2.7 กราฟการทำงานของเอนโคดเดอร์

เอนโคดเดอร์เป็นเซนเซอร์สำหรับการวัดจำนวนรอบหรือระยะทางการหมุน และความเร็วของมอเตอร์ สัญญาณพัลส์เอนโคดเดอร์ ประกอบด้วย สัญญาณ A B และ Z ซึ่งสัญญาณ A กับ B จะต่างเฟสกัน 90 องศา จะขึ้นอยู่กับทิศทางการหมุนของมอเตอร์ดังแสดงตามรูปที่ 2.7 เช่น ถ้าหมุนตามเข็มนาฬิกาเฟสA จะนำหน้าเฟสB 90 องศา ทิศทางกลับกันเฟสA ล้าหลังเฟสB 90 องศา ส่วนสัญญาณ Z จะให้สัญญาณออก 1 เฟสทุกๆ 1 รอบซึ่งทำหน้าที่เป็นสวิทซ์ดังแสดงตามรูปกราฟที่ 2.8

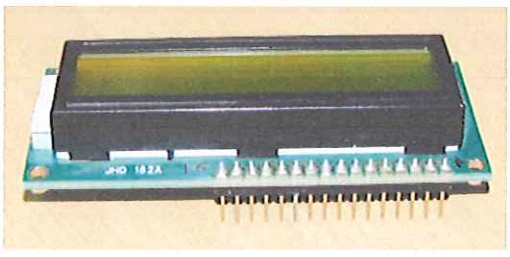


รูปที่ 2.8 กราฟสัญญาณพัลส์เอนโคดเดอร์

เมื่อเอนโคดเดอร์ทำการตรวจสอบระยะทางการขึ้นลงในทุกระยะการดำเนินการของมอเตอร์ไฟฟ้ากระแสตรง โดยที่มีการแสดงผลออกทางจอแอลซีดี

2.7 แอลซีดี (LCD : Liquid Crystal Display) หรือ โฟโต้เซลล์ (Photo Cell)

โครงงานนี้ได้นำจอแอลซีดีมาทำหน้าที่แสดงผลของจำนวนระยะทางที่มอเตอร์ไฟฟ้ากระแสตรงดำเนินการได้ ดังนั้นจึงได้ทำการศึกษาและค้นคว้าหลักการและทฤษฎีของจอแอลซีดีซึ่งมีลักษณะดังแสดงในรูปที่ 2.9 โดยมีรายละเอียดดังนี้



รูปที่ 2.9 แอลซีดี 16 ตัวอักษร 2 บรรทัด

แอลซีดี (LCD) หรือ โฟโต้เซลล์ (Photo Cell) คือเทคโนโลยีมอนิเตอร์แอลซีดีย่อมาจาก หน้าจอแสดงผลผลึกเหลว (Liquid Crystal Display) ซึ่งเป็นจอแสดงผลแบบดิจิตอล (Digital) โดย ภาพที่ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Black Light) ผ่านชั้นกรองแสง (Polarized filter) แล้ววิ่งไปยังคริสตัลเหลวที่เรียงตัวด้วยกัน 3 เซลล์คือ แสงสีแดง แสงสีเขียวและแสงสีน้ำเงินกลายเป็นพิกเซล(Pixel)ที่สว่างสดใสเกิดขึ้น นอกจากนี้เทคโนโลยีที่ พัฒนามาใช้กับ แอลซีดี สามารถแบ่งออกเป็น 2ประเภทคือ

2.7.1 โพสซิธีฟเมทริกต์ (Super-Twisted Nematic) เป็นเทคโนโลยีแบบเก่าที่ให้ความ คมชัดและความสว่างน้อยกว่า ใช้ในจอโทรศัพท์มือถือทั่วไป

2.7.2 แอกซิธีฟเมทริกต์ (Thin Film Transistors) สามารถแสดงภาพได้คมชัดและสว่าง กว่าแบบแรก ใช้ในจอมอนิเตอร์หรือโน้ตบุ๊ก สทิวเท็ดนีมาติก (Twisted Nematic) คือสารประเภทนี้ จะมีการจัดโครงสร้างโมเลกุลเป็นเกลียว แต่ถ้าผ่านกระแสไฟฟ้าเข้าไปมันก็จะคลายตัวออกเป็น เส้นตรง เราใช้ปรากฏการณ์นี้เป็นตัวกำหนดว่าจะให้แสงผ่านได้หรือไม่ได้สทิวเท็ดนีมาติกผลึก เหลวชนิดนี้จะให้เราสามารถเปลี่ยนทิศทางการสั่นของคลื่นแสงได้ 90 องศาเซลเซียส ถึง 150 องศา เซลเซียส คือเปลี่ยนจากแนวตั้งให้กลายเป็นแนวนอนหรือเปลี่ยนกลับกันจากแนวนอนให้เป็น แนวตั้งก็ได้ ด้วยจุดนี้เองทำให้การค่าตอบสนองสัญญาณเทียบกับเวลา (Response Time) มีค่าสูง

หลักการคือการบังคับให้หยดของผลึกเหลว (Liquid crystal) ซึ่งมีแผ่นแก้วกักเอาไว้ให้ไป ปิดรูช่องแสง ซึ่งแสงถูกฉายมาจากด้านหลังของหน้าจอก่อให้เกิดการแสดงผลเป็นตัวอักษรหรือ ตัวเลขในรูปแบบต่างๆได้ตามต้องการ จุดเด่นของหน้าจอแอลซีดีขาว-ดำหรือเรียกว่าแบบโมโน โกลด์ โดยการทำงานที่ไม่อาศัยปืนยิงอิเล็กตรอน จึงช่วยให้ด้านลึกของจอภาพมีขนาดสั้นกว่า มอนิเตอร์แบบซีดีที (CDT) ถึง 3 เท่า และด้วยรูปร่างที่แบนราบทางด้านหน้าและด้านหลัง เมื่อมี แสงส่องถูกขั้วลบจะเกิดอิเล็กตรอน อิเล็กตรอนจะวิ่งจากขั้วลบไปยังขั้วบวกทำให้เกิดกระแสไฟฟ้า ขึ้นในวงจร กระแสไฟฟ้านี้จะมีมากหรือน้อยขึ้นอยู่กับปริมาณของแสงที่โฟโต้อิเล็กทริกเซลล์อยู่ และเมื่อมีแสงแล้วการทำงานของเครื่องก็จะทำงานปกติ แต่เมื่อเครื่องทำงานแล้วไม่มีแสงเกิดขึ้น

บทที่3
ขั้นตอนการดำเนินงาน

โครงงานนี้ในส่วนของบทที่ 3 ผู้จัดทำจะกล่าวถึงขั้นตอนการดำเนินงานแต่ละขั้นตอนของการออกแบบและพัฒนาระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน โดยเริ่มต้นจากการศึกษาและค้นคว้าข้อมูลเครื่องเอกซเรย์คอมพิวเตอร์ ศึกษาและค้นคว้าข้อมูลของไมโครคอนโทรลเลอร์และภาษาซี ศึกษาและค้นคว้าข้อมูลของสเตปปิ้งมอเตอร์ ศึกษาและค้นคว้าข้อมูลของมอเตอร์ไฟฟ้ากระแสตรงและเอนโคดเดอร์ ศึกษาและค้นคว้าข้อมูลของจอแอลซีดี ออกแบบและพัฒนาโครงสร้างระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน ออกแบบระบบควบคุมการทำงานของสเตปปิ้งมอเตอร์และมอเตอร์ไฟฟ้ากระแสตรง ออกแบบลายปริ้นวงจรและกล่องควบคุมชุดไมโครคอนโทรลเลอร์ PIC 16F886 ทดสอบระบบการทำงานของการพัฒนาระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน วิเคราะห์ผลการทดสอบ สรุปผลการทดสอบ ปัญหาและข้อเสนอแนะ และจัดทำรูปเล่มปริญญานิพนธ์ โดยที่มีรายละเอียดของขั้นตอนดังนี้

3.1 ทบทวนวรรณกรรม

ระบบเอกซเรย์คอมพิวเตอร์เป็นการถ่ายภาพเอกซเรย์ที่มีศักยภาพในการเห็นเนื้อเยื่อสูงซึ่งสามารถเห็นรายละเอียดโดยจากที่การถ่ายเอกซเรย์ทั่วไปไม่สามารถเห็นได้ การสแกนด้วยเครื่องเอกซเรย์คอมพิวเตอร์มีจุดประสงค์หลักในการใช้งานเพื่อตรวจหาความผิดปกติในเนื้อเยื่อกระดูก หรือโครงสร้างของร่างกาย สามารถช่วยบอกตำแหน่งที่แม่นยำในการทำหัตถการทางการแพทย์อีกทั้งยังช่วยในการวินิจฉัยในขั้นแรกเริ่ม หรือช่วยให้เห็นความผิดปกติได้ดี และนำไปสู่ความสำเร็จในการรักษาโรคได้มากขึ้นทั้งยังช่วยในการตรวจสอบโครงสร้างภายในของวัตถุต่างๆ ได้โดยไม่ต้องทำลายส่วนประกอบอื่นจนกลายเป็นเครื่องมือเฉพาะที่มีบทบาทสำคัญในโรงพยาบาลและห้องปฏิบัติการวิจัย (โดยมีรายละเอียดในหัวข้อ 2.1 ของบทที่ 2)

3.2 ศึกษาและค้นคว้าข้อมูลของไมโครคอนโทรลเลอร์และภาษาซี

โครงงานได้ศึกษาคุณสมบัติของไมโครคอนโทรลเลอร์ PIC 16F886 และความรู้ภาษาซีมาใช้ทำหน้าที่การประมวลผล เพื่อควบคุมการทำงานของ สเตปปิ้งมอเตอร์ มอเตอร์ไฟฟ้ากระแสตรง และแสดงผลออกทางจอแอลซีดี (โดยมีรายละเอียดในหัวข้อ 2.2 และ 2.3 ของบทที่ 2)

3.3 ศึกษาและค้นคว้าข้อมูลของสเตปปิ้งมอเตอร์

โครงงานนี้ได้ศึกษาและค้นคว้าการนำไปใช้งานของสเตปปิ้งมอเตอร์เพื่อทำหน้าที่ในปรับการหมุนของฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กตามรอบแกนแนวดิ่ง (โดยมีรายละเอียดในหัวข้อ 2.4 ของบทที่ 2)

3.4 ศึกษาและค้นคว้าข้อมูลของมอเตอร์ไฟฟ้ากระแสตรง และเอนโคดเดอร์

โครงงานนี้ได้ศึกษาและค้นคว้าข้อมูลของมอเตอร์ไฟฟ้ากระแสตรงซึ่งทำหน้าที่เลื่อนระดับขึ้นลงของแผ่นฟิล์มตามแนวดิ่งโดยมีเอนโคดเดอร์ทำการตรวจจับระยะทางในการขึ้นลงในแนวดิ่งของแผ่นฟิล์ม (โดยมีรายละเอียดในหัวข้อ 2.5 และ 2.6 ของบทที่ 2)

3.5 ศึกษาและค้นคว้าข้อมูลของจอแอลซีดี

โครงงานได้ศึกษาและค้นคว้าข้อมูลของจอแอลซีดีเพื่อมาทำหน้าที่แสดงผลของจำนวนระยะทางที่มอเตอร์ไฟฟ้ากระแสตรงดำเนินการได้ (โดยมีรายละเอียดในหัวข้อ 2.7 ของบทที่ 2)

3.6 ออกแบบและพัฒนาโครงสร้างระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

การออกแบบและพัฒนาโครงสร้างระบบเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน คือการออกแบบการทำงานทั้งหมดของระบบ และการนำองค์ความรู้ที่ได้ศึกษาและค้นคว้าดังที่กล่าวมาในบทที่ 2 มาใช้ โดยจะแบ่งการออกแบบโครงสร้างออกแบบเป็น 3 ส่วนหลัก ได้แก่

3.6.1 ออกแบบส่วนโครงรับการทำงานสเตปปิ้งมอเตอร์

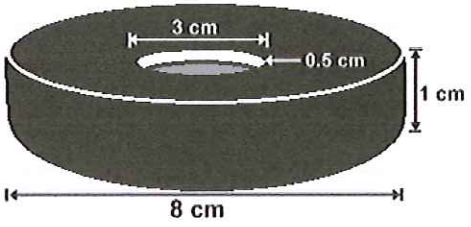
3.6.2 ออกแบบส่วนโครงรับการทำงานมอเตอร์กระแสตรง

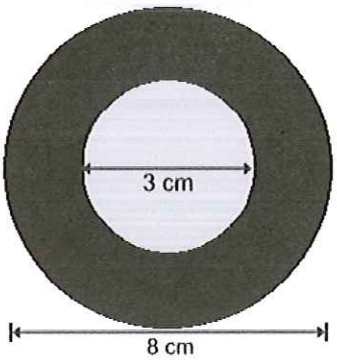
3.6.3 ออกแบบระบบการพัฒนาเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมทำการทดสอบ

มีรายละเอียดของแต่ละส่วน ดังนี้

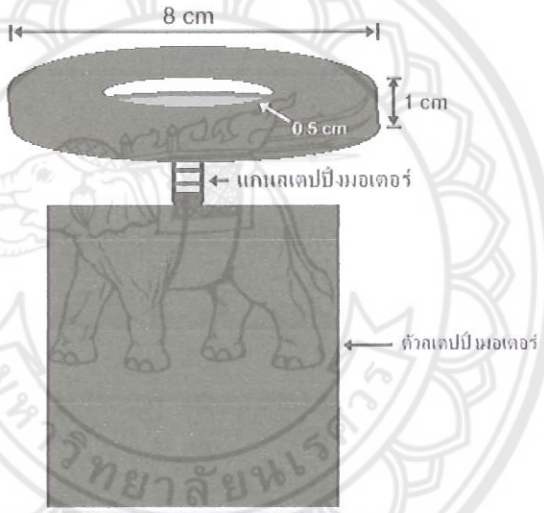
3.6.1 ออกแบบส่วนโครงรับการทำงานสเตปปิ้งมอเตอร์ มีรายละเอียด ดังนี้

3.6.1.1 ฐานรองรับหุ่นจำลองสัตว์ทดลองขนาดเล็ก ทำหน้าที่วางหุ่นจำลองเพื่อทำการฉายแสง โดยฐานนี้ใช้อะลูมิเนียมผ่านการกลึงเป็นแผ่นวงกลมหนา 1 เซนติเมตรมีเส้นผ่านศูนย์กลาง 8 เซนติเมตร เจาะช่องตรงกลางแผ่นอะลูมิเนียมเป็นวงกลมซึ่งจะให้มีเส้นผ่านศูนย์กลางขนาด 4 เซนติเมตร และเจาะลึกลงไป 0.5 เซนติเมตรจากนั้นนำไปประกอบกับแกนสเตปปิ้ง ตามที่แสดงดังรูปที่ 3.1 (รูปที่ 3.1a แสดงด้านข้างของฐาน รูปที่ 3.1b แสดงด้านบนของฐาน และ รูปที่ 3.1c แสดงด้านข้างเมื่อประกอบเข้ากับแกนสเตปปิ้งมอเตอร์)



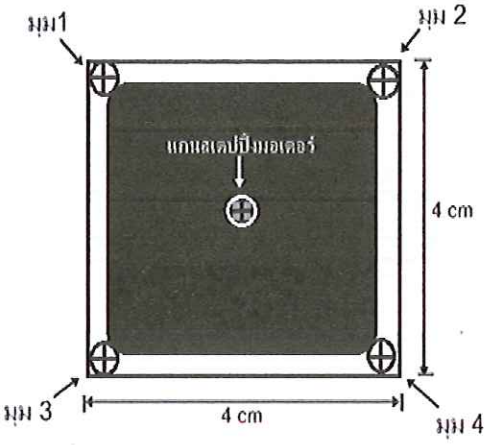


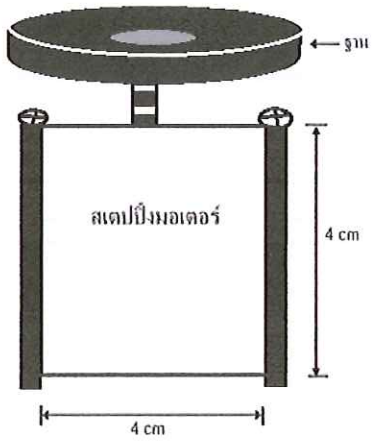
รูปที่ 3.1a ด้านข้างบนของฐาน **รูปที่ 3.1b ด้านบนของฐาน**



รูปที่ 3.1c ด้านข้างฐานกับสเตปปิ้งมอเตอร์

3.6.1.2 ออกแบบการวางโครงรับสเตปปิ้งมอเตอร์ เพื่อทำหน้าที่ยึดกับระบบโดยใช้สกรูเกลียวปล่อยขนาดความยาว 5 เซนติเมตรล็อกลงในแนวดิ่งที่ 4 มุมของสเตปปิ้งมอเตอร์ซึ่งขนาดของจุดที่ล็อกจะเท่ากับมอเตอร์มีขนาดความกว้างและความยาว 4 × 4 เซนติเมตร (จุดที่ 1 คือแกนหมุนของสเตปปิ้งมอเตอร์ซึ่งจะประกอบกับฐานวางหุ่นจำลองสัตว์ทดลองขนาดเล็ก) ตามที่แสดงในรูปที่ 3.2 (รูป 3.2a แสดงด้านบนของการยึดสกรูเกลียวปล่อยของสเตปปิ้งมอเตอร์ รูป 3.2b แสดงด้านข้างของมอเตอร์เมื่อยึดสกรูเกลียวปล่อยของสเตปปิ้งมอเตอร์)

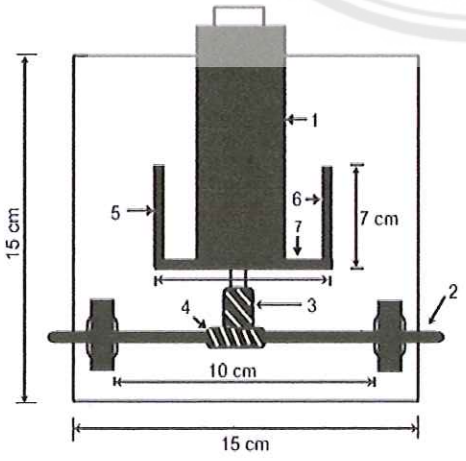


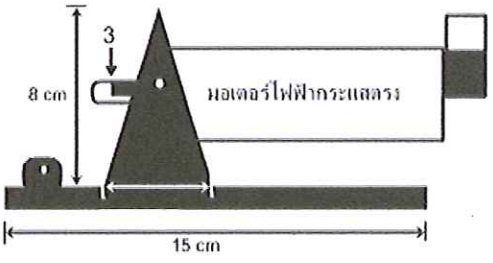


รูปที่ 3.2a ด้านบนสเตปปิ้งมอเตอร์ **รูปที่ 3.2b ด้านข้างสเตปปิ้งมอเตอร์**

3.6.2 ออกแบบส่วนโครงรับการทำงานมอเตอร์กระแสตรง เพื่อใช้ในการหมุนเลื่อนระดับขึ้นลงของแผ่นฟิล์มตามแนวดิ่ง โดยมีรายละเอียดการออกแบบดังนี้

3.6.2.1 โครงรับมอเตอร์ไฟฟ้ากระแสตรง ใช้ฐานวางมอเตอร์ขนาดความกว้างและความยาวขนาด 15×15 เซนติเมตร จากนั้นนำมอเตอร์ที่ประกบเอนโคดเดอร์เข้ากับด้านหลังมอเตอร์มาใส่ส่วนที่ยึดมอเตอร์เข้ากับฐานวาง ซึ่งถูกออกแบบให้มีลักษณะเป็นแผ่นเหล็กสามเหลี่ยมฐานกว้างและความสูงเท่ากับ 7×8 เซนติเมตร จำนวน 2 แผ่นตั้งให้ขนาดกันมีระยะห่าง 6 เซนติเมตร ปิดแกนมอเตอร์ด้วยแผ่นเหล็กสี่เหลี่ยมขนาดความกว้างและความสูง 6×8 เซนติเมตร ประกอบเข้ากับเฟืองตัวหนอน 2 ตัวเพื่อเป็นกลไกยกระดับกล่องใส่ฟิล์ม แสดงตามในรูปที่ 3.3 (รูปที่3.3a แสดงด้านบนของมอเตอร์ และ รูปที่3.3b แสดงด้านบนของมอเตอร์ไฟฟ้ากระแสตรง)





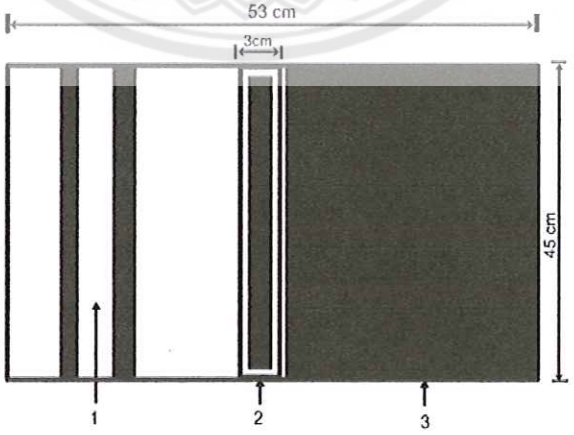
รูปที่ 3.3a ด้านบนของมอเตอร์ไฟฟ้ากรแสตรง **รูปที่ 3.3b ด้านข้างของมอเตอร์ไฟฟ้ากรแสตรง**

จากรูปที่ 3.3a และ รูปที่ 3.3b แสดงรายละเอียดแต่ละจุดดังนี้

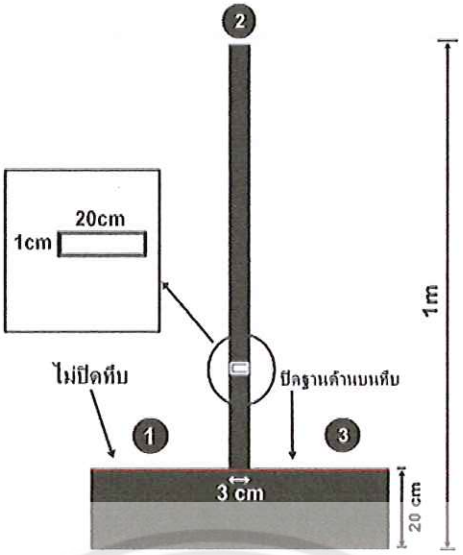
- หมายเลข 1 มอเตอร์ฟ้ากระแสตรงติดเอนโคดเดอร์
- หมายเลย 2 แกนหมุนเฟืองตัวหนอนตัวที่ 2
- หมายเลข 3 เฟืองตัวหนอนตัวที่ 1
- หมายเลข 4 เฟืองตัวหนอนตัวที่ 2
- หมายเลข 5 เหล็กแผ่นสามเหลี่ยมแผ่นที่ 1
- หมายเลข 5 เหล็กแผ่นสามเหลี่ยมแผ่นที่ 2
- หมายเลข 6 เหล็กแผ่นสี่เหลี่ยมผิดแกนมอเตอร์

3.6.3 ออกแบบระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมทำการทดสอบ เป็นการนำส่วนต่างๆที่ได้ออกแบบเบื้องต้นมาระบุตำแหน่งให้เป็นระบบการทำงาน โดยแสดงรายละเอียดการออกแบบ ดังนี้

3.6.3.1 ออกแบบความกว้างและความยาวของอุปกรณ์การพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน โดยสร้างฐานที่มีขนาดความกว้างและความยาว 53×45 เซนติเมตร มีเหล็กกั้นรังสีเอกซเรย์ความสูง 1 เมตร ซึ่งเจาะช่องรับรังสีขนาดความกว้างและความยาว 1×20 เซนติเมตร อยู่สูงจากพื้น 40 เซนติเมตร โดยที่กำหนดเป็นด้านหน้าอุปกรณ์ซึ่งปิดทึบฐานเพื่อวางสเตปปิ้งมอเตอร์ ส่วนด้านหลังจะสร้างกลไกการทำงานของมอเตอร์ไฟฟ้ากระแสตรงนำเหล็ก 2 ท่อนขนาดเท่ากับความกว้างของฐานเชื่อมขนานไว้เพื่อวางมอเตอร์ไฟฟ้ากระแสตรง โดยแสดงในรูปที่ 3.4a และรูปที่ 3.4b



รูปที่ 3.4a แสดงด้านบนของอุปกรณ์

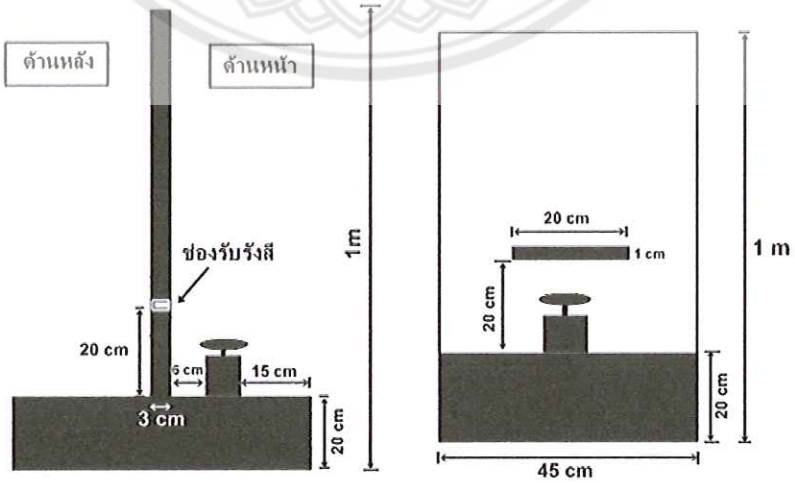


รูปที่ 3.4b แสดงด้านข้างของอุปกรณ์

จากรูปที่ 3.4a และรูปที่ 3.4b แสดงรายละเอียดแต่ละจุดดังนี้

- จุดที่ 1 คือด้านหลังอุปกรณ์ มีเหล็ก 2 ท่องเชื่อมขนานไว้เพื่อวางมอเตอร์ไฟฟ้ากระแสตรง
- จุดที่ 2 คือแผ่นเหล็กกั้นรังสีเอกซเรย์ที่เจาะช่องรับรังสีมีขนาดความกว้างและความยาว 1×20 เซนติเมตร
- จุดที่ 3 คือด้านหน้าเครื่องฐานด้านบนปิดทึบทั้งหมดเพื่อวางสเตปปิ้งมอเตอร์

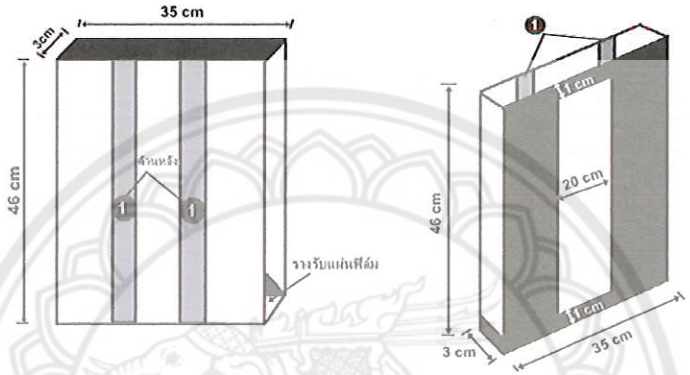
3.6.3.2 ออกแบบการวางสเตปปิ้งมอเตอร์ ซึ่งจะวางไว้ด้านหน้าอุปกรณ์ให้ลึกเข้าไปทางเหล็กกั้นรังสีเอกซเรย์ 15 เซนติเมตร และห่างออกจากเหล็กกั้นรังสีเอกซเรย์ 6 เซนติเมตร ตามที่แสดงในรูปที่ 3.5a แสดงด้านข้างของอุปกรณ์ และรูปที่ 3.5b



รูปที่ 3.5a ด้านข้างของตำแหน่งสเตปปิ้งมอเตอร์ รูปที่ 3.5b ด้านหน้าของตำแหน่งสเตปปิ้งมอเตอร์

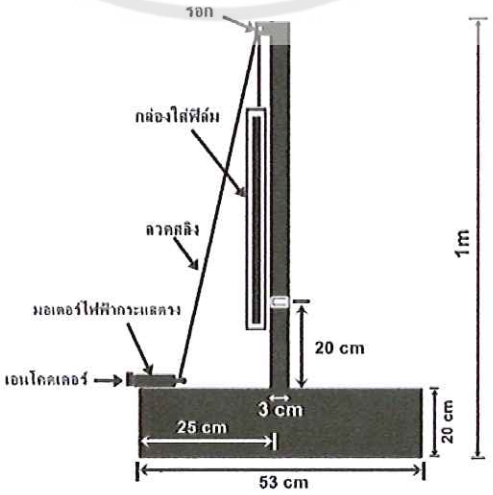
เ๗198534 ป พ3521 2557 17 ตค. 2560 สำนักหอสมุด

3.6.3.3 ออกแบบการวางมอเตอร์ไฟฟ้ากระแสตรง ซึ่งการออกแบบนี้รวบไปถึงกลไกการทำงานการเลื่อนระดับของแผ่นฟิล์มในการจะให้แผ่นฟิล์มเลื่อนระดับได้นั้นต้องมีอุปกรณ์รองรับแผ่นฟิล์มโดยมีการออกแบบเป็นกล่องใส่พอดีกับแผ่นฟิล์มที่นำมาทำการทดสอบส่วนประกอบที่ออกแบบขึ้นมาคือมีแผ่นปิดด้านหน้าที่มีช่องรับรังสีเอกซเรย์ขนาดความกว้างและความยาว 1×20 มีแผ่นเหล็กประกบประกบปิดด้านหลังขนาดความกว้างและความยาว 10×46 จำนวน 2 แผ่น (หมายเลข 1) และรางรับฟิล์มด้านล่าง จากนั้นนำส่วนต่างๆมาประกอบกันในตำแหน่งที่แสดงในรูปที่ 3.6a และ รูปที่ 3.6b



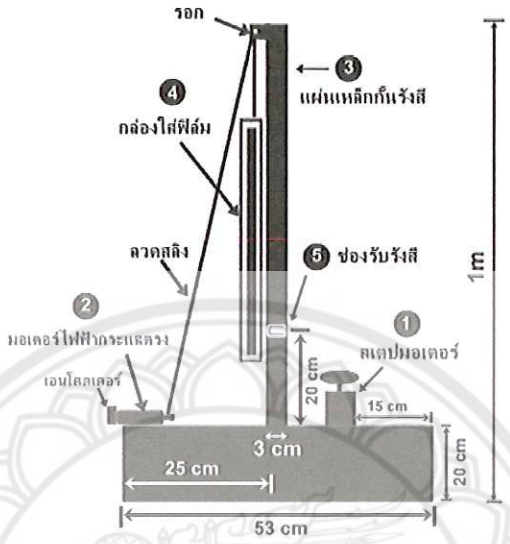
รูปที่ 3.6a ด้านหลังของกล่องใส่ฟิล์ม รูปที่ 3.6b ด้านหน้าของกล่องใส่ฟิล์ม

เมื่อมีการออกแบบกล่องใส่ฟิล์มแล้วจึงได้ออกแบบตำแหน่งการวางกลไกของมอเตอร์ไฟฟ้ากระแสตรง โดยนำโครงรับมอเตอร์ที่ได้ออกแบบไว้ในหัวข้อที่ 3.6.2 ในบทที่ 3 มาวางบนเหล็ก 2 ท่อนที่ได้ออกแบบไว้ในหัวข้อ 3.6.2.1 ในบทที่ 3 จากนั้นออกแบบกลไกการเลื่อนระดับขึ้นลงตามแนวดิ่งโดยใช้ลวดสลิงพ่วงกับรอก 2 ตัวโยงกับกล่องรับฟิล์ม และผูกลวดสลิงกับมอเตอร์ไฟฟ้ากระแสตรง ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 แสดงด้านข้างของตำแหน่งและกลไกการเลื่อนระดับขึ้นลงของแผ่นฟิล์ม

จากการออกแบบส่วนประกอบมานั้นเมื่อนำทุกส่วนมาประกอบกันได้ระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมทำการทดสอบ แสดงดังรูปที่ 3.8 มีรายละเอียดของตำแหน่งดังนี้ ได้ดังนี้



รูปที่ 3.8 ระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

จากภาพที่ 3.8 การใช้อุปกรณ์ในหมายเลขต่างๆรวมทั้งบอกตำแหน่งการต่ออุปกรณ์ของระบบเข้ากับวงจรควบคุมโดยมีการออกแบบวงจรตามรูปที่ 3.12 (หัวข้อที่ 3.8 บทที่ 3 หน้าที่ 31) แสดงรายละเอียดดังนี้

- หมายเลข 1 สเตปปิ้งมอเตอร์ต่อเข้ากับพอร์ตที่ 3 ของวงจรควบคุม
- หมายเลข 2 มอเตอร์ไฟฟ้ากระแสตรงต่อเข้ากับพอร์ตที่ 4 ของวงจรโดยที่เอนโคดเดอร์ต่อตรงกับมอเตอร์ไฟฟ้ากระแสตรง
- หมายเลข 3 ฉากกั้นรังสีเจาะเป็นช่องว่างสูงห่างจากฐานขึ้นมาก 20 เซนติเมตรซึ่งมีขนาดความกว้างและความยาว 1×20 เซนติเมตร
- หมายเลข 4 กล่องใส่แผ่นฟิล์มโดยถูกโยงกับลวดสลิงซึ่งเมื่อมอเตอร์ไฟฟ้ากระแสตรงหมุนแผ่นฟิล์มจะปรับระดับขึ้นลงตามแนวดิ่ง
- หมายเลข 5 ช่องว่างที่ถูกเจาะมาเพื่อรับรังสีที่ฉายไปยังหุ่นจำลองสัตว์ทดลองขนาดเล็กทำให้ได้ภาพบนฟิล์มมีขนาดความกว้างและความยาว 1×20 เซนติเมตร

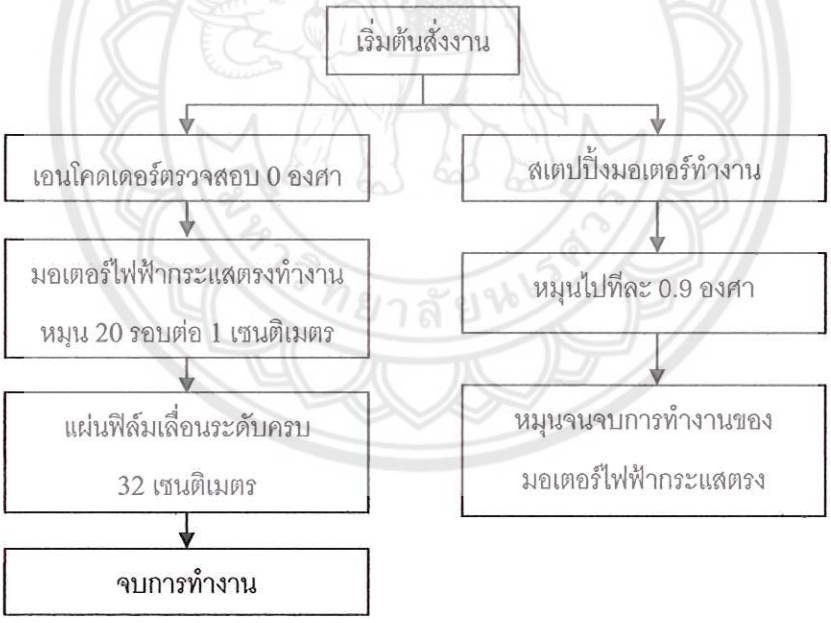
จากที่ออกแบบระบบการทำงานเมื่อเลื่อนระดับขึ้นลงตามแนวดิ่งต้องแสดงผลออกทางจอแอลซีดีเพื่อทราบผลระยะทางที่มอเตอร์ไฟฟ้ากระแสตรงได้หมุนตามการตรวจสอบองศาของเอนโคดเดอร์ซึ่งต่อแอลซีดีที่พอร์ต 5 (จากรูปที่ 3.12 หน้า 31) เป็นสัญญาณส่งออกของวงจร

3.7 ออกแบบระบบควบคุมการทำงานของสเตปปิ้งมอเตอร์และมอเตอร์ไฟฟ้ากระแสตรง

ออกแบบระบบการประมวลผลการทำงานของไมโครคอนโทรลเลอร์ PIC 16F886 เพื่อควบคุมการหมุนของสเตปปิ้งมอเตอร์ และมอเตอร์ไฟฟ้ากระแสตรง ให้แสดงผลออกทางจอแอลซีดี การออกแบบอุปกรณ์นี้จะออกแบบตามวิธีการดำเนินขั้นตอนการฉายแสงเพื่อนำผลการฉายรังสีไปวินิจฉัยในทางการแพทย์ โดยที่การฉายรังสีให้กับหุ่นจำลองสัตว์ทดลองขนาดเล็กแต่ละครั้งจะต้องได้ขนาดความสูงครั้งละ 1 เซนติเมตร พร้อมกับหมุนลำตัวไป 0.9 องศาจนสิ้นสุดแผ่นฟิล์ม

เมื่อสิ้นสุดแผ่นฟิล์มต้องมีการสำรองการฉายรังสีของลำตัวในองศาอื่นๆจึงมีการดำเนินการเฉพาะการหมุนสเตปปิ้งมอเตอร์โดยที่มอเตอร์กระแสตรงไม่ต้องทำงาน เพราะฉะนั้นสเตปปิ้งมอเตอร์ต้องทำงานได้ครบ 360 องศา และเมื่อดำเนินการฉายภาพครบตามขนาดแผ่นฟิล์ม 32 ระดับแล้วรีเซตกล่องใส่แผ่นฟิล์มมาที่ระดับเริ่มต้นเพื่อพร้อมใช้งานในครั้งต่อไป ซึ่งจะลำดับการทำงานของระบบควบคุมเป็น 3 ส่วนการทำงาน โดยมีรายละเอียดดังนี้

3.7.1 ควบคุมการหมุนสเตปปิ้งมอเตอร์ และการหมุนมอเตอร์ไฟฟ้ากระแสตรงพร้อมกันทั้ง 2 ตัว โดยมีการทำงานตามแผนภาพรูปที่ 3.9



รูปที่ 3.9 แผนภาพแสดงการทำงานของสเตปปิ้งมอเตอร์และมอเตอร์ไฟฟ้ากระแสตรง

จากรูปที่ 3.9 เริ่มต้นการใช้งานรับคำสั่งให้เอนโคดเดอร์ตรวจสอบองศาการหมุนของมอเตอร์ไฟฟ้ากระแสตรงเมื่อตรวจสอบได้ 0 องศาให้เริ่มดำเนินการหมุน 20 รอบจะได้ระยะทาง 1 เซนติเมตร จากนั้นทำการกำหนดให้หมุนระดับละ 20 รอบต่อการทำงานต่อไปจนครบ 32 เซนติเมตร พร้อมทั้งรับคำสั่งให้สเตปปิ้งตอเตอร์เริ่มทำงานโดยที่หมุนไปทีละ 0.9 องศา จนจบการทำงานของมอเตอร์ไฟฟ้ากระแสตรง แล้วจบการทำงาน

3.7.2 ควบคุมการหมุนสเตปปิ้งมอเตอร์ตัวเดียว โดยมีการทำงานตามแผนภาพรูปที่ 3.10

รูปที่ 3.10 แผนภาพแสดงการทำงานของสเตปปิ้งมอเตอร์

จากรูปที่ 3.10 เริ่มต้นการใช้งานรับคำสั่งให้สเตปปิ้งตอเตอร์เมื่ออยู่ที่ 0 องศาเทียบกับหน่วยวัดที่ฐานหุ่นจำลองสัตว์ทดลองขนาดเล็ก โดยมอเตอร์จะหมุนไปทีละ 0.9 องศาทำงานจนครบ 360 องศา จบการทำงาน

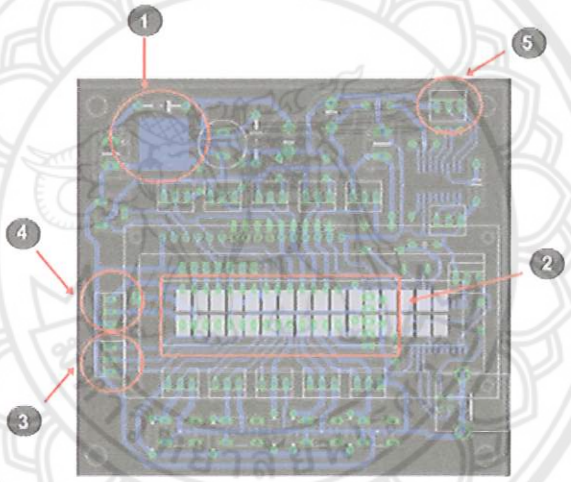
3.7.3 ควบคุมการหมุนกลับทิศมอเตอร์ไฟฟ้ากระแสตรง โดยมีการทำงานตามแผนภาพรูปที่ 3.11

รูปที่ 3.11 แผนภาพแสดงการหมุนกลับทิศของมอเตอร์ไฟฟ้ากระแสตรง

จากรูปที่ 3.11 เริ่มต้นการใช้งานโดยรับคำสั่งให้เอนโคดเดอร์ตรวจสอบองศาการหมุนไปที่มอเตอร์ไฟฟ้ากระแสตรงเมื่อได้ดำเนินการเลื่อนระดับไปที่ 32 เซนติเมตรจากนั้นเอนโคดเดอร์จะกำหนดให้มอเตอร์ไฟฟ้ากระแสตรงหมุนกลับทิศทางซึ่งได้กำหนดให้หมุน 20 รอบต่อการเลื่อนระดับแผ่นฟิล์ม 1 เซนติเมตร และจะแสดงผลทุกระยะออกทางจอแอลซีดีจนครบ 0 เซนติเมตร จบการทำงาน

3.8 ออกแบบลายปริ้นวงจรและกล่องควบคุมชุดไมโครคอนโทรลเลอร์ PIC 16F886

3.8.1 การออกแบบลายปริ้นวงจร ซึ่งวงจรนี้ได้ทำหน้าที่ควบคุมการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน การออกแบบได้ใช้โปรแกรม Eagle layout edition จากนั้นจึงได้ออกแบบกล่องเพื่อเก็บอุปกรณ์ชุดไมโครคอนโทรลเลอร์ PIC F886 แสดงอุปกรณ์ที่ใช้ออกแบบวงจรในรูปที่ 3.12 มีรายละเอียดดังนี้

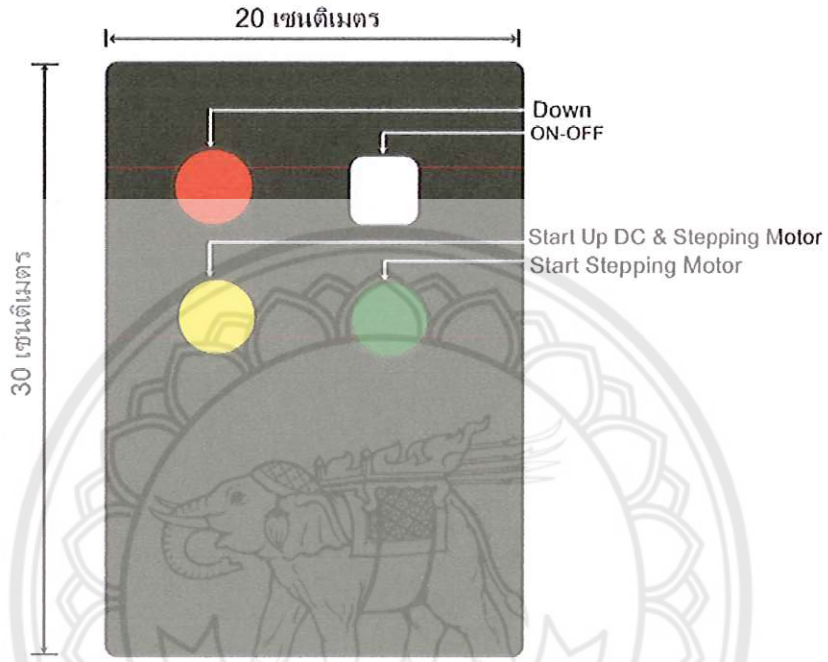


รูปที่ 3.12 รูปแบบลายปริ้นที่ออกแบบ

สำหรับบอร์ดไมโครคอนโทรลเลอร์เป็นตัวอย่างในการออกแบบก่อนทำการทดสอบซึ่งอุปกรณ์ที่ต่อภายในวงจรจะประกอบด้วย

- จุดที่ 1 พอร์ต USB เพื่อประมวลปลการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน
- จุดที่ 2 ไม่โครคอนโทรลเลอร์ PIC 16F886 เพื่อใช้ในการประมวลผลการทำงานของระบบ
- จุดที่ 3 สัญญาณรับเข้าของของสเตปปิ้งมอเตอร์เพื่อสั่งการให้มอเตอร์หมุนตามองศา
- จุดที่ 4 สัญญาณรับเข้าของเอนโคดเดอร์และมอเตอร์ไฟฟ้ากระแสตรงเพื่อตรวจสอบองศาและสั่งการให้มอเตอร์ไฟฟ้ากระแสตรงให้ทำการหมุน
- จุดที่ 5 สัญญาส่งออกของจอแอลซีดีเพื่อแสดงผลของระยะทางที่มอเตอร์ไฟฟ้ากระแสตรงได้หมุนตามการตรวจสอบของเอนโคดเดอร์

3.8.2 การออกแบบกล่องควบคุมไมโครคอนโทรเลอร์ PIC 16F886 เพื่อจัดเก็บอุปกรณ์ที่เป็นระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนให้อยู่ในส่วนของการพร้อมใช้งาน โดยโครงงานนี้ได้ออกแบบกล่องควบคุมที่มีขนาดความกว้างและความยาว 20×30 เซนติเมตร มีฟังก์ชั่นการใช้งานดังแสดงในรูปที่ 3.13



รูป 3.13 ออกแบบกล่องไมโครคอนโทรลเลอร์

จากรูปที่ 3.13 แสดงรายละเอียดของฟังก์ชั่นการใช้งานได้ดังนี้

- สีขาว เป็นปุ่มควบคุมการปิดเปิดอุปกรณ์พร้อมใช้งานทั้งระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน
- สีเหลือง เป็นปุ่มควบคุมการเลื่อนระดับขึ้นลงในแนวดิ่งของมอเตอร์ไฟฟ้ากระแสตรงและหมุนตามแนวแกนดิ่งของสเตปปิ้งมอเตอร์ซึ่งทำงานพร้อมกัน
- สีเขียว เป็นปุ่มควบคุมการหมุนของสเตปปิ้งมอเตอร์เพียงตัวเดียว
- สีแดง เป็นปุ่มควบคุมการหมุนกลับของมอเตอร์ไฟฟ้ากระแสตรงในแนวดิ่ง

3.9 ทดสอบระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

ทดสอบระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนขั้นตอนนี้จะเป็นการนำอุปกรณ์ที่พัฒนาขึ้นมาทดสอบเพื่อเก็บผลการทดสอบแบ่งการทดสอบได้ดังนี้

- การทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการหมุนฐานหุ่นจำลองสัตว์ทดสองขนาดเล็กรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา
- การทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร (โดยมีรายละเอียดของบทที่ 4)

3.10 วิเคราะห์ผลการทดสอบ สรุปผลการทดสอบ ปัญหาและข้อเสนอแนะ

จากการทดสอบและสรุปผลการทดสอบของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนได้แก่

- วิเคราะห์ผล การทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการหมุนฐานหุ่นจำลองสัตว์ทดสองขนาดเล็กรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา
- วิเคราะห์ผล การทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร (โดยมีรายละเอียดของบทที่ 4 และบทที่ 5)

3.11 จัดทำรูปเล่มปริญญานิพนธ์

มีการแก้ไขปรับปรุงรูปเล่มเรียบเรียงข้อมูลพร้อมทั้งการจัดทำรูปเล่มปริญญานิพนธ์

บทที่4
ผลการทดสอบและการวิเคราะห์ผลการทดสอบ

ในบทที่ 4 จะกล่าวถึงการทดสอบประสิทธิภาพการทำงานของการพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนที่ได้ออกแบบ และพัฒนาขึ้นด้วยการใช้ไมโครคอนโทรลเลอร์ PIC 16F886 โดยมีผลการทดสอบ การวิเคราะห์ผลการทดสอบ และสรุปผลการทดสอบ รายละเอียดดังนี้

ในการทดสอบประสิทธิภาพการทำงานของการพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน แบ่งออกได้ 3 ผลการทดสอบ ผลการทดสอบการออกแบบการพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน ผลการทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการหมุนฐานหุ่นจำลองสัตว์ทดสองขนาดเล็กรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา ผลการทดสอบการทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร ซึ่งเป็นการทดสอบเพื่อให้เข้าใจถึงศักยภาพการทำงานของอุปกรณ์ที่พัฒนาขึ้น

4.1 ผลการทดสอบการออกแบบระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

ผลการทดสอบการออกแบบระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพัฒนาจากการออกแบบในส่วนต่างๆแบ่งได้ 3 ส่วน ได้แก่

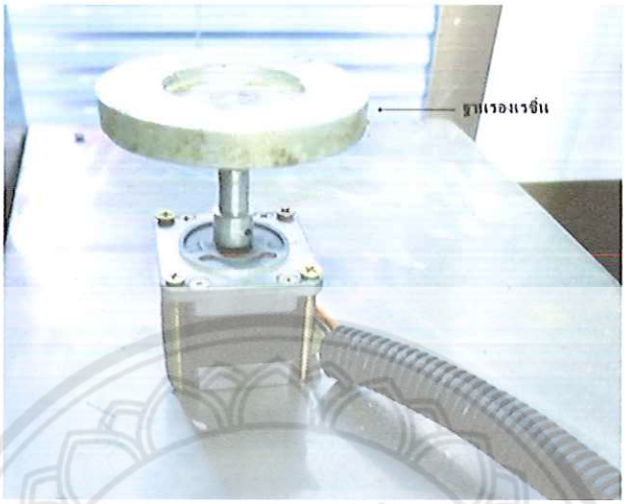
4.1.1 โครงรับการทำงานของสเตปปิ้งมอเตอร์

4.1.2 โครงรับการทำงานมอเตอร์กระแสตรง

4.1.3 โครงสร้างของอุปกรณ์การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมใช้งานได้

โดยมีรายละเอียดโครงสร้างแต่ละส่วน ดังนี้

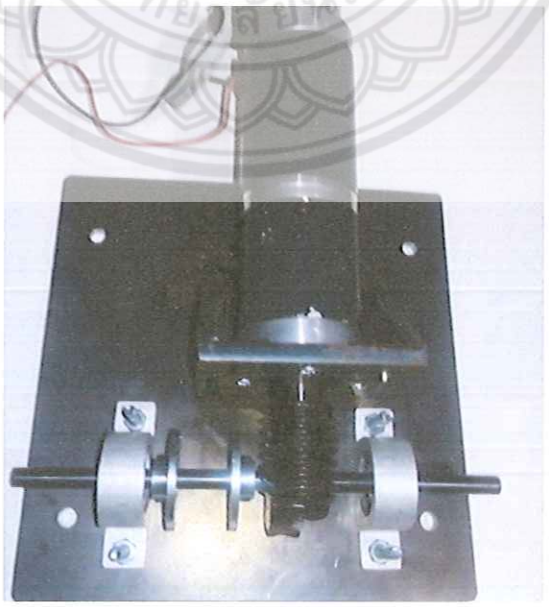
4.1.1 โครงรับการทำงานของสเตปปิ้งมอเตอร์ ซึ่งจะทำหน้าที่ในปรับการหมุนของฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กตามรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา ผลการออกแบบได้แสดงในรูปที่ 4.1



รูปที่ 4.1 โครงรับการทำงานของสเตปปิ้งมอเตอร์

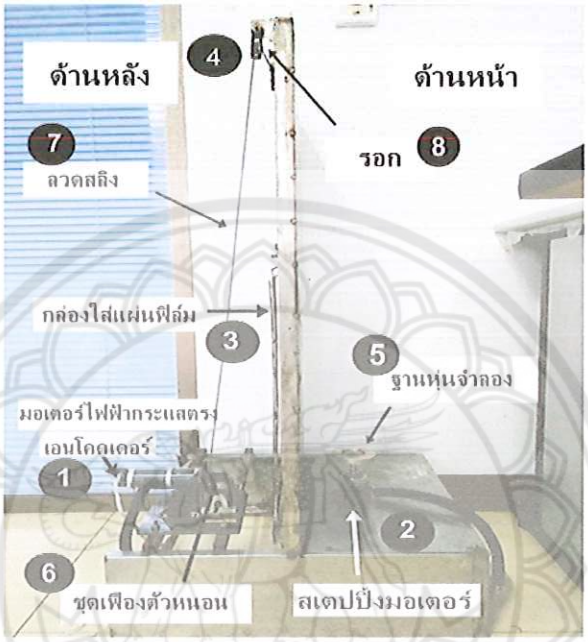
4.1.2 โครงรับการทำงานมอเตอร์กระแสตรง

โครงรับการทำงานมอเตอร์กระแสตรงการออกแบบส่วนเพื่อทำหน้าที่ปรับระดับขึ้นลงของแผ่นฟิล์มตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร ผลการออกแบบจะได้ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 โครงรับการทำงานมอเตอร์กระแสตรง

4.1.3 โครงสร้างอุปกรณ์พัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมทำการทดสอบ ซึ่งจะนำส่วนประกอบที่ได้สร้างขึ้นแล้ว ได้แก่ ฐานรองรับหุ่นจำลองสัตว์ทดลองขนาดเล็ก โครงรับสเตปปิ้งมอเตอร์ โครงรับการทำงานมอเตอร์กระแสตรง อุปกรณ์รองรับแผ่นฟิล์ม กลไกการทำงานของมอเตอร์ไฟฟ้ากระแสตรง โดยจะมีตำแหน่งของส่วนประกอบดังแสดงตามรูปที่ 4.3



รูปที่ 4.3 โครงสร้างการพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนพร้อมใช้งานได้

โดยมีรายละเอียด ดังนี้

- หมายเลข 1 คือ มอเตอร์ไฟฟ้ากระแสตรงด้านหลังมีเอนโคดเดอร์ประกอบยึดติดกับด้านหลังของอุปกรณ์
- หมายเลข 2 คือ สเตปปิ้งมอเตอร์ยึดติดกับด้านหน้าของอุปกรณ์
- หมายเลข 3 คือ กล่องใส่แผ่นฟิล์มติดกับด้านหลังอุปกรณ์
- หมายเลข 4 คือ กลไกยกระดับขึ้นลงในแนวดิ่งของแผ่นฟิล์ม
- หมายเลข 5 คือ ฐานเพื่อหุ่นจำลองสัตว์ทดลองขนาดเล็ก
- หมายเลข 6 คือ ชุดฟันเฟืองเป็นกลไกของมอเตอร์ไฟฟ้ากระแสตรง
- หมายเลข 7 คือ ลวดสลิงซึ่งผูกติดกับกล่องใส่แผ่นฟิล์มโยงผ่านรอกมายึดกับชุดฟันเฟืองของมอเตอร์ไฟฟ้ากระแสตรง
- หมายเลย 8 คือ รอก 2 ตัวเป็นกลไกการยกระดับแผ่นฟิล์มให้ขึ้นในแนวดิ่ง

4.2 ผลการทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน โดยจะทำการทดสอบการหมุนฐานหุ่นจำลองสัตว์ทดสองขนาดเล็กรอบแกนแนวดิ่ง หมุนไปทีละ 0.9 องศาจนถึง 360 องศา

การทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนที่พัฒนาขึ้นในการทดสอบนี้กล่าวถึงผลการทดสอบและการวิเคราะห์ผลการทดสอบของการหมุนสเตปปิ้งมอเตอร์โดยการประมวลผลจากไมโครคอนโทรลเลอร์ PIC 16F886 ให้สามารถหมุนรอบแนวแกนดิ่งเริ่มทำการทดสอบโดยที่ตั้งค่าไว้ 0 องศา จากนั้นจึงทำการทดสอบโดยที่เปรียบเทียบองศาไปทีละ 0.9 องศาจนถึง 360 องศา และบันทึกผลการทดสอบ ซึ่งมีผลการทดสอบแสดงในดังตารางที่ 4.1

ตารางที่ 4.1 ตารางการหมุนองศา (ผลการทดสอบบางส่วน) เริ่มต้น 0 องศา ถึง 360 องศา

องศาการหมุนสเตปปิ้งมอเตอร์ (องศา)	องศาที่วัดได้จากอุปกรณ์ (องศา)	ความคลาดเคลื่อน (%)
0	0	0
0.9	0.91	0.01
1.8	1.81	0.01
2.7	2.71	0.01
3.6	3.61	0.01
4.5	4.52	0.02
5.4	5.41	0.01
6.3	6.31	0.01
7.2	7.21	0.01
8.1	8.11	0.01
9.0	9.02	0.02
27.0	27.02	0.02
27.9	27.92	0.02
28.8	28.82	0.02
29.7	29.71	0.01
30.6	30.62	0.02

องศาการหมุนสเตปปิ้งมอเตอร์ (องศา)	องศาที่วัดได้จากอุปกรณ์ (องศา)	ความคลาดเคลื่อน (%)
32.4	32.41	0.01
33.3	33.31	0.01
34.2	34.21	0.01
35.1	35.12	0.02
72.0	72.02	0.02
72.9	72.92	0.02
73.8	73.82	0.02
74.7	74.72	0.02
75.6	75.61	0.01
76.5	76.51	0.01
77.4	77.41	0.01
78.3	78.31	0.01
79.2	79.21	0.01
99.0	99.02	0.02
99.9	99.92	0.02
100.8	100.81	0.01
101.7	101.71	0.01
102.6	102.62	0.02
103.5	103.52	0.02
104.4	104.41	0.01
105.3	105.31	0.01
106.2	106.22	0.02
107.1	107.13	0.03
108.0	108.01	0.01
252.0	252.02	0.01
252.9	252.91	0.01
253.8	253.82	0.02
254.7	254.71	0.01
255.6	255.62	0.02

องศาการหมุนสเตปปิ้งมอเตอร์ (องศา)	องศาที่วัดได้จากอุปกรณ์ (องศา)	ความคลาดเคลื่อน (%)
256.5	256.52	0.02
340.2	340.23	0.03
341.1	341.11	0.01
342.0	342.01	0.01
342.9	342.93	0.03
343.8	343.83	0.03
344.7	344.73	0.03
345.6	345.61	0.01
346.5	346.51	0.01
351.9	351.91	0.01
352.8	352.81	0.01
353.7	353.71	0.01
354.6	354.61	0.01
355.5	355.51	0.01
356.4	356.42	0.02
357.3	357.32	0.02
358.2	358.22	0.02
359.1	359.11	0.01
360.0	360.02	0.02

วิเคราะห์ผลการทดสอบ

จากตารางที่ 4.1 เป็นการทดสอบสเตปปิ้งมอเตอร์ที่ทำการหมุนฐานหุ่นจำลองสัตว์ทดสองขนาดเล็กรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา โดยการวิเคราะห์ผลการทดสอบนี้ (ผลการทดสอบการทำงานของสเตปปิ้งมอเตอร์ทั้งหมดแสดงในภาคผนวก ก) ซึ่งประเมินจากผลการทดสอบแล้วจะเห็นว่าได้ผลลัพธ์ที่ได้มีความคลาดเคลื่อนสูงสุดที่ 0.03 องศา ความคลาดเคลื่อนต่ำสุดที่ 0.01 องศา และความคลาดเคลื่อนเฉลี่ย 0.02 องศา ไม่สามารถกำหนดได้ว่าการหมุนองศาไหนแกว่งไปเท่าใด เนื่องจากความผิดพลาดเกิดขึ้นจากการแกว่งของการหมุนของมอเตอร์และฐานซึ่งผลของความคลานเคลื่อนไม่เป็นผลสะสม โดยตามจุดประสงค์กำหนดให้หมุนไปทีละ 0.9 องศาจนถึง 360 องศาความคลาดเคลื่อนที่เกิดขึ้นจึงเป็นผลการทดสอบที่ยอมรับได้

4.3 ผลการทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยจะทำการทดสอบการเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร

การทดสอบประสิทธิภาพของระบบการทำงานของระบบการพัฒนาเครื่องเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนที่พัฒนาขึ้นในการทดสอบนี้กล่าวถึงผลการทดสอบและการวิเคราะห์ผลการทดสอบของการหมุนของมอเตอร์ไฟฟ้ากระแสตรงโดยการประมวลผลจากไมโครคอนโทรลเลอร์ PIC 16F886 ให้สามารถเลื่อนระดับฟิล์มขึ้นลงตามแนวดิ่ง เริ่มทำการทดสอบโดยที่ตั้งค่าไว้ 0 เซนติเมตร จากนั้นจึงทำการทดสอบโดยที่เปรียบเทียบไปทีละ เซนติเมตรจนถึง 32 เซนติเมตร ซึ่งมีการทดสอบ 2 การทดสอบโดยมีรายละเอียด และผลการทดสอบดังนี้

4.3.1 การทดสอบการหมุนของมอเตอร์ไฟฟ้ากระแสตรงจากการประมวลผลของไมโครคอนโทรลเลอร์ PIC 16F886 โดยให้เอนโคดเดอร์ตรวจสอบองศาการหมดจากการกำหนดไว้ที่ 360 องศาต่อ 1 รอบการหมุนของมอเตอร์ไฟฟ้ากระแสตรงการทดลองนี้ไม่มีภาระของน้ำหนักกล่องใส่แผ่นฟิล์มรวมกับแผ่นฟิล์มที่ติดตั้งขึ้นมา และไม่มีการวัดระยะทางตามแนวดิ่ง ผลการทดสอบ ทดสอบมอเตอร์ไฟฟ้ากระแสตรงจากสภาพไม่มีภาระของน้ำหนักกล่องใส่ฟิล์มรวมกับแผ่นฟิล์ม ผลที่ได้คือมอเตอร์ไฟฟ้ากระแสตรงหมุนได้ 32 รอบ จากนั้นหมุนกลับทิศทางอีก 32 รอบ

4.3.2 การทดสอบนี้เป็นการทดสอบการหมุนของมอเตอร์ไฟฟ้ากระแสตรง ซึ่งการทดสอบนี้พ่วงกับอุปกรณ์การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน โดยมีภาระของน้ำหนักกล่องใส่ฟิล์มรวมกับแผ่นฟิล์ม มีน้ำหนักสุทธิ 1,730 กรัม และบันทึกผลทดสอบมีการวัดระยะทางตามแนวดิ่งเมื่อมอเตอร์ไฟฟ้ากระแสตรงหมุนให้แผ่นฟิล์มเลื่อนระดับขึ้นลงระดับละ 1 เซนติเมตร เนื่องจากมีภาระของน้ำหนักกล่องใส่ฟิล์มรวมกับแผ่นฟิล์มทำให้มอเตอร์หมุน 360 องศาไม่สามารถเลื่อนระดับได้ 1 เซนติเมตร จึงได้เพิ่มองศาการหมุนให้เอนโคดเดอร์ตรวจสอบการหมุนของมอเตอร์ไฟฟ้ากระแสตรงพบว่ามอเตอร์ไฟฟ้ากระแสตรงจะหมุน 20 รอบต่อการเลื่อนระดับขึ้นลงตามแนวดิ่งของฟิล์มได้ระยะทาง 1 เซนติเมตร จากนั้นจึงทำการทดสอบไปจนถึง 32 เซนติเมตร ผลการทดสอบที่ได้ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 ตารางการปรับระดับ เริ่มต้น 20 เซนติเมตร วัดจากพื้น ถึง 32 เซนติเมตร

ระดับของตลับฟิล์ม	ระยะที่เลื่อนระดับเริ่มจากฐาน (เซนติเมตร)	ระยะทดสอบที่วัดได้ (เซนติเมตร)	ความคลาดเคลื่อน (%)
1	1	1.2	0.2%
2	2	2.2	0.2%
3	3	3.2	0.2%
4	4	4.0	0.1%
5	5	5.1	0.1%
6	6	6 .3	0.3%
7	7	7.0	0.1%
8	8	8.1	0.1%
9	9	9.2	0.2%
10	10	10.2	0.2%
11	11	11.2	0.2%
12	12	12.1	0.1%
13	13	13.1	0.1%
14	14	14 .1	0.1%
15	15	15.3	0.3%
16	16	16.2	0.2%
17	17	17.2	0.2%
18	18	18.0	0.1%
19	19	19.1	0.1%
20	20	20.1	0.1%
21	21	21.1	0.1%
22	22	22.0	0.1%
23	23	23.1	0.1%
24	24	24.2	0.2%
25	25	25.1	0.1%
26	26	26.0	0.1%
27	27	27.1	0.1%

ระดับของตลับฟิล์ม	ระยะที่เลื่อนระดับเริ่มจากฐาน (เซนติเมตร)	ระยะทดสอบที่วัดได้ (เซนติเมตร)	ความคลาดเคลื่อน (%)
28	28	28.0	0.1%
29	29	29.2	0.2%
30	30	30.3	0.3%
31	31	31.1	0.1%
32	32	32.1	0.1%

วิเคราะห์ผลการทดสอบ

จากตารางที่ 4.2 ซึ่งการทดสอบนี้จะทำการทดสอบมอเตอร์ไฟฟ้ากระแสตรงที่ทำการเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร ประเมินผลการทดสอบแล้วจะเห็นว่าผลลัพธ์ที่ได้มีความคลาดเคลื่อนที่สูงสุด 0.3 เซนติเมตร ความคลาดเคลื่อนต่ำสุด 0.1 เซนติเมตร และความคลาดเคลื่อนเฉลี่ย 0.2 เซนติเมตร ซึ่งถ้าเปรียบเทียบผลตามตาราง พบว่าระดับที่ 6,15 และ 30 มีความคลาดเคลื่อนสูงสุด และความคลาดเคลื่อนส่วนใหญ่อยู่ที่ 0.1 เนื่องจากลวดสลิงมีการแกว่งเล็กน้อยทำให้การวัดจึงมีความผิดพลาดไปจากผลจริง ซึ่งตามจุดประสงค์มีกำหนดให้เลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตร ดังนั้นความคลาดเคลื่อนที่เกิดขึ้นจึงเป็นผลการทดสอบที่ยอมรับการใช้งานได้

4.4 สรุปผลการทดสอบ

การทดสอบการออกแบบการพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยนำส่วนที่ได้สร้างจากการออกแบบมาประกอบสามารถการทำการทดสอบได้อย่างเป็นระบบ

ผลการทดสอบของสเตปปิ้งมอเตอร์โดยหมุนฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กรอบแกนแนวดิ่งซึ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศาซึ่งสามารถหมุนจนครบรอบมอเตอร์ ซึ่งการทดสอบได้เกิดความผิดพลาดจากการแกว่งของแกนหมุนมอเตอร์และฐาน มีผลความคลาดเคลื่อนเฉลี่ยอยู่ที่ 0.02 องศา โดยความคลาดเคลื่อนนี้ไม่มีผลต่อการใช้งาน

ผลการทดสอบของมอเตอร์ไฟฟ้ากระแสตรงโดยเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 จนถึง 32 เซนติเมตรซึ่งสามารถเลื่อนระดับขึ้นลงตามแนวดิ่งของแผ่นฟิล์มได้ครบ 32 ระดับ โดยการทดสอบได้เกิดความผิดพลาดเนื่องจากลวดสลิงมีการแกว่งทำให้การวัดจึงมีความผิดพลาดไปจากผลจริงจึงมีผลความคลาดเคลื่อนเฉลี่ยอยู่ที่ 0.2 เซนติเมตร ความคลาดเคลื่อนที่เกิดขึ้นนี้เป็นที่ยอมรับการใช้งานได้

บทที่ 5

สรุปผลและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงผลสรุปในการออกแบบและพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน โดยเริ่มจากขั้นตอนการศึกษาวรรณกรรมเรื่องการสร้างระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน ศึกษาและค้นคว้าข้อมูลที่เกี่ยวข้อง ออกแบบและพัฒนาโครงสร้างระบบ และทดสอบประสิทธิภาพของการทำงาน วิเคราะห์ผลการทดสอบ สรุปผลการทดสอบ ปัญหาและข้อเสนอแนะ มีรายละเอียดดังนี้

5.1 สรุปผลการดำเนินโครงงาน

ปัจจุบันการพัฒนาระบบเอกซเรย์คอมพิวเตอร์ที่มีรายละเอียดของฟิล์มสูงโดยใช้ตัวรับภาพระบบฟิล์มสกรีนมีต้นทุนเชิงพาณิชย์ของระบบเอกซเรย์คอมพิวเตอร์ขนาดเล็กที่สูงและยังมีการใช้ในวงจำกัด ดังนั้นผู้จัดทำจึงต้องการศึกษา และพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับสร้างภาพตัดขวางในสัตว์ทดลองขนาดเล็กขึ้นโดยการใช้เครื่องเอกซเรย์ทั่วไป และลดการนำเข้าเพื่อการตรวจวินิจฉัยและติดตามผลการศึกษาในสัตว์ทดลองขนาดเล็ก โครงงานนี้เริ่มต้นด้วยการศึกษาหลักการ และทฤษฎีเพื่อนำการประมวลผลผ่านไมโครคอนโทรลเลอร์ไปใช้ในการออกแบบและพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน ซึ่งมีส่วนประกอบของระบบคือ สเตปปิ้งมอเตอร์ มอเตอร์ไฟฟ้ากระแสตรง เอนโคดเดอร์ และจอแสดงผลแอลซีดี สำหรับขั้นตอนการดำเนินโครงงานได้เริ่มจากทบทวนวรรณกรรม ศึกษาข้อมูลของไมโครคอนโทรลเลอร์ และศึกษาข้อมูลที่เกี่ยวข้องกับการสร้างระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน ต่อมาทำการออกแบบควบคุมการทำงานทั้งระบบด้วยการประมวลผลของไมโครคอนโทรลเลอร์ 16F886 แล้วแสดงผลออกทางจอแอลซีดี เมื่อออกแบบได้ตรงตามต้องการจึงนำอุปกรณ์การทำงานทุกส่วนมาประกอบกัน จากนั้นทดสอบประสิทธิภาพการทำงานของระบบ โดยแบ่งเป็น 2 ส่วนคือ สเตปปิ้งมอเตอร์เพื่อใช้หมุนฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กรอบแกนแนวดิ่งหมุนไปทีละ 0.9 องศาจนถึง 360 องศา และมอเตอร์ไฟฟ้ากระแสตรงเพื่อเลื่อนระดับขึ้นลงตามแนวดิ่งตั้งแต่ 1 เซนติเมตรจนถึง 32 เซนติเมตร

จากผลการทดสอบประสิทธิภาพการทำงานของสเตปปิ้งมอเตอร์ที่ฐานหุ่นจำลองสัตว์ทดลองขนาดเล็กพบว่ามีความคลาดเคลื่อนเฉลี่ย 0.02 องศา เนื่องจากการแกว่งของแกนหมุนมอเตอร์ และจากผลการทดสอบประสิทธิภาพการทำงานของมอเตอร์ไฟฟ้ากระแสตรงที่แผ่นฟิล์มพบว่ามีความคลาดเคลื่อนเฉลี่ย 0.2 เซนติเมตร เนื่องจากการแกว่างของกล่องใส่ฟิล์มสกรีน แต่อย่างไรก็ตามค่าความคลาดเคลื่อนที่เกิดขึ้นไม่มีผลต่อการใช้งานเมื่อทำการทดสอบอุปกรณ์แล้วได้ผลการฉายภาพของหุ่นสัตว์ทดลองขนาดเล็กสามารถนำไปวินิจฉัยในขั้นตอนทางการแพทย์ได้

5.2 ปัญหาและข้อเสนอแนะ

5.2.1 ปัญหาที่พบคือฐานรองรับหุ่นจำลองสัตว์ทดลองขนาดเล็กทุกรอบที่มีการหมุนไปจะพบว่ามีการแกว่งเล็กน้อยทำให้มีค่าองศาที่วัดออกมาไม่เสถียร แนวทางแก้ไขควรเลือกวิธีการประกบฐานรองรับหุ่นจำลองสัตว์ทดลองขนาดเล็กไม่ให้กระแทกกับแกนมอเตอร์ และฐานต้องมีน้ำหนักเบาเพื่อที่การวัดค่าจะมีความเสถียร

5.2.2 ปัญหาที่พบคือกลไกที่ให้เลื่อนระดับแผ่นฟิล์มมีการแกว่งและกระตุกในขณะที่ดำเนินการทำให้ค่าที่วัดระยะทางไม่เสถียร ไม่ควรเลือกใช้กลไกที่ใช้วิธีการโยง แนวทางแก้ไขควรใช้รางเลื่อนติดลูกล้อจะทำให้ลดปัญหาการแกว่ง และการกระตุกและค่าที่วัดได้มีความเสถียรมากขึ้น แล้วโครงงานนี้ควรปรับปรุงโปรแกรมให้มีระบบป้องกันไฟฟ้าขัดข้องเพื่อทำการเลื่อนระดับแผ่นฟิล์มจะได้ทำงานได้อย่างต่อเนื่อง

5.2.3 ปัญหาที่พบคือระบบกลไกขับเคลื่อนแผ่นฟิล์มมีน้ำหนักมากจึงเกิดผลกระทบต่อการหมุนของมอเตอร์ไฟฟ้ากระแสตรงทำให้เกิดความล้าช้าในกระบวนการเลื่อนระดับของแผ่นฟิล์มตามแนวดิ่ง แนวทางแก้ไขควรสร้างระบบกลไกการขับเคลื่อนของมอเตอร์กระแสตรงให้มีน้ำหนักเบา เพื่อไม่เป็นภาระของมอเตอร์ไฟฟ้ากระแสตรงซึ่งน้ำหนักมากจะส่งผลให้มอเตอร์ทำงานได้ไม่เต็มประสิทธิภาพ

5.3 แนวทางการพัฒนาโครงงาน

5.3.1 ในช่วงดำเนินการถ้าเกิดเหตุไฟฟ้าขัดข้องมอเตอร์ไฟฟ้ากระแสตรงไม่สามารถทำงานต่อเนื่องได้ต้องรีเซตกลับมาจุดศูนย์เนื่องจากไม่ได้มีการกำหนดค่าความจำให้กับโปรแกรมควบคุม แนวทางทางพัฒนาคือ ควรปรับปรุงโปรแกรมให้มีระบบป้องกันไฟฟ้าขัดข้องเพื่อทำการเลื่อนระดับแผ่นฟิล์มจะได้ทำงานได้อย่างต่อเนื่อง

5.3.2 เนื่องจากอุปกรณ์นี้ใช้งานร่วมกับเครื่องเอกซเรย์คอมพิวเตอร์ฐานอุปกรณ์นี้มีขนาดที่ต่ำจึงต้องวางไว้กับพื้นที่สูง แนวทางแก้ไข ถ้าปรับปรุงโครงสร้างควรสร้างฐานให้สูง 50-100 เซนติเมตรเพื่อง่ายต่อการใช้งานร่วมกับเครื่องเอกซเรย์คอมพิวเตอร์

เอกสารอ้างอิง

[1] ผู้ช่วยศาสตราจารย์ ดร.นันทวัฒน์ อู่ดี .(2555).**โครงงานการสร้างระบบเอกซเรย์คอมพิวเตอร์สำหรับถ่ายภาพสัตว์ทดลองขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน**. วิทยานิพนธ์ วท.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก.

[2] DataSheet.(2556).**ไมโครคอนโทรลเลอร์หมายเลข PIC16F886**. สืบค้นเมื่อ 2 ธันวาคม 2558 จาก www.microchip.com/PIC16F886

[3] Thaiio.(2003).**ความรู้เบื้องต้นและหลักการทำงานสเตปปิ้งมอเตอร์**. สืบค้นเมื่อ 6 ธันวาคม 2558 จาก http://www.thaiio.com/Hardware/stepmoter.htm

[4] Wenzhou Zhengke Electromotor.(2013).**มอเตอร์กระแสตรงชนิด ZYTD-38SRZ-R**. สืบค้นเมื่อ 6 ธันวาคม. จาก http://www.zhengkemotor.com/product/product39.html

[5] Mr.Adisak Chinawong.8 มกราคม 2543.**มอเตอร์กระแสตรง (DC MOTOR)**. สืบค้นเมื่อ 6 ธันวาคม 2558. จาก http://www.adisak51.com/page21.html

[6] Dept.Computer Engineering, Chiang Mai University.(2556).**Encoders**. สืบค้นเมื่อ 7 ธันวาคม 2559. จาก https://mcu56.learninginventions.org/?page_id=256

[7] Engineers Garage Inspiring Creations By Kushagra.(2552).**LCD**. สืบค้นเมื่อ 9 ธันวาคม 2558. จาก http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet

[8] วิชาการ.คอม.(2547).**การเขียนโปรแกรมภาษา C เบื้องต้น (Introduction to C Programming)**. สืบค้นเมื่อ 15 ธันวาคม. จาก http://www.vcharkarn.com/varticle/18065

ภาคผนวก ก

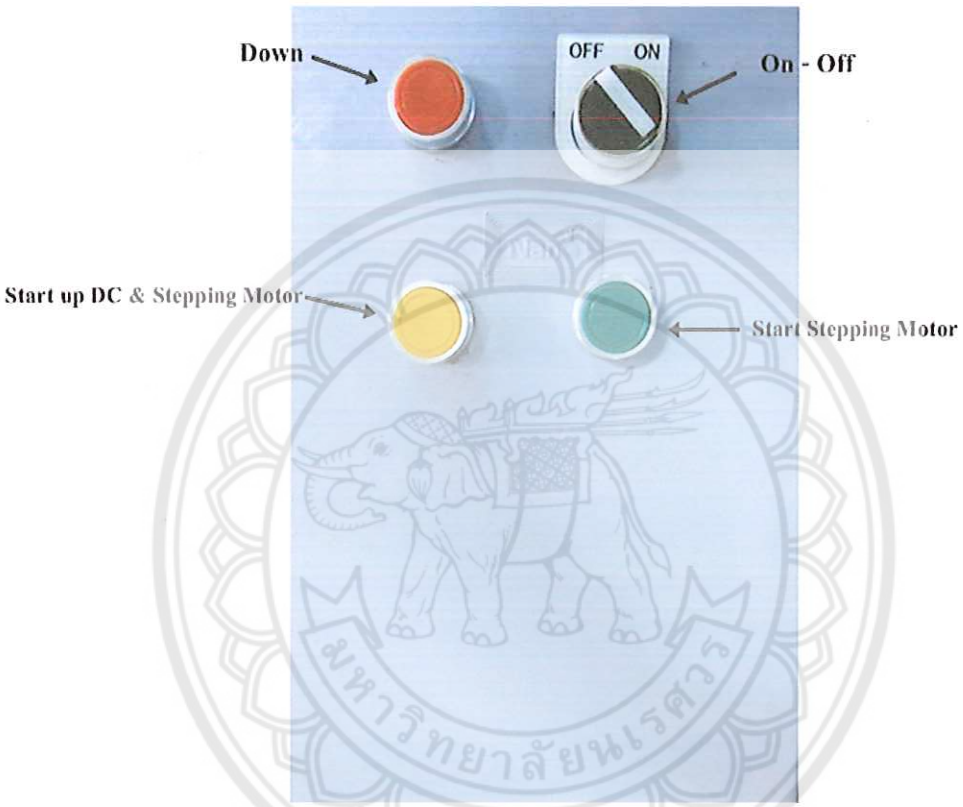
คู่มือการใช้งาน อุปกรณ์การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีน

คู่มือฉบับนี้จัดทำเพื่ออธิบายการใช้งานของอุปกรณ์การพัฒนาระบบเอกซเรย์คอมพิวเตอร์สำหรับภาพตัดขวางสัตว์ขนาดเล็กด้วยตัวรับภาพแบบฟิล์มสกรีนโดยมีรายละเอียดการใช้งานดังนี้

1.รายละเอียดของปุ่มสถานะต่างๆ

2.วิธีการใช้งานตั้งค่าและดูค่าที่อุปกรณ์ทำงาน

1.รายละเอียดของปุ่มสถานะต่างๆ

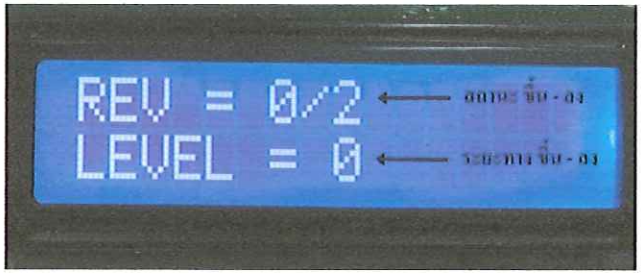


รูปที่ 1 กล่องควบคุมระบบ

- สีขาว เป็นปุ่มควบคุมการปิดเปิดอุปกรณ์พร้อมใช้งานทั้งระบบ
- สีเหลือง เป็นปุ่มควบคุมการเลื่อนระดับขึ้นลงในแนวดิ่งของมอเตอร์ไฟฟ้ากระแสตรงและหมุนตามแนวแกนดิ่งของสเตปปิ้งมอเตอร์ซึ่งทำงานพร้อมกัน
- สีเขียว เป็นปุ่มควบคุมการหมุนของสเตปปิ้งมอเตอร์เพียงตัวเดียว
- สีแดง เป็นปุ่มควบคุมการหมุนกลับของมอเตอร์ไฟฟ้ากระแสตรงในแนวดิ่ง

2.วิธีการใช้งานตั้งค่าและดูค่าที่อุปกรณ์ทำงาน

1.เปิดเครื่องรอให้เครื่องแสดงค่าพร้อมใช้งาน



รูปที่ 2 แสดงค่าพร้อมใช้งาน

- สถานะ ขึ้น แสดงเลข 2 และ ลง แสดงเลข 0
- เมื่อทำงานอุปกรณ์จะแสดงตัวเลขระยะทางที่เลื่อนแผ่นฟิล์ม

2.เลือกฟังก์ชั่นที่ต้องทำใช้งาน และเมื่อขึ้นถึงขึ้นที่ระดับ 32 ให้กดปุ่มสีแดง**เพื่อรีเซตค่าที่ 0**

ภาคผนวก ข
ผลการทดสอบสเตปปิ้งมอเตอร์ทั้งหมด 360 องศา

ตารางการหมุนองศา เริ่มต้น 0.9 องศา ถึง 360 องศาในการทดสอบ

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
0.9	0.9	0
1.8	1.8	0
2.7	2.7	0
3.6	3.6	0
4.5	4.5	0
5.4	5.4	0
6.3	6.3	0
7.2	7.2	0
8.1	8.1	0
9.0	9.0	0
9.9	9.9	0
10.8	10.8	0
11.7	11.7	0
12.6	12.6	0
13.5	13.5	0
14.4	14.4	0
15.3	15.3	0
16.2	16.2	0
17.1	17.1	0
18.0	18.0	0
18.9	18.9	0
19.8	19.8	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
20.7	20.7	0
21.6	21.6	0
22.5	22.5	0
23.4	23.4	0
24.3	24.3	0
25.2	25.2	0
26.1	26.1	0
27.0	27.0	0
27.9	27.9	0
28.8	28.8	0
29.7	29.7	0
30.6	30.6	0
31.5	31.5	0
32.4	32.4	0
33.3	33.3	0
34.2	34.2	0
35.1	35.1	0
36.0	36.0	0
36.9	36.9	0
37.8	37.8	0
38.7	38.7	0
39.6	39.6	0
40.5	40.5	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
41.4	41.4	0
42.3	42.3	0
43.2	43.2	0
44.1	44.1	0
45.0	45.0	0
45.9	45.9	0
46.8	46.8	0
47.7	47.7	0
48.6	48.6	0
49.5	49.5	0
50.4	50.4	0
51.3	51.3	0
52.2	52.2	0
53.1	53.1	0
54.0	54.0	0
54.9	54.9	0
55.8	55.8	0
56.7	56.7	0
57.6	57.6	0
58.5	58.5	0
59.4	59.4	0
60.3	60.3	0
61.2	61.2	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
62.1	62.1	0
63.0	63.0	0
63.9	63.9	0
64.8	64.8	0
65.7	65.7	0
66.6	66.6	0
67.5	67.5	0
68.4	68.4	0
69.3	69.3	0
70.2	70.2	0
71.1	71.1	0
72.0	72.0	0
72.9	72.9	0
73.8	73.8	0
74.7	74.7	0
75.6	75.6	0
76.5	76.5	0
77.4	77.4	0
78.3	78.3	0
79.2	79.2	0
80.1	80.1	0
81.0	81.0	0
81.9	81.9	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
82.8	82.8	0
83.7	83.7	0
84.6	84.6	0
85.5	85.5	0
86.4	86.4	0
87.3	87.3	0
88.2	88.2	0
89.1	89.1	0
90.0	90.0	0
90.9	90.9	0
91.8	91.8	0
92.7	92.7	0
93.6	93.6	0
94.5	94.5	0
95.4	95.4	0
96.3	96.3	0
97.2	97.2	0
98.1	98.1	0
99.0	99.0	0
99.9	99.9	0
100.8	100.8	0
101.7	101.7	0
102.6	102.6	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
103.5	103.5	0
104.4	104.4	0
105.3	105.3	0
106.2	106.2	0
107.1	107.1	0
108.0	108.0	0
108.9	108.9	0
109.8	109.8	0
110.7	110.7	0
111.6	111.6	0
112.5	112.5	0
113.4	113.4	0
114.3	114.3	0
115.2	115.2	0
116.1	116.1	0
117.0	117.0	0
117.9	117.9	0
118.8	118.8	0
119.7	119.7	0
120.6	120.6	0
121.5	121.5	0
122.4	122.4	0
123.3	123.3	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
124.2	124.2	0
125.1	125.1	0
126.0	126.0	0
126.9	126.9	0
127.8	127.8	0
128.7	128.7	0
129.6	129.6	0
130.5	130.5	0
131.4	131.4	0
132.3	132.3	0
133.2	133.2	0
134.1	134.1	0
135.0	135.0	0
135.9	135.9	0
136.8	136.8	0
137.7	137.7	0
138.6	138.6	0
139.5	139.5	0
140.4	140.4	0
141.3	141.3	0
142.2	142.2	0
143.1	143.1	0
144.0	144.0	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
144.9	144.9	0
145.8	145.8	0
146.7	146.7	0
147.6	147.6	0
148.5	148.5	0
149.4	149.4	0
150.3	150.3	0
151.2	151.2	0
152.1	152.1	0
153.0	153.0	0
153.9	153.9	0
154.8	154.8	0
155.7	155.7	0
156.6	156.6	0
157.5	157.5	0
158.4	158.4	0
159.3	159.3	0
160.2	160.2	0
161.1	161.1	0
162.0	162.0	0
162.9	162.9	0
163.8	163.8	0
164.7	164.7	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
165.6	165.6	0
166.5	166.5	0
167.4	167.4	0
168.3	168.3	0
169.2	169.2	0
170.1	170.1	0
171.0	171.0	0
171.9	171.9	0
172.8	172.8	0
173.7	173.7	0
174.6	174.6	0
175.5	175.5	0
176.4	176.4	0
177.3	177.3	0
178.2	178.2	0
179.1	179.1	0
180.0	180.0	0
180.9	180.9	0
181.8	181.8	0
182.7	182.7	0
183.6	183.6	0
184.5	184.5	0
185.4	185.4	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
186.3	186.3	0
187.2	187.2	0
188.1	188.1	0
189.0	189.0	0
189.9	189.9	0
190.8	190.8	0
191.7	191.7	0
192.6	192.6	0
193.5	193.5	0
194.4	194.4	0
195.3	195.3	0
196.2	196.2	0
197.1	197.1	0
198.0	198.0	0
198.9	198.9	0
199.8	199.8	0
200.7	200.7	0
201.6	201.6	0
202.5	202.5	0
203.4	203.4	0
204.3	204.3	0
205.2	205.2	0
206.1	206.1	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
207.0	207.0	0
207.9	207.9	0
208.8	208.8	0
209.7	209.7	0
210.6	210.6	0
211.5	211.5	0
212.4	212.4	0
213.3	213.3	0
214.2	214.2	0
215.1	215.1	0
216.0	216.0	0
216.9	216.9	0
217.8	217.8	0
218.7	218.7	0
219.6	219.6	0
220.5	220.5	0
221.4	221.4	0
222.3	222.3	0
223.2	223.2	0
224.1	224.1	0
225.0	225.0	0
225.9	225.9	0
226.8	226.8	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
227.7	227.7	0
228.6	228.6	0
229.5	229.5	0
230.4	230.4	0
231.3	231.3	0
232.2	232.2	0
233.1	233.1	0
234.0	234.0	0
234.9	234.9	0
235.8	235.8	0
236.7	236.7	0
237.6	237.6	0
238.5	238.5	0
239.4	239.4	0
240.3	240.3	0
241.2	241.2	0
242.1	242.1	0
243.0	243.0	0
243.9	243.9	0
244.8	244.8	0
245.7	245.7	0
246.6	246.6	0
247.5	247.5	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
248.4	248.4	0
249.3	249.3	0
250.2	250.2	0
251.1	251.1	0
252.0	252.0	0
252.9	252.9	0
253.8	253.8	0
254.7	254.7	0
255.6	255.6	0
256.5	256.5	0
257.4	257.4	0
258.3	258.3	0
259.2	259.2	0
260.1	260.1	0
261.0	261.0	0
261.9	261.9	0
262.8	262.8	0
263.7	263.7	0
264.6	264.6	0
265.5	265.5	0
266.4	266.4	0
267.3	267.3	0
268.2	268.2	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
269.1	269.1	0
270.0	270.0	0
270.9	270.9	0
271.8	271.8	0
272.7	272.7	0
273.6	273.6	0
274.5	274.5	0
275.4	275.4	0
276.3	276.3	0
277.2	277.2	0
278.1	278.1	0
279.0	279.0	0
279.9	279.9	0
280.8	280.8	0
281.7	281.7	0
282.6	282.6	0
283.5	283.5	0
284.4	284.4	0
285.3	285.3	0
286.2	286.2	0
287.1	287.1	0
288.0	288.0	0
288.9	288.9	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
289.8	289.8	0
290.7	290.7	0
291.6	291.6	0
292.5	292.5	0
293.4	293.4	0
294.3	294.3	0
295.2	295.2	0
296.1	296.1	0
297.0	297.0	0
297.9	297.9	0
298.8	298.8	0
299.7	299.7	0
300.6	300.6	0
301.5	301.5	0
302.4	302.4	0
303.3	303.3	0
304.2	304.2	0
305.1	305.1	0
306.0	306.0	0
306.9	306.9	0
307.8	307.8	0
308.7	308.7	0
309.6	309.6	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
310.5	310.5	0
311.4	311.4	0
312.3	312.3	0
313.2	313.2	0
314.1	314.1	0
315.0	315.0	0
315.9	315.9	0
316.8	316.8	0
317.7	317.7	0
318.6	318.6	0
319.5	319.5	0
320.4	320.4	0
321.3	321.3	0
322.2	322.2	0
323.1	323.1	0
324.0	324.0	0
324.9	324.9	0
325.8	325.8	0
326.7	326.7	0
327.6	327.6	0
328.5	328.5	0
329.4	329.4	0
330.3	330.3	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
331.2	331.2	0
332.1	332.1	0
333.0	333.0	0
333.9	333.9	0
334.8	334.8	0
335.7	335.7	0
336.6	336.6	0
337.5	337.5	0
338.4	338.4	0
339.3	339.3	0
340.2	340.2	0
341.1	341.1	0
342.0	342.0	0
342.9	342.9	0
343.8	343.8	0
344.7	344.7	0
345.6	345.6	0
346.5	346.5	0
347.4	347.4	0
348.3	348.3	0
349.2	349.2	0
350.1	350.1	0
351.0	351.0	0

องศาในการปรับการหมุนสเตปปิ้งมอเตอร์	องศาที่วัดได้จากเครื่องพัฒนาระบบเอกซเรย์คอมพิวเตอร์ฯ	ความคลาดเคลื่อน(%)
351.9	351.9	0
352.8	352.8	0
353.7	353.7	0
354.6	354.6	0
355.5	355.5	0
356.4	356.4	0
357.3	357.3	0
358.2	358.2	0
359.1	359.1	0
360.0	360.0	0

ภาคผนวก ค

รายละเอียด PIC 16F886

MICROCHIP

PIC16F882/883/884/886/887
Data Sheet

28/40/44-Pin, Enhanced Flash-Based 8-Bit CMOS Microcontrollers with nanoWatt Technology

MICROCHIP

PIC16F882/883/884/886/887

28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Only 35 instructions to learn:
  - All single-cycle instructions except branches
- Operating speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to ±1%
  - Software selectable frequency range of 8 MHz to 31 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended Temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced low-current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years
- Program memory Read/Write during run time
- In-Circuit Debugger (on board)

Low-Power Features:

- Standby Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11 µA @ 32 kHz, 2.0V, typical
  - 220 µA @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 µA @ 2.0V, typical

Peripheral Features:

- 24/35 I/O pins with individual direction control:
  - High current source/sink for direct LED drive
  - Interrupt-on-Change pin
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up (ULPWU)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (CV_{REF}) module (% of V_{DD})
  - Fixed voltage reference (0.6V)
  - Comparator inputs and outputs externally accessible
  - SR Latch mode
  - External Timer1 Gate (count enable)
- A/D Converter:
  - 10-bit resolution and 11/14 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Dedicated low-power 32 kHz oscillator
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM+ module:
  - 16-bit Capture, max. resolution 12.5 ns
  - Compare, max. resolution 200 ns
  - 10-bit PWM with 1, 2 or 4 output channels, programmable "dead time", max. frequency 20 kHz
  - PWM output steering control
- Capture, Compare, PWM module:
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Enhanced USART module:
  - Supports RS-485, RS-232, and LIN 2.0
  - Auto-Baud Detect
  - Auto-Wake-Up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I^2C™ Master and Slave Modes with I^2C address mask

PIC16F882/883/884/886/887

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	ECCP/ CCP	EUSART	MSSP	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)							
PIC16F882	2048	128	128	28	11	1/1	1	1	2	2/1
PIC16F883	4096	256	256	24	11	1/1	1	1	2	2/1
PIC16F884	4096	256	256	35	14	1/1	1	1	2	2/1
PIC16F886	8192	368	256	24	11	1/1	1	1	2	2/1
PIC16F887	8192	368	256	35	14	1/1	1	1	2	2/1

ภาคผนวก ง

โปรแกรมการทำงานที่พัฒนาขึ้นของเครื่องเอกซเรย์คอมพิวเตอร์ขนาดเล็ก สำหรับนำมาใช้กับสัตว์ทดลองขนาดเล็ก

```
**************************START PROGRAM*********************************
[HEADER]
magic_cookie={66E99B07-E706-4689-9E80-9B2582898A13}
file_version=1.0
device=PIC16F886
[PATH_INFO]
BuildDirPolicy=BuildDirIsProjectDir
dir_src=
dir_bin=output
dir_tmp=
dir_sin=
dir_inc=
dir_lib=
dir_lkr=
[CAT_FILTERS]
filter_src=*.c;*.as;*.asm;*.usb
filter_inc=*.h;
filter_obj=*.obj;*.p1;*.hex
filter_lib=*.lib;*.lpp
filter_lkr=*.unknown
[CAT_SUBFOLDERS]
subfolder_src=
subfolder_inc=
subfolder_obj=
subfolder_lib=
subfolder_lkr=
[FILE_SUBFOLDERS]
file_000=.
file_001=.
file_002=.
file_003=.
[GENERATED_FILES]
```

file_000=no

file_001=no

file_002=no

file_003=no

[OTHER_FILES]

file_000=no

file_001=no

file_002=no

file_003=no

[FILE_INFO]

file_000=main.c

file_001=lcd_4bit.c

file_002=C:\Program Files\HI-TECH Software\PICC\9.80\include\pic16f886.h

file_003=lcd_4bit.h

[SUITE_INFO]

suite_guid={507D93FD-16F1-4270-980F-0C7C0207E6D3}

suite_state=

[TOOL_SETTINGS]

TS{3FF1D5F2-E530-4850-9F70-F61D55BD1AC9}=C6=31 DB=0 DC=9 DF=0 DD=1 C2=0

C3=0 DE=1 D7=1 11E=0 121=0 122=0 123=0 124=0 125=0 11F=94FE=31 C9=2,1 EC=1 F0=0

EF=1 EE=0 104=0 E9= C4=0 F2= F3= F4= F8=1 F5= F9=0 FA=0 FB=0 C0=0 C1=0 BD=0

BC=0 BB=0 BF=0 BE=0 B8= 101=0 103= 102=0 BA= FF=0 100=0 106=0 109=0 10A=1 10B=0

10C=0 10E=0 10F=1 110=0 118=0 116=0 117= 10D=0 114=-1 113=-1 111=0 115=-1 F5=

TS{3FF1D5F2-E530-4850-9F70-F61D55BD1AC9}000=

TS{3FF1D5F2-E530-4850-9F70-F61D55BD1AC9}001=

[ACTIVE_FILE_SETTINGS]

TS{3FF1D5F2-E530-4850-9F70-F61D55BD1AC9}000_active=yes

TS{3FF1D5F2-E530-4850-9F70-F61D55BD1AC9}001_active=yes

[INSTRUMENTED_TRACE]

enable=0

transport=0

format=0

```
[CUSTOM_BUILD]
Pre-Build=
Pre-BuildEnabled=1
Post-Build=
Post-BuildEnabled=1

*******************************MAIN SETUP***********************************

#include <pic.h>
#include <stdio.h>
//#include "uart.h"
#include "lcd_4bit.h"

__EEPROM_DATA(60,1,0x06,0x0E,10,0,0,0);
//__CONFIG(HS & WDTDIS & LVPDIS & MCLRDIS);  // Config. High speed clock,Disable
watchdog and Disable LVP
__CONFIG(INTIO & WDTDIS & LVPDIS & MCLRDIS &BORDIS);

#define _XTAL_FREQ  4000000        // Define Frequency for function __delay_ms

/******* IO ********/

#define MOTOR_1_A                  RC7
#define MOTOR_1_A_TRIS                        TRISC7
#define MOTOR_1_B                  RC6
#define MOTOR_1_B_TRIS                        TRISC6

#define SW1_TRIS TRISB5
#define SW2_TRIS TRISB4
#define SW3_TRIS TRISB2
#define SW4_TRIS TRISB1
```

```
#define SW1             RB5
#define SW2             RB4
#define SW3             RB2
#define SW4             RB1

#define SW_MODE     SW4
#define SW_UP       SW3
#define SW_DOWN     SW2
#define SW_ENTER    SW1

#define STEP_1_TRIS TRISC3
#define STEP_2_TRIS TRISC4
#define STEP_3_TRIS TRISA6
#define STEP_4_TRIS TRISA7

#define STEP_1          RC3
#define STEP_2          RC4
#define STEP_3          RA6
#define STEP_4          RA7

char buff[50];

/******* TIME ********/

#define TIME_DEBOUNCE 300

//////////////////////////////////////////////////////

#define LIMIT_BUFF_RX       70
char cbuff,rxbuf,buffRx[LIMIT_BUFF_RX] ;
enum State
{
```

```
        WAIT_START_RECEIVE_DATA,
        START_RECEIVE_DATA,
        RECEIVE_DATA_FINISH,
};

////////////////////////////////////

enum State_
{
        RUNNING_STATE,
        SETTING_JOG_STATE,
        INITIAL_STATE,
        SETTING_SPEED_STATE,
        SETTING_REV_STATE,
        SETTING_COUNT_FREQ_STATE,
SETTING_STEP_STATE,
};
void Initial_IO(void);
void initial_timer1 (void);
void initial_timer0 (void);
void Delay_us(unsigned int time);

char state;
char buff_LCD[30];

char i,check_LRC,j;

void motor_1_off (void);
void motor_1_backward (void);
void motor_1_forward (void);
```

```
void clear_lcd_line1(void);
void clear_lcd_line2(void);

unsigned int analog(unsigned char channel);
void Initial_ADC(void);
void Initial_PWM(void);

////////////////////////////

int step_on = 1 ; // 1-4

void step_up(void);
void step_down(void);
void drive_step(int num);
void start_capture(void);
void fn_capture(void);
char flag_capture=0;
unsigned int freq=0;
unsigned int count_rev=0;
unsigned int count_freq=0;

unsigned int limit_rev=0;
char flag_wait = 0;
char flag_reset = 0;
unsigned int set_var=0, count_level = 0;
unsigned int limit_freq = 0,limit_step = 0;
```

```
static void interrupt isr(void)
{
if(TMR1IF)
{
                TMR1IF = 0;

                //TMR1L = 0; TMR1H = 0;
                //RA2 ^=1;


}
if(CCP2IF==1)  // Rising edge detected?
{
                CCP2CON = 0;  // Disable capture module
                CCP2IF = 0;             // Clear CCP1 interrupt flag
                CCP2IE = 0;             // Clear CCP1 interrupt flag
                TMR1ON = 0;             // Time 1 off
                freq = 125000/((CCPR2H<<8)+CCPR2L);         // Calculate frequency of
signal
                //freq = 1000000/((CCPR1H<<8)+CCPR1L);      // Calculate frequency of
signal
                flag_capture = 0;       // Clear flag indicator
        }
/*      if(RCIF)        // Ensure check RCIF flag
        {
                RCIF = 0;

                rxbuf = RCREG;
                //UART_TX(rxbuf);

                                // this algorithm is receive data only , without start and stop bit
```

```
if(rxbuf == ':')  // start bit
{
        Rx_state = START_RECEIVE_DATA;

        for(cbuff = 0 ; cbuff < LIMIT_BUFF_RX; cbuff ++)
        {
                buffRx[cbuff] = 0;
        }

        cbuff = 0;

}
else if(Rx_state == START_RECEIVE_DATA)  // start receiver to buff
{
        if(rxbuf == '!' )  //  "stop bit" is avalible
        {
                Rx_state = RECEIVE_DATA_FINISH;
        }
        else if (cbuff >= LIMIT_BUFF_RX)  // error length of frame potocol (num of data without start bit and stop bit)
        {
                Rx_state = WAIT_START_RECEIVE_DATA;
        }
        else
        {
                buffRx[cbuff] = rxbuf;    // array hex
                cbuff ++;
        }
```

```
                                    }
            }
*/

}

main()
{

            Initial_IO();
            //Initial_ADC();
            __delay_ms(100);
            Initial_LCD_port();
            InitialLCD();
            __delay_ms(100);
            GIE = 1;                    // Enable global interrupt
            PEIE=1;
            Initial_PWM();

// Timer 1 Prescaler 1:8
            T1CKPS0 = 1;
            T1CKPS1 = 1;
            TMR1CS = 0;                 // Use internal clock source
            T1SYNC = 0;                 // Synchronize external clock input

            step_on = 1;
            drive_step(step_on);

            //initial_timer1();
            //initial_timer0();
            state = INITIAL_STATE;
            //Rx_state = WAIT_START_RECEIVE_DATA;
```

```
        clear_lcd_line1(); LCDGotoXY(1,1); sprintf(buff_LCD," MOTOR  CONTROL ");
printf_LCD(buff_LCD);
        clear_lcd_line2(); //LCDGotoXY(65,1); sprintf(buff_LCD,"      V.1       ");
printf_LCD(buff_LCD);

        //motor_1_forward();

        //__delay_ms(3000);

        while(1)
        {
/*              if(Rx_state == RECEIVE_DATA_FINISH)
                {
                    // Response  =  |:|W|hr|min|day|date|month|year|!|  // first time of
initial

                    Rx_state = WAIT_START_RECEIVE_DATA;

                }
*/
                    if(state == RUNNING_STATE)
                    {

                        if(flag_wait && flag_capture==0)                    // Capture
working?
                        {
                            start_capture();              // Start new capture

                            count_freq ++;
                            if(count_freq >= limit_freq) //3100)
```

```
                {
                    count_freq = 0;
                    count_rev++;
                }

            }

            if(flag_reset)
            {
                if(count_level > 0)
                {
                    if(count_rev >= limit_rev){count_rev = 0;
step_down();  count_level--;}
                }
                else{motor_1_off(); flag_reset = 0;flag_wait = 0;}
            }
            else
            {
                if(count_rev >= limit_rev){motor_1_off(); flag_wait =
0;}
            }

            LCDGotoXY(1,1); sprintf(buff_LCD,"REV = %d/%d
",count_rev,limit_rev); printf_LCD(buff_LCD);

            LCDGotoXY(65,1); sprintf(buff_LCD,"LEVEL = %d
",count_level); printf_LCD(buff_LCD);

        if(!SW_MODE)
        {
```

```
                                        motor_1_off();
                                        flag_reset = 0; flag_wait = 0;

        __delay_ms(TIME_DEBOUNCE);while(!SW_MODE);
                                        drive_step(0);
                                        state = SETTING_SPEED_STATE;
                                        set_var = CCPR1L;
                                        clear_lcd_line1(); //LCDGotoXY(1,1);
sprintf(buff_LCD,"SET SPEED 1-100"); printf_LCD(buff_LCD);
                                        clear_lcd_line2(); //LCDGotoXY(65,1);
sprintf(buff_LCD,"SPEED : %d  ",CCPR1L); printf_LCD(buff_LCD);

                                    }
                                    else if(!SW_UP && !flag_wait && !flag_reset &&
count_level<100 ){ __delay_ms(TIME_DEBOUNCE);while(!SW_UP);     flag_wait = 1;
step_up(); motor_1_forward();count_rev = 0; count_level++;}
                                    else if(!SW_DOWN && !flag_wait && !flag_reset &&
count_level>0 ){__delay_ms(TIME_DEBOUNCE);while(!SW_DOWN); flag_wait = 1;
step_down();     motor_1_backward(); count_rev = 0; count_level--;}
                                    else if(!SW_ENTER && !flag_wait && !flag_reset &&
count_level > 0){__delay_ms(TIME_DEBOUNCE);while(!SW_ENTER);
motor_1_backward();count_rev = 0; flag_reset = 1; flag_wait = 1; }
                                    //else if(!SW_ENTER &&
!flag_wait){__delay_ms(TIME_DEBOUNCE);while(!SW_ENTER); step_down();}

                }
                else if(state == INITIAL_STATE)
                {

                    state = RUNNING_STATE;
                    clear_lcd_line1(); LCDGotoXY(1,1);
sprintf(buff_LCD,"STOP"); printf_LCD(buff_LCD);
```

```
                clear_lcd_line2(); //LCDGotoXY(65,1); sprintf(buff_LCD,"
V.1     "); printf_LCD(buff_LCD);
                motor_1_off();
                CCPR1L = eeprom_read(0); // set speed
                limit_rev = eeprom_read(1); // rev / 1cm
                limit_freq = eeprom_read(2)<< 8 | eeprom_read(3) ;
                limit_step  = eeprom_read(4); //
            }
            else if(state == SETTING_SPEED_STATE)
            {

                    if(!SW_MODE)
                    {

__delay_ms(TIME_DEBOUNCE);while(!SW_MODE);

                        state = SETTING_REV_STATE;
                        set_var = limit_rev;

                        clear_lcd_line1(); //LCDGotoXY(1,1);
sprintf(buff_LCD,"2)SET REV 1-100"); printf_LCD(buff_LCD);
                        clear_lcd_line2(); //LCDGotoXY(65,1);
sprintf(buff_LCD,"REV : %d    ",CCPR1L); printf_LCD(buff_LCD);
                        //LCDGotoXY(65,10); sprintf(buff_LCD,"%d
",set_var); printf_LCD(buff_LCD);
                    }

                    else
if(!SW_ENTER){__delay_ms(TIME_DEBOUNCE);while(!SW_ENTER); limit_rev = set_var;
eeprom_write(0,set_var);}
                    else if(!SW_UP){
__delay_ms(TIME_DEBOUNCE);while(!SW_UP);        if(set_var < 100){set_var++;} }
```

```
                                    else
if(!SW_DOWN){__delay_ms(TIME_DEBOUNCE);while(!SW_DOWN);  if(set_var >
1){set_var--;} }
                                    else{
                                    LCDGotoXY(1,1); sprintf(buff_LCD,"SET SPEED 1-
100"); printf_LCD(buff_LCD);
                                    LCDGotoXY(65,1); sprintf(buff_LCD,"SPEED : %d
",CCPR1L); printf_LCD(buff_LCD);
                                    LCDGotoXY(65,10); sprintf(buff_LCD,"%d
",set_var); printf_LCD(buff_LCD);
                                    }

                }
                else if(state == SETTING_REV_STATE) // > 1 rev for 1cm
                {

                                    if(!SW_MODE)
                                    {

        __delay_ms(TIME_DEBOUNCE);while(!SW_MODE);
                                        state = SETTING_COUNT_FREQ_STATE;

                                        set_var = limit_freq;
                                        clear_lcd_line1(); //LCDGotoXY(1,1);
sprintf(buff_LCD,"JOG MODE"); printf_LCD(buff_LCD);
                                        clear_lcd_line2(); //LCDGotoXY(65,1);
sprintf(buff_LCD,"REV : %d    ",CCPR1L); printf_LCD(buff_LCD);

                                    }
                                    else
if(!SW_ENTER){__delay_ms(TIME_DEBOUNCE);while(!SW_ENTER); limit_rev = set_var;
eeprom_write(1,set_var);}
```

```
                                    else if(!SW_UP){
__delay_ms(TIME_DEBOUNCE);while(!SW_UP);        if(set_var < 100){set_var++;} }
                                    else
if(!SW_DOWN){__delay_ms(TIME_DEBOUNCE);while(!SW_DOWN);  if(set_var >
1){set_var--;} }
                                    else{
                                    LCDGotoXY(1,1); sprintf(buff_LCD,"SET REV 1-100
cm"); printf_LCD(buff_LCD);
                                    LCDGotoXY(65,1); sprintf(buff_LCD,"REV : %d
",limit_rev); printf_LCD(buff_LCD);
                                    LCDGotoXY(65,10); sprintf(buff_LCD,"%d
",set_var); printf_LCD(buff_LCD);
                                    }

                    }
                    else if(state == SETTING_COUNT_FREQ_STATE)//  1 rev encoder
count
                    {

                                    if(!SW_MODE)
                                    {

        __delay_ms(TIME_DEBOUNCE);while(!SW_MODE);
                                            state = SETTING_STEP_STATE;
                                            clear_lcd_line1(); LCDGotoXY(1,1);
sprintf(buff_LCD,"JOG MODE"); printf_LCD(buff_LCD);
                                            clear_lcd_line2(); //LCDGotoXY(65,1);
sprintf(buff_LCD,"REV : %d    ",CCPR1L); printf_LCD(buff_LCD);
                                            set_var = limit_step;
                                    }
```

```
                                    else
if(!SW_ENTER){__delay_ms(TIME_DEBOUNCE);while(!SW_ENTER); limit_freq = set_var;
eeprom_write(2,(set_var >> 8)&0xFF);eeprom_write(3,set_var &0xFF);}
                                    else if(!SW_UP){
__delay_ms(TIME_DEBOUNCE);while(!SW_UP);        if(set_var < 10000){set_var++;} }
                                    else
if(!SW_DOWN){__delay_ms(TIME_DEBOUNCE);while(!SW_DOWN);  if(set_var >
1){set_var--;} }
                                    else{
                                    LCDGotoXY(1,1); sprintf(buff_LCD,"SET COUNT(1
REV)"); printf_LCD(buff_LCD);
                                    LCDGotoXY(65,1); sprintf(buff_LCD,"%d
",limit_freq); printf_LCD(buff_LCD);
                                    LCDGotoXY(65,8); sprintf(buff_LCD,"%d
",set_var); printf_LCD(buff_LCD);
                                    }

                }
                else if(state == SETTING_STEP_STATE) // > 1 rev for 1cm
                {

                                    if(!SW_MODE)
                                    {

        __delay_ms(TIME_DEBOUNCE);while(!SW_MODE);
                                        state = SETTING_JOG_STATE;

                                        clear_lcd_line1(); //LCDGotoXY(1,1);
sprintf(buff_LCD,"JOG MODE"); printf_LCD(buff_LCD);
                                        clear_lcd_line2(); //LCDGotoXY(65,1);
sprintf(buff_LCD,"REV : %d    ",CCPR1L); printf_LCD(buff_LCD);
```

```
                                        }
                                        else
if(!SW_ENTER){__delay_ms(TIME_DEBOUNCE);while(!SW_ENTER); limit_step = set_var;
eeprom_write(4,set_var);}
                                        else if(!SW_UP){
__delay_ms(TIME_DEBOUNCE);while(!SW_UP);        if(set_var < 200){set_var++;} }
                                        else
if(!SW_DOWN){__delay_ms(TIME_DEBOUNCE);while(!SW_DOWN);  if(set_var >
1){set_var--;} }
                                        else{
                                        LCDGotoXY(1,1); sprintf(buff_LCD,"SET STEP 1-
200"); printf_LCD(buff_LCD);
                                        LCDGotoXY(65,1); sprintf(buff_LCD,"STEP : %d
",limit_step); printf_LCD(buff_LCD);
                                        LCDGotoXY(65,10); sprintf(buff_LCD,"%d
",set_var); printf_LCD(buff_LCD);
                                        }

                    }
                    else if(state == SETTING_JOG_STATE)
                    {

                                        if(!SW_MODE)
                                        {

        __delay_ms(TIME_DEBOUNCE);while(!SW_MODE);
                                                state = RUNNING_STATE;
                                                clear_lcd_line1(); //LCDGotoXY(1,1);
sprintf(buff_LCD,"STOP"); printf_LCD(buff_LCD);
```

```
                                        clear_lcd_line2(); //LCDGotoXY(65,1);
sprintf(buff_LCD,"REV : %d   ",CCPR1L); printf_LCD(buff_LCD);
                                        drive_step(step_on);
                                    }
                                    else if(!SW_UP){
__delay_ms(TIME_DEBOUNCE);while(!SW_UP);        step_up();motor_1_forward();
__delay_ms(1000); motor_1_off();}
                                    else
if(!SW_DOWN){__delay_ms(TIME_DEBOUNCE);while(!SW_DOWN); step_down();
        motor_1_backward();  __delay_ms(1000); motor_1_off();}
                                    else
if(!SW_ENTER){__delay_ms(TIME_DEBOUNCE);while(!SW_ENTER); count_level = 0;}
                                    else{LCDGotoXY(65,1); sprintf(buff_LCD,"ENTER
TO RESET"); printf_LCD(buff_LCD);}

                }

        }
}

void motor_1_off (void)
{
        MOTOR_1_A = 0; MOTOR_1_B = 0; __delay_ms(1000);
}

void motor_1_backward (void)
{
        // go back to mag no.1
        MOTOR_1_A = 0;        MOTOR_1_B = 1;
}

void motor_1_forward (void)
```

```
{
        // go to mag no.5
        MOTOR_1_A = 1;        MOTOR_1_B = 0;
}


void Initial_IO(void)
{

        ANSEL = 0 ; // digital I/O
        ANSELH = 0 ; // digital I/O
        RBPU = 0 ; // 1 = PORTB pull-ups are disabled

        MOTOR_1_A_TRIS = 0;
        MOTOR_1_B_TRIS = 0;



        MOTOR_1_A= 0;
        MOTOR_1_B= 0;

        SW1_TRIS  = 1;
        SW2_TRIS  = 1;
        SW3_TRIS  = 1;
        SW4_TRIS  = 1;

        STEP_1_TRIS = 0;
        STEP_2_TRIS = 0;
        STEP_3_TRIS = 0;
        STEP_4_TRIS = 0;

        STEP_1 = 0;
        STEP_2 = 0;
```

```
        STEP_3 = 0;
        STEP_4 = 0;



}

void Delay_us(unsigned int time)
{
        while(time > 0)
        {
                time--;
                __delay_us(1);
        }
}

void initial_timer1 (void)
{

        // intitial setting for READ_FET capture
/*      T1CKPS0 = 1;
        T1CKPS1 = 1;
        TMR1CS = 0;             // Use internal clock source
        T1SYNC = 0;             // Synchronize external clock READ_FET
        TMR1ON = 0;             // Time 1 off
*/

        //T1CKPS0 = 1; T1CKPS1 = 1;//Prescaler 1:8
        T1CKPS0 = 0; T1CKPS1 = 0; // 15/11/56
        TMR1CS = 0;             // Use internal clock source
        T1SYNC = 0;             // Synchronize external clock READ_FET

//      TMR1H = 0x0B;           //0x0B`DB
```

```
//          TMR1L = 0xDB;
            TMR1IF  = 0;
            TMR1IE = 1;
            TMR1L = 0; TMR1H = 0;
            TMR1ON = 1;

            //65535*(1/(4e6/4/8))

            // @4MHz
            // 4MHz/4/8 = 125kHz
            // 1/250kHz = 0.000008
            // max = 8us x 65535 = 0.26214 sec

            // 0.5s /(1/(4000000/4/8)) = 62500
            // 65535-62500 = 3`035 , 0x0B`DB

}
void initial_timer0 (void)
{
            /*
            // prescale 16
            T2CKPS0 = 1;
            T2CKPS1 = 1;
            TMR2ON = 1;
            */
            // prescale 4
            T0CS = 0; //Internal instruction cycle clock (FOSC/4)
            PSA = 0; //Prescaler is assigned to the Timer0 module
            PS0 = 1;
            PS1 = 0;
            PS2 = 0;
```

```
        T0IF = 0;
        T0IE = 1;

        // 1e-3/(1/(4e6/4/4))
        // 250 clock for 1ms
        TMR0 = 5;
}
void Initial_ADC(void)
{
        //ADCS<1:0>:  A/D Conversion Clock Select bits
        ADCS1 = 1; // F RC (clock derived from a dedicated internal oscillator = 500 kHz max)
        ADCS0 = 1;

        // config port that you want
        //ANS0 = 1;

        //ANS9 = 1;
        //ANS11 = 1;
        //ANS13 = 1;

        ADFM = 0;

        //VCFG1: Voltage Reference bit
        VCFG0 = 0;  // Vss

        //VCFG0: Voltage Reference bit
        VCFG1 = 0;  // Vdd

        //ADON: ADC Enable bit
        ADON = 1 ; // ADC is enabled

}
```

```
unsigned int analog(unsigned char channel)
{
        unsigned int value = 0;
        if(channel >= 0 && channel <= 13)
        {

                //CHS<3:0>: Analog Channel Select bits
                CHS3 = (channel & 0x08) == 0x08 ;
                CHS2 = (channel & 0x04) == 0x04 ;
                CHS1 = (channel & 0x02) == 0x02 ;
                CHS0 = (channel & 0x01) == 0x01 ;

                GODONE = 1;             // Start conversion
                while(GODONE);          // Wait until conversion success
                value = (ADRESH<<2)+(ADRESL>>6);   // Get ADC value

        }
        return value ;
}
void clear_lcd_line1(void)
{
        LCDGotoXY(1,1); sprintf(buff_LCD,"               "); printf_LCD(buff_LCD);
}
void clear_lcd_line2(void)
{
        LCDGotoXY(65,1); sprintf(buff_LCD,"               "); printf_LCD(buff_LCD);
}

void fn_capture(void)
{
        if(flag_capture==0)             // Capture working?
```

```
        {
                start_capture();            // Start new capture
        }
}

void start_capture(void)
{
        CCP2IE = 0;             // Disable CCP1 interrupt
        TMR1H = 0x00;// Initial value for Timer 1
        TMR1L = 0x00;
        CCP2IF = 0;             // Clear CCP1 interrupt flag
        CCP2CON = 0x05;         // Detect rising edge
        CCP2IF = 0;             // Clear CCP1 interrupt flag
        while(CCP2IF==0);       // Wait until rising edge detected
        flag_capture = 1;       // Set flag indicator
        TMR1ON = 1;             // Time 1 off
        CCP2IF = 0;             // Clear CCP1 interrupt flag
        GIE =1;                 // Enable Global interrupt
        PEIE = 1;               // Pheripheral interrupt enable
        CCP2IE = 1;             // Enable CCP1 interrupt

}

void step_up(void)
{
int i;
        for(i=0;i<limit_step;i++)
{
        if(step_on == 4){step_on = 1;}
        else {step_on++;}
        drive_step(step_on); __delay_ms(1500);
}
```

```
}
void step_down(void)
{
int i;
        for(i=0;i<limit_step;i++)
{
        if(step_on == 1){step_on = 4;}
        else {step_on--;}
        drive_step(step_on); __delay_ms(1500);
}

}
void drive_step(int num)
{
        if(num == 1){STEP_1 = 1; STEP_2 = 0; STEP_3 = 0; STEP_4 = 0; }
        else if(num == 2){STEP_1 = 0; STEP_2 = 1; STEP_3 = 0; STEP_4 = 0; }
        else if(num == 3){STEP_1 = 0; STEP_2 = 0; STEP_3 = 1; STEP_4 = 0; }
        else if(num == 4){STEP_1 = 0; STEP_2 = 0; STEP_3 = 0; STEP_4 = 1; }
        else{STEP_1 = 0; STEP_2 = 0; STEP_3 = 0; STEP_4 = 0; }
}
void Initial_PWM(void)
{
        // Enhanced PWM Single output
//      STRA = 1;               // Use P1A(RC2 pin)generate PWM output signal
//      STRB = 1;               // Use P1B(RD5 pin)generate PWM output signal
        CCP1CON = 0x0C;         // CCP1 PWM
mode,CCP1M3:CCP1M0=1100,P1M1:P1M0=00
        PR2 = 0xFF;             // Set Period counting
        TMR2IF =0;              // Ensure Clear overflow flag timer2
        T2CON |= 0x01;// Prescaler 4
        T2CON |= 0x04;          // Start timer2
```

```
        CCP1CON |= 0x30;        // CCP1 PWM mode
        CCPR1L = 50;//100;      // Set duty for CCP1 40%

        TRISC2 = 0;
        RC2 = 0;
}

#include "lcd_4bit.h"


void Delay_ms(unsigned int x)
{ while(x!=0) x--; }

void Initial_LCD_port(void)
{
  LCD_D4_TRIS  = 0 ; // set output pin
  LCD_D5_TRIS  = 0 ; // set output pin
  LCD_D6_TRIS  = 0 ; // set output pin
  LCD_D7_TRIS  = 0 ; // set output pin

        LCD_E_TRIS = 0 ; // set output pin
        LCD_RS_TRIS = 0 ; // set output pin
        LCD_RW_TRIS = 0 ; // set output pin
        LCD_RW = 0;
}
void LCD_Epluse(void)
{
         LCD_E = 1 ;
         Delay_ms(LCD_EPLUSE_TM);
         LCD_E = 0 ;
         Delay_ms(5);            //Wait for LCD Ready
}
```

```
void WriteLCD4Bit(char DAT)
{
 LCD_D4 = ( DAT &  (1 << 4)) == (1 << 4);
 LCD_D5 = ( DAT & (1 << 5)) == (1 << 5);
 LCD_D6 = ( DAT & (1 << 6)) == (1 << 6);
 LCD_D7 = ( DAT & (1 << 7)) == (1 << 7);

 LCD_Epluse();
}
//Write 4 bit Data twice
void WriteLCDData(char DAT)
{
 LCD_RS=1;          //Data Mode
 WriteLCD4Bit(DAT);
 DAT =  DAT << 4 ;
 WriteLCD4Bit(DAT);
}
//Write 4 bit Command twice
void WriteLCDCmd(char CMD)
{
 LCD_RS = 0;          //Command Mode
 WriteLCD4Bit(CMD);
 CMD =  CMD << 4  ;
 WriteLCD4Bit(CMD);
}
void InitialLCD(void)
{
  char BITS;

//Prepare LCD Control Logic
   LCD_E = 0;
```

```
        LCD_RS =0;          //Command Mode

//Initial LCD send Initial Code 0B00110000 3 time
 WriteLCD4Bit(0B00110000);
 LCD_Epluse();
 WriteLCD4Bit(0B00110000);
 LCD_Epluse();
 WriteLCD4Bit(0B00110000);
 LCD_Epluse();
//Set as 4 bit
 WriteLCD4Bit(LCD_FUNCTION_SET|LCD4BIT_MODE);

//Set LCD
  WriteLCDCmd(LCD_FUNCTION_SET|LCD4BIT_MODE|LCD_LINE2|LCD_FONT_5X7);
  WriteLCDCmd(DISPLAY_LCD|LCD_ON|LCD_CURSOR_OFF);
  WriteLCDCmd(LCD_ENTRY_MODE|INCREMENTS_LCD|NO_SCROLL_LCD);
  WriteLCDCmd(LCD_CLEAR_SCREEN);

}

//DD-RAM command
void LCDGotoXY(char X,char Y)
{
  X--;                  //Adjust to Count number 1...n
  switch(Y)
  {
   case 1:
      WriteLCDCmd(LCD_ROW1|LCD_DDRAM_ADDRESS+X);      //0x00|0x80
      break;
   case 2:
      WriteLCDCmd(LCD_ROW2|LCD_DDRAM_ADDRESS|+X);    //64|0x80
      break;
```

```
    }
}
void printf_LCD ( char* str)
{
   char c;

   while( (c = *str++) )
      WriteLCDData(c);
}
#include<pic.h>
//Define LCD I/O PIN
//LCD Control
#define  LCD_E          RA4
#define  LCD_E_TRIS     TRISA4

#define  LCD_RW                    RA5
#define  LCD_RW_TRIS               TRISA5

#define  LCD_RS         RC0
#define  LCD_RS_TRIS    TRISC0

//LCD Data
#define  LCD_D4         RA3
#define  LCD_D5         RA2
#define  LCD_D6         RA1
#define  LCD_D7         RA0
#define  LCD_D4_TRIS        TRISA3
#define  LCD_D5_TRIS        TRISA2
#define  LCD_D6_TRIS        TRISA1
#define  LCD_D7_TRIS        TRISA0

#define  LCD_ROW1       0x00
```

```
#define LCD_ROW2        0x40
#define LCD_ROW3        0x14
#define LCD_ROW4        0x54

#define LCD_EPLUSE_TM 20

//*** START LCD Command ***
#define LCD_CLEAR_SCREEN  0B00000001
#define LCD_RETURN_HOME   0B00000010

//Entry mode set (BIT2=1)
#define LCD_ENTRY_MODE    0B00000100
#define INCREMENTS_LCD    0B00000010
#define DECREMENTS_LCD    0B00000000
#define NO_SCROLL_LCD     0B00000000
#define SCROLL_LCD        0B00000001

//Display ON/OFF (BIT3=1)
#define DISPLAY_LCD       0B00001000
#define LCD_ON            0B00000100
#define LCD_OFF           0B00000000
#define LCD_CURSOR_ON     0B00000010
#define LCD_CURSOR_OFF    0B00000000
#define LCD_BLINK_ON      0B00000001
#define LCD_BLINK_OFF     0B00000000

//Scroll Display/Shift Cursor (BIT4=1)
#define LCD_SCROLL_SHIFT  0B00010000
#define LCD_SHIFT_CURSOR  0B00000000
#define LCD_SHIFT_DISPLAY 0B00001000
#define LCD_SHIFT_LEFT    0B00000000
#define LCD_SHIFT_RIGHT   0B00000100
```

```
//Function set (BIT5=1)
#define  LCD_FUNCTION_SET   0B00100000
#define  LCD4BIT_MODE       0B00000000
#define  LCD8BIT_MODE       0B00010000
#define  LCD_LINE1          0B00000000
#define  LCD_LINE2          0B00001000
#define  LCD_FONT_5X7       0B00000000
#define  LCD_FONT_5X10      0B00000100

//Move To CGRAM Address (BIT6=1)
#define  LCD_CGRAM_ADDRESS  0B01000000

//Move To DDRAM Address (BIT7=1)
#define  LCD_DDRAM_ADDRESS  0B10000000

//*** END LCD Command ***

void Delay_ms(unsigned int x);
void Initial_LCD_port(void);
void LCD_Epluse(void);
void WriteLCDCmd(char CMD);
void WriteLCDData(char c);
void InitialLCD(void);
void LCDGotoXY(char  X,char  Y);
void LCDGotoXY(char  X,char  Y);
void printf_LCD ( char* str);

********************************** END PROGRAM ***************************
```

ประวัติผู้ดำเนินโครงงาน

ชื่อ นางสาวพรเพ็ญ บุญแสวง

ภูมิลำเนา 99/1 หมู่ 10 ต.พรหมพิราม อ.พรหมพิราม จ.พิษณุโลก

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพรหมพิรามวิทยา
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: ponpen.ee@gmail.com