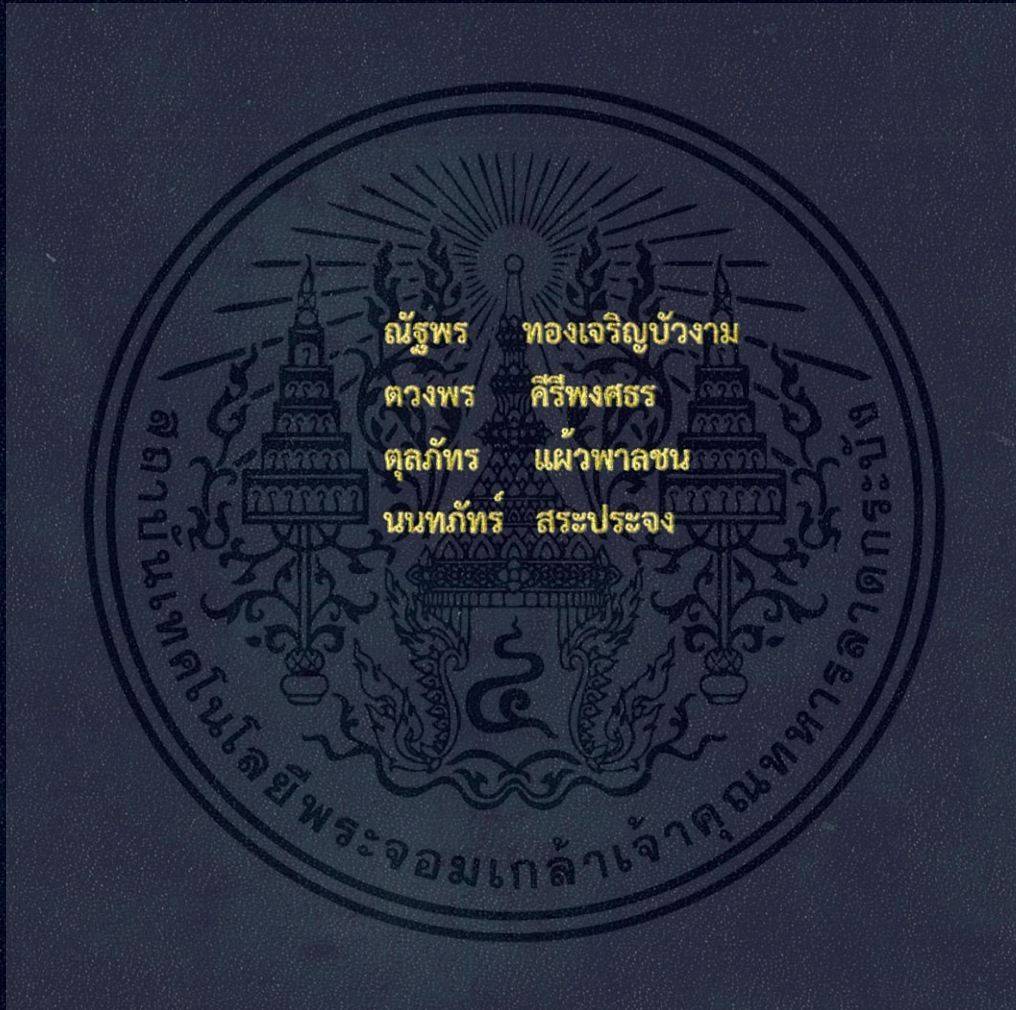


โครงการ บอร์ดแสดงผลผ่านหลอดแอลอีดี
LED Display Board



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

โครงการ บอร์ดแสดงผลผ่านหลอดแอลอีดี
LED Display Board



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LED Display Board



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปฏิญญานิพนธ์	โครงการ บอร์ดแสดงผลผ่านหลอดแอลอีดี	
นักศึกษาผู้จัดทำ	นางสาวณัฐพร	ทองเจริญบัวงาม รหัสนักศึกษา 54010426
	นางสาวดวงพร	ศิริพงศธร รหัสนักศึกษา 54010485
	นายตุลภัทร	แผ้วพาลชน รหัสนักศึกษา 54010488
	นายนนทภัทร์	สระประจง รหัสนักศึกษา 54010653
อาจารย์ที่ปรึกษา	ผศ.ดร.พงษ์ชัย นิลาศ	
ปีการศึกษา	2557	

บทคัดย่อ

โครงการบอร์ดแสดงผลผ่านหลอดแอลอีดี เป็นโครงการที่ได้ทำงานศึกษา ออกแบบ และประกอบ ฮาร์ดแวร์ของบอร์ดแอลอีดี โครงการนี้ได้ใช้ไมโครคอนโทรลเลอร์ PIC16F877A หลอดแอลอีดี 256 หลอด ชิพตรีจิสเตอร์ วงจรนับ ไทป์พีซีไอสายระบบปฏิบัติการแอนดรอยด์ พอร์ตอนุกรม RS232 และการนำเอาตัวเลขหรือตัวอักษรมาใช้ในการสื่อความหมายระหว่างผู้ใช้งานกับตัวโปรแกรม ซึ่งเป็นการเขียนโปรแกรม Microsoft Visual Studio 2010 ให้สามารถรับคำสั่ง รับภาพตัวอักษรจากกล้อง รับไฟล์รูปภาพตัวอักษรและรับตัวอักษรจากแป้นพิมพ์ได้ และได้ใช้การเขียนโปรแกรม Eclipse สำหรับสร้างแอปพลิเคชันแอนดรอยด์ แล้วส่งข้อมูลไปยัง PIC16F877A เพื่อควบคุมการแสดงผลบนบอร์ดแอลอีดี ไมโครคอนโทรลเลอร์ จะทำการรับข้อมูลผ่านพอร์ตอนุกรมจากคอมพิวเตอร์หรือโทรศัพท์ไร้สาย และแสดงผลบนบอร์ดแอลอีดี ในส่วนการทดลองนั้น สามารถรับภาพจากกล้อง ไฟล์รูปภาพ และแป้นพิมพ์ แล้วแสดงตัวเลข ตัวอักษรภาษาอังกฤษ และตัวอักษรภาษาไทยได้ และยังสามารถเลื่อนตัวอักษรบนบอร์ดแอลอีดีอย่างที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title LED Display Board
Authors Miss Nattaporn Thongjaroenbuangam
Miss Tuangporn Kerepongsatorn
Mr. Toonlapat Paaopanchon
Mr. Nontapat Sraprajong
Thesis Advisor Assit.Prof.Dr.Phongchai Nilas
Year 2014

ABSTRACT

This LED Display Board project focuses on study, design as well as hardware assembly the LED Display Board. The project employs PIC 16F877A microcontroller, 256 LEDs, Shift register, Counter, Smartphone with Android system, Serial port RS232 and a Graphical User Interface (GUI) program. The GUI is written in Microsoft visual Studio 2010 to receive commands, Alphabet image from camera, Alphabet Image from files, Alphabet from Keyboard. And building an Android application on Eclipse program for transmitted to PIC16F877A to control the LED Display Board. Microcontroller receives data via Serial Port from computer or Smartphone and display them on LED Display Board. The experiment results could receive image from camera, files, keyboard and show visible number, English alphabet, Thai alphabet and could scroлл alphabet on LED Display Board.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความเมตตาจากผู้ช่วยศาสตราจารย์ ดร.พงษ์ชัย นิลาศ ที่ได้ให้คำแนะนำและคำสอนแก่ผู้วิจัยปริญญาานิพนธ์ตลอดมา อีกทั้งยังเอื้อเฟื้อสถานที่ เครื่องมือต่างๆที่ใช้ในการทำปริญญาานิพนธ์ ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณพี่ธนาคุณ รัตนจรัสโรจน์ รุ่น 49 ที่ให้คำปรึกษาอันเป็นประโยชน์ และช่วยเหลือด้วยดีตลอดมา

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และเพื่อนอันเป็นที่รัก ที่คอยสนับสนุน และเป็นแรงบันดาลใจในการทำปริญญาานิพนธ์ฉบับนี้ ให้เสร็จลุล่วงไปได้

ผู้วิจัยขอให้ปริญญาานิพนธ์ฉบับนี้ได้ให้ ความรู้ ประโยชน์ และคุณค่า กับผู้อ่านไม่มากนักน้อย

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	1
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและหลักการ.....	3
2.1 ไมโครคอนโทรลเลอร์.....	3
2.1.1 หน่วยประมวลผลกลางหรือซีพียู.....	3
2.1.2 หน่วยความจำ.....	3
2.1.3 ส่วนติดต่ออุปกรณ์ภายนอก หรือ พอร์ต.....	3
2.1.4 ช่องทางเดินของสัญญาณ หรือ บัส (Bus).....	3
2.1.5 วงจรกำเนิดสัญญาณนาฬิกา.....	4
2.2 PIC16F877A.....	4
2.2.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F877A.....	4
2.2.2 วงจรพื้นฐานของไมโครคอนโทรลเลอร์ PIC16F877A.....	6
2.2.3 คุณสมบัติของไมโครคอนโทรลเลอร์ PIC16F877A.....	6
2.3 หลอดแอลอีดี (light – emitting diode).....	7
2.3.1 หลักการทำงานของหลอดแอลอีดี.....	7
2.3.2 ข้อดีของหลอดแอลอีดี.....	8
2.3.3 ข้อจำกัดของหลอดแอลอีดี.....	8
2.4 Dot Martrix Display.....	8
2.5 Altium.....	9
2.6 Microsoft Visual Studio 2010.....	9
2.7 Eclipse.....	9

สารบัญ(ต่อ)

	หน้า
2.8 MikroC Compiler.....	10
2.9 PICKIT2.....	10
2.10 PX-700.....	11
2.11 การประมวลผลภาพดิจิทัล.....	12
2.11.1 ภาพสี.....	12
2.11.2 ภาพระดับสีเทา.....	13
2.11.3 ภาพสองระดับ.....	13
2.12 Optical Character Recognition.....	15
2.12.1 โครงสร้างระบบโอซีอาร์.....	15
2.12.1.1 ขบวนการประมวลผลขั้นต้น.....	15
2.12.1.2 การรู้จำ.....	16
2.12.1.3 การประมวลผลขั้นปลาย.....	17
2.13 บลูทูธโมดูล.....	18
2.14 ชิพตรีจีสเตอร์.....	18
2.15 วงจรนับ.....	19
2.15.1 วงจรการนับแบบไม่เข้าจังหวะ.....	19
2.15.2 วงจรการนับแบบเข้าจังหวะ.....	19
2.16 ระบบบัส SPI.....	20
บทที่ 3 วิธีการดำเนินงาน.....	21
3.1 ฮาร์ดแวร์.....	22
3.1.1 LED Display Board ขนาด 8x8 บิต.....	22
3.1.2 วงจร Control LED Display Board.....	23
3.1.3 พาวเวอร์ซัพพลาย.....	25
3.1.4 Module แปลง USB เป็นพอร์ตอนุกรม TTL.....	25
3.1.5 บลูทูธโมดูล.....	26
3.2 ซอฟต์แวร์.....	27
3.2.1 ส่วนโปรแกรม Microsoft Visual Studio 2010.....	27
3.2.1.1 การใช้กล้องเว็บแคมในการถ่ายภาพ.....	29
3.2.1.2 การเลือกไฟล์รูปภาพ.....	31
3.2.1.3 การพิมพ์ตัวอักษรจากคีย์บอร์ด.....	33
3.2.2 ส่วนโปรแกรม Eclipse.....	37

สารบัญ(ต่อ)

	หน้า
3.2.3 การใช้งาน MikroC Compiler.....	41
บทที่ 4 ผลการทดลอง.....	44
4.1 การทดสอบโปรแกรมสำหรับสั่งให้บอร์ดแอลอีดีแสดงค่าต่างๆ.....	44
4.2 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ บลูทูธ ซึ่งสามารถรับค่าจากโปรแกรม Text Recognition (English Version) และ android application.....	45
4.2.1 ส่งตัวเลขทีละหนึ่งตัว.....	45
4.2.2 ส่งตัวอักษรภาษาอังกฤษทีละหนึ่งตัว.....	47
4.2.3 ส่งคำภาษาอังกฤษที่มี 4 ตัวอักษรพร้อมกัน.....	49
4.3 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ บลูทูธ ซึ่งสามารถรับค่าจากโปรแกรม Text Recognition (Thai Version).....	51
4.4 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ บลูทูธ โดยให้ตัวอักษรเลื่อนจากขวาไปซ้าย.....	53
4.5 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ บลูทูธ โดยให้ตัวอักษรเลื่อนจากล่างขึ้นบน.....	54
บทที่ 5 สรุปผลการทดลอง และข้อเสนอแนะ.....	57
5.1 สรุปผลการทดลอง.....	57
5.2 ข้อเสนอแนะ.....	57
บรรณานุกรม.....	58
ภาคผนวก.....	59
ภาคผนวก ก. แสดงแบบวงจรจากการใช้โปรแกรม Altium Designer 14.3.....	60
ภาคผนวก ข. โปรแกรม.....	63

สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F877A.....	5
2.2 แสดงวงจรพื้นฐานของไมโครคอนโทรลเลอร์ PIC16F877A.....	6
2.3 แสดงรูปหลอดแอลอีดี.....	7
2.4 แสดงรูปโครงสร้างของหลอดแอลอีดี.....	7
2.5 แสดงรูป Dot Matrix Display แบบต่างๆ.....	9
2.6 แสดงรูปอุปกรณ์โปรแกรมไมโครคอนโทรลเลอร์รุ่น PX-700.....	11
2.7 แสดงรูปสี RGB.....	12
2.8 แสดงรูประดับสีเทา.....	13
2.9 แสดงรูปสองระดับ.....	14
2.10 แสดงรูปความสัมพันธ์ของภาพสองระดับ.....	14
2.11 แสดงรูปในระดับสีเทา.....	15
2.12 แสดงรูปที่มีการแบ่ง Threshold.....	15
2.13 แสดงรูปขั้นตอนการทำงานของ OCR.....	15
2.14 แสดงรูปโครงสร้างของ Neural Network.....	17
2.15 แสดงรูปบลูทูธโมดูล.....	18
2.16 แสดงรูปตำแหน่งขาของไอซี 74HC595.....	18
2.17 แสดงรูปขาทั้งหมดของไอซี 74HC595.....	18
2.18 แสดงรูปวงจรการนับแบบไม่เข้าจังหวะ.....	19
2.19 แสดงรูปวงจรการนับแบบเข้าจังหวะ.....	19
3.1 แสดงรูปแผนผังการทำงานของบอร์ดแสดงผลแอลอีดี.....	21
3.2 แสดงรูป Schematic ของ LED Display Board.....	22
3.3 แสดงรูปด้านบนของ LED Display Board ขนาด 8x8 บิต.....	22
3.4 แสดงรูปด้านล่างของ LED Display Board ขนาด 8x8 บิต.....	23
3.5 แสดงรูป Schematic ของวงจร Control LED Display Board.....	24
3.6 แสดงรูป PCB ของวงจร.....	24
3.7 แสดงรูปบอร์ด Control LED Display Board.....	25
3.8 แสดงรูปพาวเวอร์ซัพพลาย.....	25
3.9 แสดงรูปโมดูล USB to TTL UART (CP2102).....	25
3.10 แสดงรูป Bluetooth Module HC-05.....	26
3.11 แสดงรูปส่วนประกอบฮาร์ดแวร์ทุกส่วนมาประกอบกัน และทำการเดินสายไฟ.....	26
3.12 แสดงรูปการทำงานของโปรแกรม.....	27

สารบัญรูป

รูปที่	หน้า
3.13 แสดงรูปหน้าจอเริ่มต้นโปรแกรม Text Recognition (English Version).....	28
3.14 แสดงรูปหน้าต่างโปรแกรม เลือกพอร์ตอนุกรม.....	28
3.15 แสดงรูปหน้าต่างโปรแกรม เลือก Baud rate.....	28
3.16 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม ‘Connect’.....	29
3.17 แสดงรูปหน้าต่างโปรแกรมที่กำลังจะจับภาพด้วยกล้องเว็บแคม.....	29
3.18 แสดงรูปหน้าต่างโปรแกรมที่จับภาพโดยกดปุ่ม ‘Capture’.....	30
3.19 แสดงรูปหน้าต่างโปรแกรมที่กดปุ่ม ‘GetText’ และแสดงผลลัพธ์ที่ได้.....	30
3.20 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม ‘Write’.....	31
3.21 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม ‘Browse’.....	31
3.22 แสดงรูปหน้าต่างโปรแกรมเมื่อเลือกไฟล์รูปภาพและแสดงผลลัพธ์.....	32
3.23 แสดงรูปหน้าต่างโปรแกรมเมื่อกดปุ่ม ‘Write’.....	32
3.24 แสดงรูปหน้าต่างโปรแกรม พิมพ์ ‘KeyboardInput’.....	33
3.25 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม ‘Go’.....	33
3.26 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม ‘Write’.....	34
3.27 แสดงรูปหน้าต่างโปรแกรม ตัวอักษรภาษาไทย.....	34
3.28 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม ‘Symbol’.....	35
3.29 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม ‘Write’.....	35
3.30 Flow Chart แสดงขั้นตอนการทำงานของ Text Recognition.....	36
3.31 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการสร้าง Android Application Project.....	37
3.32 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการตั้งค่าต่างๆ.....	37
3.33 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการกำหนดไอคอน.....	38
3.34 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการเลือก Activity.....	38
3.35 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการกำหนดชื่อ Activity Name และ Layout Name.....	39
3.36 แสดงรูปหน้าต่างโปรแกรม Eclipse เมื่อทำการสร้าง Android Application Project เสร็จเรียบร้อย.....	39
3.37 Flow Chart แสดงขั้นตอนการทำงานของ Android Application.....	40
3.38 โปรแกรม MikroC Compiler.....	41
3.39 หน้าต่างสำหรับกำหนดรายละเอียดของไมโครคอนโทรลเลอร์.....	41
3.40 หน้าต่างเมื่อ Compile โปรแกรม.....	42
3.41 ไฟล์นามสกุล .hex สำหรับใช้โปรแกรมลงไมโครคอนโทรลเลอร์.....	42
3.42 แสดงรูปขั้นตอนการทำงานของโปรแกรม MikroC Compiler.....	43
4.1 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดครบทุกดวง.....	44

สารบัญรูป

รูปที่	หน้า
4.2 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดหลักคู่ (2, 4, 6, 8).....	44
4.3 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดแถวคี่ (1, 3, 5, 7).....	44
4.4 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดเป็นรูปหน้ายิ้ม.....	45
4.5 แสดงรูปการส่งเลข 0 จากโปรแกรม Text Recognition (English Version) และ android application.....	45
4.6 แสดงรูปการส่งเลข 1 จากโปรแกรม Text Recognition (English Version) และ android application.....	46
4.7 แสดงรูปการส่งอักษร Z จากโปรแกรม Text Recognition (English Version) และ android application.....	47
4.8 แสดงรูปการส่งอักษร V จากโปรแกรม Text Recognition (English Version) และ android application.....	48
4.9 แสดงรูปการส่งคำว่า WORK จากโปรแกรม Text Recognition (English Version) และ android application.....	49
4.10 แสดงรูปการส่งคำว่า TIME จากโปรแกรม Text Recognition (English Version) และ android application.....	50
4.11 แสดงรูปการส่งอักษร ก จากโปรแกรม Text Recognition (Thai Version).....	51
4.12 แสดงรูปการส่งอักษร น จากโปรแกรม Text Recognition (Thai Version).....	51
4.13 แสดงรูปการส่งอักษร ข จากโปรแกรม Text Recognition (Thai Version).....	52
4.14 แสดงรูปการส่งอักษร ร จากโปรแกรม Text Recognition (Thai Version).....	52
4.15 แสดงรูปการเลื่อนตัวอักษรจากขวาไปซ้าย.....	53
4.16 แสดงรูปการเลื่อนตัวอักษรจากล่างขึ้นบน.....	54
4.17 แสดงรูปตัวอักษร A-Z.....	55
4.18 แสดงรูปตัวเลข 0-9.....	56
4.19 แสดงรูปจำลองตัวอักษรภาษาไทยทั้งหมดที่สามารถทำได้.....	56

บทที่ 1

บทนำ

1.1 ความสำคัญของปฏิญญานิพนธ์

เนื่องจากในปัจจุบันนี้เทคโนโลยีทางด้านสารสนเทศเข้ามามีบทบาทเป็นอย่างมากในชีวิตประจำวัน เพราะปัจจุบันเป็นโลกของข่าวสารข้อมูล ที่เป็นหนึ่งเดียวกันทั่วโลกโดยเฉพาะอย่างยิ่ง ด้านธุรกิจและการโฆษณา ดังนั้นในปัจจุบันจึงต้องการสื่อที่สามารถแสดงข่าวสารข้อมูลรวมทั้ง ภาพ การเคลื่อนไหวต่างๆ เพื่อให้สามารถดึงดูดความสนใจ และสามารถสื่อสารให้ทุกคนได้เข้าใจในข้อมูลข่าวสารนั้นๆ

กระดานหรือแผ่นป้ายแสดงข่าวสาร (Display Board) เป็นอีกสื่อหนึ่งที่นิยมใช้กันมากในปัจจุบัน ซึ่งสามารถพบเห็นได้ตามแหล่งชุมชน ย่านธุรกิจ ศูนย์การค้า โรงพยาบาลและตามบริษัทห้างร้านต่างๆ ซึ่งในอดีตยังคงใช้ระบบการแสดงผลเป็นเซเวนเซกเมนต์ (7-Segment) และต่อมาจึงมีการพัฒนาเป็นแบบจุดแสดงผล (Dot Matrix Display) ซึ่งจะมีความละเอียดของภาพ (Resolution) มากกว่าแบบเดิม นอกจากนี้แล้วปัจจุบันป้ายโฆษณาที่มีลักษณะเป็นตัวอักษรเลื่อนก็เป็นที่ยอมรับ ดังจะเห็นได้จากตามห้างสรรพสินค้าทั่วไป หรือแม้แต่ตามร้านค้าใหญ่ๆ มักจะมีป้ายโฆษณาในลักษณะนี้ติดตั้งอยู่มาก ซึ่งสามารถดึงดูดความสนใจได้มากกว่าป้ายโฆษณาทั่วๆ ไป เนื่องจากมองเห็นได้เด่นชัดเจน มีรูปแบบในการแสดงผลที่น่าสนใจ จึงมีผู้นิยมใช้กันมากขึ้นเรื่อยๆ

จากเหตุผลที่กล่าวมาข้างต้น จึงได้เกิดแนวคิดในการทำบอร์ดแสดงผลผ่านหลอดแอลอีดี ขึ้นมา เนื่องจากหลอดแอลอีดีนั้นประหยัดพลังงานเพราะใช้พลังงานน้อยมาก แต่ให้ประสิทธิภาพในการส่องสว่างสูง อีกทั้งในระบบอินสทรูเมนต์นั้นจำเป็นที่จะต้องมียูนิตเคเตอร์ในการแสดงสถานะการทำงานของระบบ ซึ่งหลอดแอลอีดีก็เป็นอีกทางเลือกหนึ่งที่โรงงานอุตสาหกรรมนิยมเลือกใช้ โดยในการศึกษาการทำบอร์ดแสดงผลผ่านหลอดแอลอีดีนั้นยังสามารถประยุกต์ใช้ความรู้ในหลายๆ ด้าน ไม่ว่าจะเป็น อิเล็กทรอนิกส์ ฮาร์ดแวร์ ซอฟต์แวร์ และ คอมพิวเตอร์

1.2 วัตถุประสงค์ของปฏิญญานิพนธ์

เพื่อศึกษาและออกแบบวงจรการใช้งาน LED Display Board ให้สามารถแสดงตัวเลข และตัวอักษรตามที่กำหนด จากระบบปฏิบัติการแอนดรอยด์ คีย์บอร์ด และจากรูปภาพ รวมถึงกล้อง webcam ทั้งแบบมีสายและไร้สาย โดยที่ตัวอักษรและตัวเลขนั้นสามารถที่จะเลื่อนไปจนจบข้อความได้

1.3 ขอบเขตของปฏิญญานิพนธ์

1. ทำการศึกษาและแสดงผลตัวเลขและตัวอักษรผ่าน LED Display Board
2. ทำการศึกษาและออกแบบโปรแกรม Optical Character Recognition ใน Microsoft Visual Studio 2010 ด้วยภาษา VB.NET
3. ทำการศึกษาและออกแบบแอปพลิเคชันแอนดรอยด์ เพื่อส่งตัวเลขและตัวอักษร โดยใช้โปรแกรม Eclipse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการศึกษาและเขียนโปรแกรมสำหรับควบคุมการแสดงผลของ LED Display Board ผ่านไมโครคอนโทรลเลอร์ PIC16F877A

1.4 ขั้นตอนการศึกษา

1. ศึกษาหลักการทำงานของหลอดแอลอีดี
2. ศึกษาและออกแบบวงจรของ LED Display Board
3. ศึกษาการทำงานของไมโครคอนโทรลเลอร์ PIC16F877A
4. ทำการทดลองโดยการจำลองการแสดงผลบน LED Display Board ผ่านโปรแกรม Proteus
5. ศึกษาการออกแบบวงจรโดยใช้โปรแกรม Altium Designer 14.3
6. ศึกษาการทำวงจร Electronic เช่น การทำแผ่นPCB และการต่อวงจรลงบนโปรโตบอร์ด
7. ศึกษาการแปลงภาพสีเป็นภาพขาวดำ
8. ศึกษาเกี่ยวกับการสร้างและออกแบบ Optical Character Recognition ใน Microsoft Visual Studio 2010 ด้วยภาษา VB.NET
9. ศึกษาการเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์เพื่อทำการควบคุมการรับค่าและส่งค่าไปแสดงผลที่ LED Display Board โดยใช้โปรแกรม MikroC Compiler
10. ศึกษาการเขียนแอปพลิเคชันแอนดรอยด์ โดยใช้โปรแกรม Eclipse
11. ศึกษาการสื่อสารแบบมีสายและไร้สายระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ต่างๆ เช่น คอมพิวเตอร์ และ โทรศัพท์มือถือ โดยใช้พอร์ตอนุกรม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการทำงานและการเขียนโปรแกรมเพื่อควบคุมไมโครคอนโทรลเลอร์ PIC16F877A
2. สามารถออกแบบวงจรโดยใช้โปรแกรม Altium Designer 14.3 ได้
3. สามารถออกแบบและเขียนโปรแกรม Microsoft Visual Studio 2010 ได้
4. สามารถออกแบบและสร้างแอปพลิเคชันแอนดรอยด์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ เป็นอุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวเดียวกัน โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

2.1.1 หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) ทำหน้าที่เป็นศูนย์กลางควบคุมการทำงานของระบบคอมพิวเตอร์ทั้งหมด โดยนำข้อมูลจากอุปกรณ์รับข้อมูลมาทำงานประมวลผลข้อมูลตามคำสั่งของโปรแกรม และส่งผลลัพธ์ออกไปหน่วยแสดงผล

2.1.2 หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของ เครื่องคอมพิวเตอร์ตั้งโต๊ะคือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดานขุดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำ (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

2.1.3 ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะใช้ด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

2.1.4 ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

2.1.5 วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะ หากสัญญาณนาฬิกามีความถี่สูงจังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้นมีความเร็วในการประมวลผลสูงตามไปด้วย

นอกจากนี้ยังมีส่วนพิเศษอื่นๆ จะขึ้นอยู่กับกระบวนการผลิตของแต่ละบริษัทที่จะผลิตขึ้นมาใส่คุณสมบัติพิเศษลงไปเช่น

- ADC (Analog to Digital) ส่วนภาครับสัญญาณอนาล็อกแปลงไปเป็นสัญญาณดิจิทัล
- DAC (Digital to Analog) ส่วนภาคส่งสัญญาณดิจิทัลแปลงไปเป็นสัญญาณอนาล็อก
- I2C (Inter Integrate Circuit Bus) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสาร ระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถ เชื่อมต่ออุปกรณ์ จำนวนหลายๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น
- SPI (Serial Peripheral Interface) เป็นการเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส (Synchronize) มีสัญญาณนาฬิกาเข้ามาเกี่ยวข้องระหว่างไมโครคอนโทรลเลอร์ (Microcontroller) หรือจะเป็นอุปกรณ์ภายนอกที่มีการรับส่งข้อมูลแบบ SPI อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์ (Master) โดยปกติแล้วจะเป็นไมโครคอนโทรลเลอร์ หรืออาจกล่าวได้ว่าอุปกรณ์ Master จะต้องควบคุมอุปกรณ์ Slave ได้ โดยปกติตัว Slave มักจะเป็นไอซี (IC) หน้าที่พิเศษต่างๆ เช่น ไอซีอุณหภูมิ, ไอซีฐานเวลานาฬิกาจริง (Real-Time Clock) หรืออาจเป็นไมโครคอนโทรลเลอร์ที่ทำหน้าที่ในโหมด Slave ก็ได้เช่นกัน
- PWM (Pulse Width Modulation) การสร้างสัญญาณพัลส์แบบสแควร์เวฟ ที่สามารถปรับเปลี่ยนความถี่และ Duty Cycle ได้เพื่อนำไปควบคุมอุปกรณ์ต่างๆเช่น มอเตอร์
- UART (Universal Asynchronous Receiver Transmitter) ทำหน้าที่รับส่งข้อมูลแบบอะซิงโครนัสสำหรับมาตรฐานการรับส่งข้อมูลแบบ RS-232

2.2 PIC16F877A

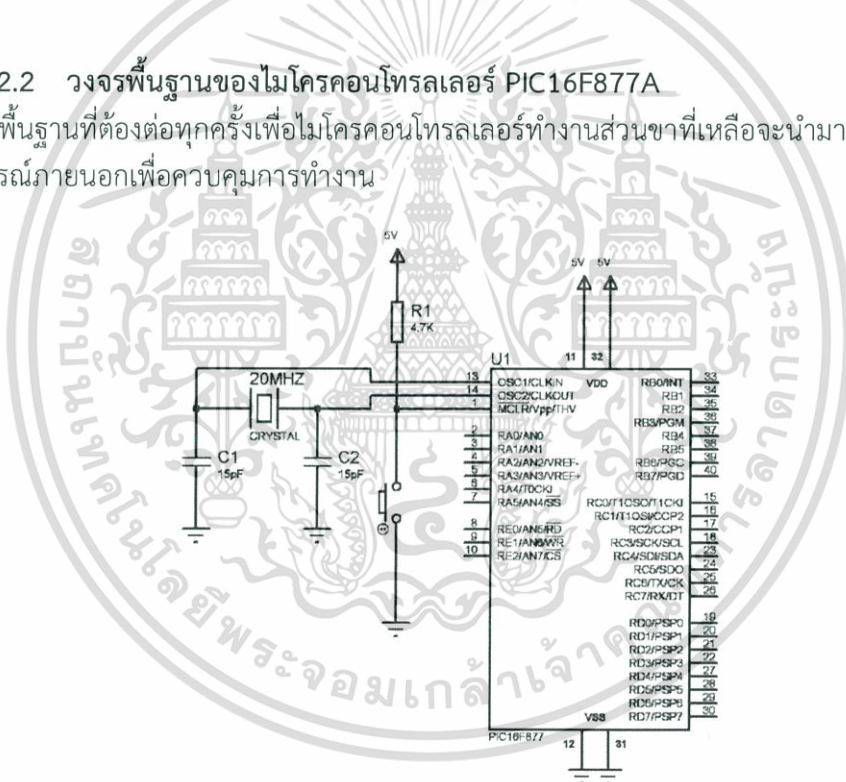
2.2.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F877A

PIC16F877A มีขนาด 40 ขา เป็นแบบ EEPROM/Flash ROM และมีหน่วยความจำโปรแกรม 8 กิโลไบต์ จากรูปที่ 2.1 ขาของ PIC16F877A แต่ละขาจะมีหน้าที่แตกต่างกันไปซึ่งแยกออกเป็น พอร์ต A จำนวน 6 บิต พอร์ต B จำนวน 8 บิต พอร์ต C จำนวน 8 บิต พอร์ต D จำนวน 8 บิต และพอร์ต E จำนวน 3 บิต โดยพื้นฐานแล้วพอร์ตแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาต์พุตเป็นดิจิทัลได้ทุกพอร์ทยกเว้น พอร์ต A และพอร์ต E ที่มีความสามารถพิเศษสามารถทำงานเป็นตัวรับสัญญาณอนาล็อกและแปลงเป็นค่าดิจิทัลเพื่อนำมาวัดปริมาณทางฟิสิกส์ต่างๆ สัญญาณนาฬิกาต้องใช้จากภายนอกต่อเข้าที่ขา 13 และ 14 การดาวน์โหลดโปรแกรมลงไมโครคอนโทรลเลอร์แบบ SPI ผ่านทางขา RB7/CSPDAT

and Logic Unit) โดยนำข้อมูลกระทำกับรีจิสเตอร์หรือรีจิสเตอร์ W (Accumulator) ผลลัพธ์สามารถเลือกเก็บได้ 2 ทางคือ การเก็บผลลัพธ์ไว้ที่ W หรือนำไปเก็บไว้ที่ไฟล์รีจิสเตอร์ทั่วไป (RAM File Registers) หรือรีจิสเตอร์ใช้งานเฉพาะเช่น PORT A หรือไปยังโมดูลที่ต่อร่วมไว้ในไมโครคอนโทรลเลอร์อันได้แก่ TMR0, TIMER1, EEPROM, 10 bit A/D, Comparator ทางบัสข้อมูล (Data Bus) อีกส่วนหนึ่งที่ได้จากหน่วยประมวลผลทางคณิตศาสตร์และลอจิก ALU (Arithmetic and Logic Unit) โดยนำข้อมูลกระทำกับรีจิสเตอร์หลักหรือรีจิสเตอร์ W (Accumulator) ผลลัพธ์สามารถเลือกเก็บได้ 2 ทางคือ การเก็บผลลัพธ์ไว้ที่ W หรือนำไปเก็บไว้ที่ไฟล์รีจิสเตอร์ทั่วไป (RAM File Registers) หรือรีจิสเตอร์ใช้งานเฉพาะเช่น PORT A หรือไปยังโมดูลที่ต่อร่วมไว้ในไมโครคอนโทรลเลอร์อันได้แก่ TMR0, TIMER1, EEPROM, 10 bit A/D, Comparator ทางบัสข้อมูล (Data Bus) อีกส่วนหนึ่งที่ได้จากหน่วยประมวลผลทางคณิตศาสตร์และลอจิก จำนวน 3 บิต ได้แก่ บิตตัวทด (Carry/borrow bit) บิตศูนย์ (Zero bit) และบิตตัวทดเมื่อมีการทศระหว่างสี่บิตต่างไปยังสี่บิตบน (Digit carry/borrow bit)

2.2.2 วงจรพื้นฐานของไมโครคอนโทรลเลอร์ PIC16F877A

วงจรพื้นฐานที่ต้องต่อทุกครั้งเพื่อไมโครคอนโทรลเลอร์ทำงานส่วนขาที่เหลืองนำมาเชื่อมต่อกับอุปกรณ์ภายนอกเพื่อควบคุมการทำงาน



รูปที่ 2.2 แสดงวงจรพื้นฐานของไมโครคอนโทรลเลอร์ PIC16F877A

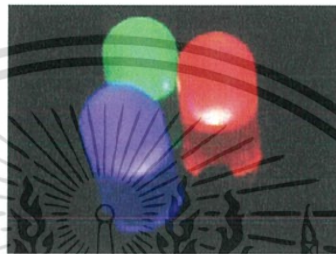
2.2.3 คุณสมบัติของไมโครคอนโทรลเลอร์ PIC16F877A

- ขาสัญญาณที่ต่อใช้งาน : 40 ขา
- ไฟเลี้ยง : 2 – 5.5 โวลต์
- พอร์ตที่ใช้งานทั่วไป : 33 บิต
- หน่วยความจำทั่วไป : 8192x14 words

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความเร็วในการประมวลผล : 4 MHz
- โมดูลจับเวลาหรือไทม์เมอร์ : 8 บิต 2 ตัว , 16 บิต 1 ตัว
- หน่วยความจำข้อมูลชนิด EEPROM : 256x8 bytes
- โมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัล : 10 บิต 8 ช่อง
- หน่วยความจำฟลैชรีจิสเตอร์ (SRAM) : 368 bytes

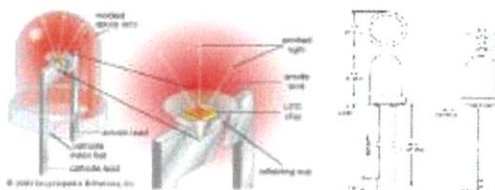
2.3 หลอดแอลอีดี (light-emitting diode)



รูปที่ 2.3 แสดงรูปหลอดแอลอีดี

2.3.1 หลักการทำงานของหลอดแอลอีดี

หลอดแอลอีดี หรือไดโอดเปล่งแสง โครงสร้างประกอบไปด้วยสารกึ่งตัวนำสองชนิด (สารกึ่งตัวนำชนิด N และสารกึ่งตัวนำชนิด P ประกอบเข้าด้วยกัน มีผิวข้างหนึ่งเรียบคล้ายกระจกเมื่อจ่ายไฟฟ้ากระแสตรงผ่านหลอดแอลอีดี โดยจ่ายไฟบวกให้ขาแอนโนด (A) จ่ายไฟลบให้ขาแคโทด (K) ทำให้อิเล็กตรอนที่สารกึ่งตัวนำชนิด N มีพลังงานสูงขึ้น จนสามารถวิ่งข้ามรอยต่อจากสารชนิด N ไปรวมกับโฮลในสารชนิด P การที่อิเล็กตรอนเคลื่อนที่ผ่านรอยต่อ PN ทำให้เกิดกระแสไหล เป็นผลให้ระดับพลังงานของอิเล็กตรอนเปลี่ยนไปและคายพลังงานออกมาในรูปคลื่นแสง สีของแสงที่เกิดจากรอยต่อจะขึ้นอยู่กับชนิดของวัสดุที่นำมาใช้ในการสร้างหลอดแอลอีดี ทั้งชนิดที่เป็นของเหลวและก๊าซ เช่น ใช้แกเลียมฟอสไฟด์ (GALLIUM PHOSPHIDE, GaP) ทำให้เกิดแสงสีแดง ใช้แกเลียมอาร์เซไนด์ ฟอสไฟด์ (GALLIUM ARSENIDE PHOSPHIDE, GaAsP) เกิดแสงสีเหลืองและเขียวการควบคุมปริมาณแสงสว่างจะควบคุมกระแสที่ไหลผ่านหลอดแอลอีดี หากกระแสที่ไหลสูงมากไปจะทำให้หลอดมีความสว่างมาก แต่หากป้อนกระแสสูงมากไปจะทำให้บริเวณรอยต่อของสารกึ่งตัวนำเกิดความร้อนปริมาณมากจนทำให้โครงสร้างหลอดเสียหายไม่สามารถใช้งานได้



รูปที่ 2.4 แสดงรูปโครงสร้างของหลอดแอลอีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ข้อดีของหลอดแอลอีดี

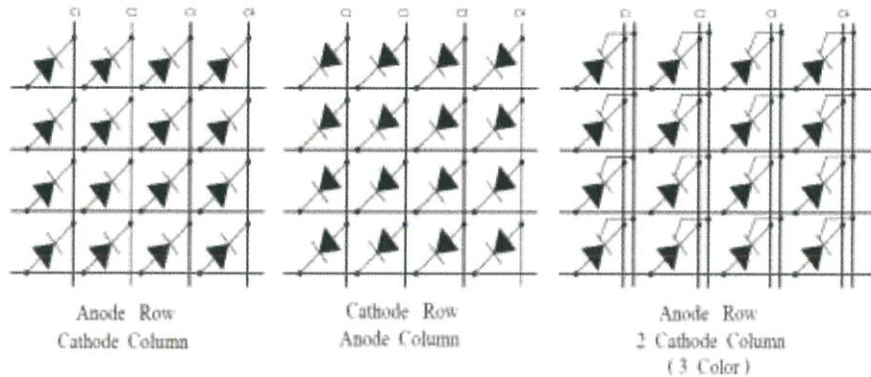
- มีประสิทธิภาพการให้แสงสว่างสูง และทิศทางแสงสว่างของหลอดแอลอีดี จะส่องไปเฉพาะด้านหน้าเท่านั้นลดการสูญเสียเปลวของแสงสว่าง
- ใช้พลังงานน้อย และอายุการใช้งานยาวนานถึง 100,000 ชั่วโมง
- สามารถเปิดปิดได้บ่อยครั้ง และเมื่อเปิดจะให้แสงสว่างโดยทันที
- สามารถควบคุมคุณภาพของแสงที่ปล่อยออกมาได้จึงสามารถนำไปใช้ให้แสงสว่างในบางสถานที่ได้ เช่น การให้แสงสว่างกับภาพเขียน เนื่องจากสามารถควบคุมแสงสว่างจากหลอดอีดี ไม่ให้มีส่วนผสมของแสงที่เป็นอันตรายต่อภาพเขียน เช่น แสงอินฟราเรด และแสงอัลตราไวโอเลต
- ปล่อยความร้อนออกมาน้อยมากทำให้ลดการสูญเสียพลังงานไฟฟ้าในส่วนเครื่องปรับอากาศ
- การดูแลรักษาต่ำ
- น้ำหนักเบา, ขนาดเล็ก

2.3.3 ข้อจำกัดของหลอดแอลอีดี

- ในการนำหลอดแอลอีดี มาใช้งานต้องมีการทดสอบสีว่า แสงที่ออกมาเป็นแสงสีที่ถูกต้องหรือไม่
- ราคาหลอดแอลอีดี ยังแพงกว่าหลอดฟลูออเรสเซนต์อยู่มาก

2.4 Dot Matrix Display

หลอดแอลอีดี ทั่วไปจะมีลักษณะเป็นหลอดเดี่ยว ๆ จะใช้เป็นตัวแสดงผลแบบ สว่าง /ดับ (ON-OFF) ถ้านำเอาหลอดแอลอีดี มารวมกันให้เป็นกลุ่มแถวในแนวนอนและแนวตั้งแล้ว รูปแบบของการแสดงผลก็จะเปลี่ยนไป การสว่าง /ดับของหลอดแอลอีดี สามารถที่จะสร้างให้ เกิดเป็นข้อความหรือรูปลักษณะต่าง ๆ ตามที่เราต้องการได้ กลุ่มของหลอดแอลอีดี ที่ว่านี้เรียกว่า Dot Matrix LED Display หรือ Dot Matrix Display ซึ่งมีอยู่ด้วยกันหลายแบบหลายขนาด เช่น ขนาด 4x4 , 5x7 , 5x8 , 7x5 , 8x8 , 16x6 มีทั้ง แบบสีเดี่ยว (Single Color) หรือแบบหลายสี (Multi Color) และ ยังแบ่งตามการใช้งานเป็นแบบAnode Row Cathode Column, Cathode Row Anode Column และ Anode Row 2 Cathode Column เป็นต้น



รูปที่ 2.5 แสดงรูป Dot Matrix Display แบบต่างๆ

2.5 Altium

การใช้ Altium Designer ความสามารถหลักๆครอบคลุมฟังก์ชันการออกแบบทางด้านอิเล็กทรอนิกส์ 6 ด้านดังนี้

- Board level design : การออกแบบ Printed circuit board (PCB)
- Programmable device : การจัดการอุปกรณ์โปรแกรมได้ ทั้ง embedded และ Field Programmable Gate Array (FPGA)
- FPGA/PCB integration : การประสานระหว่างการใช้งาน Field Programmable Gate Array (FPGA) และการออกแบบ Printed circuit board (PCB)
- Design to manufacture : การออกแบบให้สามารถนำไปสู่กระบวนการผลิตได้
- Library management : การจัดการไลบรารี
- Document management : การจัดการเอกสารทั้งหมดของชิ้นงาน

2.6 Microsoft Visual Studio 2010

Microsoft Visual Studio.NET เป็นเครื่องมือที่ใช้พัฒนาโปรแกรม Visual Programming บนระบบปฏิบัติการ Window ซึ่งรองรับภาษาในการเขียนโปรแกรมที่หลากหลายภาษา เช่น VB, C#, C++ เป็นต้น รวมทั้งคิดค้นภาษา Java เพื่อให้ใช้งานได้บนระบบปฏิบัติการใด ๆ ก็ได้ไม่ว่า จะเป็น DOS , Windows 7 , Windows XP , Linux หรือ UNIX (Mayo, 2010) และในปัจจุบันยัง สามารถใช้งานได้ใ้ในอุปกรณ์ไร้สายได้อีกด้วย นอกจากนั้นยังถูกผนวกเข้ากับโปรแกรมอื่นๆของ ไมโครซอฟท์เช่น Microsoft Access , Excel , Word เป็นต้น เพื่อใช้เขียนโปรแกรมลักษณะ Script หรือ Macro8

2.7 Eclipse

Eclipse คือโปรแกรมที่ใช้สำหรับพัฒนาภาษา Java ซึ่งโปรแกรม Eclipse เป็นโปรแกรมหนึ่งที่ใช้ในการพัฒนา Application Server ได้อย่างมีประสิทธิภาพ และเนื่องจาก Eclipse เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์ OpenSource ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆจากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมีองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug-ins หากต้องการให้ Eclipse ทำงานใดเพิ่มเติม จะต้องพัฒนา Plug-in สำหรับงานนั้นขึ้นมา และนำ Plug-in นั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse Plug-in ที่มีมาพร้อมกัน Eclipse เมื่อเรา download มาครั้งแรกก็คือองค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ debug โปรแกรมภาษา Java

ข้อดีของโปรแกรม คือ ติดตั้งง่าย สามารถใช้ได้กับ J2SDK ได้ทุกเวอร์ชัน รองรับภาษาต่างประเทศ อีกหลายภาษา มี plugin ที่ใช้เสริมประสิทธิภาพของโปรแกรม สามารถทำงานได้กับไฟล์หลายชนิด เช่น HTML, Java, C, JSP, EJB, XML และ GIF ใช้งานได้กับระบบปฏิบัติการ Windows, Linux และ Mac OS

2.8 MikroC Compiler

โปรแกรม MikroC Compiler เป็นซอฟต์แวร์สำหรับแปลโปรแกรมภาษา C ของไมโครคอนโทรลเลอร์ตระกูล PIC เป็นรหัสเครื่องหรือแมคชีนโค้ด (machine code) ผลิตโดยบริษัท MikroElektronika โดยมีคุณสมบัติที่น่าสนใจดังนี้

- มีความสามารถสูง มีเครื่องมือที่ช่วยอำนวยความสะดวกให้กับนักพัฒนาโปรแกรมได้เป็นอย่างดี
- มีตัวอย่างรองรับการเรียนรู้สำหรับผู้เริ่มต้นใช้งาน
- รองรับไมโครคอนโทรลเลอร์ตระกูล PIC/dsPIC ทุกเบอร์ เนื่องจากมีการพัฒนาอย่างต่อเนื่อง

2.9 PICKIT2

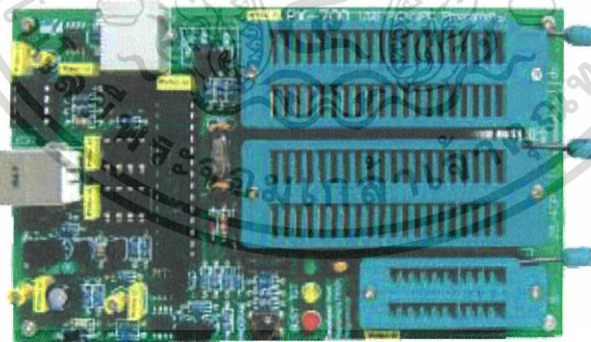
โปรแกรม PICKit2 พัฒนาโดยบริษัท Microchip เป็นซอฟต์แวร์ใช้สำหรับโปรแกรมภาษาเครื่อง (.hex) ลงไปบน PIC โดยมีฟังก์ชันในการเขียนข้อมูล (Write), อ่านข้อมูล (Read) , ตรวจสอบข้อมูล (Verify), ลบข้อมูล (Erase) และยังสามารถตรวจสอบความผิดพลาดของวงจรภายในไมโครคอนโทรลเลอร์ได้อีกด้วย ซึ่งจะต้องใช้ร่วมกับอุปกรณ์ในการโปรแกรมไมโครคอนโทรลเลอร์ด้วย

2.10 PX-700

อุปกรณ์ในการโปรแกรมไมโครคอนโทรลเลอร์รุ่น PX-700 ใช้สำหรับโปรแกรมภาษาเครื่องลงไปบนไมโครคอนโทรลเลอร์ โดยจะต้องใช้ร่วมกับซอฟต์แวร์บนคอมพิวเตอร์เพื่อใช้ในการโปรแกรมข้อมูลลงไปบนไมโครคอนโทรลเลอร์ผ่านพอร์ต USB

คุณสมบัติโดยทั่วไป

- เชื่อมต่อคอมพิวเตอร์ผ่านพอร์ต USB
- ใช้ไฟเลี้ยงจากพอร์ต USB
- มีหลอดแอลอีดี แสดงสถานะการทำงาน 3 ดวง (POWER, TARGET และBUSY)
- สามารถอัปเดตเฟิร์มแวร์ได้โดยผ่านทางซอฟต์แวร์
- บนบอร์ดมีซ็อกเก็ต ZIF 3 ชุดสำหรับโปรแกรม
- ไมโครคอนโทรลเลอร์ PIC 8-20 ขา PIC12Fxxx, PIC16Fxxx
- ไมโครคอนโทรลเลอร์ PIC 28/40 ขา PIC16F/PIC18F
- ไมโครคอนโทรลเลอร์ dsPIC 28/40 dsPIC30F
- การดีบั๊กขึ้นอยู่กับรุ่นของเฟิร์มแวร์ โดยต้องใช้ PX-700 ร่วมกับโปรแกรม MPLAB V8 ขึ้นไป โดยเลือก Debug Tools เป็น Pickit2
- สามารถโปรแกรมไมโครคอนโทรลเลอร์ PIC แบบแฟลชได้ทุกเบอร์



รูปที่ 2.6 แสดงรูปอุปกรณ์โปรแกรมไมโครคอนโทรลเลอร์รุ่น PX-700

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

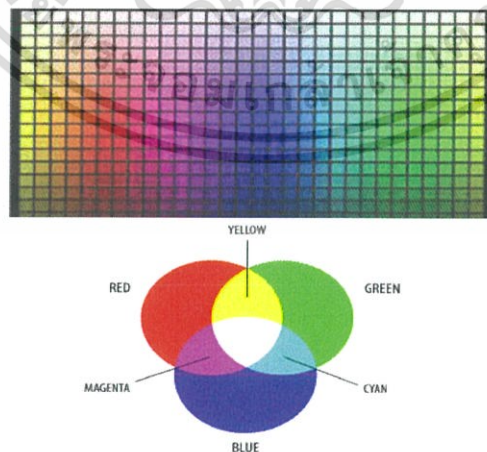
2.11 การประมวลผลภาพดิจิทัล (Digital Image Processing)

ภาพดิจิทัล (Digital Image) หมายถึง ภาพที่เก็บอยู่ในรูปแบบดิจิทัล ซึ่งภาพที่มนุษย์มองเห็นด้วยสายตาทั่วไปนั้นเป็นภาพในลักษณะของสามมิติ คือมีมิติของความกว้าง ความยาว และความลึก ภาพดิจิทัลเป็นการแสดงผลในลักษณะของฟังก์ชันสองมิติ $f(x, y)$ โดยที่ x และ y เป็นตำแหน่งพิกัดของภาพ ส่วนแอมพลิจูดของ f ที่พิกัด (x, y) ใดๆภายในภาพคือ ค่าความเข้มแสง (Intensity) ที่ตำแหน่งนั้นๆ และเมื่อ x, y และแอมพลิจูดของ f เป็นค่าจำกัด (Finite value) จึงเรียกรูปภาพนี้ว่าเป็นภาพดิจิทัล (Digital Image) และถ้ากำหนดให้ภาพ $f(x, y)$ มีขนาด M แถว N คอลัมน์ พิกัดของจุดกำเนิด (Origin) ของภาพคือที่ตำแหน่ง $(x, y) = (0, 0)$ แล้วค่าแต่ละค่าที่มีอยู่ในเมตริกซ์จะเรียกว่าพิกเซล (Pixel)

การประมวลผลภาพ (Image Processing) คือการใช้ขั้นตอนหรือกระบวนการต่างๆมากระทำกับภาพดิจิทัล โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพเพื่อให้ได้ภาพที่มีคุณสมบัติตามต้องการ เช่น ทำให้ภาพมีความคมชัดมากขึ้น ทำให้ภาพสว่างขึ้นหรือมืดลง ทำให้ภาพมีสีสันทันมากยิ่งขึ้น ย่อ-ขยายภาพเพื่อประหยัดพื้นที่ในการเก็บข้อมูล เป็นต้น โดยชนิดของข้อมูลภาพที่ใช้ในการประมวลผลภาพดิจิทัลจะประกอบได้ดังนี้

2.11.1 ภาพสี RGB (RGB Image)

สีของรูปภาพต่างๆบนจอคอมพิวเตอร์ที่ใช้กันอยู่ในปัจจุบันเกิดจากการผสมสีของแสง 3 สี คือ แดง(Red) เขียว(Green) และน้ำเงิน(Blue) เรียกกันง่ายๆว่า “RGB” ซึ่งสีอื่นๆที่ต่างไปจากนี้เกิดจากการผสมสีของแสงมาผสมกัน เช่น สีแดงผสมสีเขียวได้สีเหลือง สีแดงผสมสีน้ำเงินได้สีม่วงหรือหากเอาทุกสีมาผสมกันก็ได้สีขาว โดยระบบสีแบบ 8 บิต จะเป็นการใช้หน่วยความจำดิจิทัลขนาด 8 บิต มาแทนเฉดสีแต่ละสีในเชิงคณิตศาสตร์ ซึ่งในแต่ละบิตมีค่าเป็น “0” กับ “1” ซึ่งแปลว่าจะมีค่าตัวเลขที่เป็นไปได้จำนวน 2 ยกกำลัง 8 ($2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$) หรือตั้งแต่ “0000000” ฐาน 2” (มีค่าเท่ากับ 0 ในฐาน 10) จนถึง “11111111” ฐาน 2” (มีค่าเท่ากับ 255 ในฐาน 10) ซึ่งหมายความว่า มีเฉดสีต่างๆกันในสีใดสีหนึ่งได้ 256 เฉดสี



รูปที่ 2.7 แสดงรูปสี RGB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2 ภาพระดับสีเทา (Gray Scale Image)

เนื่องจากภาพสี RGB เกิดจากการผสมสีกันของสีแดง เขียว และน้ำเงิน ในอัตราส่วนที่ต่างกัันในแต่ละพิกเซล ทำให้เกิดเป็นภาพสี จึงเป็นเรื่องยากที่จะนำภาพสีมาประมวลผลเลยดังนั้นเพื่อให้ง่ายขึ้นก่อนที่จะนำภาพไปประมวลผลจึงควรทำภาพสีให้เป็นภาพในระดับสีเทาก่อน ซึ่งภาพระดับสีเทานั้น พิกเซลแต่ละพิกเซลจะมีค่าความเข้มสีเป็นค่าเฉลี่ยของค่าสีแม่แบบทั้ง 3 ในพิกเซลนั้น ซึ่งอัลกอริทึมที่นิยมใช้ในการทำภาพสีเป็นภาพระดับสีเทานั้นก็คือ การเฉลี่ยค่าแม่สีทั้ง 3 ใน 1 พิกเซล เขียนสมการได้ดังนี้

$$\text{ค่าระดับสีเทา} = ((0.299 \times R) + (0.587 \times G) + (0.114 \times B)) \quad (2.1)$$

R คือ ค่าอินพุตพิกเซลสีแดง

G คือ ค่าอินพุตพิกเซลสีเขียว

B คือ ค่าอินพุตพิกเซลสีน้ำเงิน



รูปที่ 2.8 แสดงรูประดับสีเทา

2.11.3 ภาพสองระดับ (Binary Image)

ภาพสองระดับหมายถึงภาพที่มีระดับความเข้มของสีเพียง 2 ระดับ คือแต่ละพิกเซลจะมีค่าเพียงสองค่าคือ 0 และ 1 โดยพิกเซลที่มีค่าเป็น 1 จะแสดงผลเป็นสีดำ และพิกเซลที่มีค่าเป็น 0 จะแสดงผลเป็นสีขาว ดังภาพที่ 1 ซึ่งในการประมวลภาพดิจิทัลโดยทั่วไปนั้นจะนิยมแปลงภาพสี RGB ให้เป็นภาพระดับสีเทา แล้วแปลงข้อมูลภาพระดับสีเทาให้เป็นภาพสองระดับเพื่อลดข้อมูลในภาพลงซึ่งจะทำให้ประมวลผลได้เร็วขึ้นและยังลดเนื้อที่ในการเก็บข้อมูลได้อีกด้วย

1 0

2 3

รูปที่ 2.9 แสดงรูปสองระดับ

การสร้างภาพสองระดับนั้น (Binary Image) สามารถทำได้โดยการกำหนดค่า Threshold ซึ่งมีค่าอยู่ระหว่าง 0-255 โดยมีเงื่อนไขว่าถ้าความเข้มแสงของจุดภาพใดมีค่าต่ำกว่าหรือเท่ากับค่า Threshold ให้จุดภาพนั้นมีค่าเป็น 0 หรือเป็นสีดำ และจุดภาพใดมีค่าสูงกว่าค่า Threshold ให้จุดภาพนั้นมีค่าเป็น 1 หรือสีขาว เช่น การกำหนดค่า Threshold เท่ากับ 127 ซึ่งเป็นค่าช่วงกลางๆของภาพระดับสีเทา จุดพิกเซลที่มีค่าระดับสีเทามากกว่า 127 จะเปลี่ยนจากสีเทาเป็นสีขาว ส่วนจุดพิกเซลที่มีค่าระดับสีเทาน้อยกว่า 127 จะเปลี่ยนจากสีเทาเป็นสีดำ โดยมีสูตรดังนี้

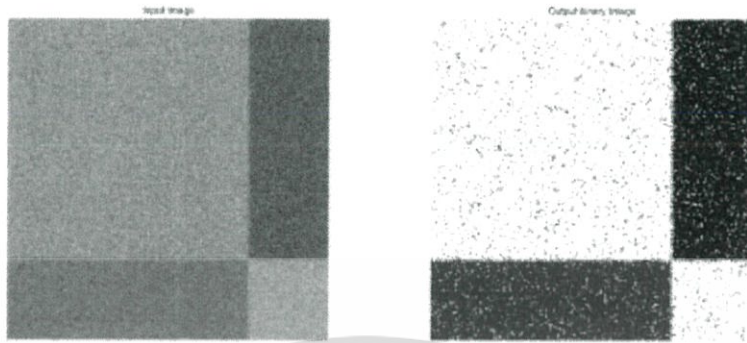
$$\text{Grayscale} = \begin{cases} 255 & ; \text{ if grayscale} > 127 \\ 0 & ; \text{ else} \end{cases}$$

แสดงความสัมพันธ์ได้ดังรูปที่ 2.10



รูปที่ 2.10 แสดงรูปความสัมพันธ์ของภาพสองระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



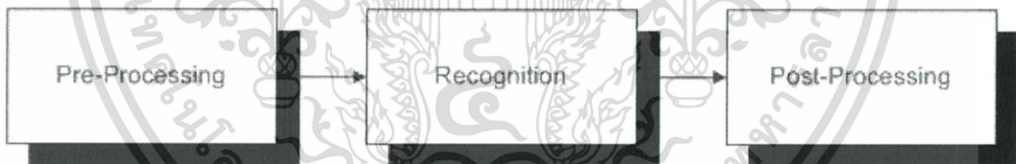
รูปที่ 2.11 แสดงรูปในระดับสีเทา รูปที่ 2.12 แสดงรูปที่มีการแบ่ง Threshold

2.12 Optical Character Recognition (OCR)

Optical character recognition หรือที่เรียกกันสั้นๆว่า OCR คือ การแปลงภาพตัวอักษร ให้เป็นข้อความโดยอัตโนมัติ

2.12.1 โครงสร้างระบบโอซีอาร์

โครงสร้างของระบบโอซีอาร์โดยทั่วไปประกอบไปด้วยขั้นตอนการทำงานหลัก 3 ขั้นตอน



รูปที่ 2.13 แสดงรูปขั้นตอนการทำงานของ OCR

2.12.1.1 ขบวนการประมวลผลขั้นต้น (Pre-Processing)

ในการทำงานของโปรแกรมโอซีอาร์นั้น ก่อนที่โปรแกรมจะสามารถบอกได้ว่ารูปภาพที่ส่งเข้าไปประกอบด้วยตัวอักษรอะไรบ้าง จำเป็นจะต้องผ่านขั้นตอนที่สำคัญหลายขั้น ขั้นตอนดังกล่าวนี้มักถูกเรียกรวมกันว่า กระบวนการประมวลผลขั้นต้น (Pre-Processing) ซึ่งเป็นขั้นตอนในการปรับแต่งและจัดเตรียมข้อมูลให้เหมาะสมกับขั้นตอนการรู้จำต่อไป ได้แก่

1.) การกรองข้อมูลแทรกซ้อน (Noise Filtering)

การกรองข้อมูลแทรกซ้อนมีจุดประสงค์เพื่อลดทอนส่วนของรูปภาพที่เป็นสิ่งแปลกปลอมอันไม่พึงประสงค์ออกไป โดยข้อมูลแทรกซ้อนที่เป็นต้นเหตุสำคัญที่ทำให้ความถูกต้องของโปรแกรมลดลง จึงจำเป็นที่จะต้องจัดการกับส่วนเกินเหล่านี้ออกไปให้ได้มากที่สุดเท่าที่จะเป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) การปรับแต่งข้อมูล (Normalization)

การปรับแต่งข้อมูลเป็นการปรับภาพตัวอักษรให้อยู่ในรูปแบบที่ระบบต้องการเพื่อนำไปใช้ในขั้นต่อไป เช่น การปรับขนาดรูปตัวอักษร, การปรับตัวอักษรที่เอียงให้ตรง, การแปลงรูปสี่หรือเกรย์สเกลให้เป็นขาวดำ เป็นต้น

3.) การตัดแบ่งพื้นที่ใช้งาน (Cropping)

การตัดแบ่งพื้นที่เป็นการตัดแยกเอาเฉพาะรูปตัวอักษรออกมาจากภาพ เพื่อส่งให้ขั้นตอนการรู้จำในการระบุตัวอักษรนั้นเป็นรหัสตัวอักษรอะไร หลักการพอสั่งเขตที่ใช้สำหรับการตัดรูปตัวอักษรโดยทั่วไปจะใช้พื้นที่สีขาว (สีพื้น) รอบรูปเป็นตัวกำหนดขอบเขตในการตัด

4.) การสกัดลักษณะสำคัญ (Feature Extraction)

การสกัดลักษณะสำคัญเป็นการดึงเอาโครงสร้างพื้นฐานที่สำคัญของตัวอักษรนั้นออกมา โดยโครงสร้างพื้นฐานที่ว่าจะต้องมีการกำหนดไว้ก่อนว่าจะมีอะไรบ้าง มีการนิยามอย่างไร เช่น สำหรับภาษาไทยอาจกำหนดว่าตัวอักษรภาษาไทยทั้งหมดประกอบด้วยโครงสร้างพื้นฐานคือ เส้นตรง (แนวตั้ง/นอน) เส้นเอียง หัว (วงกลม) ส่วนโค้ง ส่วนหัว จุดแตกกิ่ง จุดตัด เป็นต้น เมื่อแยกเอาองค์ประกอบของตัวอักษรแต่ละตัวออกมาได้แล้ว จากนั้นรูปภาพของตัวอักษรนั้นในรูปแบบของรายการขององค์ประกอบพื้นฐานต่างๆ แทน ซึ่งจะถูกส่งต่อเป็นอินพุตสำหรับขั้นตอนการรู้จำต่อไป

2.12.1.2 การรู้จำ (Recognition)

ขั้นตอนนี้ถือเป็นหัวใจของระบบ เพราะเป็นส่วนที่จะตัดสินว่ารูปตัวอักษรที่ส่งเข้าไปเป็นรหัสตัวอักษรอะไร โดยแนวทางการรู้จำสามารถแบ่งออกได้เป็น 4 กลุ่ม ดังนี้

1.) วิธีการเข้าคู่รูปแบบ (Template Matching)

วิธีการเข้าคู่รูปแบบเป็นวิธีการแรกๆ ที่มาใช้ในการรู้จำตัวอักษร หลักการโดยทั่วไปคือ จะต้องมีรูปแบบ (template) ที่สร้างขึ้นมาสำหรับอ่านตัวอักษร โดยมีการกำหนดตำแหน่งสำคัญที่สามารถใช้แยกแยะความแตกต่างระหว่างตัวอักษรแต่ละตัว เวลาทำงานก็ให้นำรูปภาพที่ต้องการอ่านไปหาบนแบบเพื่อวัดความคล้ายคลึงกันของภาพกับตัวแบบ จากนั้นก็ระบุว่าเป็นรหัสตัวอักษรอะไร โดยใช้ค่าผ่านระดับหรือวิธีการบางอย่างในการตัดสิน

2.) วิธีการสถิติ (Statistical Approach)

วิธีการสถิติเป็นวิธีการที่ใช้หลักการทางสถิติ โดยนำค่าความน่าจะเป็นหรือฟังก์ชันการแจกแจงความน่าจะเป็นมาใช้ในการตัดสินใจ รูปภาพอินพุตที่ได้มาจากขั้นตอนการสกัดลักษณะสำคัญ จะถูกส่งเข้าไปในส่วนการรู้จำเฉพาะของแต่ละตัวอักษร ซึ่งได้ผลลัพธ์ออกมาเป็นค่าความน่าจะเป็นที่อินพุตเป็นตัวอักษรใด เมื่ออินพุตได้ผ่านส่วนการรู้จำครบทุกตัวแล้ว ก็นำเอาผลลัพธ์ที่ได้ทั้งหมดมาเปรียบเทียบกับว่าได้ค่าความน่าจะเป็นของตัวอักษรใดมากที่สุด ผลลัพธ์จะออกเป็นตัวอักษรนั้น

3.) วิธีการวิเคราะห์ทางโครงสร้าง (Structural Analysis)

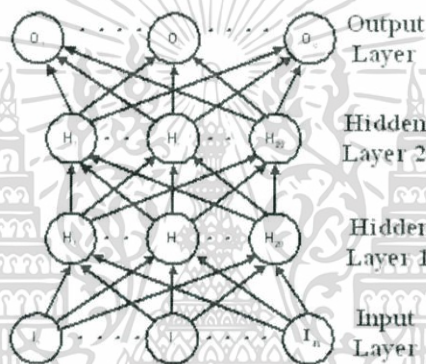
วิธีการวิเคราะห์ทางโครงสร้างคือการวิเคราะห์โครงสร้างตัวอักษร โดยถือว่าตัวอักษรทุกตัวประกอบด้วยองค์ประกอบพื้นฐาน ซึ่งได้มาจากการสกัดลักษณะสำคัญ เช่นเดียวกันกับวิธีการทางสถิติต่างกันตรงที่ลักษณะสำคัญ ที่ส่งมาให้กับขั้นตอนการรู้จำแบบการวิเคราะห์ทางโครงสร้างนี้ มักจะใช้เป็นชื่อหรือค่าที่บอกว่าคุณลักษณะโครงสร้างสำคัญนั้นเป็นอะไร เช่น เส้นตรง วงกลม เป็นต้น ในขั้นตอนการรู้จำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะสำคัญทั้งหลายที่ประกอบเป็นตัวอักษรนั้น จะถูกส่งเข้าไปให้กับส่วนที่ตรวจวิเคราะห์กฎการเขียนตัวอักษร เพื่อระบุว่าเป็นตัวอะไร ซึ่งจะตัดสินใจโดยการดูที่รูปแบบการเชื่อมต่อขององค์ประกอบต่างๆ เข้าเป็นตัวอักษรนั้น

4.) วิธีทางโครงข่ายประสาทเทียม (Neural Network)

วิธีทางโครงข่ายประสาทเทียมเป็นเทคนิคที่พยายามเลียนแบบการทำงานของสมองมนุษย์ ที่มีโครงข่ายเชื่อมต่อกันของหน่วยความจำย่อยๆ จำนวนมากที่สะสมความรู้เอาไว้ ความรู้เหล่านี้จะได้รับการฝึกสอนไว้ก่อน เช่นการสอนให้รู้จักตัวอักษร “ก” ถึง “ฮ” โดยการส่งภาพตัวอักษรเหล่านี้เข้าไปพร้อมกับบอกว่ามีค่าเป็นรหัสตัวอักษรอะไร โครงข่ายประสาทเทียมจะเรียนรู้ถึงรูปแบบตัวอักษรที่หลากหลายของตัวอักษรตัวนั้น เพื่อว่าเวลาทำงานจริงจะได้มีความสามารถพอที่จะรับมือกับภาพตัวอักษรในหลายๆ รูปแบบ อินพุตที่ส่งให้จะผ่านขั้นตอนการสกัดลักษณะสำคัญ และกระบวนการประมวลผลเบื้องต้นอื่นๆ ก่อนเสมอ



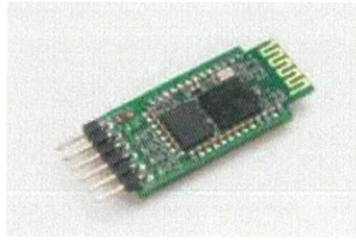
รูปที่ 2.14 แสดงรูปโครงสร้างของ Neural Network

2.12.1.3 การประมวลผลขั้นปลาย(Post-Processing)

หลังจากที่ผ่านขั้นตอนการรู้จำแล้ว รูปตัวอักษรที่ถูกส่งเข้าไปได้ผลลัพธ์ออกมาเป็นรหัสตัวอักษร ซึ่งก็ไม่ได้หมายความว่าเอาท์พุตที่ได้มาจะถูกต้องทั้งหมด ดังนั้นเพื่อเพิ่มความถูกต้องให้แก่โปรแกรมจึงได้มีการเสริมส่วนการตรวจสอบและแก้ไขข้อความเข้ามา

ส่วนขบวนการประมวลผลขั้นปลายในโปรแกรมโอซีอาร์เป็นส่วนที่มีความสำคัญมาก ดังนั้นส่วนนี้สามารถเพิ่มอัตราความถูกต้องให้แก่โปรแกรมได้ โดยเฉพาะในส่วนที่นอกเหนือจากความสามารถของส่วนการรู้จำ

2.13 บลูทูธโมดูล



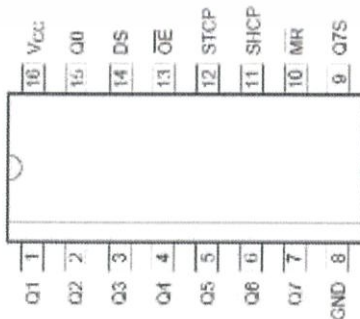
รูปที่ 2.15 แสดงรูปบลูทูธโมดูล

มีความสามารถสื่อสารแบบไร้สายโดยใช้บลูทูธ ใช้งานง่ายด้วยอินเตอร์เฟซพอร์ตอนุกรม เขียนโปรแกรมง่ายเหมือนส่งผ่านพอร์ตอนุกรมทั่วไป จ่ายไฟ 3.3V แล้วต่อขา RX กับ TX ไปที่ไมโครคอนโทรลเลอร์ สามารถส่งงานควบคุมจากอุปกรณ์ที่มีบลูทูธ เช่น เครื่องคอมพิวเตอร์หรือโทรศัพท์มือถือที่มีบลูทูธ

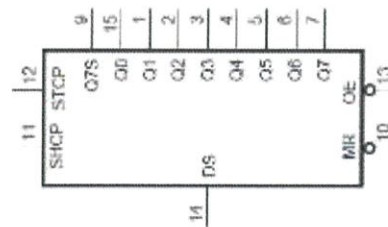
2.14 ชิพตรีจิสเตอร์ (74HC595)

ไอซี 74HC595 เป็นไอซีลอจิก (Logic IC) ที่ทำหน้าที่เป็นวงจรถอนบิตขนาด 8 บิต (8-bit Shift Register) โดยรับข้อมูลเข้าและส่งข้อมูลออกทีละบิต (Serial-In & Serial-Out) และมีเอาต์พุตแบบ 8 บิต (Parallel Data Output) ถ้าต้องการจะส่งข้อมูลไปกำหนดสถานะลอจิกให้ขาเอาต์พุตทั้ง 8 ขา (Q0 ถึง Q7) จะต้องใช้วิธีเลื่อนบิตทีละบิต เข้าที่ขาอินพุต DS (Serial Data Input) และสร้างสัญญาณ Clock ป้อนเข้าที่ขา SHCP (Shift Clock Input) เพื่อกำหนดจังหวะในการทำงาน และสามารถเลือกใช้ความถี่ของสัญญาณอินพุต SHCP ได้เกินกว่า 10MHz สัญญาณ SHCP หนึ่งไซเคิล (Clock Cycle) หมายถึงการเลื่อนบิตหนึ่งครั้ง ดังนั้นจึงต้องใช้ทั้งหมด 8 ไซเคิล สำหรับ 8 บิต ถ้าเลื่อนบิตมากกว่า 8 ไซเคิล ข้อมูลในรีจิสเตอร์ ก็จะถูกส่งออกมาตามลำดับที่ขา Q7S

การเลื่อนบิตเพื่อส่งข้อมูลจะส่งแบบ MSB First เมื่อข้อมูลถูกนำไปใส่ไว้ในรีจิสเตอร์ภายในจนครบ 8 บิตแล้ว และต้องการให้ขาเอาต์พุต Q0 ถึง Q7 มีสถานะลอจิกตรงตามข้อมูลในรีจิสเตอร์ จะต้องสร้างสัญญาณพัลส์ เข้าที่ขา STCP (Storage Clock Input) และต้องกำหนดขาอินพุต OE (Output Enable) ให้เป็นลอจิก LOW ถ้าต้องการจะเคลียร์ข้อมูลในรีจิสเตอร์ ให้ป้อนอินพุตที่ขา MR (Master Reset) เป็น LOW แล้วจึงเปลี่ยนเป็น HIGH



รูปที่ 2.16 แสดงรูปตำแหน่งขาของไอซี 74HC595



รูปที่ 2.17 แสดงรูปขาทั้งหมดของไอซี 74HC595

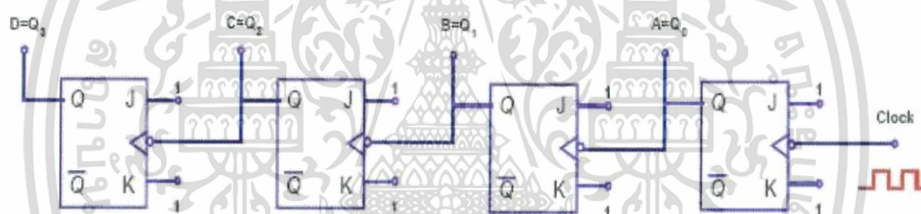
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15 วงจรนับ (Counter)

วงจรถับ คือ เกิดจากการนำเอาฟลิปฟล็อปที่ต่อกันเป็นวงจรมาใช้งาน นับจำนวนคล็อก (Clock) หรือพัลส์ (Pulse) ที่ป้อนเข้าทางอินพุต หรือบางที่อาจเรียกว่าวงจรถับความถี่ หลักการทำงานของวงจรถับส่วนใหญ่จะเป็นวงจรถับแบบเลขฐานสอง วงจรถับสามารถแบ่งออกเป็น 2 แบบใหญ่ๆ คือ

2.15.1 วงจรถับแบบไม่เข้าจังหวะ (Asynchronous Counter หรือ Ripple Counter)

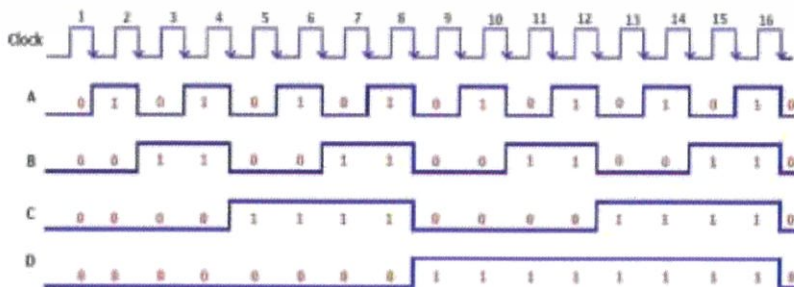
โดยพื้นฐานจะใช้ J-K Flip Flop มาต่อเรียงกันดังวงจรถับในรูปข้างล่าง สถานะเอาต์พุตของฟลิปฟล็อปแต่ละตัว ขึ้นอยู่กับสถานะเอาต์พุตของฟลิปฟล็อปตัวก่อนหน้า คือ ฟลิปฟล็อปตัวแรกจะส่งสัญญาณ (Pulse) จาก Q ไปกระตุ้น (Trigger) ที่ Clk ของฟลิปฟล็อปตัวที่สอง และฟลิปฟล็อปตัวที่สองจะส่งสัญญาณไปกระตุ้น (Trigger) ที่ Clk ของฟลิปฟล็อปตัวที่สาม ไปเรื่อยๆ ตามขั้นตอนการทำงานของวงจรถับแบบนี้มีลักษณะไหลเป็นระลอก จึงทำให้มีชื่ออีกอย่างหนึ่งว่า วงจรถับแบบริบเบิล (Ripple Counter) วงจรถับแบบไม่เข้าจังหวะ (Asynchronous Counter) ที่ได้ศึกษามาแล้วนั้นใช้ได้ดีกับความถี่ต่ำๆ เนื่องจากปัญหาความล่าช้าของสัญญาณเอาต์พุตหรือเกิดเวลาหน่วงในการส่งข้อมูลจากอินพุตไปยังเอาต์พุต ของฟลิปฟล็อปแต่ละตัว



รูปที่ 2.18 แสดงรูปวงจรถับแบบไม่เข้าจังหวะ

2.15.2 วงจรถับแบบเข้าจังหวะ (Synchronous Counter หรือ Parallel Counter)

คือการทำงานของวงจรถับที่ความถี่สูงๆ จำเป็นที่จะต้องให้ฟลิปฟล็อปหลายๆตัวต่อกัน ได้รับสัญญาณกระตุ้นไปพร้อมๆ กัน เพื่อหลีกเลี่ยงปัญหาที่เกิดจากการหน่วงเวลาของฟลิปฟล็อป เป็นวงจรถับที่เอาต์พุตของฟลิปฟล็อป ที่เปลี่ยนแปลงพร้อมๆ กันตามสัญญาณคล็อก (Clock) วงจรถับชนิดนี้เรียกว่า วงจรถับแบบเข้าจังหวะ (Synchronous Counter)



รูปที่ 2.19 แสดงรูปวงจรถับแบบเข้าจังหวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16 ระบบบัส SPI

การเชื่อมต่อด้วยโหมด SPI เป็นการรับส่งข้อมูลขนาด 8 บิต แบบซิงโครนัส โดยสามารถกำหนดการใช้งานขาพอร์ตได้ทั้งแบบ 3 ขา และ 4 ขา ขึ้นอยู่กับการเลือกใช้งาน โดยแต่ละขาพอร์ตมีหน้าที่ดังนี้

- Serial Data Out (SDO)
ขาพอร์ตส่งข้อมูล (อุปกรณ์สแลฟ) ต่อเข้ากับขาพอร์ตรับข้อมูล (อุปกรณ์มาสเตอร์) สำหรับ PIC 16F877A คือขาพอร์ต RC5/SDO
- Serial Data In (SDI)
ขาพอร์ตรับข้อมูล (อุปกรณ์สแลฟ) ต่อเข้ากับขาพอร์ตส่งข้อมูล (อุปกรณ์มาสเตอร์) สำหรับ PIC16F877A คือขาพอร์ต RC4/SDI/SDA
- Serial Clock (SCK)
ขาพอร์ตกำหนดสัญญาณนาฬิกา (อุปกรณ์มาสเตอร์) สำหรับ PIC16F877A คือขาพอร์ต RC3/SCK/SCL/LVDIN
- Slave Select (/SS)
ขาเลือกอุปกรณ์สแลฟสำหรับ PIC16F877A คือขาพอร์ต RA5/SS/AN4 (เชื่อมต่อแบบ 4 ขา)

บทที่ 3

วิธีการดำเนินงาน

การออกแบบการแสดงผลบนบอร์ดแสดงผลแอลอีดีนั้น ในเบื้องต้นต้องเข้าใจถึงหลักการและการทำงานภาพรวมของระบบการแสดงผลก่อนว่าจะรับอินพุตจากที่ใดบ้าง ซึ่งโครงการนี้ได้กำหนดให้มีอินพุต 4 ทาง คือ รับค่าจากคีย์บอร์ดคอมพิวเตอร์ ภาพถ่ายจากกล้องเว็บแคม ไฟล์รูปภาพและคีย์บอร์ดโทรศัพท์มือถือในระบบแอนดรอยด์ จากนั้นจึงกำหนดช่องทางการสื่อสารในการรับส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ ที่ใช้ในการควบคุมการแสดงผลของบอร์ดแสดงผลแอลอีดี ซึ่งแสดงให้เห็นภาพรวมดังรูปที่ 3.1



รูปที่ 3.1 แสดงแผนผังการทำงานของบอร์ดแสดงผลแอลอีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับปริญญาโทฉบับนี้สามารถแบ่งส่วนการทำงานได้ 2 ส่วน คือ ทางด้านฮาร์ดแวร์ และ ทางด้านซอฟต์แวร์

- ทางด้านฮาร์ดแวร์ ประกอบไปด้วย
 1. LED Display Board ขนาด 8x8 บิต
 2. วงจร Control LED Display Board
 3. พาวเวอร์ซัพพลาย
 4. Module แปลง USB เป็น พอร์ตอนุกรม TTL
 5. บลูทูธโมดูล
- ทางด้านซอฟต์แวร์ ประกอบไปด้วย
 1. โปรแกรม Microsoft Visual Studio 2010
 2. โปรแกรม Eclipse
 3. โปรแกรม MikroC Compiler

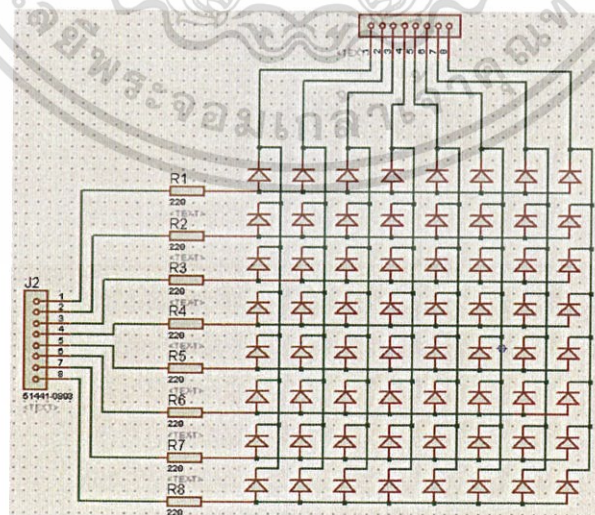
3.1 ฮาร์ดแวร์

3.1.1 LED Display Board ขนาด 8x8 บิต

ส่วนประกอบหลัก

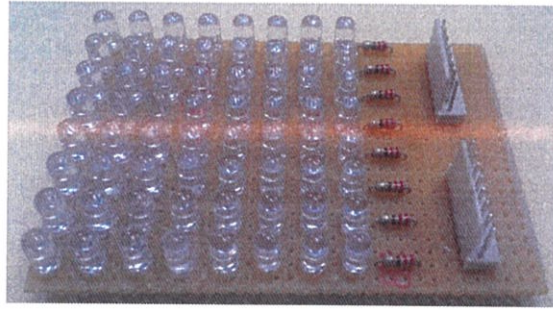
- | | | | |
|-------------------------|----|-------|-------------|
| 1. หลอดแอลอีดี 3mm | 64 | หลอด | ต่อ 1 บอร์ด |
| 2. ตัวต้านทาน 220 โอห์ม | 8 | ตัว | ต่อ 1 บอร์ด |
| 3. คอนเน็คเตอร์ 8 pin | 2 | ตัว | ต่อ 1 บอร์ด |
| 4. บอร์ดไข่ปลา | 1 | บอร์ด | ต่อ 1 บอร์ด |

โดย LED Display Board มีหน้าที่แสดงภาพ โดยรับคำสั่งจาก PIC16F877A

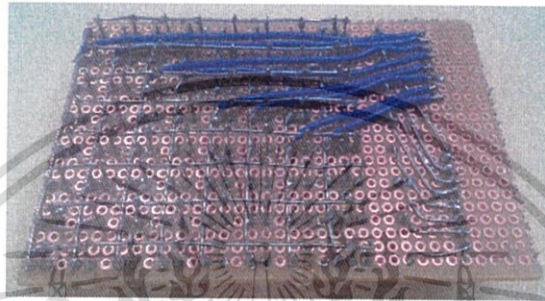


รูปที่ 3.2 แสดงรูป Schematic ของ LED Display Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงรูปด้านบนของ LED Display Board ขนาด 8x8 บิต



รูปที่ 3.4 แสดงรูปด้านล่างของ LED Display Board ขนาด 8x8 บิต

3.1.2 วงจร Control LED Display Board

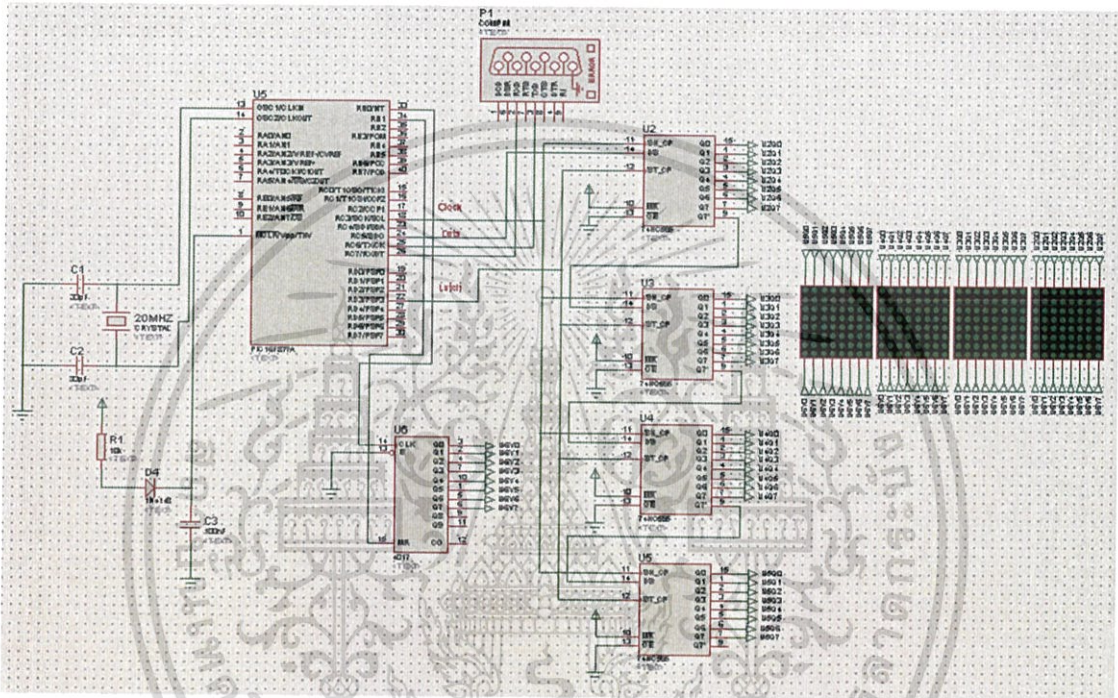
ส่วนประกอบหลัก

1. PIC16F877A	1	ตัว	10. คอนเน็คเตอร์ 2 Pin	1	ตัว
2. ตัวเก็บประจุ 33 pF	2	ตัว	11. คอนเน็คเตอร์ 5 Pin	1	ตัว
3. ตัวเก็บประจุ 100 nF	1	ตัว	12. คอนเน็คเตอร์ 8 Pin	1	ตัว
4. คริสตัล 20 MHz	1	ตัว	13. ซี็อกเก็ต 16 Pin	1	ตัว
5. ตัวต้านทาน 10k ohm	1	ตัว	14. ซี็อกเก็ต 40 Pin	1	ตัว
6. ไดโอด 1N4148	1	ตัว	15. สายไฟ		
7. IC Counter CD4017	1	ตัว	16. บลูทูธโมดูล		
8. ชิฟตรีจิสเตอร์ 74HC595	1	ตัว	17. โมดูลแปลง USB เป็นพอร์ตอนุกรม		
9. คอนเน็คเตอร์ 4 Pin	1	ตัว			
10. ซี็อกเก็ต 16 Pin	1	ตัว			
11. ซี็อกเก็ต 40 Pin	1	ตัว			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

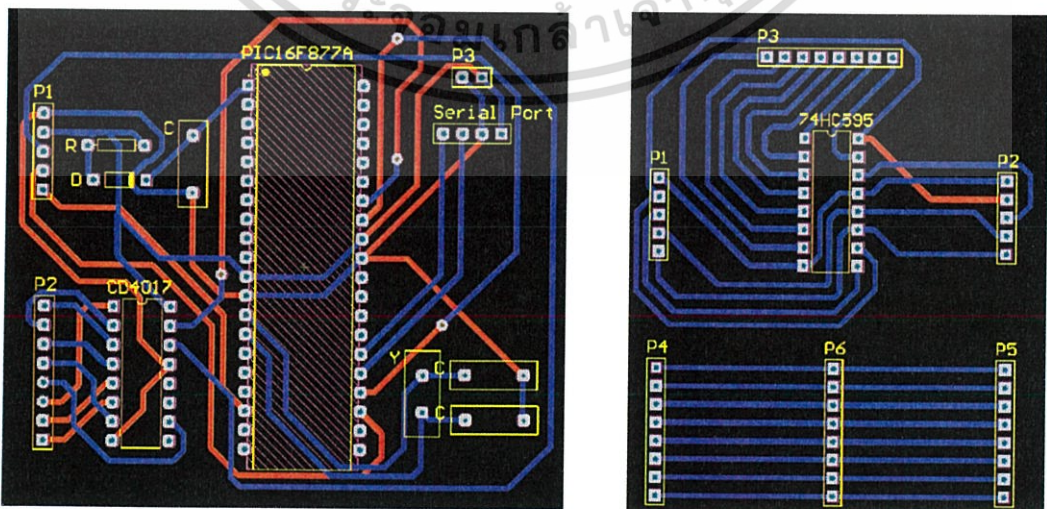
หลักการทํางานของวงจร Control LED Display Board

เมื่อต่อตัวคอนเน็คเตอร์เข้าที่ไมโครคอนโทรลเลอร์ PIC16F877A โดยขาแอนอด(Anode)ของหลอดแอลอีดีต่อกับ Port B ของ PIC16F877A และขาแคโทด(Cathode) ต่อกับ Port D ของ PIC16F877A ซึ่งหลอดแอลอีดี จะติดก็ต่อเมื่อไมโครคอนโทรลเลอร์ส่งเอาท์พุตเป็น Logic 1 ออกที่ Port B และส่งเอาท์พุตเป็น Logic 0 ออกที่ Port D และ หลอดแอลอีดี จะดับก็ต่อเมื่อไมโครคอนโทรลเลอร์ส่งเอาท์พุตเป็น Logic 0 ออกที่ Port B และส่งเอาท์พุตเป็น Logic 0 ออกที่ Port D



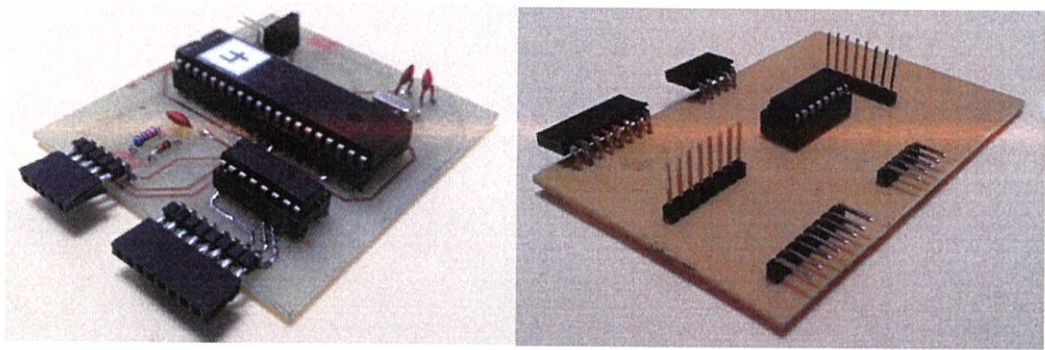
รูปที่ 3.5 แสดงรูป Schematic ของวงจร Control LED Display Board

โดยทำการออกแบบวงจร Control LED Display Board โดยใช้โปรแกรม Altium Designer 14.3



รูปที่ 3.6 แสดงรูป PCB ของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงรูปบอร์ด Control LED Display Board

3.1.3 พาวเวอร์ซัพพลาย

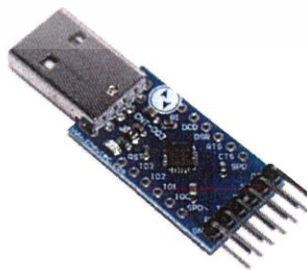
พาวเวอร์ซัพพลายที่ใช้ในโครงการคือ พาวเวอร์ซัพพลายขนาด 5 VDC สำหรับป้อนไฟให้บอร์ด Control LED Display Board



รูปที่ 3.8 แสดงรูปพาวเวอร์ซัพพลาย

3.1.4 Module แปลง USB เป็น พอร์ตอนุกรม TTL

ในโครงการนี้ใช้ โมดูล USB to TTL UART (CP2102) สำหรับส่งข้อมูลจากคอมพิวเตอร์ผ่านพอร์ตอนุกรม

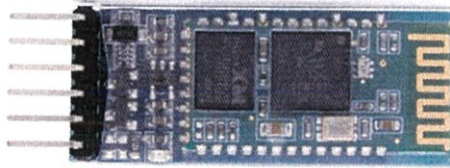


รูปที่ 3.9 แสดงรูปโมดูล USB to TTL UART (CP2102)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

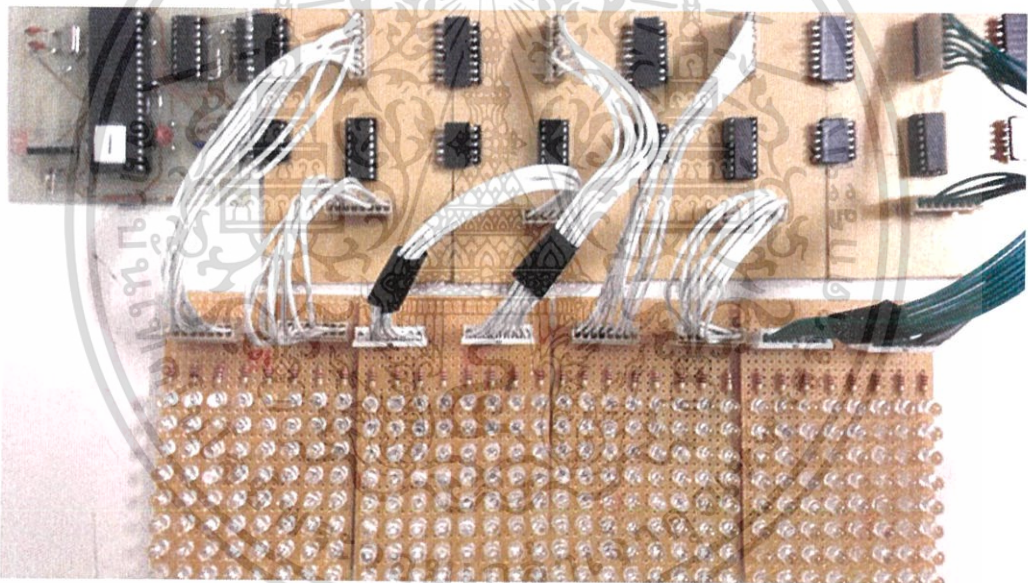
3.1.5 บลูทูธโมดูล

บลูทูธที่นำมาใช้เป็นบลูทูธโมดูล HC-05 นำมาใช้ในการส่งข้อมูลแบบไร้สายจาก โทรศัพท์มือถือ และ คอมพิวเตอร์



รูปที่ 3.10 แสดงรูปบลูทูธโมดูล HC-05

เมื่อนำส่วนประกอบฮาร์ดแวร์ทุกส่วนมาประกอบกัน และทำการเดินสายไฟ จะได้ดังรูป 3.11

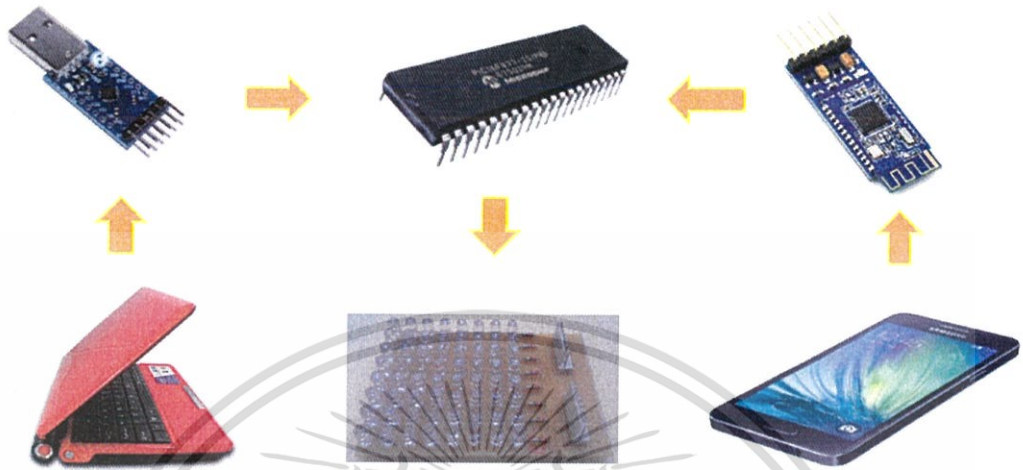


รูปที่ 3.11 แสดงรูปส่วนประกอบฮาร์ดแวร์ทุกส่วนมาประกอบกัน และทำการเดินสายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ซอฟต์แวร์

ซอฟต์แวร์ในโครงการฉบับนี้ประกอบด้วย 3 ส่วน คือ โปรแกรม Microsoft Visual Studio 2010 โปรแกรม Eclipse และ โปรแกรม MikroC Compiler



รูปที่ 3.12 แสดงรูปการทำงานของโปรแกรม

ขั้นตอนการทำงานของโปรแกรม

ในส่วนของการแสดงผลนั้นสามารถรับข้อมูลได้ 2 ช่องทางคือ จากคอมพิวเตอร์และโทรศัพท์มือถือ

3.2.1 ส่วนโปรแกรม Microsoft Visual Studio 2010

แบ่งเป็น 2 โปรแกรม คือ Text Recognition (English Version) และ Text Recognition (Thai Version)

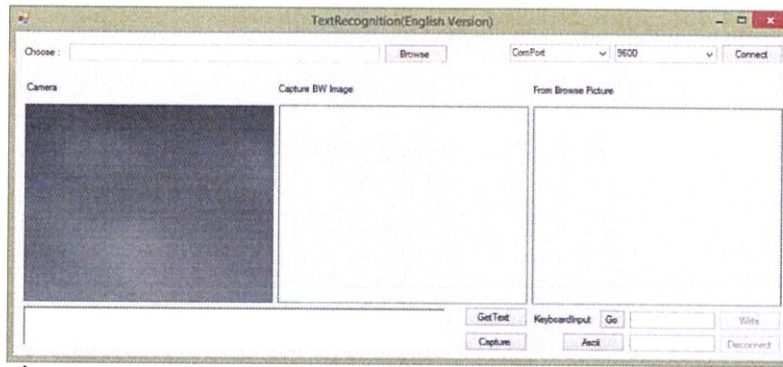
โปรแกรม Text Recognition (English Version) มีการใช้งาน 3 แบบ คือ

- 1.) ถ่ายภาพจากกล้องเว็บแคม
- 2.) เลือกไฟล์รูปภาพ
- 3.) พิมพ์ตัวอักษรจากคีย์บอร์ด

โดยการเริ่มต้นใช้โปรแกรมให้สามารถใช้งานได้ต้องทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ดังขั้นตอนต่อไปนี้

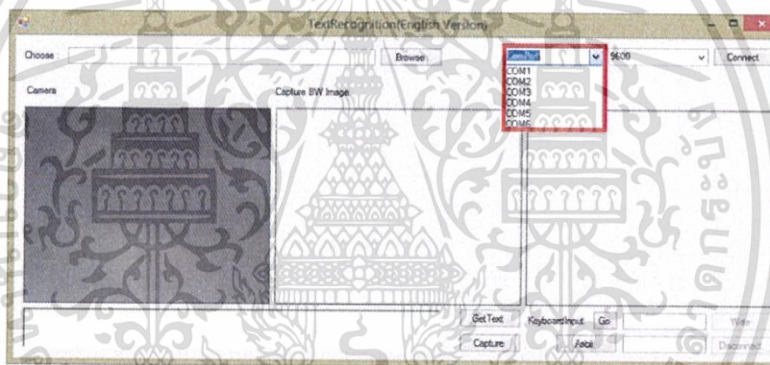
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โดยเมื่อเปิดโปรแกรมขึ้น หน้าจอเริ่มต้น ก่อนเริ่มทำงาน ดังรูปที่ 3.13



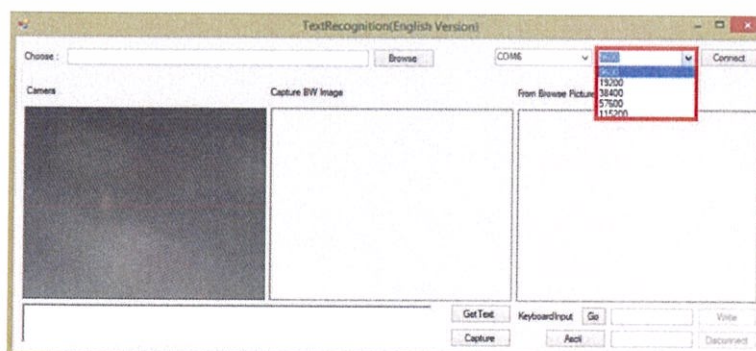
รูปที่ 3.13 แสดงรูปหน้าจอเริ่มต้นโปรแกรม Text Recognition (English Version)

2. ทำการเลือก Comport เนื่องจากในคอมพิวเตอร์แต่ละเครื่องพอร์ตอนุกรมจะไม่เหมือนกัน โครงการงานเราจึงทำปุ่มที่สามารถเลือกพอร์ตอนุกรมได้ จึงทำการเลือกพอร์ตอนุกรมให้ตรงกับเครื่องคอมพิวเตอร์ โดยกดปุ่ม 'ComPort' ดังรูปที่ 3.14



รูปที่ 3.14 แสดงรูปหน้าต่างโปรแกรม เลือกพอร์ตอนุกรม

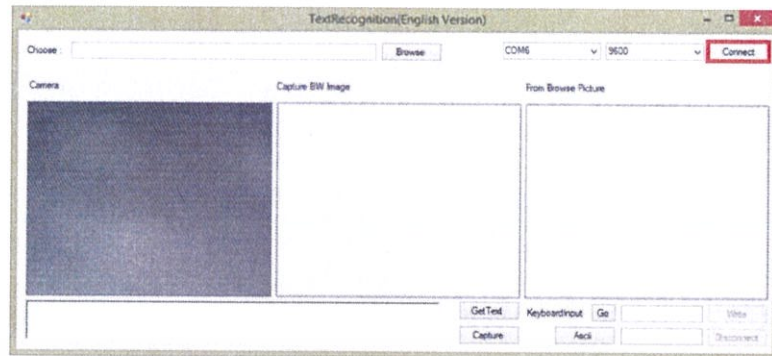
3. เมื่อเลือก Comport เสร็จแล้ว ให้เลือก Baud rate ที่ใช้งานเพื่อจะได้ใช้ Baud rate ที่ตรงกับไมโครคอนโทรลเลอร์ ดังรูปที่ 3.15



รูปที่ 3.15 แสดงรูปหน้าต่างโปรแกรม เลือก Baud rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อเลือก Comport และ Baud rate เสร็จแล้ว จึงทำการเชื่อมต่อกับ ไมโครคอนโทรลเลอร์โดยกดปุ่ม 'Connect' ดังรูปที่ 3.16

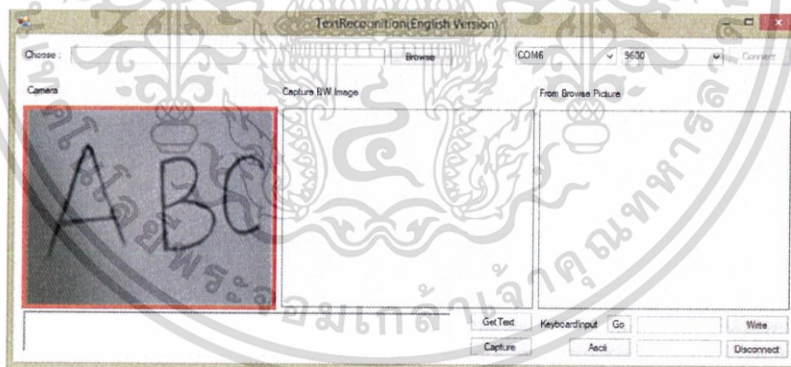


รูปที่ 3.16 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม 'Connect'

หลังจากเชื่อมต่อกับไมโครคอนโทรลเลอร์เรียบร้อยแล้วก็จะเข้ามาในส่วนการใช้งานคือ

3.2.1.1 การใช้กล้องเว็บแคมในการถ่ายภาพ

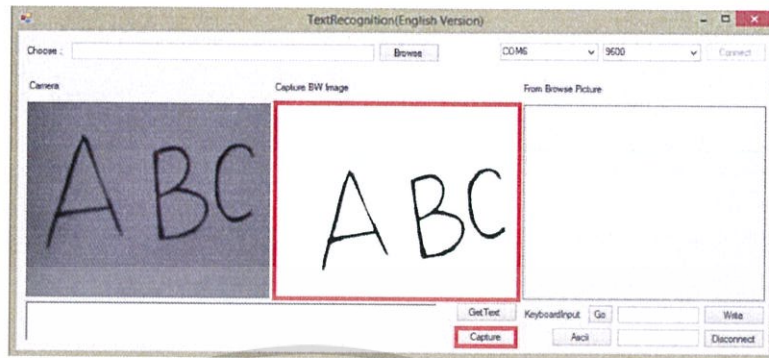
1. นำกล้องเว็บแคมไปจับภาพตัวอักษรที่ต้องการ โดยควรจะมีแสงที่เพียงพอ และพื้นหลังของตัวอักษรต้องเป็นสีอ่อน โดยรูปที่จะจับด้วยกล้องเว็บแคมจะแสดงอยู่ที่กล่องแสดงรูปภาพที่ชื่อว่า 'Camera' ดังรูปที่ 3.17



รูปที่ 3.17 แสดงรูปหน้าต่างโปรแกรมที่กำลังจะจับภาพด้วยกล้องเว็บแคม

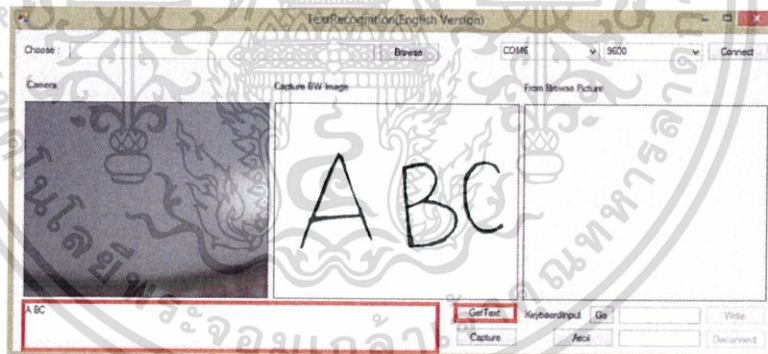
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อได้ภาพให้กดปุ่ม 'Capture' แล้วรูปที่ได้จะแสดงในกล่องแสดงรูปภาพที่ชื่อว่า 'Capture BW Image' โดยจะเป็นภาพที่จับภาพมาแล้วทำการแปลงภาพนั้นให้เป็นภาพสองระดับเพื่อให้ง่ายต่อตรวจรู้ตัวอักษร ดังรูปที่ 3.18



รูปที่ 3.18 แสดงรูปหน้าต่างโปรแกรม ที่จับภาพโดยกดปุ่ม 'Capture'

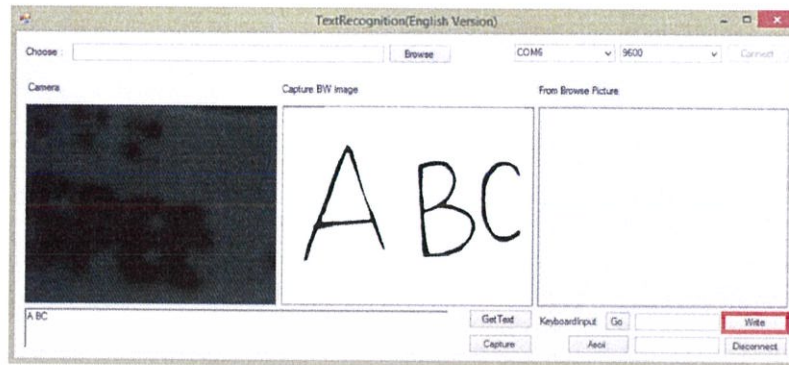
- เมื่อจับภาพและแปลงภาพเรียบร้อยแล้ว จากนั้นกดปุ่ม 'GetText' เพื่อทำการดูว่าตัวอักษรในภาพคือตัวอักษรอะไรบ้าง แล้วผลลัพธ์จะแสดงในกล่องข้อความด้านล่าง ดังรูปที่ 3.19



รูปที่ 3.19 แสดงรูปหน้าต่างโปรแกรม ที่กดปุ่ม 'GetText' และแสดงผลที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

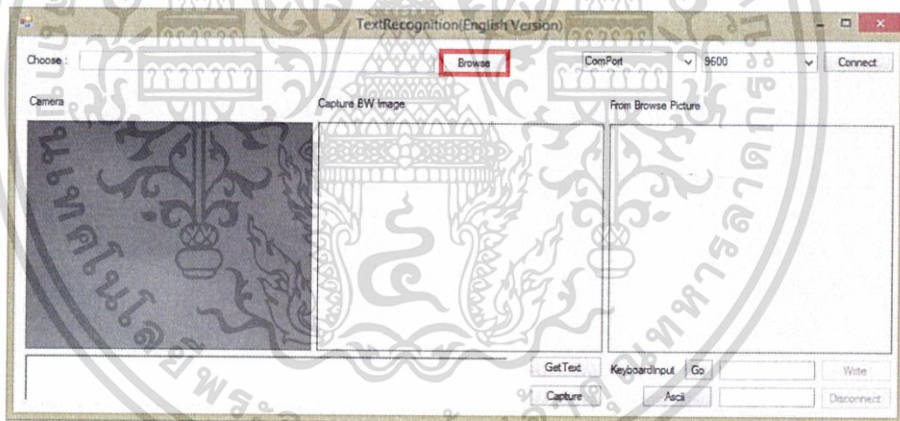
4. เมื่อได้ตัวอักษรที่ต้องการแล้ว จากนั้นกดปุ่ม 'Write' เพื่อส่งข้อมูลไปที่ ไมโครคอนโทรลเลอร์ เพื่อไปแสดงผลใน LED Display Board ดังรูปที่ 3.20



รูปที่ 3.20 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม 'Write'

3.2.1.2 การเลือกไฟล์รูปภาพ

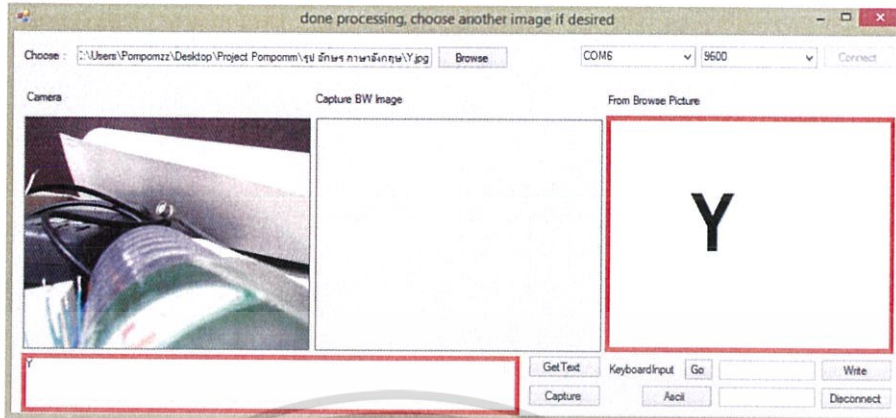
1. กดปุ่ม 'Browse' เพื่อเข้าไปเลือกไฟล์รูปภาพที่เราต้องการ ดังรูปที่ 3.21



รูปที่ 3.21 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม 'Browse'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อเลือกรูปภาพที่ต้องได้แล้ว รูปจะแสดงที่กล่องแสดงรูปภาพที่ชื่อว่า 'From Browse Picture' และ แสดงผลลัพธ์ที่กล่องข้อความด้านล่าง ดังรูปที่ 3.22



รูปที่ 3.22 แสดงรูปหน้าต่างโปรแกรมเมื่อเลือกไฟล์รูปภาพและแสดงผลลัพธ์

3. เมื่อได้ตัวอักษรที่ต้องการแล้ว จากนั้นกดปุ่ม 'Write' เพื่อส่งข้อมูลไปที่ไมโครคอนโทรลเลอร์ เพื่อแสดงผลใน LED Display Board ดังรูปที่ 3.23

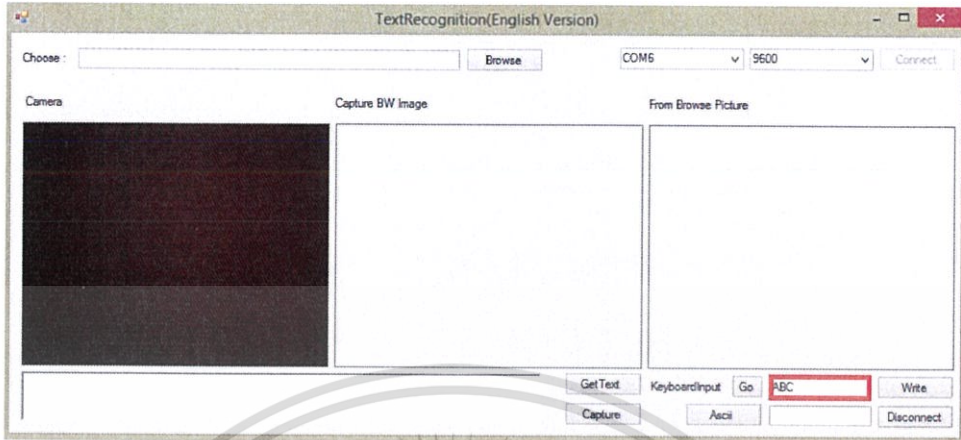


รูปที่ 3.23 แสดงรูปหน้าต่างโปรแกรมเมื่อกดปุ่ม 'Write'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

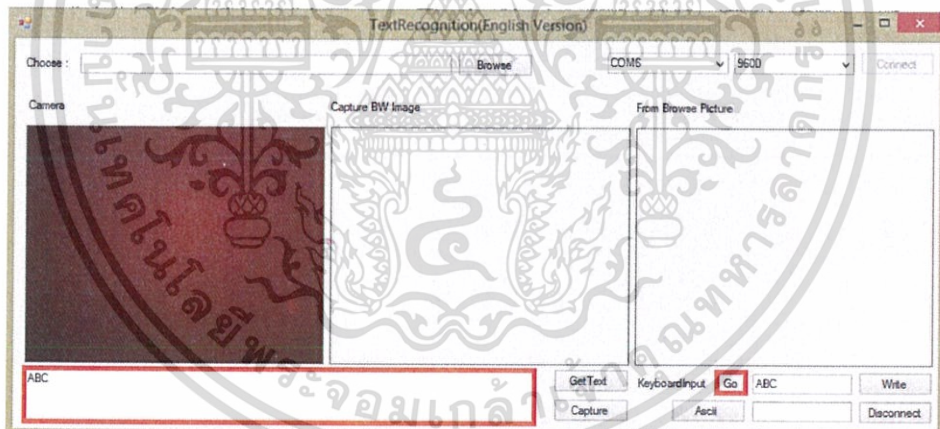
3.2.1.3 การพิมพ์ตัวอักษรจากคีย์บอร์ด

- พิมพ์ตัวอักษรที่ต้องการจะให้เห็นใน LED Display Board ลงในกล่องข้อความที่ชื่อว่า 'KeyboardInput' ดังรูปที่ 3.24



รูปที่ 3.24 แสดงรูปหน้าต่างโปรแกรม พิมพ์ 'KeyboardInput'

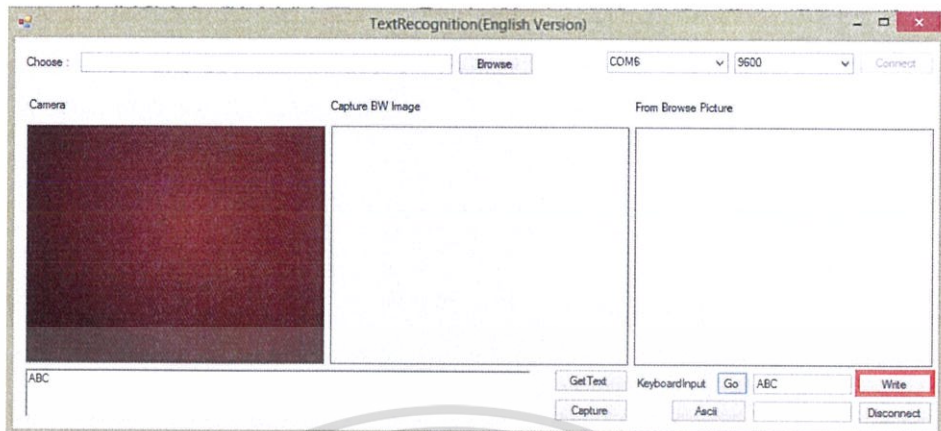
- เมื่อพิมพ์ตัวอักษรที่ต้องการลงในกล่องข้อความแล้ว กดปุ่ม 'Go' ตัวอักษรก็จะไปแสดงในกล่องข้อความด้านล่าง ดังรูปที่ 3.25



รูปที่ 3.25 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม 'Go'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

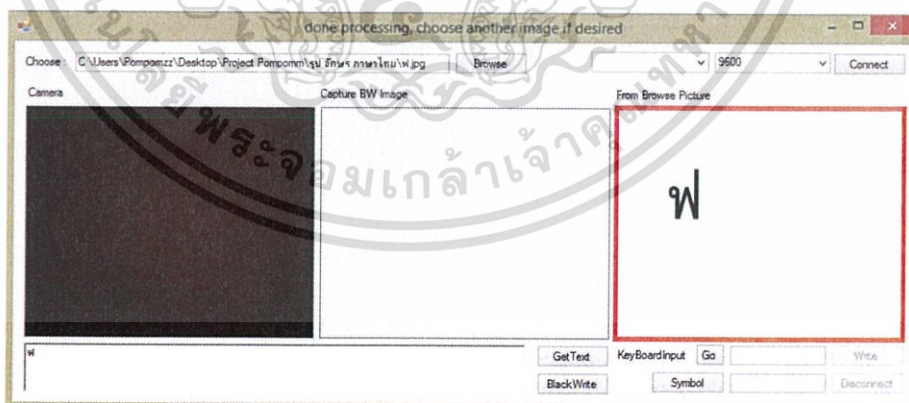
- เมื่อได้ตัวอักษรที่ต้องการแล้ว จากนั้นกดปุ่ม 'Write' เพื่อส่งข้อมูลไปที่ไมโครคอนโทรลเลอร์ เพื่อแสดงผลใน LED Display Board ดังรูปที่ 3.26



รูปที่ 3.26 แสดงรูปหน้าต่างโปรแกรม กดปุ่ม 'Write'

ส่วนของโปรแกรม Text Recognition(Thai Version) ก็จะมีวิธีใช้งานเหมือนกันเพียงแต่จะเพิ่มอีกส่วนหนึ่ง คือ เวลาส่งข้อมูลไปที่ไมโครคอนโทรลเลอร์ ตัวไมโครคอนโทรลเลอร์จะไม่สามารถรับตัวอักษรภาษาไทยได้ ดังนั้นจึงต้องทำการแปลงตัวอักษรภาษาไทยให้เป็นรหัสแอสกีและแปลงจากรหัสแอสกีไปเป็นสัญลักษณ์ต่างๆแทน แล้วจึงทำการส่งข้อมูลไปที่ไมโครคอนโทรลเลอร์ ดังขั้นตอนที่เพิ่มขึ้นต่อไปนี้

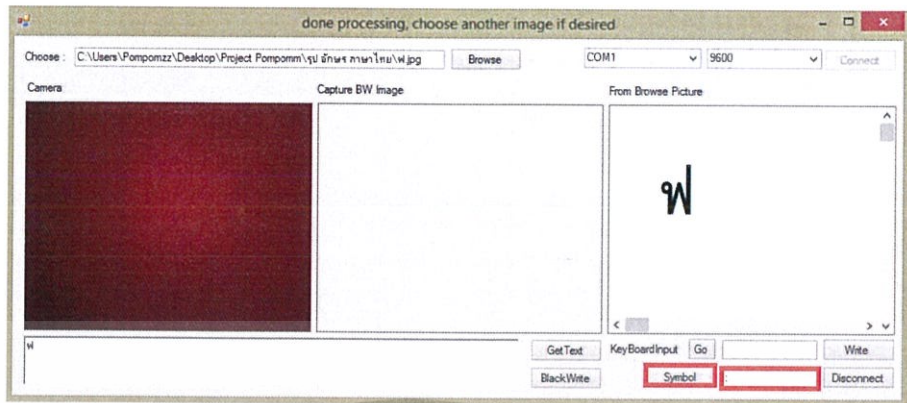
- เมื่อเราได้ตัวอักษรที่ต้องการแล้ว ดังรูปที่ 3.27



รูปที่ 3.27 แสดงรูปหน้าต่างโปรแกรม ตัวอักษรภาษาไทย

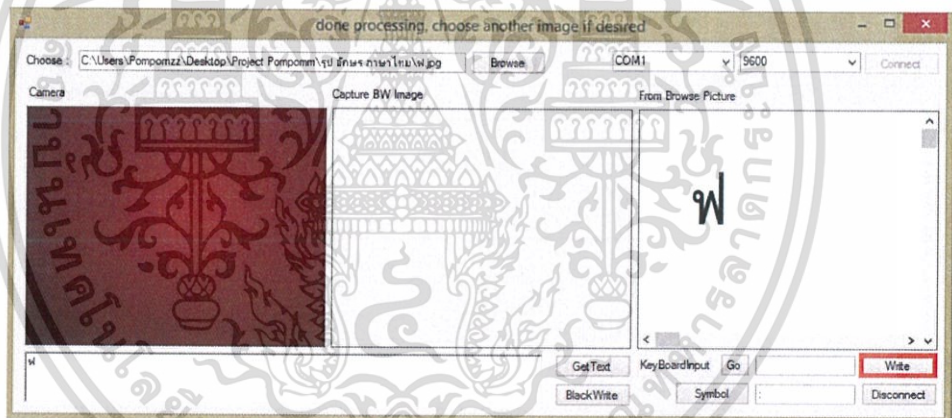
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กดปุ่ม 'Symbol' เพื่อแปลงตัวอักษรภาษาไทยให้เป็นสัญลักษณ์ ดังรูปที่ 3.28



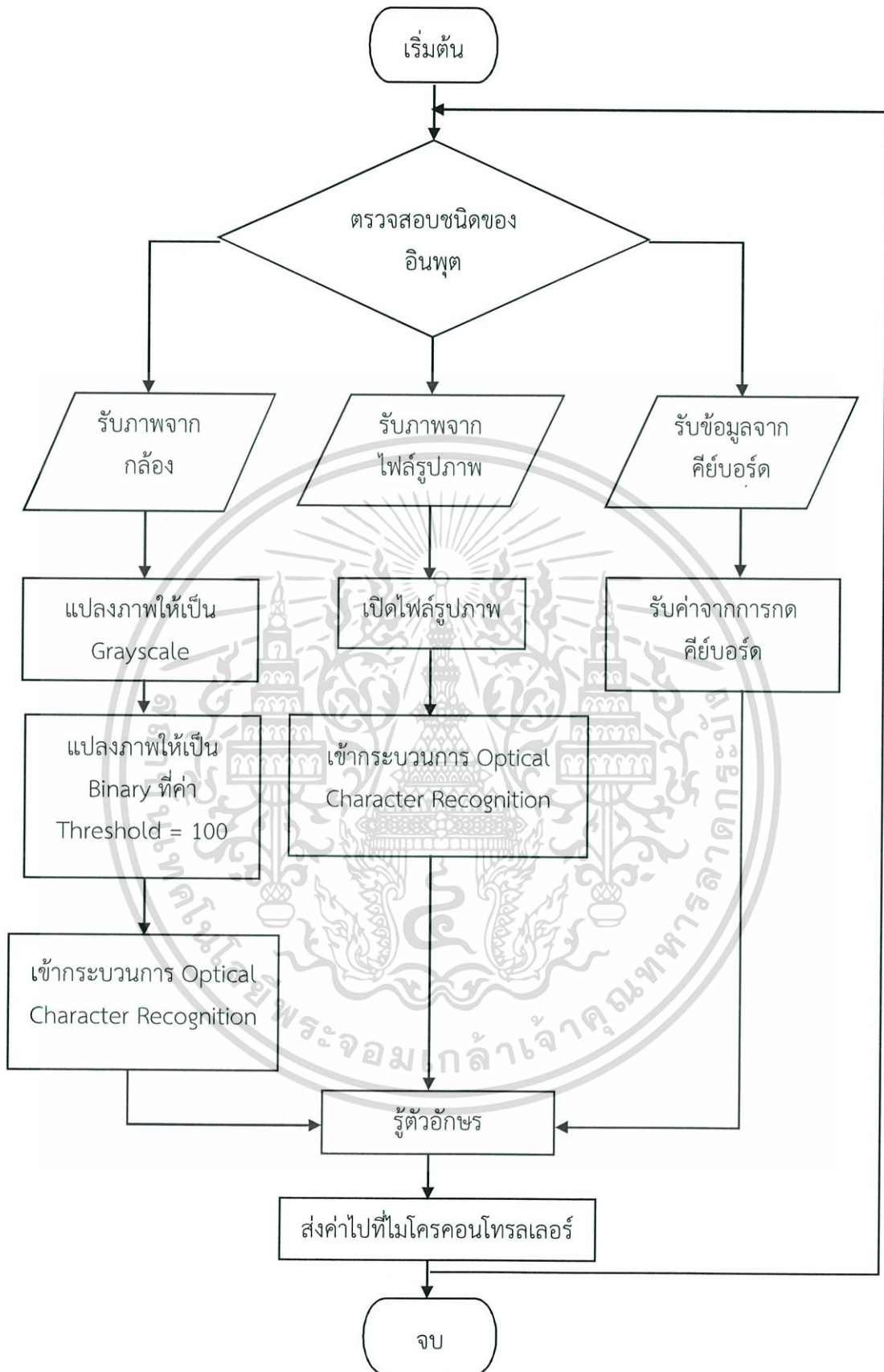
รูปที่ 3.28 แสดงรูปภาพต่างโปรแกรม กดปุ่ม 'Symbol'

3. เมื่อกดปุ่ม 'Symbol' แล้วจึงกดปุ่ม 'Write' เพื่อส่งข้อมูลไปยัง LED Display Board ดังรูปที่ 3.29



รูปที่ 3.29 แสดงรูปภาพต่างโปรแกรม กดปุ่ม 'Write'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



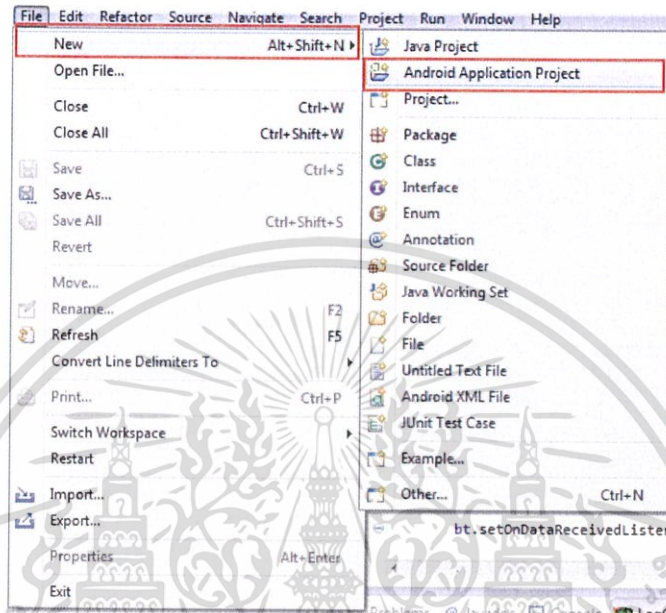
รูปที่ 3.30 Flow Chart แสดงขั้นตอนการทำงานของ Text Recognition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ส่วนโปรแกรม Eclipse

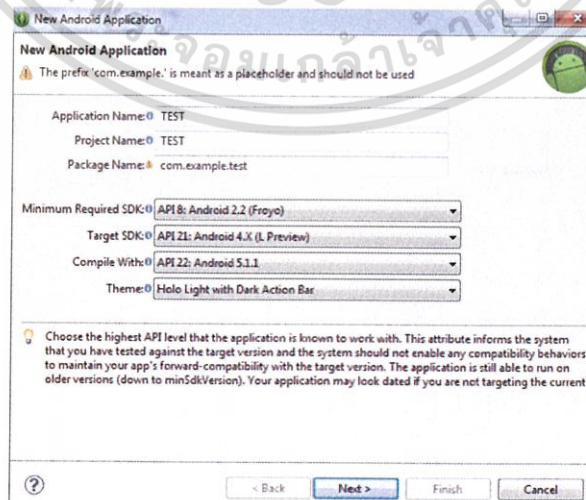
ในการส่งข้อมูลผ่านโทรศัพท์มือถือนั้น จะต้องมีการพัฒนา Android Application โดยการใช้โปรแกรม Eclipse ในการพัฒนา ซึ่งมีการขั้นตอนการสร้าง Android Application ดังนี้

1. ไปที่ File > New > Android Application Project ดังรูปที่ 3.31



รูปที่ 3.31 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการสร้าง Android Application Project

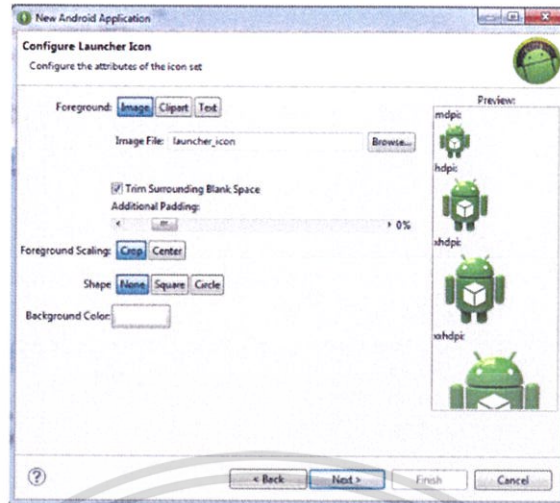
2. จากนั้นทำการใส่ Application Name, Project Name และเลือก Minimum Required SDK, Target SDK และ Compile With ตามที่ต้องการ ดังรูปที่ 3.32



รูปที่ 3.32 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการตั้งค่าต่างๆ

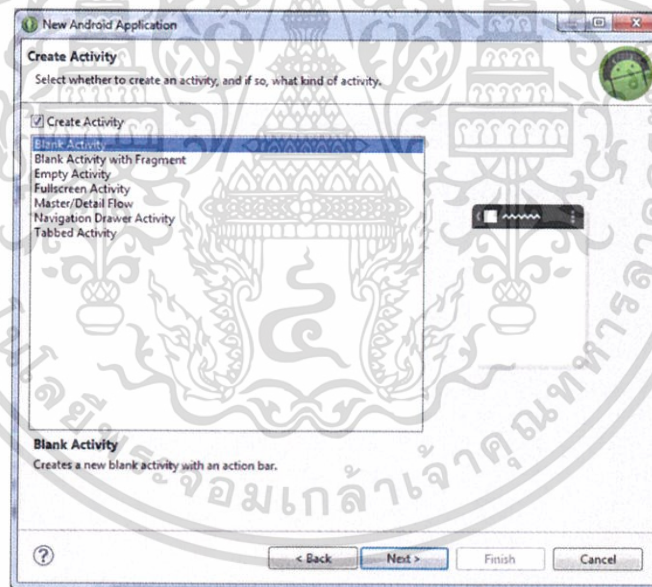
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการกำหนดขนาดและรูปร่างของไอคอนของแอปพลิเคชัน ดังรูปที่ 3.33



รูปที่ 3.33 แสดงหน้าต่างโปรแกรม Eclipse ในการกำหนดไอคอน

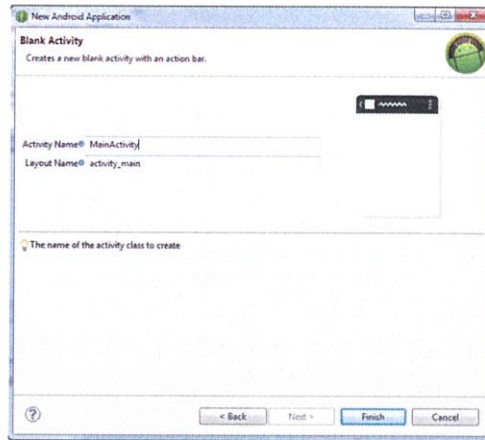
4. กำหนด Activity ของแอปพลิเคชันที่จะสร้าง ดังรูปที่ 3.34



รูปที่ 3.34 แสดงหน้าต่างโปรแกรม Eclipse ในการเลือก Activity

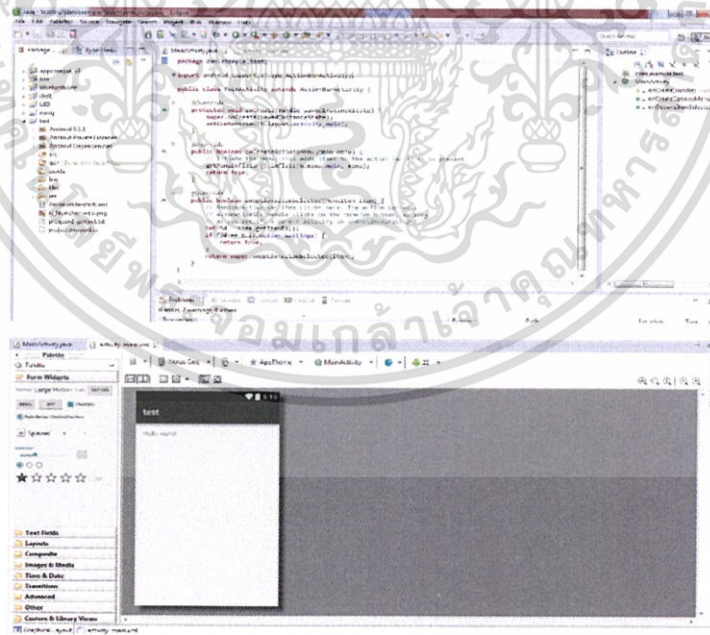
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. กำหนดชื่อของคลาส Activity ที่จะสร้าง ชื่อ Layout และชื่อ Fragment ดังรูป 3.35



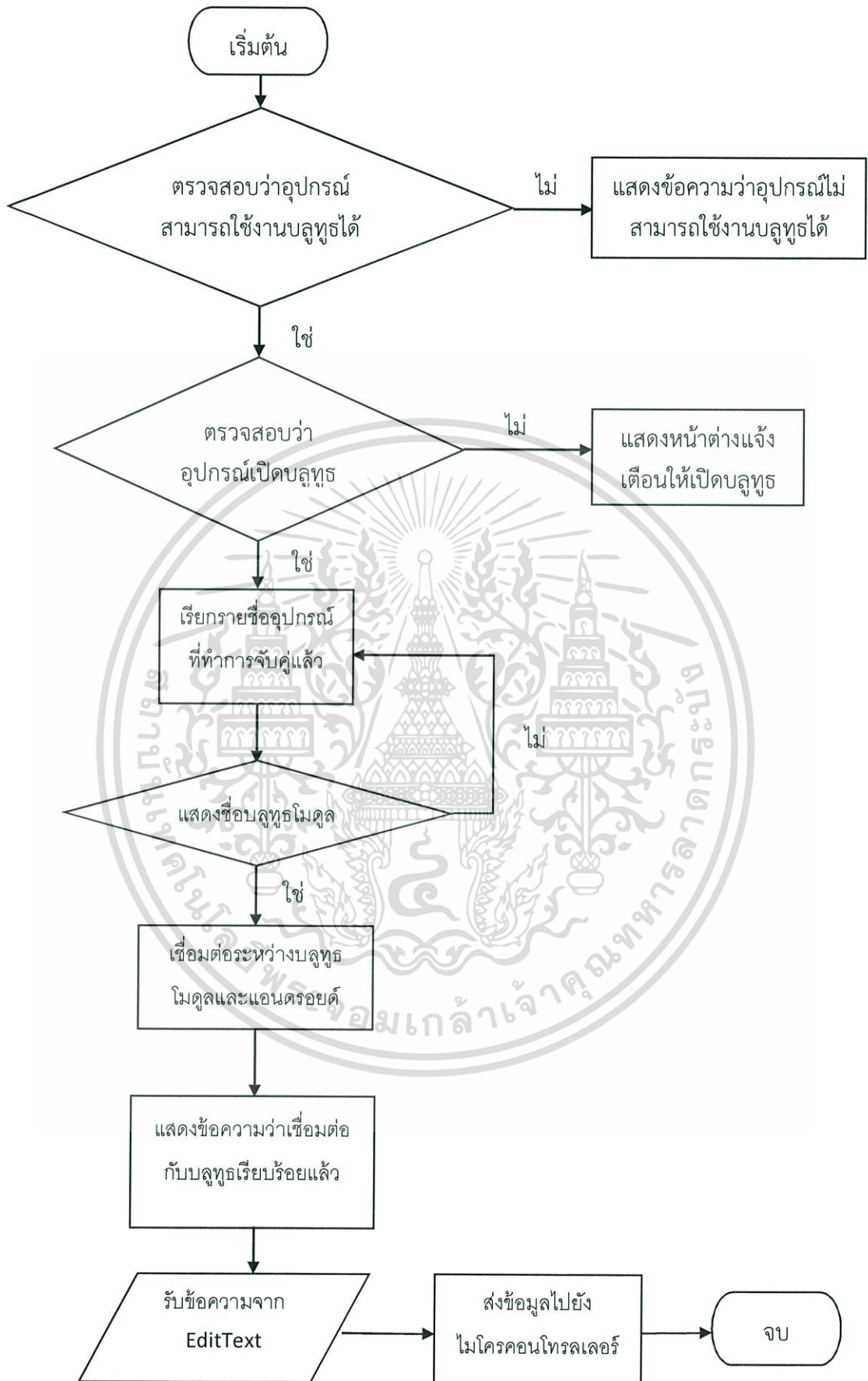
รูปที่ 3.35 แสดงรูปหน้าต่างโปรแกรม Eclipse ในการกำหนดชื่อ Activity Name และ Layout Name

6. เมื่อทำการสร้าง Android Application Project เสร็จเรียบร้อยแล้วจะแสดงหน้าต่าง MainActivity.java ซึ่งสามารถทำการแก้ไขโค้ดที่ใช้ควบคุมการทำงานของแอปพลิเคชัน และหน้าต่าง activity_main.xml ซึ่งแสดงหน้าต่างของ Android Application ดังภาพที่ 3.36



รูปที่ 3.36 แสดงรูปหน้าต่างโปรแกรม Eclipse เมื่อทำการสร้าง Android Application Project เสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



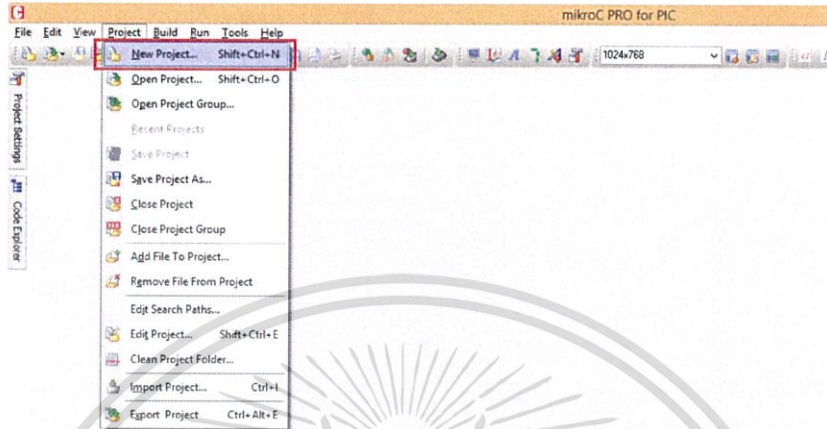
รูปที่ 3.37 Flow Chart แสดงขั้นตอนการทำงานของ Android Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การใช้งาน MikroC Compiler

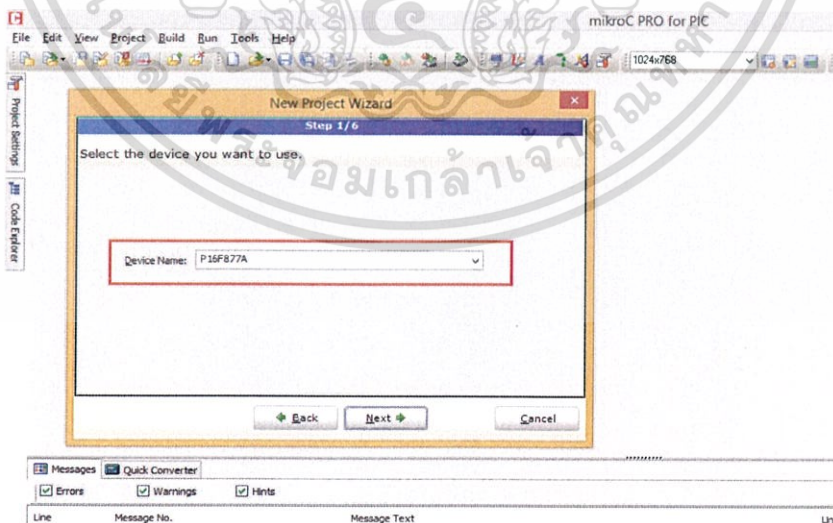
ขั้นตอนในการเขียนโปรแกรม

1. เมนู Project เลือก New Project เพื่อสร้างไฟล์ใหม่ และ เลือกโฟลเดอร์สำหรับจัดเก็บไฟล์ ดังรูปที่ 3.38



รูปที่ 3.38 โปรแกรม MikroC Compiler

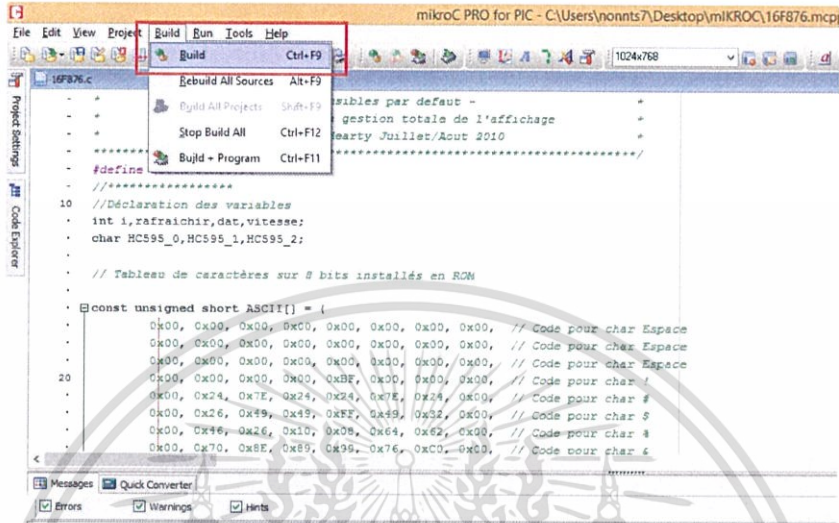
2. ทำการกำหนดรายละเอียดของไมโครคอนโทรลเลอร์ เบอร์ไมโครคอนโทรลเลอร์ที่ใช้และสัญญาณนาฬิกา



รูปที่ 3.39 หน้าต่างสำหรับกำหนดรายละเอียดของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หลังจากทำการเขียนโปรแกรมตามที่ต้องการเสร็จสิ้นแล้วจะต้องทำการ Compile โปรแกรม เพื่อตรวจสอบความถูกต้องของโค้ดที่ได้เขียนขึ้นมา และเป็นการสร้างไฟล์ .hex ที่จะนำไปใช้ สำหรับโปรแกรมลงไมโครคอนโทรลเลอร์ โดยไปที่เมนู Compile เลือกไอคอน Compile ดัง รูปที่ 3.40



รูปที่ 3.40 หน้าต่างเมื่อ Compile โปรแกรม

4. เมื่อ Compile โปรแกรมเสร็จสิ้นจะได้ไฟล์นามสกุล .hex ชื่อเดียวกับโปรเจกต์ที่ได้ตั้งชื่อไว้ สำหรับใช้โปรแกรมลงไมโครคอนโทรลเลอร์ ดังรูปที่ 3.41

TESTPORT	4/30/2015 12:07 AM	C Debug File	13 KB
TESTPORT	4/30/2015 12:07 AM	Error log	1 KB
TESTPORT.resym	4/30/2015 12:07 AM	ESYM File	10 KB
TESTPORT	4/26/2015 10:13 PM	H File	1 KB
TESTPORT	4/30/2015 12:07 AM	HEX File	1 KB
TESTPORT	4/30/2015 12:07 AM	LST File	4 KB
TESTPORT	4/30/2015 12:07 AM	PIT File	1 KB

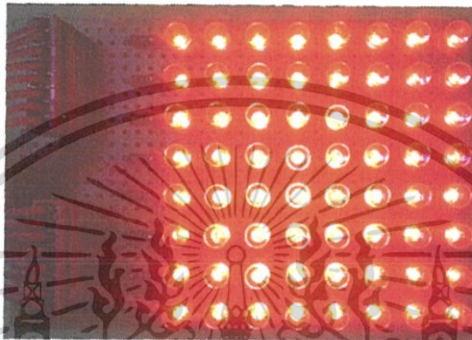
รูปที่ 3.41 ไฟล์นามสกุล .hex สำหรับใช้โปรแกรมลงไมโครคอนโทรลเลอร์

บทที่ 4

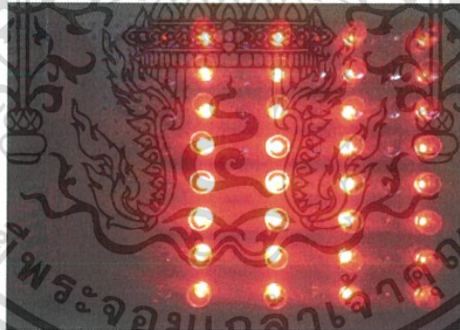
ผลการทดลอง

4.1 การทดสอบโปรแกรมสำหรับสั่งให้ LED Display Board แสดงค่าต่างๆ

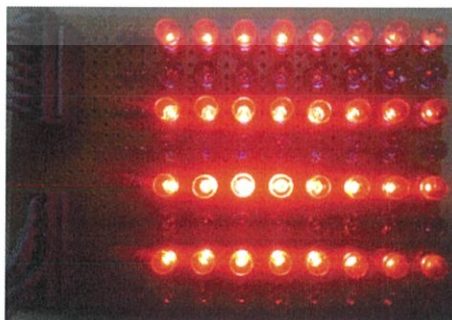
เป็นการเขียนโปรแกรมเพื่อให้ LED Display Board ติดในรูปแบบต่างๆกันไป



รูปที่ 4.1 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดครบทุกดวง

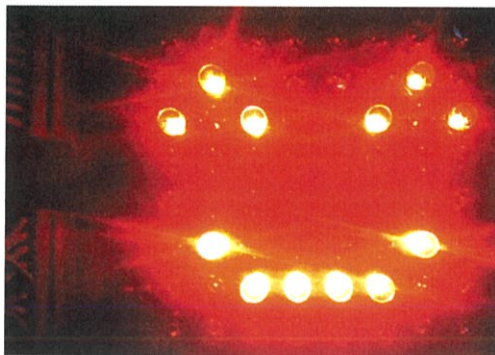


รูปที่ 4.2 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดหลักคู่ (2, 4, 6, 8)



รูปที่ 4.3 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดแถวคี่ (1, 3, 5, 7)

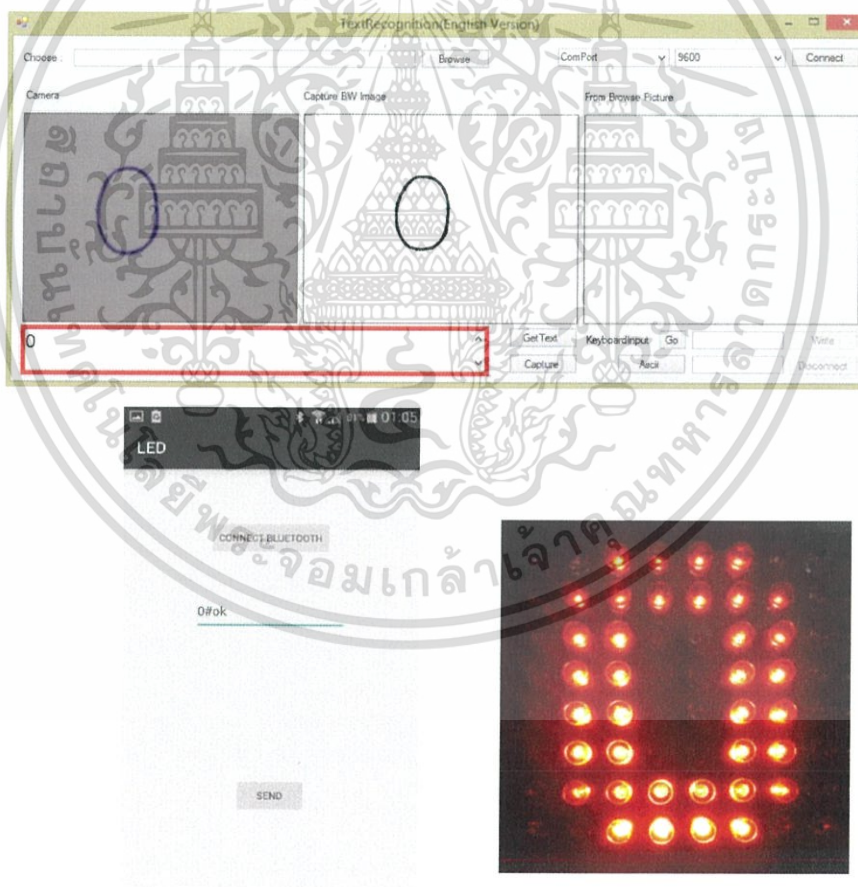
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงรูปการทดสอบโปรแกรมโดยการสั่งให้ติดเป็นรูปหน้ายิ้ม

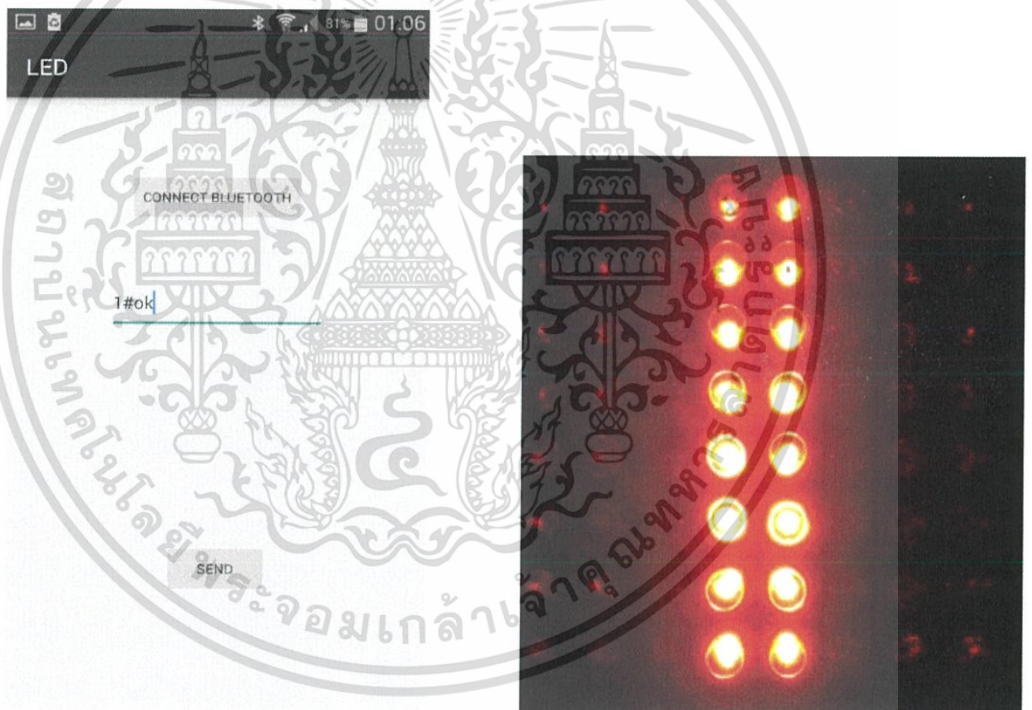
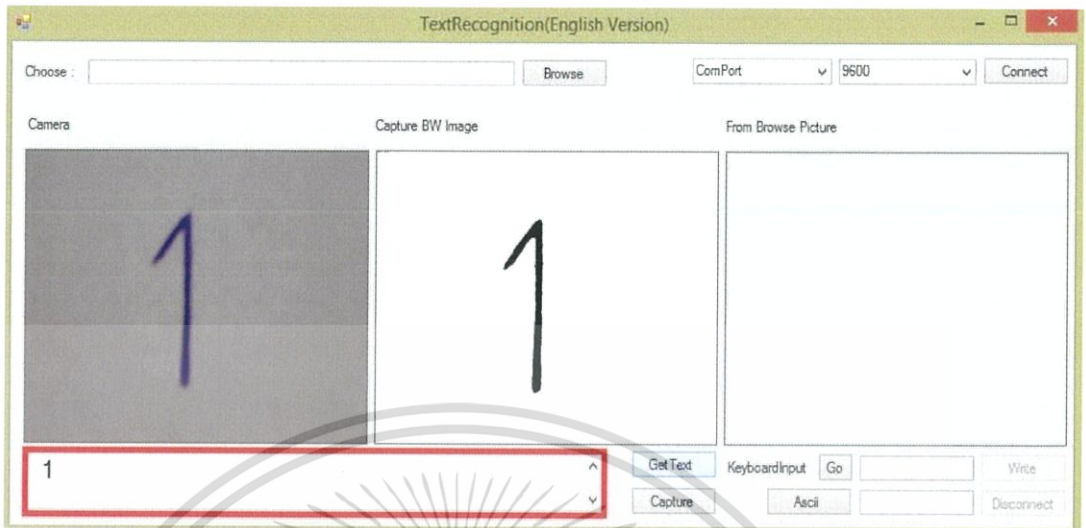
4.2 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ Bluetooth ซึ่งสามารถรับค่าจากโปรแกรม Text Recognition (English Version) และ android application

4.2.1 ส่งตัวเลขที่ละหนึ่งตัว



รูปที่ 4.5 แสดงรูปการส่งเลข 0 จากโปรแกรม Text Recognition (English Version) และ android application

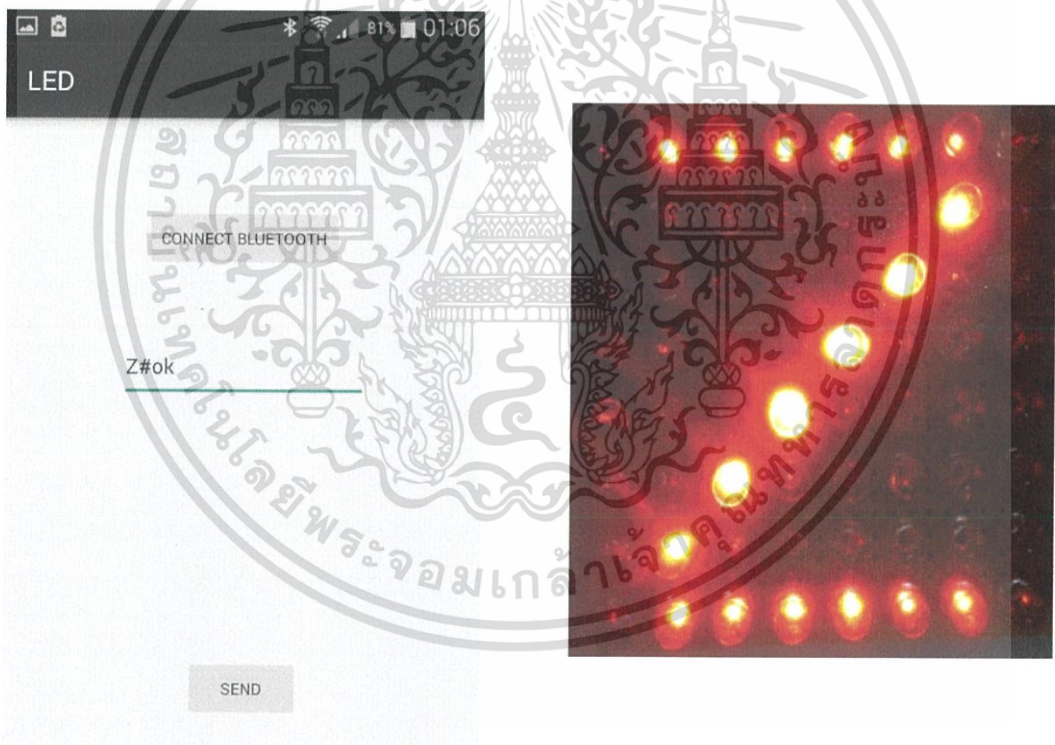
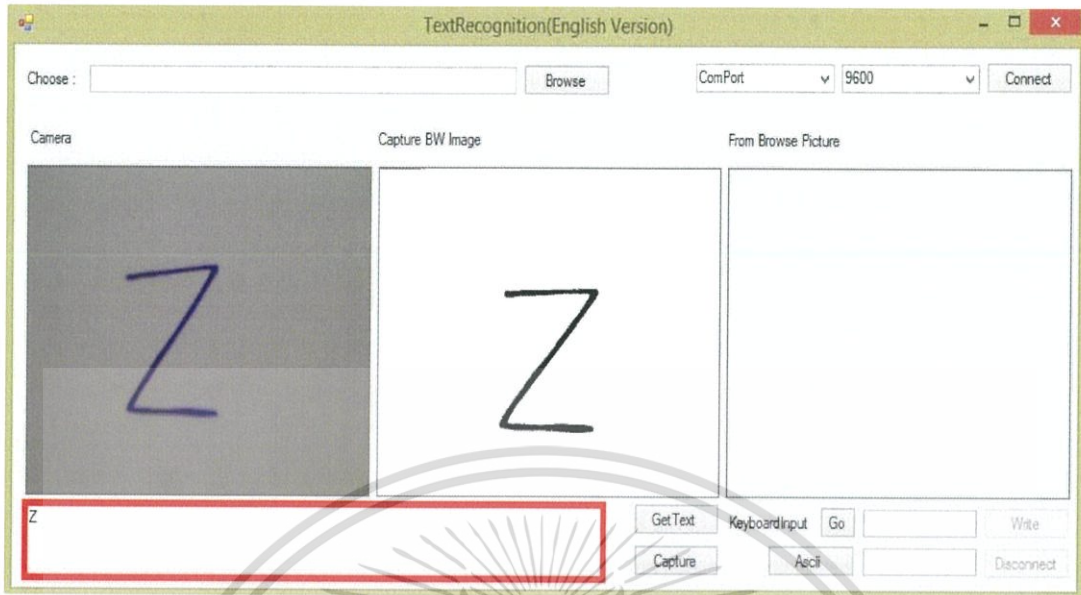
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงรูปการส่งเลข 1 จากโปรแกรม Text Recognition (English Version) และ android applicatio android application

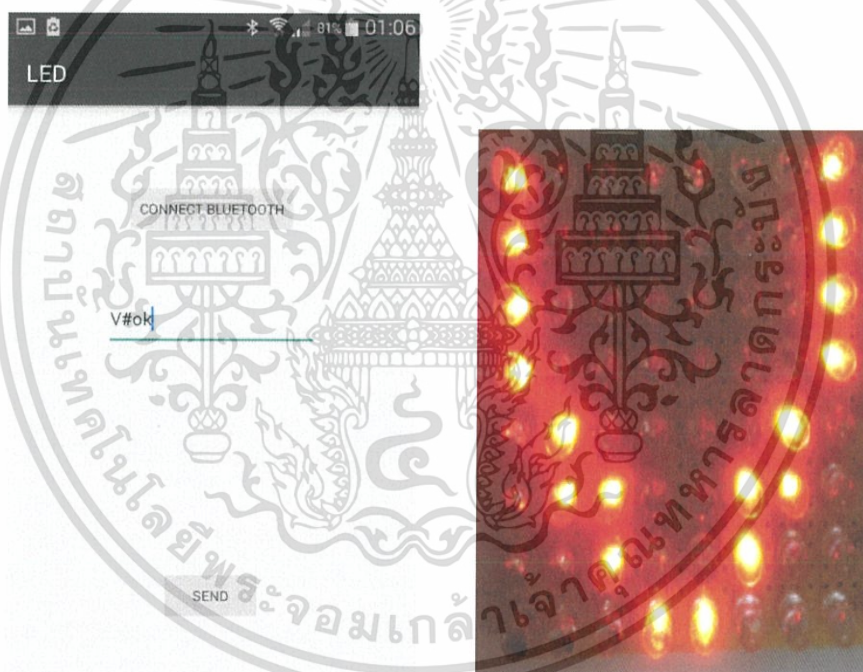
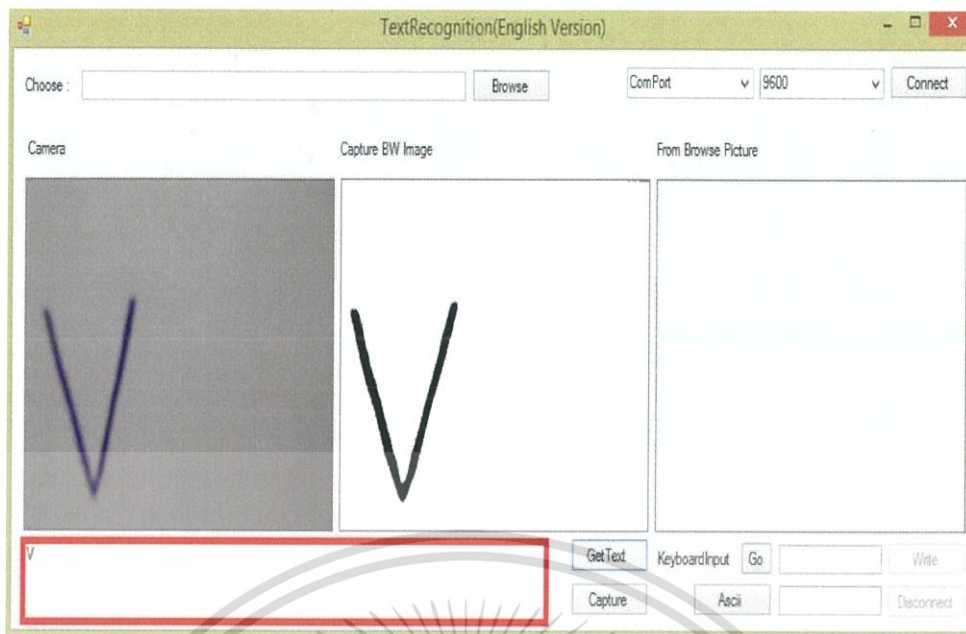
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ส่งตัวอักษรภาษาอังกฤษที่ละหนึ่งตัว



รูปที่ 4.7 แสดงรูปการส่งอักษร Z จากโปรแกรม Text Recognition (English Version) และ android application

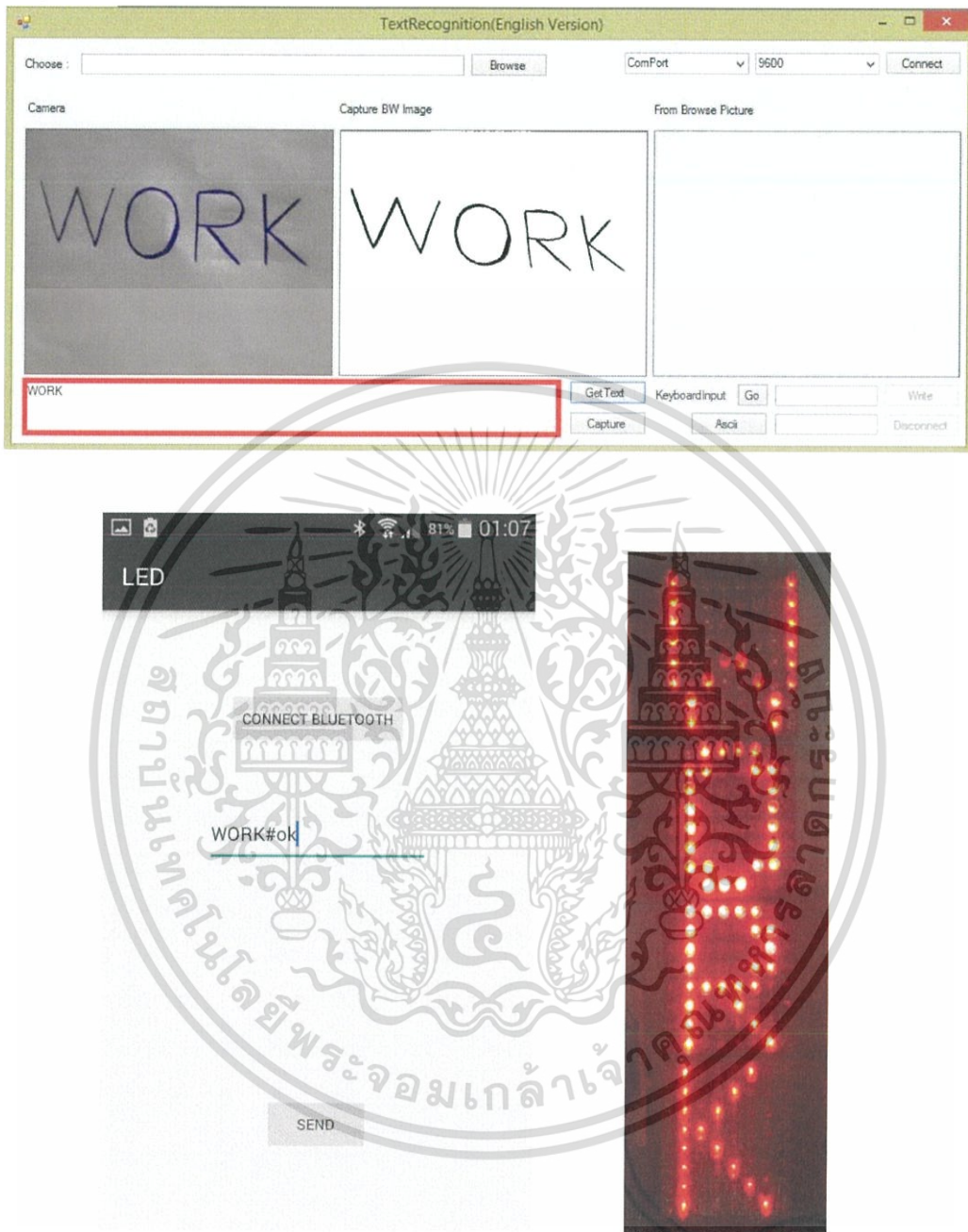
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงรูปการส่งอักษร V จากโปรแกรม Text Recognition (English Version) และ android application

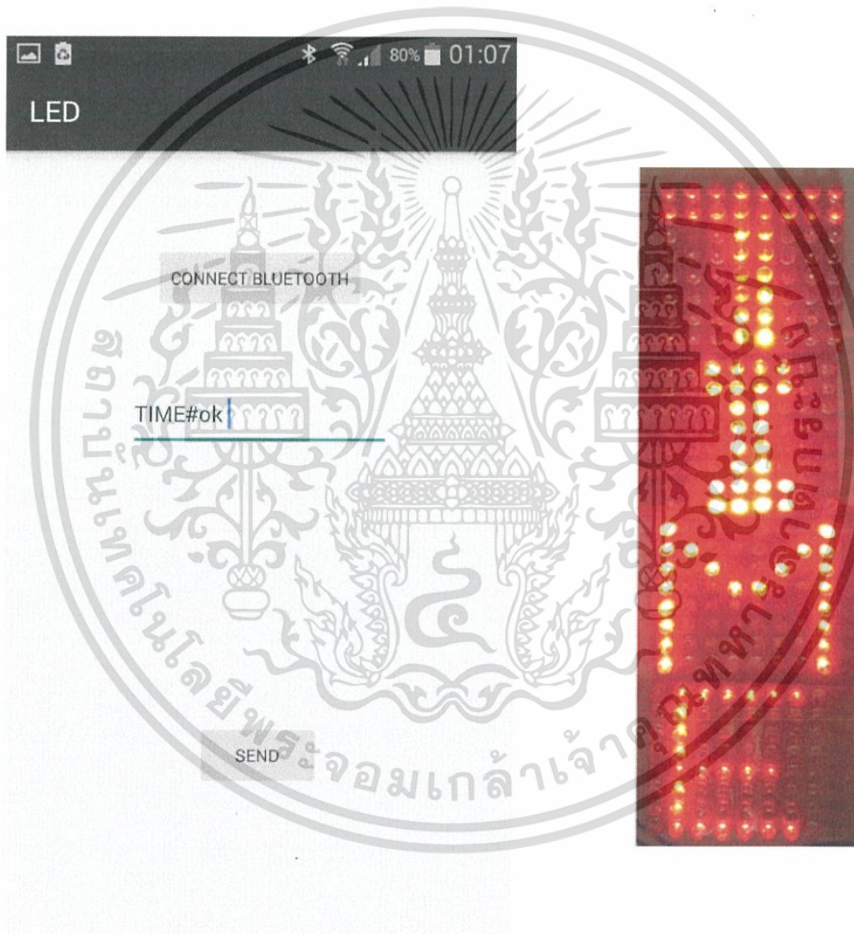
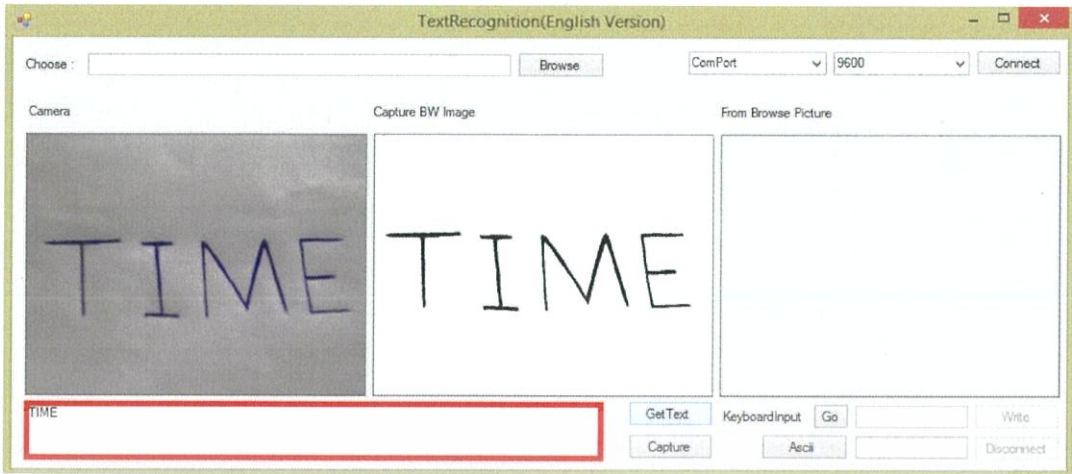
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ส่งคำภาษาอังกฤษที่มี 4 ตัวอักษรพร้อมกัน



รูปที่ 4.9 แสดงรูปการส่งคำว่า WORK จากโปรแกรม Text Recognition (English Version) และ android application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

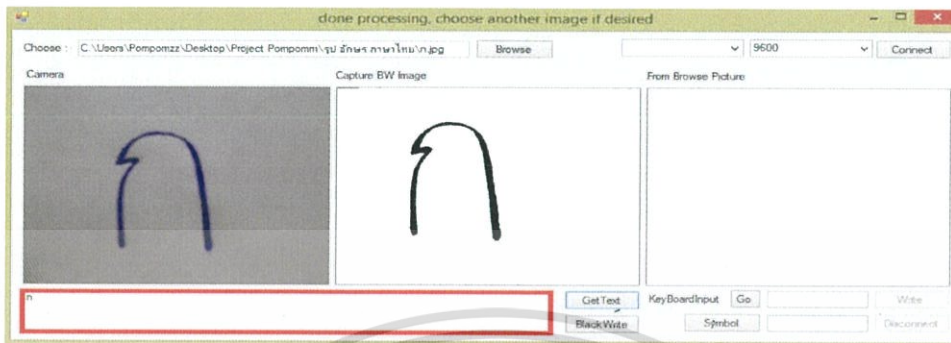


รูปที่ 4.10 แสดงรูปการส่งคำว่า TIME จากโปรแกรม Text Recognition (English Version) และ android application

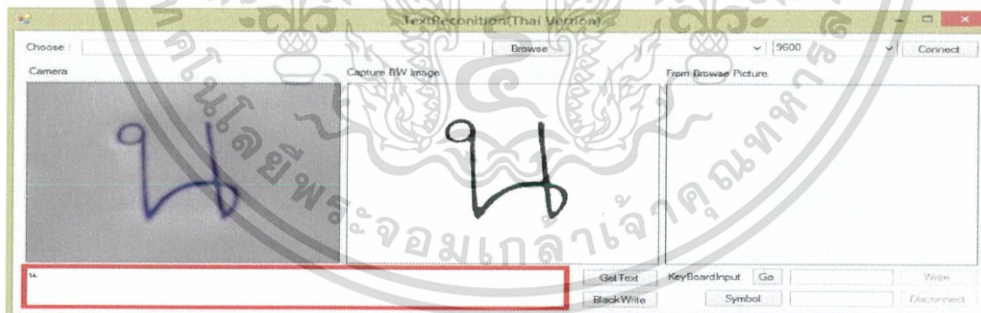
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ Bluetooth ซึ่งสามารถรับค่าจากโปรแกรม Text Recognition (Thai Version)

ทำการส่งตัวอักษรภาษาไทยทีละหนึ่งตัว

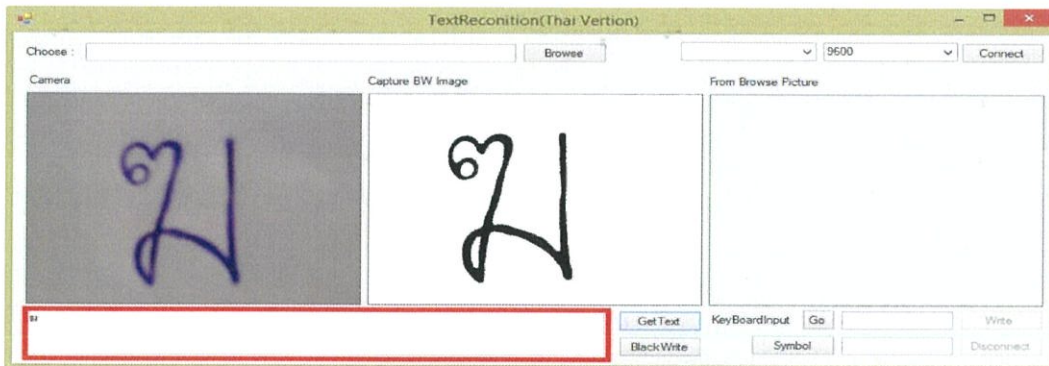


รูปที่ 4.11 แสดงรูปการส่งอักษร ก จากโปรแกรม Text Recognition (Thai Version)



รูปที่ 4.12 แสดงรูปการส่งอักษร น จากโปรแกรม Text Recognition (Thai Version)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



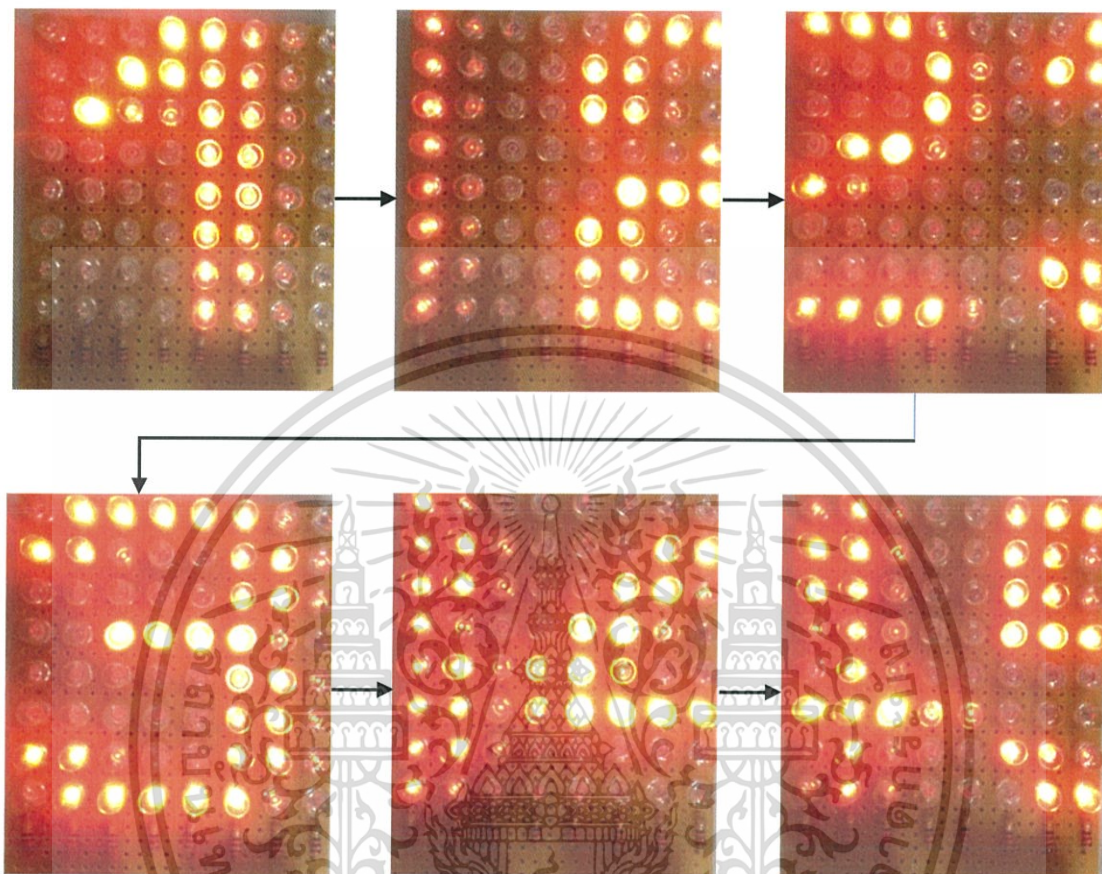
รูปที่ 4.13 แสดงรูปการส่งอักษร ม จากโปรแกรม Text Recognition (Thai Version)



รูปที่ 4.14 แสดงรูปการส่งอักษร ร จากโปรแกรม Text Recognition (Thai Version)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

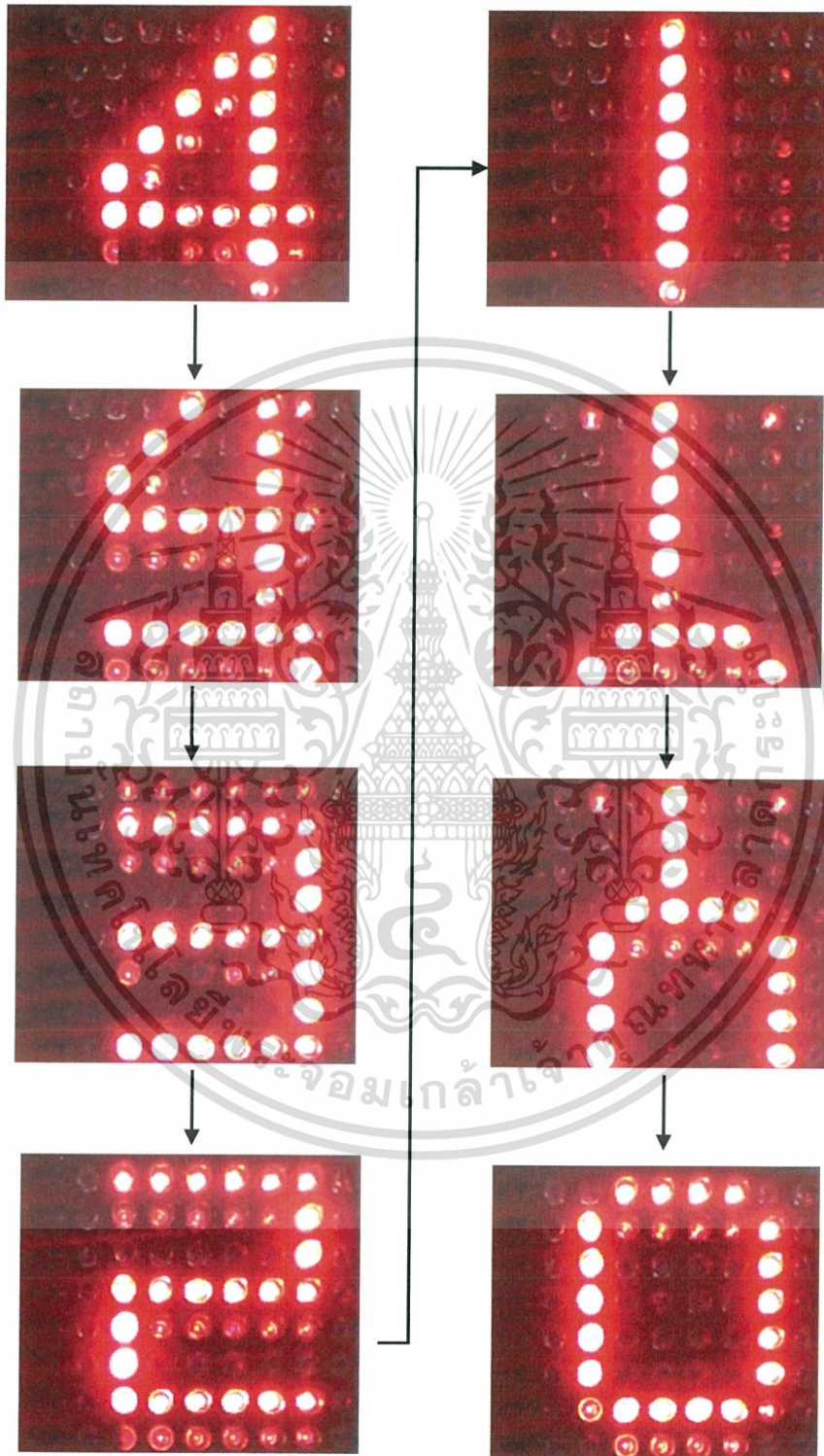
4.4 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ Bluetooth โดยให้ตัวอักษรเลื่อนจากขวาไปซ้าย



รูปที่ 4.15 แสดงรูปการเลื่อนตัวอักษรจากขวาไปซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

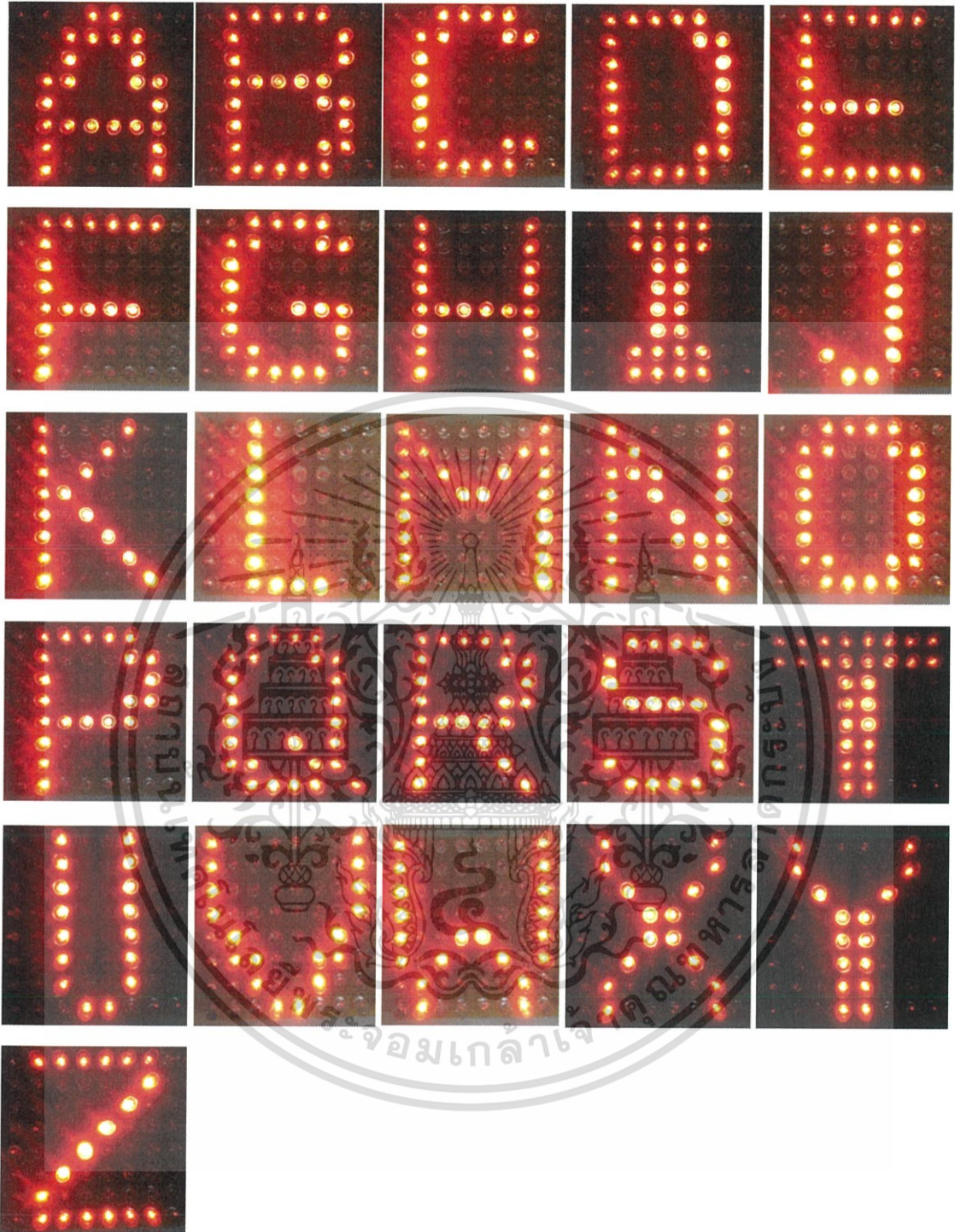
4.5 การทดสอบการแสดงผลโดยส่งค่าผ่าน RS232 หรือ Bluetooth โดยให้ตัวอักษร เลื่อนจากล่างขึ้นบน



รูปที่ 4.16 แสดงรูปการเลื่อนตัวอักษรจากล่างขึ้นบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

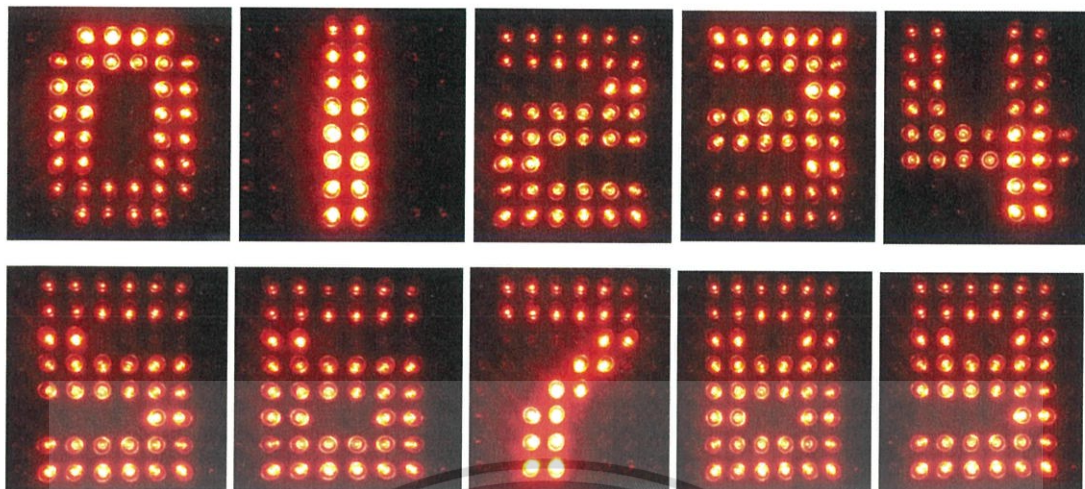
รูปแสดงตัวอักษร A-Z



รูปที่ 4.17 แสดงรูปตัวอักษร A-Z

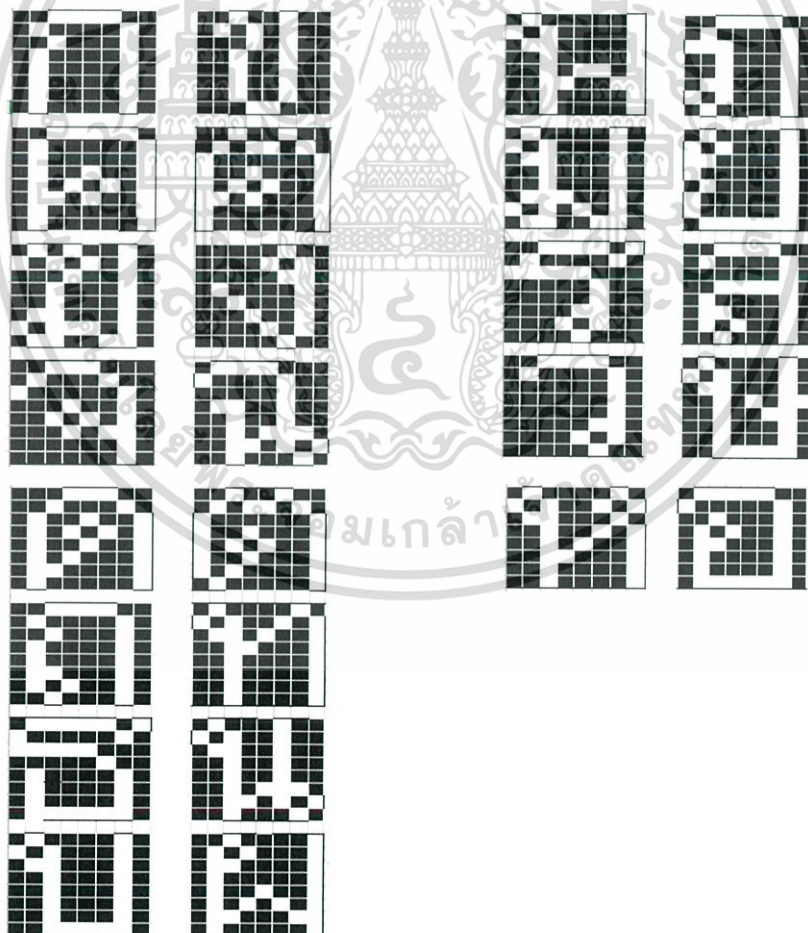
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแสดงตัวเลข 0-9



รูปที่ 4.18 แสดงรูปตัวเลข 0-9

รูปจำลองแสดงตัวอักษรภาษาไทยทั้งหมดที่สามารถทำได้



รูปที่ 4.19 แสดงรูปจำลองตัวอักษรภาษาไทยทั้งหมดที่สามารถทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง และข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ปฏิญานิพนธ์ฉบับนี้เป็นการศึกษาและออกแบบโปรแกรมเพื่อใช้ในการควบคุม LED Display Board โดยสั่งการผ่านไมโครคอนโทรลเลอร์ PIC16F877A จากการศึกษาเบื้องต้นนั้นทำให้รู้ถึงการออกแบบวงจร LED Display Board โดยใช้โปรแกรม Altium การใช้โปรแกรม MiKroC Compiler เพื่อเขียนโปรแกรมควบคุมการแสดงผลบนไมโครคอนโทรลเลอร์ การเขียนโปรแกรม Microsoft Visual Basic 2010 เพื่อตรวจรู้ตัวอักษรและส่งค่าไปยังไมโครคอนโทรลเลอร์ และการเขียนแอปพลิเคชันแอนดรอยด์ เพื่อส่งค่าไปที่ไมโครคอนโทรลเลอร์ โดยใช้พอร์ตอนุกรมทั้งแบบมีสาย ไร้สายในการสื่อสารระหว่างไมโครคอนโทรลเลอร์และอุปกรณ์ต่างๆ ได้แก่ คอมพิวเตอร์ และ โทรศัพท์มือถือ

จากการทดลองในส่วนของารรับค่าโปรแกรม Microsoft Visual Basic 2010 เมื่อได้รับภาพ สามารถตรวจรู้ได้ว่าเป็นตัวอักษรใด และสามารถส่งค่าตัวอักษร ตัวเลข หรือข้อความไปยังไมโครคอนโทรลเลอร์ PIC16F877A เพื่อแสดงผลที่ LED Display Board โดยผ่านพอร์ตอนุกรมได้ทั้งแบบมีสายและไร้สาย และในส่วนของแอปพลิเคชันแอนดรอยด์ สามารถพิมพ์ตัวอักษร หรือตัวเลข แล้วส่งค่าตัวอักษรไปแสดงยัง LED Display Board โดยผ่านพอร์ตอนุกรมแบบไร้สายได้ และสามารถเลื่อนตัวอักษรได้

5.2 ข้อเสนอแนะ

1. ขนาดของ LED Display Board ที่ใช้ในการทดลองมีขนาดเพียง 8x8 บิตต่อหนึ่งบอร์ด สามารถเพิ่มขนาดของบอร์ดให้มีขนาดใหญ่มากขึ้น เพื่อให้สามารถรองรับการแสดงผลที่มีความละเอียดได้มากขึ้น และข้อความที่มีความยาวมากขึ้น
2. เนื่องจากหลอดแอลอีดี ที่ใช้ในการทดลองสามารถแสดงผลได้เพียงสีเดียวคือสีแดง แต่สามารถเปลี่ยนเป็นหลอด 3 สีได้ เพื่อให้สามารถแสดงผลเป็นภาพที่มีหลายสีได้
3. LED Display Board ที่ใช้ในการทดลอง สามารถแสดงผลได้เพียง ตัวเลข ตัวอักษร ภาษาอังกฤษ และตัวอักษรภาษาไทยบางตัวเท่านั้น สามารถพัฒนาให้แสดงผลเป็นรูปภาพต่างๆได้ และภาษาไทยครบทุกตัวอักษร
4. ฮาร์ดแวร์ควรออกแบบให้สามารถใช้งานได้อย่างยืดหยุ่นมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] PIC Microcontroller Learning-By-Doing ด้วยภาษา C. พิมพ์ครั้งที่ 1. กรุงเทพฯ : สมาร์ทเลิร์นนิ่ง, 2550
- [2] ดอนสัน ปงผาบ. ไมโครคอนโทรลเลอร์ PIC และการประยุกต์ใช้งาน -- กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2550
- [3] ประจัน พลังสันติกุล, PIC Works Examples and C Source Code. กรุงเทพฯ : บริษัท แอพซอฟต์แวร์เทคจำกัด, 2537
- [4] ประจัน พลังสันติกุล, All About CCS C PIC C Programming with CCS C Compiler. กรุงเทพฯ : บริษัท แอพซอฟต์แวร์เทคจำกัด, 2551
- [5] บัญชา ปะสีละเตสัง, พัฒนาแอปพลิเคชันด้วย Visual Basic 2010. พิมพ์ครั้งที่ 1. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2554
- [6] Sleeping For Less. 2556. “สารบัญบทความแอนดรอยด์” [ระบบออนไลน์]. แหล่งที่มา <http://www.akexorcist.com/2013/02/android-article-index.html>
- [7] บริษัท เอ็มเบ็ดเต็ด เทคโนโลยี จำกัด. “Altium Designer 14” [ระบบออนไลน์]. แหล่งที่มา <http://www.altiumthai.com/node/209>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



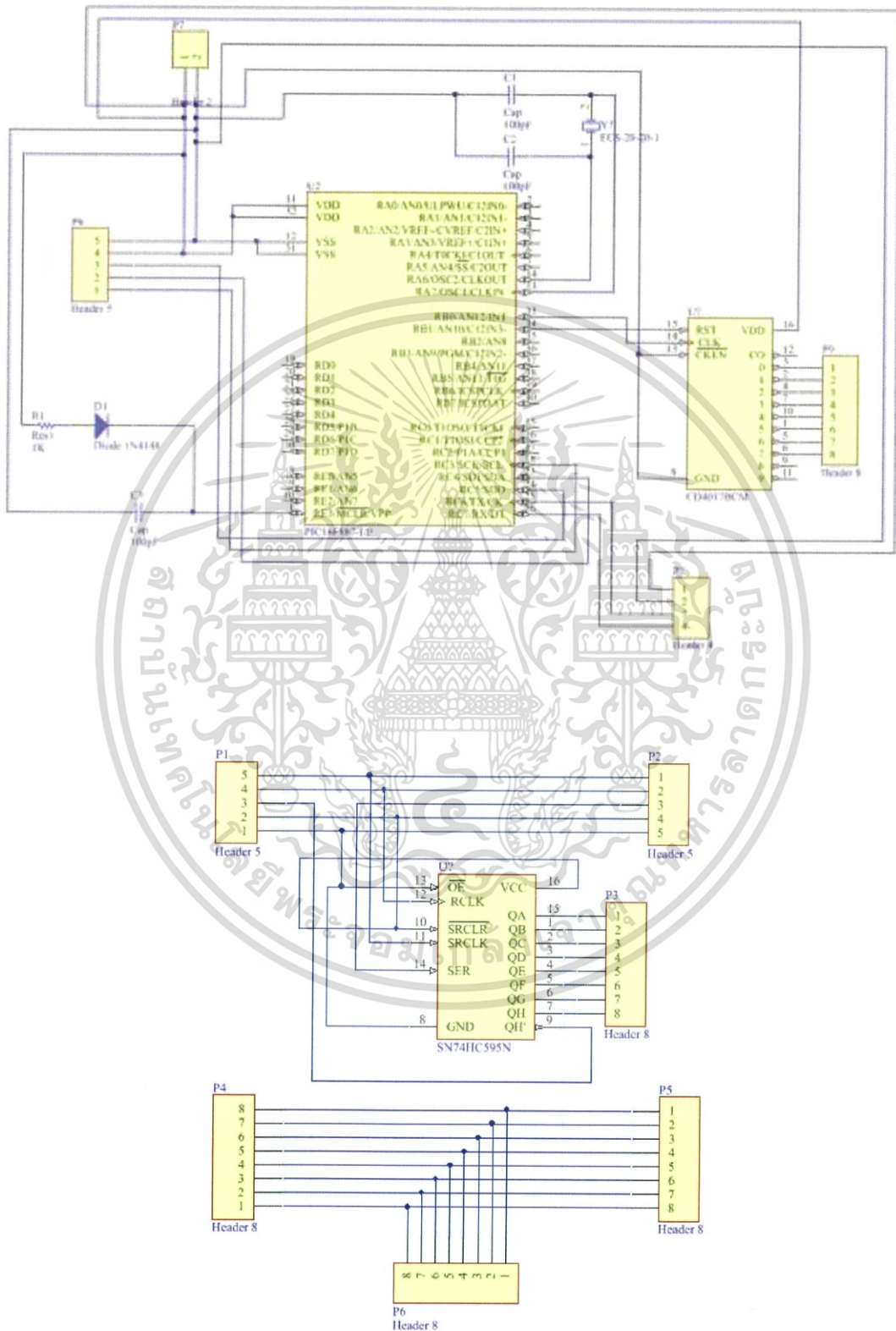
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.
แสดงแบบวงจรจากการใช้โปรแกรม Altium Designer 14.3

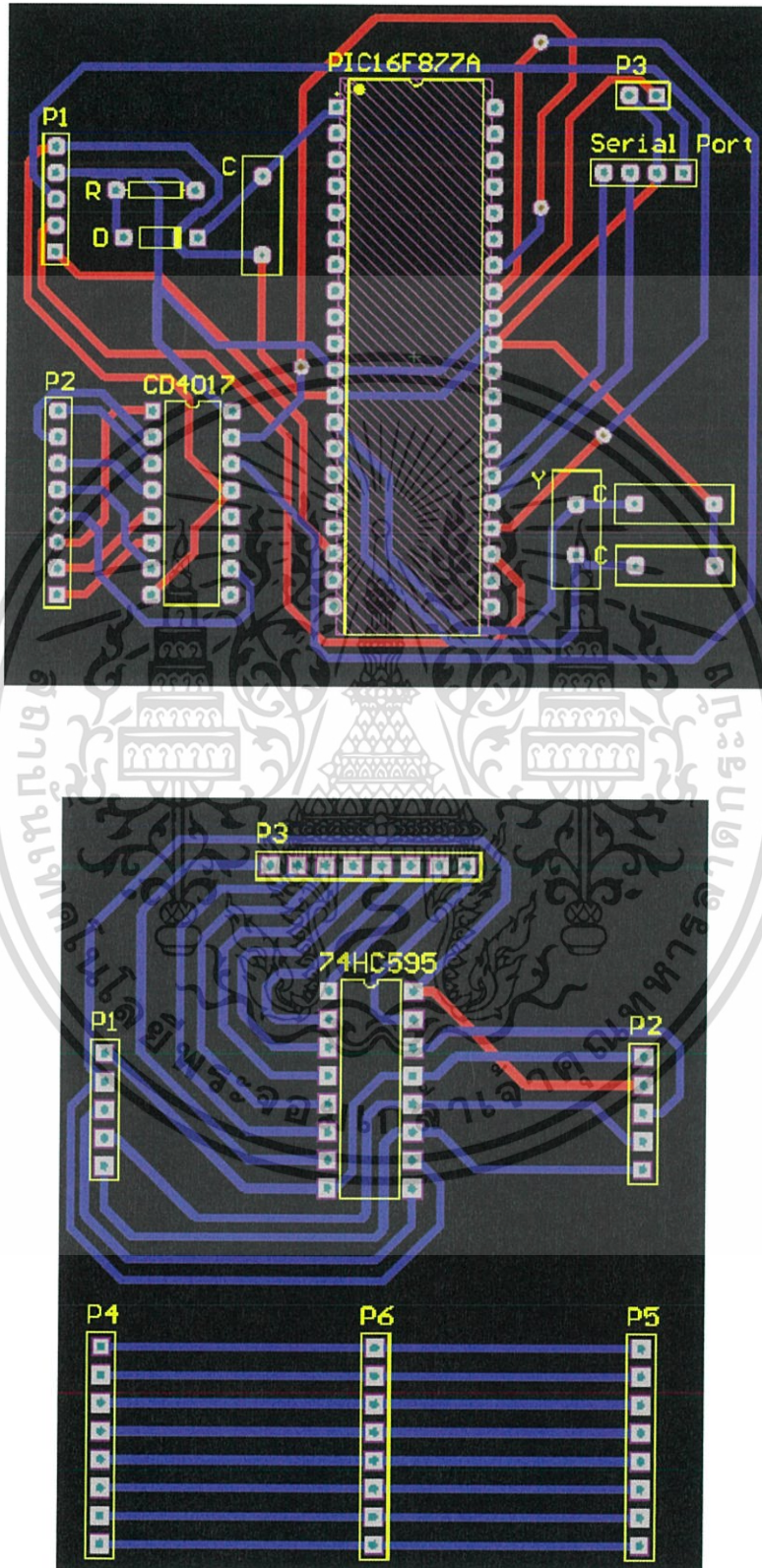
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผงวงจรแบบ Schematic



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผงวงจรแบบ Printed Circuit Board (PCB)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.
โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Visual Basic2010

- โปรแกรม Text Recognition(English Version)

```
Imports Emgu.CV           'อิมพอตไลบรารี EmguCV เพื่อใช้รันโปรแกรม OCR
Imports Emgu.Util
Imports Emgu.CV.OCR
Imports Emgu.CV.Structure
Imports Emgu.CV.UI
Imports Emgu.CV.CvEnum
```

```
Imports System           'อิมพอต System สำหรับใช้ Serial Port
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
```

```
Public Class Form1       'กำหนดตัวแปรต่างๆ
    Dim OCRz As Tesseract = New Tesseract("tessdata", "eng",
    Tesseract.OcrEngineMode.OEM_TESSERACT_ONLY)
    Dim capturez As Capture = New Capture
    Dim blnFirtTimeInResizeEvent As Boolean = True
    Dim intOrigFormWidth As Integer
    Dim intOrigFormHeight As Integer
    Dim intOrigTableLayoutPanelWidth As Integer
    Dim intOrigTableLayoutPanelHeight As Integer
    Dim imgImage As Image(Of Bgr, Byte) = Nothing
    Dim imgBlank As Image(Of Bgr, Byte) = Nothing
    Dim tess As Tesseract
    Dim word As String
```

```
Sub New()
    InitializeComponent()
    intOrigFormWidth = Me.Width
    intOrigFormHeight = Me.Height
    intOrigTableLayoutPanelWidth = ibImage.Width
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

intOrigTableLayoutPanelHeight = ibImage.Height
Try      'ส่วนที่กำหนดว่าจะตรวจรู้ตัวอักษรว่าจะใช้ ภาษาอะไร และใช้กระบวนแบบไหน
    tess = New Tesseract("tessdata", "eng", Tesseract.OcrEngineMode.OEM_DEFAULT)
Catch ex As Exception
    Me.Text = "error instantiating Tesseract object"
    RichTextBox1.Text = "error instantiating Tesseract object"
    txtFile.Enabled = False
    btnFile.Enabled = False
End Try
End Sub

```

```
Private Property myPort As String()
```

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Me.KeyPreview = True
    myPort = IO.Ports.SerialPort.GetPortNames() 'หา Comport ใช้งาน
    BaudComboBox.Items.Add(9600) 'กำหนด Baudrate ใช้งาน
    BaudComboBox.Items.Add(19200)
    BaudComboBox.Items.Add(38400)
    BaudComboBox.Items.Add(57600)
    BaudComboBox.Items.Add(115200)
    BaudComboBox.Text = CStr(BaudComboBox.Items.Item(0))
    PortComboBox.Items.AddRange(CType(myPort, Object()))
    WriteButton.Enabled = False
    CloseButton.Enabled = False
End Sub

```

```

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick      'ทำให้กล้องเว็บแคมแสดงภาพแบบ Real time
    Dim imagez As Image(Of Bgr, Byte) = capturez.QueryFrame()
    PictureBox1.Image = imagez.ToBitmap()
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
```

```
    PictureBox2.Image = PictureBox1.Image
```

```
    Dim bm As New Bitmap(PictureBox2.Image)
```

```
    Dim X As Integer
```

```
    Dim Y As Integer
```

```
    Dim clr As Integer
```

```
    For X = 0 To bm.Width - 1           'Loop ในการทำภาพให้เป็น Grayscale
```

```
        For Y = 0 To bm.Height - 1
```

```
            clr = (CInt(bm.GetPixel(X, Y).R * 0.2989) +
                (bm.GetPixel(X, Y).G * 0.587) +
                (bm.GetPixel(X, Y).B * 0.114))
```

```
            If clr > 100 Then           'เมื่อ pixel นั้นๆแปลงเป็น Grayscale แล้ว
```

```
                clr = 255             'ก็จะมีค่าความสว่างระหว่าง 0 - 255
```

```
            Else                       'จึงนำมาแบ่ง Threshold เพื่อแปลงภาพ
```

```
                clr = 0               'ให้เป็นภาพ Binary
```

```
            End If
```

```
            bm.SetPixel(X, Y, Color.FromArgb(clr, clr, clr))
```

```
        Next Y
```

```
    Next X
```

```
    PictureBox2.Image = bm
```

```
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
    OCRz.Recognize(New Image(Of Bgr, Byte)(PictureBox2.Image)) 'ทำการแปลงภาพ
ตัวอักษรให้เป็น ตัวอักษร
```

```
    RichTextBox1.Text = OCRz.GetText
```

```
End Sub
```

```
Private Sub InitButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles InitButton.Click
```

```
    SerialPort1.PortName = PortComboBox.Text           'ปุ่มที่ทำการเชื่อมต่อ Serial Port
```

```
    SerialPort1.BaudRate = CInt(BaudComboBox.Text)
```

```
    SerialPort1.Open()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

InitButton.Enabled = False
WriteButton.Enabled = True
CloseButton.Enabled = True
End Sub

```

```

Private Sub WriteButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles WriteButton.Click
    Dim ok As String = "#ok"
    SerialPort1.Write(RichTextBox1.Text & ok.ToString) 'ปุ่มที่ทำการส่งค่าไปยัง PIC
End Sub

```

```

Private Sub CloseButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CloseButton.Click
    SerialPort1.Close() 'ปุ่มยกเลิกการเชื่อมต่อ
    InitButton.Enabled = True
    WriteButton.Enabled = False
    CloseButton.Enabled = False
End Sub

```

```

Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    ReceivedText(SerialPort1.ReadExisting()) 'ส่วนการรับค่าจาก PIC
End Sub

```

```

Private Sub ReceivedText(ByVal [text] As String)
    If Me.OutputTextBox.InvokeRequired Then
        Dim x As New ContextCallback(AddressOf ReceivedText)
        Me.Invoke(x, New Object() {{text}})
    Else
        Me.OutputTextBox.Text &= [text]
    End If
End Sub

```

```

Private Sub Form1_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Resize

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If (blnFirtTimeInResizeEvent = True) Then'ส่วนของการกำหนดความกว้างความยาวของ
Image box
    blnFirtTimeInResizeEvent = False
Else
    iblImage.Width = Me.Width - (intOrigFormWidth - intOrigTableLayoutPanelWidth)
    iblImage.Height = Me.Height - (intOrigFormHeigh -
intOrigTableLayoutPanelHeigh)
End If
End Sub

```

```

Private Sub btnFile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFile.Click
    Dim dialogResult As DialogResult = ofdFile.ShowDialog() 'ส่วนของการเลือกไฟล์
ภาพ
    If (dialogResult = Windows.Forms.DialogResult.OK Or dialogResult =
Windows.Forms.DialogResult.Yes) Then
        txtFile.Text = ofdFile.FileName
    End If
End Sub
Private Sub txtFile_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtFile.TextChanged
    txtFile.SelectionStart = txtFile.Text.Length
    If (txtFile.Text <> String.Empty) Then
        ProcessImageAndUpdateGUI()
    End If
End Sub
Sub ProcessImageAndUpdateGUI() 'ส่วนของการตรวจรู้ตัวอักษรจากการเลือกไฟล์ภาพ
Try
    imgImage = New Image(Of Bgr, Byte)(txtFile.Text)
Catch ex As Exception
    Me.Text = "error openig file"
Return
End Try

If (imgImage Is Nothing) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Return
End If
Me.Text = "processing, please wait . . . "
pictureBox1.Image = pictureBoxBlank
Application.DoEvents()
tesseract.Recognize(pictureBox1.Image)
RichTextBox1.Text = tesseract.GetText()
pictureBox1.Image = pictureBoxImage
Me.Text = "done processing, choose another image if desired"

```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
```

```
Dim abc As Integer 'ส่วนของการแปลงตัวอักษรให้เป็นรหัส Ascii
```

```
abc = Asc(RichTextBox1.Text)
```

```
OutputTextBox.Text = abc
```

```
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged
```

```
word = TextBox1.Text 'ส่วนรับข้อมูลจากคีย์บอร์ด
```

```
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
```

```
RichTextBox1.Text = word 'ส่วนการส่งค่าจากส่วนรับคีย์บอร์ดไปยังกล่องข้อความเพื่อ
ส่งไปยัง PIC
```

```
End Sub
```

```
End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรม Text Recognition(Thai Version)

```
Imports Emgu.CV 'อิมพอตไลบรารี EmguCV เพื่อใช้รันโปรแกรม OCR
Imports Emgu.Util
Imports Emgu.CV.OCR
Imports Emgu.CV.Structure
Imports Emgu.CV.UI
Imports Emgu.CV.CvEnum
```

```
Imports System 'อิมพอต System สำหรับใช้ Serial Port
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
```

```
Public Class Form1 'กำหนดตัวแปรต่างๆ
    Dim OCRz As Tesseract = New Tesseract("tessdata", "tha",
Tesseract.OcrEngineMode.OEM_DEFAULT)
    Dim capturez As Capture = New Capture
    Dim blnFirtTimeInResizeEvent As Boolean = True
    Dim intOrigFormWidth As Integer
    Dim intOrigFormHeight As Integer
    Dim intOrigTableLayoutPanelWidth As Integer
    Dim intOrigTableLayoutPanelHeight As Integer
    Dim imgImage As Image(Of Bgr, Byte) = Nothing
    Dim imgBlank As Image(Of Bgr, Byte) = Nothing
    Dim tess As Tesseract
    Dim word As String
```

```
Sub New()
    InitializeComponent()
    intOrigFormWidth = Me.Width
    intOrigFormHeight = Me.Height
    intOrigTableLayoutPanelWidth = ibImage.Width
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
intOrigTableLayoutPanelHeight = iblImage.Height
```

```
Try      'ส่วนที่กำหนดว่าจะตรวจรู้ตัวอักษรว่าจะใช้ ภาษาไทย
        tess = New Tesseract("tessdata", "tha", Tesseract.OcrEngineMode.OEM_DEFAULT)
Catch ex As Exception
        Me.Text = "error instantiating Tesseract object"
        RichTextBox1.Text = "error instantiating Tesseract object"
        txtFile.Enabled = False
        btnFile.Enabled = False
End Try
End Sub
```

```
Private Property myPort As String()
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Me.KeyPreview = True
    myPort = IO.Ports.SerialPort.GetPortNames() 'หา Comport ใช้งาน
    BaudComboBox.Items.Add(9600)              'กำหนด Baudrate ใช้งาน
    BaudComboBox.Items.Add(19200)
    BaudComboBox.Items.Add(38400)
    BaudComboBox.Items.Add(57600)
    BaudComboBox.Items.Add(115200)

    BaudComboBox.Text = CStr(BaudComboBox.Items.Item(0))
    PortComboBox.Items.AddRange(C Type(myPort, Object)))
    WriteButton.Enabled = False
    CloseButton.Enabled = False
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick      'ทำให้กล้องเว็บแคมแสดงภาพแบบ Real time
    Dim imagez As Image(Of Bgr, Byte) = capturez.QueryFrame()
    PictureBox1.Image = imagez.ToBitmap()
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

PictureBox2.Image = PictureBox1.Image

Dim bm As New Bitmap(PictureBox2.Image)

Dim X As Integer

Dim Y As Integer

Dim clr As Integer

For X = 0 To bm.Width - 1 'Loop ในการทำภาพให้เป็น Grayscale

For Y = 0 To bm.Height - 1

clr = (CInt(bm.GetPixel(X, Y).R * 0.2989) +
(bm.GetPixel(X, Y).G * 0.587) +
(bm.GetPixel(X, Y).B * 0.114))

If clr > 100 Then 'เมื่อ pixel นั้นๆแปลงเป็น Grayscale แล้ว

clr = 255 'ก็จะมีค่าความสว่างระหว่าง 0 - 255

Else 'จึงนำมาแบ่ง Threshold เพื่อแปลงภาพ

clr = 0 'ให้เป็นภาพ Binary

End If

bm.SetPixel(X, Y, Color.FromArgb(clr, clr, clr))

Next Y

Next X

PictureBox2.Image = bm

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

OCRz.Recognize(New Image(Of Bgr, Byte)(PictureBox2.Image)) 'ทำการแปลงภาพ
ตัวอักษรให้เป็น ตัวอักษร

RichTextBox1.Text = OCRz.GetText

End Sub

Private Sub InitButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles InitButton.Click

SerialPort1.PortName = PortComboBox.Text 'ปุ่มที่ทำการเชื่อมต่อ Serial Port

SerialPort1.BaudRate = CInt(BaudComboBox.Text)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SerialPort1.Open()
InitButton.Enabled = False
WriteButton.Enabled = True
CloseButton.Enabled = True
End Sub

```

```

Private Sub WriteButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles WriteButton.Click
    Dim ok As String = "#ok"
    SerialPort1.Write(OutputTextBox.Text & ok.ToString) 'ปุ่มที่ทำการส่งค่าไปยัง PIC
End Sub

```

```

Private Sub CloseButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CloseButton.Click
    SerialPort1.Close() 'ปุ่มยกเลิกการเชื่อมต่อ
    InitButton.Enabled = True
    WriteButton.Enabled = False
    CloseButton.Enabled = False
End Sub

```

```

Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    ReceivedText(SerialPort1.ReadExisting()) 'ส่วนการรับค่าจาก PIC
End Sub

```

```

Private Sub ReceivedText(ByVal [text] As String)
    If Me.OutputTextBox.InvokeRequired Then
        Dim x As New ContextCallback(AddressOf ReceivedText)
        Me.Invoke(x, New Object() {(text)})
    Else
        Me.OutputTextBox.Text &= [text]
    End If
End Sub

```

```

Private Sub Form1_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Resize

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If (blnFirtTimeInResizeEvent = True) Then      'ส่วนของการกำหนดความกว้างความยาว
ของ Imagebox
    blnFirtTimeInResizeEvent = False
Else
    lblImage.Width = Me.Width - (intOrigFormWidth - intOrigTableLayoutPanelWidth)
    lblImage.Height = Me.Height - (intOrigFormHeight -
intOrigTableLayoutPanelHeight)
End If
End Sub

```

```

Private Sub btnFile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFile.Click
    Dim dialogResult As DialogResult = ofdFile.ShowDialog() 'ส่วนของการเลือกไฟล์ภาพ
    If (dialogResult = Windows.Forms.DialogResult.OK Or dialogResult =
Windows.Forms.DialogResult.Yes) Then
        txtFile.Text = ofdFile.FileName
    End If
End Sub

```

```

Private Sub txtFile_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtFile.TextChanged
    txtFile.SelectionStart = txtFile.Text.Length
    If (txtFile.Text <> String.Empty) Then
        ProcessImageAndUpdateGUI()
    End If
End Sub

```

```

Sub ProcessImageAndUpdateGUI()      'ส่วนของการตรวจรู้ตัวอักษรจากการเลือกไฟล์ภาพ
Try
    lblImage = New Image(Of Bgr, Byte)(txtFile.Text)
Catch ex As Exception
    Me.Text = "error openig file"
Return
End Try

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If (imgImage Is Nothing) Then
```

```
    Return
```

```
End If
```

```
Me.Text = "processing, please wait . . . "
```

```
ibImage.Image = imgBlank
```

```
Application.DoEvents()
```

```
tess.Recognize(imgImage)
```

```
RichTextBox1.Text = tess.GetText()
```

```
ibImage.Image = imgImage
```

```
Me.Text = "done processing, choose another image if desired"
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button3.Click
```

```
    Dim abc As Integer
```

```
    Dim korkai As String = Nothing
```

```
    abc = Asc(RichTextBox1.Text)
```

```
    If abc = 161 Then
```

```
        korkai = "@"
```

```
    End If
```

```
    If abc = 162 Then
```

```
        korkai = "^"
```

```
    End If
```

```
    If abc = 164 Then
```

```
        korkai = "#"
```

```
    End If
```

```
    If abc = 165 Then
```

```
        korkai = "\"
```

```
    End If
```

```
    If abc = 166 Then
```

```
        korkai = "_"
```

```
    End If
```

```
    If abc = 167 Then
```

```
        korkai = "$"
```

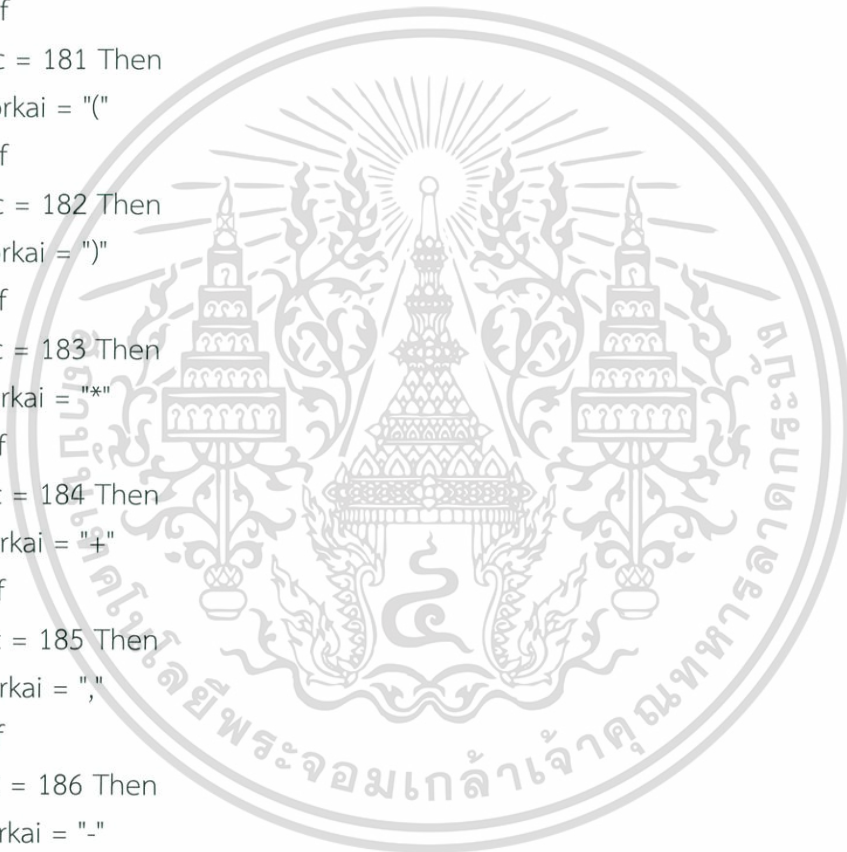
ส่วนของการแปลงตัวอักษรภาษาไทยให้เป็นสัญลักษณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
If abc = 168 Then
    korkai = "%"
End If
If abc = 169 Then
    korkai = "&"
End If
If abc = 180 Then
    korkai = ""
End If
If abc = 181 Then
    korkai = "("
End If
If abc = 182 Then
    korkai = ")"
End If
If abc = 183 Then
    korkai = "*"
End If
If abc = 184 Then
    korkai = "+"
End If
If abc = 185 Then
    korkai = ","
End If
If abc = 186 Then
    korkai = "-"
End If
If abc = 187 Then
    korkai = "."
End If
If abc = 190 Then
    korkai = "/"
End If
If abc = 192 Then

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    korkai = ";"
End If
If abc = 193 Then
    korkai = "["
End If
If abc = 194 Then
    korkai = "{"
End If
If abc = 195 Then
    korkai = "}"
End If
If abc = 196 Then
    korkai = "<"
End If
If abc = 197 Then
    korkai = "="
End If
If abc = 199 Then
    korkai = ">"
End If
If abc = 201 Then
    korkai = ":"
End If
If abc = 203 Then
    korkai = "?"
End If
OutputTextBox.Text = korkai
End Sub

Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged
    word = TextBox1.Text      'ส่วนรับข้อมูลจากคีย์บอร์ด
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RichTextBox1.Text = word 'ส่วนการส่งค่าจากส่วนรับคีย์บอร์ดไปยังกล่องข้อความเพื่อส่งไป
ยัง PIC
End Sub
End Class

```

โปรแกรม Eclipse

- ในส่วนของ MainActivity.java

```

package com.example.led;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import app.akexorcist.bluetoothspp.BluetoothSPP;
import app.akexorcist.bluetoothspp.BluetoothSPP.BluetoothConnectionListener;
import app.akexorcist.bluetoothspp.BluetoothSPP.OnDataReceivedListener;
import app.akexorcist.bluetoothspp.BluetoothState;
import app.akexorcist.bluetoothspp.DeviceList;

public class MainActivity extends ActionBarActivity {
    BluetoothSPP bt;
    EditText edit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

edit = (EditText)findViewById(R.id.editText1);

bt = new BluetoothSPP(this); //ประกาศตัวและกำหนดค่าตัวแปร BluetoothSPP
if(!bt.isBluetoothAvailable()){ //ตรวจสอบว่าอุปกรณ์สามารถใช้บลูทูธได้หรือไม่

    Toast.makeText(getApplicationContext(), "Bluetooth is Not Available",
Toast.LENGTH_SHORT).show(); // แสดงข้อความ Bluetooths is Not Available
    finish();
}

bt.setOnDataReceivedListener(new OnDataReceivedListener() {
@Override
public void onDataReceived(byte[] data, String message) {
Toast.makeText(getApplicationContext(), message, Toast.LENGTH_SHORT).show();
}
}); //บลูทูธรับข้อมูล

bt.setBluetoothConnectionListener(new BluetoothConnectionListener() {
//แสดงสถานะการเชื่อมต่อของบลูทูธ

@Override
public void onDeviceDisconnected() // หยุดทำการเชื่อมต่อ
{Toast.makeText(getApplicationContext(), "Connection Lost",
Toast.LENGTH_SHORT).show();} // แสดงข้อความ Connection Lost

@Override
public void onDeviceConnectionFailed() // ไม่สามารถเชื่อมต่อได้
{Toast.makeText(getApplicationContext(), "Unable to Connect",
Toast.LENGTH_SHORT).show();} // แสดงข้อความ Unable to Connect

@Override
public void onDeviceConnected(String name, String address)
// เชื่อมต่อกับอุปกรณ์นั้นๆได้
{Toast.makeText(getApplicationContext(), "Connect To" + name + "\n" + address,
Toast.LENGTH_SHORT).show();}
// แสดงข้อความ Connect ชื่อบลูทูธ ที่สามารถเชื่อมต่อได้ และ แสดง address

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

});

    Button b = (Button)findViewById(R.id.button_connect);
    b.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
    if
(bt.getServiceState()==BluetoothState.STATE_CONNECTED){bt.disconnect(); }
        else {Intent intent = new Intent(getApplicationContext(),
DeviceList.class);
            startActivityForResult(intent,
BluetoothState.REQUEST_CONNECT_DEVICE);
        }
    }
});
}
public void onDestroy(){
    super.onDestroy();
    bt.stopService();
}

public void onStart(){
    super.onStart(); //ตรวจสอบว่าอุปกรณ์เปิดใช้งานบลูทูธอยู่หรือไม่
    if(!bt.isBluetoothEnabled()){ //ทำการร้องขอให้เปิดใช้บลูทูธ

Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(intent, BluetoothState.REQUEST_ENABLE_BT);
} //ทำการเลือกอุปกรณ์ที่จะทำการเชื่อมต่อด้วย
else {
        if(!bt.isServiceAvailable()){
            bt.setupService();
            bt.startService(BluetoothState.DEVICE_OTHER);
            setup();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

private void setup() {
    Button s = (Button)findViewById(R.id.button_send);
    s.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
String message = edit.getText().toString(); //แปลงค่าจาก editText ให้เป็น string
bt.send(message, true); //ส่งข้อมูลที่รับจากedittext
}
});
}

public void onActivityResult(int requestCode, int resultCode, Intent
data){
if(requestCode==BluetoothState.REQUEST_CONNECT_DEVICE){
if(resultCode==Activity.RESULT_OK) bt.connect(data);}
else if (requestCode == BluetoothState.REQUEST_ENABLE_BT){
if(resultCode == Activity.RESULT_OK){
bt.setupService();
bt.startService(BluetoothState.DEVICE_ANDROID);
setup();
}
else {
Toast.makeText(getApplicationContext(), "Bluetooth was
not enabled", Toast.LENGTH_SHORT).show();
finish();
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

}

```

- ในส่วนของ activity_main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.ibelt.MainActivity" >

```

```

<Button
    android:id="@+id/button_connect"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"
    android:text="@string/button_title" />

```

```

<Button
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="92dp"
    android:text="@string/button_send" />

```

```

<EditText

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/button_connect"
android:layout_centerHorizontal="true"
android:layout_marginTop="57dp"
android:ems="10"
android:hint="@string/edit_hint"
android:inputType="textPersonName" >

<requestFocus />
</EditText>

</RelativeLayout>

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม MikroC Compiler

- โปรแกรมแสดงตัวอักษรบน Dot Matrix LED ขนาด 8x8 โดยรับอินพุตจาก RS232

```
#define Latch PORTD.F3 // กำหนดให้พอร์ต D3 เป็นขาแลตซ์
int k=0,i=0,j=0,l=0,g=0,o=0,rat,dat,speed; //สร้างและกำหนดตัวแปรชนิด integer
char HC595_0,HC595_1,HC595_2,HC595_3; // สร้างตัวแปรชนิด character
char Msg[15]; // สร้างอาร์เรย์ 1มิติ ขนาด15 ชนิด character
unsigned short ASCII[80]; //สร้างอาร์เรย์ขนาด 1มิติขนาด 80 โดยไม่มีเครื่องหมาย
const unsigned short RAM[44][8] = {
//สร้างอาร์เรย์ 2 มิติ ขนาด 44x8 เพื่อเก็บค่าคงที่ของรูปแบบตัวอักษร
    0x00, 0x7E, 0xC1, 0xB1, 0x8D, 0x83, 0x7E, 0x00, // Code pour char 0
    0x00, 0x04, 0x02, 0x01, 0xFF, 0x00, 0x00, 0x00, // Code pour char 1
    0x00, 0xE6, 0x91, 0x91, 0x89, 0x89, 0x86, 0x00, // Code pour char 2
    0x00, 0x42, 0x81, 0x89, 0x89, 0x89, 0x76, 0x00, // Code pour char 3
    0x00, 0x30, 0x28, 0x24, 0x22, 0xFF, 0x20, 0x00, // Code pour char 4
    0x00, 0x4F, 0x89, 0x89, 0x89, 0x89, 0x71, 0x00, // Code pour char 5
    0x00, 0x7E, 0x89, 0x89, 0x89, 0x89, 0x72, 0x00, // Code pour char 6
    0x00, 0x01, 0x01, 0xF1, 0x09, 0x05, 0x03, 0x00, // Code pour char 7
    0x00, 0x76, 0x89, 0x89, 0x89, 0x89, 0x76, 0x00, // Code pour char 8
    0x00, 0x4E, 0x91, 0x91, 0x91, 0x91, 0x7E, 0x00, // Code pour char 9
    0x00, 0x00, 0xc3, 0xff, 0xff, 0xc3, 0x00, 0x00, // |
    0x00, 0xff,0x06, 0x0c, 0x30, 0x60, 0xff, 0x00, // N
    0x00, 0x46,0x89, 0x89, 0x89, 0x89, 0x72, 0x00, // S
    0x3f,0xc0,0x20, 0x10, 0x10, 0x20, 0xc0, 0x3f, // W
    0x00,0x7e,0xff, 0xc3, 0xc3, 0xff, 0x7e, 0x00, // 0
    0x00,0xff,0x11, 0x11, 0x31, 0x5b, 0x8e, 0x00, // R
    0x00,0xff,0x08, 0x14, 0x22, 0x41, 0x80, 0x00, // K
    0x00,0xf8,0x2e, 0x23, 0x23, 0x2e, 0xf8, 0x00, // A
    0x00,0xff,0x89, 0x89, 0x89, 0xd9, 0x76, 0x00, // B
    0x00,0x7e,0xc3, 0x81, 0x81, 0xc3, 0x42, 0x00, // C
    0x00,0xff,0x81, 0x81, 0x81, 0xc3, 0x7e, 0x00, // D
    0x00,0xff,0x91, 0x91, 0x91, 0x91, 0x81, 0x00, // E
    0x00,0xff,0x11, 0x11, 0x11, 0x11, 0x01, 0x00, // F
    0x00,0x7e,0xc3, 0x81, 0x91, 0x93, 0x72, 0x00, // G
    0x00,0xff,0x10, 0x10, 0x10, 0x10, 0xff, 0x00, // H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0x00,0x00,0x40, 0x80, 0x81, 0x7f, 0x01, 0x00, // J
0x00,0x00,0xff, 0x80, 0x80, 0x80, 0x00, 0x00, // L
0xff,0x02,0x04, 0x08, 0x08, 0x04, 0x02, 0xff, // M
0x00,0xff,0x11, 0x11, 0x11, 0x1d, 0x0e, 0x00, // P
0x00,0x7e,0xc3, 0x81, 0xa1, 0xc3, 0x7e, 0x80, // Q
0x03,0x03,0x03, 0xff, 0xff, 0x03, 0x03, 0x03, // T
0x00,0x00,0x7f, 0x80, 0x80, 0x7f, 0x00, 0x00, // U
0x0f,0x30,0x60, 0x80, 0x80, 0x60, 0x30, 0x0f, // V
0x00,0xc3,0x24, 0x18, 0x18, 0x24, 0xc3, 0x00, // X
0x00,0x03,0x04, 0xf8, 0xf8, 0x04, 0x03, 0x00, // Y
0x00,0xc3,0xa1, 0x91, 0x89, 0x85, 0x83, 0x00, // Z
0x00,0xfa,0x05, 0x01, 0x01, 0x01, 0xfe, 0x00, // ก
0x00,0x02,0x79, 0x86, 0x80, 0x7f, 0x00, 0x00, // ข
0x42,0xa1,0xfa, 0x45, 0x42, 0x80, 0xff, 0x00, // ค
0x04,0x02,0x11, 0x29, 0x11, 0x22, 0xfc, 0x00, // จ
0x02,0x09,0xf9, 0x41, 0x21, 0x7e, 0xa0, 0x40, // ฉ
0x02,0x05,0xfe, 0x40, 0x20, 0x7f, 0xa0, 0x40, // ญ
0x04,0x06,0x05, 0x45, 0xa5, 0x79, 0x02, 0x01 // ร
};

void interrupt() { //สร้างฟังก์ชันอินเทอร์รัปต์
  rat++; //ให้ตัวแปร rat เพิ่มค่าทีละค่า
  speed++; //ให้ตัวแปร speed เพิ่มค่าทีละค่า
  TMR0 = 240 // กำหนดให้รีจิสเตอร์ไทม์เมอร์ TMR0 = 240
  INTCON = 0x20; // กำหนดให้รีจิสเตอร์ INTCON = 0x20
}

*****

void send_data(char HC595_0, char HC595_1, char HC595_2,char HC595_3)
{
  Spi1_Write(HC595_0); // เขียนข้อมูลไปยัง HC595 ตัวที่ 1 ผ่านการสื่อสารแบบ SPI
  Spi1_Write(HC595_1); // เขียนข้อมูลไปยัง HC595 ตัวที่ 2 ผ่านการสื่อสารแบบ SPI
  Spi1_Write(HC595_2); // เขียนข้อมูลไปยัง HC595 ตัวที่ 3 ผ่านการสื่อสารแบบ SPI
  Spi1_Write(HC595_3); // เขียนข้อมูลไปยัง HC595 ตัวที่ 4 ผ่านการสื่อสารแบบ SPI
  Latch=1 // สร้างสัญญาณแลตช์
  Latch=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void test() {
if(rat>=10){          // สร้างเงื่อนไขเมื่อตัวแปร rat >= 10
if (speed>=9000){    // สร้างเงื่อนไขเมื่อตัวแปร speed >= 9000
    dat++;           // เพิ่มค่าตัวแปร dat ละหนึ่ง
    speed=0; }      // กำหนดให้ตัวแปร speed = 0
    rat =0;         // กำหนดให้ตัวแปร rat = 0
    RB0_bit=1;     // สร้างสัญญาณนาฬิกาไปยัง IC CD4017BE
    RB0_bit=0;
if(i>=8){           // สร้างเงื่อนไขเมื่อ i >= 8
    i=0;           // กำหนดให้ i=0
    RB1_bit=1;     // สร้างสัญญาณเพื่อไป Reset IC CD4017BE
    RB1_bit=0;
    } i++;         // กำหนดให้ตัวแปร i เพิ่มค่าทีละหนึ่ง
    HC595_0=~ASCII[i+dat]; // กำหนดให้ตัวแปร HC595_0 = อาเรย์ ASCII ตำแหน่ง i+dat
    HC595_1=~ASCII[(i+8)+dat]; // กำหนดให้ตัวแปร HC595_1 = อาเรย์ ASCII ตำแหน่ง i+8+dat
    HC595_2=~ASCII[(i+16)+dat]; // กำหนดให้ตัวแปร HC595_2 = อาเรย์ ASCII ตำแหน่ง i+16+dat
    HC595_3=~ASCII[(i+24)+dat]; // กำหนดให้ตัวแปร HC595_3 = อาเรย์ ASCII ตำแหน่ง i+24+dat
    send_data(HC595_0,HC595_1,HC595_2,HC595_3); // ส่งข้อมูลไปยังฟังก์ชัน send_data
    Delay_ms(1); } // หน่วงเวลา 1มิลลิวินาที
if(dat>=48)         // สร้างเงื่อนไขเมื่อตัวแปร dat >= 48
{dat=0; }          // กำหนดให้ตัวแปร dat = 0
}

```

สร้างฟังก์ชัน FillArray , FillArray_2 , FillArray_Thai เพื่อนำข้อมูลภายในอาเรย์ RAM ไปใส่ในอาเรย์ ASCII เมื่อมีการส่งข้อมูลตัวอักษรและตัวเลขผ่าน RS232 มายัง Microcontroller โดยจะเก็บตัวอักษรไว้ในอาเรย์ Msg

```

void FillArray() {
do{
    if(Msg[i] == '0' { // สร้างเงื่อนไขเมื่อมีข้อมูลภายในอาเรย์ Msg เท่ากับ '0'
        j=(8*i)+30; k=0; // กำหนดให้ j = 8 x i + 30 , กำหนดให้ k=0
    }
do{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ASCII[j] = RAM[0][k];    // กำหนดให้อาเรย์ ASCII ตำแหน่งที่ j เท่ากับ อาเรย์ RAM ตำแหน่ง
                                // ที่ 0 , k
        j++; k++; }            // กำหนดให้ j และ k เพิ่มค่าทีละหนึ่ง
while(k<=7);                // กำหนดเงื่อนไขให้ลูป do-while ยังคงเป็นจริงต่อเมื่อ k<=7
    }
if(Msg[i] == '1') {k=0; j=(8*i)+30;
    do{ ASCII[j] = RAM[1][k];
        j++; k++; }while(k<=7);
}
if(Msg[i] == '2') {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[2][k];
        j++;k++;;} while(k<=7);
}

if(Msg[i] == '3') {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[3][k];
        j++; k++;; }while(k<=7);}
if(Msg[i] == '4') {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[4][k];
        j++; k++;; }while(k<=7); }
if(Msg[i] == '5') {k=0; j=(8*i)+30;
    do {ASCII[j] = RAM[5][k];
        j++;k++;;}while(k<=7); }
if(Msg[i] == '6') {k=0; j=(8*i)+30;
    do{ ASCII[j] = RAM[6][k];
        j++;k++;;}while(k<=7); }
if(Msg[i] == '7') {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[7][k];
        j++;k++;;}while(k<=7); }
if(Msg[i] == '8') {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[8][k];
        j++;k++;;}while(k<=7); }
if(Msg[i] == '9') {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[9][k];
        j++;k++;;}while(k<=7); }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(Msg[i] == 'I' || Msg[i] == 'i') {k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[10][k];
    j++;k++;}while(k<=7); }
if(Msg[i] == 'N' || Msg[i] == 'n')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[11][k];
    j++;k++;}while(k<=7);}
if(Msg[i] == 'S' || Msg[i] == 's')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[12][k];
    j++;k++;}while(k<=7);}
if(Msg[i] == 'W' || Msg[i] == 'w')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[13][k];
    j++;k++;}while(k<=7);}
if(Msg[i] == 'O' || Msg[i] == 'o')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[14][k];
    j++;k++;}while(k<=7);}
if(Msg[i] == 'R' || Msg[i] == 'r')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[15][k];
    j++;k++;}while(k<=7); }
if(Msg[i] == 'K' || Msg[i] == 'k')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[16][k];
    j++;k++;}while(k<=7); }
if(Msg[i] == 'A' || Msg[i] == 'a')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[17][k];
    j++;k++;}while(k<=7);
}
  i++;
}while(i<=15);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void FillArray_2()
{i=0;
  do{if(Msg[i] == 'B' || Msg[i] == 'b')
    {k=0; j=(8*i)+30;
      do{ASCII[j] = RAM[18][k]; j++;k++;
    }while(k<=7);
    }
  if(Msg[i] == 'C' || Msg[i] == 'c')
  {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[19][k];
      j++;k++;;}while(k<=7);
  }
  if(Msg[i] == 'D' || Msg[i] == 'd')
  {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[20][k];
      j++;k++;;}while(k<=7);
  }
  if(Msg[i] == 'E' || Msg[i] == 'e')
  {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[21][k];
      j++;k++;;}while(k<=7);
  }
  if(Msg[i] == 'F' || Msg[i] == 'f')
  {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[22][k];
      j++;k++;;}while(k<=7);
  }
  if(Msg[i] == 'G' || Msg[i] == 'g')
  {k=0; j=(8*i)+30;
    do{ASCII[j] = RAM[23][k];
      j++;k++;;}while(k<=7);
  }
  if(Msg[i] == 'H' || Msg[i] == 'h')
  {k=0; j=(8*i)+30;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{ASCII[j] = RAM[24][k];
  j++;k++;}while(k<=7);
}
if(Msg[i] == 'J' || Msg[i] == 'j')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[25][k];
    j++;k++;}while(k<=7);
}
if(Msg[i] == 'L' || Msg[i] == 'l')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[26][k];
    j++;k++;}while(k<=7);
}
if(Msg[i] == 'M' || Msg[i] == 'm')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[27][k];
    j++;k++;}while(k<=7);
}
if(Msg[i] == 'P' || Msg[i] == 'p')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[28][k];
    j++;k++;}while(k<=7);
}
if(Msg[i] == 'Q' || Msg[i] == 'q')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[29][k];
    j++;k++;}while(k<=7);
}
if(Msg[i] == 'T' || Msg[i] == 't')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[30][k];
    j++;k++;}while(k<=7);
}
if(Msg[i] == 'U' || Msg[i] == 'u')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[31][k];
    j++;k++; }while(k<=7);
}
if(Msg[i] == 'V' || Msg[i] == 'v')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[32][k];
    j++;k++;}while(k<=7);
}
if(Msg[i] == 'X' || Msg[i] == 'x')
{k=0; j=(8*i)+30;
  do {ASCII[j] = RAM[33][k];
    j++;k++;;}while(k<=7);
}
if(Msg[i] == 'Y' || Msg[i] == 'y')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[34][k];
    j++;k++;;}while(k<=7);
}
if(Msg[i] == 'Z' || Msg[i] == 'z')
{k=0; j=(8*i)+30;
  do{ASCII[j] = RAM[35][k];
    j++;k++;;}while(k<=7);
}
  i++;
}while(i<=15);
}
FillArray_Thai() {
  i=0;
  do{
  if(Msg[i] == '!')
  {
    k=0;
    j=(8*i)+30;
    do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        ASCII[j] = RAM[36][k];
        j++;k++;}while(k<=7);
    }
    if(Msg[i] == '^')
    {k=0; j=(8*i)+30;
        do{ASCII[j] = RAM[37][k];
            j++;k++;}while(k<=7);
    }
    if(Msg[i] == '_')
    {k=0; j=(8*i)+30;
        do{ASCII[j] = RAM[38][k];
            j++;k++;}while(k<=7);
    }
    if(Msg[i] == '%')
    {k=0; j=(8*i)+30;
        do{ASCII[j] = RAM[39][k];
            j++;k++;}while(k<=7);
    }
    if(Msg[i] == '&')
    {k=0;j=(8*i)+30;
        do{ASCII[j] = RAM[40][k];
            j++;k++;}while(k<=7);
    }
    if(Msg[i] == ',')
    {k=0;j=(8*i)+30;
        do{ASCII[j] = RAM[41][k];
            j++;k++;}while(k<=7);
    }
    if(Msg[i] == '}')
    {k=0; j=(8*i)+30;
        do{ASCII[j] = RAM[41][k];
            j++;k++;}while(k<=7);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i++;
    }while(i<=15);
}
void main() {
INTCON=0xA0;    // เปิดการใช้งานอินเทอร์พรีตไทมเมอร์โดยกำหนดให้รีจิสเตอร์ INTCON = 0xA0
OPTION_REG = 0x00;    // กำหนดให้ค่าพรีสเกลและ TMR0 = 0
TMR0=0;
ADCON0=0;    // ปิดการใช้งานโมดูลแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล
TRISB=0;    // กำหนดให้พอร์ต B เป็นเอาต์พุต
TRISD=0;    // กำหนดให้พอร์ต D เป็นเอาต์พุต
PORTA=0;    // กำหนดให้พอร์ต A = 0
PORTB=0;    // กำหนดให้พอร์ต B = 0
PORTC=0;    // กำหนดให้พอร์ต C = 0
PORTD=0;    // กำหนดให้พอร์ต D = 0
rat=0;    // กำหนดให้ตัวแปร rat = 0
i=0;    // กำหนดให้ตัวแปร i = 0
dat=0;    // กำหนดให้ตัวแปร dat =0

Spi1_Init_Advanced(_SPI_MASTER_OSC_DIV4,_SPI_DATA_SAMPLE_MIDDLE,_SPI_CLK_IDLE_
HIGH,_SPI_LOW_2_HIGH);    // ตั้งค่าการสื่อสารแบบ SPI

*****
for(o=0;o<=80;o++)
{ASCII[o] = 0x00;    // กำหนดให้อาเรย์ ASCII = 0 ตั้งแต่ตำแหน่งที่ 0 - 80
}
uart1_Init(9600);    // กำหนดบอตรัตของ UART = 9600 bps
uart1_read_text(Msg,"#ok" ,255); //อ่านข้อมูลสตริงจนกว่าจะได้รับอักขระ 'ok' จึงจะนำมา
    เก็บไว้ในอาเรย์ Msg
uart1_write_text(Msg);    // เขียนข้อมูลสตริงภายในอาเรย์ Msg ออกทาง UART
FillArray();    // เรียกใช้งานฟังก์ชัน FillArray
FillArray_2();    // เรียกใช้งานฟังก์ชัน FillArray_2
FillArray_Thai();    // เรียกใช้งานฟังก์ชัน FillArray_Thai
while(1)    // วงลูปไม่รู้จบ
    {test();    // เรียกใช้งานฟังก์ชัน test
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้