โปรแกรมออกแบบและจำลองการแปรอักษร 3มิติ

Program for Design and Simulation

of Three-Dimensional Character

นางสาวกันติยาภรณ์ ขวัญพรม รหัส 53363324

นางสาวนลิน อุตสาหฉันท์ รหัส 53363553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2556

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 30 / มี.ค. / 57
เลขทะเบียน 1655845
เลขเรียกหนังสือ วร.
มหาวิทยาลัยนเรศวร ว389 ป 2556

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	**โปรแกรมออกแบบและจำลองการแปรอักษร 3 มิติ**		
ผู้ดำเนินโครงงาน	**นางสาวกันติยากรณ์**	**ขวัญพรม**	**รหัส 53363324**
	นางสาวนลิน	อุตสาหฉันท์	**รหัส 53363553**
ที่ปรึกษาโครงงาน	ดร.สุวิทย์ กิระวิทยา		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2556		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์**ฉบับนี้เป็นส่วนหนึ่ง**ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอม**พิวเตอร์**

Suwit Kiravittaya.....**ที่ปรึกษาโครงงาน**

(ดร. สุวิทย์ กิระวิทยา)

.....**กรรมการ**

(อ. รัฐภูมิ วรานุสาสน์)

เศรษฐา ตั้งค้าวานิช.....**กรรมการ**

(อ. เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	โปรแกรมออกแบบและจำลองการแปรอักษร 3 มิติ		
ผู้ดำเนินโครงงาน	นางสาวกันติยาภรณ์	ขวัญพรม	รหัส 53363324
	นางสาวนลิน	อุตสาหฉันท์	รหัส 53363553
ที่ปรึกษาโครงงาน	ดร.สุวิทย์ กิระวิทยา		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2556		

.....

บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อศึกษาและนำเสนอการแสดงผลในแบบ 3 มิติ โดยการพัฒนาโปรแกรมคอมพิวเตอร์เพื่อจำลองการแสดงผล 3 มิติ จากแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ เป็นการพัฒนาโปรแกรมการออกแบบและจำลองการแปรอักษร 3 มิติ เพื่อผู้ที่สนใจในเรื่องการแสดงผล 3 มิติ สามารถออกแบบและจำลองการแสดงผลด้วยไดโอดเปล่งแสงทรงลูกบาศก์ในคอมพิวเตอร์ด้วยโปรแกรมที่ผู้ศึกษาพัฒนาขึ้นได้ ทั้งนี้ตัวอักษรหรือข้อความยังสามารถประมวลผลออกมาเป็นสีต่างๆ ได้ตามที่เราต้องการได้อีกด้วย

Project title	Program for Design and Simulation of Three-Dimensional Character		
Name	Miss. Gantiyapron	Khwanprom	ID. 53363324
	Miss. Nalin	Utsahachant	ID. 53363553
Project advisor	Dr. Suwit Kiravittaya		
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic year	2013		

.....

Abstract

This project presents a development of three-dimensional (3D) display by using a computer program to simulate the display on an LED cube. People who are interested in the 3D display can use this program to design and simulate with virtual LED cube in the program. Both 3D Thai and English characters can be displayed in this program. It can be used to design and simulate various display method. Texts can be processed into different colors.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์เรื่องโปรแกรมออกแบบและจำลองการแปรอักษร 3 มิติ ฉบับนี้สำเร็จลุล่วงได้ เนื่องมาจากทางคณะผู้จัดทำได้รับความอนุเคราะห์จาก ดร. สุวิทย์ กิระวิทยา ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานนี้ ที่กรุณาให้คำปรึกษา แนะนำวิธีการทำงานพร้อมทั้งแนะแนวทางในการแก้ไขปัญหาต่างๆ และคอยกระตุ้นให้คณะผู้จัดทำทำงานอย่างต่อเนื่อง ตลอดจนสละเวลาอันมีค่าเพื่อตรวจสอบและแก้ไขข้อบกพร่องต่างๆ คณะผู้จัดทำรู้สึกเป็นเกียรติอย่างมากที่ได้รับความอนุเคราะห์จากอาจารย์

ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบพระคุณอาจารย์ทุกท่าน บิดา มารดา ที่คอยสั่งสอนให้ความรู้จนคณะผู้จัดทำสำเร็จการศึกษา ทั้งนี้ขอขอบคุณเพื่อนๆ ที่ได้ช่วยในการทดสอบโปรแกรมและคอยให้กำลังใจ ช่วยให้คำปรึกษาทั้งในเรื่องเรียนและในเรื่องส่วนตัวจนสำเร็จลุล่วงมาด้วยดี

นางสาวกันติยาภรณ์ ขวัญพรม
นางสาวนลิน อุตสาหฉันท์

สารบัญ

หน้า

ใบรับรองโครงงานวิจัย.....ก
บทคัดย่อภาษาไทย.....ข
บทคัดย่อภาษาอังกฤษ.....ค
กิตติกรรมประกาศ.....ง
สารบัญ.....จ
สารบัญตาราง.....ช
สารบัญรูป.....ซ

บทที่ 1 บทนำ
1.1 ที่มาและความสำคัญของโครงงาน.....1
1.2 วัตถุประสงค์ของโครงงาน.....1
1.3 ขอบข่ายของโครงงาน.....2
1.4 ขั้นตอนของการดำเนินงานและแผนการดำเนินงาน.....3
1.5 ผลที่คาดว่าจะได้รับจากโครงงาน.....3
1.6 งบประมาณ.....3

บทที่ 2 หลักการและทฤษฎี
2.1 แมตแล็บ.....4
2.2 ทฤษฎีสี.....8
2.3 ตัวอักษร.....9
2.4 หลักการพื้นฐานของ Light Emitting Diode Cube (LED Cube).....11

บทที่ 3 วิธีการดำเนินโครงงาน
3.1 อุปกรณ์ที่ใช้.....13
3.2 วิธีการดำเนินโครงงาน.....14

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการดำเนินโครงงาน

4.1 การแสดงผลแบบกระพริบ (Flashing).....32

4.2 การแสดงผลแบบเลื่อนในแนวนอน (Horizontal).....35

4.3 การแสดงผลแบบเลื่อนในแนวตั้ง (Vertical).....39

บทที่ 5 บทสรุป

5.1 บทสรุปการดำเนินงาน.....43

5.2 ปัญหาที่พบ.....44

5.3 ข้อเสนอแนะ.....44

5.4 แนวทางในการประยุกต์และพัฒนาต่อไป.....45

เอกสารอ้างอิง.....46

ภาคผนวก ก.....47

ภาคผนวก ข.....53

ประวัติผู้ดำเนินงาน.....57

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงาน.....2

5.1 การเปรียบเทียบระหว่างแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของจริง(ฮาร์ดแวร์) และแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของโปรแกรม (ซอฟต์แวร์).....44

สารบัญรูปภาพ

รูปที่ หน้า

2.1 หน้าต่างการแก้ไขเค้าโครง GUIDE....6
2.2 ตัวอย่างการเขียนโปรแกรมประยุกต์ M-file....7
2.3 แสดงระบบสี RGB....8
2.4 แสดง RGB Model....8
2.5 ลักษณะตัวอักษรแบบ Block....9
2.6 ส่วนประกอบต่างๆของตัวอักษร....10
2.7 ระยะต่างๆ ในการออกแบบตัวอักษร....10
2.8 LED Cube 16x16x16....12
3.1 ตัวอย่างการสร้างตัวอักษร 2 มิติ....14
3.2 ตัวอย่างการใช้คำสั่ง Unicode ของแมตแล็บ (MATLAB)....16
3.3 แผนภาพการทำงานของการตรวจสอบสระในภาษาไทย....18
3.4 แผนภาพแสดงการทำงานของการตรวจสอบชนิดตัวอักษรของผู้ใช้....20
3.5 แผนภาพแสดงการทำงานของการแปลงตัวอักษร 2 มิติ เป็นตัวอักษร 3 มิติ....22
3.6 แผนภาพแสดงการทำงานแบบ Flashing....23
3.7 แผนภาพแสดงการทำงานแบบ Horizontal....24
3.8 แผนภาพแสดงการทำงานแบบ Vertical....25
3.9 ตัวอย่างการแสดงผลการแปลงอักษร 2 มิติ เป็นอักษร 3 มิติ....26
3.10 ตัวอย่างการแสดงผลของสี....28
3.11 ตัวอย่างการแสดงผลความหนาในแต่ละ bits....29
3.12 GUI การแสดงผลของโปรแกรม....31
4.1 แสดงชุด binary ของตัวอักษร 2 มิติ แบบ Flashing....32
4.2 แสดงการตัดชุด binary ของชุดตัวอักษร 2 มิติ ออกมาทีละ 12 bits....33
4.3 แสดงตัวอักษร 3 มิติ แบบกระพริบ (Flashing)....34

สารบัญรูปภาพ(ต่อ)

รูปที่ หน้า

4.4 แสดงชุด binary ของตัวอักษร 2 มิติ แบบ Horizontal....36

4.5 แสดงชุด binary ของชุดตัวอักษร 2 มิติ ที่ทำการ shift bit ไปทางด้านซ้าย 10 bits....36

4.6 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวนอน (Horizontal)....37

4.7 แสดงชุด binary ของตัวอักษร 2 มิติ แบบ Vertical....40

4.8 แสดงชุด binary ของตัวอักษร 2 มิติ ที่ทำการ shift bit ไปทางด้านบน 10 bits....41

4.9 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวตั้ง (Vertical)....41

ผ.1 ตัวอย่าง code การตรวจสอบสระ....48

ผ.2 ตัวอย่าง code การตรวจสอบชนิดของตัวอักษร....48

ผ.3 ตัวอย่าง code การนำพยัญชนะและสระในภาษาไทยมารวมกัน....50

ผ.4 ตัวอย่าง code การแปลงตัวอักษร 2 มิติเป็นตัวอักษร 3 มิติ....51

ผ.5 ตัวอย่าง code สี RGB, CMYK และสีผสม....51

ผ.6 ตัวอย่าง code การเรียกใช้สี....52

ผ.7 ตัวอย่าง code การเลือกความหนาของตัวอักษร....52

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันเทคโนโลยีที่ใช้ในการแสดงผลตัวอักษรหรือข้อความในลักษณะการแสดงผลแบบ 3 มิติ ได้รับความนิยมอย่างแพร่หลาย ทั้งในลักษณะการแสดงผลตัวอักษรแบบธรรมดาหรือการแสดงผลอักษรแบบเคลื่อนไหวได้ ดังนั้นทางกลุ่มผู้ศึกษาจึงต้องการพัฒนาการแสดงผลอักษรในรูปแบบ 3 มิติ ในทางด้านซอฟแวร์ ซึ่งตัวโปรแกรมสามารถออกแบบและจำลองการแปรอักษรในรูปแบบ 3 มิติ ตามความสนใจของผู้ศึกษา เพื่อที่ผู้ศึกษาจะได้รับความรู้และความเข้าใจในด้านการแสดงผล 3 มิติ, ความแตกต่างในการเก็บข้อมูลที่เป็นอาร์เรย์ 2 มิติ และ 3 มิติ และการใช้เทคนิคทางด้านความหลากหลายของแสงสี RGB ประมวลผลอักษรได้ชัดเจนยิ่งขึ้นและยังสามารถมองได้หลากหลายมุมมองอีกด้วย

โครงการที่ทำการศึกษานี้จะเป็นการพัฒนาโปรแกรมการออกแบบและจำลองการแปรอักษร 3 มิติ เพื่อผู้ที่สนใจในเรื่องการแสดงผล 3 มิติ สามารถออกแบบและจำลองการแสดงผลด้วย Light Emitting Diode Cube (LED Cube) ในคอมพิวเตอร์ด้วยโปรแกรมที่ผู้ศึกษาพัฒนาขึ้นได้ ทั้งนี้ตัวอักษรหรือข้อความยังสามารถประมวลผลออกมาเป็นสีต่างๆได้ตามที่เราต้องการ และยังสามารถนำไปประยุกต์ใช้กับการแสดงผลแบบฮาร์ดแวร์ได้อีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อจำลองการออกแบบตัวอักษร 3 มิติ เพื่อจำลองการออกแบบตัวอักษร 3 มิติ ผ่านโปรแกรมจำลองไดโอดเปล่งแสงทรงลูกบาศก์
2. เพื่อศึกษาและออกแบบโปรแกรมจำลองอักษร 3มิติ จาก library ของแมตแล็บ (MATLAB)
3. เพื่อศึกษาและทำความเข้าใจการประมวลผลภาพ 3 มิติ

1.3 ขอบข่ายของโครงงาน

1. ภาพ 3 มิติ ที่แสดงใช้ระดับความเข้มสีในระบบ RGB ที่แตกต่างกันประมวลผลออกมาเป็นสีต่างๆ
2. โปรแกรมที่พัฒนามีอินเตอร์เฟสเป็น Graphical User Interface (GUI)
3. เทคนิคในการแสดงผลตัวอักษรสามารถกระพริบได้ และเคลื่อนที่ได้ทั้งแนวตั้งและแนวนอน
4. ลักษณะของตัวอักษรสามารถแสดงผลได้ทั้งตัวเลขอารบิก ตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ ตัวอักษรภาษาอังกฤษตัวพิมพ์เล็ก ตัวเลขไทย ตัวพยัญชนะ สระ และวรรณยุกต์ภาษาไทย
5. การเพิ่ม-ลด ความหนาของตัวอักษร

1.4 ขั้นตอนของการดำเนินงานและแผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	ระยะเวลาการดำเนินงาน ปีการศึกษา 2556								
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาและทำความเข้าใจการแสดงภาพ 3 มิติ และคำสั่งใน Library ของแมตแล็บ (MATLAB) ที่ใช้แสดงผลภาพ 3 มิติ	■	■	■						
2. พัฒนาโปรแกรมออกแบบและจำลองการแปรอักษร 3 มิติ โดยใช้คำสั่งใน Library ของแมตแล็บ (MATLAB)				■	■				
3. ตรวจสอบความถูกต้องและ debug โปรแกรมออกแบบและจำลองการแปรอักษร 3 มิติ						■	■		

ตารางที่ 1.1 แผนการดำเนินงาน (ต่อ)

กิจกรรม	ระยะเวลาการดำเนินงาน ปีการศึกษา 2556								
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
4. แก้ไขโปรแกรมและตรวจสอบโปรแกรมอีกครั้ง								■	
5. สรุปผลการดำเนินงานและจัดทำปริญญานิพนธ์ฉบับสมบูรณ์									■

1.5 ผลที่คาดว่าจะได้รับจากโครงงาน

1. สามารถนำโปรแกรมที่ได้ไปประยุกต์ใช้กับไดโอดเปล่งแสงทรงลูกบาศก์
2. ได้รับความรู้เกี่ยวกับการประมวลผลและแสดงผลภาพ 3 มิติ
3. โปรแกรมที่พัฒนาขึ้นมีประโยชน์ต่อผู้ที่สนใจพัฒนาต่อ
4. ได้รับความรู้เกี่ยวกับการเก็บข้อมูลอาร์เรย์ 3 มิติ

1.6 งบประมาณ

1. ค่าวัสดุสำนักงาน	เป็นเงิน	500 บาท
2. ค่าถ่ายเอกสารและเข้าเล่มรายงานฉบับสมบูรณ์	เป็นเงิน	1500 บาท
	รวมเป็นเงินทั้งสิ้น	2,000 บาท
		(สองพันบาทถ้วน)

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่ 2

หลักการและทฤษฎี

เนื้อหาในบทนี้ได้แบ่งรายละเอียดที่เกี่ยวข้องกับหลักการและทฤษฎีที่ใช้ในการพัฒนาโครงงานออกเป็นส่วนต่างๆ หลายส่วน ได้แก่ เรื่องของการใช้งานแมตแล็บ (MATLAB) เรื่องทฤษฎีสี และทฤษฎีที่เกี่ยวข้องกับการพัฒนาโปรแกรม เป็นต้น เพื่อใช้เป็นแนวทางในการพัฒนาโครงงาน ซึ่งมีรายละเอียดดังนี้

2.1 แมตแล็บ (MATLAB)

แมตแล็บ (MATLAB) เป็นซอฟต์แวร์ในการคำนวณและการเขียนโปรแกรมโปรแกรมหนึ่ง ที่มีความสามารถครอบคลุมตั้งแต่ การพัฒนาอัลกอริธึม การสร้างแบบจำลองทางคณิตศาสตร์ กระบวนการออกแบบแบบจำลอง (Simulation) ของระบบ และการสร้างระบบควบคุม

2.1.1 อาร์เรย์ (Array)

อาร์เรย์ คือกลุ่มของข้อมูลชนิดเดียวกัน ข้อมูลแต่ละตัวในอาร์เรย์จะเรียกว่า "สมาชิกอาร์เรย์" อาร์เรย์แต่ละอาร์เรย์จะมีสมาชิกกี่ตัวก็ได้และจำนวนสมาชิกของอาร์เรย์จะมีเท่ากับจำนวนข้อมูลที่เก็บในอาร์เรย์เสมอ

2.1.1.1 อาร์เรย์หลายมิติ

การกำหนดอาร์เรย์หลายมิติทำได้โดยการใช้เครื่องหมายจุลภาค (Comma หรือ ,) คั่นระหว่างตำแหน่งในอาร์เรย์

อาร์เรย์ 3 มิติ สามารถเขียนได้ว่า A(m, n, k) หมายถึง การอ้างอิงตำแหน่งข้อมูลที่ m ในมิติที่ 1 ตำแหน่งที่ n ในมิติที่ 2 ตำแหน่งที่ k ในมิติที่ 3

2.1.2 คำสั่งที่ในการเขียนโปรแกรม

unicode2native

คือคำสั่งที่ใช้แปลงจากตัวอักษรภาษไทยไปเป็นรหัส ASCII (0-255) เพื่อจัดเก็บหรือใช้งานฟังก์ชันสตริง จะใช้คำสั่ง unicode2native ในการแก้ไขปัญหาการใช้ภาษาไทยในแมตแล็บ

fliplr (X)

คือคำสั่งที่ใช้ในการกลับตำแหน่งในแนวนอน

flipud (X)

คือคำสั่งที่ใช้ในการกลับตำแหน่งในแนวตั้ง

scatter3 (X,Y,Z,S,C)

คือคำสั่งที่ใช้ในการแสดงสีวงกลมในสถานที่ที่ระบุโดยเวกเตอร์ X, Y, และ Z

S ใช้กำหนดขนาดของตำแหน่ง (ที่ระบุไว้ในจุด)

C ใช้กำหนดสีของแต่ละตำแหน่ง

circshift (A,SHIFTSIZE)

คือคำสั่งที่ใช้หมุนเลื่อนแถวหรือหลัก

- ระบุจำนวนครั้งในการเลื่อนขึ้นลงในมิติที่ 1 โดยขึ้นเป็น – ลงเป็น +
- ระบุจำนวนครั้งในการเลื่อนซ้ายขวาในมิติที่ 2 โดยซ้ายเป็น – ขวาเป็น +

linspace (X1, X2, N)

คือคำสั่งที่ใช้สร้าง row vector ขนาด 1xN โดยค่าแรกจะเท่ากับ X1 **และค่าสุดท้ายเท่ากับ** X2

2.1.3 การใช้ภาพเป็นตัวประสานกับผู้ใช้ (Graphical User Interface หรือ GUI)

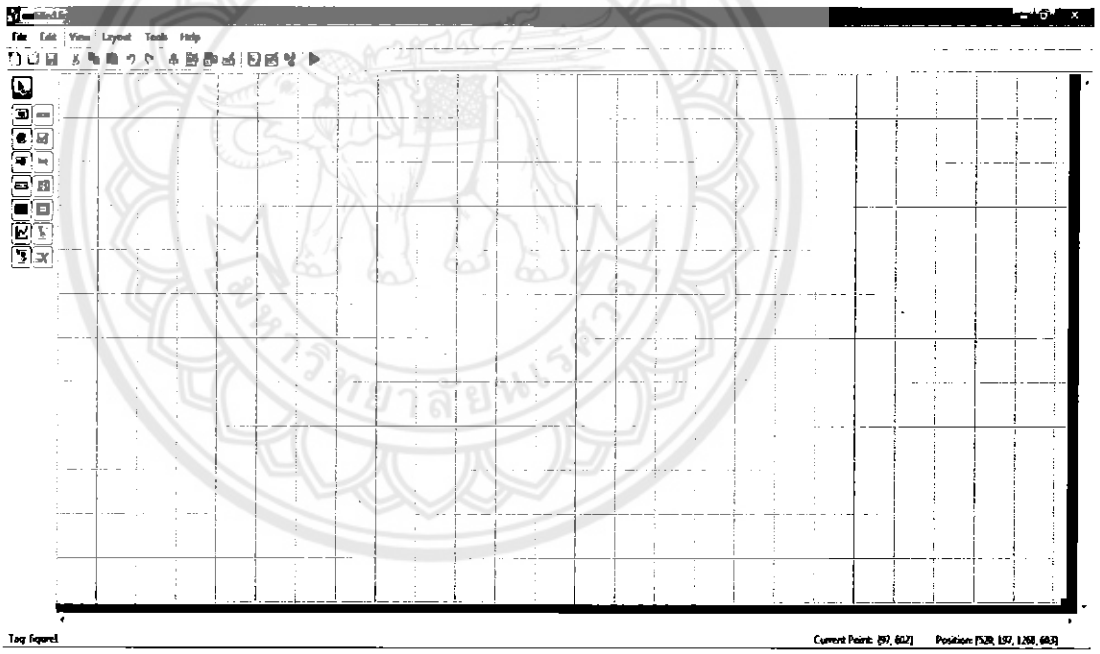
การใช้ภาพเป็นตัวประสานกับผู้ใช้ (Graphical User Interface หรือ GUI) เป็นส่วนติดต่อผู้ใช้ที่สร้างขึ้นด้วยวัตถุของกราฟิกแบบต่างๆ เช่น ไอคอน หน้าต่างการใช้งาน เมนู ปุ่มเลือก และการใช้เมาส์ โปรแกรมประยุกต์ต่างๆที่ออกแบบมาเป็นการใช้ภาพเป็นตัวประสานกับผู้ใช้ (Graphical User Interface หรือ GUI) พบว่าสามารถทำให้ผู้ใช้เข้าใจการใช้โปรแกรมประยุกต์นั้นได้อย่างรวดเร็ว และการทำงานของโปรแกรมประยุกต์จะเกิดขึ้นทันทีที่ได้รับค่าอินพุทจากผู้ใช้

2.1.2.1 การใช้ภาพเป็นตัวประสานกับผู้ใช้ (GUI) ด้วย GUIDE

แมตแล็บ (MATLAB) จะสร้างภาพที่เป็นตัวประสานกับผู้ใช้ (GUI) อยู่บนหน้าต่างรูปภาพ ซึ่งภายในหน้าต่างนี้จะมีส่วนประกอบต่างๆ อยู่ ไม่ว่าจะเป็น Axes, Uicontrol หรือวัตถุอื่นๆ และ แมตแล็บ (MATLAB) ได้สร้าง Graphical User Interface Development Environment หรือ GUIDE ขึ้นเพื่อช่วยให้เราสร้าง บันทึก และแก้ไขภาพเป็นตัวประสานกับผู้ใช้ (GUI) ได้สะดวกขึ้น การใช้ภาพเป็นตัวประสานกับผู้ใช้ (GUI) จะประกอบด้วยสองขั้นตอน ได้แก่

1. กำหนดและวางส่วนประกอบต่างๆ ลงบนการใช้ ภาพเป็นตัวประสานกับผู้ใช้ (GUI)
2. เขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนประกอบต่างๆ ใน การใช้ภาพเป็นตัวประสานกับผู้ใช้ (GUI)

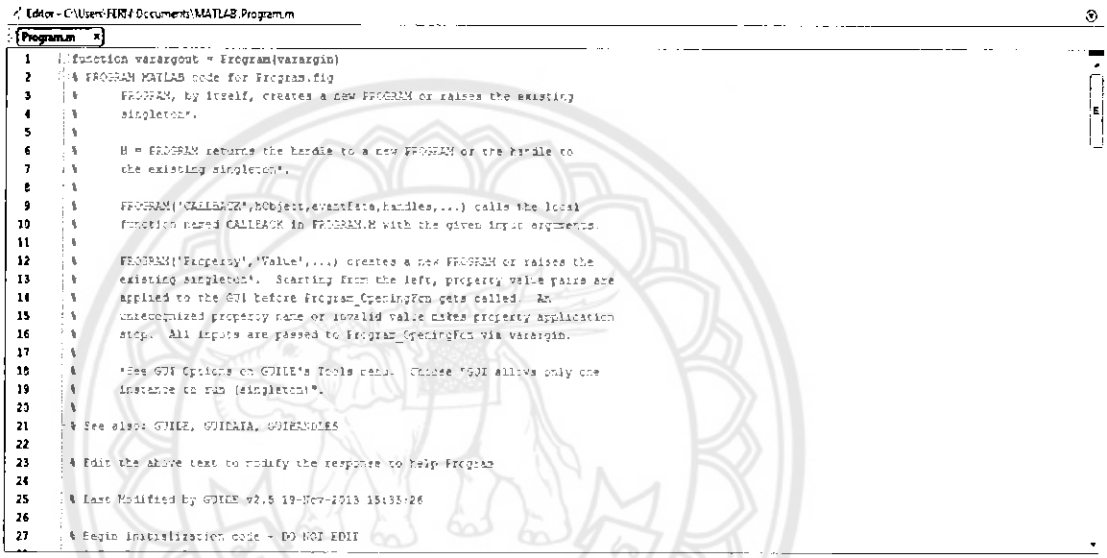
ดังภาพที่ 2.1



ภาพที่ 2.1 หน้าต่างการแก้ไขเค้าโครง GUIDE

2.1.2.2 โปรแกรมประยุกต์ M-file ที่สร้างโดย GUIDE

โปรแกรมประยุกต์ M-file เป็นโครงร่างโปรแกรมที่ใช้ควบคุมการทำงานของภาพซึ่งเป็นตัวประสานกับผู้ใช้ (GUI) ซึ่งจะถูกสร้างขึ้นโดยอัตโนมัติพร้อมกับ Fig-file เมื่อใช้ GUIDE ในการสร้าง โดยที่โปรแกรมประยุกต์ M-file จะช่วยให้มีความสะดวกและรวดเร็วมากขึ้น ซึ่งโปรแกรมหรือคำสั่งทุกส่วนรวมถึงฟังก์ชั่นรอง จะรวมอยู่ในโปรแกรมประยุกต์ M-file ดังภาพที่ 2.2



ภาพที่ 2.2 ตัวอย่างการเขียนโปรแกรมประยุกต์ M-file

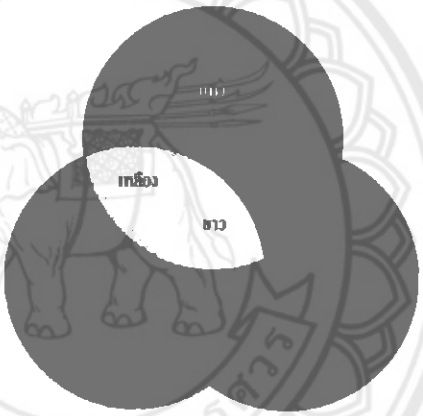
เพื่อกำหนดการทำงานของส่วนประกอบต่างๆ

2.2 ทฤษฎีสี

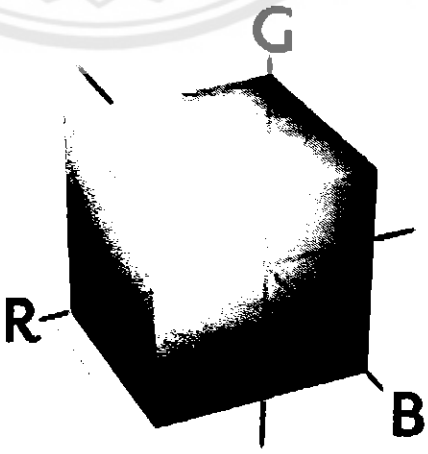
2.2.1 ระบบสี RGB

ระบบสีแบบ RGB คือระบบสีพื้นฐานที่ใช้ในการแสดงผลภาพสีทางคอมพิวเตอร์ และยังเป็นสีพื้นฐานที่ใช้ในการเก็บภาพแบบดิจิตอล ซึ่งระบบสีแบบ RGB นี้ประกอบด้วยสีแดง (R) สีเขียว (G) และสีน้ำเงิน (B) โดยทั่วไปแล้วในแต่ละสีจะถูกควบคุมด้วยจำนวนบิต 8 บิต โดยมีค่าต่ำสุดของสีคือ 0 (00 ในฐานสิบหก) และค่าสูงสุดคือ 255 (FF ในฐานสิบหก)

ภาพสีแบบ RGB ในแต่ละจุดจะถูกควบคุมด้วยจำนวนบิตทั้งหมด 24 บิต โดยมีค่าเริ่มต้นที่ 000000 – FFFFFF ในเลขฐานสิบหก โดยบิตที่ 0-7 ใช้ควบคุมระดับสีแดง บิตที่ 8-15 ใช้ควบคุมระดับสีเขียว และบิตที่ 16-23 ใช้ในการควบคุมระดับสีน้ำเงิน ดังภาพที่ 2.3 และภาพที่ 2.4



ภาพที่ 2.3 แสดงระบบสี RGB[1]



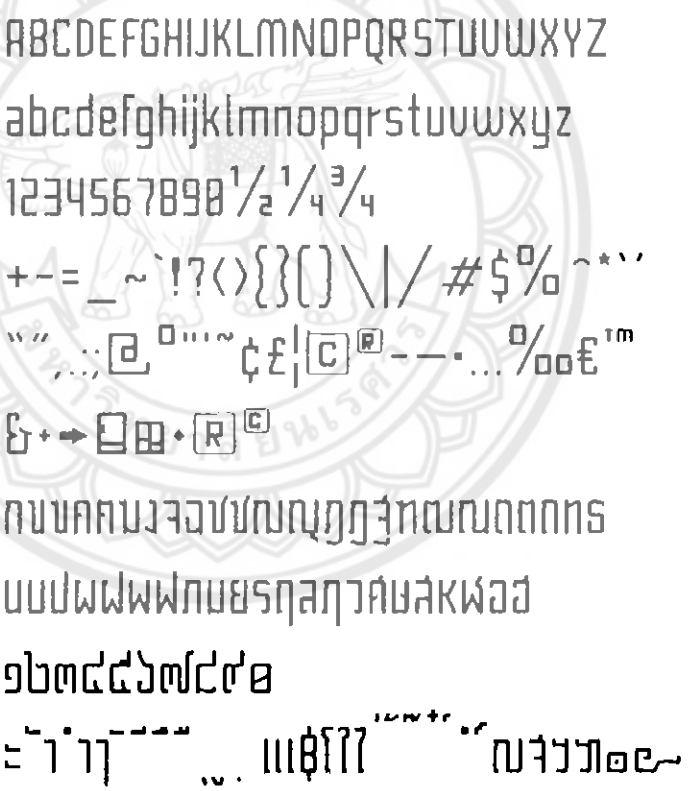
ภาพที่ 2.4 แสดง RGB Model[2]

2.3 ตัวอักษร

ตัวอักษร เรียกว่า อักขระ (character) ตัวอักษรที่มีลักษณะเหมือนกันเป็นชุดเดียวกัน เรียกว่า ฟอนต์ (Font)

2.3.1 แบบตัวอักษร

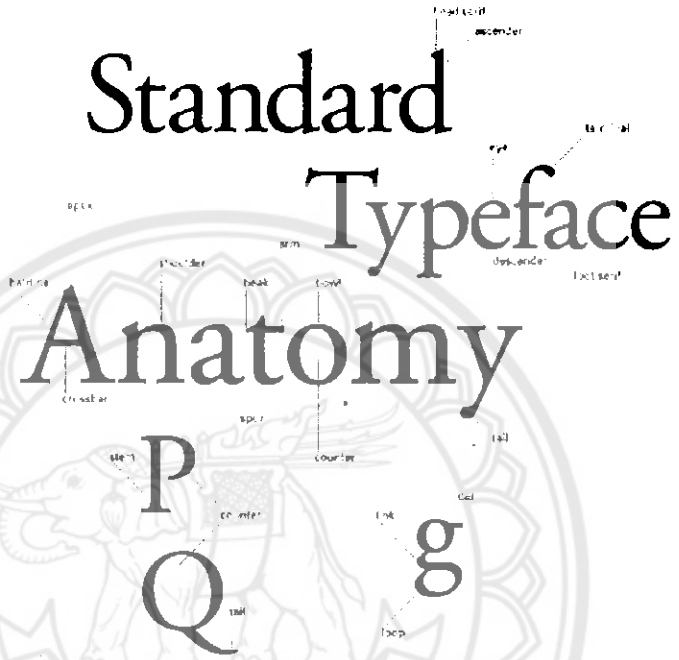
ในการสร้างโปรแกรมออกแบบและจำลองการแปรอักษร 3 มิติ รูปแบบตัวอักษรเป็นสิ่งจำเป็นในการแสดงผล การเลือกชุดแบบจำลองต้องคำนึงถึงข้อความที่แสดงผลออกมา เพื่อให้ผู้ใช้สามารถเข้าใจได้ง่าย ทางคณะผู้จัดทำจึงเลือกใช้แบบตัวอักษร "Block" เนื่องจากตัวอักษร Block มีรายละเอียดเหมาะสมกับการแสดงผลผ่านแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ ซึ่งลักษณะของตัวอักษร Block จะแสดงดังภาพที่ 2.5



ภาพที่ 2.5 ลักษณะตัวอักษรแบบ Block[3]

2.3.2 ส่วนประกอบของตัวอักษร

การออกแบบฟอนต์ต้องคำนึงถึงระยะต่างๆที่มีส่วนเกี่ยวข้องกับการใช้งาน ซึ่งจะมีผลต่อการอ่านและลักษณะเฉพาะของตัวอักษรนั้นๆ ทำให้การออกแบบแบบจำลองไดโอดเปล่งแสงต้องคำนึงถึงส่วนประกอบของตัวอักษรที่แตกต่างกัน ดังภาพที่ 2.6



ภาพที่ 2.6 ส่วนประกอบต่างๆของตัวอักษร[4]

นอกจากนี้ตัวอักษรยังมีระยะมาตรฐานต่างๆ ที่ต้องคำนึงในการออกแบบ 4 ระยะ ได้แก่ x-height, capline, topline, base line และ beardline ซึ่งความสูงโดยรวมของฟอนต์ จะวัดจากเส้น beardline จนถึงเส้น top line ดังภาพที่ 2.7

ภาพที่ 2.7 ระยะต่างๆ ในการออกแบบตัวอักษร[5]

2.3.3 องค์ประกอบพื้นฐานในการสร้างตัวอักษร

- ขนาดตัวอักษร คือจะต้องมีความสัมพันธ์กันระหว่างความกว้างกับความหนา เพราะจะทำให้ตัวอักษรมีความสมส่วน
- ช่องไฟ คือพื้นที่ว่างระหว่างตัวอักษรรวมกับพื้นที่ภายในตัวอักษร
- ความสมดุล คือต้องมีความเท่ากันของจำนวนตัวอักษร ขนาดหรือสี

2.3.3.1 การจัดวางตัวอักษร

- ระยะห่างระหว่างตัวอักษร (Letter spacing)
- ระยะห่างระหว่างคำ (Word spacing)
- ระยะระหว่างบรรทัด (Leading)

2.4 หลักการพื้นฐานของ Light Emitting Diode Cube (LED Cube)

2.4.1 ระบบสมองกลฝังตัว

ระบบสมองกลฝังตัว (Embedded System)[6] เปรียบได้กับเครื่องคอมพิวเตอร์ขนาดเล็กที่ซ่อนตัวอยู่ในอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ซึ่งทำหน้าที่ควบคุมการทำงานของอุปกรณ์นั้นๆ ให้ทำงานได้อย่างอัตโนมัติ แต่อาจมีประสิทธิภาพไม่เทียบเท่าเครื่องคอมพิวเตอร์ทั่วไปเนื่องจากการถูกจำกัดด้วยขนาดที่เล็กกว่านั่นเอง โดยโครงงานนี้ได้เลือกใช้ FiO Board และ RapidSTM32 blockset ในการพัฒนาโครงงาน

2.4.1.1 FiO Boards

อุปกรณ์บอร์ดFiO [4] คือชุดปฏิบัติการทดลองระบบสมองกลฝังตัวที่ใช้ไมโครคอนโทรเลอร์ตระกูล STM32 (สถาปัตยกรรม 32-bits ARM Cortex-M3) ที่ถูกออกแบบมาโดยเฉพาะเพื่อการใช้งานร่วมกับ RapidSTM32 blockset

2.4.1.2 RapidSTM32 blockset

เป็นชุดกล่องคำสั่งแบบรูปภาพที่สามารถใช้งานร่วมกับโปรแกรม [9] Matlab/Simulink และกลไกของ Code Generation ได้ทำให้สามารถแปลคำสั่งโปรแกรมแบบกราฟิกให้เป็นคำสั่งของระบบสมองกลฝังตัวได้ง่าย

2.4.2 หลักการสร้างไดโอดเปล่งแสงทรงลูกบาศก์

2.4.2.1 ไดโอดเปล่งแสง

ไดโอดเปล่งแสงเป็นไดโอดชนิดหนึ่ง ซึ่งภายในไดโอดเปล่งแสงจะประกอบไปด้วยสารกึ่งตัวนำ โดยจะมีการจ่ายไฟบวกไปยังขั้วบวก (แอโนด) และจ่ายไฟลบไปยังขั้วลบ (แคโทด) จะทำให้เกิดการวิ่งของอิเล็คตรอนส่งผลให้เกิดการคลายพลังงานออกมาในรูปคลื่นแสง โดยความสว่างของแต่ละสีขึ้นอยู่กับค่าความต่างศักย์ของขั้วบวกและขั้วลบ ซึ่งไดโอดเปล่งแสงแบบ 3 สี (RGB) มีอยู่ด้วยกัน 2 ชนิด ได้แก่

- ไดโอดเปล่งแสง 3 สีชนิดขั้วบวกร่วม ไดโอดชนิดนี้จะใช้ขาไฟบวกร่วมกัน โดยจะควบคุมความสว่างของแต่ละสีด้วยการจ่ายไฟลบเข้าไปที่ขาลบของแต่ละสี

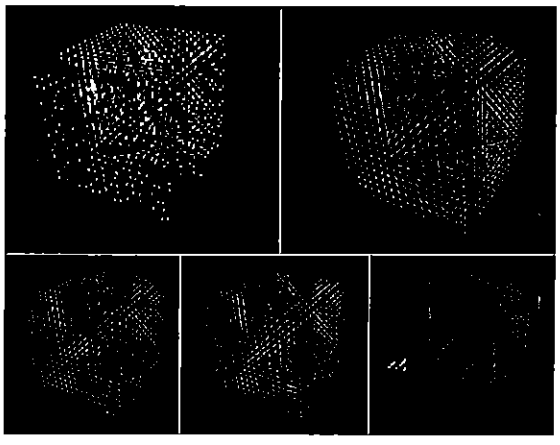
- ไดโอดเปล่งแสง 3 สีชนิดขั้วลบร่วม ไดโอดชนิดนี้จะใช้ขาไฟลบร่วมกัน โดยจะควบคุมความสว่างของแต่ละสีด้วยการจ่ายไฟบวกเข้าไปที่ขาบวกของแต่ละสี

2.4.2.2 ระนาบ

ระนาบมีอยู่ด้วยกัน 2 แบบคือ ระนาบแนวตั้งและระนาบแนวนอน

- ระนาบแนวตั้ง คือระนาบที่ตั้งฉากกับพื้นโดยระนาบแนวนี้จะเรียงตัวต่อกันในแนวตั้ง ซึ่งจะซ้อนต่อกันจากข้างหน้าไปข้างหลัง

- ระนาบแนวนอน คือระนาบที่ขนานกับพื้นโดยระนาบแนวนี้จะเรียงตัวซ้อนทับกันในแนวนอนซึ่งจะซ้อนทับกันจากข้างบนลงข้างล่าง ซึ่งระนาบทั้งสองแบบนั้นสามารถแสดงภาพได้เช่นเดียวกัน โดยระนาบแนวนอนจะต้องต่อสายเข้าไปตั้งฉากกับแนวแกนที่จะแสดงคือต่อในแนวตั้ง ส่วนระนาบแนวตั้งจะต้องต่อสายควบคุมเข้าไปในแกนนอน



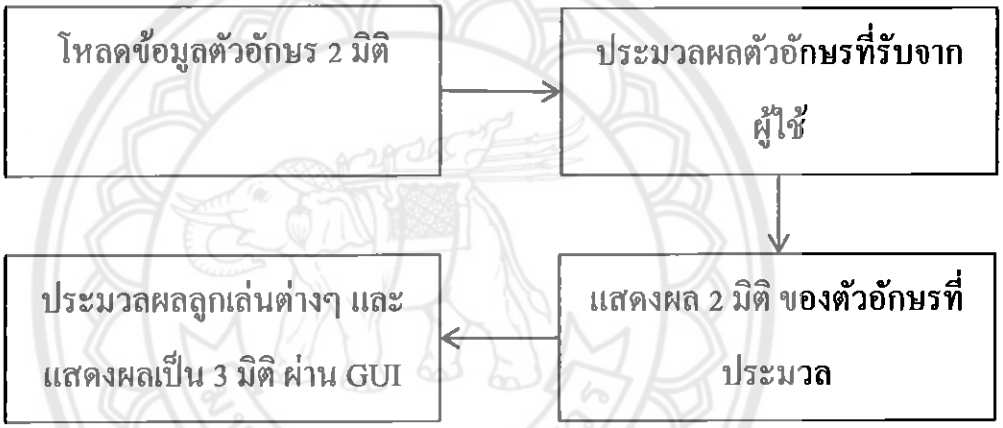
ภาพที่ 2.8 LED Cube 16x16x16 [8]

บทที่ 3

วิธีการดำเนินโครงงาน

เนื้อหาในบทนี้ได้อธิบายถึงขั้นตอนการดำเนินโครงงาน โดยเริ่มตั้งแต่รายละเอียดการออกแบบ อุปกรณ์ที่นำมาใช้ในการจัดทำโครงงาน ทั้งเครื่องมือที่เป็นฮาร์ดแวร์และซอฟแวร์ ตลอดจนถึงขั้นตอนการพัฒนาโปรแกรม ซึ่งมีรายละเอียดดังนี้

ภาพรวมของโครงงาน



3.1 อุปกรณ์ที่ใช้

3.1.1 เครื่องคอมพิวเตอร์

- หน่วยประมวลผล Intel Core i3-330M (2.1 GHz L3 3MB)
- หน่วยความจำ DDR3 ขนาด 2 GB
- การ์ดจอ Intel HD Graphics
- ระบบปฏิบัติการ Windows 7 แบบ 64-bits

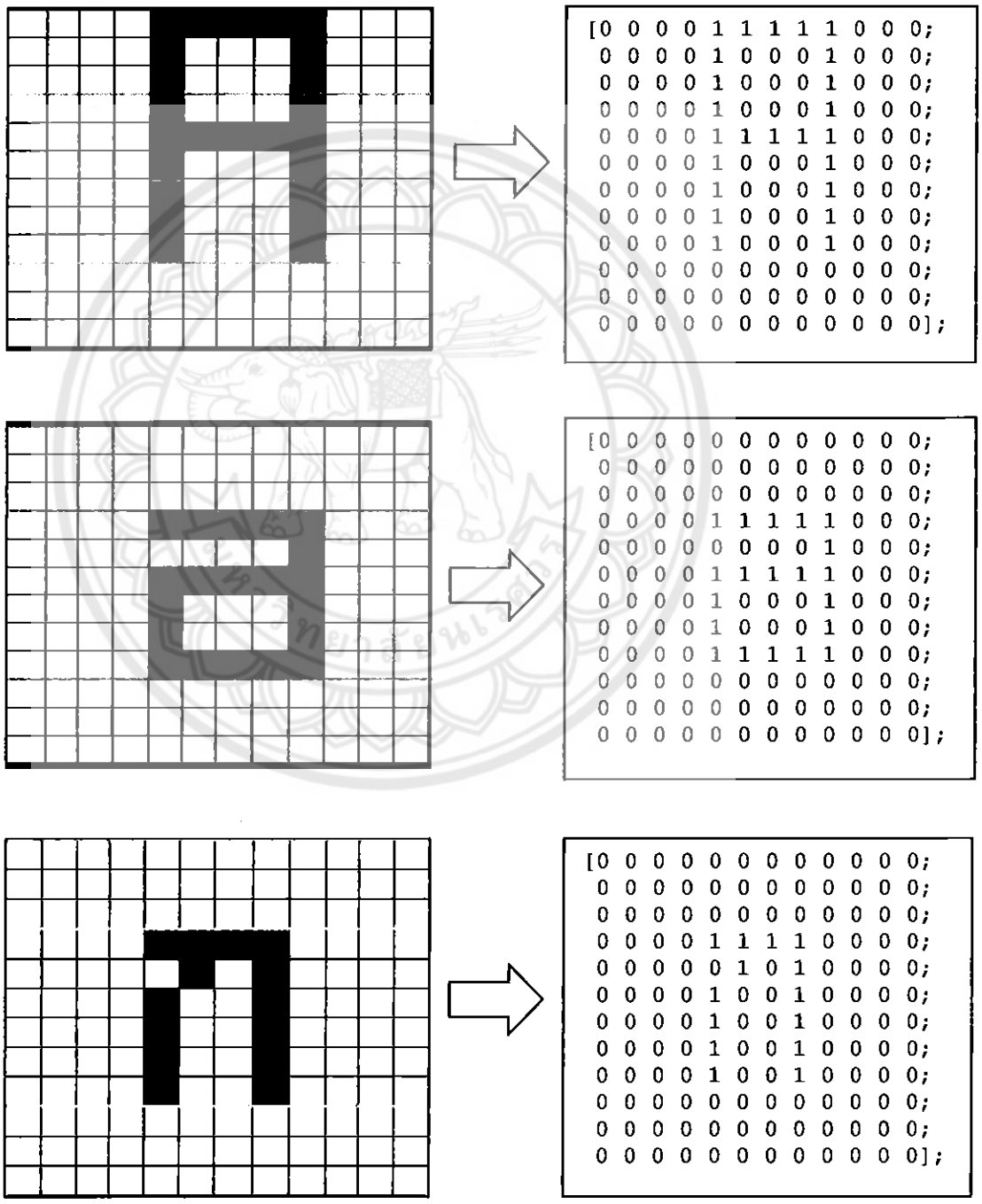
3.1.2 โปรแกรม

- Matlab 2013
- Microsoft Excel 2010

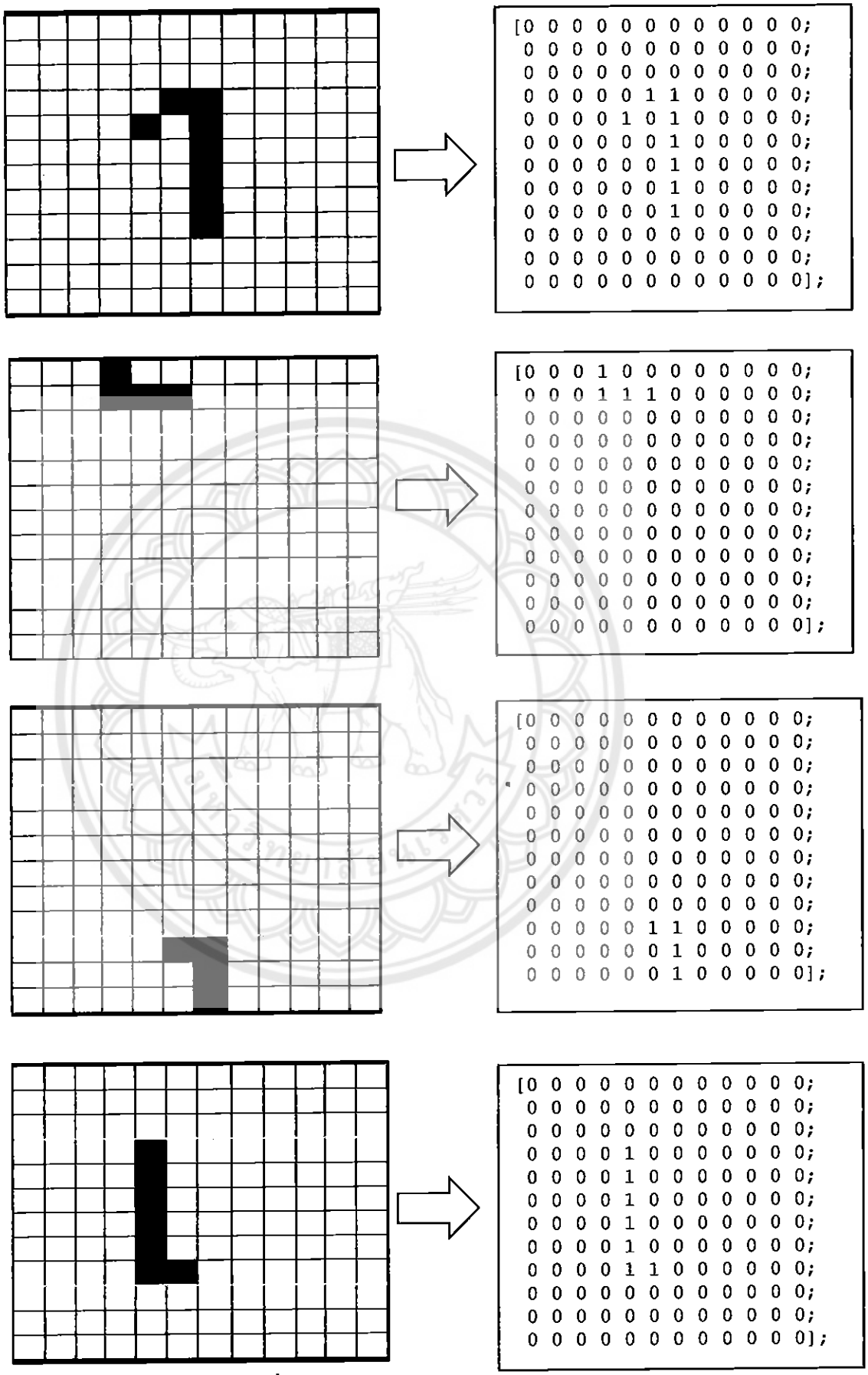
3.2 วิธีการดำเนินโครงงาน

3.2.1 การสร้างตัวอักษร 2 มิติ

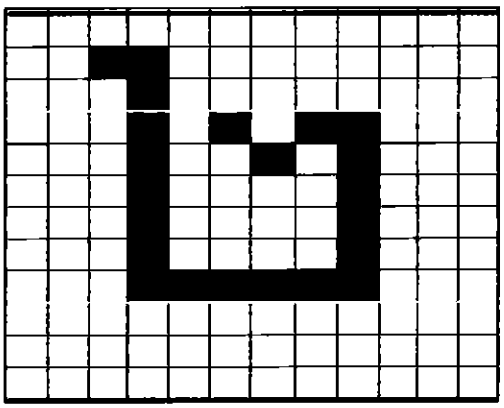
นำรูปแบบตัวอักษรที่เลือกไว้มาออกแบบ โดยการจำลองบล็อก LED ที่แสดงผล เพื่อกำหนดขนาดและตำแหน่งของตัวอักษร ตัวอักษร 2 มิติ จะถูกเก็บในรูปแบบ Binary ในแมตแล็บ (MATLAB) ดังภาพที่ 3.1



ภาพที่ 3.1 ตัวอย่างการสร้างตัวอักษร 2 มิติ



ภาพที่ 3.1 ตัวอย่างการสร้างตัวอักษร 2 มิติ (ต่อ)



```
[0 0 0 0 0 0 0 0 0 0 0 0;
 0 0 1 1 0 0 0 0 0 0 0 0;
 0 0 0 1 0 0 0 0 0 0 0 0;
 0 0 0 1 0 1 0 1 1 0 0 0;
 0 0 0 1 0 0 1 0 1 0 0 0;
 0 0 0 1 0 0 0 0 1 0 0 0;
 0 0 0 1 0 0 0 0 1 0 0 0;
 0 0 0 1 0 0 0 0 1 0 0 0;
 0 0 0 1 1 1 1 1 1 0 0 0;
 0 0 0 0 0 0 0 0 0 0 0 0;
 0 0 0 0 0 0 0 0 0 0 0 0;
 0 0 0 0 0 0 0 0 0 0 0 0];
```

ภาพที่ 3.1 ตัวอย่างการสร้างตัวอักษร 2 มิติ (ต่อ)

แล้วเก็บข้อมูลทั้งหมดไว้ในไฟล์ข้อมูล

3.2.2 การแปลงตัวอักษรให้อยู่ในรูป Unicode ของแมตแล็บ (MATLAB)

เมื่อทำการสร้างตัวอักษร 2 มิติ เก็บไว้ในไฟล์ข้อมูลเรียบร้อยแล้ว ในการเรียกใช้ตัวอักษรที่เก็บในไฟล์ข้อมูล จะใช้คำสั่ง Unicode ของแมตแล็บ (MATLAB) ดังนี้

```
unicode = unicode2native(inputcharacter);
```

คำสั่งดังกล่าวจะช่วยในการเรียกใช้ตัวอักษร เพื่อแก้ไขปัญหาตัวอักษรภาษาไทยในแมตแล็บ (MATLAB) ซึ่งการแปลงตัวอักษรให้อยู่ในรูปแบบของ Unicode มีตัวอย่างดังภาพที่ 3.2

```
>> unicode = unicode2native('A')

unicode =

   65
```

```
>> unicode = unicode2native('a')

unicode =

   97
```

```
>> unicode = unicode2native('1')

unicode =

   49
```

```
>> unicode = unicode2native('ก')

unicode =

  161
```

ภาพที่ 3.2 ตัวอย่างการใช้คำสั่ง Unicode ของแมตแล็บ (MATLAB)

```
>> unicode = unicode2native('ั')

unicode =

  209
```

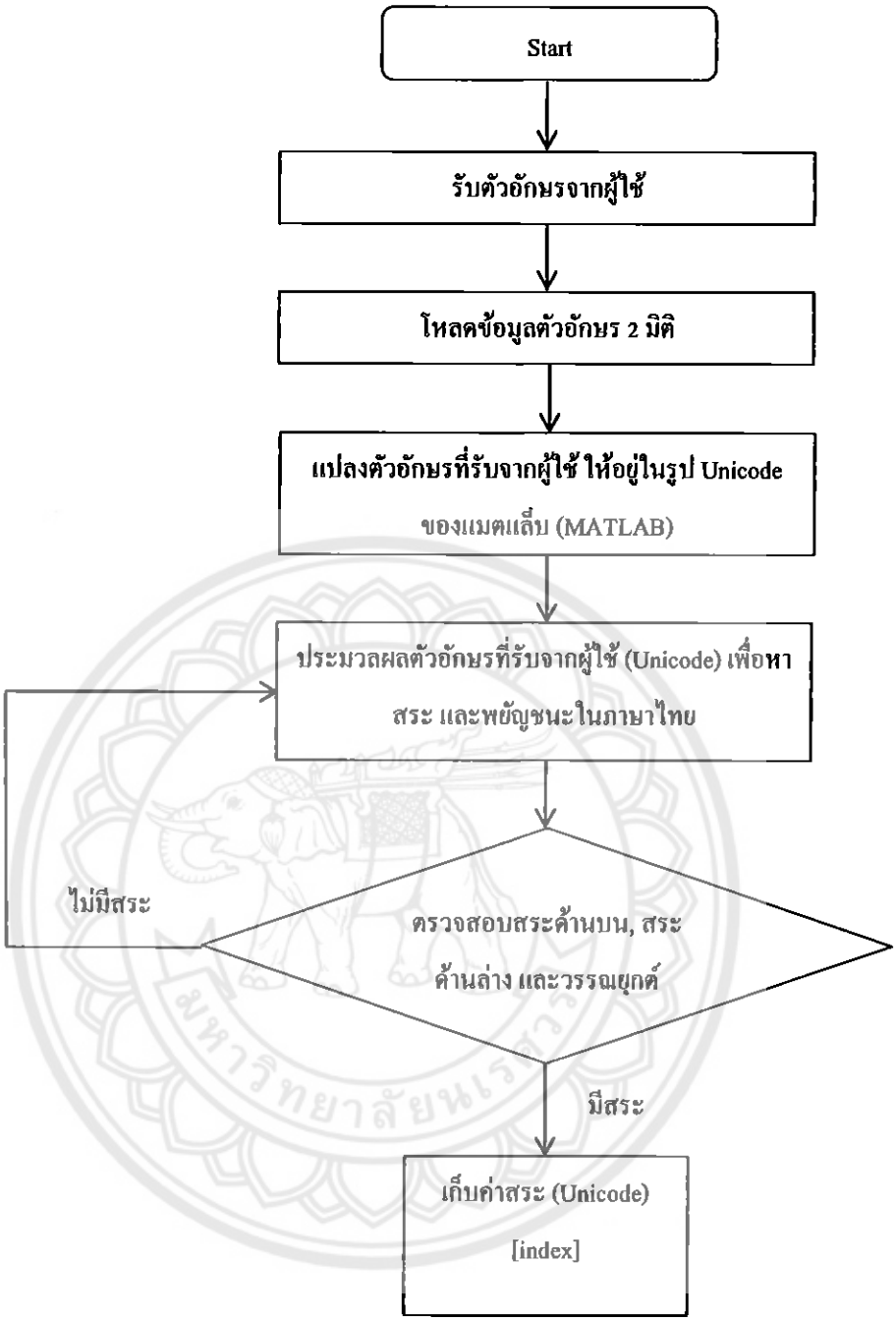
```
>> unicode = unicode2native('ุ')

unicode =

  216
```

```
>> unicode = unicode2native('ิ')

unicode =

  224
```

ภาพที่ 3.2 ตัวอย่างการใช้คำสั่ง Unicode ของแมตแล็บ (MATLAB) (ต่อ)

3.2.3 การตรวจสอบสระด้านบน สระด้านล่าง และวรรณยุกต์ ในภาษาไทย

เนื่องจากสระในภาษาไทยมีทั้งตัวที่อยู่บนพยัญชนะ และอยู่ด้านล่างของพยัญชนะ ดังนั้น จึงจำเป็นที่จะต้องนำมารวมกับพยัญชนะตัวนั้นๆ โดยมีขั้นตอนดังภาพที่ 3.3

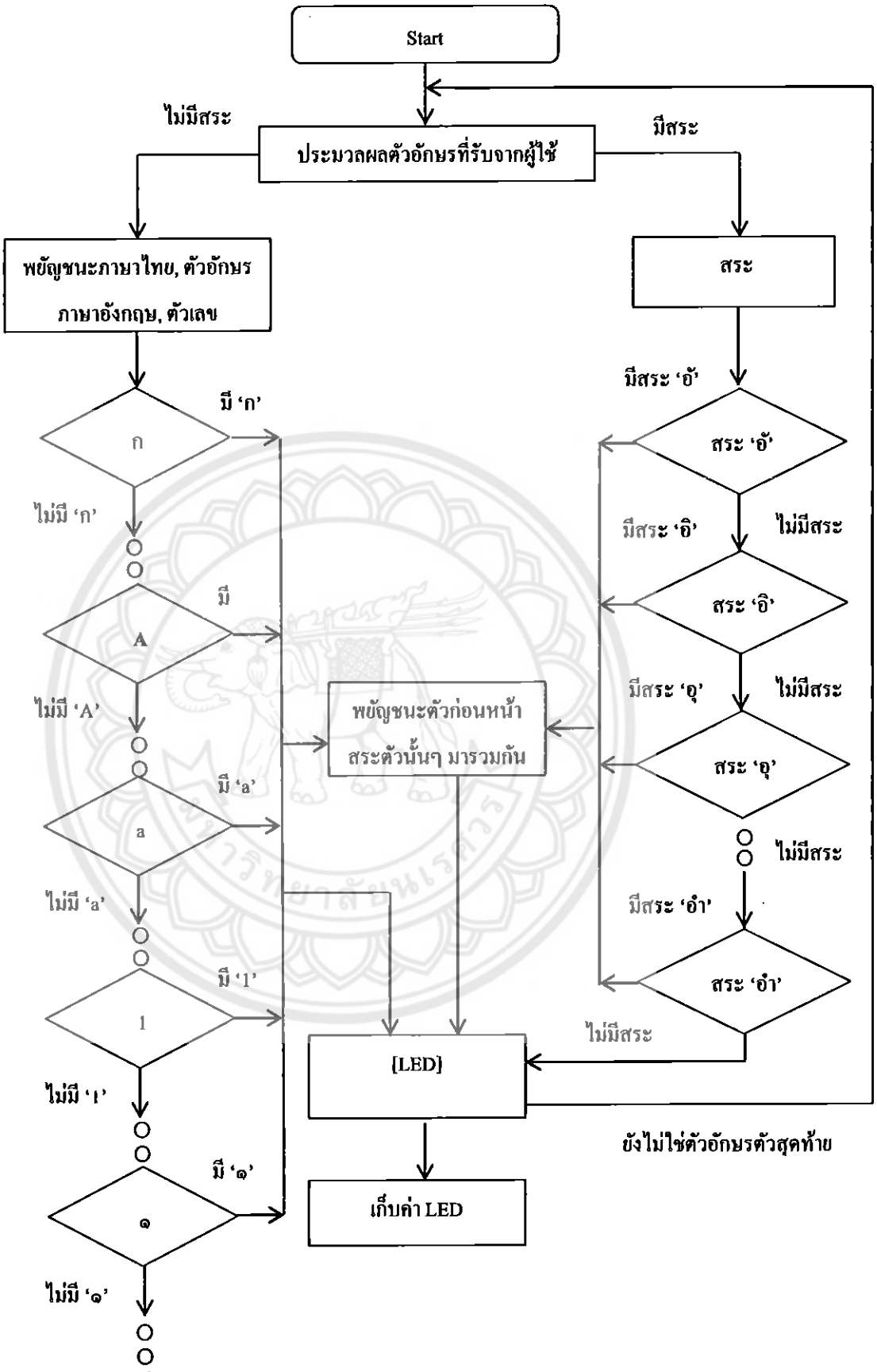


ภาพที่ 3.3 แผนภาพการทำงานของการตรวจสอบสระในภาษาไทย

ซึ่งมี code การทำงานดังภาคผนวก (ภาพที่ ผ.1) การตรวจสอบจะแยกออกเป็น 2 ส่วน คือ ตรวจสอบสระที่อยู่ข้างบนตัวอักษรรวมไปถึงวรรณยุกต์ด้วยและตรวจสอบสระที่อยู่ด้านล่างตัวของอักษร โดยจะเช็คค่าของตัวอักษรที่รับเข้ามา เมื่ออินพุตที่รับเข้ามาเป็นสระโปรแกรมจะเช็คสระว่าเป็นสระที่อยู่ข้างบนหรือไม่ก่อน ถ้าไม่ใช่สระที่อยู่ข้างบน จึงจะไปเช็คสระที่เป็นสระข้างล่างและเมื่อเจอสระที่รับเข้ามาแล้ว จะเก็บค่าของดัชนีและเก็บ Unicode ของสระนั้นๆ เพื่อนำไปใช้ในขั้นตอนต่อไป

3.2.4 การตรวจสอบชนิดตัวอักษรของข้อมูลที่รับเข้ามา

เมื่อรับข้อมูลตัวอักษรเข้ามา โปรแกรมจะทำการตรวจสอบชนิดของตัวอักษร ถ้าตัวอักษรนั้นไม่ใช่สระด้านบนและด้านล่างจะทำการตรวจสอบว่า เป็นตัวอักษรภาษาอังกฤษ หรือภาษาไทย หรือตัวเลข แต่ถ้าเป็นสระด้านบนหรือสระด้านล่างจะทำการตรวจสอบว่าเป็นสระอะไร จากนั้นทำการตรวจสอบตัวอักษรไปเรื่อยๆจนถึงตัวอักษรตัวสุดท้าย ซึ่งมี flow chart การทำงานของการตรวจสอบชนิดตัวอักษรของผู้ใช้ ดังภาพที่ 3.4

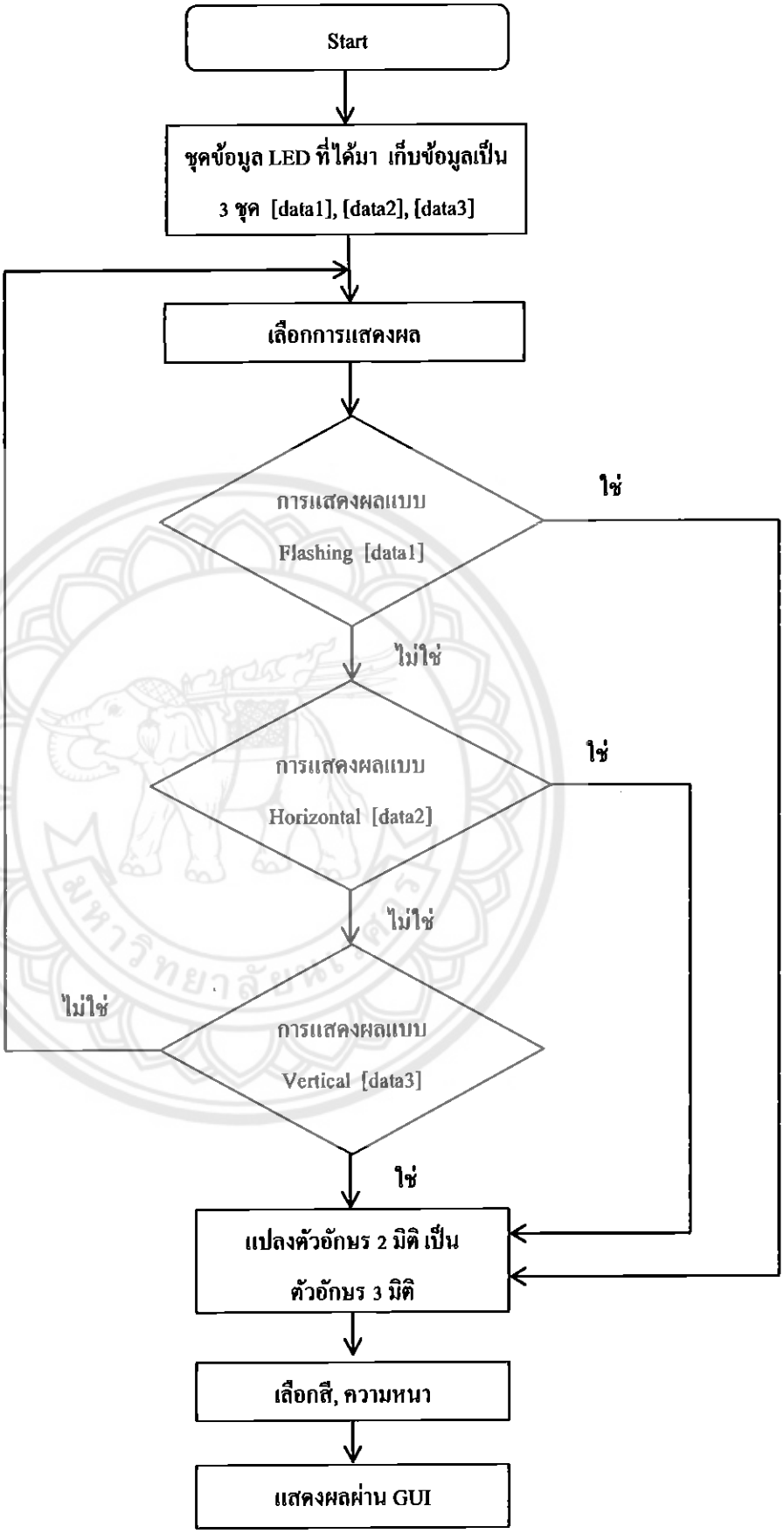


ภาพที่ 3.4 แผนภาพแสดงการทำงานของการตรวจสอบชนิดตัวอักษรของผู้ใช้

เมื่อรับตัวอักษรจากผู้ใช้เข้ามาโปรแกรมจะตรวจเช็คว่าเป็นอักษรอะไร ดัง code การตรวจสอบตัวอักษรในภาคผนวก (ภาพที่ ผ.2) และเมื่อเจอสระโปรแกรมจะนำค่าดัชนีและ Unicode ที่เก็บไว้มาเช็คค่าของพยัญชนะตัวก่อนหน้าของสระ โดยนำค่าของดัชนีมาลบ 1 เพื่อหาพยัญชนะก่อนหน้า จากนั้นจึงนำพยัญชนะและสระมาผสมกัน ดัง code ในภาคผนวก (ภาพที่ ผ.3) ซึ่ง LEDvowel คือตัวสระด้านบน, ด้านล่าง และวรรณยุกต์ ส่วน LED คือพยัญชนะตัวก่อนหน้าของสระ จากนั้นจะใช้ | (OR) เพื่อนำสระและพยัญชนะมารวมกัน

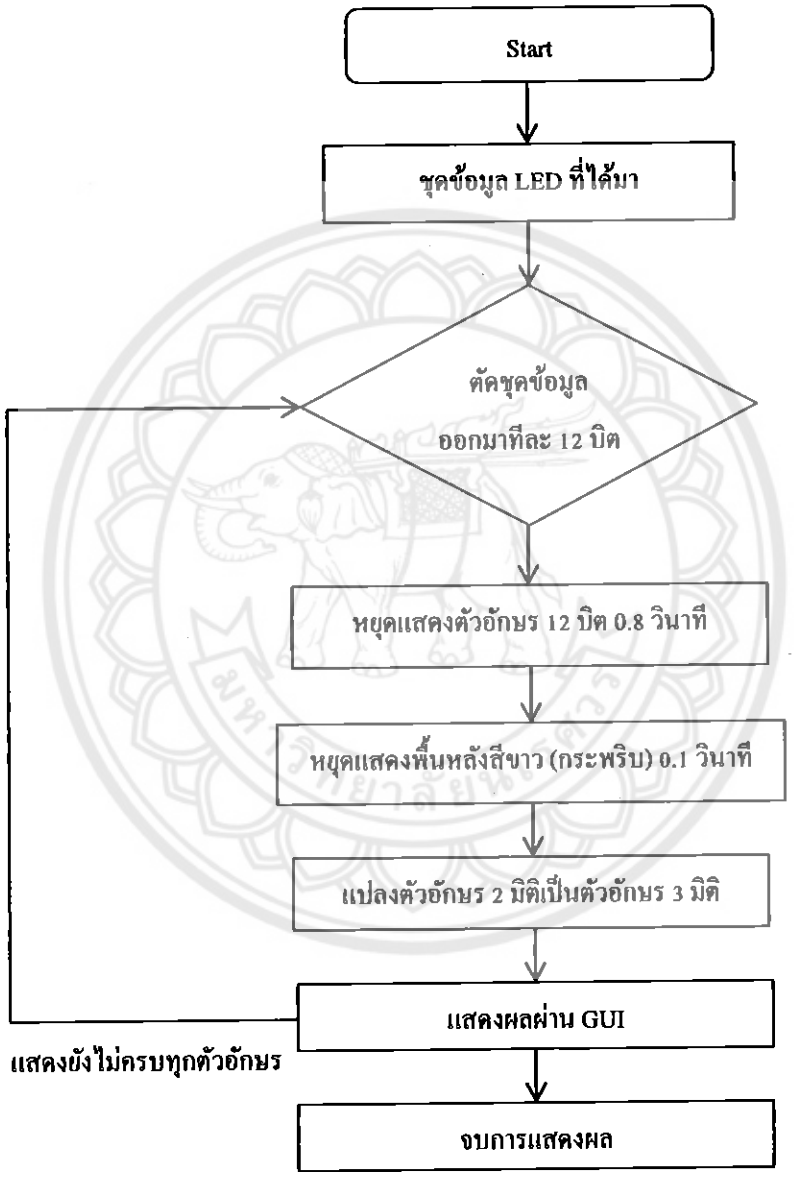
3.2.5 การแปลงตัวอักษร 2 มิติ เป็นตัวอักษร 3 มิติ

เมื่อรับข้อมูลเข้ามาโปรแกรมจะตรวจสอบตัวอักษรและเมื่อตรวจสอบตัวอักษรเสร็จจะทำการแปลงตัวอักษร 2 มิติ เป็น 3 มิติ ในการแสดงผลตัวอักษร จะต้องเลือกรูปแบบการแสดงผล ซึ่งมีรูปแบบการแสดงผลให้ผู้ใช้เลือกทั้งหมด 3 รูปแบบ คือ กระพริบ (Flashing), เคลื่อนที่ในแนวนอน (Horizontal), เคลื่อนที่ในแนวตั้ง (Vertical) โดยมีการทำงาน ดังภาพที่ 3.5



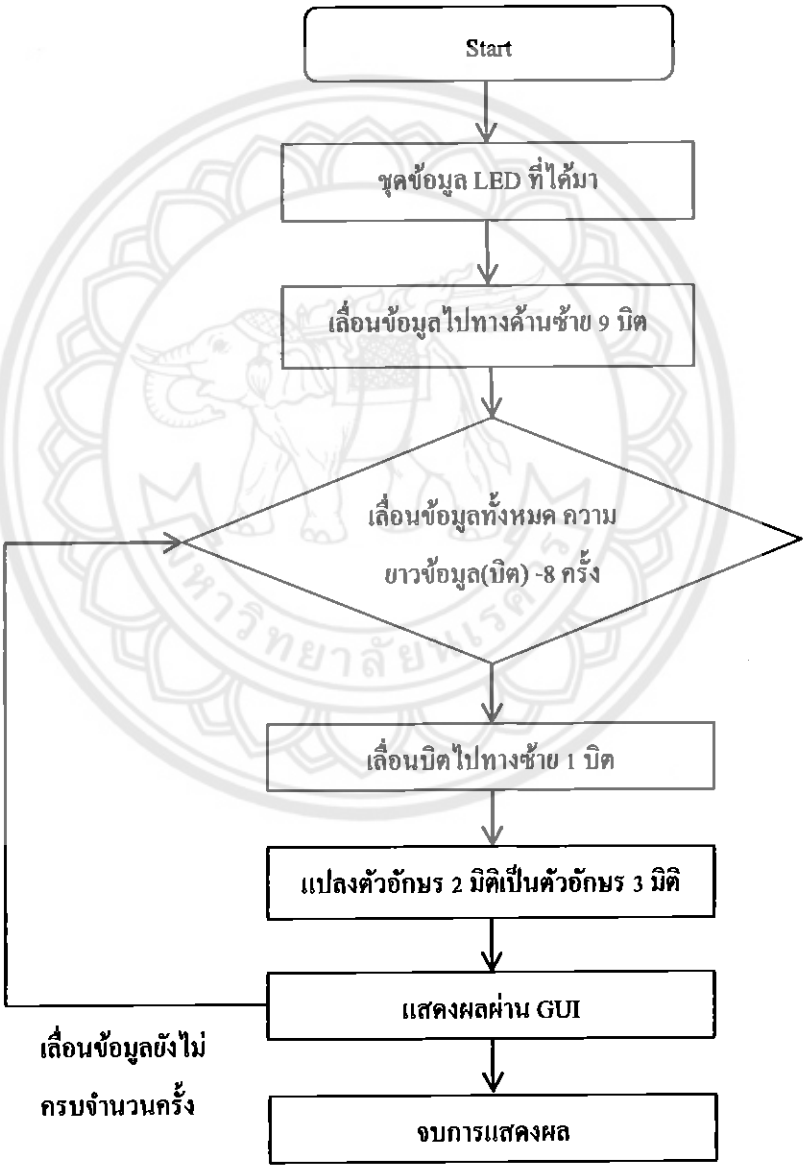
ภาพที่ 3.5 แผนภาพแสดงการทำงานของการแปลงตัวอักษร 2 มิติ เป็นตัวอักษร 3 มิติ

การแสดงผลแบบ Flashing หรือแบบกระพริบนั้น การทำงานจะเริ่มจากการรับชุดข้อมูลที่ได้มาตัดออกมาทีละ 12 บิต เพื่อให้ตัวอักษรแสดงผลทีละตัว จากนั้นจะนำตัวอักษรที่เป็น 2 มิติมาแปลงเป็นตัวอักษร 3 มิติ ซึ่งจะแสดงผลตัวอักษรตัวละ 0.8 วินาที และใช้เวลาในการกระพริบ 0.1 วินาที การทำงานจะแสดงดังแผนภาพที่ 3.6



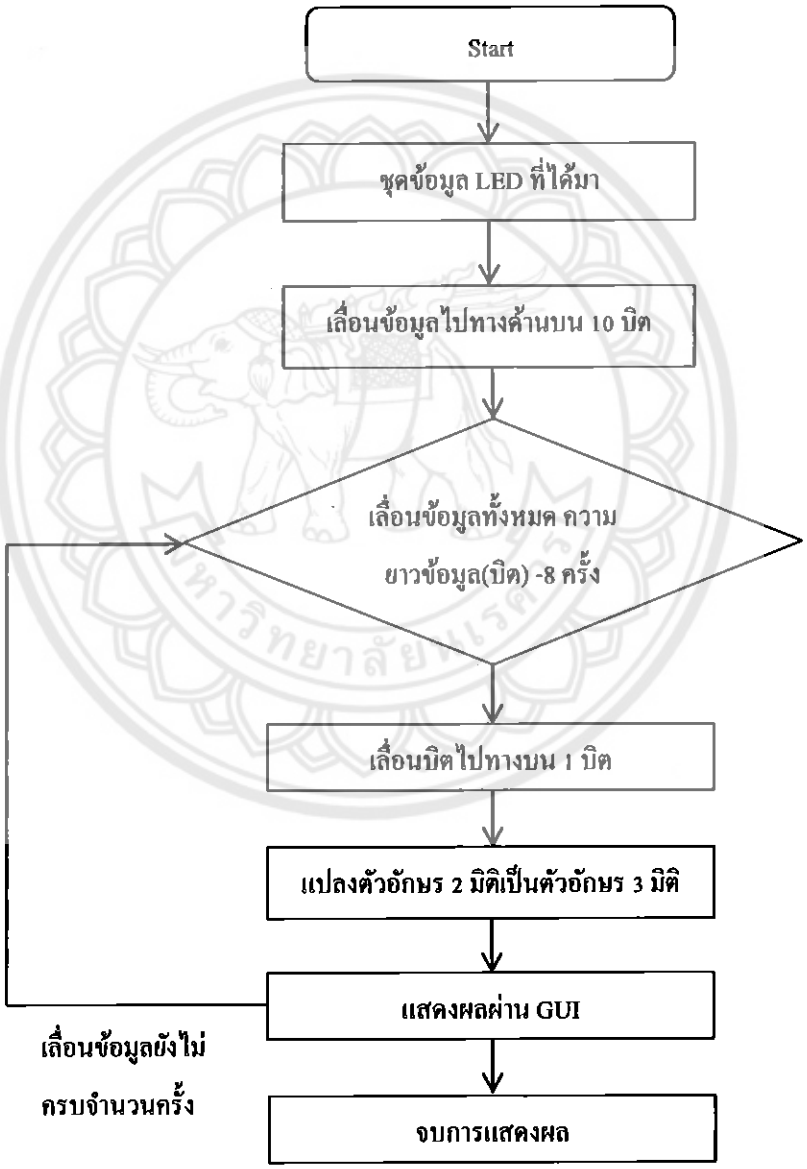
ภาพที่ 3.6 แผนภาพแสดงการทำงานแบบ Flashing

การแสดงผลแบบ Horizontal หรือการเลื่อนในแนวนอน การทำงานจะเริ่มจากการรับชุดข้อมูลที่ได้มา มาทำการเลื่อนข้อมูลไปทางด้านซ้าย 9 บิต โดนจะทำการเลื่อนทั้งหมด ด้วยการนำความยาวของชุดข้อมูลมาลบด้วย 8 ผลลัพธ์ที่ได้คือจำนวนครั้งที่ใช้ในการเลื่อนผ่านหน้าต่าง GUI ในการเลื่อนแต่ละครั้งจะทำการเลื่อนบิตไปทางด้านซ้ายทีละ 1 บิต ทุกครั้ง จากนั้นจะนำตัวอักษรที่เป็น 2 มิติมาแปลงเป็นตัวอักษร 3 มิติ ซึ่งจะแสดงผลตัวอักษรในการเลื่อนบิตข้อมูล ครั้งละ 0.1 วินาที การทำงานจะแสดงดังแผนภาพที่ 3.7



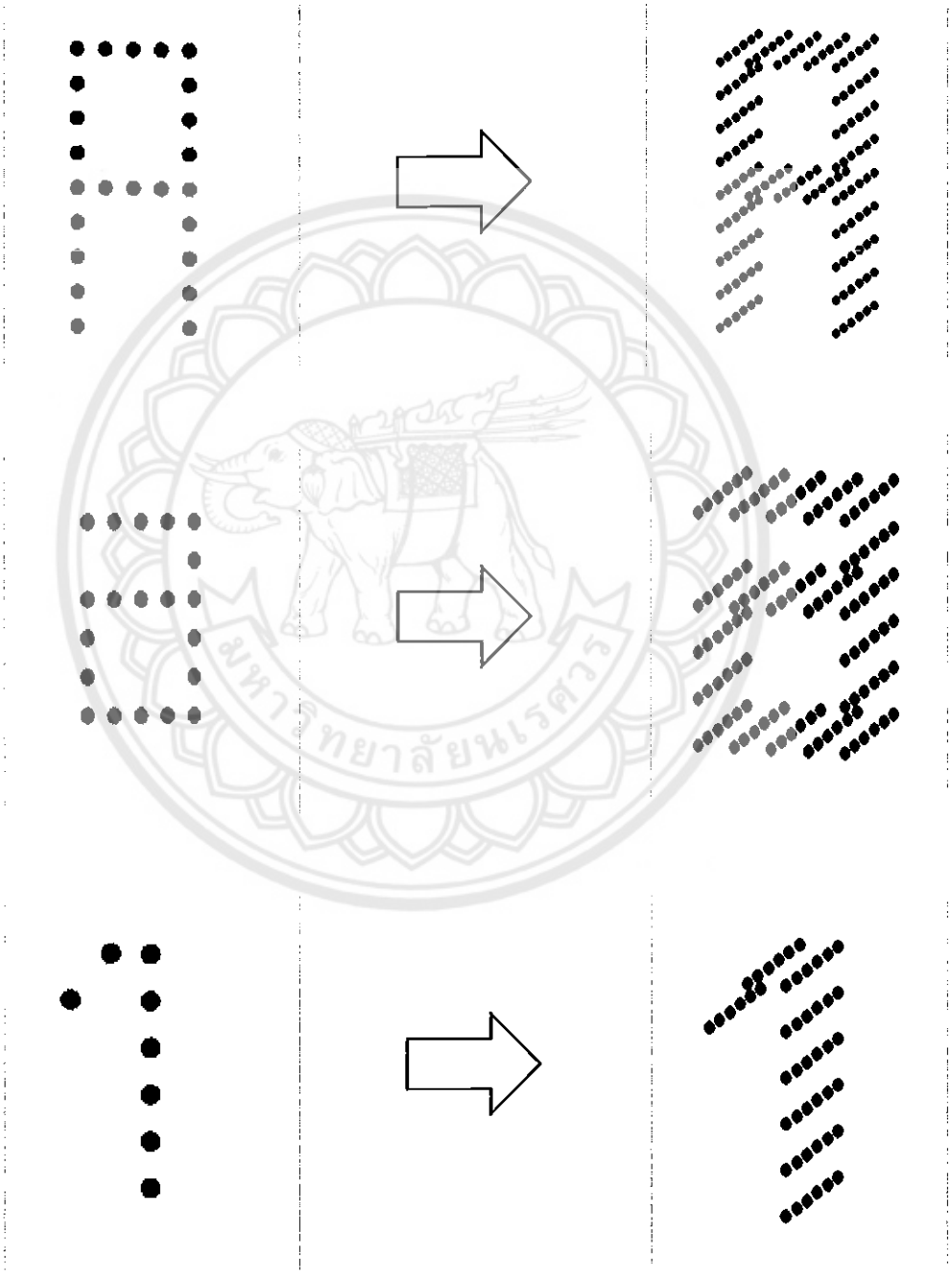
ภาพที่ 3.7 แผนภาพแสดงการทำงานแบบ Horizontal

การแสดงผลแบบ Vertical หรือการเลื่อนในแนวตั้ง การทำงานจะเริ่มจากการรับชุดข้อมูลที่ได้มา มาทำการเลื่อนข้อมูลไปทางด้านบน 10 บิต โดนจะทำการเลื่อนทั้งหมด ด้วยการนำความยาวของชุดข้อมูลมาลบด้วย 8 ผลลัพธ์ที่ได้คือจำนวนครั้งที่ใช้ในการเลื่อนผ่านหน้าต่าง GUI ในการเลื่อนแต่ละครั้งจะทำเลื่อนบิตไปทางด้านบนทีละ 1 บิต ทุกครั้ง จากนั้นจะนำตัวอักษรที่เป็น 2 มิติมาแปลงเป็นตัวอักษร 3 มิติ ซึ่งจะแสดงผลตัวอักษรในการเลื่อนบิตข้อมูล ครั้งละ 0.1 วินาที การทำงานจะแสดงดังแผนภาพที่ 3.8

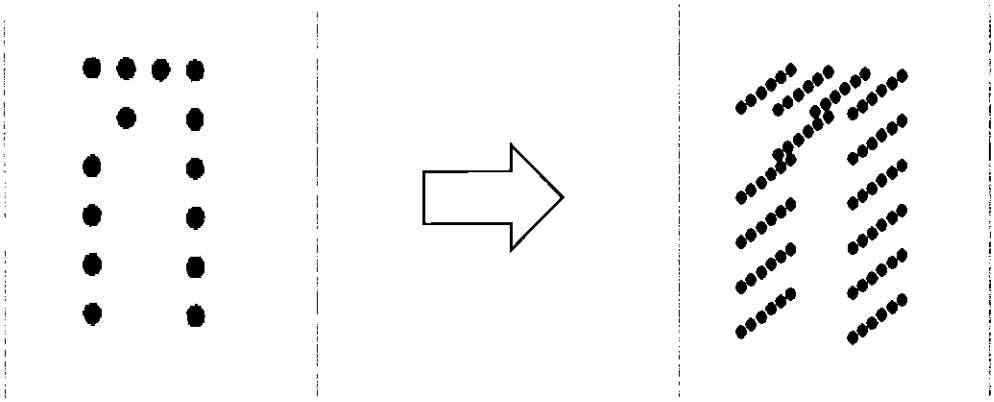


ภาพที่ 3.8 แผนภาพแสดงการทำงานแบบ Vertical

เมื่อตรวจสอบตัวอักษรที่รับเข้ามาซึ่งเป็นตัวอักษรในรูปแบบ 2 มิติเสร็จเรียบร้อยแล้ว โปรแกรมจะทำการวนลูปเช็คค่าตัวอักษรตามแกน y และ z แล้วกำหนดค่าความหนาของตัวอักษรตามแกน x จากนั้นจึงใช้คำสั่ง scatter3 เพื่อสร้างตัวอักษร 3 มิติโดยมี code การทำงานดังภาพที่ ผ.4 ตัวอย่างแสดงผลการแปลงอักษร 3 มิติ เป็นตัวอักษร 3 มิติ ดังภาพที่ 3.9



ภาพที่ 3.9 ตัวอย่างการแสดงผลการแปลงอักษร 2 มิติ เป็นอักษร 3 มิติ



ภาพที่ 3.9 ตัวอย่างการแสดงผลการแปลงอักษร 2 มิติ เป็นอักษร 3 มิติ (ต่อ)

3.2.4 สีและความหนา

3.2.4.1 การเลือกสี

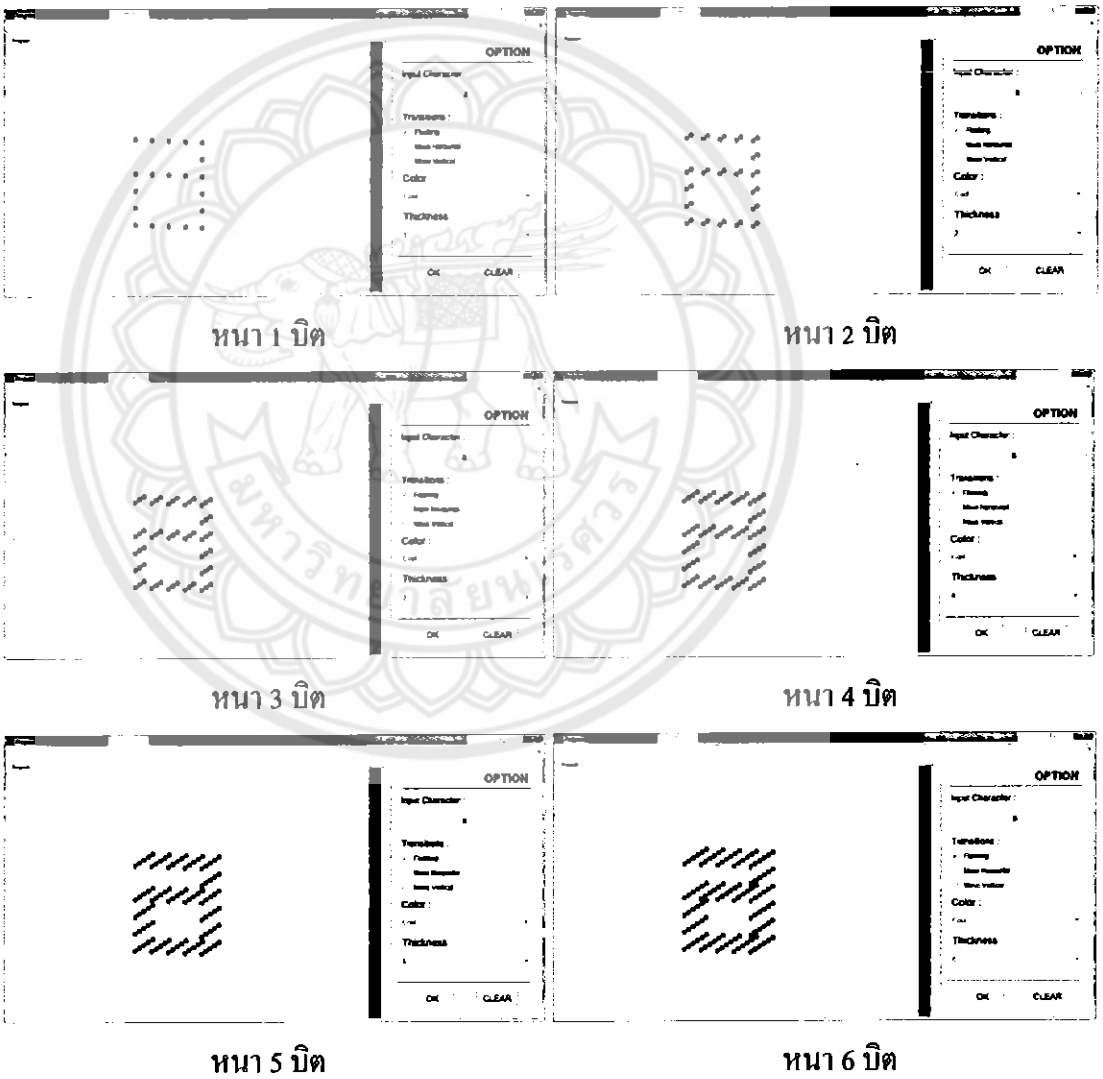
การควบคุมสีของแมตแล็บ (MATLAB) ก็เหมือนการควบคุมสีของโทรทัศน์ซึ่งจะเป็นการผสมระหว่างแม่สี 3 สีคือ แดง-เขียว-น้ำเงิน หรือที่เรียกว่า Red-Green-Blue (RGB) Color นั่นก็คือสี โดยสีดำจะมี RGB color เป็น (0, 0, 0) ส่วนสีขาวจะมี RGB color เป็น (1 , 1, 1) ดังนั้นหากต้องการสร้างสีใหม่ก็เพียงแต่เลือกระดับสี RGB ให้มีสัดส่วนที่เหมาะสมตามความต้องการของเราเท่านั้น โดยค่าที่กำหนดของแต่ละสี RGB จะต้องอยู่ ระหว่าง 0 ถึง 1 โดย color map จะเป็น matrix ที่มีขนาด m x 3 โดยแต่ละ row ซึ่งมี 3 ตัวแต่ละตัวจะแทน RGB color โดยมีการสร้างสีดัง code ดังภาคผนวก (ภาพที่ ผ.5)

โปรแกรมสามารถเลือกสีตามที่ต้องการได้โดยการเลือกจากเมนูที่กำหนดไว้ในหน้าต่างโปรแกรมซึ่งได้สร้างสีไว้ก่อนแล้ว เมื่อผู้ใช้เลือกสีตามที่ต้องการ โปรแกรมจะดึงสีมาแสดงโดยใช้คำสั่ง colormap ดัง code ภาคผนวก (ภาพที่ ผ.6) และแสดงตัวอย่างสีดังภาพที่ 3.10

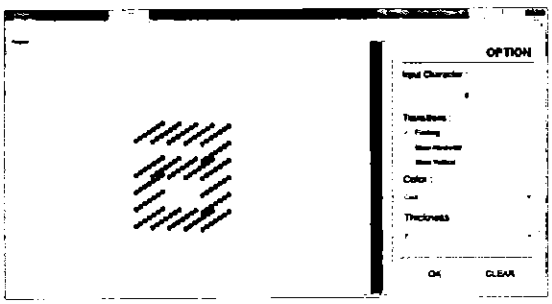
ภาพที่ 3.10 ตัวอย่างการแสดงผลของสี

3.2.4.2 การเลือกความหนา

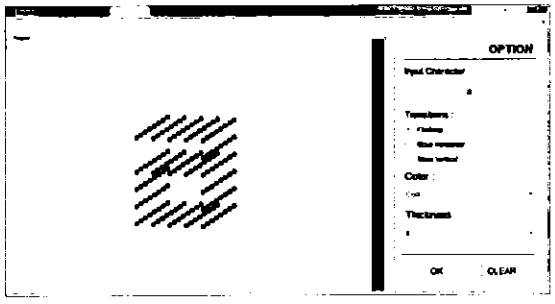
ความหนาของตัวอักษรเป็นสิ่งสำคัญอีกอย่างหนึ่งในการแสดงผลตัวอักษร การเลือกความหนาที่เหมาะสมกับการนำไปใช้ ถือว่าผู้ใช้ให้ความสำคัญ ซึ่งโปรแกรมสามารถเลือกความหนาตามที่ต้องการตั้งแต่ 1–12 บิตได้โดยการเลือกจากเมนูที่กำหนดไว้ในหน้าต่างโปรแกรม ซึ่งความหนาที่เหมาะสมกับตัวอักษร คือ 5-7 บิต แต่ทั้งนี้ก็ขึ้นอยู่กับผู้ใช้ว่าต้องการความหนาของตัวอักษรเท่าไหร่ โดย code การทำงานในการเลือกความหนาเป็นไปตามภาคผนวก (ภาพที่ ผ.7) และการแสดงผลความหนาของตัวอักษรเป็นดังภาพที่ 3.11



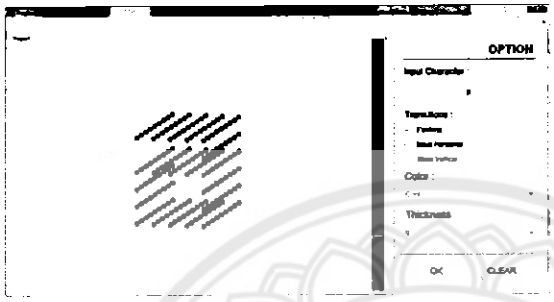
ภาพที่ 3.11 ตัวอย่างการแสดงผลความหนาในแต่ละ bits



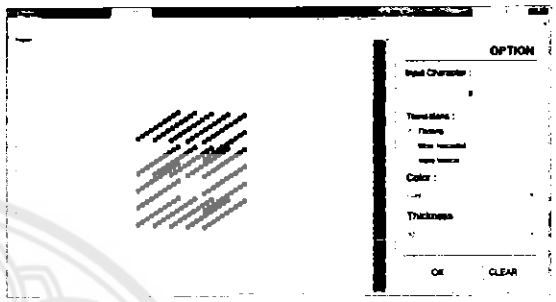
หนา 7 บิต



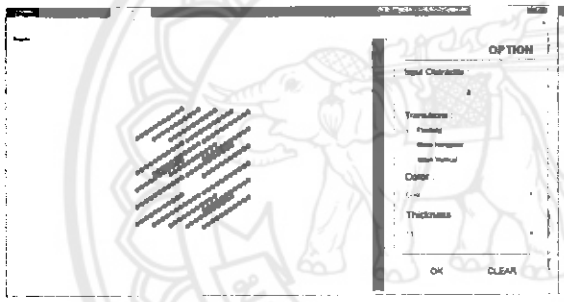
หนา 8 บิต



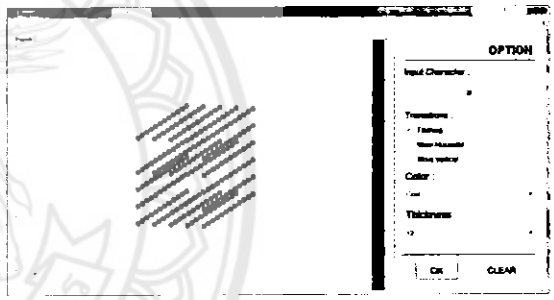
หนา 9 บิต



หนา 10 บิต



หนา 11 บิต

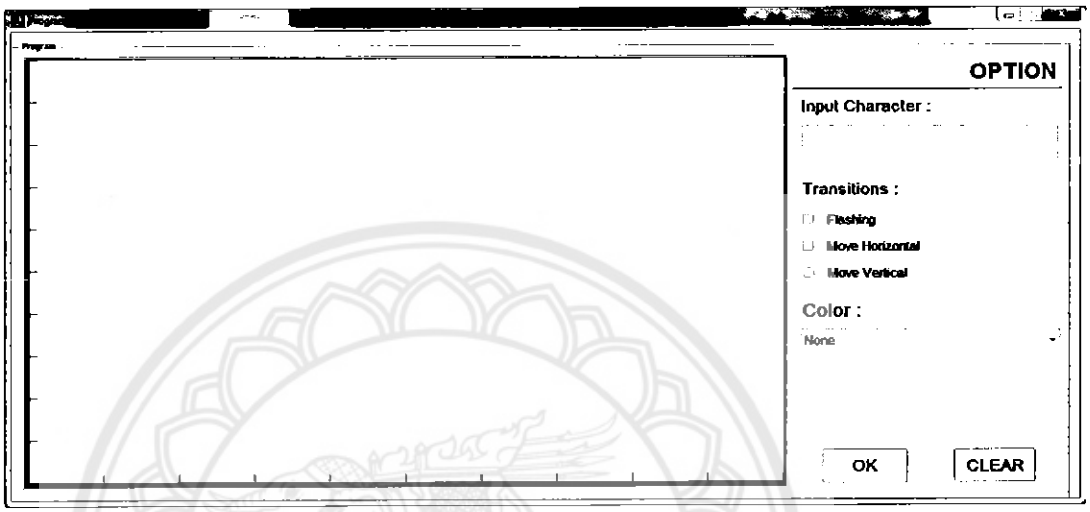


หนา 12 บิต

ภาพที่ 3.11 ตัวอย่างการแสดงผลความหนาในแต่ละบิต (ต่อ)

3.2.4 การออกแบบโปรแกรม

การออกแบบหน้า GUI ของโปรแกรม จำเป็นต้องคำนึงถึงความน่าสนใจ และการใช้งานที่ง่าย เพื่อให้ผู้ใช้สามารถใช้งานได้อย่าสงสะดวกและง่ายต่อการออกแบบตัวอักษร 3 มิติ ดังภาพที่ 3.12



ภาพที่ 3.12 GUI การแสดงผลของโปรแกรม

บทที่ 4

ผลการดำเนินโครงงาน

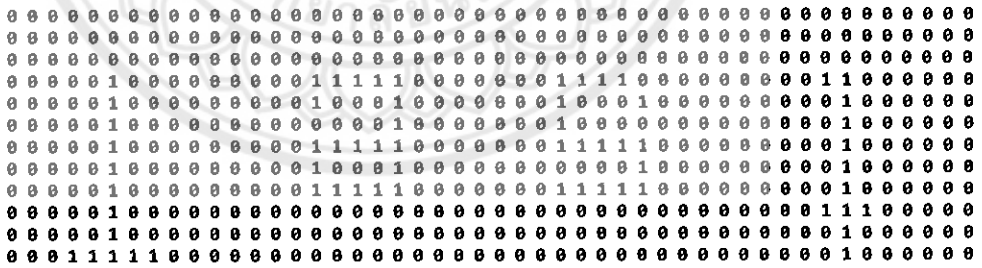
เนื้อหาในบทนี้ได้รวบรวมผลการทดลองการแสดงผล 3 มิติ ในลักษณะต่างๆ โดยผ่านแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ ซึ่งมีรายละเอียดดังนี้

4.1 การแสดงผลแบบกระพริบ (Flashing)

การแสดงผล 3 มิติ ในลักษณะของการกระพริบโดยผ่านแบบจำลอง เมื่อรับข้อมูลตัวอักษร 2 มิติ มาแปลงเป็น 3 มิติแล้ว ถ้าผู้ใช้กำหนด Transition เป็น Flashing ข้อความจะปรากฏทีละตัวอักษร เหมือนการกระพริบตัวอักษร ซึ่งทางคณะผู้จัดทำได้นำตัวอย่างการแสดงผลแบบกระพริบโดยใช้ข้อความในตัวอย่างว่า "Test" ซึ่งใช้เวลาในการแสดงตัวอักษรตัวละ 0.8 วินาที และใช้เวลาในการเปลี่ยนตัวอักษร 0.1 วินาที

4.1.1 ตัวอักษร 2 มิติ

เมื่อทำการป้อนข้อมูลตัวอักษรเข้ามาในโปรแกรม ซึ่งในตัวอย่างคือ ตัวอักษร 'T', ' e', 's' และ 't' โปรแกรมจะทำการดึงชุด binary ของตัวอักษร 2 มิติดังกล่าว ดังภาพที่ 4.1



ภาพที่ 4.1 แสดงชุด binary ของตัวอักษร 2 มิติ แบบ Flashing

นำตัวอักษร 2 มิติที่เป็น binary ที่ได้มา นำมาตัดออกทีละ 12 บิต จากนั้นก็จะนำ binary ที่ได้มาแปลงเป็นตัวอักษร 3 มิติ ดังภาพที่ 4.2

cutdata =

```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
```

cutdata =

```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
```

cutdata =

```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
```

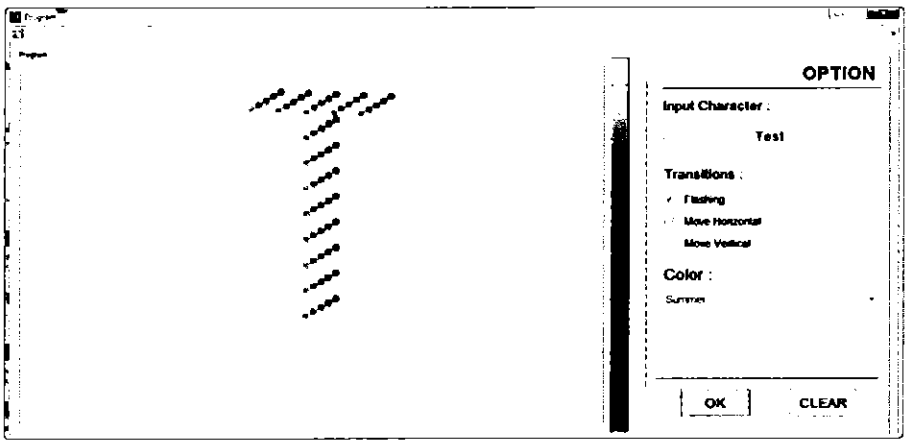
cutdata =

```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0
```

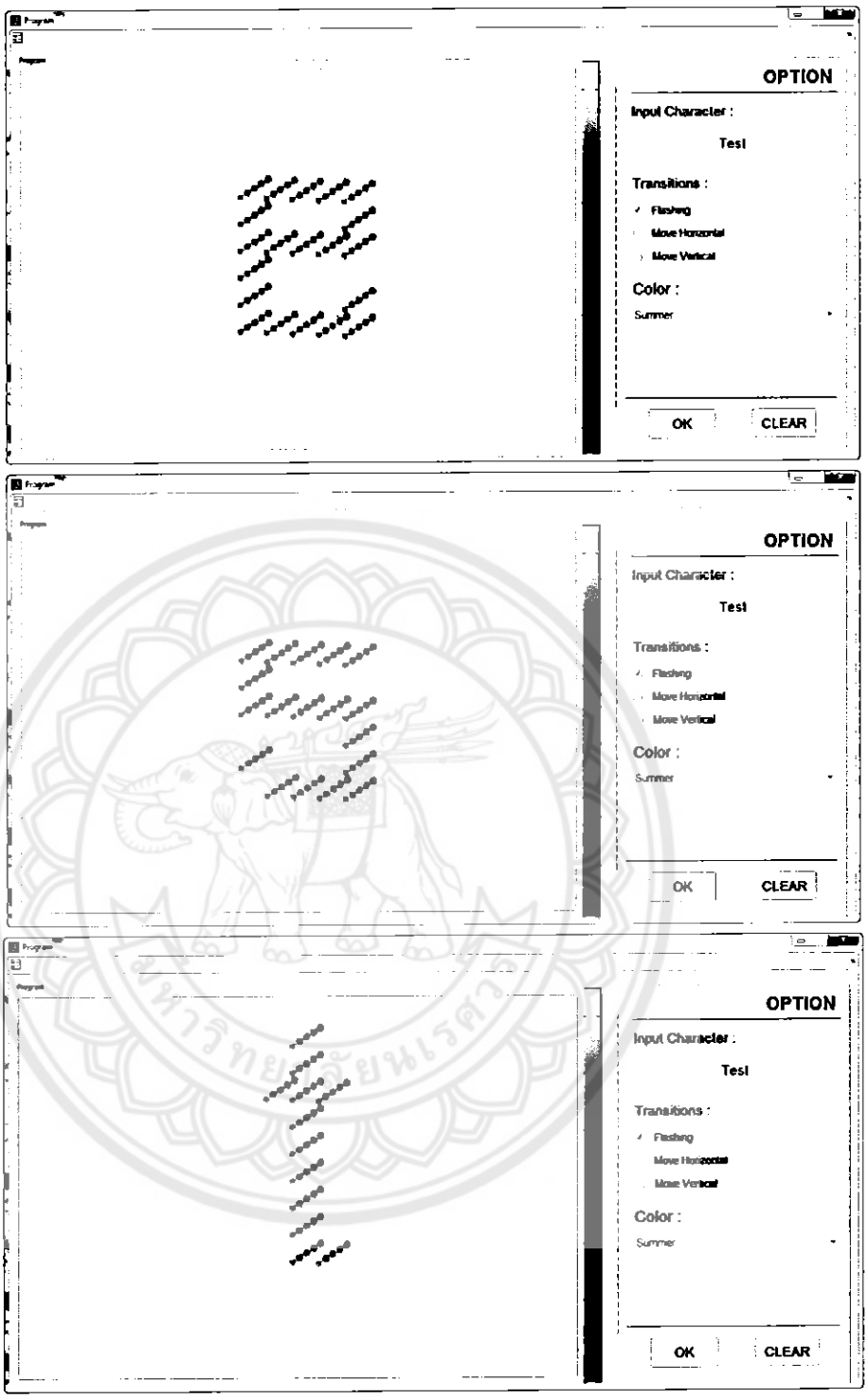
ภาพที่ 4.2 แสดงการตัดชุด binary ของชุดตัวอักษร 2 มิติ ออกมาทีละ 12 บิต

4.1.2 ตัวอักษร 3 มิติ

เมื่อได้ตัวอักษร 2 มิติที่เป็น binary มาแล้ว ก็จะทำการแปลงให้เป็นตัวอักษร 3 มิติ จะได้อักษร 3 มิติ ดังแสดงในภาพที่ 4.3



ภาพที่ 4.3 แสดงตัวอักษร 3 มิติ แบบกระพริบ (Flashing)



ภาพที่ 4.3 แสดงตัวอักษร 3 มิติ แบบกระพริบ (Flashing) (ต่อ)

4.2 การแสดงผลแบบเลื่อนในแนวนอน (Horizontal)

การแสดงผล 3 มิติ ในลักษณะของการเลื่อนในแนวนอนโดยผ่านแบบจำลอง เมื่อรับข้อมูลตัวอักษร 2 มิติ มาแปลงเป็น 3 มิติแล้ว ถ้าผู้ใช้กำหนด Transition เป็น Horizontal ข้อความจะแสดงเป็นชุดตัวอักษรที่ผู้ใช้ป้อนเข้ามาในโปรแกรม โดยตัวอักษรจะเลื่อนในแนวนอนผ่านหน้าต่างของ GUI ไปทีละบิต เลื่อนบิตทั้งหมด จำนวนบิตทั้งหมดของชุดตัวอักษรที่ผู้ใช้ป้อนเข้ามาลบด้วย 8 ครั้ง ซึ่งทางคณะผู้จัดทำได้นำตัวอย่างการแสดงผลแบบเลื่อนในแนวนอน โดยใช้ข้อความในตัวอย่างว่า “วิศวกร” ซึ่งใช้เวลาในการแสดงตัวอักษรบิตละ 0.1 วินาที

4.2.1 ตัวอักษร 2 มิติ

เมื่อทำการป้อนข้อมูลตัวอักษรเข้ามาในโปรแกรม ซึ่งในตัวอย่างคือ ตัวอักษร ‘วิ’, ‘ศ’, ‘ว’, ‘ก’ และ ‘ร’ ชุด binary ของตัวอักษร 2 มิติที่ได้มา จะถูกจัดเรียงแบบ queues (abstract data type) และได้ตัดบิต 2 คอลัมน์ด้านหน้าและ 2 คอลัมน์ด้านหลังที่เป็นบิต 0 ทั้งหมดออก เพื่อให้ระยะห่างระหว่างตัวอักษรไม่กว้างมากเกินไป ดังภาพที่ 4.4

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0
0 0 1 1 1 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0
0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

ภาพที่ 4.4 แสดงชุด binary ของตัวอักษร 2 มิติ แบบ Horizontal

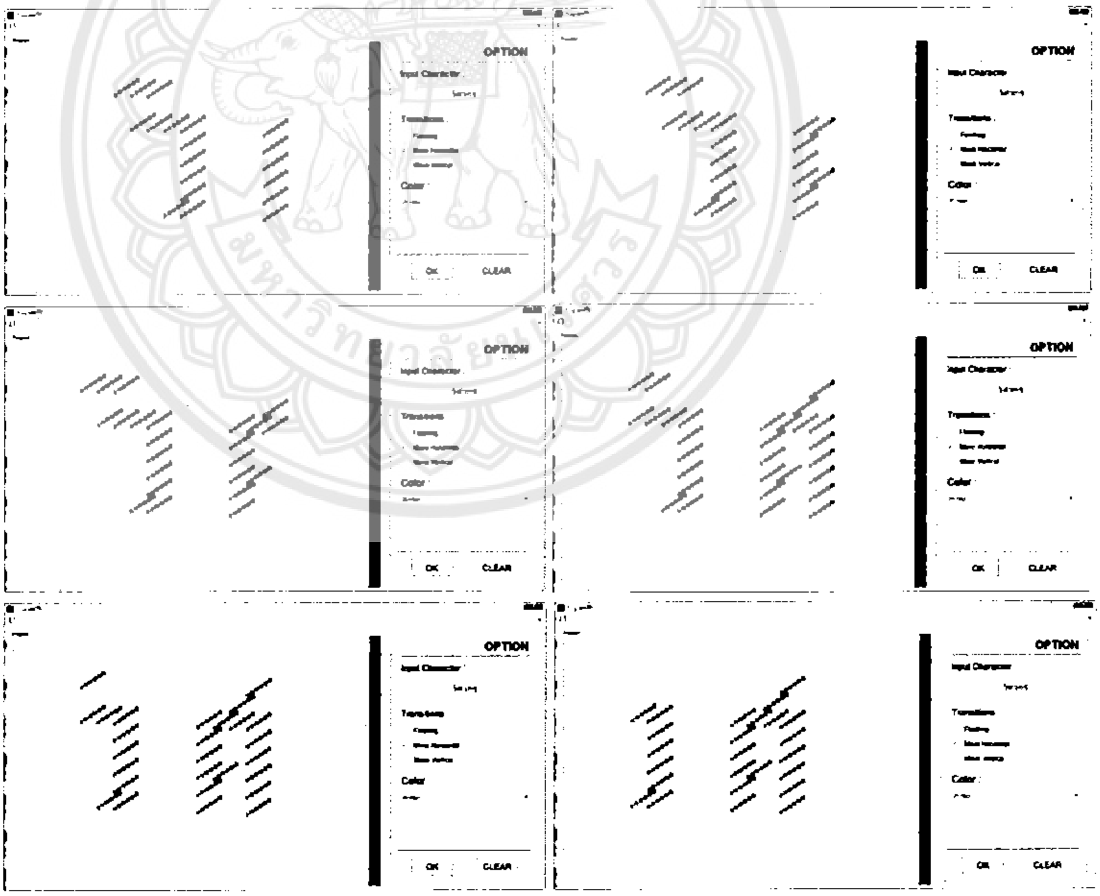
จากนั้นทำการเลื่อนบิตไปทางด้านซ้าย 9 บิต เพื่อให้ตัวอักษรที่เก็บไว้ลำดับแรกสุดแสดงก่อน จะได้ชุด binary ของตัวอักษร 2 มิติ ดังภาพที่ 4.5

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 1
1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0
1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0
1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

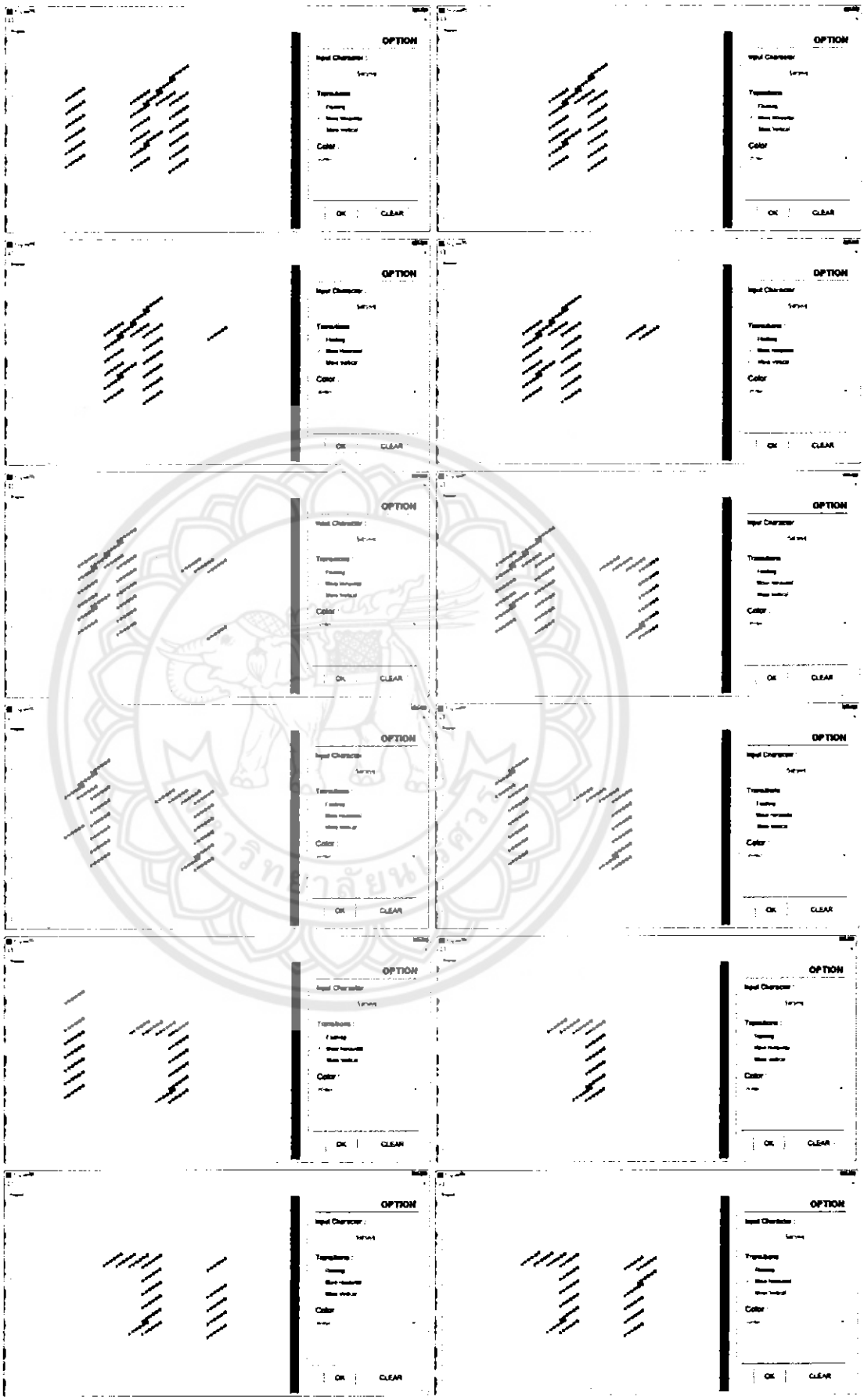
ภาพที่ 4.5 แสดงชุด binary ของชุดตัวอักษร 2 มิติ ที่ทำการเลื่อนบิตไปทางด้านซ้าย 9 บิต

4.2.2 ตัวอักษร 3 มิติ

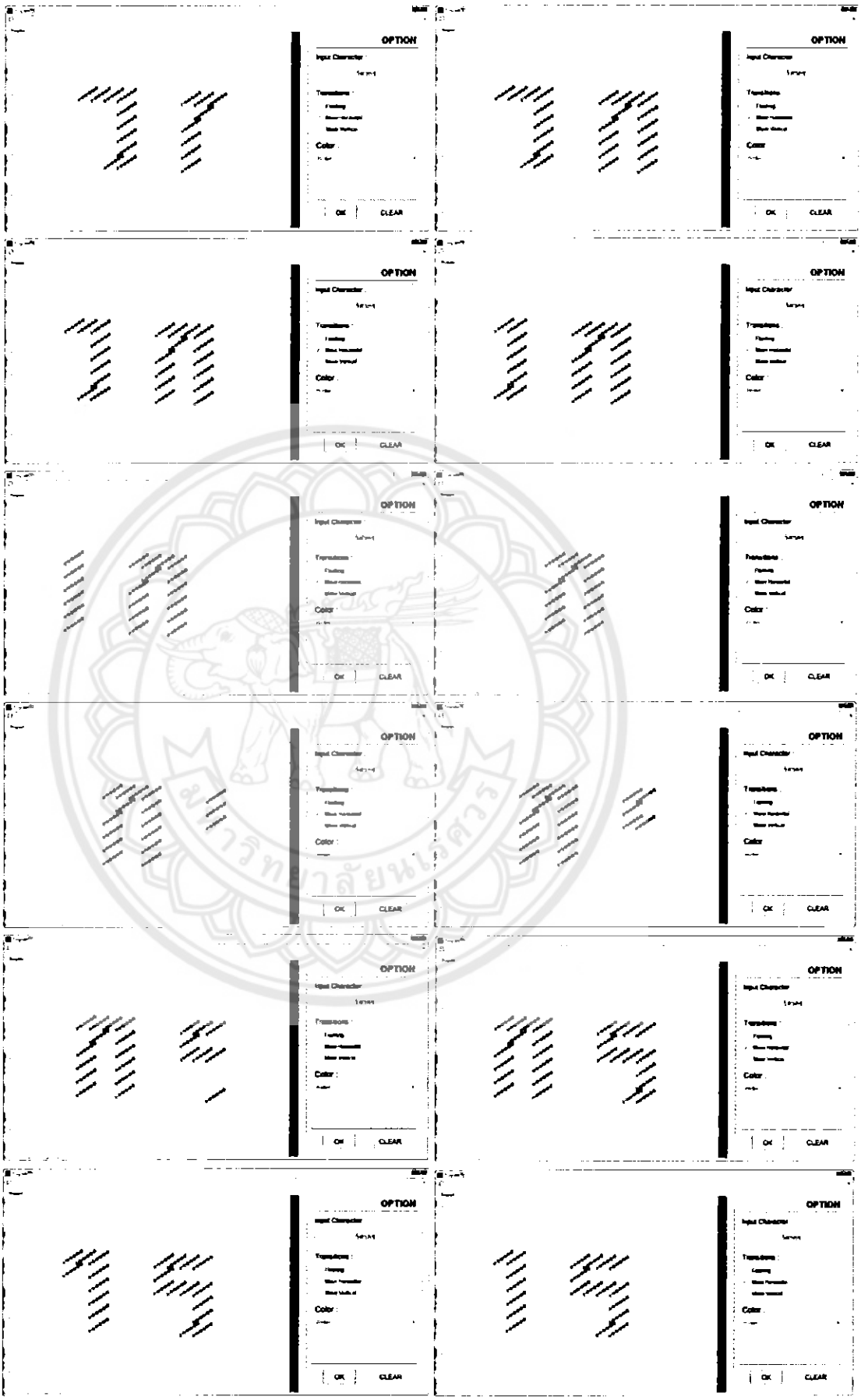
เมื่อได้ตัวอักษร 2 มิติที่เป็น binary มาแล้ว ก็จะทำการแปลงให้เป็น**ตัวอักษร 3 มิติ จะได้**อักษร 3 มิติ ดังแสดงในภาพที่ 4.6



ภาพที่ 4.6 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวนอน (Horizontal)



ภาพที่ 4.6 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวนอน (Horizontal) (ต่อ)



ภาพที่ 4.6 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวนอน (Horizontal) (ต่อ)

4.3 การแสดงผลแบบเลื่อนในแนวตั้ง (Vertical)

การแสดงผล 3 มิติ ในลักษณะของการเลื่อนในแนวตั้ง โดยผ่านแบบจำลอง เมื่อรับข้อมูลตัวอักษร 2 มิติ มาแปลงเป็น 3 มิติแล้ว ถ้าผู้ใช้กำหนด Transition เป็น Vertical ข้อความจะแสดงเป็นชุดตัวอักษรที่ผู้ใช้ป้อนเข้ามาในโปรแกรม โดยตัวอักษรจะเลื่อนในแนวตั้งผ่านหน้าต่างของ GUI ไปทีละบิต เลื่อนบิตทั้งหมด จำนวนบิตทั้งหมดของชุดตัวอักษรที่ผู้ใช้ป้อนเข้ามาลบด้วย 8 ครั้ง ซึ่งทางคณะผู้จัดทำได้นำตัวอย่างการแสดงผลแบบเลื่อนในแนวตั้งโดยใช้ข้อความในตัวอย่างว่า "0๑2๓" ซึ่งใช้เวลาในการแสดงตัวอักษรบิตละ 0.1 วินาที

4.3.1 ตัวอักษร 2 มิติ

เมื่อทำการป้อนข้อมูลตัวอักษรเข้ามาในโปรแกรม ซึ่งในตัวอย่างคือ ตัวอักษร '0', '๑', '2' และ '๓' ชุด binary ของตัวอักษร 2 มิติที่ได้มา จะถูกจัดเรียงแบบ queues (abstract data type) และได้ตัดบิต 2 แถวด้านบนและ 2 แถวด้านล่างที่เป็นบิต 0 ทั้งหมดออก ดังภาพที่ 4.7

```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 1 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 1 0 1 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 1 0 0 1 1 0 0 0 0
0 0 0 1 0 1 0 1 0 0 0 0
0 0 0 1 1 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
```

ภาพที่ 4.7 แสดงชุด binary ของตัวอักษร 2 มิติ แบบ Vertical

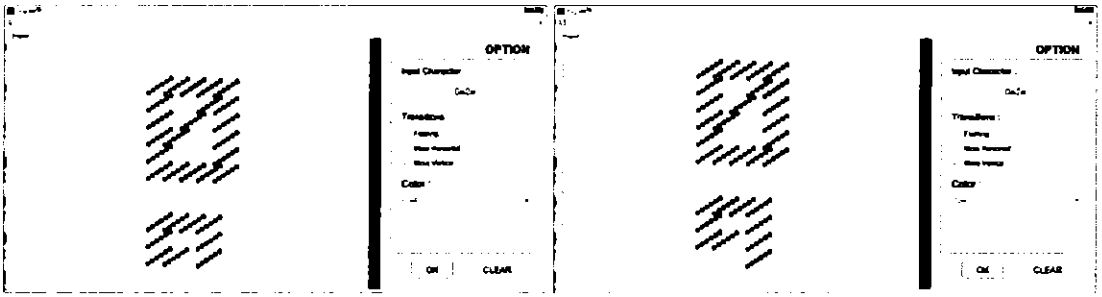
จากนั้นทำการเลื่อนบิตไปทางด้านบน 10 บิต เพื่อให้ตัวอักษรที่เก็บไว้ลำดับแรกสุดแสดงก่อน จะได้ชุด binary ของตัวอักษร 2 มิติ ดังภาพที่ 4.8

```
0 0 0 0 1 0 1 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 1 0 0 1 1 0 0 0 0
0 0 0 1 0 1 0 1 0 0 0 0
0 0 0 1 1 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 1 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 0 1 0 0 1 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0
```

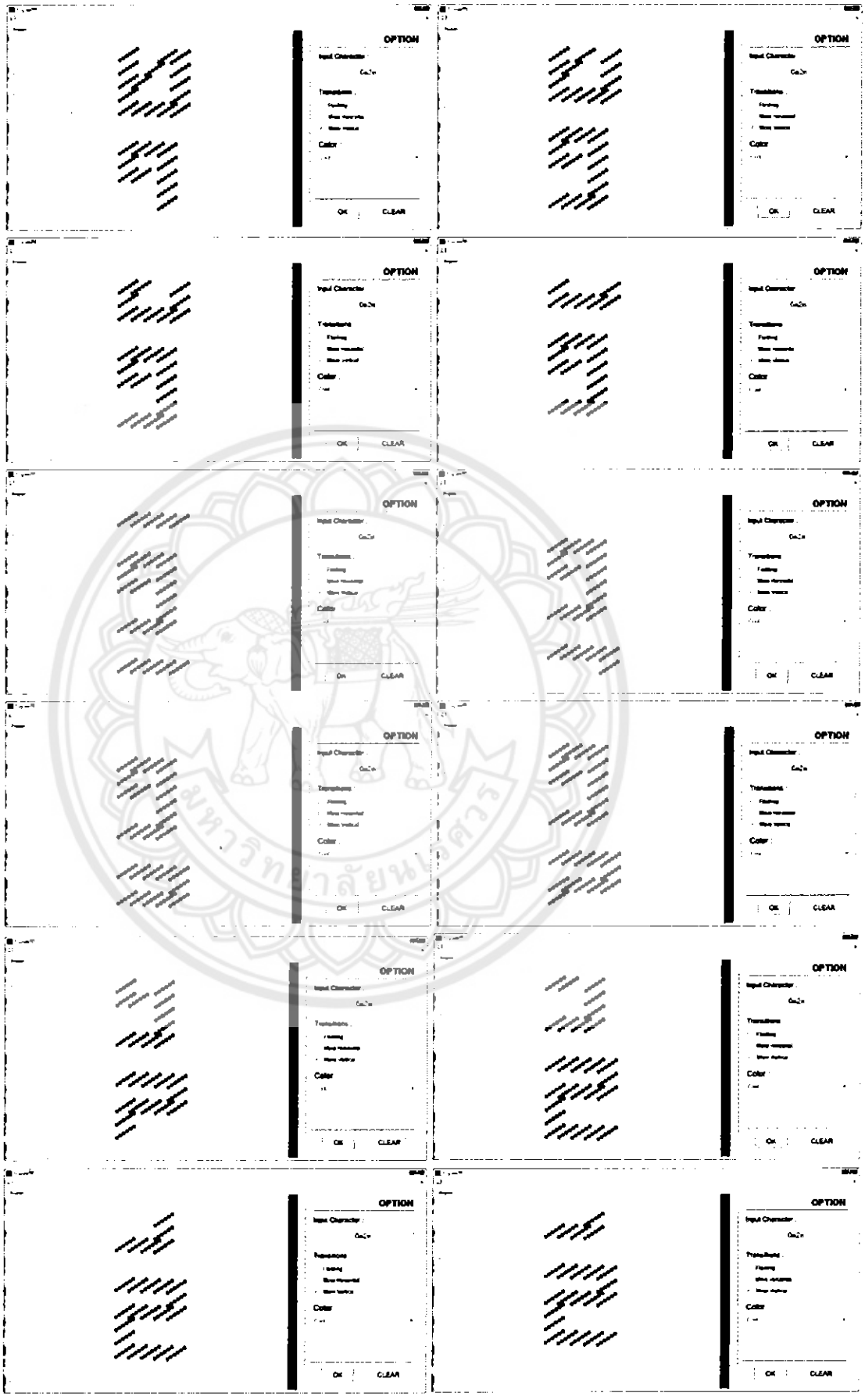
ภาพที่ 4.8 แสดงชุด binary ของตัวอักษร 2 มิติ ที่ทำการเลื่อนบิตไปทางด้านบน 10 บิต

4.2.2 ตัวอักษร 3 มิติ

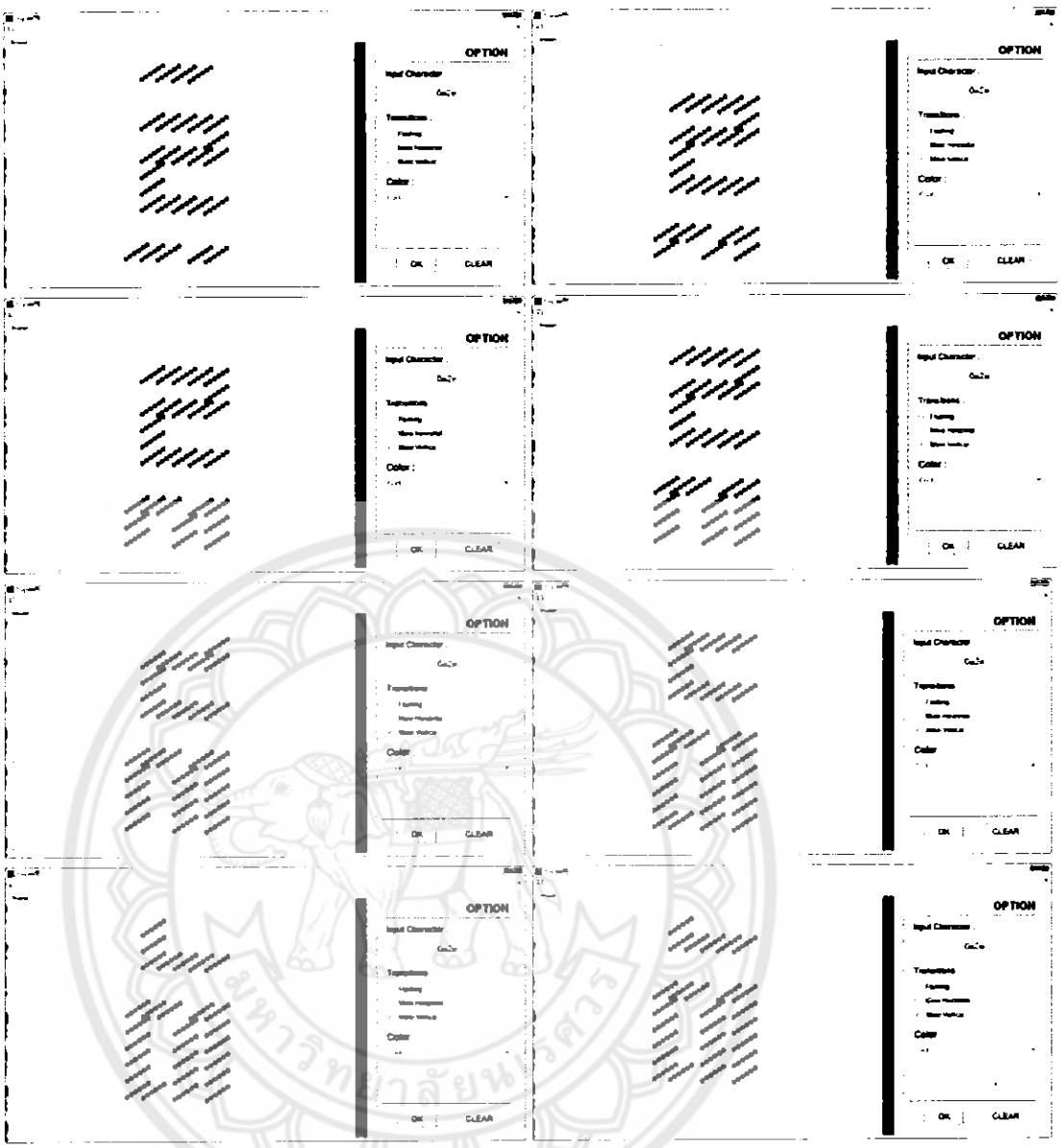
เมื่อได้ตัวอักษร 2 มิติที่เป็น binary มาแล้ว ก็จะทำการแปลงให้เป็นตัวอักษร 3 มิติ จะได้อักษร 3 มิติ ดังแสดงในภาพที่ 4.9



ภาพที่ 4.9 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวตั้ง (Vertical)



ภาพที่ 4.9 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวตั้ง (Vertical) (ต่อ)



ภาพที่ 4.9 แสดงตัวอักษร 3 มิติ แบบเลื่อนในแนวตั้ง (Vertical) (ต่อ)

บทที่ 5
บทสรุป

เนื้อหาในบทนี้เป็นส่วนที่สรุปผลการดำเนินงานทั้งหมด ได้แก่ ปัญหาที่พบระหว่างการดำเนินงาน ข้อเสนอแนะในการทำโครงงาน และแนวทางในการประยุกต์พัฒนาโครงงานต่อไปซึ่งรายละเอียดมีดังนี้

5.1 บทสรุปการดำเนินงาน

จากการดำเนินงานทางคณะผู้จัดทำได้ออกแบบโปรแกรมและแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ขนาด 12x12x12 ดวง เพื่อใช้เป็นตัวจำลองภาพในการแสดงผลการแปลงอักษร 2 มิติให้กลายเป็นอักษร 3 มิติ ซึ่งตัวอักษรนั้นมีทั้งตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก ตัวพยัญชนะ สระ และวรรณยุกต์ภาษาไทย เลขอาราบิกและเลขไทย ในการแสดงผลจะสามารถแสดงได้ไม่จำกัดเวลา และจำนวนตัวอักษร โดยตัวอักษรจะสามารถเลื่อนได้ทั้งแนวตั้งและแนวนอน โดยตัวอักษรจะทำงานในระบบสี RGB, CMYK และ สีผสม สามารถเลือกสีในการแสดงผลได้ครั้งละสีและไม่สามารถผสมสีได้ภายในประโยคเดียวกัน

เมื่อนำแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของจริงที่เป็นฮาร์ดแวร์กับแบบจำลองไดโอเปล่งแสงทรงลูกบาศก์ในโปรแกรมมาพิจารณาในแง่มุมต่างๆเปรียบเทียบกันจะเกิดความแตกต่างเช่นตัวอย่างในตารางที่ 5.1 นี้

ตารางที่ 5.1 การเปรียบเทียบระหว่างแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของจริง (ฮาร์ดแวร์) และแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของโปรแกรม (ซอฟต์แวร์)

ลักษณะ	แบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของจริง (ฮาร์ดแวร์)	แบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของโปรแกรม (ซอฟต์แวร์)
สีของ LED	จะเป็น LED สีเดียว	จะสามารถเปลี่ยนสีได้โดยไม่จำกัดสีและมีความละเอียดของสีมากกว่า

ตารางที่ 5.1 การเปรียบเทียบระหว่างแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของจริง (ฮาร์ดแวร์) และแบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ของโปรแกรม (ซอฟต์แวร์) (ต่อ)

ขนาดของแบบจำลอง	มีข้อจำกัดในการเพิ่มขนาดของแบบจำลอง	สามารถเพิ่มขนาดของแบบจำลองได้ตามที่ต้องการ
ความสว่างของหลอด LED	มีความสว่างของแสงในหลอด LEDมากกว่า	มีความสว่างของแสงที่แสดงผลออกมาน้อย

5.2 ปัญหาที่พบ

- ในการประมวลผลถ้าใช้แบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ขนาด 8x8x8 รายละเอียดของตัวอักษรจะน้อยแต่การประมวลผลจะเร็ว แต่ถ้าใช้แบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ขนาด 16x16x16 รายละเอียดของตัวอักษรจะเพิ่มมากขึ้นแต่การประมวลผลตัวอักษรจะช้ามาก จึงใช้แบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ขนาด 12x12x12 ซึ่งให้รายละเอียดของอักษรชัดเจนกว่า 8x8x8 และประมวลผลเร็วกว่า 16x16x16
- การใส่แสงในแบบจำลองให้สว่างขึ้นทำให้โปรแกรมประมวลผลช้าลงมาก จึงไม่สามารถใส่แสงเพื่อเพิ่มความสว่างแก่แบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ได้
- เมื่อมีอินพุตที่เป็นสระซ้อนเข้ามาโปรแกรมจะยังไม่สามารถแสดงสระซ้อนได้
- โปรแกรมไม่สามารถจัดการเมื่อมีข้อมูลอักษรทั้งด้านบนและด้านล่างเข้ามาพร้อมกัน เช่น สระในคำว่า พันธุ์ โปรแกรมจะประมวลผลเฉพาะสระ อุ แต่จะไม่สามารถประมวลผลตัวการันต์ได้ และจะหยุดการวิเคราะห์สระด้านบนและสระด้านล่าง เช่น คาว่า พันธุ์ดุ จะไม่สามารถวิเคราะห์สระ อุ ในคำว่า ดุ ได้

5.3 ข้อเสนอแนะ

- แบบจำลองไดโอดเปล่งแสงทรงลูกบาศก์ควรเพิ่ม - ลดขนาดได้ตามความต้องการของผู้ใช้เพื่อความเหมาะสมของข้อความ
- ระบบสีที่ใช้สามารถกำหนดโทนสีอ่อน – เข้ม มีความยืดหยุ่นในการเลือกสีมากขึ้น เพื่อการสร้างสรรค์ตัวอักษรที่หลากหลายมาขึ้น
- ทำให้เกิดความสว่าง โปร่งแสงของหลอดไดโอดเปล่งแสง เพื่อให้มีมุม แสงเงาทำให้ข้อความมีความลึก ตื้น ดูมีมิติขึ้น

- ควรมีการเพิ่มการวิเคราะห์สระที่ซับซ้อนกว่าเดิม เช่น สระในคำว่า พันธุ์ และสระซ้อน เช่น สระในคำว่า วิ่ง

5.4 แนวทางในการประยุกต์และพัฒนาต่อไป

5.4.1 แนวทางในการประยุกต์

สามารถนำข้อความอักษร 3 มิติที่ได้มาประยุกต์ใช้ทำแอนิเมชั่นเป็นข้อความวิ่ง 3 มิติ หรือนำไปประยุกต์ใช้กับโฆษณาบนเว็บเพื่อให้ดูน่าสนใจมากยิ่งขึ้น ซึ่งแบบอักษรที่เลือกได้ทั้งภาษาอังกฤษ ภาษาไทย ทำให้มีตัวเลือกในการทำข้อความที่หลากหลาย และระบบสีที่ใช้เป็นสี RGB, CMYK และสีผสมทำให้ข้อความดูมีสีสันมากเหมาะสำหรับนำไปตกแต่งหรือประยุกต์ใช้งานในสื่อ

5.4.2 แนวทางในการพัฒนาต่อไป

- สามารถเพิ่มเทคนิคการแสดงผลของตัวอักษรได้ เช่น การทำให้ตัวอักษรหมุน
- สามารถเพิ่มระบบสีและระดับสีที่หลากหลาย
- เพิ่มชนิดของอักษรเป็นตัวอักษรภาษาอื่นๆได้ เช่น ภาษาจีน ภาษาเกาหลี ภาษาญี่ปุ่น เพื่อให้ข้อความมีความหลากหลายในการสื่อสารต่างๆ
- เพิ่มฟอนต์ของตัวอักษรให้หลากหลายมากยิ่งขึ้น
- เพิ่มสระซ้อน เช่น วิ่ง เนื่องจากขนาดของไอโอดเปล่งแสงทรงลูกบาศก์มีพื้นที่ไม่มากพอในการแสดงจึงแสดงผลสระซ้อนได้ลำบาก

เอกสารอ้างอิง

[1] **Introduction To The RGB Color System.** Retrieved June 26, 2013, from http://www.web-colors-explained.com/rgb_01.php

[2] ColoRotate. (2002). **Color Models.** Retrieved June 26, 2013, from http://learn.colorotate.org/color-models.html

[3] **Bock,** Retrieved July 26, 2013, from http://www.f0nt.com/release/block/

[4] [5] **ส่วนประกอบของตัวอักษร,** Retrieved July 26, 2013, from http://typographyfordesign.blogspot.com/p/blog-pase.html

[6] Seksun Sartsatit. **Embedded System Technology Laboratory.** Retrieved December 7, 2013, from http://www.nectec.or.th/index.php/2011-07-12-03-00-06/2011-07-12-03-01-25/-research-unit-ru/2011-05-18-02-45-04/2011-07-01-12-33-39.html

[7] Srantonio. (2012). **FiO Std.** Retrieved December 7, 2013, from https://www.aimagin.com/fio-std.html

[8] **LED Cube Displays 3d Image in Color.** Retrieved December 23, 2013, from http://technabob.com/blog/2008/04/06/led-3d-cube-color-display/

ภาคผนวก ก.

ตัวอย่าง Code การทำงานของโปรแกรม

1. Code โปรแกรมออกแบบและจำลองการแปรอักษร 3 มิติ

```
if isequal(unicode(m), 209) || isequal(unicode(m), 211) || ...
     isequal(unicode(m), 212) || isequal(unicode(m), 213) || ...
     isequal(unicode(m), 214) || isequal(unicode(m), 215) || ...
     isequal(unicode(m), 232) || isequal(unicode(m), 233) || ...
     isequal(unicode(m), 234) || isequal(unicode(m), 235) || ...
     isequal(unicode(m), 236)

        strvowel = [strvowel unicode(m)];
        index = [index m];

elseif isequal(unicode(m), 216) || isequal(unicode(m), 217)
        strvowel = [strvowel unicode(m)];
        index = [index m];
end
```

ภาพที่ ผ.1 ตัวอย่าง code การตรวจสอบสระ

```
      switch str
      %LEDThaiConsonant
        case 161 %ก
            LED = fliplr (flipud(U0E01));
        case 162 %ข
            LED = fliplr (flipud(U0E02));
        case 163 %ฃ
            LED = fliplr (flipud(U0E03));
        case 164 %ค
            LED = fliplr (flipud(U0E04));
        case 165 %ฅ
            LED = fliplr (flipud(U0E05));
                .
                .
      %LEDEngConsonant
        case 65 %A;
            LED = fliplr (flipud(U0041));
        case 66 %B
            LED = fliplr (flipud(U0042));
        case 67 %C
            LED = fliplr (flipud(U0043));
        case 68 %D
            LED = fliplr (flipud(U0044));
        case 69 %E
            LED = fliplr (flipud(U0045));
                .
                .
```

ภาพที่ ผ.2 ตัวอย่าง code การตรวจสอบชนิดของตัวอักษร

```
    %EngSmall
      case 97 %a
          LED = fliplr (flipud(U0061));
      case 98 %b
          LED = fliplr (flipud(U0062));
      case 99 %c
          LED = fliplr (flipud(U0063));
      case 100 %d
          LED = fliplr (flipud(U0064));
      case 101 %e
          LED = fliplr (flipud(U0065));
                .
                .
    %LEDNumber
      case 48 %0
          LED = fliplr (flipud(U0030));
      case 49 %1
          LED = fliplr (flipud(U0031));
      case 50 %2
          LED = fliplr (flipud(U0032));
      case 51 %3
          LED = fliplr (flipud(U0033));
      case 52 %4
          LED = fliplr (flipud(U0034));
                .
                .
    %NumberThai
      case 240 %๐
          LED = fliplr (flipud(U0E50));
      case 241 %๑
          LED = fliplr (flipud(U0E51));
      case 242 %๒
          LED = fliplr (flipud(U0E52));
      case 243 %๓
          LED = fliplr (flipud(U0E53));
      case 244 %๔
          LED = fliplr (flipud(U0E54));
                .
                .
    %LEDVowel
      case 207 %ๆ
          LED = fliplr (flipud(U0E2F));
      case 208 %ะ
          LED = fliplr (flipud(U0E30));
      case 210 %ิ
          LED = fliplr (flipud(U0E32));
      case 224 %เ
          LED = fliplr (flipud(U0E40));
      case 225 %แ
          LED = fliplr (flipud(U0E41));
                .
                .
end
```

ภาพที่ ผ.2 ตัวอย่าง code การตรวจสอบชนิดของตัวอักษร (ต่อ)

```
switch strvowel(iii)
      case 209 %ั
          LEDvowel = fliplr (flipud(U0E31));
          LED = LED|LEDvowel;
      case 211 %ิ
          LEDvowel = fliplr (flipud(U0E33));
          LED = LED|LEDvowel;
      case 212 %ี
          LEDvowel = fliplr (flipud(U0E34));
          LED = LED|LEDvowel;
      case 213 %ึ
          LEDvowel = fliplr (flipud(U0E35));
          LED = LED|LEDvowel;
      case 214 %ื
          LEDvowel = fliplr (flipud(U0E36));
          LED = LED|LEDvowel;
      case 215 %ุ
          LEDvowel = fliplr (flipud(U0E37));
          LED = LED|LEDvowel;
      case 216 %ู
          LEDvowel = fliplr (flipud(U0E38));
          LED = LED|LEDvowel;
      case 217 %ฺ
          LEDvowel = fliplr (flipud(U0E39));
          LED = LED|LEDvowel;
      case 232 %่
          LEDvowel = fliplr (flipud(U0E48));
          LED = LED|LEDvowel;
      case 233 %้
          LEDvowel = fliplr (flipud(U0E49));
          LED = LED|LEDvowel;
      case 234 %๊
          LEDvowel = fliplr (flipud(U0E4A));
          LED = LED|LEDvowel;
      case 235 %๋
          LEDvowel = fliplr (flipud(U0E4B));
          LED = LED|LEDvowel;
      case 236 %์
          LEDvowel = fliplr (flipud(U0E4C));
          LED = LED|LEDvowel;
  end
```

ภาพที่ ผ.3 ตัวอย่าง code การนำพยัญชนะและสระในภาษาไทยมารวมกัน

```
% Display 3D
   x=0.5:11.5; y=x; z=x;
   S=100;
   C=[]; xx=[]; yy=[]; zz=[];

   for ii=1:12
       for jj=1:12
           for kk=1:12
              if ( (showdata(kk,jj)==1) && ( (ii==4) || (ii==5)
|| (ii==6) || (ii==7) || (ii==8) || (ii==9)) )

                  xx=[xx x(ii)]; yy=[yy y(jj)];  zz=[zz z(kk)];
                  C=[C+1; 1];

               end
           end
       end
   end

   scatter3(xx,yy,zz,S,C,'filled'); grid off;
   daspect([1 1 1]); view(3); axis equal;
   axis([0 12 0 12 0 12]);
   xlabel('x');ylabel('y');zlabel('z');
```

ภาพที่ ผ.4 ตัวอย่าง code การแปลงตัวอักษร 2 มิติเป็นตัวอักษร 3 มิติ

```
alpha(0.3); nlevel = 2^8;

none = [ones(1,nlevel);ones(1,nlevel);ones(1,nlevel)]';
red = [ones(1,nlevel);zeros(1,nlevel);zeros(1,nlevel)]';
green= [zeros(1,nlevel);ones(1,nlevel);zeros(1,nlevel)]';
blue = [zeros(1,nlevel);zeros(1,nlevel);ones(1,nlevel)]';

cyan = [linspace(0,1,nlevel);ones(1,nlevel);ones(1,nlevel)]';
magenta = [ones(1,nlevel);linspace(0,1,nlevel);ones(1,nlevel)]';
yellow = [ones(1,nlevel);ones(1,nlevel);linspace(0,1,nlevel)]';
black = [zeros(1,nlevel);zeros(1,nlevel);zeros(1,nlevel)]';

hot = [ones(1,nlevel);linspace(0,1,nlevel);zeros(1,nlevel)]';
summer = [linspace(0,1,nlevel);ones(1,nlevel);zeros(1,nlevel)]';
winter = [zeros(1,nlevel);ones(1,nlevel);linspace(0,1,nlevel)]';
cool = [linspace(0,1,nlevel);zeros(1,nlevel);ones(1,nlevel)]';
```

ภาพที่ ผ.5 ตัวอย่าง code สี RGB

```
if inputcolor == 1 %None
    colormap(none);
elseif inputcolor == 2 %Red
    colormap(red);
elseif inputcolor == 3 %Green
    colormap(green);
elseif inputcolor == 4 %Blue
    colormap(blue);
elseif inputcolor == 5 %Cyan
    colormap(cyan);
elseif inputcolor == 6 %Magenta
    colormap(magenta);
elseif inputcolor == 7 %Yellow
    colormap(yellow);
elseif inputcolor == 8 %Black
    colormap(black);
elseif inputcolor == 9 %Hot
    colormap(hot);
elseif inputcolor == 10 %Summer
    colormap(summer);
elseif inputcolor == 11 %Winter
    colormap(winter);
elseif inputcolor == 12 %Cool
    colormap(cool);
end
```

ภาพที่ ผ.6 ตัวอย่าง code การเรียกใช้สี

```
if inputThickness == 1
    Thickness = ((ii==1));
elseif inputThickness == 2
    Thickness = ((ii==1) || (ii==2));
elseif inputThickness == 3
    Thickness = ((ii==1) || (ii==2) || (ii==3));
elseif inputThickness == 4
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4));
elseif inputThickness == 5
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5));
elseif inputThickness == 6
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5) || (ii==6));
elseif inputThickness == 7
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5) || (ii==6) || (ii==7));
elseif inputThickness == 8
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5) || (ii==6) || (ii==7) || (ii==8));
elseif inputThickness == 9
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5) || (ii==6) || (ii==7) || (ii==8) ||
                (ii==9));
```

ภาพที่ ผ.7 ตัวอย่าง code การเลือกความหนาของตัวอักษร

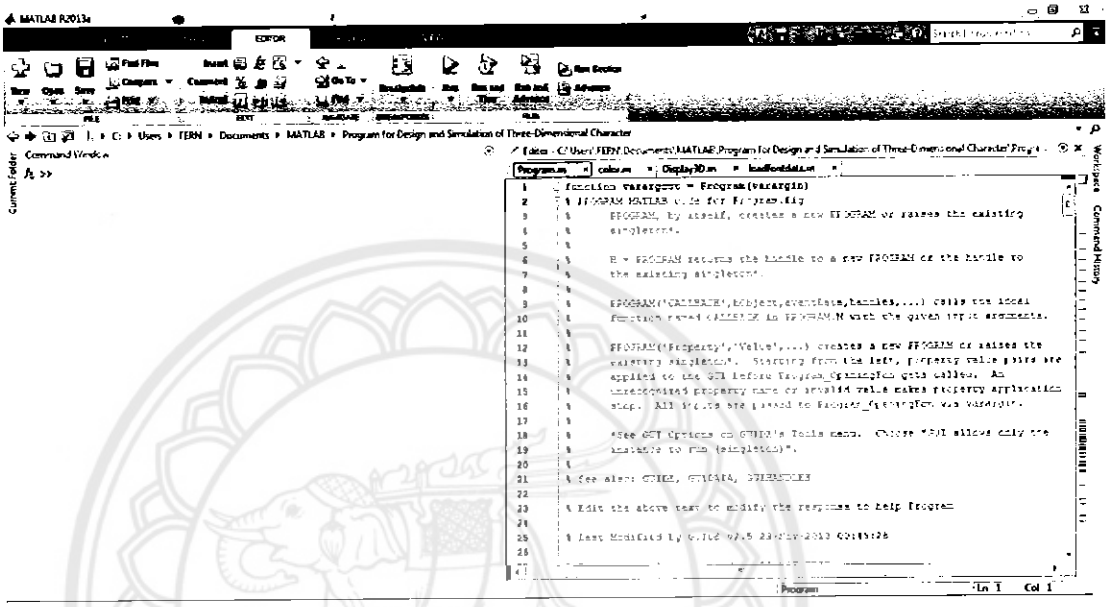
```
elseif inputThickness == 10
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5) || (ii==6) || (ii==7) || (ii==8) ||
                (ii==9) || (ii==10));
elseif inputThickness == 11
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5) || (ii==6) || (ii==7) || (ii==8) ||
                (ii==9) || (ii==10) || (ii==11));
elseif inputThickness == 12
    Thickness = ((ii==1) || (ii==2) || (ii==3) || (ii==4) ||
                (ii==5) || (ii==6) || (ii==7) || (ii==8) ||
                (ii==9) || (ii==10) || (ii==11) || (ii==12));
end
```

ภาพที่ ผ.7 ตัวอย่าง code การเลือกความหนาของตัวอักษร (ต่อ)

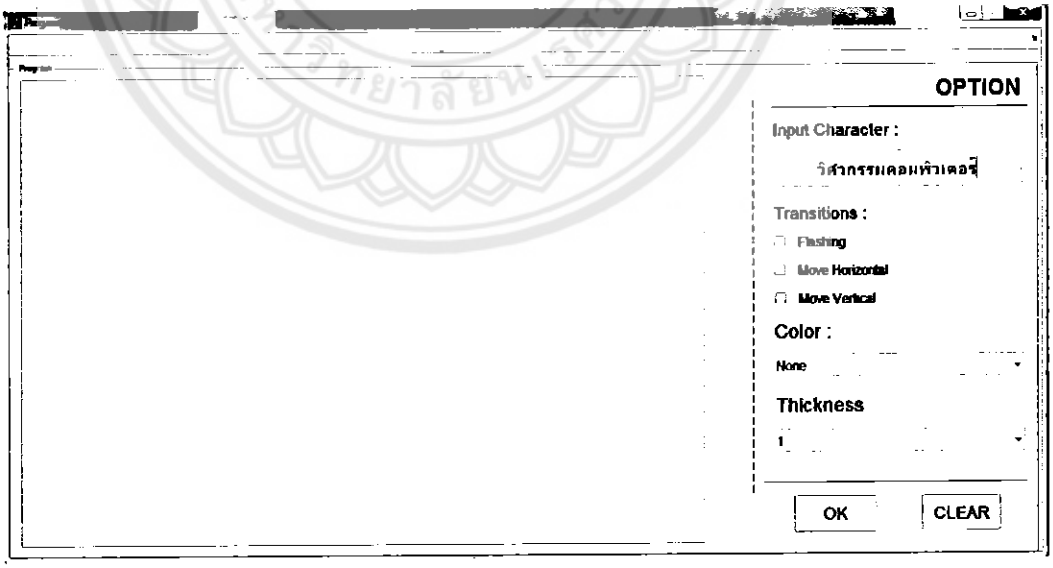
ภาคผนวก ข.

การใช้งานโปรแกรม

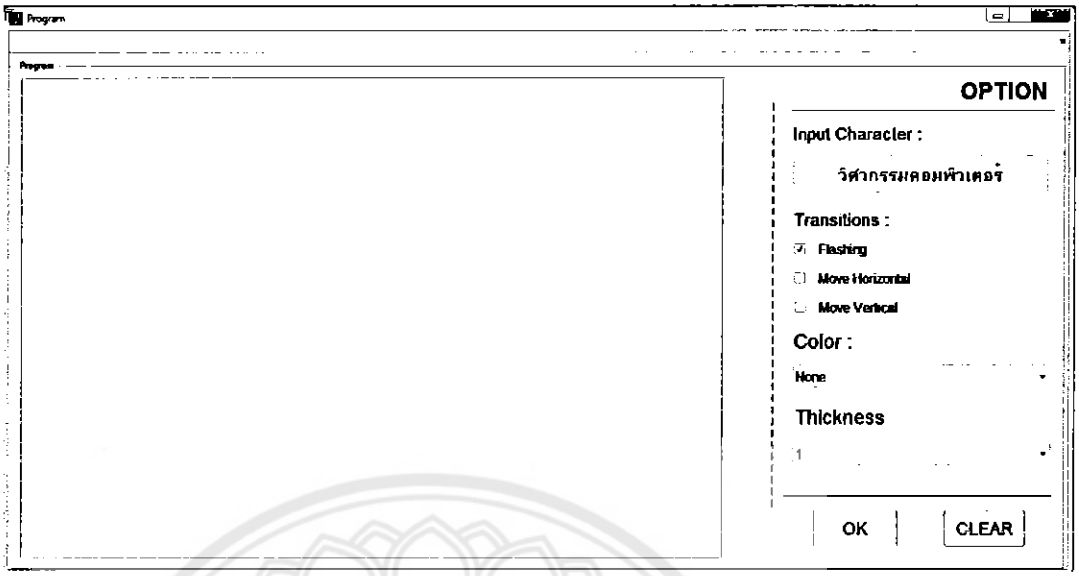
1. เปิดโปรแกรมแมตแล็บขึ้นมา และทำการรันโปรแกรม



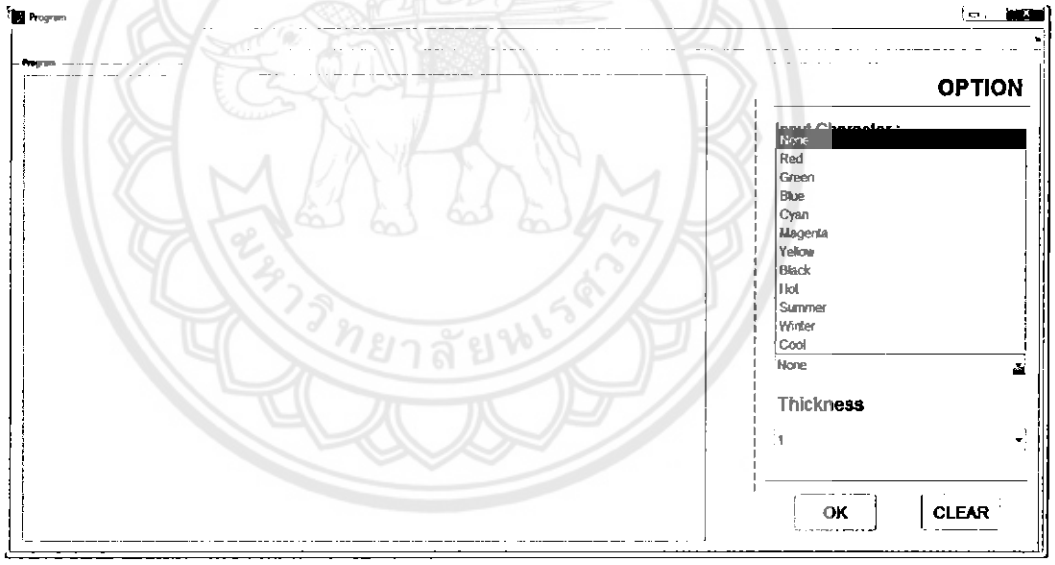
2.ป้อนข้อมูลตัวอักษรที่ต้องการแสดง



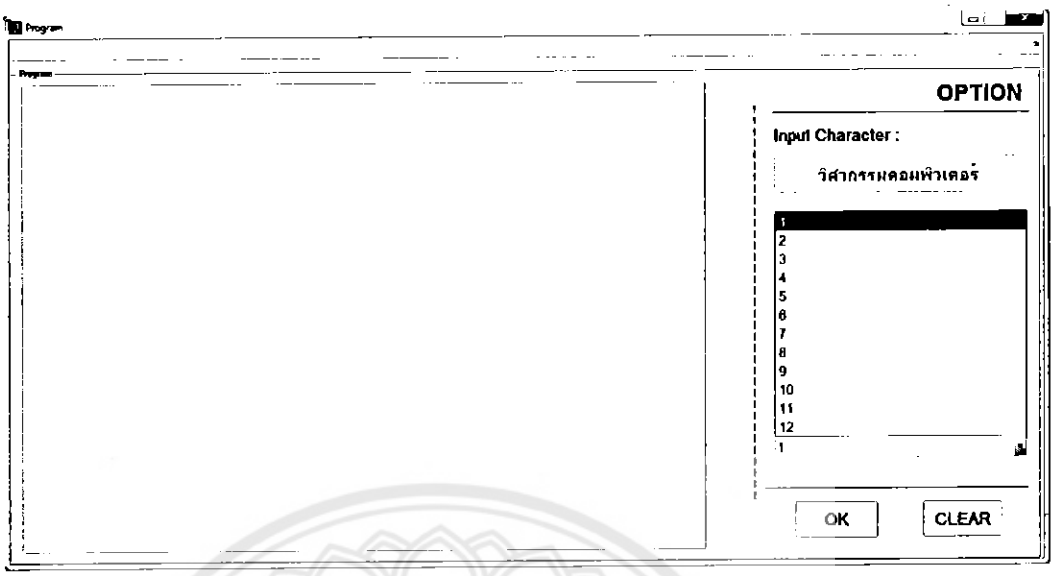
3.เลือกรูปแบบการแสดงผล



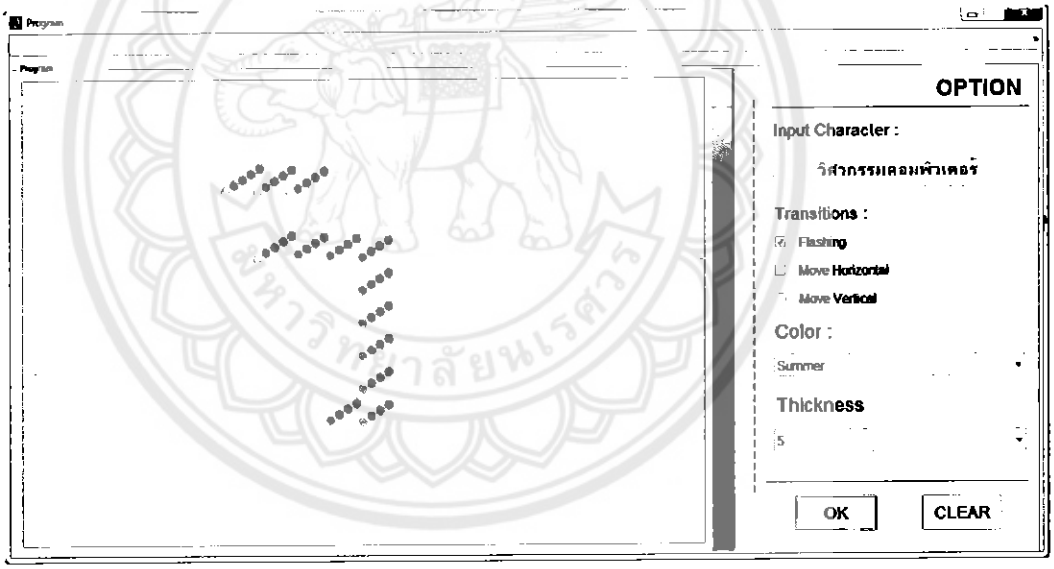
4.เลือกสีสำหรับชุดตัวอักษร



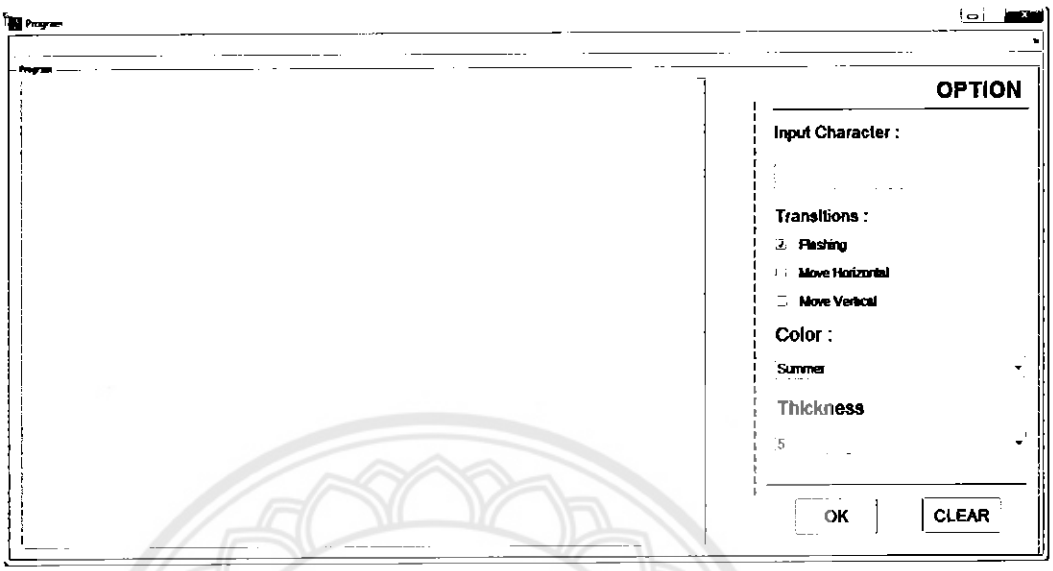
5.เลือกความหนาของตัวอักษร



6.จากนั้นคลิกที่ปุ่ม "OK"



7.หากต้องการป้อนชุดตัวอักษรชุดใหม่ สามารถคลิกที่ปุ่ม “CLEAR” เพื่อลบชุดตัวอักษรชุดเก่าออกไป



หรือหากต้องการจบโปรแกรม สามารถคลิกที่ X ตรงมุมบนด้านขวามือได้เลย