วิธีการหาขอบเขตตัวอักษรและจับคำภาษาอังกฤษบนหน้าจอคอมพิวเตอร์

A Method of English Character Capturing on Monitor

นายพิชิต ธรรมวงศ์ รหัส 46380033

นายพีระพล นกเผือก รหัส 46380191

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงการ	วิธีการกำหนดขอบเขตและจับคำภาษาอังกฤษบนหน้าจอคอมพิวเตอร์
ผู้ดำเนินโครงงาน	นายพิชิต ธรรมวงศ์ รหัส 46380033
	นายพีระพล นกเผือก รหัส 46380191
อาจารย์ที่ปรึกษา	ดร. อัครพันธ์ วงศ์กังแห
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

.....

คณะวิศวกรรมศาสตร์ มหาลัยนเรศวร อนุมัติให้โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร. อัครพันธ์ วงศ์กังแห)

.....กรรมการ

(ดร. สุรเดช จิตประไพกุลศาล)

.....กรรมการ

(อาจารย์ศิริพร เดชะศิลารักษ์)

หัวข้อโครงการ	วิธีการหาขอบเขตตัวอักษรและจับคำภาษาอังกฤษบนหน้าจอคอมพิวเตอร์		
ผู้ดำเนินโครงงาน	นายพิชิต	ธรรมวงศ์	รหัส 46380033
	นายพีระพล	นกเผือก	รหัส 46380191
อาจารย์ที่ปรึกษา	ดร.อัครพันธ์	วงศ์กังแห	
	ดร.สุรเดช	จิตประไพกุลศาล	
	อาจารย์ศิริพร	เดชะศิลารักษ์	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

บทคัดย่อ

โครงงานนี้เป็นการศึกษาเกี่ยวกับโปรแกรมแปลคำศัพท์โดยการจับภาพหน้าจอคอมพิวเตอร์ ซึ่งประกอบไปด้วย ส่วนที่เป็นการป้อนคำศัพท์เหมือนโปรแกรมแปลโดยทั่วไป และส่วนที่เป็นตัวจับคำที่หน้าจอภาพ และส่วนที่เป็นตัวแสดงคำศัพท์ที่ได้จากการค้นหา เพื่อทำให้ผู้ใช้สามารถเข้าใจความหมายของคำศัพท์เหล่านั้น โดยผู้ใช้จะลดเวลาในการเปิดพจนานุกรม ในการจัดทำโครงงานนี้ผู้จัดทำได้ใช้โปรแกรม Visual Basic และ Microsoft Access ในการจัดทำโปรแกรมขึ้นมา

ผลที่ได้จากการทำโครงงานนี้ โปรแกรมสามารถช่วยเหลือผู้ใช้ในการแปลความหมายของคำศัพท์ได้เป็นอย่างดี อีกทั้งผู้ใช้ไม่จำเป็นที่ต้องป้อนข้อมูลทางแป้นพิมพ์ เพื่อลดเวลาการใช้งาน และง่ายต่อการใช้งานอีกด้วย

Project Title	Character Capturing on monitor for English-Thai Language Translation.		
Name	Mr Pichit	Thammawong	ID. 46380033
	Mr Peerapol	Nokphuak	ID. 46380191
Project Advisor	Akaraphunt	Vongkunghae, PhD.	
	Suradet	Jitprapaikulsarn, PhD.	
	Siriporn	Dachasilaruk	
Major	Computer Engineering.		
Department	Electrical and Computer Engineering.		
Academic Year	2007		

ABSTRACT

This project is the study about a program the vocabulary translates by screen capture for computer, which includes inputting vocabulary by keyboard as same as general program, the part of screen capture and result display for give user understands their vocabulary. The user will decrease time in the open dictionary. This project uses Visual Basic and Microsoft Access Program for created

This project result is a program enabling the user to translate English into Thai easier than the other translation tools. Users don't have to type the input data. It is a convenient dictionary program interesting on for us.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงาน คือ ดร.อัครพันธ์ วงศ์กังแห ที่คอยช่วยเหลือ เอาใจใส่ และช่วยแนะนำในทุกๆด้าน รวมถึงอาจารย์ที่ปรึกษาร่วม และอาจารย์ท่านอื่นที่ คอยให้คำปรึกษาและแนะนำต่างๆ และเพื่อนทุกๆ คนที่ให้กำลังใจเสมอมา ในโอกาสนี้ทางคณะผู้จัดทำจึงขอขอบคุณทุกๆท่านที่มีส่วนช่วยเหลือโครงงานนี้ให้ประสบความสำเร็จได้ด้วยดี

รวมทั้งบิดามารดาของข้าพเจ้าที่สนับสนุนในทุกๆด้าน และให้โอกาสทางการศึกษาอย่างเต็มที่ ข้าพเจ้า ขอระลึกถึงในบุญคุณของท่านและขอกราบขอบพระคุณมา ณ ที่นี้

นายพิชิต ธรรมวงศ์

นายพีระพล นกเผือก

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงาน

คอมพิวเตอร์ เป็นอุปกรณ์ที่มีความจำเป็นอย่างยิ่งในปัจจุบัน โดยเฉพาะในยุคโลกาภิวัฒน์ ที่สังคมเปิดกว้าง ทั้งในเรื่องข้อมูลข่าวสาร การติดต่อสื่อสาร การคมนาคม การแพทย์ ฯลฯ สิ่งเหล่านี้ล้วนจำเป็นที่ต้องมีคอมพิวเตอร์เข้ามาช่วยเป็นเครื่องมือ และในอนาคตก็ยิ่งจำเป็นมากขึ้น โดยเฉพาะในยุคไอทีอย่างนี้

ในประเทศไทยก็ได้นำคอมพิวเตอร์เข้ามาใช้ในด้านต่างๆ มากมาย เช่น ด้านการศึกษา ด้านการเงิน การทหาร การสื่อสาร การคมนาคม การบันเทิง ฯลฯ และอื่นๆอีกมากมาย จึงมีการผลิตซอฟต์แวร์ออกมาให้ใช้มากมายและเพื่อให้ใช้ซอฟต์แวร์นั้นๆมีประสิทธิภาพยิ่งขึ้นก็ต้องศึกษาคู่มือที่ติดมากับซอฟต์แวร์นั้น ทำให้เกิดปัญหาในเรื่องของภาษา ส่วนใหญ่แล้วภาษาที่ใช้มักจะเป็นภาษาอังกฤษทำให้เกิดปัญหาที่ผู้ใช้อ่านคู่มือไม่ได้ จึงทำให้ใช้แซฟต์แวร์นั้นๆได้ไม่เต็มประสิทธิภาพเท่าที่ควร พจนานุกรมอังกฤษ - ไทย จึงเป็นหนทางหนึ่งที่คอยช่วยเหลือให้กับผู้ที่ต้องการจะแปลคู่มือของซอฟต์แวร์นั้นๆ แต่พจนานุกรมอังกฤษ - ไทย นั้นต้องคอยป้อนคำศัพท์ผ่านทางแป้นพิมพ์ ซึ่งจะช้าต่อการแปล หรือ บางครั้งข้อมูลที่เรานำมาเป็นไฟล์รูปภาพไม่สามารถที่จะแปลได้ จึงมีความคิดว่า ทำอย่างไรจึงจะทำให้ปัญหานี้หมดไปได้ โดยไม่ต้องป้อนคำศัพท์ทางแป้นพิมพ์ จะทำอย่างไรที่จะแปลงไฟล์ที่เป็นรูปภาพให้มาเป็นไฟล์ตัวหนังสือ ในยุคของสารสนเทศ (Information age) นี้ ข้อมูลข่าวสารเป็นสิ่งจำเป็นที่ทุกคนสามารถเสาะแสวงหาหรือในทางกลับกัน ก็สามารถเผยแพร่ให้บุคคลทั่วไปได้รับทราบถึงข้อมูล ซึ่งจะเป็นประโยชน์ต่อวงการต่าง ๆ อย่างกว้างขวาง การส่งและการรับข้อมูลข่าวสารที่เป็นภาษาไทยนับว่ามีความสำคัญต่อทุกวงการ ไม่ว่าจะเป็นด้านวิชาการหรือด้านธุรกิจ ทั้งนี้เพื่อจะได้ติดต่อสื่อสารกันด้วยความสะดวก โดยไม่จำกัดเพศ หรือการศึกษา โดยใช้ภาษาไทยเป็นสื่อกลางในการติดต่อ เมื่อคอมพิวเตอร์เข้ามามีบทบาทในการศึกษาและการประกอบอาชีพมากขึ้น ความต้องการที่จะทำให้คอมพิวเตอร์แสดงผลภาษาไทยไปจนถึงสามารถเข้าใจภาษาไทย และโต้ตอบด้วยภาษาไทยได้ก็เพิ่มมากขึ้น และการประยุกต์ใช้ภาษาไทยบนคอมพิวเตอร์ นับตั้งแต่เรื่องการประมวลผลอักษร (Character processing) การประมวลผลคำ (Word processing) การประมวลผลข้อความ (Text processing) ไปจนถึงการประมวลผลภาพ (Image processing) ของภาษาไทย หนทางต่างๆเป็นการทำเพื่อจะทำการแปลภาษาอังกฤษให้เป็นภาษาไทย

ปัจจุบันจึงมีการนำเอาเทคโนโลยี โอซีอาร์ (Optical Character Recognition: OCR) เข้ามาใช้ โอซีอาร์ คือ โปรแกรมรู้จำอักษรด้วยแสง หรือโอซีอาร์ (Optical Character Recognition)

โอซีอาร์เป็นคำย่อของภาษาอังกฤษว่า "Optical Character Recognition: OCR" แปลเป็นภาษาไทยได้ว่า "การรู้จักอักษรด้วยแสง" เป็นงานประยุกต์งานหนึ่งของสาขาวิทยาการคอมพิวเตอร์ ที่ได้รับความสนใจและพัฒนามานานกว่า 70 ปีแล้ว โอซีอาร์เป็นการรู้จำรูปแบบตัวอักษร ซึ่งเป็นงานวิจัยในสาขาการรู้จำรูปแบบ (Pattern Recognition) เป็นเทคโนโลยีที่ส่งผลให้ระบบคอมพิวเตอร์สามารถระบุรูปแบบได้อย่างถูกต้อง เช่น สามารถจะบอกได้ว่า ภาพนั้นคือภาพอะไร ภาพตัวอักษรนั้นคือตัวอักษรอะไร หรือเสียงนั้นคือเสียงของคำสั่งอะไร เป็นต้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อเรียนรู้และศึกษาทางด้านการเขียนโปรแกรม เช่น Visual Basic, MS Access, Image Processing

1.2.3 เพื่อเป็นการศึกษาถึงระบบฐานข้อมูลและการสืบค้นข้อมูล (คำศัพท์)

1.2.4 เพื่อศึกษาถึงหลักการของ OCR (Optical Character Recognition)

1.2.5 เพื่อเป็นการนำซอฟต์แวร์มาประยุกต์ใช้กับฐานข้อมูล(คำศัพท์) เพื่อเป็นพจนานุกรมในการแปลอังกฤษ – ไทย

1.2.6 เพื่อทำโปรแกรมสำหรับแปลทันทีจากหน้าจอแล้วนำมาประยุกต์ใช้งาน

1.2.7 เพื่อช่วยเหลือผู้ที่มีความรู้ทางด้านภาษาอังกฤษน้อย ก็สามารถเข้าใจคำศัพท์นั้นได้

1.3 ขอบข่ายของโครงงาน

1.3.1 ศึกษาและออกแบบ OCR (Optical Character Recognition)

1.3.2 ทำการออกแบบฐานข้อมูล (คำศัพท์)

1.3.3 ศึกษาการทำงาน และออกแบบจากโปรแกรม OCR (Optical Character Recognition) ฐานข้อมูล (คำศัพท์)

1.3.4 สามารถแปลตัวอักษรได้เฉพาะตัวอักษรมาตรฐาน ไม่สามารถแปลตัวอักษรที่เขียนด้วยลายมือหรือการสแกนด้วยการเขียนลายมือ

1.3.5 โครงงานนี้สามารถแปลงตัวอักษรได้ไม่เกินขนาด 26 พิกเซล และเป็นฟอนต์มาตรฐาน

1.3.6 โครงงานนี้ไม่สามารถแปลงตัวอักษรที่เป็นสีขาว และตัวอักษรที่ขีดเส้นใต้ได้

1.3.7 โครงงานนี้สามารถจับตัวอักษรในเอกสารพวก Microsoft Word และ เอกสาร PDF ได้ดี

1.4 แผนการดำเนินงาน

ตารางที่ 1.1 แสดงความคืบหน้าในการทำโครงงาน

กิจกรรม	ปี 2550										ปี 2551		
	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. รวบรวมและศึกษาข้อมูลของ OCR	←	→											
2. ศึกษาถึงการแปลงไฟล์จากไฟล์รูปภาพเป็นไฟล์ตัวหนังสือ	←	→											
3. ศึกษาถึงทฤษฎีการเขียนโปรแกรม Visual Basic	←	→											
4. ศึกษาถึงการเขียนโปรแกรมฐานข้อมูล Access	←	→											
5.ทำการออกแบบโปรแกรม		←	→										
6.สร้างส่วนติดต่อระหว่างผู้ใช้กับโปรแกรม							←	→					
7. เขียนโปรแกรม							←	→	→	→	→	→	
8. ทดสอบโปรแกรม								←	→	→	→	→	→
9.แก้ไขข้อผิดพลาดตรวจสอบความต้องการ								←	→	→	→	→	
10. สรุปผล								←	→	→	→	→	→
11. จัดเตรียมเอกสาร							←	→	→	→	→	→	→

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 ได้รับความรู้จากการเขียนโปรแกรม

1.5.2 ได้เข้าใจถึงการสืบค้นข้อมูลจากฐานข้อมูล

1.5.3 ได้รับประสบการณ์ของการทำงานร่วมกันในกลุ่ม

1.5.4 ได้รู้ถึงกระบวนการการทำงานของคอมพิวเตอร์

1.5.5 ได้พัฒนาโปรแกรมที่สามารถช่วยเหลือผู้ที่ไม่มีความรู้ทางภาษาอังกฤษมากนัก

1.5.6 ได้โปรแกรมแปลอังกฤษ – ไทย จากหน้าจอคอมพิวเตอร์ หรือ เอกสาร

1.6 งบประมาณ

1.6.1 ค่าหนังสือประกอบการทำโครงงาน	700 บาท
1.6.2 ค่าหมึกพิมท์	300 บาท
1.6.3 ค่ากระดาษสำหรับพิมพ์	200 บาท
1.6.4 ค่าซอฟต์แวร์ที่ใช้ในการพัฒนา	400 บาท
1.6.5 ค่าเข้าเล่มโครงงาน	400 บาท
รวมทั้งสิ้น	2000 บาท
	(สองพันบาทถ้วน)

บทที่ 2
หลักการและทฤษฎี

2.1 หลักการจับภาพหน้าจอคอมพิวเตอร์ (Capture on Computer)

หลักการของโครงงานนี้คือ เราจะมองหน้าจอคอมพิวเตอร์ของเรา เป็นภาพหนึ่งเท่านั้น อาจจะกล่าวว่า เป็นภาพ Wallpaper ก็ได้ เมื่อได้ภาพมาแล้วก็จะใช้หลักการเกี่ยวเมาส์ เข้ามาช่วยเพื่อระบุตำแหน่งหรือเก็บค่าของตำแหน่งนั้นไว้ และยึดถือตำแหน่งนั้นเป็นจุดกึ่งกลางภาพ หลังจากนั้นจะทำการวิเคราะห์เพื่อหาตำแหน่งอ้างอิงของจุด x และเก็บค่านั้นไว้ เพื่อที่จะนำไปสู่ขั้นตอนการหาจุด y กระบวนการต่อไป จะทำการหาขอบบน (Top Margin) และขอบล่าง (Botton Margin) เพื่อตัดแถวตามยาวของคำๆนั้น ในส่วนนี้ สิ่งสำคัญคือ เงื่อนไขในการหาขอบบนและขอบล่าง เพราะจะมีการตรวจสอบถึงพิกเซลที่เป็นสีขาวที่มีความยาวตลอดแนวแกน y หลังจากที่ตัดขอบบนและขอบล่างเรียบร้อยแล้ว ก็จะทำการสแกนตามแนวแกน x หรือแนวตั้ง เพื่อหาขอบเขตของคำๆนั้น ทั้งซ้าย-ขวา หรือเรียกว่าการหาขอบซ้าย (Left Margin) และขอบขวา (Right Margin) เมื่อได้ค่าของขอบซ้าย-ขวาแล้ว เราจะทำการสแกนใหม่อีกหนึ่งรอบ โดยครั้งนี้เป็นการสแกนเพื่อตัดตัวอักษรออกมาทีละตัว แล้วเก็บอักษรเหล่านี้ไว้ในอะเรย์ เพื่อที่จะลำเลียงเข้าไปประมวลผลในโอซีอาร์ และแปลงออกเป็นคำหรือประโยคที่เป็นตัวอักษร ในหัวข้อต่อไปจะเป็นรายละเอียดของการจับคำภาษาอังกฤษ

2.1.1 กระบวนการจับภาพทางจอภาพ

ในส่วนนี้เป็นการแสดงให้เห็นถึงขั้นตอนการจับภาพของโปรแกรมนี้ว่าใช้อัลกอริทึมใดบ้างในการจับภาพ (คำที่ต้องการ) มาแปลงเป็นตัวอักษร หลักการการจับภาพสรุปได้ดังต่อไปนี้

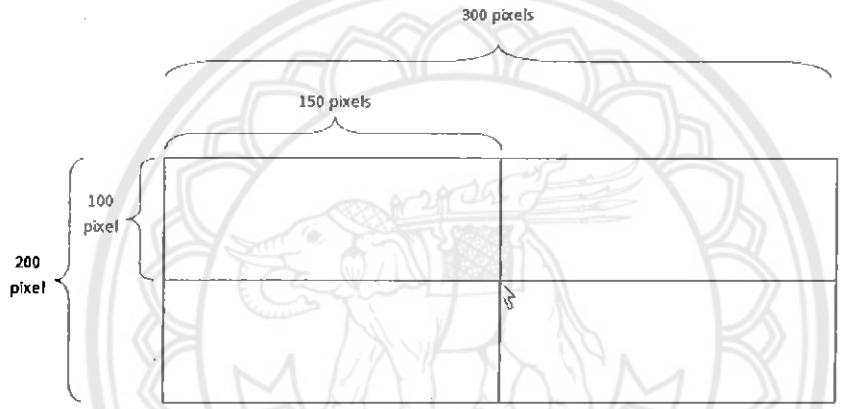
API (Application Programming Interface) เป็นฟังก์ชันที่ใช้เกี่ยวกับกราฟิก ในโครงงานนี้มีการเรียกใช้ VBAPI ใน Visual Basic 6.0 เพื่อเป็นตัวจับภาพ เปรียบเสมือนเป็นกระบวนการทางวินโดวส์ ที่จะใช้จัดการหน้าจอคอมพิวเตอร์ เป็นอัลกอริทึมแรกที่มีการเรียกใช้หลังจากผู้ใช้ กดปุ่ม Capture ตัวอย่างฟังก์ชันที่สำคัญ เช่น Function GetScreen(Optional ByVal hwnd As Long) As IPictureDisp เป็นฟังก์ชันที่ใช้จับภาพหน้าจอคอมพิวเตอร์ Public Declare Function GetCursorPos Lib "User32" (lpPoint As POINTAPI) เป็นฟังก์ชันเกี่ยวกับตำแหน่งที่เมาส์ชี้ Public Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long เป็นฟังก์ชันที่ใช้เกี่ยวกับการรับค่า pixel ในภาพ เพื่อนำมาทำการแปลงอักษร Public Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long เป็นฟังก์ชันเกี่ยวกับการจัด context device เหล่านี้เป็นตัวอย่างของฟังก์ชัน เกี่ยวกับ API ทั้งสิ้น

2.1.2 การนำภาพที่จับได้มาแปลงเป็นตัวอักษร

หลังจากกระบวนการจับภาพ จากข้อที่ 2.1.1 จะได้ภาพมา 1 ภาพที่มีขนาด 300 x 200 pixel ในส่วนนี้เราเรียกว่า กระบวนการแปลงภาพ จะมีการแปลงภาพจากภาพสี (RGB) ให้เป็นภาพขาว-ดำ (gray scale) แล้วจะเข้าสู่ขั้นตอนต่อไปนี้

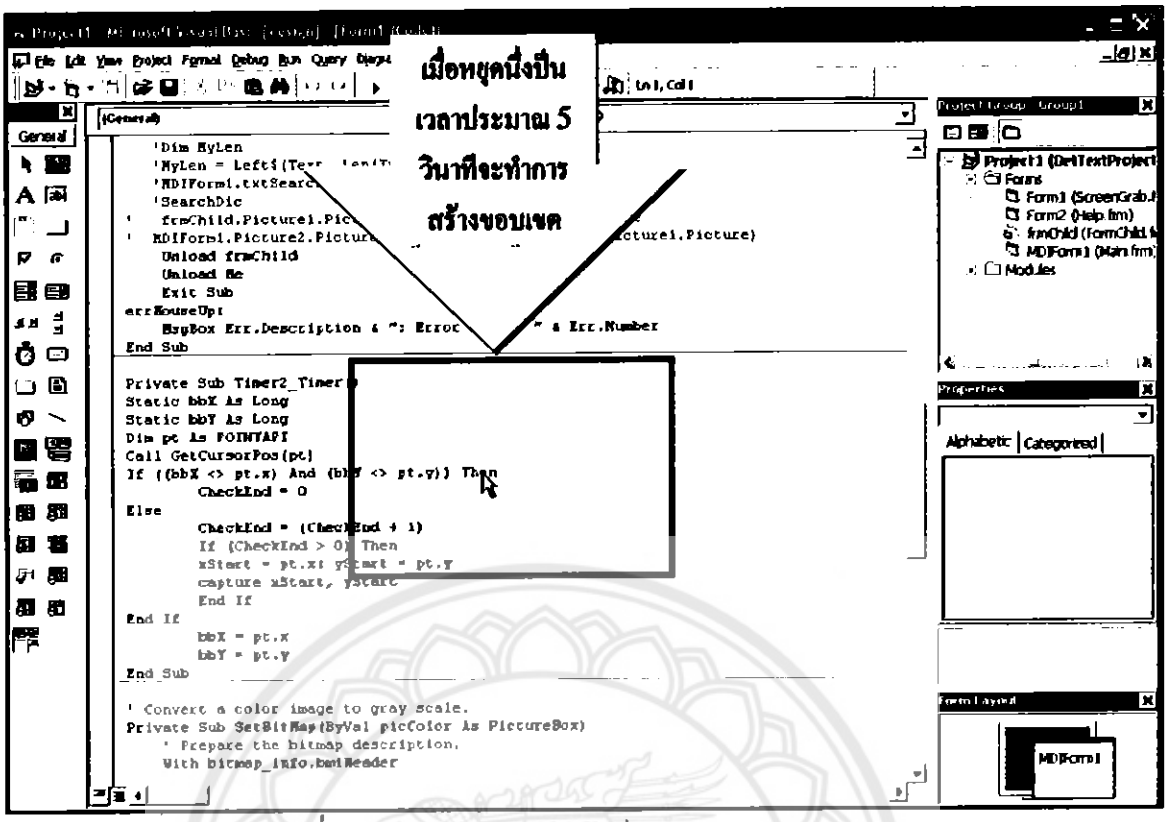
2.1.2.1 การกำหนดขอบเขตภาพ

เมื่อเข้าสู่ฟังก์ชัน Capture () ฟังก์ชันนี้จะนำเอาคู่อันดับ x, y ที่เป็นการกำหนดขอบเขตที่จะตัดภาพ ซึ่งในโครงงานตัดภาพเป็นขนาด width 300 และ height 200 pixel โดยจะกำหนดเคอร์เซอร์ให้อยู่ตรงกลางจากฟังก์ชัน Public Declare Function GetCursorPos Lib "User32" (IpPoint As POINTAPI) As Long ผลจะได้ดังรูปข้างล่างนี้



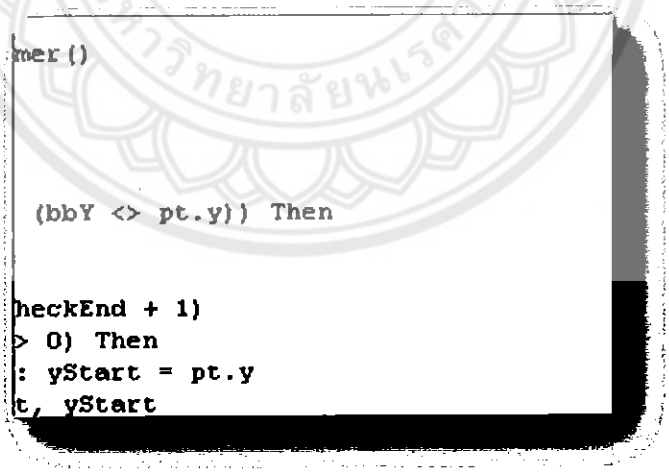
รูปที่ 2.1 แสดงการกำหนดขนาดภาพ

แต่ก่อนที่จะมากำหนดขอบเขตภาพนั้น จะมีอีกอัลกอริทึม ที่จะใช้เป็นตัวจับเวลาว่าตอนนี้เมาส์ของเราหยุดเคลื่อนไหวหรือยัง ถ้าเมาส์หยุดการเคลื่อนไหว เป็นเวลา 5 วินาที จะใช้คำสั่งจับภาพและกำหนดขนาดของภาพ



รูปที่ 2.2 ผลการทำงานของฟังก์ชัน Timer และ Capture

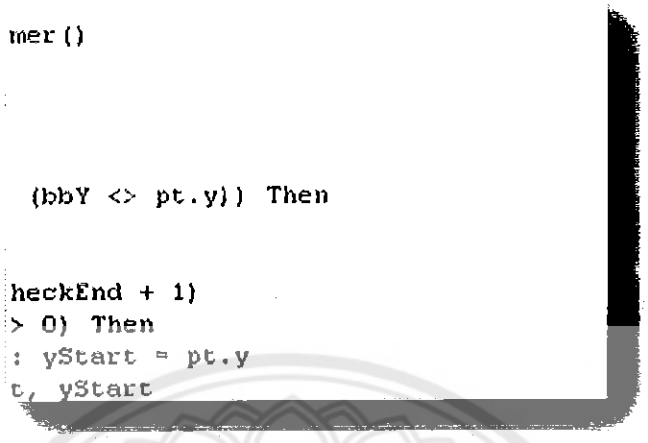
และผลลัพธ์สุดท้ายที่จะได้จะมีลักษณะดังนี้



รูปที่ 2.3 ภาพสีขนาด 300 x 200

รูปที่ 2.3 **เป็นภาพที่เราจับได้ แต่จะเห็นว่าเป็นภาพสี และกระบวนการต่อไปคือ การทำให้ รูปที่** 2.3 **กลายเป็นภาพขาว-ดำ หรือ gray scale และคำสั่งที่ใช้คือ** Private Sub **SetBitMap(ByVal picColor As** PictureBox) **และหลังจากนั้นจะนำค่าที่ได้จากคำสั่งSetBitMap() มาใช้กับ**

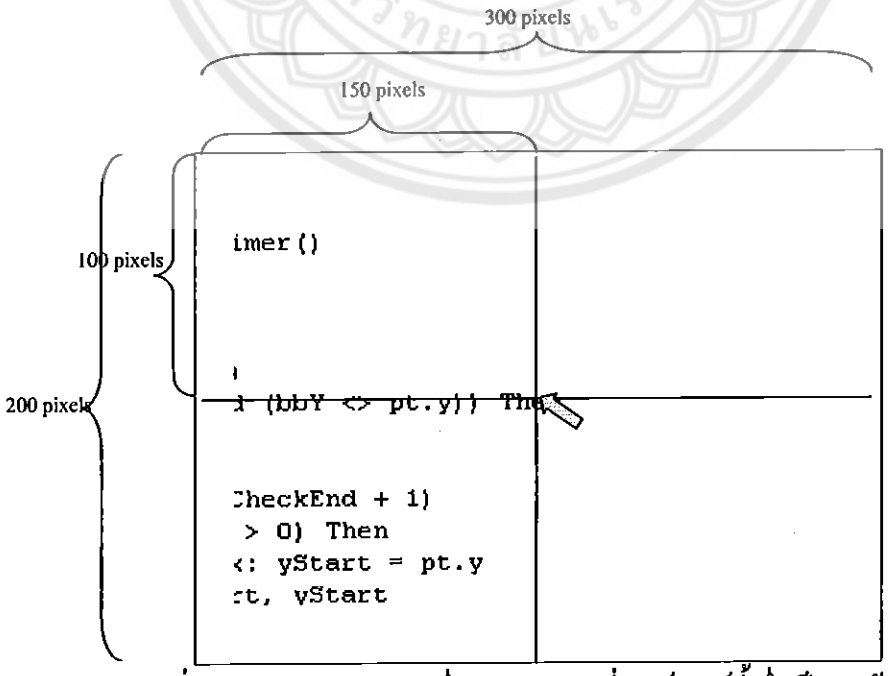
Private Sub MakeToGray(ByVal picColor As PictureBox) และคำสั่ง MakeToGray() นั้นเป็นคำสั่งที่ทำให้ภาพสี กลายเป็นขาวดำ จะได้ดังนี้



รูปที่ 2.4 แสดงการใช้คำสั่ง SetBitMap () และ MakeToGray()

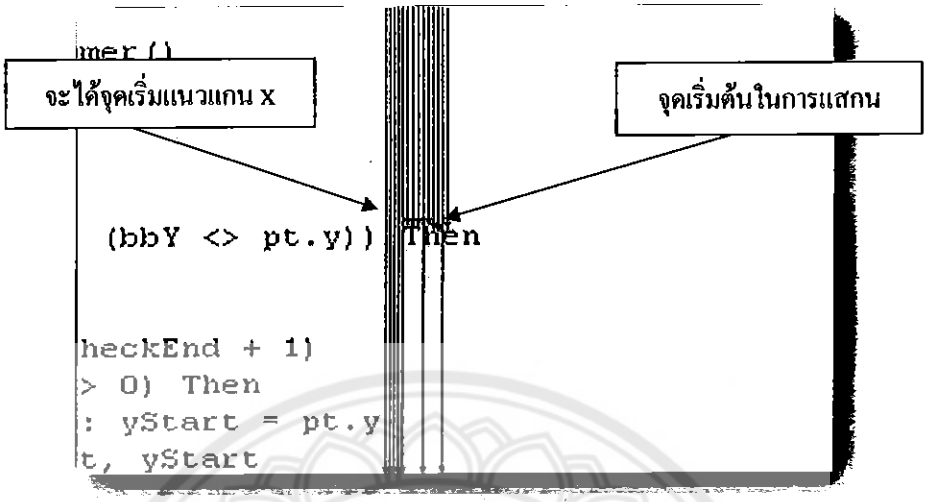
2.1.2.2 กระบวนการแปลงภาพเป็นภาพตัวอักษรย่อย

หลักการคือ เมื่อเราต้องการข้อความไหนก็นำเคอร์เซอร์ชี้ที่คำๆนั้น จึงทำให้ คำๆนั้นอยู่ตรงกลางระหว่างภาพขนาด 300 x 200 pixel ดังนั้นจะทำการสแกนจากจุดกึ่งกลางภาพนั้นขึ้นไปด้านบนก่อน เพื่อหาขอบบนของภาพ สามารถอธิบายได้ดังรูปต่อไปนี้



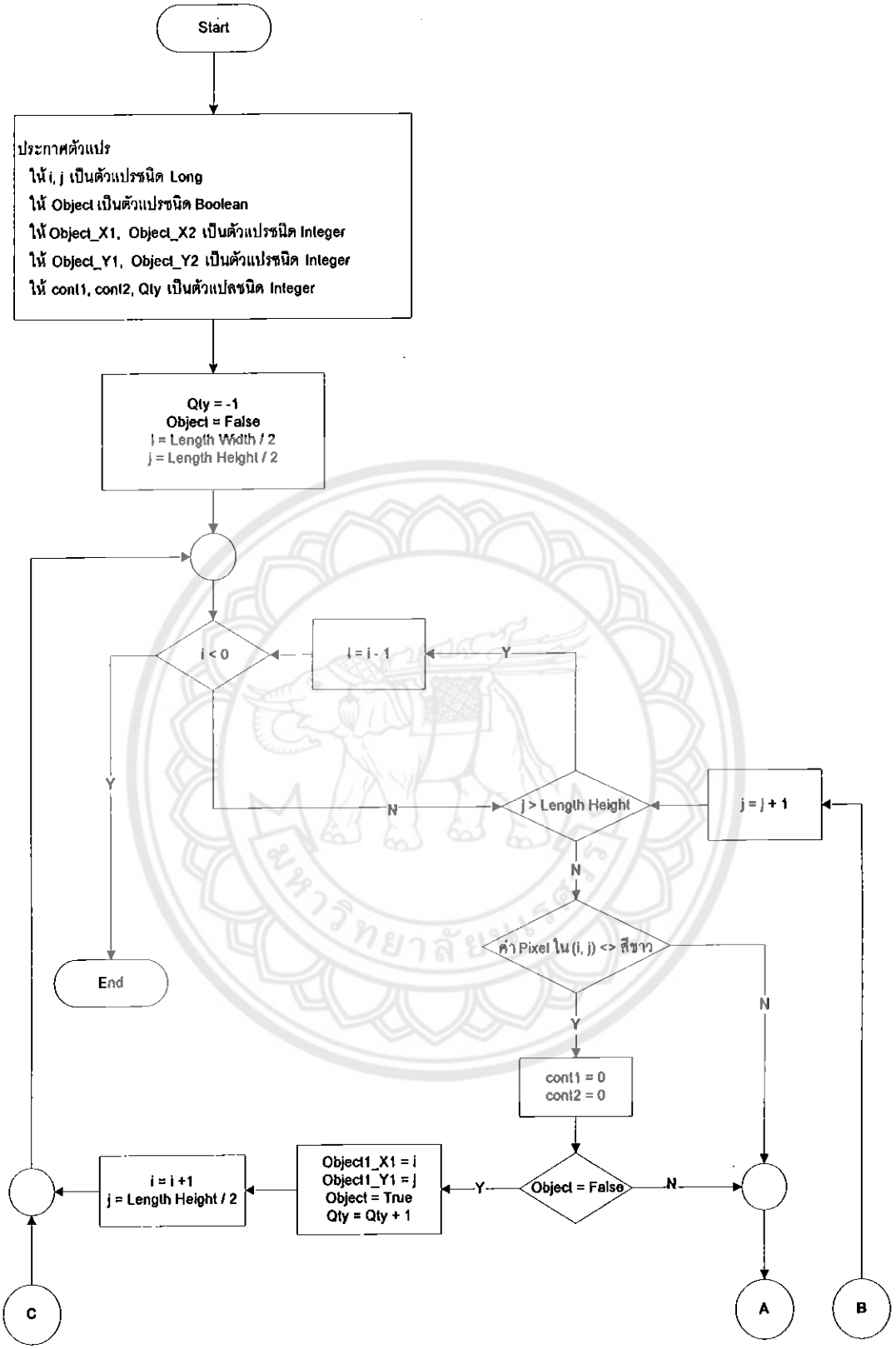
รูปที่ 2.5 แสดงการแบ่งจุดกึ่งกลางจากจุดที่เคอร์เซอร์ชี้ ซึ่งเป็นจุดอ้างอิง

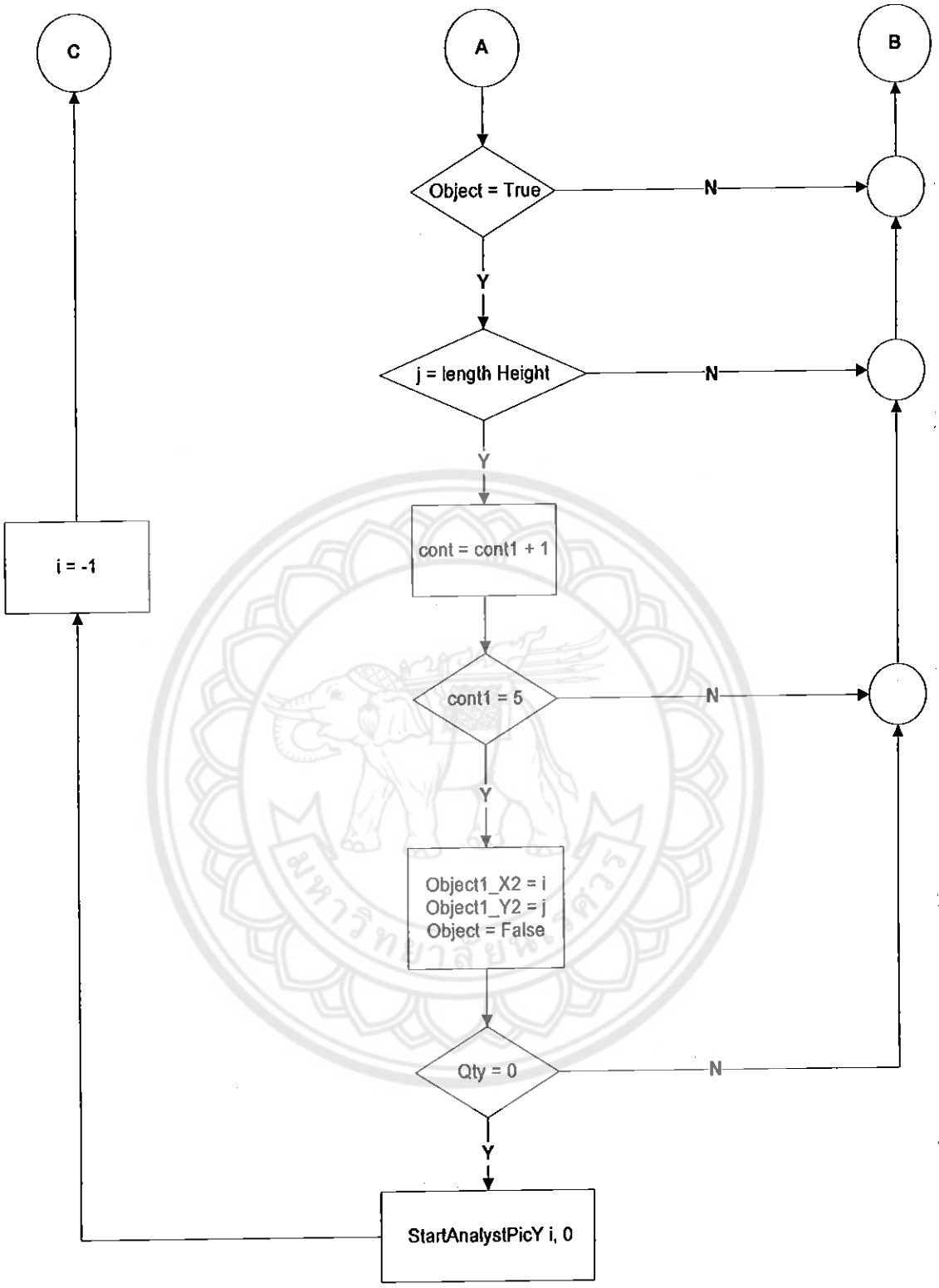
หลังจากที่หาจุดที่ตำแหน่งเมาส์ชี้ได้แล้ว ต่อไปจะทำการวิเคราะห์หาจุดเริ่มต้นของแนวแกน y เพื่อใช้เป็นตำแหน่งอ้างอิงของแนวแกน x และการวิเคราะห์หาแกน y นั้น



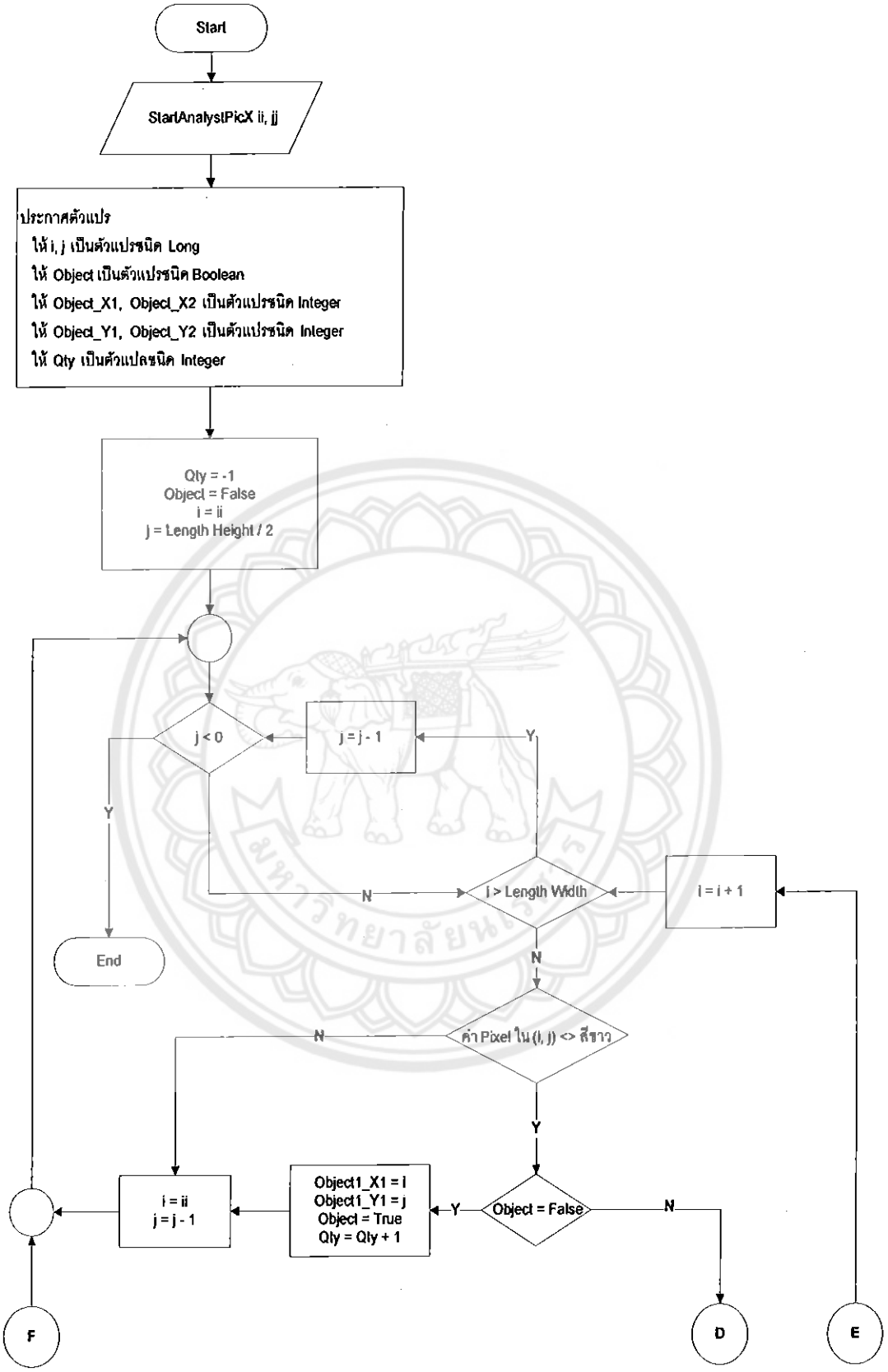
รูปที่ 2.6 แสดงการวิเคราะห์เพื่อหาแนวแกน x

จากรูปเป็นการจำลองการทำงานของอัลกอริทึม สำหรับเส้นลูกศรเป็นเพียงเส้นอ้างอิงเท่านั้น จะสังเกตเห็นว่า จากเส้นลูกศรที่เป็นการสแกนแนวตั้งที่เป็นพิกเซลสีขาวจะทอดยาว หมายถึงความยาวในแนวตั้งหรือคู่อันดับแนวแกน x จะทำการแบ่งครึ่งของความยาวเพื่อจะใช้เป็นจุดอ้างอิงของแนวแกน y ของภาพ และในส่วนของการหาจุดอ้างอิง y ก็เพื่อหลีกเลี่ยงไอคอนที่อยู่ด้านหน้าคำๆ นั้น เพราะจะทำให้เกิดข้อผิดพลาดขึ้นได้ จะใช้อัลกอริทึมต่อนี้

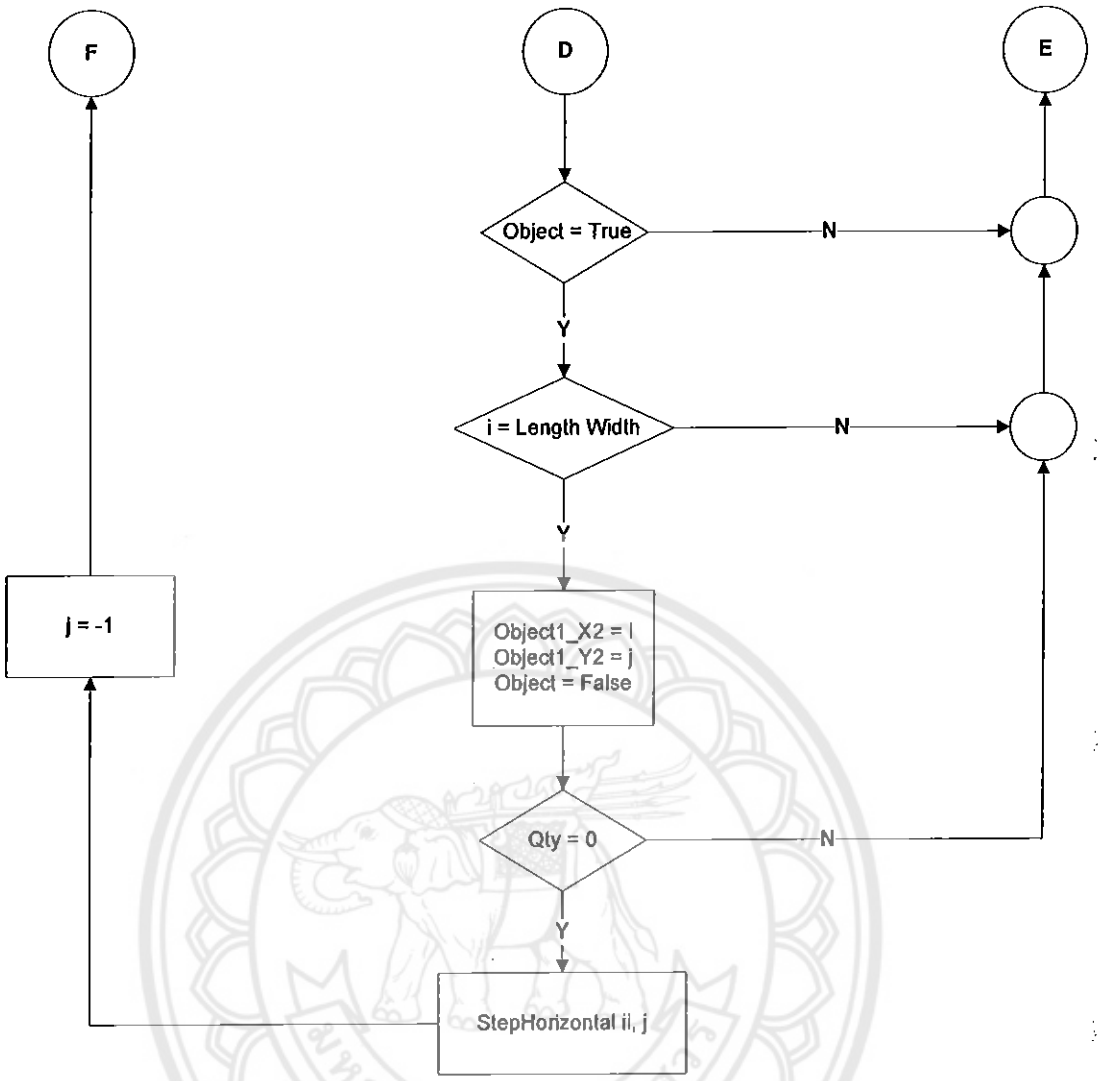

Start
ประกาศตัวแปร
ให้ i, j เป็นตัวแปรชนิด Long
ให้ Object เป็นตัวแปรชนิด Boolean
ให้ Object_X1, Object_X2 เป็นตัวแปรชนิด Integer
ให้ Object_Y1, Object_Y2 เป็นตัวแปรชนิด Integer
ให้ cont1, cont2, Qty เป็นตัวแปลชนิด Integer
Qty = -1
Object = False
i = Length Width / 2
j = Length Height / 2
i < 0
i = i - 1
Y
Y
N
j > Length Height
j = j + 1
N
End
ค่า Pixel ใน (i, j) <> สีขาว
N
Y
cont1 = 0
cont2 = 0
Object = False
Y
N
Object1_X1 = i
Object1_Y1 = j
Object = True
Qty = Qty + 1
i = i +1
j = Length Height / 2
C
A
B



รูปที่ 2.7 Flow Chart ของการวิเคราะห์จุดอ้างอิงแกน x

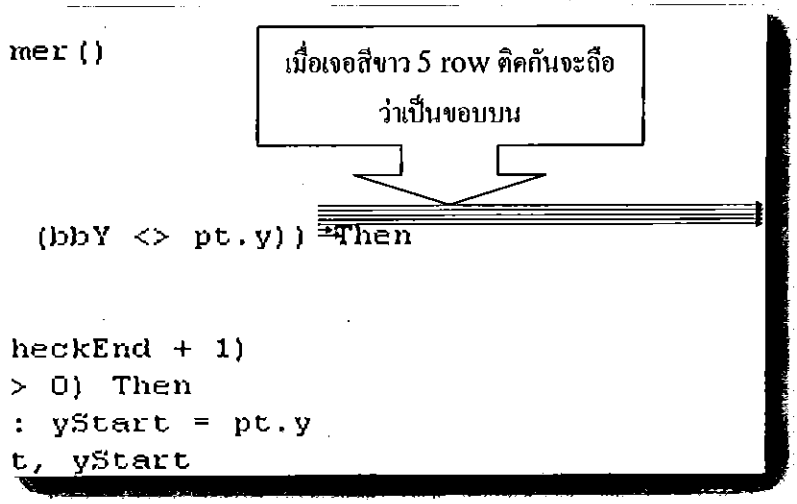


รูปที่ 2.8 Flow Chart ของการวิเคราะห์จุดอ้างอิง y



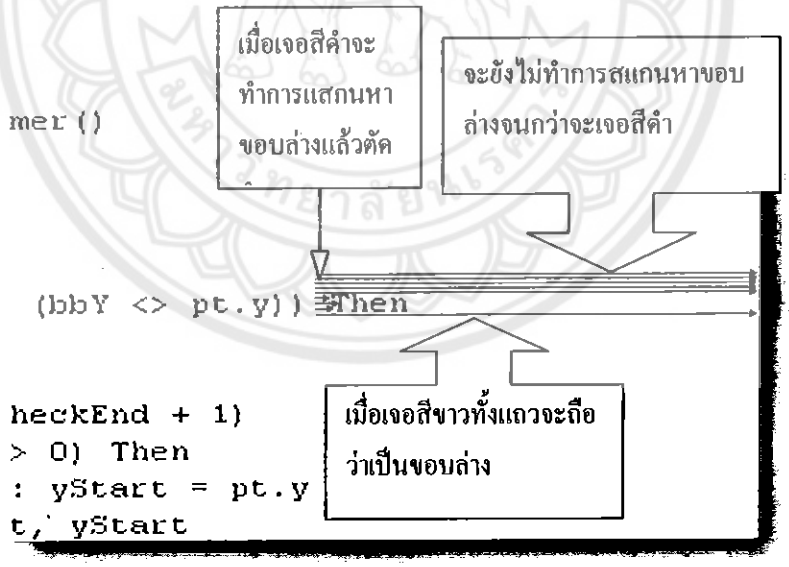
รูปที่ 2.8 Flow Chart ของการวิเคราะห์จุดอ้างอิง y (ต่อ)

จากอัลกอริทึมด้านบนนั้น มีหลักการคือ นำค่าที่แบ่งได้จากแนวแกน x เป็นจุดอ้างงอิงแล้วทำการสแกนขึ้นด้านบน โดยมีเงื่อนไขว่า ถ้าหากแถวในแนวการสแกนขึ้นหรือตามแนวแกน y ของแถวนั้นมีพิกเซลสีดำให้ทำการแสกนใหม่ จนกว่าจะพบแถวของพิกเซลสีขาวเป็นจำนวน 5 แถว และถือว่าแถวที่5 ของแถวพิกเซลสีขาวเป็นขอบบน (Top Margin) ดังแสดงในรูปต่อไปนี้



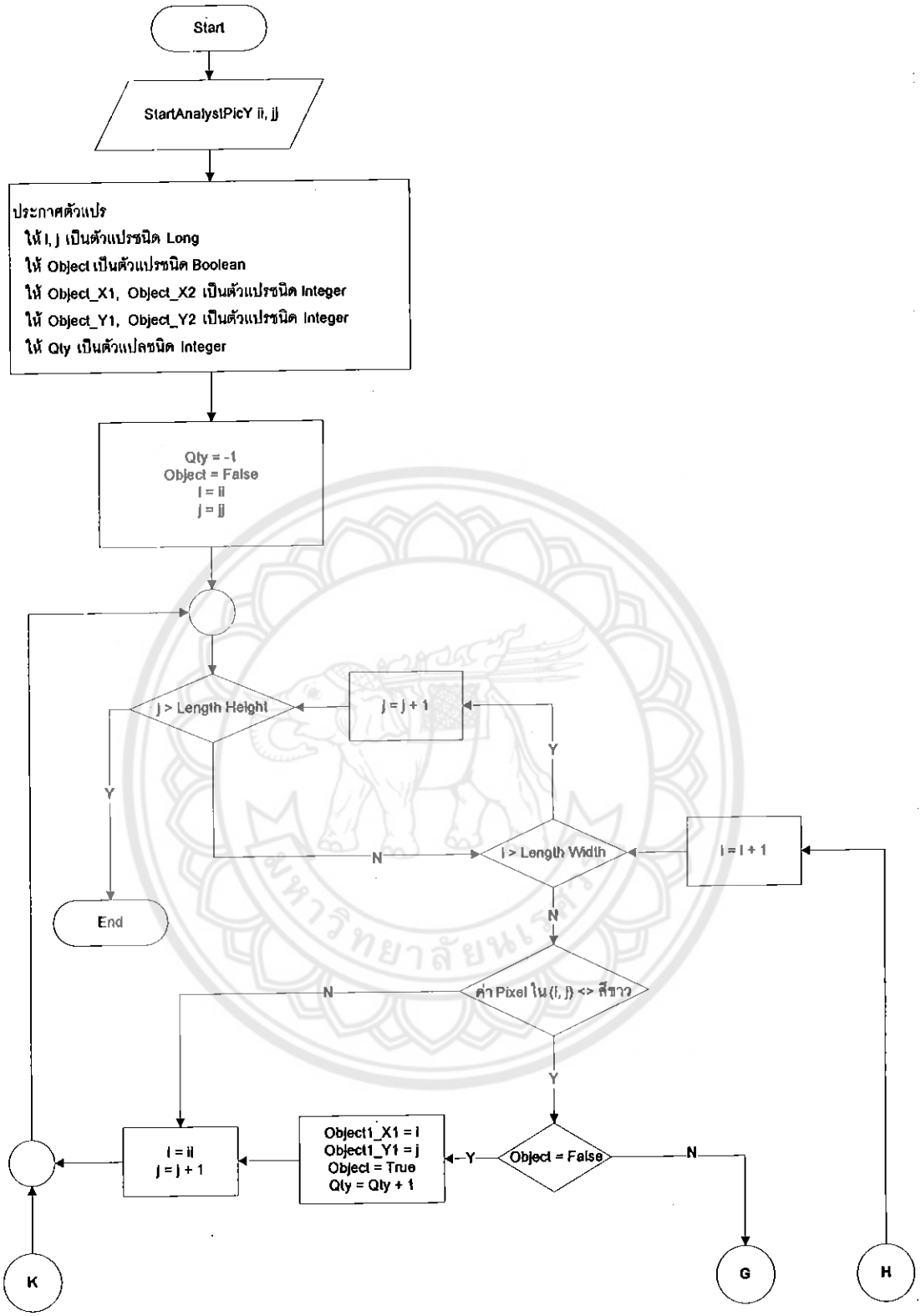
รูปที่ 2.9 แสดงการหาขอบบนของภาพ

จากอัลกอริทึมข้างต้น เราจะได้ค่าของแกน x และ y เพื่อที่จะได้นำค่าเหล่านี้มาทำการหาขอบบนและขอบล่างเพื่อตัดเอาเพียงส่วนที่ต้องการเท่านั้น โดยการกำหนดเริ่มต้นคือ (จุดเริ่มต้นแนวแกน x, จุดเริ่มต้นแนวแกน y) โดยการแสกนหาสีดำ แต่ถ้าหากเจอสีขาวทั้งแถวก็ให้ตัดออกมาเลย ซึ่งจะใช้อัลกอริทึงดังต่อไปนี้

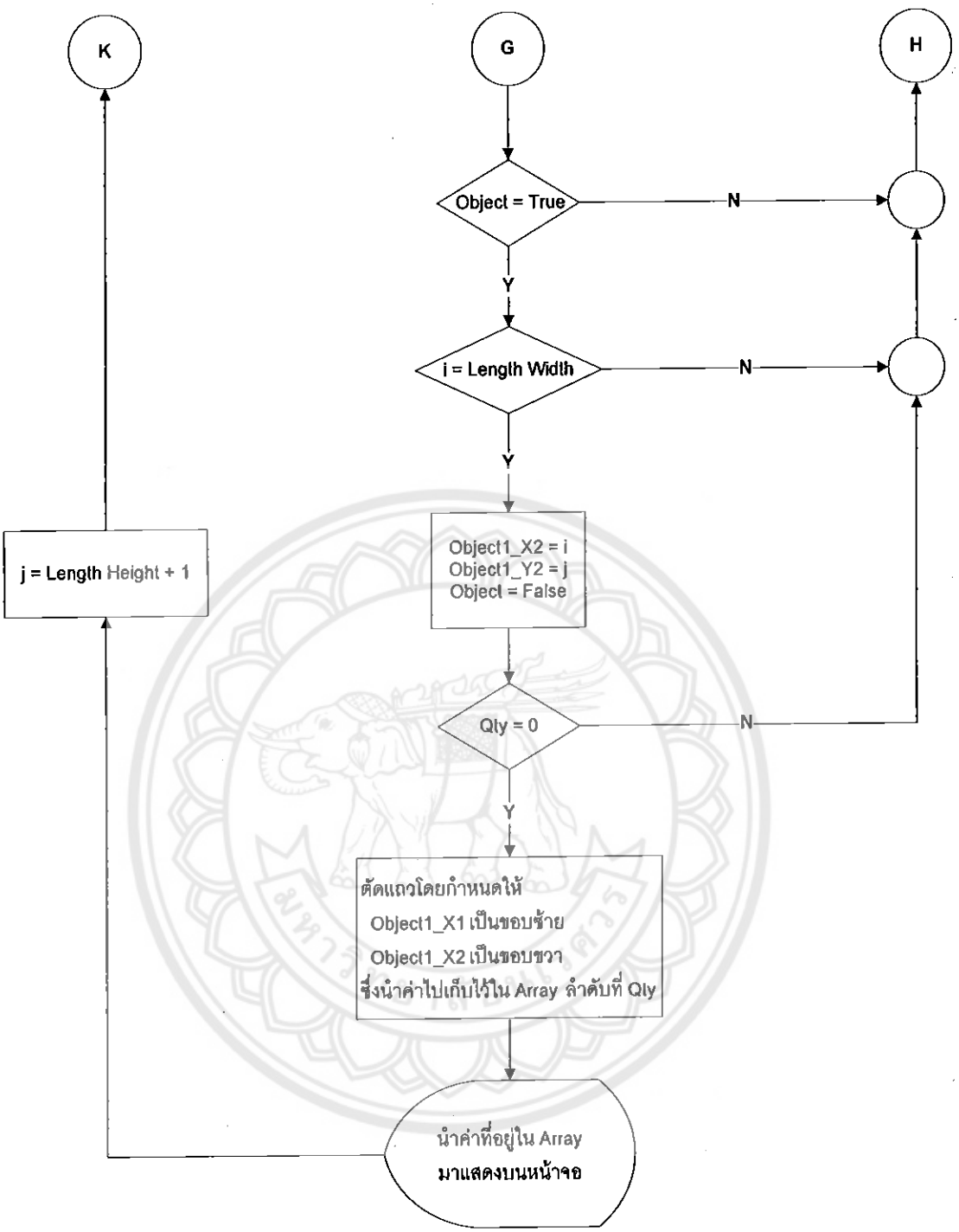


รูปที่ 2.11 แสดงการหาขอบบนและขอบล่างตามแนวแกน y

ซึ่งสามารถเขียนเป็นFlow Chart ได้ดังนี้

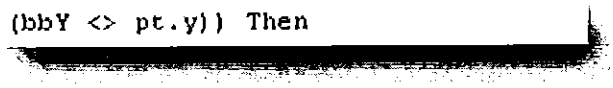


รูปที่ 2.10 Flow Chart แสดงการหาขอบบนและขอบล่าง



รูปที่ 2.10 Flow Chart แสดงการหาขอบบนและขอบล่าง (ต่อ)

และผลลัพธ์สุดท้ายจะได้ดังนี้

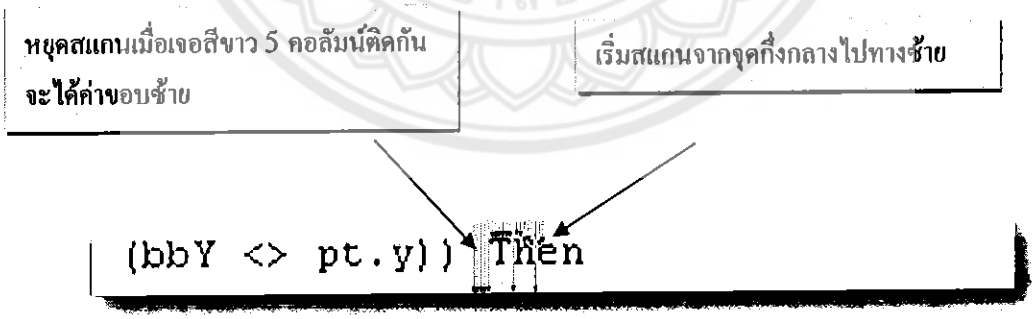


รูปที่ 2.12 ผลลัพธ์การจากตัดขอบบนและขอบล่าง

กระบวนการที่ผ่านมา เรียกว่า กระบวนการสแกนแนวนอน (Horizontal Scan) จากนั้นจะนำผลที่ได้จากกระบวนการสแกนแนวนอน มาใช้เพื่อแยกภาพที่ได้ ให้กลายเป็นภาพย่อย ตัวอย่าง คำว่า Then เป็นภาพเพียงภาพเดียว เราจะนำมาแยก ออกเป็น ภาพของอักษร T ภาพของอักษร h ภาพของอักษร e และภาพของอักษร n ตามลำดับ เราจะเรียกการสแกนแบบนี้ว่า กระบวนการสแกนแนวตั้ง (Vertical Scan) ซึ่งการสแกนแนวตั้ง จะเริ่มต้นจากจุดกึ่งกลางของภาพดังภาพที่แสดงต่อไปนี้

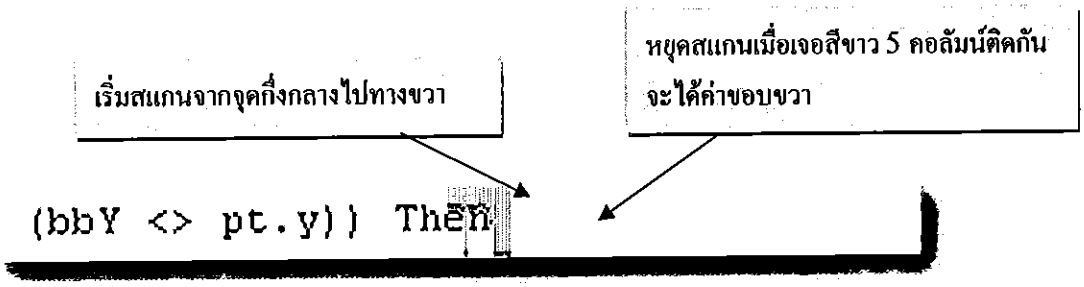
รูปที่ 2.13 แสดงเส้นประกึ่งกลางภาพ

โดยจะสแกนไปทางด้านซ้ายก่อน โดยมีเงื่อนไขว่า ถ้าสแกนพบสีขาวจำนวน 5 แถวติดต่อกันตามแนวตั้ง จะถือว่าจุดนั้นเป็นขอบซ้าย (Left Margin) และเก็บค่าขอบซ้ายไว้ ดังรูปข้างล่างนี้



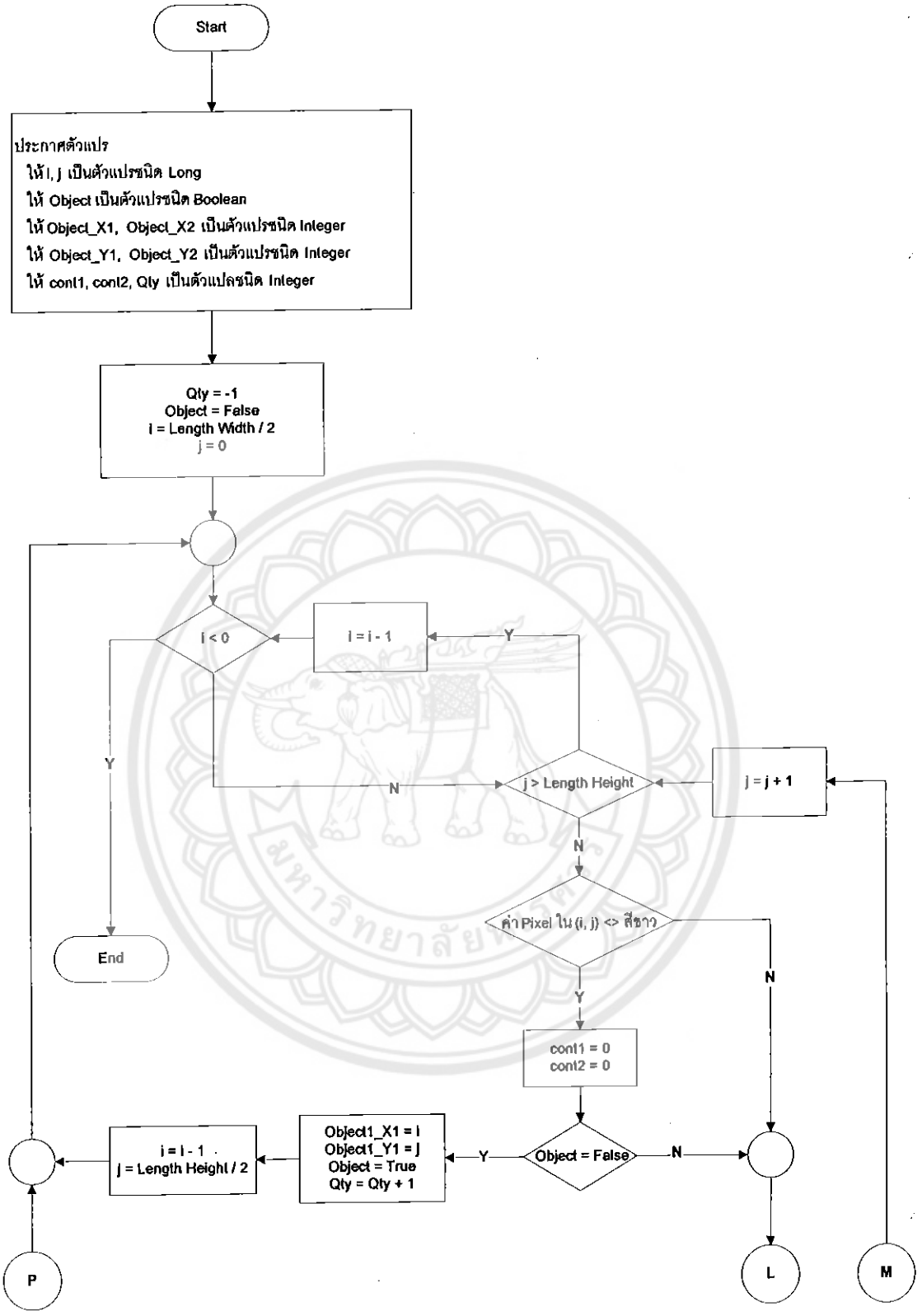
รูปที่ 2.14 แสดงการสแกนหาขอบซ้าย (Left Margin)

หลังจากนั้นสแกนจากจุดกึ่งกลางมาทางด้านขวาเงื่อนไขก็เช่นเดิม คือถ้าพบแถวที่เป็นสีขาวจำนวน 5 แถวติดต่อกันตามแนวตั้งจะถือว่าเป็นขอบขวา (Right Margin)

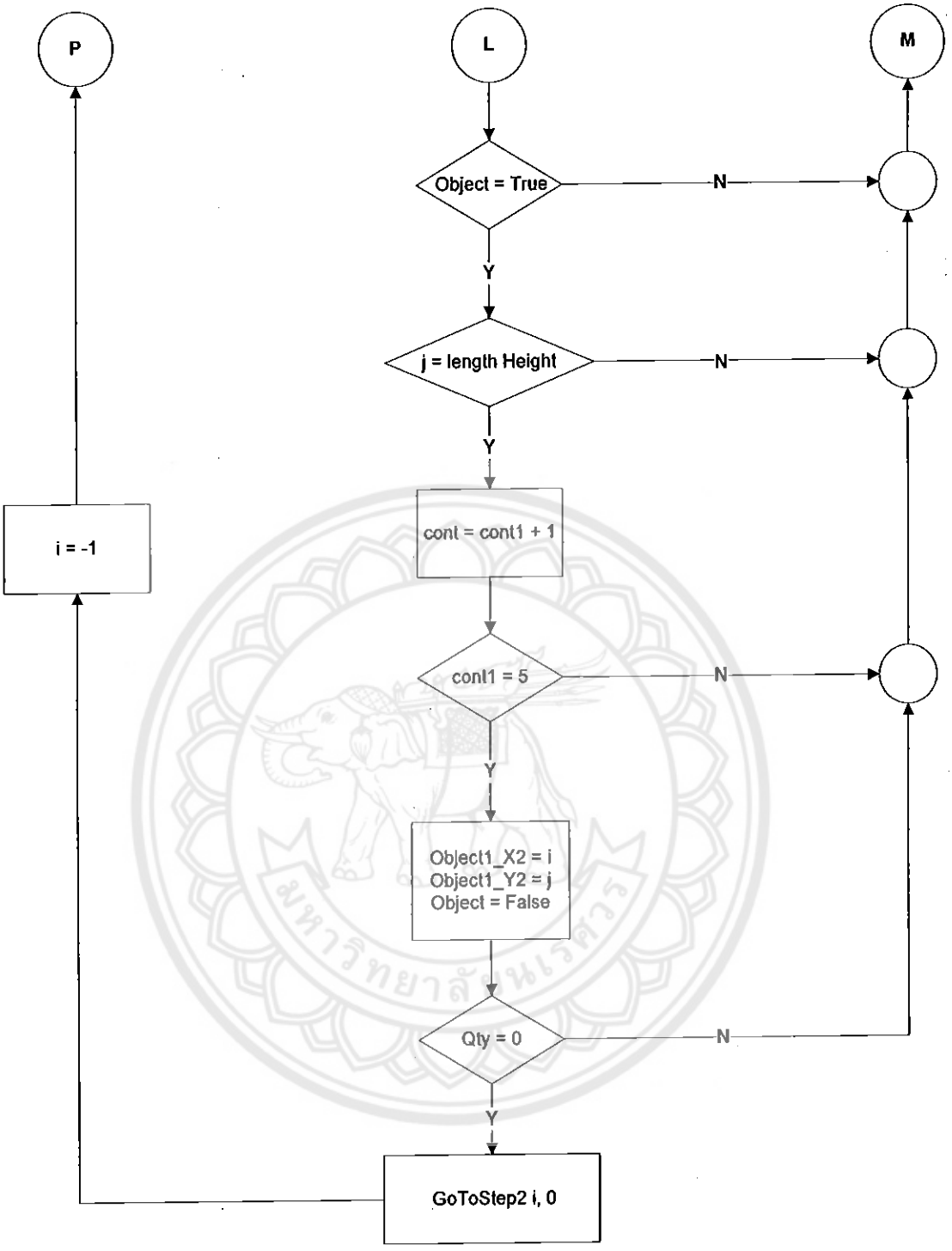


รูปที่ 2.15 แสดงการสแกนหาขอบขวา (Right Margin)

เมื่อได้ค่าของขอบซ้าย-ขวา แล้ว ก็จะทำการสแกนใหม่อีกครั้ง โดยการสแกนครั้งนี้จะตัดตัวอักษร โดยเงื่อนไขของการสแกนคือจะตัดตัวอักษร เมื่อพบ Pixel สีขาวในแนวตั้ง 1 แถว ก็จะตัดตัวอักษร และจะทำอย่างนี้ไปเรื่อยๆเมื่อพบแถวสีขาว และมีอัลกอริทึมดังนี้

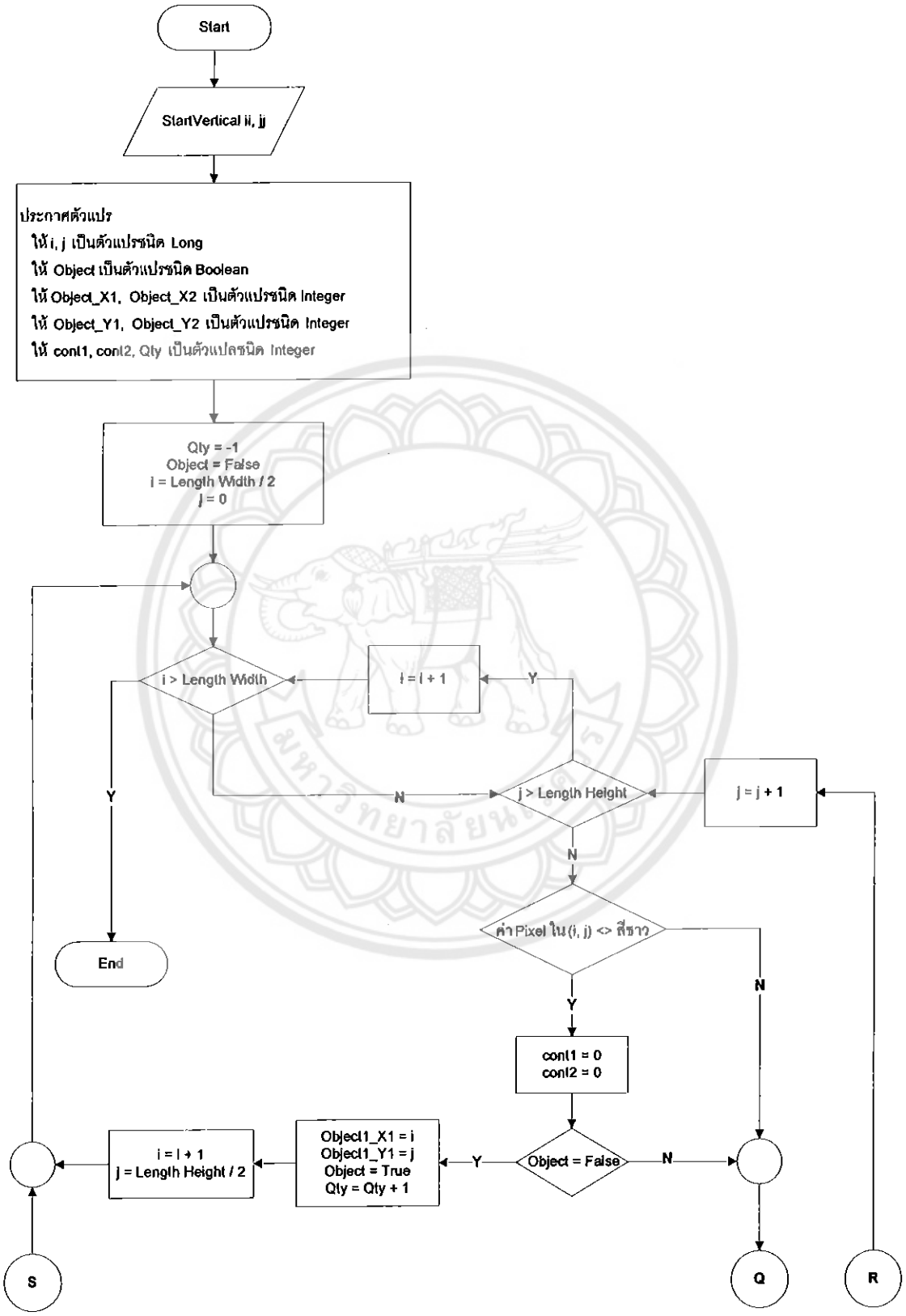


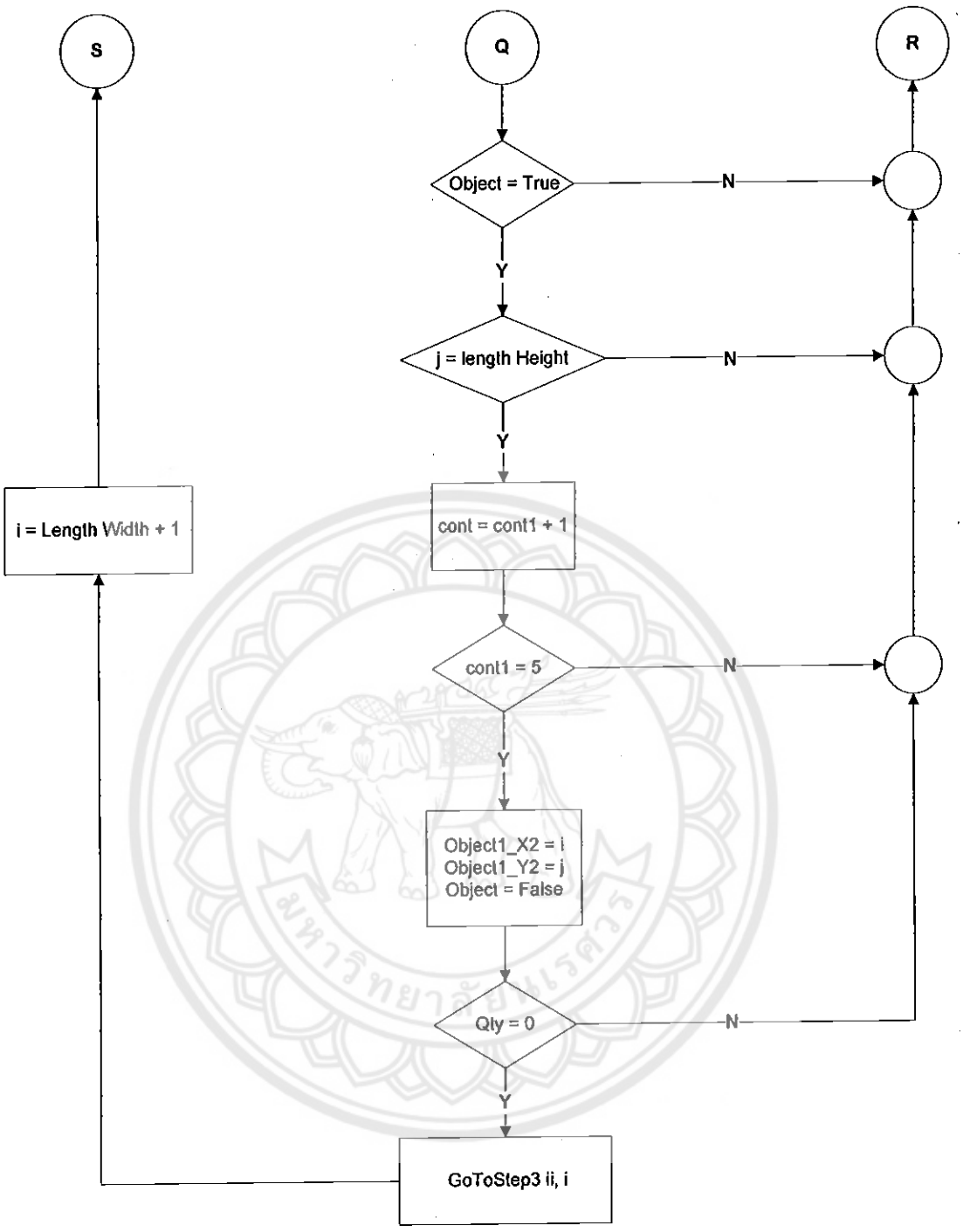
รูปที่ 2.16 Flow Chart การหาขอบซ้าย-ขวาและการแยกภาพตัวอักษร



รูปที่ 2.16 Flow Chart การหาขอบซ้าย-ขวาและการแยกภาพตัวอักษร (ต่อ)

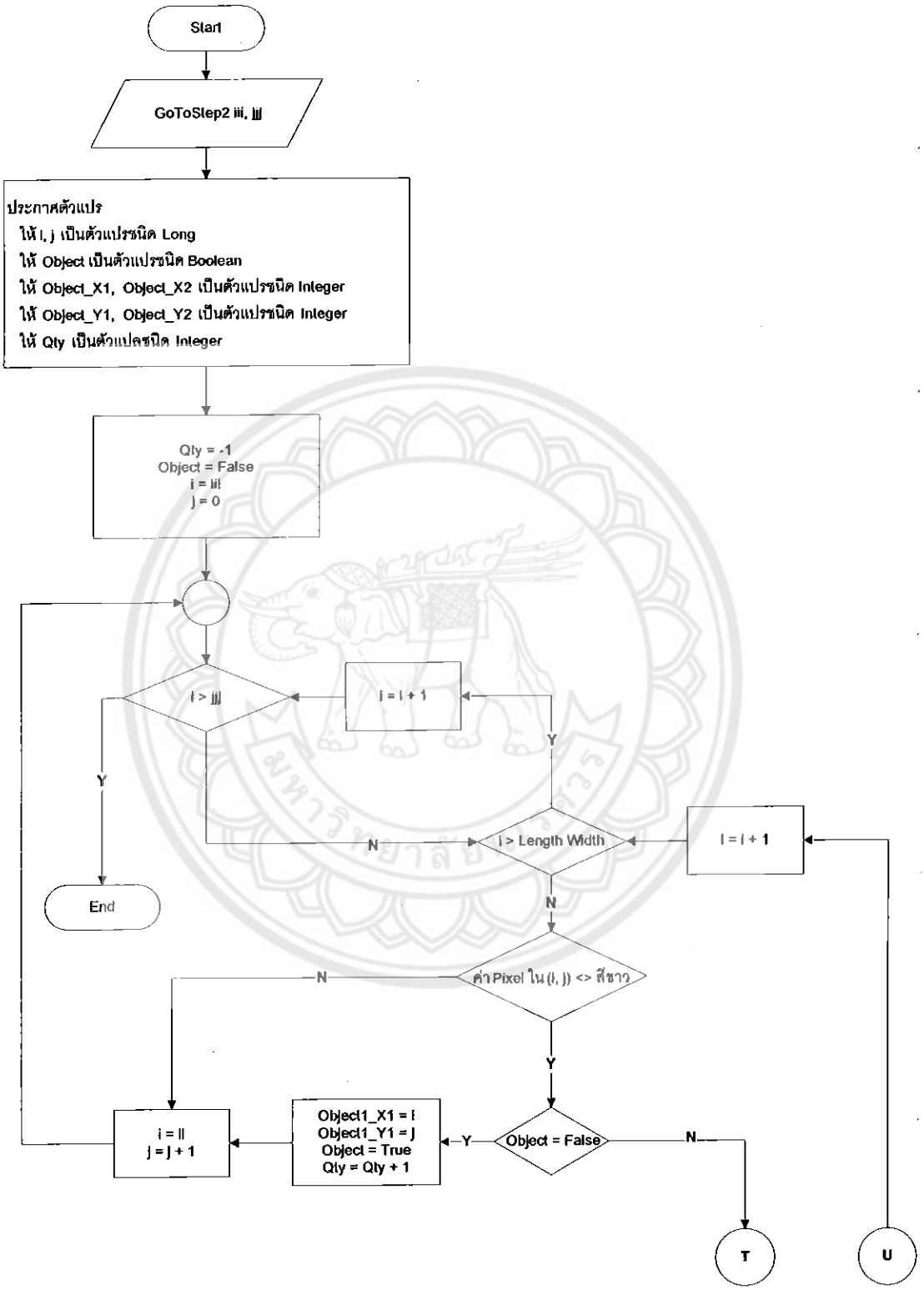
และ Flow Chart ของ GoToStep2 Algorithm มีดังนี้

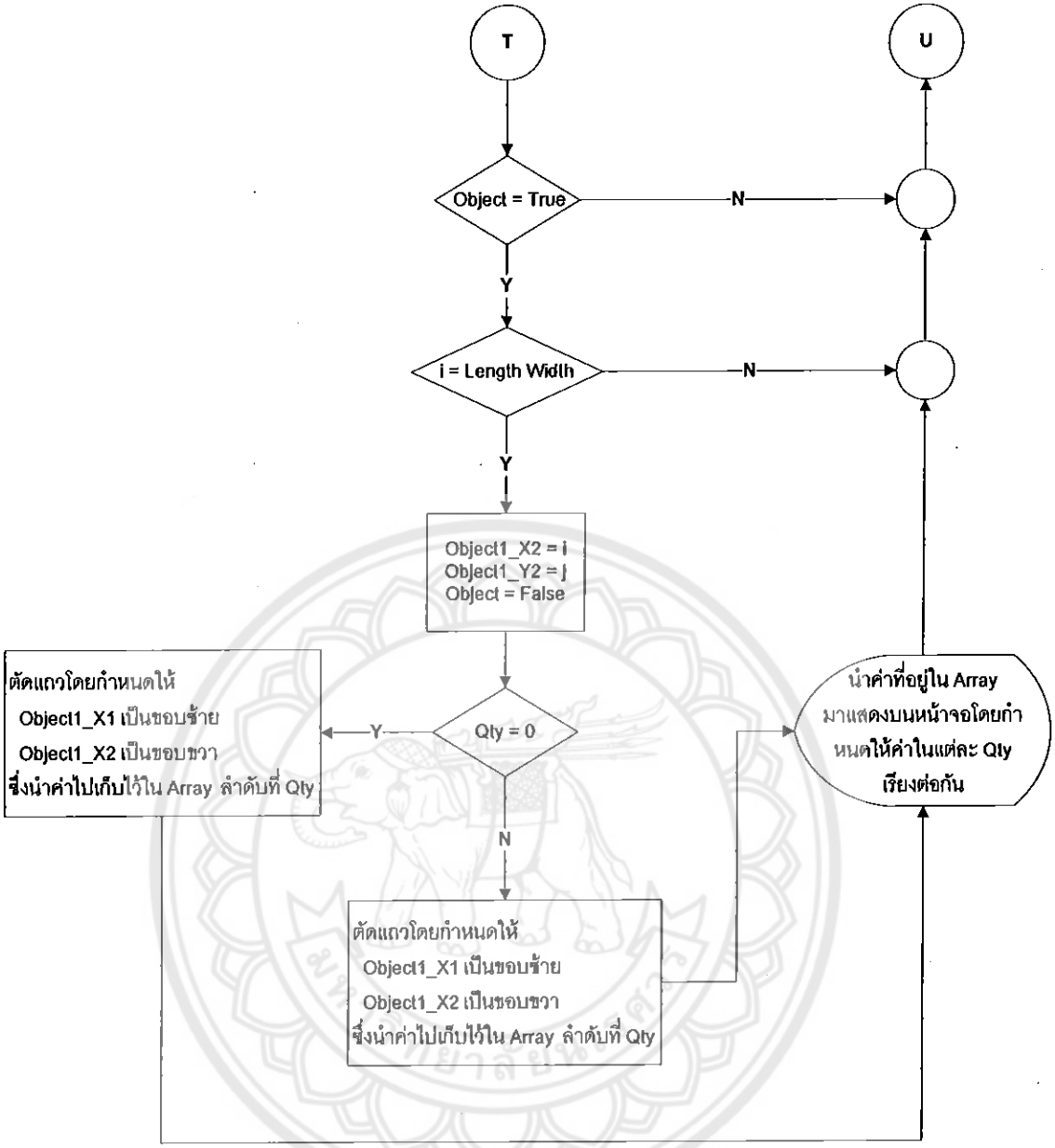




รูปที่ 2.17 Flow Chart ของ GoToStep2 Algorithm

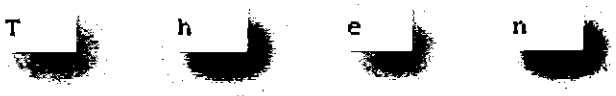
และอัลกอริทึมสุดท้ายคือ





รูปที่ 2.18 Flow Chart ของ GoToStep3 Algorithm

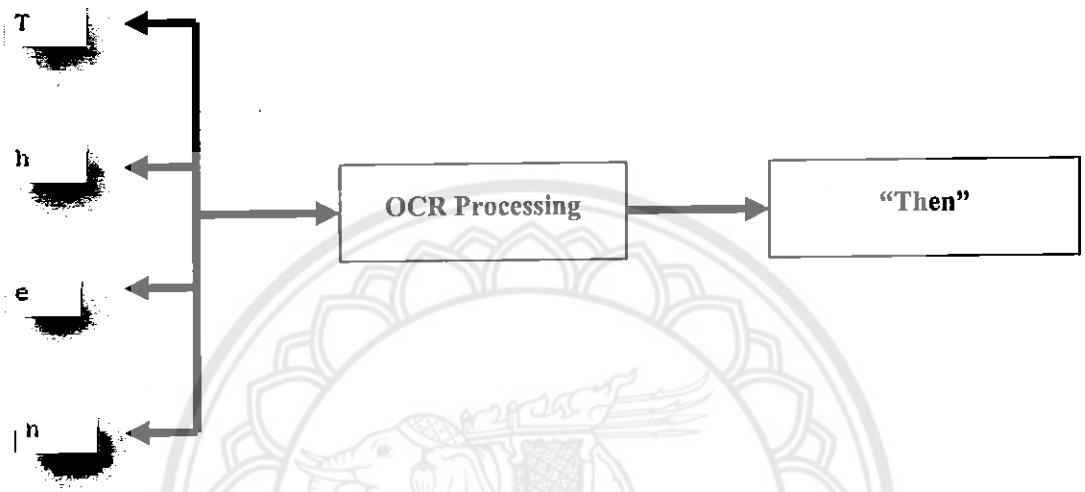
และผลลัพธ์สุดท้าย คือ



รูปที่ 2.19 แสดงตัวอักษรหลังการใช้ Vertical Scan

2.1.2.3 การนำตัวอักษรเข้าสู่กระบวนการ OCR

ฟังก์ชันของ OCR นั้น เป็น open source ที่นำมาจากแหล่งที่มาอื่น แต่มีการประกาศและเรียกใช้ดังนี้ Public Declare Function OCR Lib "OCR.dll" (ByVal file As String, ByVal imageType As Long) As String และสามารถอธิบายให้ง่ายต่อการเข้าใจได้ดังนี



รูปที่ 2.20 การแปลงภาพเป็นอักษรของกระบวนการ OCR

2.2 ฐานข้อมูลเบื้องต้น (Foundation of Database)

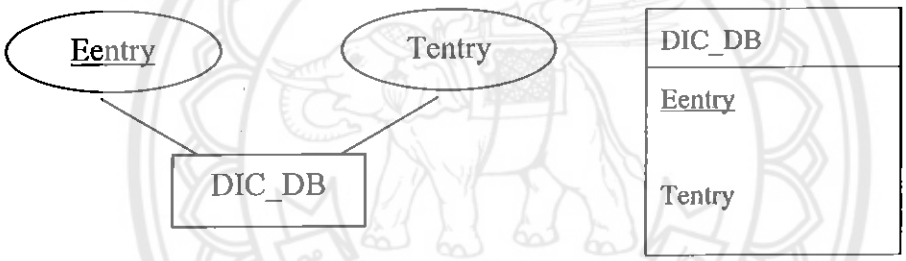
2.2.1 ความหมายของฐานข้อมูล

ฐานข้อมูล หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องสร้างเก็บไว้ในแฟ้มเดียวกัน หรือแยกเก็บหลายๆ แฟ้มข้อมูล ที่สำคัญคือ จะต้องสร้างความสัมพันธ์ระหว่างระเบียนและเรียกใช้ความสัมพันธ์นั้นได้ ในฐานข้อมูลนั้นย่อมมีความซ้ำซ้อนของข้อมูล เราจำเป็นต้องกำจัดความซ้ำซ้อนของข้อมูลออกเสีย และเก็บข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง มีการควบคุมดูแลรักษาการใช้และผู้มีสิทธิ์จะใช้ข้อมูล เพราะบางข้อมูลอาจจะมีการใช้ที่จำกัดสิทธิ์ และบางข้อมูลสามารถใช้ร่วมกันได้ ระบบฐานข้อมูล หรือที่เราเรียกว่า DBMS (Data Base Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลโดยง่าย สะดวก และมีประสิทธิภาพ การเข้าถึงนั้นอาจจะเป็น การสร้าง การแก้ไข ฐานข้อมูล เป็นต้น

ความสำคัญของฐานข้อมูลคือ ช่วยลดการเก็บข้อมูลที่ซ้ำซ้อน รักษาความถูกต้องของข้อมูล สามารถใช้ข้อมูลร่วมกันได้ สามารถแก้ไข ตรวจสอบได้ง่ายขึ้น ลดระยะเวลาการแก้ไขให้น้อยลง และการป้องกันและรักษาความปลอดภัยของข้อมูล

2.2.2 การออกแบบฐานข้อมูล (Database Design)

เราต้องการสร้างฐานข้อมูลเพื่อเก็บรวบรวมคำศัพท์ทั้งหมดเพื่อนำมาประยุกต์ใช้กับโปรแกรมแปลคำศัพท์ หรือใช้งานอื่นๆ ขั้นตอนการออกแบบ คือ เราต้องกำหนด entity หรือถ้าในการเขียนโปรแกรมเราอาจจะเรียก การประกาศตัวแปร ในการประกาศ entity นั้น ก็เพื่อเป็นการสร้างตัวแปรมารองรับข้อมูลที่เราจะนำไปเก็บไว้นั้นเอง ในที่นี้ เราจะกำหนด entity ทั้งหมด 2 ตัวคือ Eentry และ Tentry เพื่อเก็บคำศัพท์และความหมาย ตามลำดับ เมื่อได้ entity มาแล้ว สิ่งต่อไปนั้นคือการกำหนดความสำคัญของ entity ก็คือการกำหนด Primary Key ซึ่งจะกำหนดให้ Eentry เป็น Primary Key สามารถเขียนเป็นแผนผังความสัมพันธ์ ได้ดังนี้



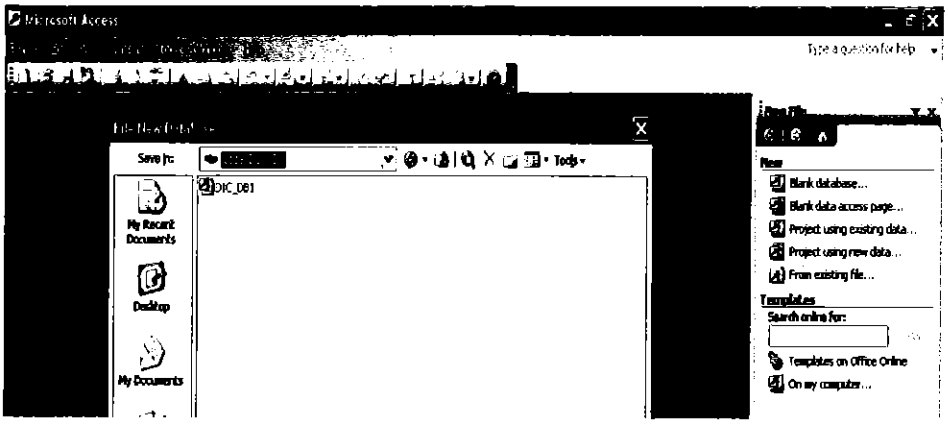
รูปที่ 2.21 แสดงความสัมพันธ์ของ E-R Diagram

2.2.3 วิธีการสร้างฐานข้อมูล

ฐานข้อมูลถือเป็นแหล่งเก็บสะสมรวบรวมคำศัพท์พร้อมทั้งความหมายเป็นส่วนที่ application นี้ใช้อ้างอิงถึงและมีการติดต่อระหว่าง application กับ database ในโครงงานนี้ฐานข้อมูลที่สร้างขึ้นเป็นฐานข้อมูลอย่างง่าย ไม่ซับซ้อนมากมาย การออกแบบฐานข้อมูลได้กล่าวไว้ในบทที่ 2 แล้วและมีวิธีการสร้างดังนี้

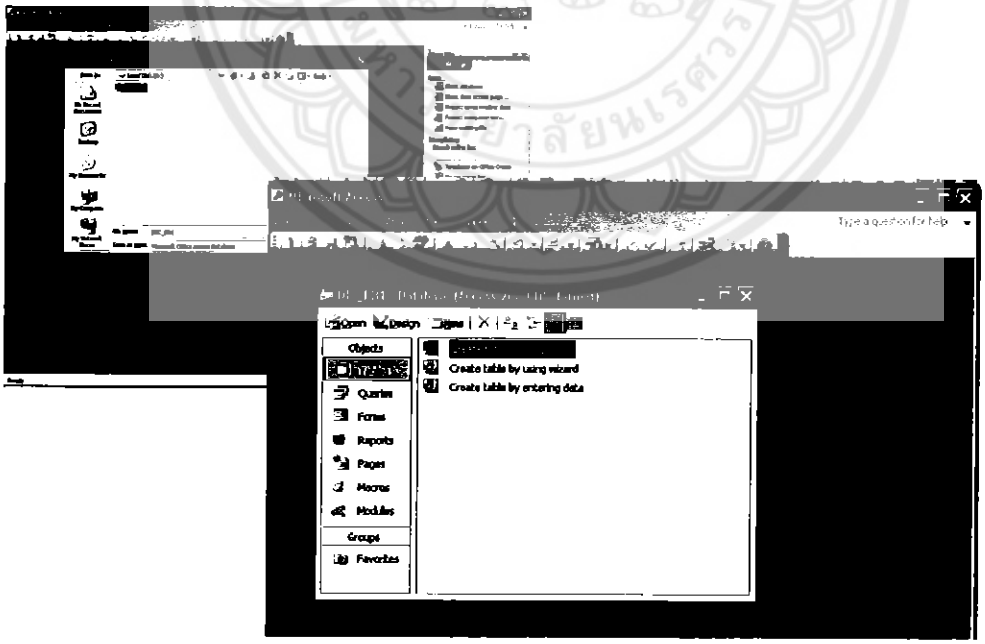
เมื่อเข้าสู่โปรแกรม Microsoft Access 2003 เราจะพบกับหน้าต่าง New File ที่ด้านขวาของหน้าต่างเป็นอันดับแรกซึ่งหน้าต่าง New File นี้ได้เก็บรวบรวมสิ่งต่างๆที่จำเป็นสำหรับการเริ่มต้นใช้งาน Access สำหรับขั้นตอนการสร้างฐานข้อมูลใหม่มีดังนี้

1. คลิกที่ Blank database บนหน้าต่าง New File
2. คลิกเลือกตำแหน่งสำหรับเก็บไฟล์ฐานข้อมูล



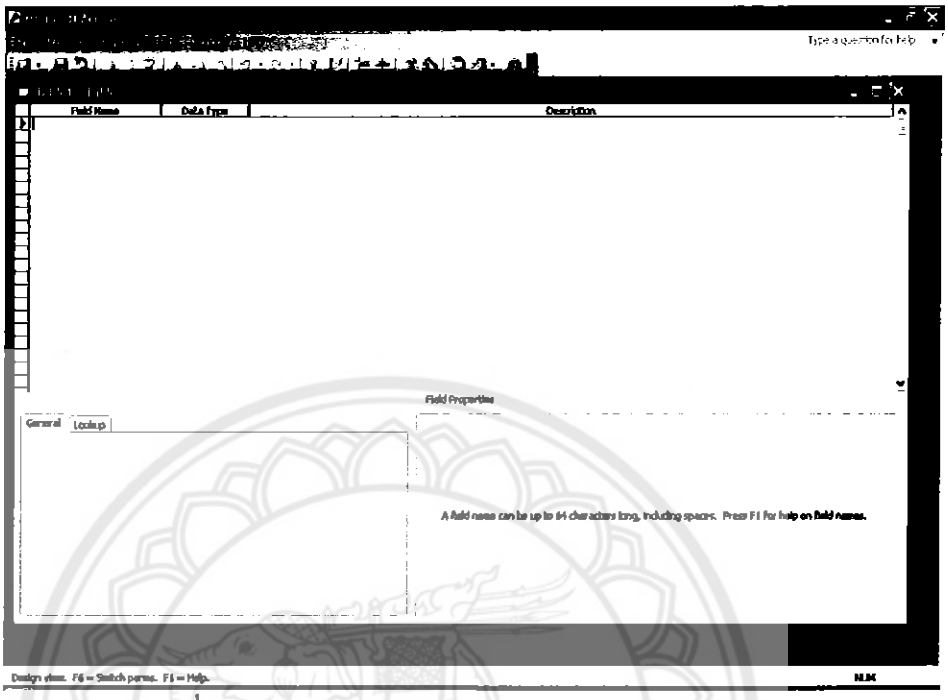
รูปที่ 2.22 แสดงการสร้างฐานข้อมูลใหม่

1. ตั้งชื่อให้กับฐานข้อมูล ใหม่ ควรจะตั้งชื่อให้สื่อความหมาย ซึ่งสามารถตั้งได้ทั้งภาษาไทย และภาษาอังกฤษ
2. คลิกปุ่ม Create ก็จะได้ฐานข้อมูลใหม่ตามที่ตั้งชื่อไว้ โดยจะปรากฏชื่อฐานข้อมูลที่หน้าต่างDatabase



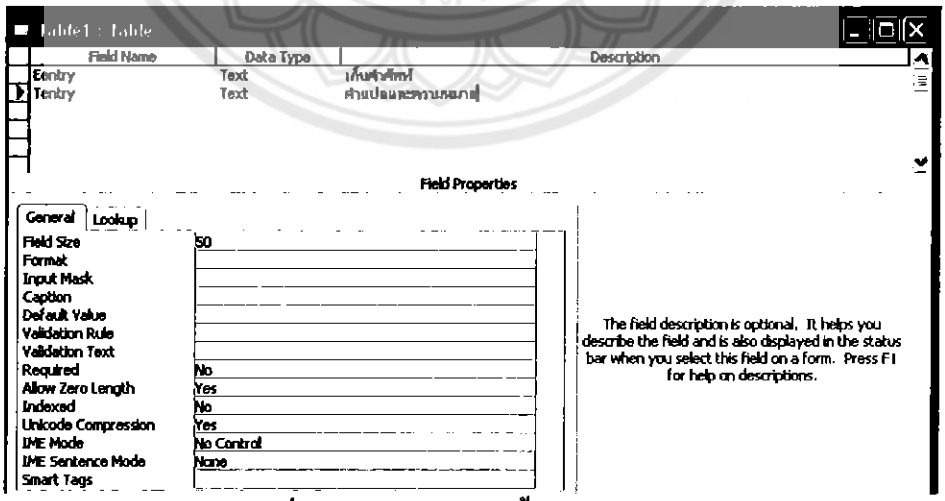
รูปที่ 2.23 แสดงชื่อและหน้าต่างที่ได้จากการสร้างฐานข้อมูล

3. คลิกที่แถบ Tables ทางด้านซ้ายมือ และดับเบิลคลิกที่แถบรายการ Create table in Design view จะปรากฏหน้าต่างของTables เพื่อให้ทำการ กำหนด Entity



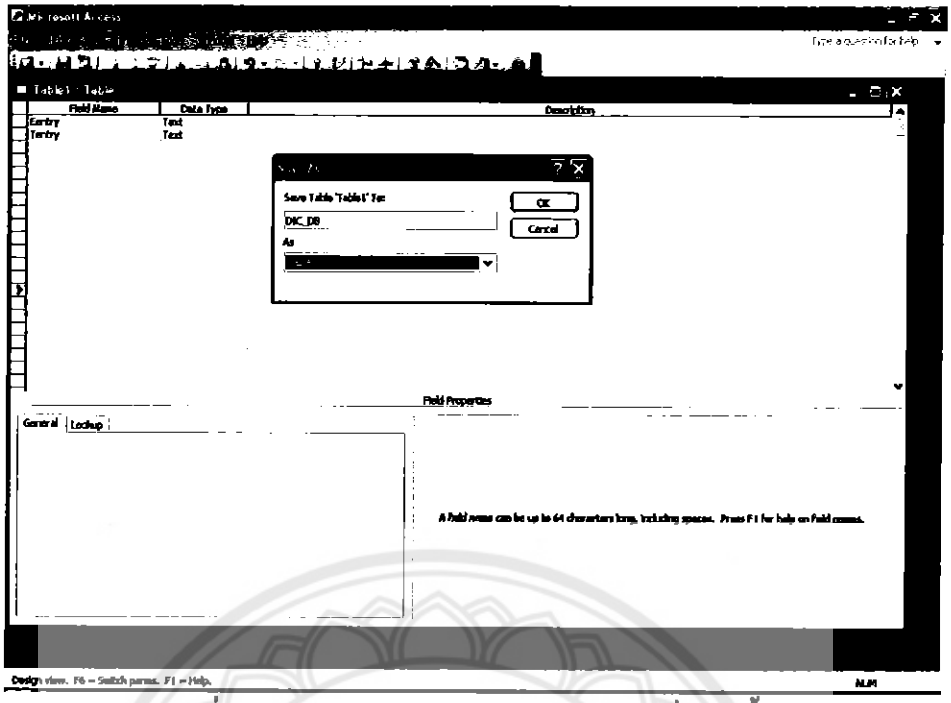
รูปที่ 2.24 แสดงหน้ากำหนดค่า Entity ของ Table

หลังจากที่หน้าTable ปรากฏขึ้นมานั้น จะมีการกำหนด Entity และ Primary Key ในที่นี้เราจะกำหนดให้ Eentry เป็น Primary Key ดังนี้



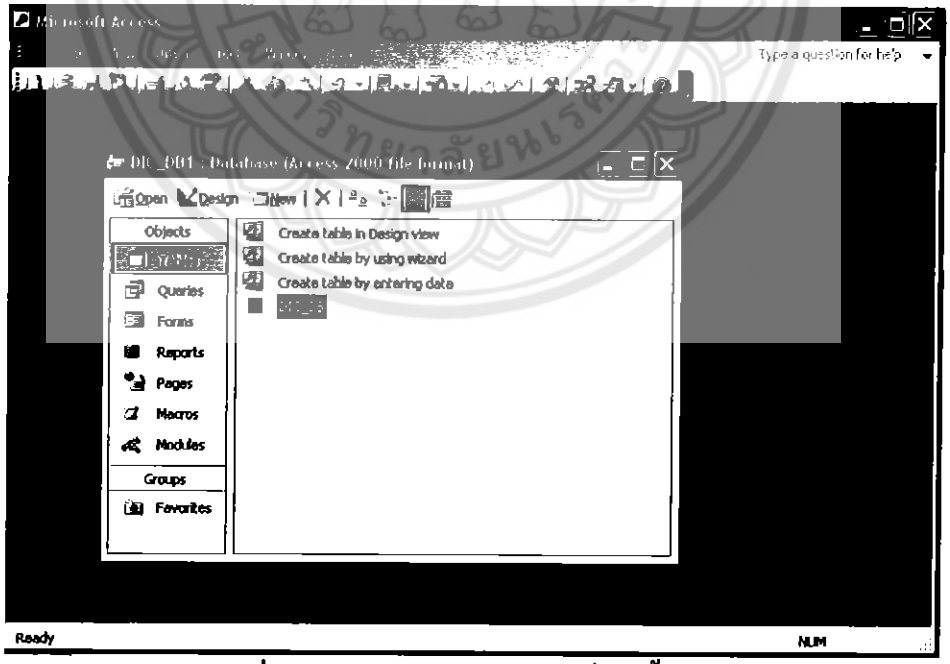
รูปที่ 2.25 แสดง Entity ทั้งหมดในตาราง

แล้วการ Save as...ข้อมูลตารางที่สร้างขึ้นมาแล้วจะปรากฏหน้าต่างต่อไปนี้



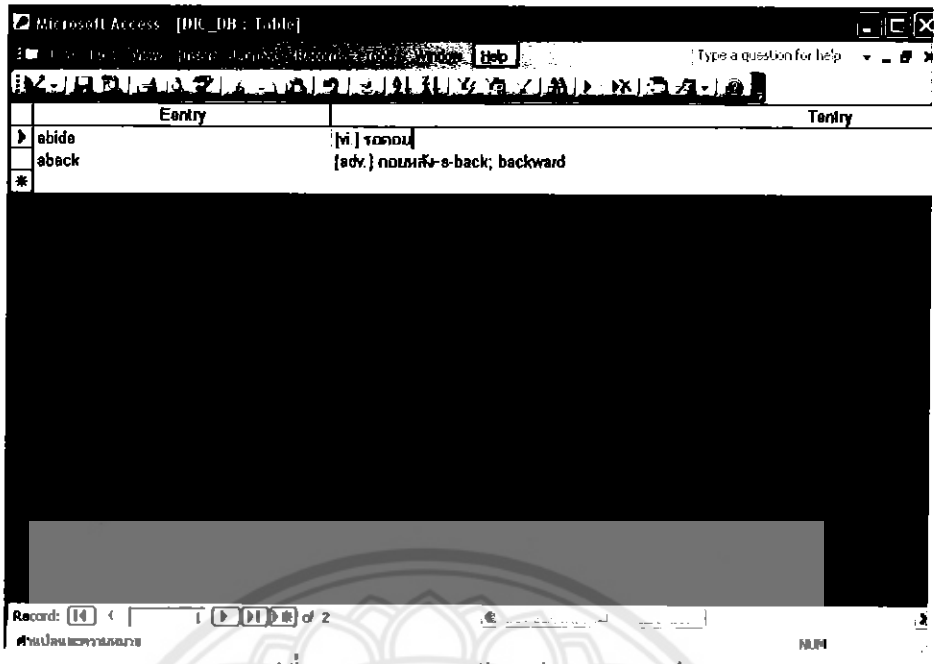
รูปที่ 2.26 แสดงหน้าต่างการ Save as Table ที่สร้างขึ้น

หลังจากทำการ Save as แล้ว และปิดหน้าต่าง Table จะปรากฏหน้าต่างพร้อมทั้งมี Table ใหม่ที่ชื่อว่า DIC_DB เกิดขึ้นเพิ่มที่หน้าต่างนี้



รูปที่ 2.27 แสดงตารางฐานข้อมูลที่เกิดขึ้นใหม่

ทำการ ดับเบิลคลิกที่ DIC_DB ก็จะปรากฏหน้าต่างสำหรับ การป้อนข้อมูลลงไป โดย ในคอลัมน์ Eentry จะป้อนคำศัพท์ ส่วน Tentry จะป้อนคำแปลหรือความหมายลงไป ดังนี้



รูปที่ 2.28 แสดงการป้อนข้อมูลคำศัพท์

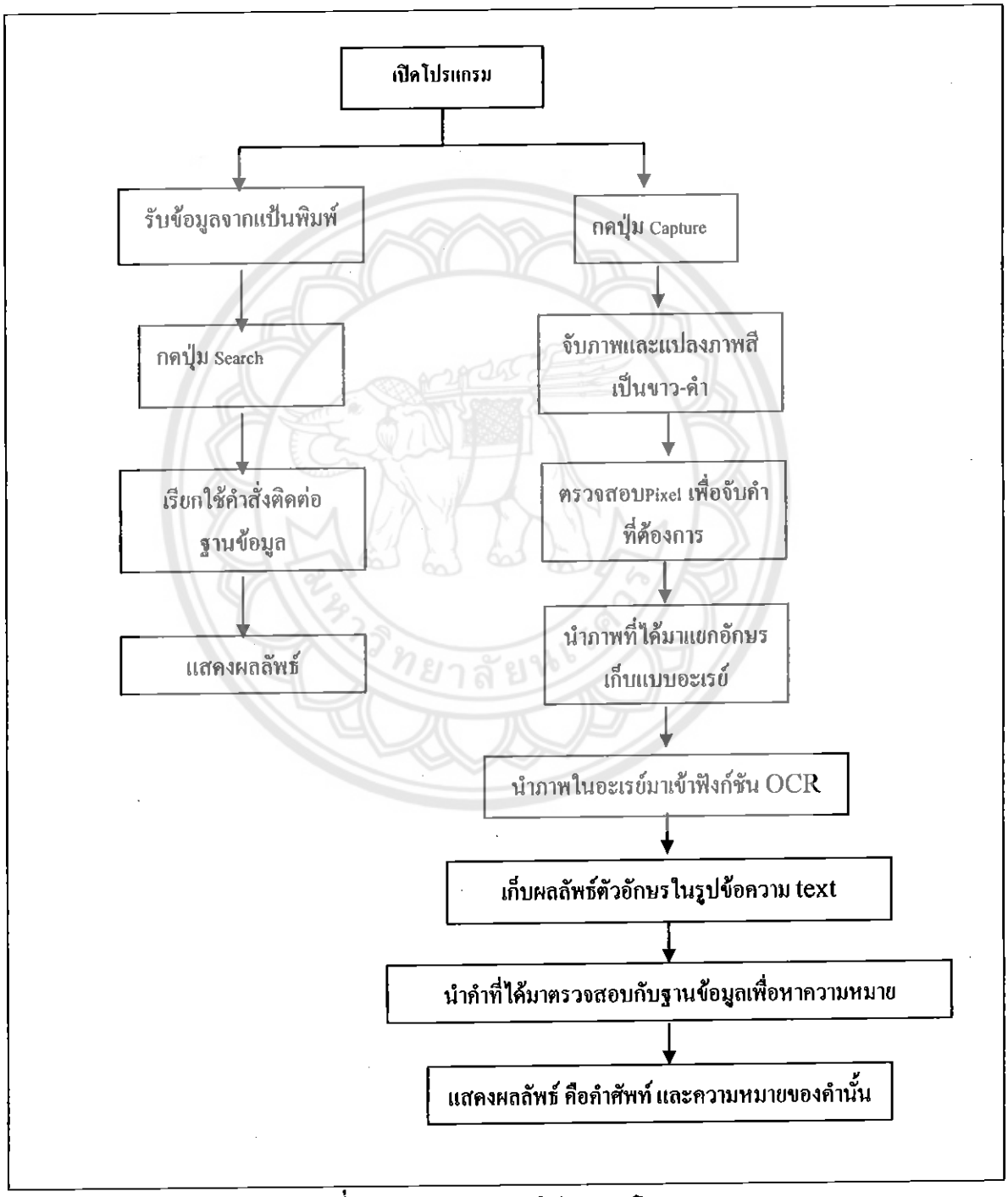
ให้ทำการป้อนคำศัพท์ที่ต้องจนครบ แล้วทำการ บันทึก ข้อมูล ก็จะได้ ไฟล์ฐานข้อมูล

2.3 หลักการของ OCR เบื้องต้น

OCR (Optical Character Recognition) เป็นเทคโนโลยีที่มีการศึกษามานานแล้ว แต่ยังไม่แพร่หลายในปัจจุบันเท่าไรนัก OCR นั้นเป็นกระบวนการรู้จำรู้จัก ตัวอักษร สำหรับอัลกอริทึมหรือโครงสร้างของตัวโปรแกรมก็แตกต่างกันออกไป ในโครงงานี้ไม่ได้เน้นในเรื่องของการสร้าง OCR ออกมาแต่ เราได้มีการนำเอา OCR จากแหล่งที่มาอื่นๆมาใช้เป็น Open Source ยังไม่ได้เป็นอัลกอริทึมที่สมบูรณ์อาจจะยังมีข้อผิดพลาดอยู่บ้างในกระบวนการรู้จำรู้จักตัวอักษร หลักการของ OCR นั้นจะครอบคลุมเนื้อหาทางด้านการประมวลผลภาพดิจิตอล (Digital Image Processing) มีการนำเอาทฤษฎีโครงข่ายประสาทเทียม (Neural Network) มาใช้ร่วมกับกระบวนการรู้จำรู้จัก เมื่อโปรแกรม OCR รับภาพ เข้ามาเพื่อทำการประมวลผล ในส่วนนี้จะมีการแปลงภาพที่ได้ให้เป็น gray scale มีการใช้ทฤษฎี threshold และการทำ segmentation ทำ การ Normalize ภาพเพื่อจะแยกตัวอักษร และ นำเข้าสู่ โครงข่ายประสาทเทียม เพื่อเรียนรู้และกระบวนการจดจำตัวอักษร (recognition Process) และอาศัยหลักการความน่าจะเป็นมาช่วยในการแปลงตัวอักษรออกมา ขั้นสุดท้ายของกระบวนการ OCR คือการเรียงตัวอักษรที่ผ่านกระบวนการข้างต้น เพื่อนำเอาผลลัพธ์ มาแสดงหรือประยุกต์ใช้กับงานด้านอื่น

บทที่ 3
ขั้นตอนการดำเนินงาน

3.1 โครงสร้างของโปรแกรม มีขั้นตอนดังต่อไปนี้



รูปที่ 3.1 แสดงแผนภาพโปรแกรมโดยรวม

จากแผนภาพทำให้เห็นถึงขั้นตอนการรับข้อมูลของโปรแกรมซึ่งมีอยู่ด้วยกัน 2 วิธี คือ การรับข้อมูลทางแป้นพิมพ์โดยการพิมพ์คำที่ต้องการของผู้ใช้ และการรับภาพโดยการนำเมาส์ไปชี้ที่คำที่ต้องการ

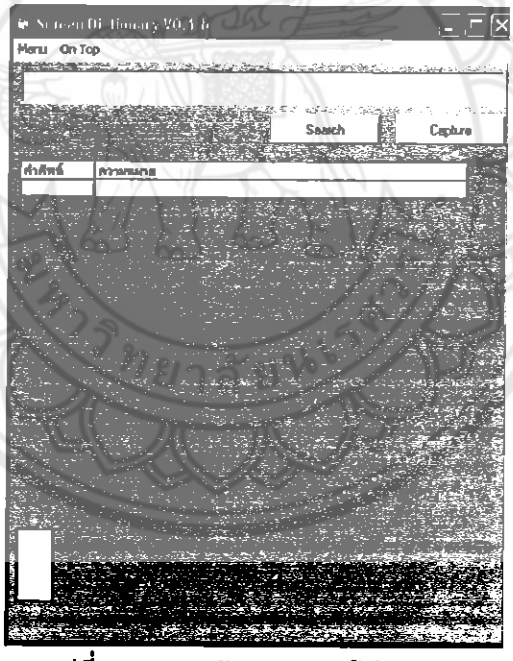
3.2 กระบวนการสร้างโครงงาน

3.2.1 การสร้างที่ติดต่อกับผู้ใช้ (User Interface)

ซึ่งมีองค์ประกอบดังต่อไปนี้

1. เมนูบาร์ (Menu Bar)
2. กล่องรับและแสดงข้อมูล (Textbox)
3. ปุ่มคำสั่ง Search และ Capture
4. ตารางแสดงคำศัพท์และความหมาย
5. ช่องแสดงภาพที่ผ่านกระบวนการแยกย่อยแล้ว

ทั้ง 5 องค์ประกอบสามารถนำมารวมกันเพื่อก่อให้เกิดเป็นหน้าของโปรแกรมขึ้นมา ดังนี้



รูปที่ 3.2 แสดงลักษณะของโปรแกรม

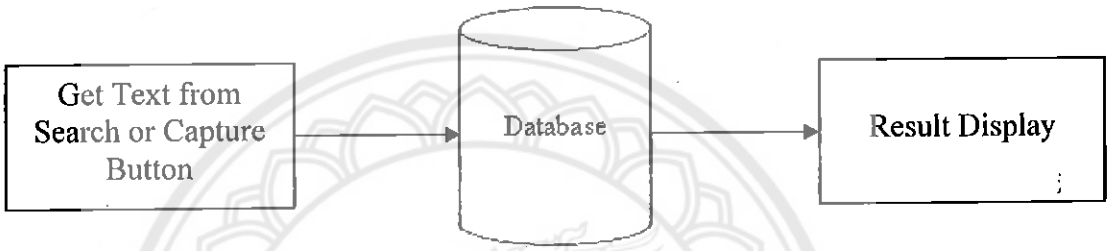
3.2.2 การเขียนโปรแกรมควบคุมการทำงานของโปรแกรม

โดยโปรแกรมนี้ได้แบ่งออกเป็นดังนี้ Form ประกอบด้วย form1 (Screen Grab .frm) ซึ่งมีการเขียนโค้ดของกระบวนการจับภาพหน้าจอคอมพิวเตอร์ การตั้งพิกเซล การแปลงภาพขาว-ดำ และการเรียกใช้ฟังก์ชันจาก form อื่นที่ต้องการ Form2 (Help.frm) เป็นฟอร์มของ help ซึ่งเป็นเครื่องมือที่ช่วยในการแสดงผล และ frmChild (FormChild.frm) เป็นฟอร์มลูก และฟอร์มสุดท้ายคือ MDIForm1

(Main.frm) เป็นฟอร์มหลักของโปรแกรมที่เก็บรวบรวมฟังก์ชันที่สำคัญไว้ เช่นฟังก์ชันของคำสั่งปุ่ม Search และ Capture เป็นต้น Modules เป็นฟอร์มที่เก็บฟังก์ชันที่มีการเรียกใช้ของโปรแกรมบ่อย เช่น ฟังก์เกี่ยวกับการจัดการหน้า Desktop ฟังก์ชันเกี่ยวกับ Device Contxt และฟังก์ชันของ OCR สำหรับโค้ดของโปรแกรมสามารถดูได้จากภาคผนวก ท้ายเล่ม

3.2.3 การสร้างและใช้ฐานข้อมูล

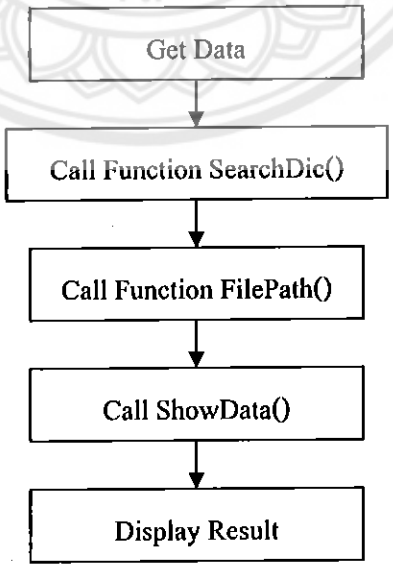
สำหรับฐานข้อมูลมีลักษณะการสร้างดังที่ได้กล่าวมาแล้ว ในบทที่ผ่านมา ซึ่งเป็นนำเครื่องมือของไมโครซอฟต์ (Miceosoft Access) มาใช้งาน มีคำสั่งอยู่หลายคำสั่งที่ต้องมีการติดต่อกับฐานข้อมูล เช่น ปุ่มคำสั่ง Search และ Capture



รูปที่ 3.3 แสดงการติดต่อของฐานข้อมูล

3.3 กระบวนการของปุ่มคำสั่ง Search และ Capture

เป็นกระบวนการที่ผู้ใช้ มีการป้อนข้อมูล (คำศัพท์)ทางแป้นพิมพ์โดยตรง ซึ่งไม่ได้ใช้กระบวนการจับภาพ และกดปุ่ม Search เพื่อต้องการความหมายของคำศัพท์นั้น มีขั้นตอนดังนี้



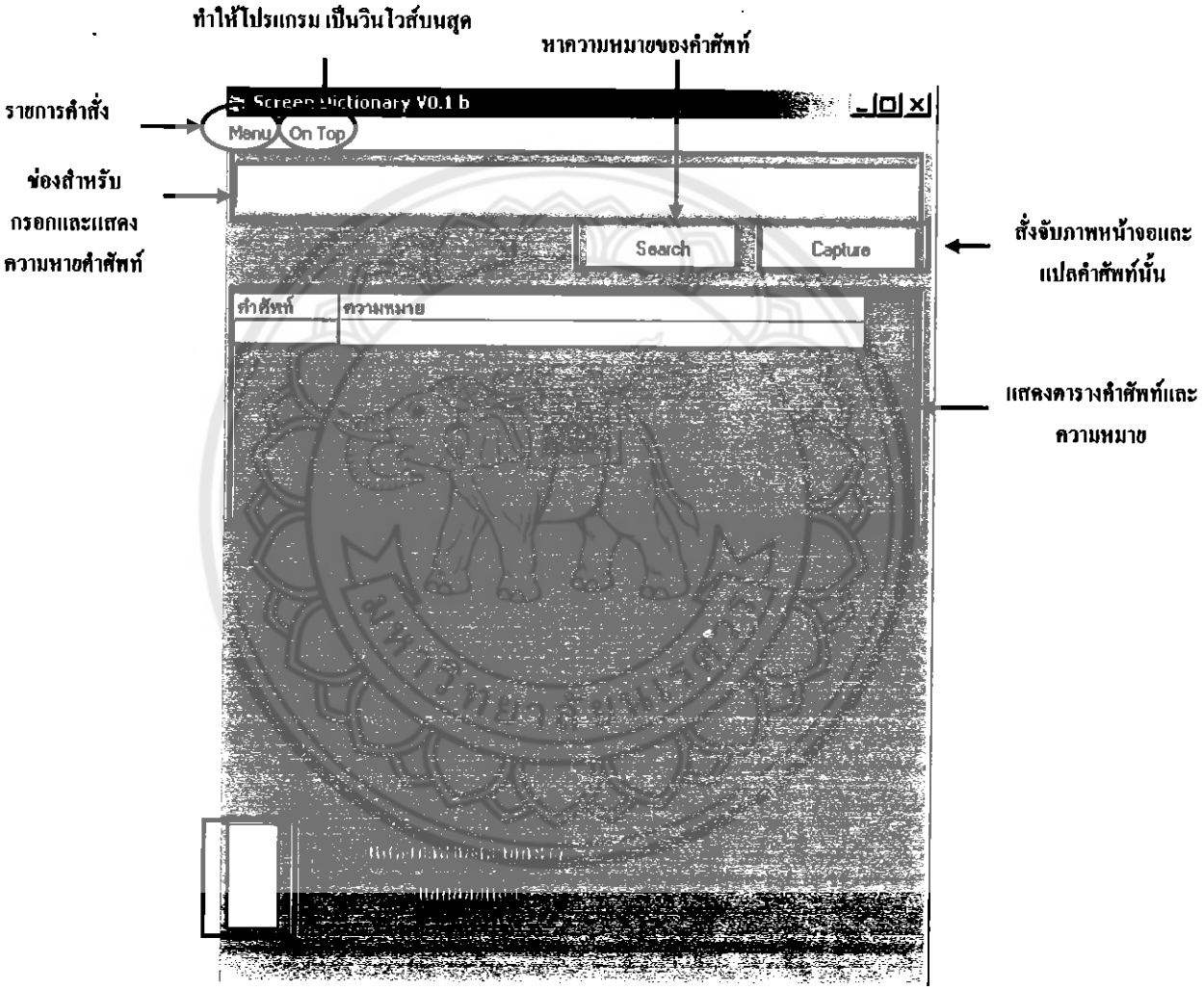
รูปที่ 3.4 แสดงการค้นหาคำศัพท์ด้วยการกดปุ่ม Search

เหตุการณ์จะเกิดเมื่อผู้ใช้ กดปุ่ม Capture ซึ่งจะมีกระบวรการจับภาพหน้าจอคอมพิวเตอร์ และทำการแปลงเป็นตัวอักษร และนำตัวอักษรเหล่านั้นมาค้นหาความหมายผ่านการติดต่อฐานข้อมูล แต่อินพุตที่รับมานั้นไม่ใช่การป้อนทางแป้นพิมพ์ คำศัพท์ ที่ได้นั้นผ่านกระบวนการแปลงภาพเป็นตัวอักษร ดังที่ได้กล่าวมาแล้ว ถึงจะแตกต่างกันทางการนำเข้าของข้อมูล แต่ข้อมูลก็เป็นตัวอักษรเช่นเดียวกัน ซึ่งขั้นตอนการเชื่อมต่อฐานข้อมูลและผลลัพธ์จึงเหมือนกับการกดปุ่ม Search

บทที่ 4

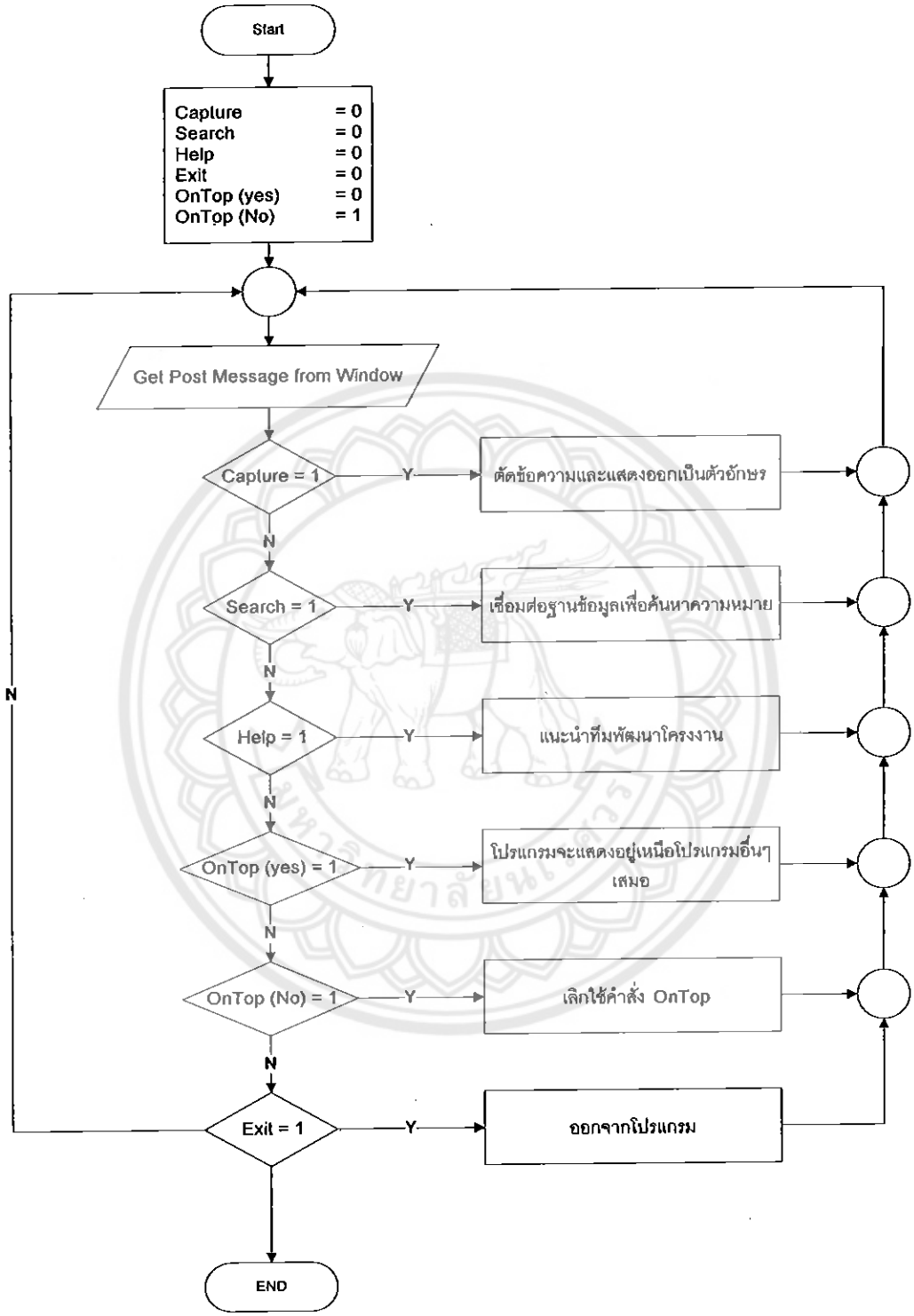
ผลการทดลอง

4.1 ส่วนประกอบของโปรแกรม



รูปที่ 4.1 แสดงลักษณะของโปรแกรมเมื่อทำการเปิดโปรแกรม

4.2 ขั้นตอนการทำงานของโปรแกรม

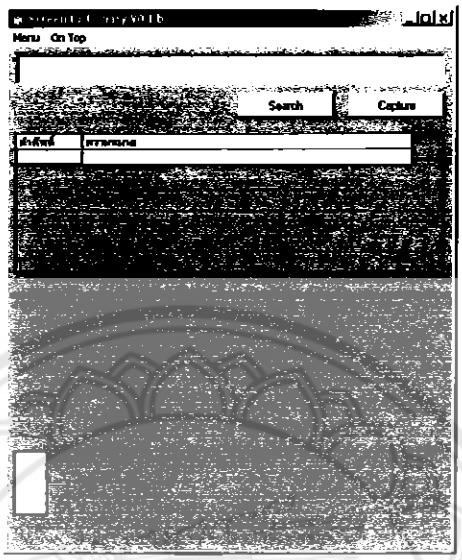


รูปที่ 4.2 แสดงขั้นตอนในการทำงานของโปรแกรม

4.3 ผลการทดสอบโปรแกรม

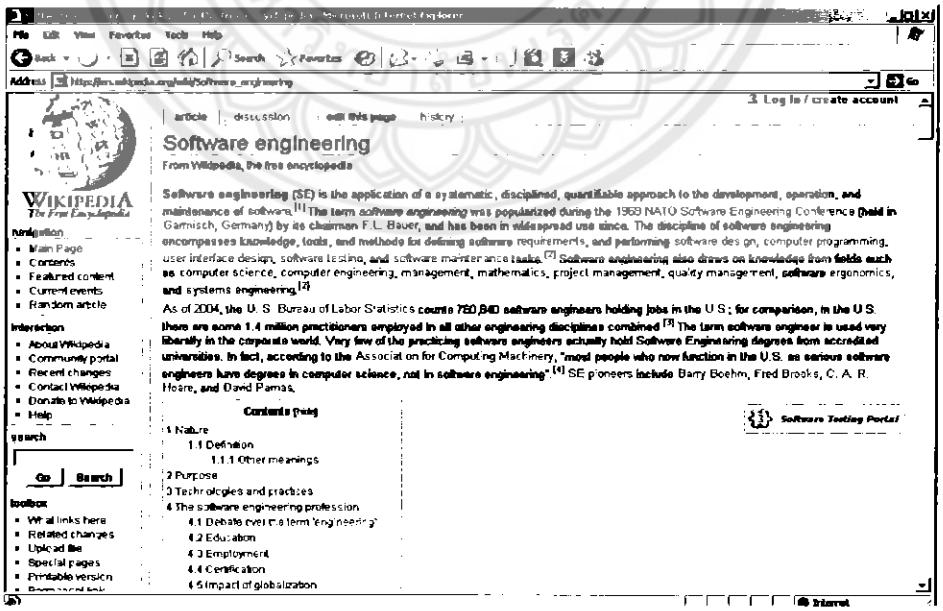
หลังจากการที่ทดลองใช้คำสั่งในโปรแกรมผลที่ได้สามารถอธิบายเป็นขั้นตอนได้ดังนี้

1. เปิดโปรแกรม Screen Dictionary ขึ้นมาเพื่อเตรียมพร้อมสำหรับการเรียกใช้งาน



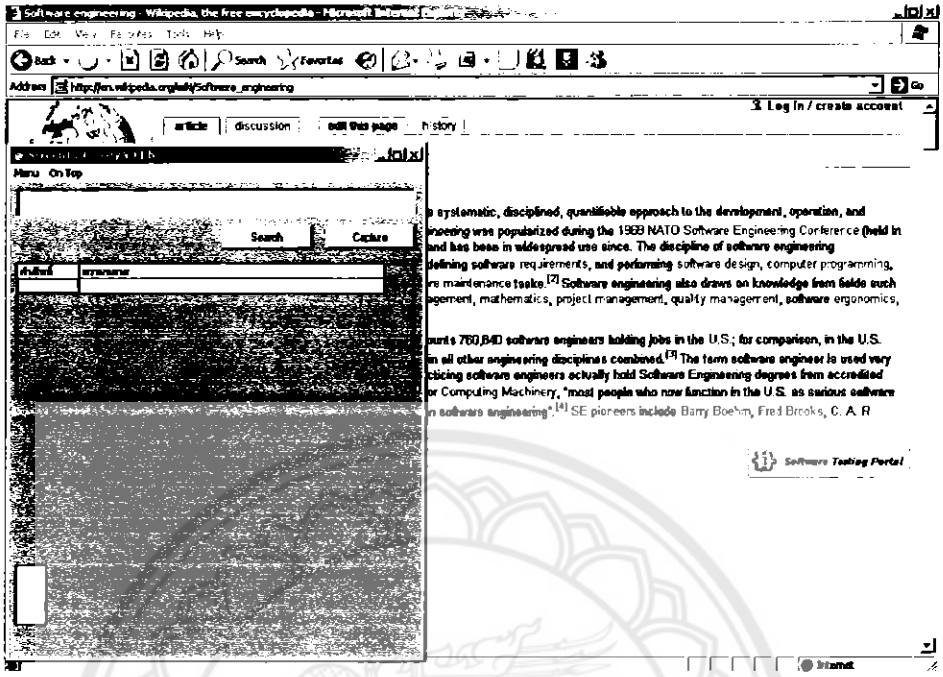
รูปที่ 4.3 แสดงการใช้งานโปรแกรม Screen Dictionary

2. ทำการเปิดโปรแกรม Internet Explorer และbrowser หน้าเว็บเพจขึ้นมา เพื่อทำการทดสอบ



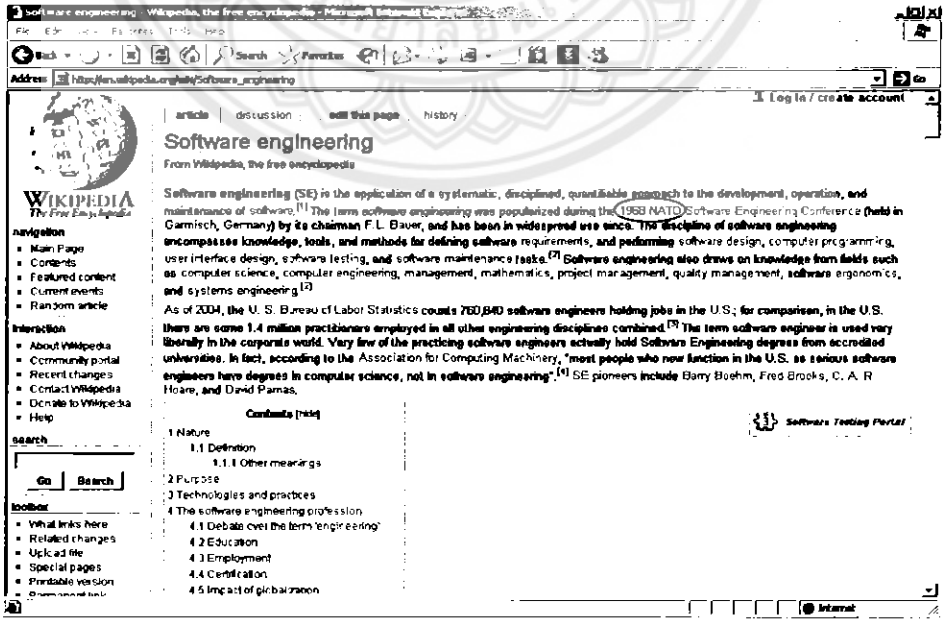
รูปที่ 4.4 แสดงการเรียกใช้หน้าเว็บเพจโดยผ่าน Internet Explorer

3. ผู้ใช้ต้องการทราบความหมายของคำว่า approach และทำการเรียกใช้โปรแกรม ดังนี้



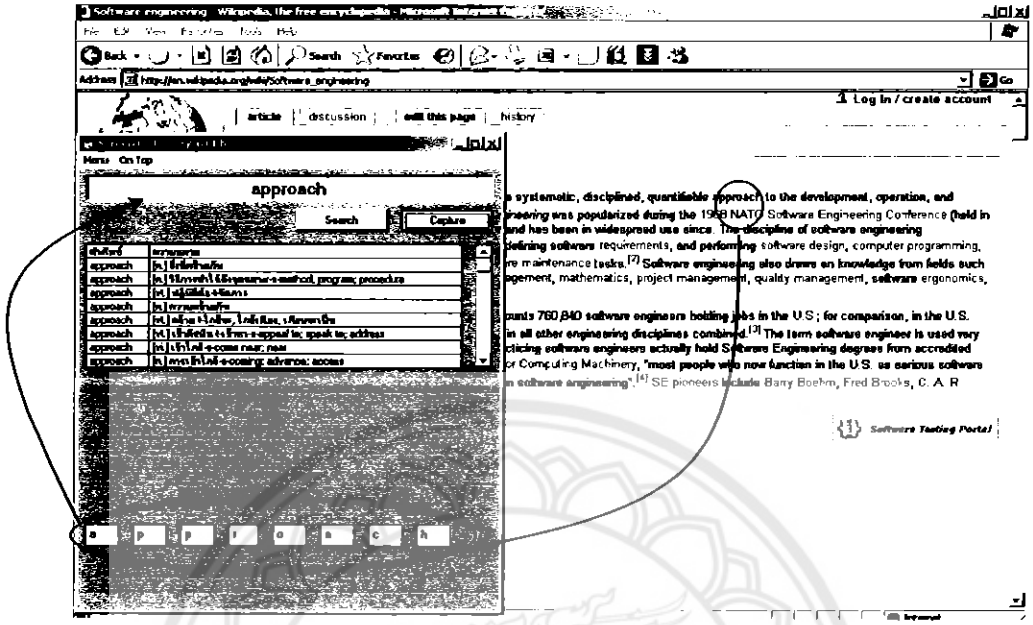
รูปที่ 4.5 แสดงการเรียกใช้โปรแกรม Screen Dictionary และ Internet Explorer

4. ทำการกดปุ่ม Capture หลังจากนั้นให้นำเมาส์ไปชี้คำที่ต้องจะให้โปรแกรมแปล ในที่นี้ จะชี้ที่คำว่า approach



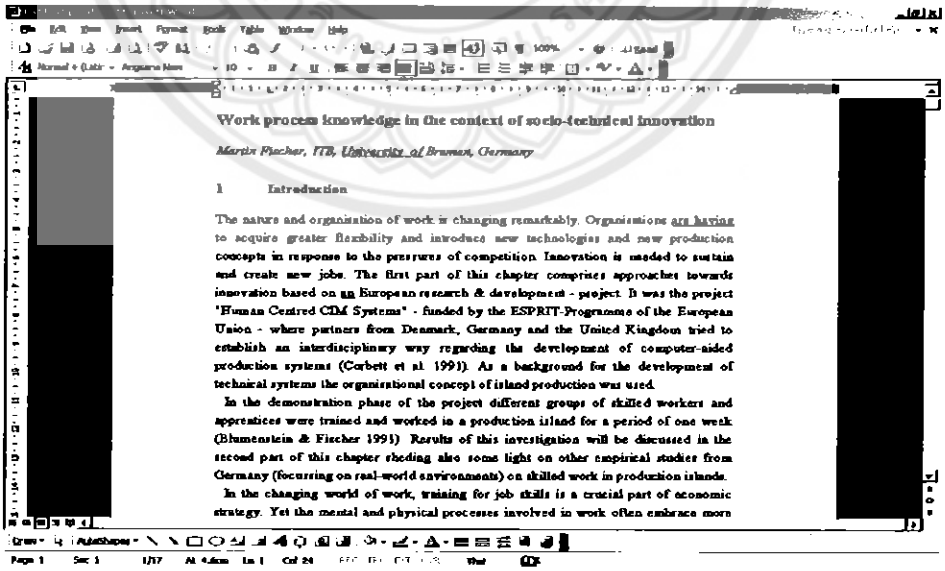
รูปที่ 4.6 แสดงตำแหน่งการชี้ของเมาส์

5. ผลที่ได้จากการทดสอบโปรแกรม เป็นดังนี้

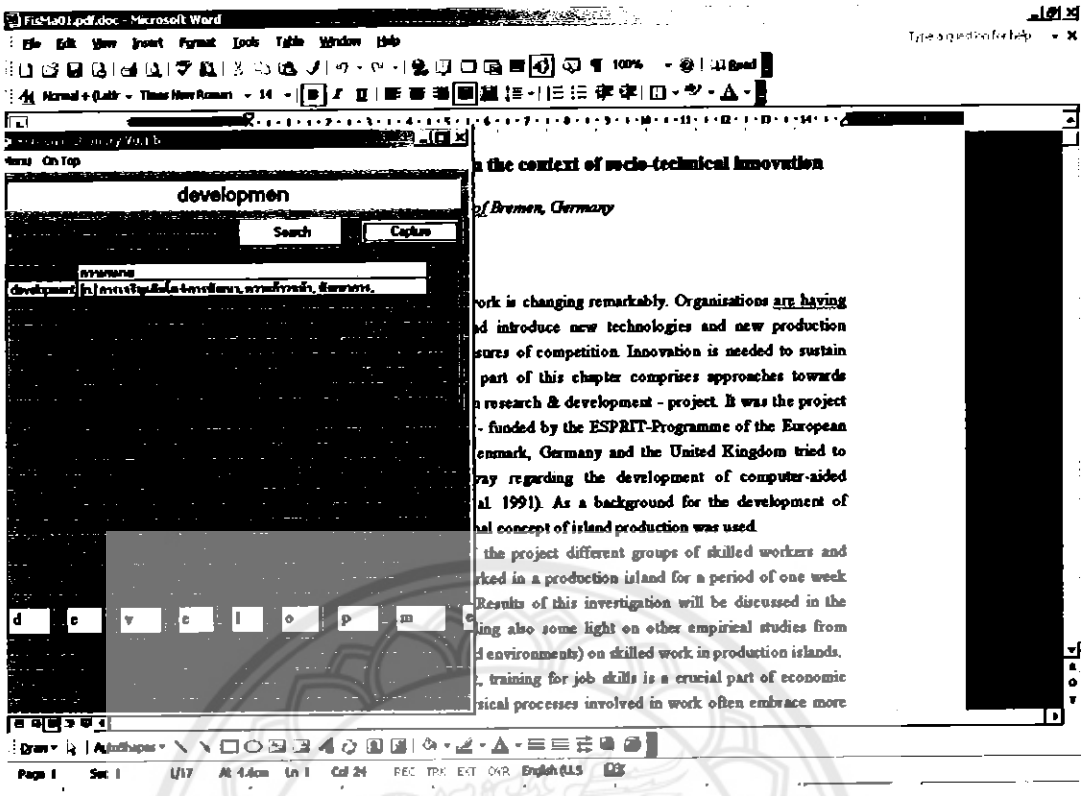


รูปที่ 4.7 แสดงผลลัพธ์จากการจับคำและการแปลความหมาย

6. การทดสอบโปรแกรมกับเอกสาร Microsoft Word มีดังนี้

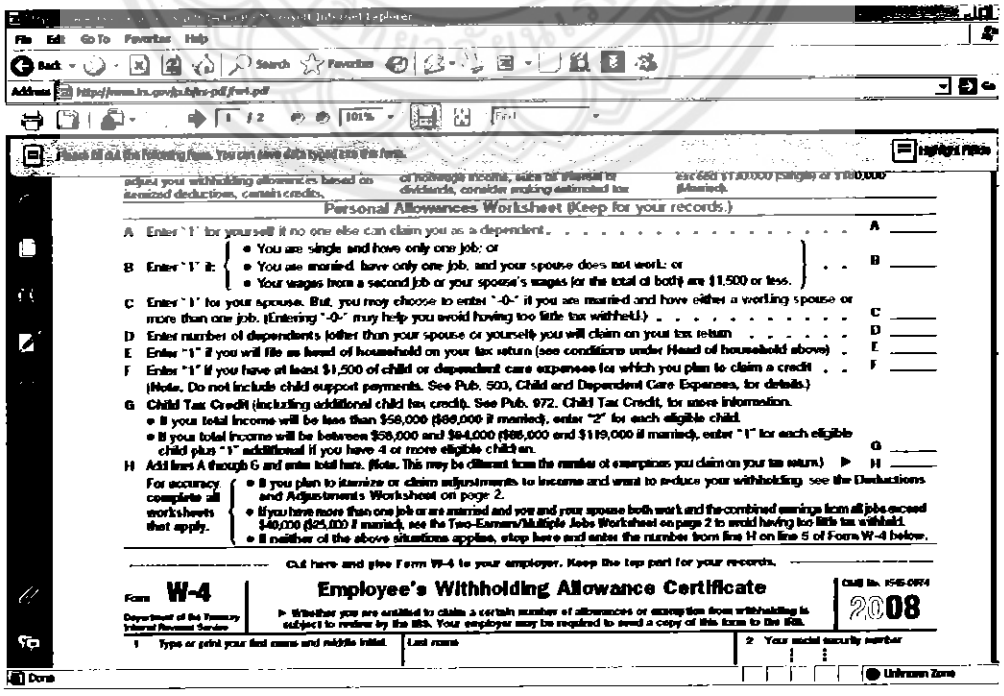


รูปที่ 4.8 แสดงการเปิดเอกสาร Microsoft Word

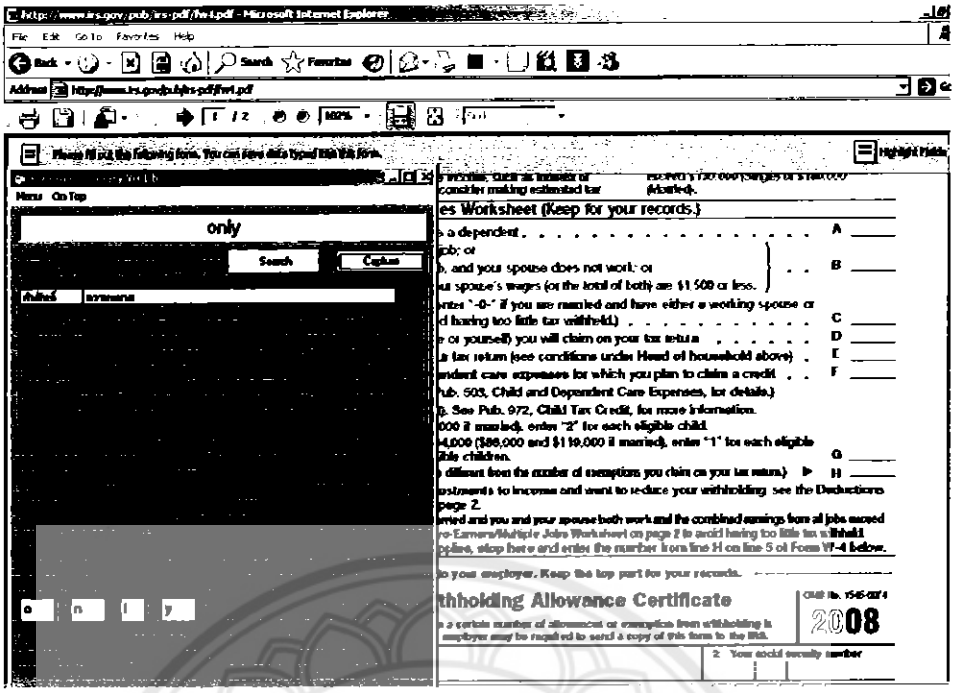


รูปที่ 4.9 แสดงผลลัพธ์โปรแกรมกับเอกสาร Microsoft Word

7. การทดสอบโปรแกรมกับเอกสารประเภท Adobe Acrobat PDF มีดังนี้

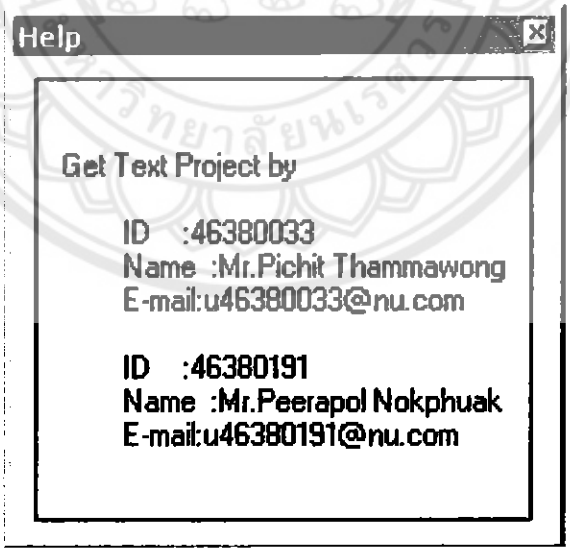


รูปที่ 4.10 แสดงการใช้งานเอกสารประเภท PDF



รูปที่ 4.11 แสดงผลลัพธ์ของการใช้งานกับเอกสาร PDF

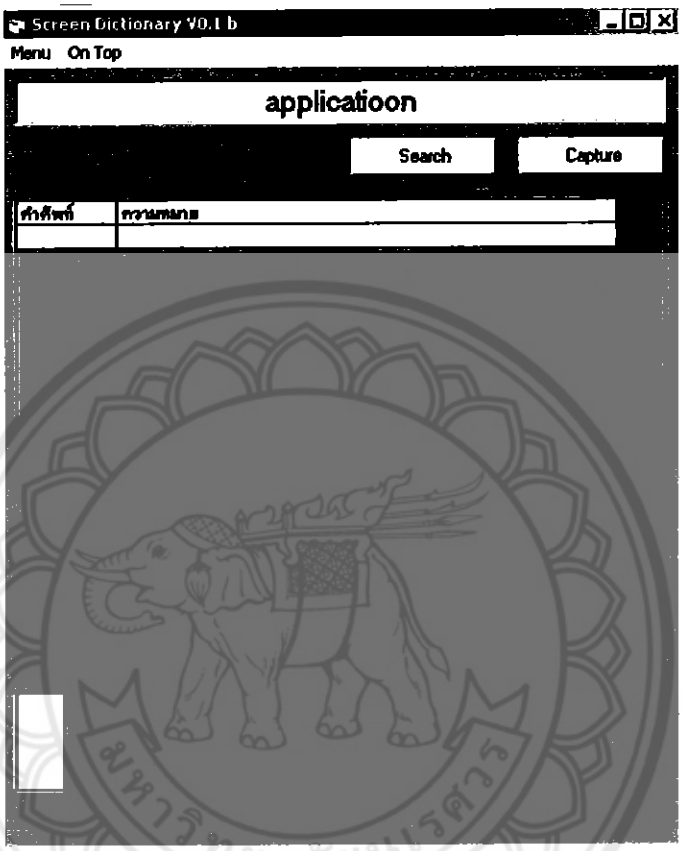
6. คำสั่งอื่นๆบนโปรแกรม เช่น คำสั่ง Help และคำสั่ง On Top



รูปที่ 4.12 แสดงการเรียกใช้คำสั่ง Help

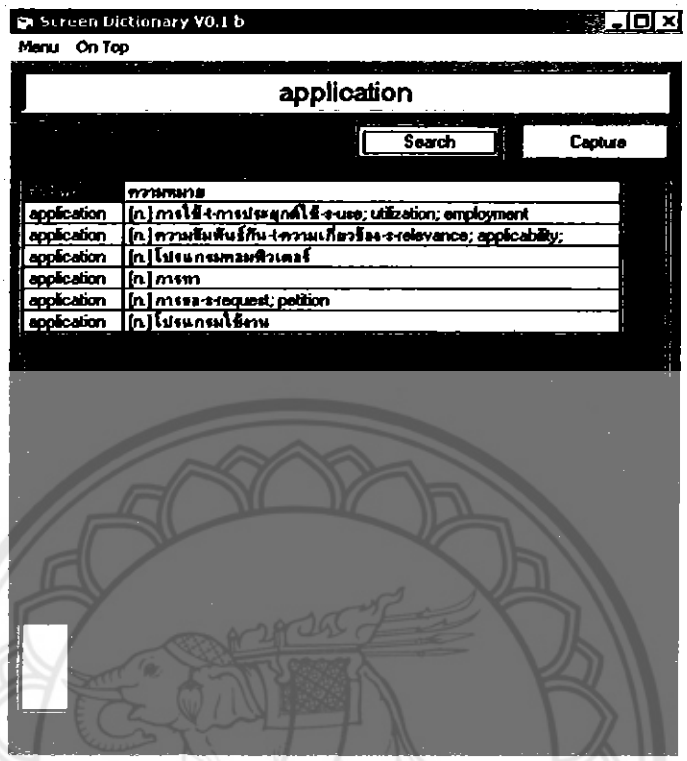
ส่วนคำสั่ง On Top นั้น เมื่อเราเลือกที่ yes โปรแกรมของเรา จะปรากฏอยู่บนสุดเสมอ ถ้าเลือก no โปรแกรมก็จะคืนค่ากลับเหมือนเดิม

7. การใช้งานปุ่ม Search ปุ่มนี้มีไว้สำหรับการรับคำศัพท์ผ่านแป้นพิมพ์ เมื่อป้อนคำศัพท์ที่ช่อง Textbox แล้ว ผู้ใช้สามารถกดปุ่มนี้เพื่อดูความหมายของคำศัพท์นั้นได้ จะใช้ในกรณีที่ไม่มีการับภาพที่หน้าจอคอมพิวเตอร์ มีวิธีดังนี้



รูปที่ 4.13 การป้อนคำศัพท์ทางแป้นพิมพ์

ผลที่ได้จากโปรแกรม Screen Dictionary ดังนี้



รูปที่ 4.14 แสดงการแปลความหมายของโปรแกรม

จากการทดสอบกับฟอนต์มาตรฐาน3 ชนิดคือ Angsana UPC, MS Sans Serif, Tahoma ที่ขนาด 16 pt และ18 pt กับเอกสารต่อไปนี้ เอกสาร HTML, MS Word และ PDF ตามลำดับ จำนวนคำที่ทดสอบ 50คำ ผลที่ได้ดังต่อไปนี้

ตารางที่ 4.1 เปรียบเทียบประสิทธิภาพของการทำงานของโปรแกรม

ชนิด	ขนาด (pt)	จำนวนอักขระ 50 คำ		
		Angsana (คำ)	MS San Serif (คำ)	Tahoma (คำ)
MS Word	16	20	42	45
	18	25	45	47
PDF	16	19	41	43
	18	25	44	45
HTML	16	19	35	34
	18	22	39	36

บทที่ 5

บทสรุป

5.1 สรุปและอภิปรายผลการทดลอง

จากผลการทดลองโครงงานที่ได้นั้น โครงงานนี้แสดงถึงการจับภาพบนหน้าจอคอมพิวเตอร์ พร้อมทั้งการกำหนดขอบเขตของภาพนั้น เพื่อให้ได้มาซึ่งภาพหรือจุดที่เราต้องนั่นก็หมายถึงคำในภาพนั้น และนำภาพนั้นมาเข้าสู่กระบวนการประมวลผลภาพ เพราะบางครั้งภาพที่หน้าจอคอมพิวเตอร์หรือคำที่เราต้องการไม่ได้มีเพียงแค่สีดำเพียงสีเดียว แต่อาจจะปะปนสีอื่นอีกมากมาย โครงงานนี้ได้ใช้การแปลงภาพสีให้กลายเป็นภาพขาว- ดำ เพื่อที่จะได้นำผลลัพธ์ไปสู่กระบวนการวิเคราะห์ตัวอักษรที่ง่ายขึ้น แต่โครงงานนี้ก็มีข้อจำกัดที่ไม่สามารถแปลงภาพตัวอักษรให้ออก มาเป็นตัวอักษร กับภาพตัวอักษรจำพวกที่เป็นตัวอักษรสีขาวกับภาพตัวอักษรที่มีการขีดเส้นใต้คำๆนั้น เพราะในส่วนนี้ยังต้องลงรายละเอียดเกี่ยวกับการซ้อนทับหรือติดกันของพิกเซลอีก ในสำหรับการทดลองที่ได้ปฏิบัตินั้น ได้ทดสอบงานเอกสารเป็นจำนวน 3 ชนิด คือ เอกสาร HTML บนหน้าเว็บเพ็จ เอกสาร PDF และเอกสาร Word จากเอกสารทั้ง 3 ชนิดนั้นจะเห็นว่าเอกสาร Word กับ PDF นั้น ผลลัพธ์ค่อนข้างดี ส่วนเอกสาร HTML นั้นยังมีข้อผิดพลาดมากมาย เพราะในเอกสารชนิดนี้ มีคำที่เป็นลิงค์ ถ้าเรานำเมาส์ไปชี้ที่คำๆ นี้จะเกิดการขีดเส้นใต้ และก็จะส่งผลต่อข้อจำกัดของเรา โครงงานนี้ สามารถค้นหาความหมายของคำศัพท์ได้ 2 วิธี คือ การกดทางปุ่ม Search และการกดปุ่ม Capture สำหรับปุ่ม Search เป็นการทำงานที่คล้ายคลึงกับโปรแกรมแปลคำศัพท์โดยทั่วๆไป ที่มีการรับข้อมูลคำศัพท์ทางแป้นพิมพ์และแสดงผลออกมาทางโปรแกรม และสำหรับปุ่ม Capture นั้นเป็นการนำเอากระบวนการจับจอภาพคอมพิวเตอร์มาช่วยลดระยะเวลาในการพิมพ์ข้อมูลลงไป อีกทั้งพบว่าง่ายต่อการใช้งานอีกด้วย

5.2 ปัญหาและแนวทางในการพัฒนา

โครงงานนี้มีแนวคิดหลัก และเป็นที่มาของการสร้างสรรค์งานนี้ขึ้นมา คือ การลดระยะเวลาในการพิมพ์คำศัพท์ และง่ายต่อการใช้งานโปรแกรมที่ใช้งานนั้นเป็นอิสระในตัวของมันเอง ไม่ต้องนำงานหรือข้อมูลยัดใส่ลงไปในโปรแกรมเพื่อให้เสียเวลา แต่โครงงานนี้ก็มีข้อจำกัดของขอบข่ายงานดังนี้

5.2.1 ชนิด ขนาดและลักษณะของตัวอักษร

จากผลของการทดลอง ได้ทำการทดลองกับกลุ่มอักษรมาตรฐาน 3 ชนิดด้วยกัน คือ อักษรชนิด Angsana UPC, MS Sans Serif และTahoma และขนาดตัวอักษรต่างกัน ทั้งหมด 3 ขนาด คือ 16 pt 18 pt และ 24 pt ผลที่ได้คือ ฟอนต์ตัวอักษร Angsana UPC มีปัญหาเกี่ยวกับการตัดอักษรมาเป็นอักษรย่อยเป็น

อย่างมาก สาเหตุ เพราะรูปแบบของฟอนต์นี้มีความเหลื่อมล้ำของตัวอักษรที่คร่อมกันอยู่ ยกตัวอย่างเช่น roof จะเห็นว่าตัวอักษร r และตัวอักษร o นั้นคร่อมกันอยู่ซึ่งมีผลต่ออัลกอริทึมการแปลงตัวอักษรของโครงงานนี้ ส่วนฟอนต์ที่เหลือ คือ MS Sans Serif และ Tahoma นั้น แทบไม่พบข้อบกพร่องเลย

5.2.2 อักษรที่ขีดเส้นใต้และอักษรสีขาว

อักษรที่ขีดเส้นใต้นั้นทำให้โครงงานนี้ประสบปัญหาในด้านการวิเคราะห์หาตำแหน่งของ x,y และการสแกนแยกอักษรออกจากกัน โครงงานนี้จะตรวจจับอักษรที่เป็นสีดำเท่านั้น ในทางกลับกันการที่จะทราบถึงขอบเขตของคำหรือการเว้นวรรคนั้น สามารถทำได้โดยการตรวจจับพิกเซลสีขาว ถ้าหากมีภาพอักษรที่เป็นสีขาว โปรแกรมก็จะเกิดการผิดพลาดออกมาทันที

5.2.3 ความผิดพลาดของโอซีอาร์ (OCR)

เนื่องด้วย OCR เป็นโค้ดที่เราได้มาจากแหล่งที่อื่น ไม่สามารถแก้ไขได้ ตัวอย่างของความผิดพลาดที่เห็นได้ คือ การแปลงตัวอักษร l, a, s, g เมื่อแปลงแล้วจะได้ดังนี้ i, 8, 5 และ9 ตามลำดับ แนวทางการแก้ไข คือ การสร้าง OCR ขึ้นมาใหม่ที่สามารถแก้ไขและเพิ่มเติมได้

จากปัญหาทั้ง 3 ข้อข้างต้น ดังที่ได้กล่าวมานั้น พบว่าข้อผิดพลาดของกระบวนการจะเกิดขึ้นที่ขั้นตอนการแปลงตัวอักษร เพราะอัลกอริทึมจะจำกัดในด้านของพิกเซล ไม่มีความยืดหยุ่นในเรื่องของพิกเซลเท่าที่ควร แนวทางการพัฒนาปรับปรุงนั้น สามารถนำทฤฎีหรืออัลกอริทึมในเรื่องต่อไปนี้ การแยกบรรทัดตัวอักษร (Character Segmentation) อาจจะใช้ความรู้ทางด้าน horizon projection profile method และ vertical projection profile method ซึ่งเป็นการแยกข้อความออกจากบรรทัด ซึ่งสามารถนำมาใช้พัฒนาร่วมกันได้ วิธีการดังกล่าวจะได้บรรทัดอักษรที่แม่นยำขึ้น และการแปลงตัวอักษร หรือตัดตัวอักษรซึ่งเป็นกระบวนที่สำคัญที่สุดเพื่อคัดแยกตัวอักษร และวิธีการตัดคำ (Character extraction) และวิธี Bob's Coloring จะหา pixel รอบๆ coordinate ปัจจุบัน โดยไล่หาทีละทิศทั้งหมด 8 ทิศ และหลักการ text detection สามารถนำวิธีการอีกทั้งอัลกอริทึมเหล่านี้มาประยุกต์ใช้เพื่อแก้ไขโครงงานนี้ ให้มีประสิทธิภาพมากยิ่งขึ้น

เอกสารอ้างอิง

[1] GREGORY A. BAXES. **Digital Image Processing: Principle and Application.** John Wiley & Sons, Inc. 1994

[2] M.A. SID – AHMED. **Image Processing Theory, Algorithms and Architectures.** McGraw – Hill Co., 1995

[3] Appleman, Dan. **Dan appleman's Visual Basic 5.0 programmer's guide to the Win 32 API / Dan Appleman.** Emeryville, Calif.: Ziff-Davis Press, 1997

[4] **Optical Character Recognition.** http://wara.com. 2007

[5] **ฉัททวุฒิ** พืชผล, พิชิต สันติกุลานนท์. **คู่มือเรียน Visual Basic 6.** กรุงเทพฯ: **บริษัท** โปรวิชั่น จำกัด, 2542

[6] **Visual Basic 6.0 Win32 API เทคนิคและการประยุกต์.** นนทบุรี: **บริษัท อัปเปอร์** แมเนจเมนท์ เอ็กซ์เซลเลนซ์ จำกัด. 2547

[7] **พีรภัทร์** สว่างเพียร. **เทคนิคการเขียนโปรแกรมและเกมด้วย Visual C++.** กรุงเทพฯ: **ซีเอ็ดยูเคชั่น**, 2545

[8] **สิทธิโชค** ยอดระยับ. **การเขียนโปรแกรม Digital Image Processing ด้วย Visual Basic.** **พิมพ์ครั้งที่** 1. กรุงเทพมหานคร. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2550.

ภาคผนวก ก

ฟังก์ชัน API (Application Programming Interface) ใน Visual Basic 6.0 มีดังต่อไปนี้

GetPixel Function

Getpixel เป็นฟังก์ชันเกี่ยวกับการเก็บค่าพิกเซล รูปแบบดังต่อไปนี้

```
Public Declare Function GetPixel Lib "gdi32"
        (ByVal hdc As Long, ByVal x As Long,
         ByVal y As Long) As Long
```

CreateCompatibleBitmap Function

ฟังก์ชัน CreateCompatibleBitmap ที่สร้างภาพ bitmap ที่เข้ากันระหว่างภาพที่หน้าจอกับภาพของ Device Context มีรูปแบบดังต่อไปนี้

```
Declare Function CreateCompatibleBitmap Lib "gdi32" Alias "CreateCompatibleBitmap"
        (ByVal hdc As Long,
         ByVal nWidth As Long,
         ByVal nHeight As Long) As Long
```

BitBlt Function

ฟังก์ชัน BitBlt จะกระทำการถ่ายทอดบิต-บล็อก ของข้อมูลสีตอบสนองกับสี่เหลี่ยมของพิกเซลจาก DC แหล่งกำเนิดไปสู่ DC เป้าหมาย มีรูปแบบดังนี้

```
Declare Function BitBlt Lib "gdi32" Alias "BitBlt"
        (ByVal hDestDC As Long,
         ByVal x As Long,
         ByVal y As Long,
         ByVal nWidth As Long,
         ByVal nHeight As Long,
         ByVal hSrcDC As Long,
         ByVal xSrc As Long,
         ByVal ySrc As Long,
         ByVal dwRop As Long) As Long
```

ซึ่ง hdcDest – เป็น DC เป้าหมาย

nXDest, nYDest - เป็นคู่อันดับของแกน x y ของมุมบนซ้ายของสี่เหลี่ยมปลายทาง

nWidth, nHeight - ขนาดความกว้างและสูงของสี่เหลี่ยมปลายทาง

hdcSrc - DC ต้นกำเนิด

nXSrc, nYSrc – คู่อันดับของแกน x y ของมุมซ้ายบนของสี่เหลี่ยนต้นกำเนิด

CreateDC Function

ฟังก์ชัน CreateDc จะสร้าง device context ขึ้นมาซึ่งจะอ้างอิงวัตถุ เมื่อสิ้นสุดการใช้ DC ในโปรแกรม เราจะใช้ฟังก์ชัน DeleteDC เพื่อทำลายมันเสีย จะไม่ใช้ฟังก์ชัน ReleaeDC กับ DC ที่ถูกสร้างในโปรแกรม ฟังก์ชันจะมีการคืนค่ากลับ ถ้าโปรแกรมทำงานสำเร็จ แต่จะคืนค่า 0 ถ้าเกิดความผิดพลาดเกิดขึ้น ซึ่งมีรูแบบดังต่อไปนี้

```
Declare Function CreateDC Lib "gdi32.dll" Alias "CreateDCA"
        (ByVal lpDriverName As String,
         ByVal lpDeviceName As String,
         ByVal lpOutput As String,
         lpInitData As DEVMODE) As Long
```

ซึ่ง IpDriverName – โดยทั่วไปแล้วจะไม่มีค่า คาดว่าใช้สำหรับ DISPLAY เพื่ออ้างอิงไดรว์เวอร์ปสดงผลภายใต้ Windows NT

IpDeviceName – ชื่อของ device ที่จะสร้างเป็น DC

IpInitData – การเก็บรวบรวมข้อมูลเพื่อใช้ initialize DC นั้น

CallWindowProc Function

ฟังก์ชัน CallWindowProc จะส่งข้อมูล message ไปที่โครงสร้างวินโดวส์

มีรูปแบบดังนี้

```
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA"
        (ByVal lpPrevWndFunc As Long,
         ByVal hWnd As Long,
         ByVal Msg As Long,
         ByVal wParam As Long,
         ByVal lParam As Long)As Long
```

ซึ่ง lpPrevWndFunc – เป็นตัวชี้ที่โครงสร้างหรือกระบวนการก่อนหน้า ถ้าค่าที่ได้นี้ถูกเก็บโดยการเรียกฟังก์ชัน GetWindowLong กับตัวพารามิเตอร์ nIndex จะเซตค่าที่ GWL_WNDPROC or DWL_DLGPROC มันคือการกระทำภายในที่อยู่ของวินโดวส์ หรือกระการของข้อความ หรือค่าที่มันได้รับจากที่อยู่นั้น

hWnc - กระบวนการทางวินโดวส์มีไว้เพื่อรับ mesaage จากวินโดวส์

Msg – เป็น message ของวินโดวส์

wParam, Iparam – เป็นตัวแปรที่ใช้ในการเพิ่มข้อมูล message เนื้อหาเกี่ยวกับพารามิเตอร์ตัวนี้จะ

ขึ้นอยู่กับค่าของตัวแปร Msg การคืนค่าผลลัพธ์กลับของกระบวนการ message และขึ้นอยู่กับ message ที่ส่งมา

GetObject Fuction

ฟังก์ชัน GetObject เป็นฟังก์ชันที่เก็บข้อมูลเกี่ยวกับกราฟิกวัตถุ ซึ่งมีรูปแบบดังนี้

```
Declare Function GetObject Lib "gdi32" Alias "GetObjectA"
        (ByVal hObject As Long,
         ByVal nCount As Long,
         lpObject As Any) As Long
```

hgdiobj - เป็นการจับกราฟิกวัตถุที่สนใจ มันสามารถจับได้เพียงอย่างเดียว ดังต่อไปนี้ bitmap, brush, font,palatte,penหรืออุปกรณ์ที่เป็นอิสระจากภาพถูกสร้างขึ้นโดยการเรียกใช้ฟังก์ชัน

CreateDIBSection

cbBuffer – เป็นจำนวนของไบต์ของข้อมูลที่จะถูกเขียนที่บัฟเฟอร์

lpvObject - ชี้ไปที่บัฟเฟอร์ซึ่งเป็นการรับข้อมูลเกี่ยวกับกราฟิกวัตถุ

GetWindowDc Function

ฟังก์ชัน GetWindowDC จะรับค่าจาก Device Context สำหรับการทำวินโดวส์ ประกอบไปด้วย titlebar, menu และ scroll bar ซึ่งวินโดวส์ DC จะทำการวาดสิ่งเหล่านี้ในวินโดวส์ เพราะว่าจุดกำเนิดของ DC คือมุมซ้ายบนของวินโดวส์จะแทนที่ด้วยพื้นที่ลูกข่าย ซึ่งมีรูปแบบดังนี้

```
Declare Function GetWindowDC Lib "user32" Alias "GetWindowDC"
        (ByVal hwnd As Long) As Long
```

ซึ่ง hWnd – เป็นวินโดวส์ที่ Device Context ได้รับมา

GetWindowRect Function

เป็นฟังก์ชันที่อ่านค่าขนาดและตำแหน่งของวินโดวส์ ข้อมูลเหล่านี้จะใส่ไว้ที่ตัวแปรส่งผ่านนั่นคือ lpRect ขนาดที่ได้รับเป็นคู่อันดับของมุมบนซ้าย และมุมล่างขวา ของวินโดวส์ มีรูปแบบดังนี้

```
Declare Function GetWindowRectLib"user32.dll" (ByVal hwnd As Long, lpRect As RECT) As Long
```

ซึ่ง hwnd – เป็นค่า handle ของวินโดวส์ที่อ่านขนาดและความกว้าง

lpRect – ค่าที่รับจากคู่อันดับของมุมซ้ายบนและมุมขวาล่าง

GetDesktopWindow Function

ฟังก์ชันนี้จะคืนค่า handle ให้กับ Desktop Window ซึ่งก็คือวินโดวส์ที่ทำขึ้นจาก Desktop ของคอมพิวเตอร์มันก็คือหน้าจอคอมพิวเตอร์ แต่ถ้าฟังก์ชันนี้ล้มเหลวมันจะคืนค่า 0 แทนที่จะเป็นค่าที่ handle ซึ่งมีรูปแบบดังต่อไปนี้

```
Declare Function GetDesktopWindow Lib "user32.dll" () As Long
```

GetCursorPos Function

เป็นฟังก์ชันที่ใช้อ่านค่าตำแหน่งปัจจุบันของเคอร์เซอร์เมาส์ เป็นคู่อันดับ x y ของเมาส์ (เกี่ยวกับหน้าจอคอมพิวเตอร์) เป็นการใส่ค่าลงไปที่ตัวแปรที่ใช้ส่งนั่นคือ IpPoint ฟังก์ชันนี้คืนค่า 0 ถ้าเกิดข้อผิดพลาด หรือ 1 ถ้ามันสำเร็จ มีรูปบบดังนี้

```
Declare Function GetCursorPos Lib "user32.dll" (lpPoint As POINT_TYPE) As Long
```

ซึ่ง IpPoint – เป็นคู่อันดัน x y ของเคอร์เซอร์เมาส์

GetDIBits Function

เป็นฟังก์ชันที่ค่าบิตของภาพและคัดลอกภาพเหล่านั้นไปสู่บัฟเฟอร์ เป็นการใช้รูปแบบเฉพาะ มีดังนี้

```
Declare Function GetDIBits Lib "gdi32" (ByVal aHDC As Long,
        ByVal hBitmap As Long,
        ByVal nStartScan As Long,
        ByVal nNumScans As Long,
        lpBits As Any,
        lpBI As BITMAPINFO,
        ByVal wUsage As Long) As Long
```

hdc – เป็น Device Context

hbmp – ภาพ bitmap

uStartScan - การสแกนแถวแรกเพื่อเก็บค่า

cScanLine – สแกนจำนวนแถว

lpvBits - จุดที่บัฟเฟอร์รับข้อมูลภาพ ถ้าพารามิเตอร์มีค่าเป็น NULL แล้ว ฟังก์ชันจะส่ง ขนาดและรูปแบบ ของภาพไปที่โครงสร้าง BITMAPINFO ชี้โดยพารามิเตอร์ lpbi

ipbi – จะชี้ไปที่ BITMAPINFO ซึ่งต้องการรูปแบบที่มีไว้สำหรับสร้างข้อมูล DIB

uUsage - เป็นรูปแบบของสมาชิก bmiColor ของโครงสร้าง BITMAPINFO

CoppyMemory Function

เป็นการคัดลอกไบต์แบบอนุกรม จากที่หนึ่งในหน่วยความจำคอมพิวเตอร์ไปสู่หน่วยความจำที่อื่นๆ ฟังก์ชันนี้สามารถใช้ส่งข้อมูลระหว่างตัวแปร อะเรย์ โครงสร้างข้อมูล ชนิดใดๆหรือที่อยู่หน่วยความจำนามธรรม คำว่า ByVal จะต้องนำหน้าพารามิเตอร์ ฟังก์ชันจะไม่คืนค่ากลับ มีรูปแบบดังนี้

```
Declare Sub CopyMemory Lib "kernel32.dll"Alias" RtlMoveMemory" (hpvDest As Any,_
        hpvSource As Any,_
        ByVal cbCopy As Long)_
```

hpvDest – ถ้าไม่มีการนำหน้าด้วย ByVal ในตัวแปร หรืออะเรย์ หรือวัตถุ ซึ่งที่อยู่หน่วยความจำที่คัดลอกเป็นไบต์

ภาคผนวก ข

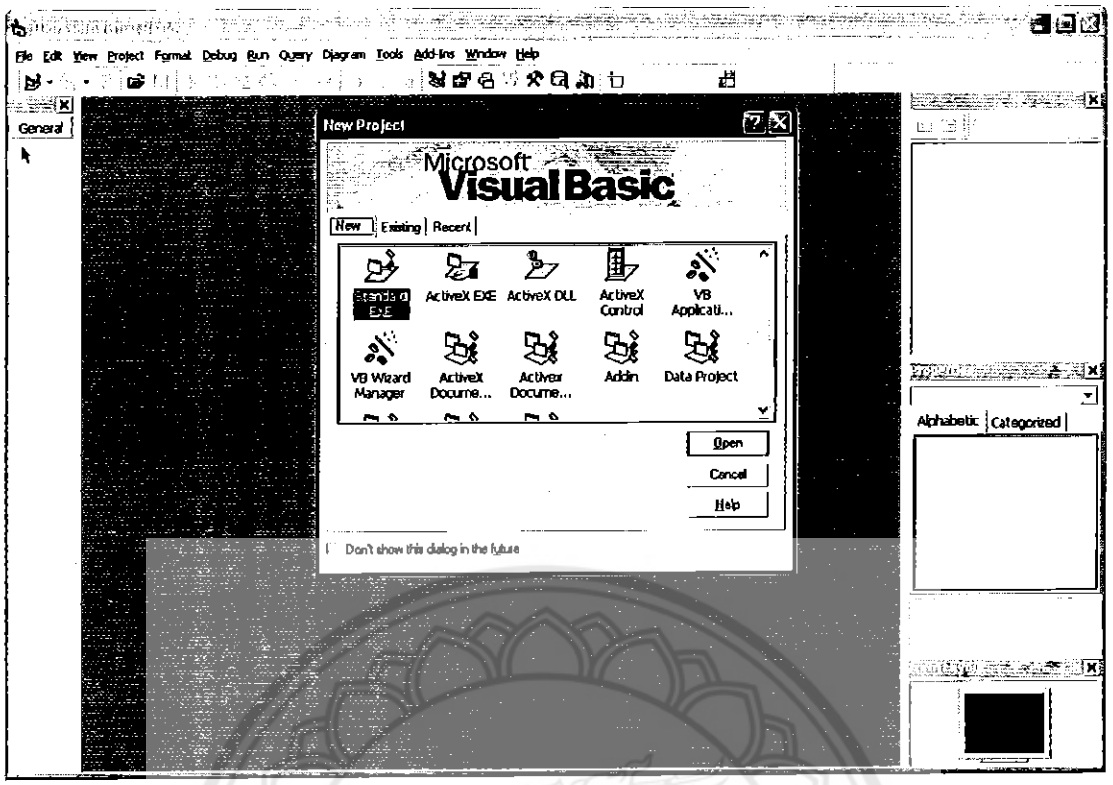
Win32 API คือ API (Application Promgramming Interface) ที่รวบรวมฟังก์ชันที่ใช้พัฒนา Application บน MS Windows ทั้งหมดเอาไว้ การพัฒนา Application โดยติดต่อกับ Win32 API โดยตรงจะสามารถควบคุมการทำงานของ Application ได้ทุกอย่างตามที่ต้องการ

โปรแกรม Screen Dictionary เป็นแกรม Dictionary ที่ทำให้ภาพบนหน้าจอคอมพิวเตอร์กลายเป็นไฟล์รูปภาพแล้วตัดคำในส่วนที่ต้องการมาแปรงเป็นตัวอักษรเพื่อนำไปค้นหาความหมายในฐานข้อมูล Access โปรแกรมนี้ถูกพัฒนามาจากภาษา Visual Basic 6.0 For 32-bit Windows Development เพื่อที่จะสามารถรัน Visual Basic ได้อย่างไม่มีปัญหา ต้องแน่ใจก่อนว่า hardware และ software พร้อมแล้วซึ่งความต้องการของระบบมีดังนี้

- Microsoft Windows 95 หรือมากกว่า, หรือ Microsoft Windows NT Workstation 4.0 (Service Pack 3 recommended) หรือมากกว่า
- 486DX/66 MHz หรือสูงกว่า (แนะนำให้ใช้ Pentium หรือสูงกว่า), หรือชิป Alpha processor ที่สามารถรัน Microsoft Windows NT Workstation ได้
- ไดรฟ์ CD-ROM
- การ์ด VGA หรือสูงกว่า สนับสนุนการแสดงผลระบบ Windows
- แรม 16 MB สำหรับ Windows 95, 32 MB สำหรับ Windows NT Workstation
- เมาส์ และอื่นๆ ที่ระบบ Windows รองรับ

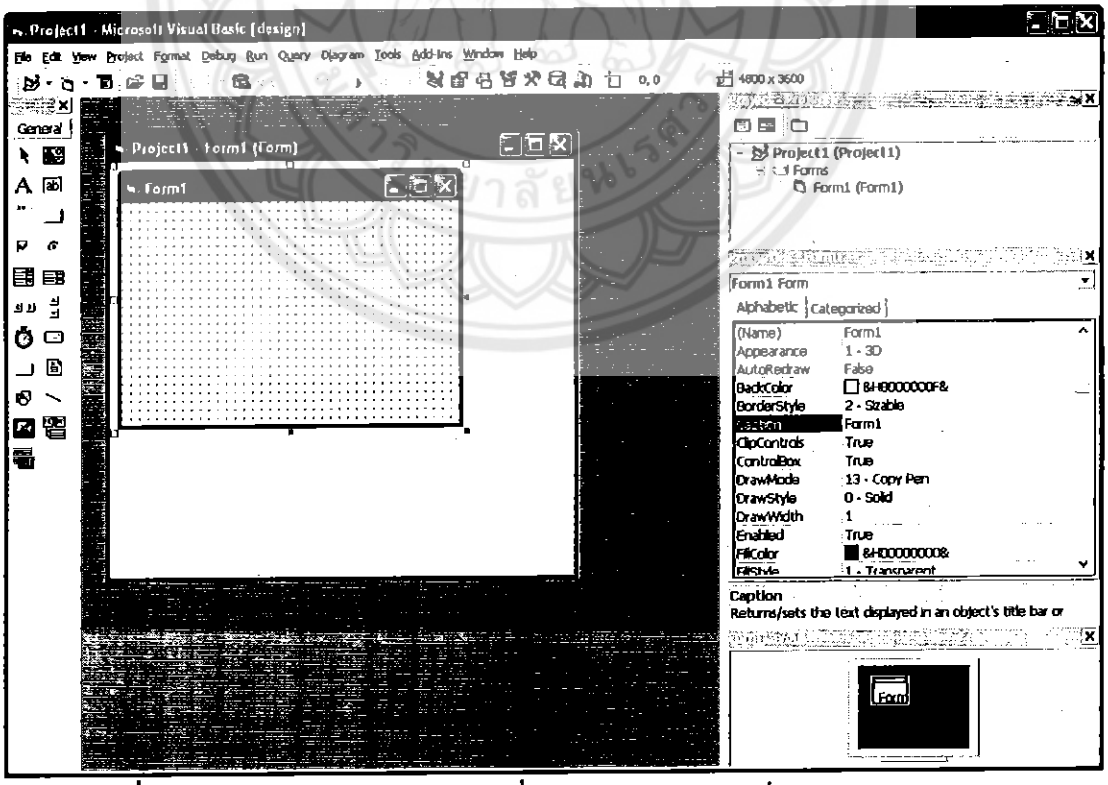
ขั้นตอนในการสร้างโปรแกรม

1. เปิดโปรแกรม Visual Basic 6.0 ได้จาก Start > All Program > Microsoft Visual Studio 6.0 > Microsoft Visual Basic 6.0 หรือ C:\Program Files\Microsoft Visual Studio\VB98\ VB6.exe



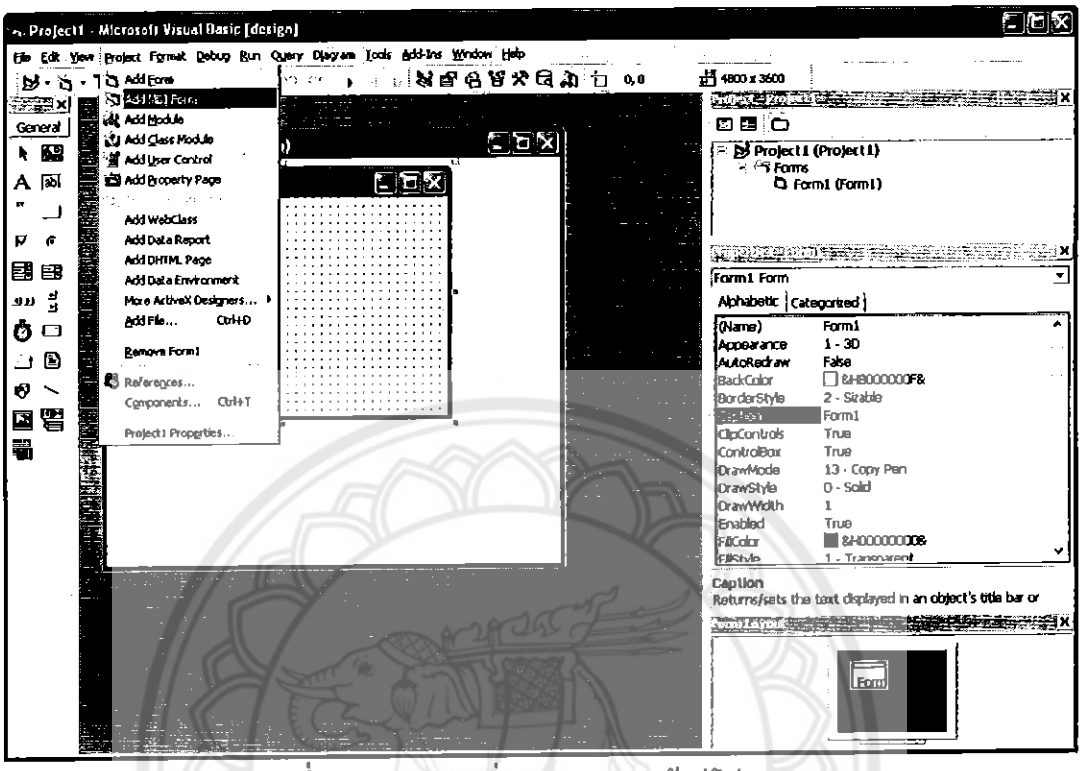
รูปที่ ข.1 แสดงหน้าต่างของโปรแกรม Visual Basic เวอร์ชัน 6

2. เลือกแอพพลิเคชันชนิด Standard EXE



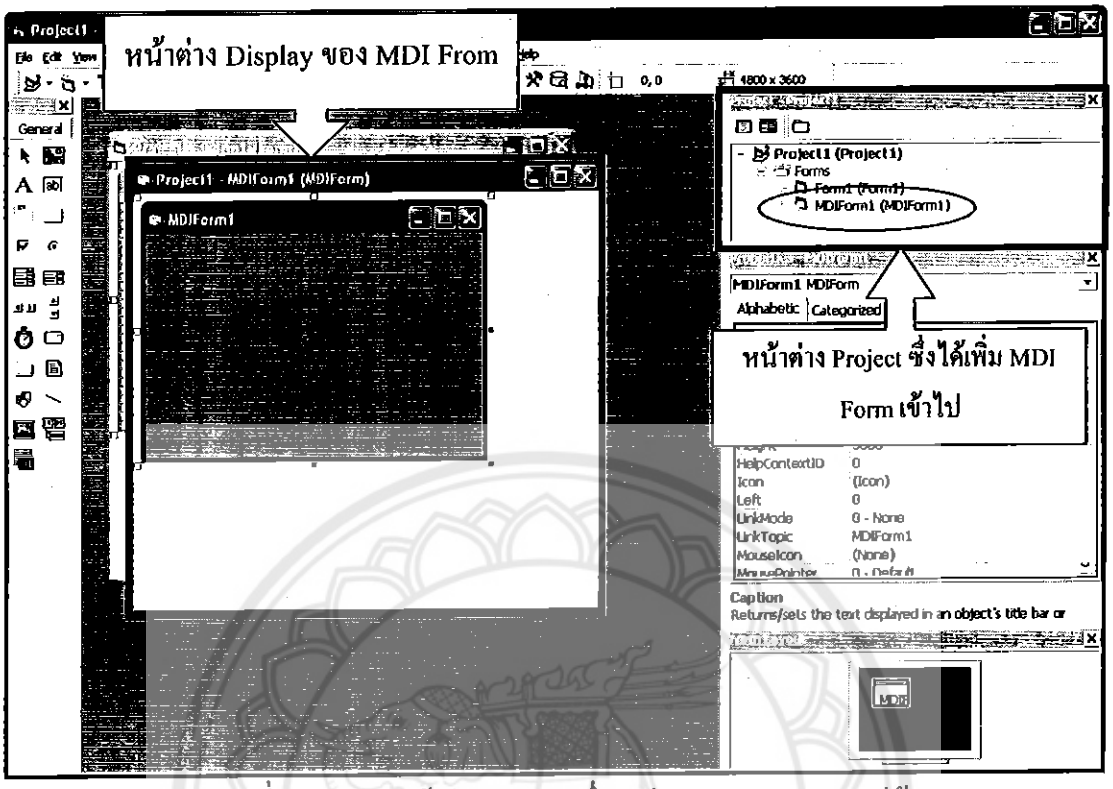
รูปที่ ข.2 แสดงหน้าตาของโปรแกรมเมื่อเลือกแอพพลิเคชันเป็นชนิด Standard EXE

3. สร้าง MDI Form ขึ้นมาเพื่อทำตัวโปรแกรมโดยที่เลือก Project >>เลือก Add MDI Form ดังรูปที่ ข.3



รูปที่ ข.3 แสดงการเพิ่ม MDI Form เข้าสู่โปรแกรม

เมื่อเพิ่ม MDI Form เข้าไปในโปรเจ็กแล้ว Form นี้จะถูกเพิ่มเข้าไปในหน้าต่างของ Project ซึ่งอยู่ทางซ้ายของตัวโปรแกรม Visual Basic 6.0 ดังแสดงในรูปที่ ข.4

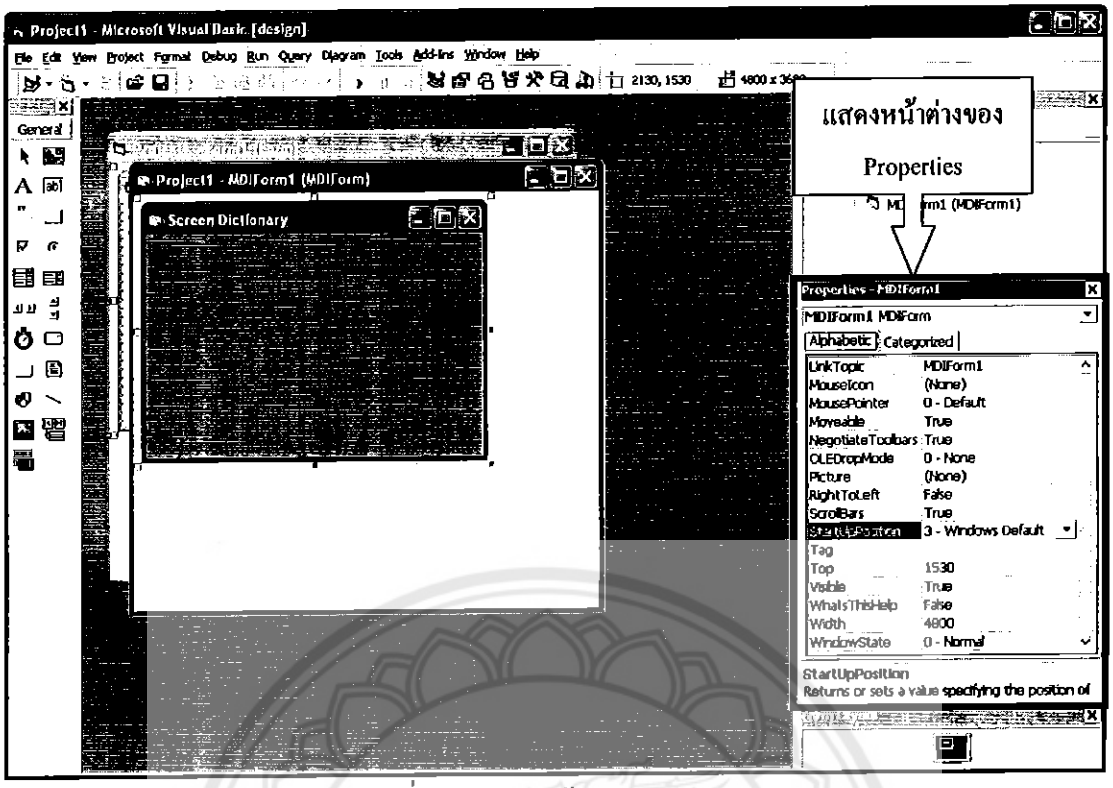


รูปที่ ข.4 แสดงหน้าต่าง Project ซึ่งจะมี MDI Form บรรจุอยู่ด้วย

4. เมื่อได้ MDI Form มาแล้วต่อไปจะเป็นการเซตค่าของ MDI Form ใน Properties ซึ่งช่องของ Properties จะอยู่ข้างล่างต่อกับหน้าต่าง Project สามารถเซตค่าได้ดังนี้

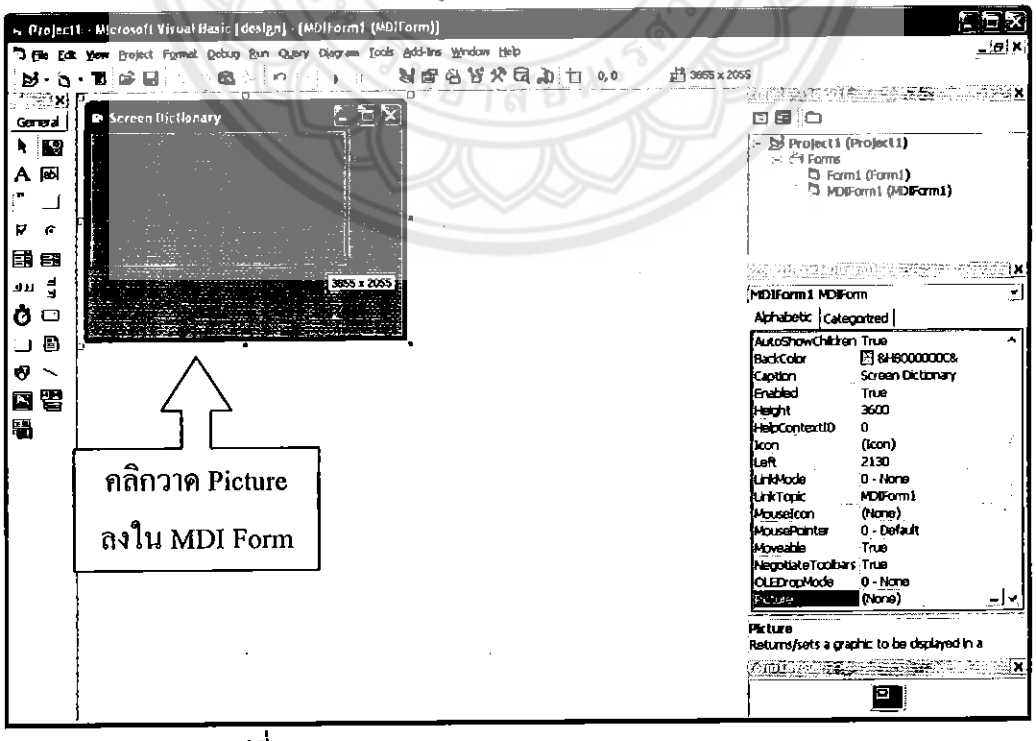
   ใน Tab Alphabetic

   - Caption : Screen Dictionary (เปลี่ยนTitle บน Form เป็น Screen Dictionary)

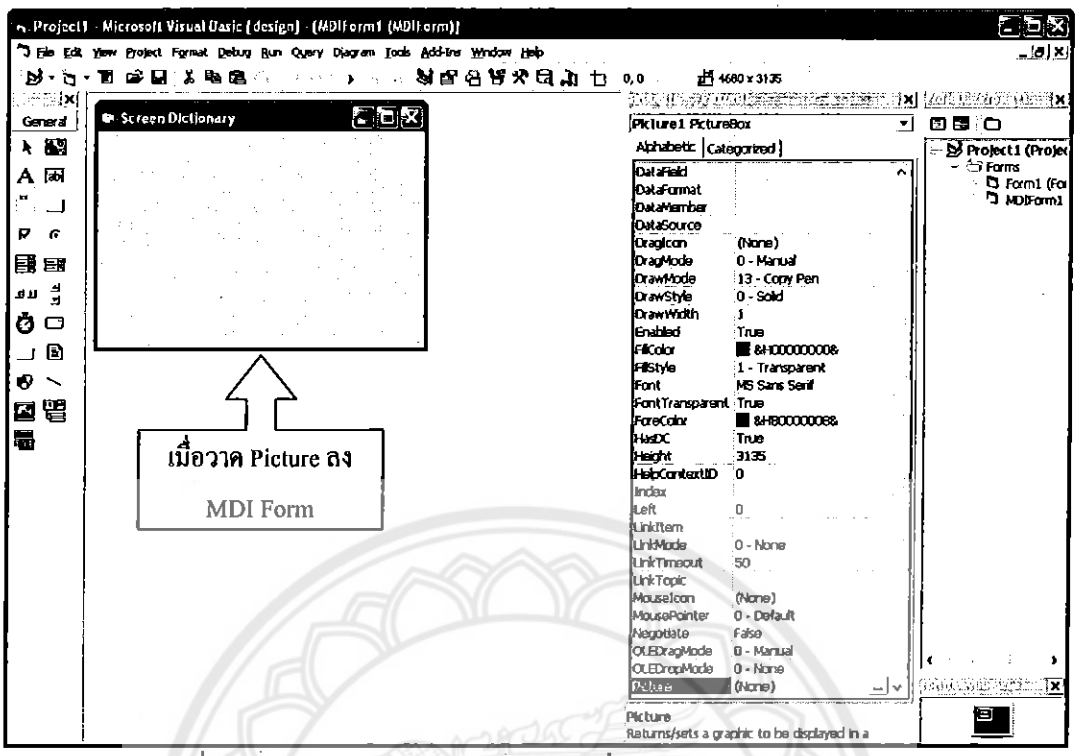


รูปที่ ข.5 แสดงหน้าต่างของ Properties

5. ต่อไปจะเป็นการวาด Tool Box ลงใน MDI From โดยที่ต้องวาด Picture Box ลงไปใน MDI Form เพื่อทำเป็น Background สามารถวาด Picture Box ได้จากคลิกที่ จากนั้นทำการวาด Picture Box ลงใน MDI Form ดังรูปที่ ข.6



รูปที่ ข.6 แสดงการวาด Picture Box ลงบน MDI From

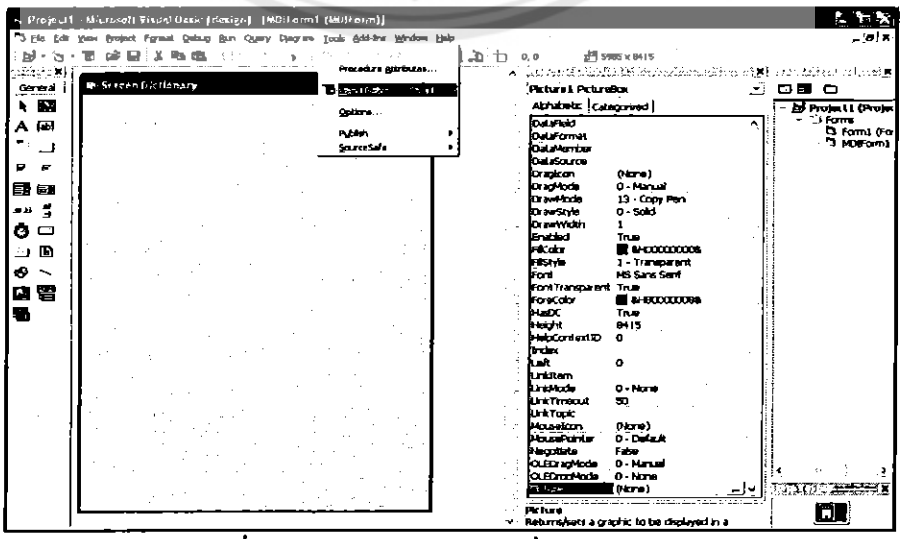


รูปที่ ข.7 แสดงหน้าตาของ MDI Form เมื่อทำการวาด Picture Box ลงไป

กำหนดค่าใน Properties ได้ดังนี้

- Appearance : 0 – Flat
- BackColor : &H00FFC0C0&
- BorderStyle : 0 - None

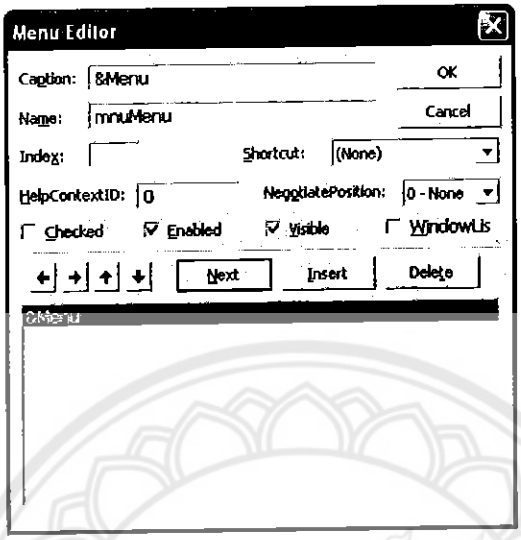
6. ต่อไปจะเป็นการสร้าง Menu Bar โดยที่เลือก Tools >>เลือก Menu Editor... Ctrl+E ดังรูปที่ ข.8



รูปที่ ข.8 แสดงการเรียกใช้คำสั่ง Menu Editor

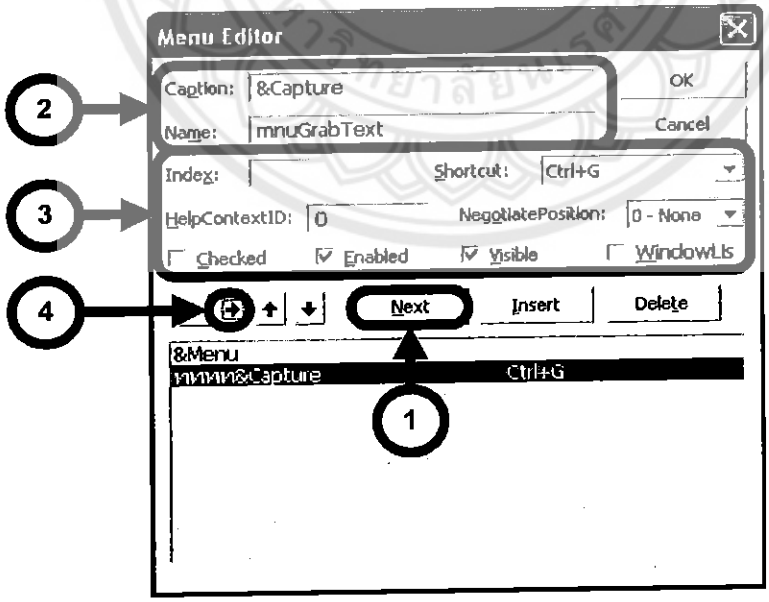
7. เมื่อเรียกคำสั่ง Menu Editor จะแสดงหน้าต่าง Menu Editor ขึ้นมาจากนั้นเซตค่าใน Menu Editor เพื่อสร้างคำสั่งลงใน Menu Bar สามารถสร้างได้ดังนี้

   1. การสร้างคำสั่ง Menu สามารถสร้างได้ดังรูป



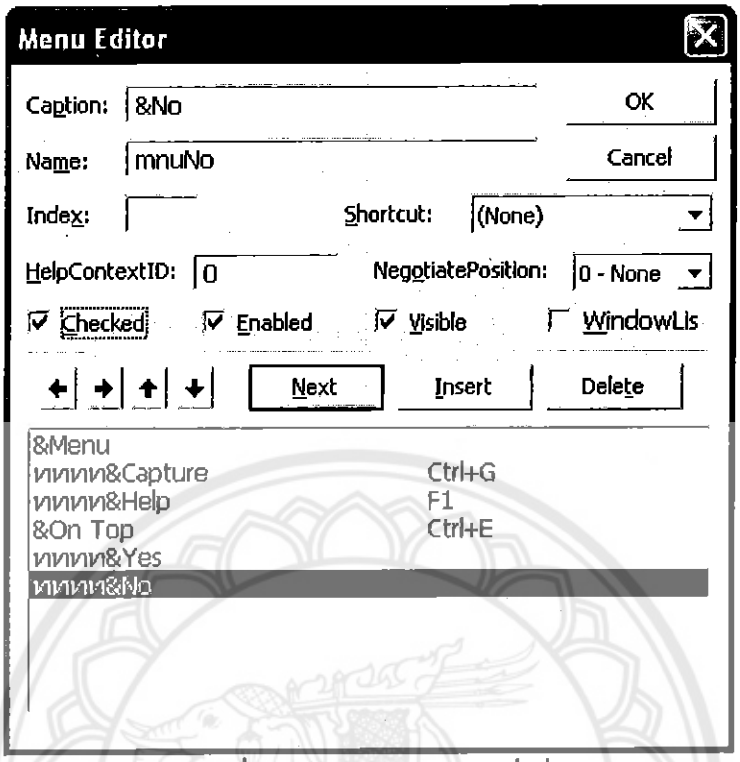
รูปที่ ข.9 แสดงการสร้างคำสั่ง Menu

2. การสร้างคำสั่ง Capture ซึ่งเป็นคำสั่งย่อยในคำสั่ง Menu สามารถสร้างตาม Step ได้ดังรูปที่ ข.10



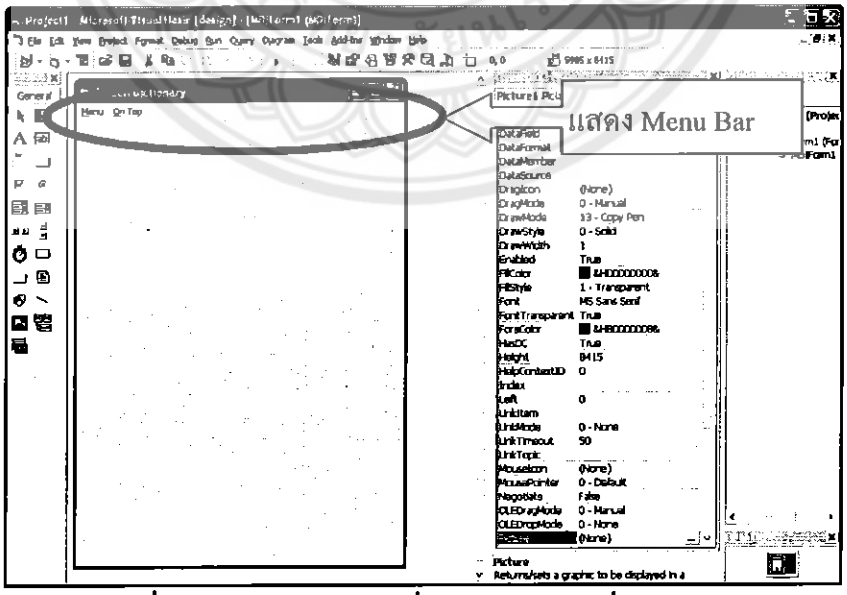
รูปที่10 แสดง Step ในการสร้างคำสั่ง Capture

3. ลำดับในการสร้างคำสั่งที่เหลือก็มี Step ในการสร้างเช่นเดียวกับ Step การสร้างคำสั่ง Menu และคำสั่ง Capture จะได้ดังรูปที่ ข.11



รูปที่ข.11 แสดงการสร้างคำสั่งอื่นๆ

4. เมื่อเซตค่าใน Menu Editor ได้ตามต้องการแล้วก็เลือก OK จะได้ Menu Bar ดังรูปที่ ข.12

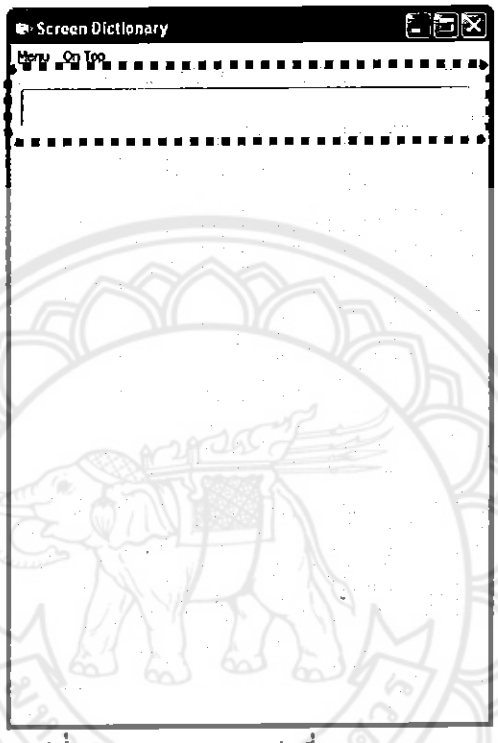


รูปที่ข.12 แสดง Menu Barที่ได้จากการใช้คำสั่ง Menu Editor

8. ต่อไปจะเป็นการสร้างช่อง Text Box เพื่อใช้แสดงตัวอักษร โดยการเรียกใช้คำสั่ง [abl] เมื่อทำวาด Text Box ไว้ใน MDI Form เสร็จแล้วกำหนดค่าใน Properties ดังนี้

ใน Tab Alphabetic

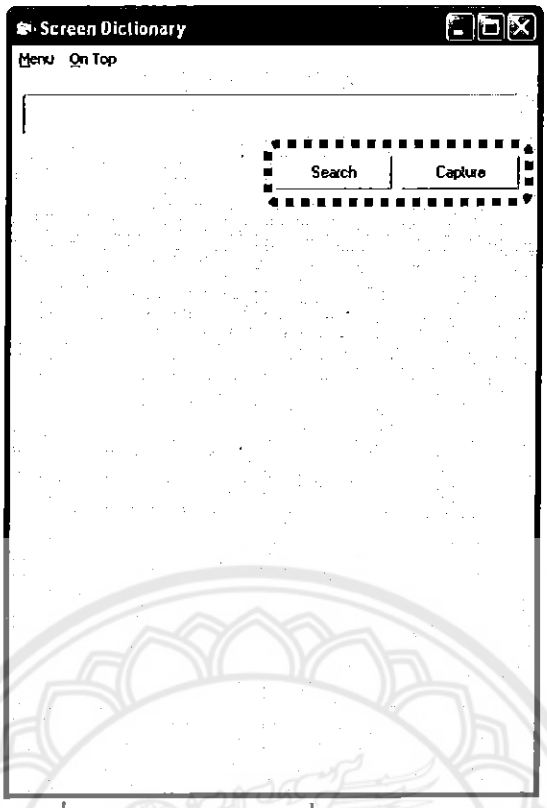
- Name : txtSearch
- Alignment : 2-Center
- Text : ลบคำว่า Text1 ออก



รูปที่ข.13 แสดงตำแหน่งที่วาด Text Box

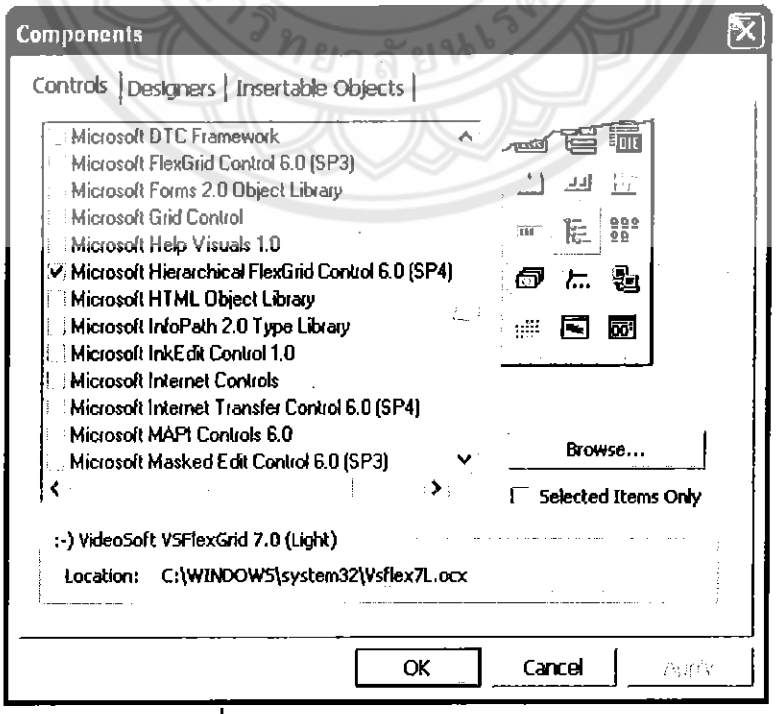
9. การสร้าง Command Button [ ] จะมี 2 Button ด้วยกันคือ Search Button และ Capture Button และกำหนดค่าใน Properties ได้ดังนี้

- Search Button ใน Tab Alphabetic
  - Name : cmdSearch
  - Caption : Search
- Capture Button ใน Tab Alphabetic
  - Name : capture
  - Caption : Capture



รูปที่ข.14 แสดงตำแหน่งที่วาด Command Button

10. การสร้าง MSHFlexGrid นั้นในแอพพลิเคชันแบบ Standard EXE จะไม่มีคำสั่งนี้ดังนั้นจึงต้องเพิ่มโดยที่เลือก > Project >> Components จะปรากฏหน้าต่าง Components ให้เลือก Microsoft Hierarchical FlexGrid Control 6.0 (SP4) จากนั้นเลือก OK

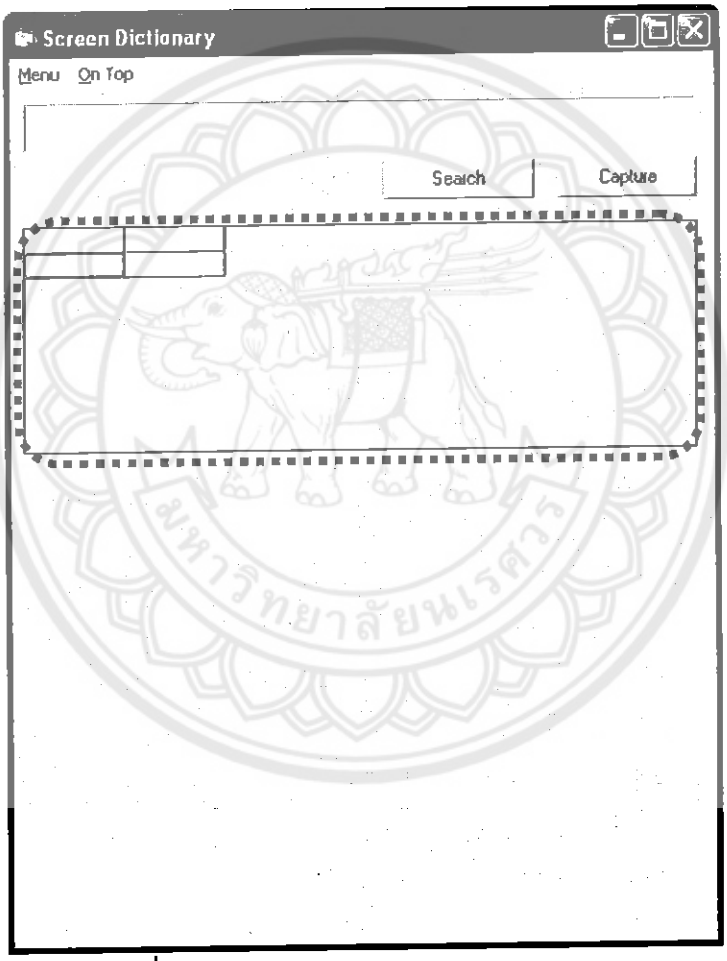


รูปที่ข.15 แสดงหน้าต่างของ Components

จะปรากฏคำสั่ง MSHFlexGrid  เมื่อวาด MSHFlexGrid ลงบน MDI Form เสร็จแล้วให้กำหนดค่าใน Properties ดังนี้

ใน Tab Alphabetic

- Name : mfgCustomers
- Appearance : 0 – flexFlat
- BackColorBkg : &H00FFC0C0&
- FixedCols : 0
- RowHeightMin : 1
- WordWrap : True



รูปที่ข.16 แสดงตำแหน่งในการวาง MSHFlexGrid

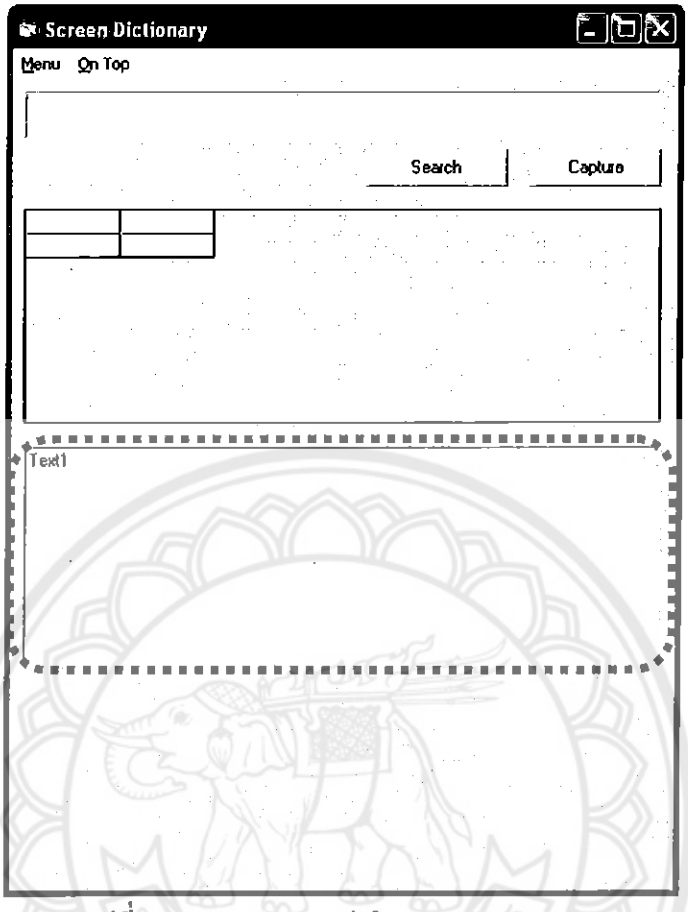
11. สร้าง Text Box เพื่อรับค่าจาก MSHFlexGrid กำหนด Properties ให้กับ Text Box ได้ดังนี้

ใน Tab Alphabetic

- MultiLine : True
- ScrollBars : 2 - Vertical

- Visible : False

Tool Box เมื่อ Run จะยังไม่แสดงจะแสดงก็ต่อเมื่อมีการคลิกใน MSHFlexGrid

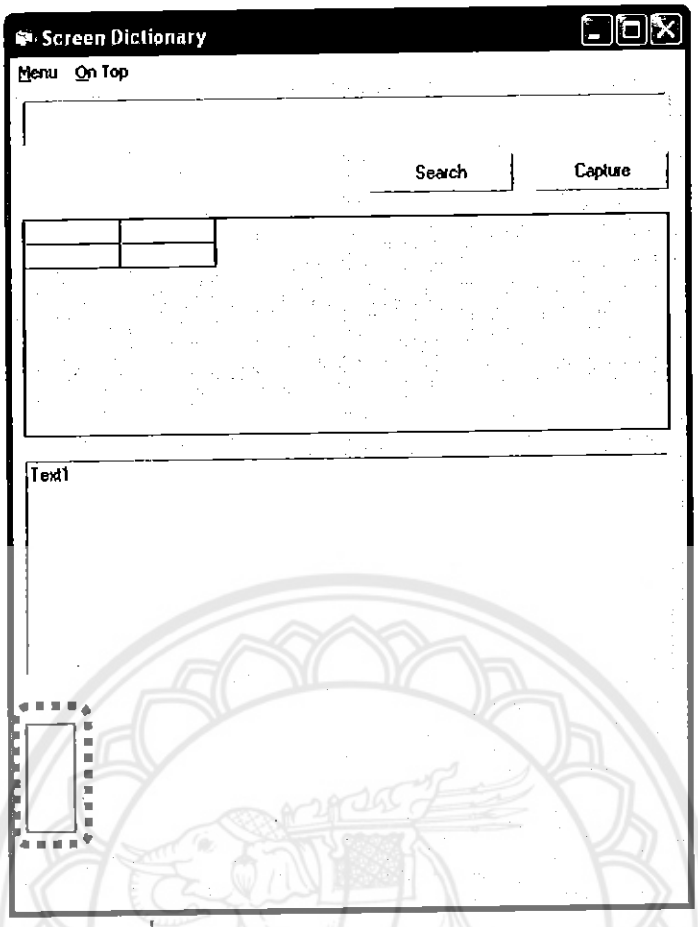


รูปที่ ข.17 แสดงตำแหน่งในการวาง Text Box

12. สุดท้ายเป็นการการสร้าง Picture Box เพื่อแสดงผลลัพท์ที่เป็นรูปภาพสามารถกำหนด Properties ได้ดังนี้

ใน Tab Alphabetic

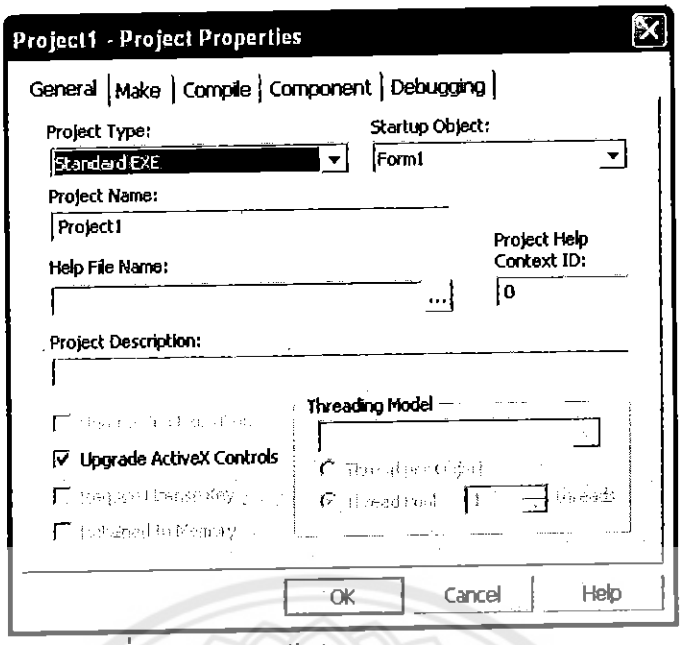
- Name : Picture4
- Appearance : 0 – Flat
- AutoRedraw : True
- AutoSize : True
- Index : 0



รูปที่ ข.18 แสดงตำแหน่งในการวาง Picture Box

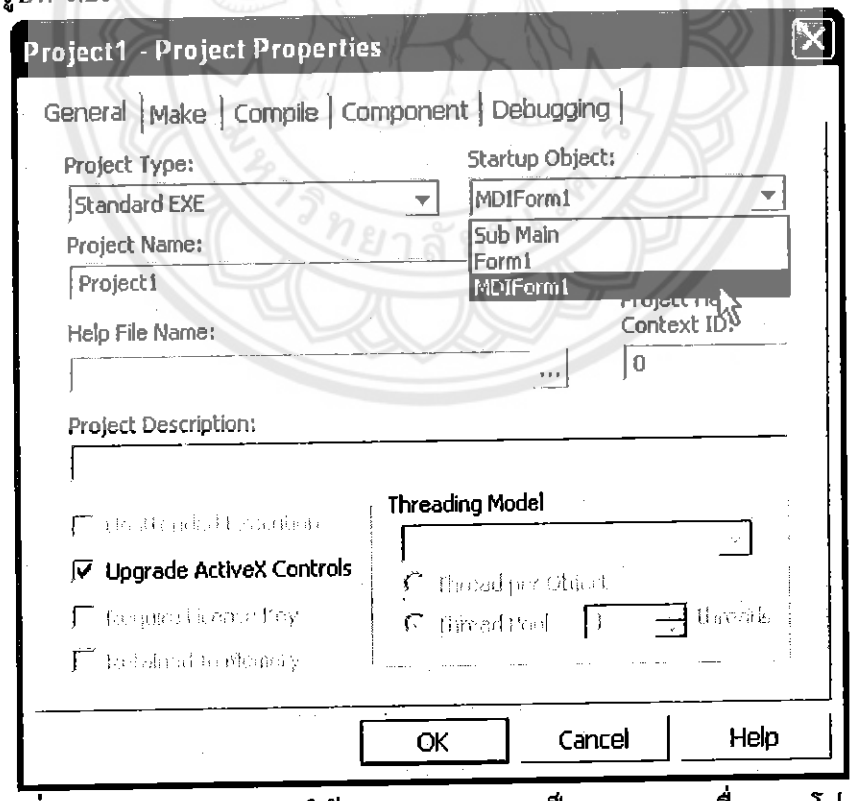
หลังจากที่เราได้สร้าง Layout ของโปรแกรม Screen Dictionaryเสร็จแล้ว Step ต่อไปจะเป็นการเขียน Code เพื่อกำหนดหน้าที่ของ Tool Box ต่างเพื่อให้โปรแกรมใช้ได้

13. เมื่อเรา Run ▶ โปรแกรมจะเห็นว่า Form แสดงขึ้นมานั้นไม่ใช่ Screen Dictionary กลับเป็น Form1 ที่แสดงขึ้นมาแทนนั้นเป็นเพราะว่าโปแกรมได้กำหนดให้ Form1 แสดงเป็น Form แรก เราจึงต้องตั้งค่าให้ Screen Dictionary แสดงเป็น Form แรกเมื่อทำการ Run โดยที่ก่อนอื่นหยุด ■ โปรแกรมก่อนจากนั้นเลือก > Project >> Project1 Properties จะแสดงหน้าต่าง Project Properties ดังรูปที่ ข.19



รูปที่ข.19 แสดงหน้าต่างของ Project Properties

14. หน้าต่าง Project Properties ในโหมด Startup Object: ให้เลือก Form ที่ต้องการแสดงเป็น Form แรกเมื่อ Run ในที่นี้เราจะเลือกให้ MDI Form แสดงเป็น Form แรกจากนั้นคลิก OK ดังรูปที่ ข.20

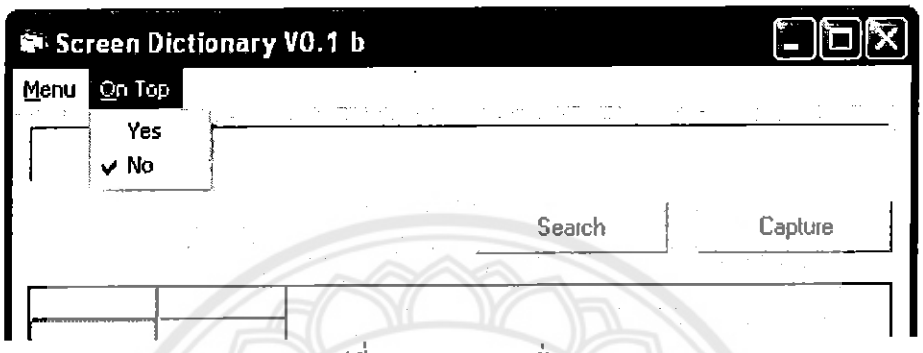


รูปที่ ข.20 แสดงการกำหนดให้ MDI Form แสดงเป็น From แรกเมื่อ Run โปรแกรม

15. จากนั้นลอง Run ▶ โปรแกรมจะเป็นว่า Screen Dictionary แสดงขึ้นมาแทน Form1 ต่อไปจะเป็นการเขียน Code ให้กับ Tool Box ต่างๆ โดยที่จะเขียนให้กับคำสั่ง On Top

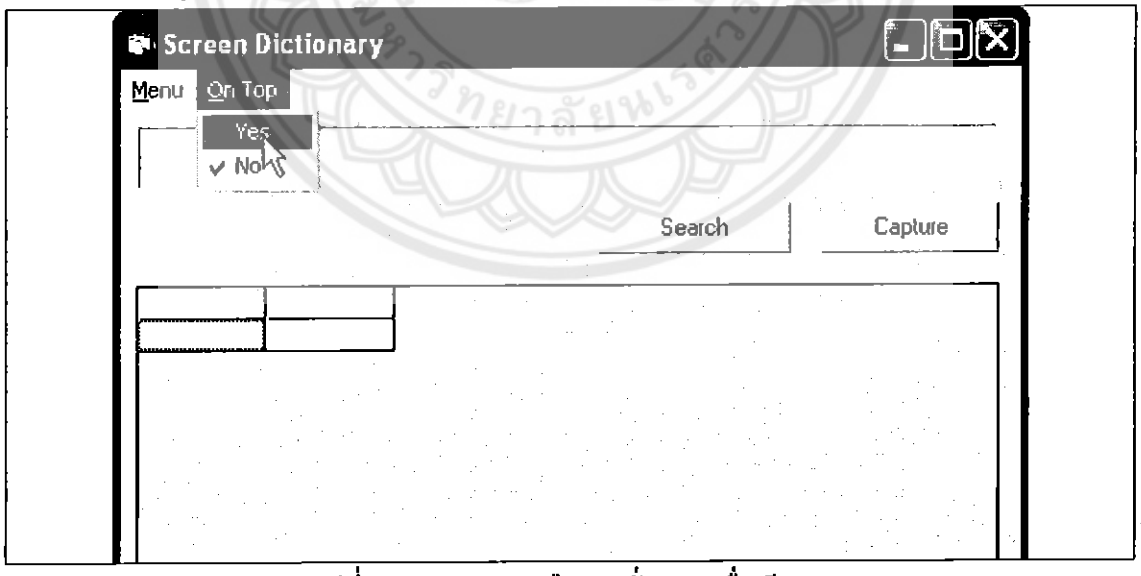
คำสั่ง On Top

คำสั่งนี้เป็นคำสั่งเสริมซึ่งจะทำให้หน้าต่างโปรแกรมอยู่บนสุดเสมอ ซึ่งจะมีตัวเลือกให้เลือกระหว่าง Yes หรือ No ดังรูปที่ ข.21 เขียนฟังก์ชันดังนี้

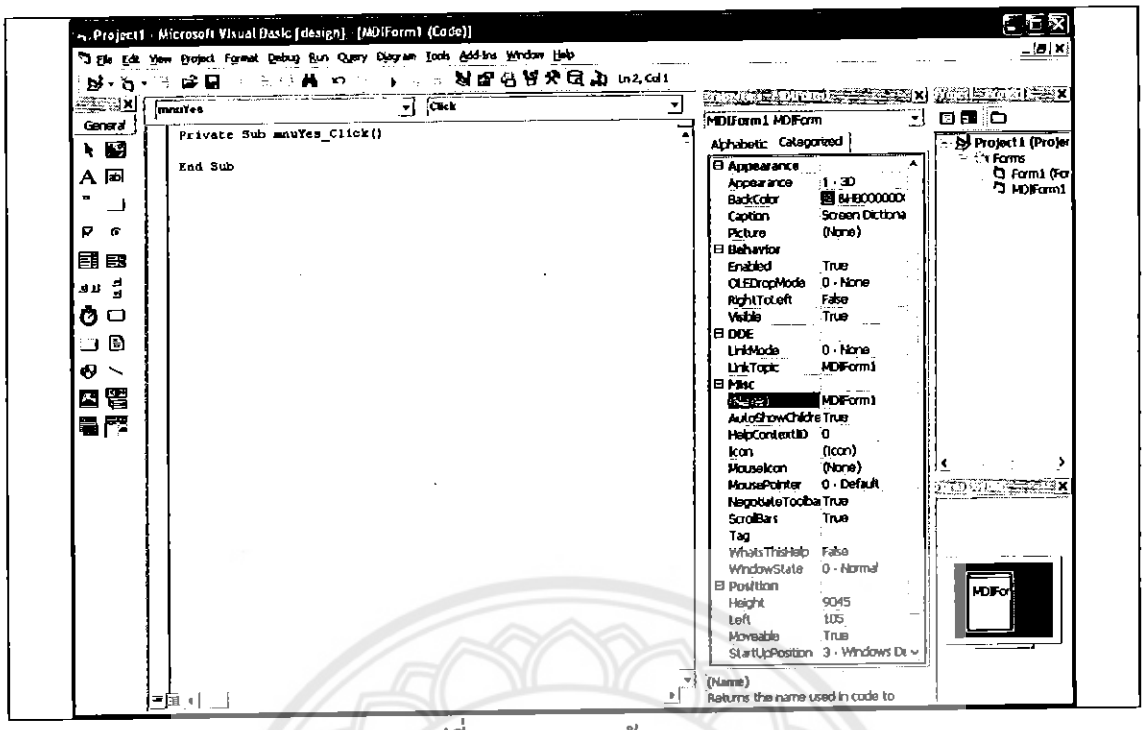


รูปที่ ข.21 แสดงคำสั่ง On Top

จากรูปที่ข.21 จะเห็นว่ามีสองคำสั่งด้วยกันคือ คำสั่ง Yes และ คำสั่ง No ซึ่งในการเขียน Code เข้าไปในคำสั่งได้โดยการคลิกคำสั่งที่ต้องการเขียน Code เข้าไปเช่นหากต้องการเขียน Code ให้คำสั่ง Yes ก็คลิกที่ Yes โปรแกรม Visual Basic 6.0 จะทำการ generate ฟังก์ชันขึ้นมาให้ในส่วนของ View Code 📄 ดังรูปที่ข.22 และรูปที่ ข.23 ตามลำดับ



รูปที่ข.22 แสดงการเลือกคำสั่ง Yes เพื่อเขียน Code

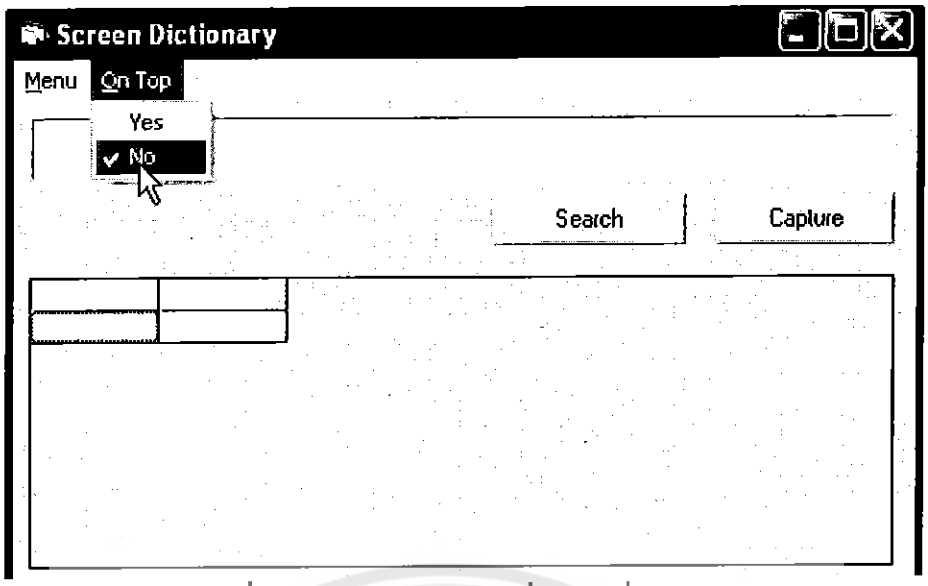


รูปที่ ข.23 แสดงหน้า View Code

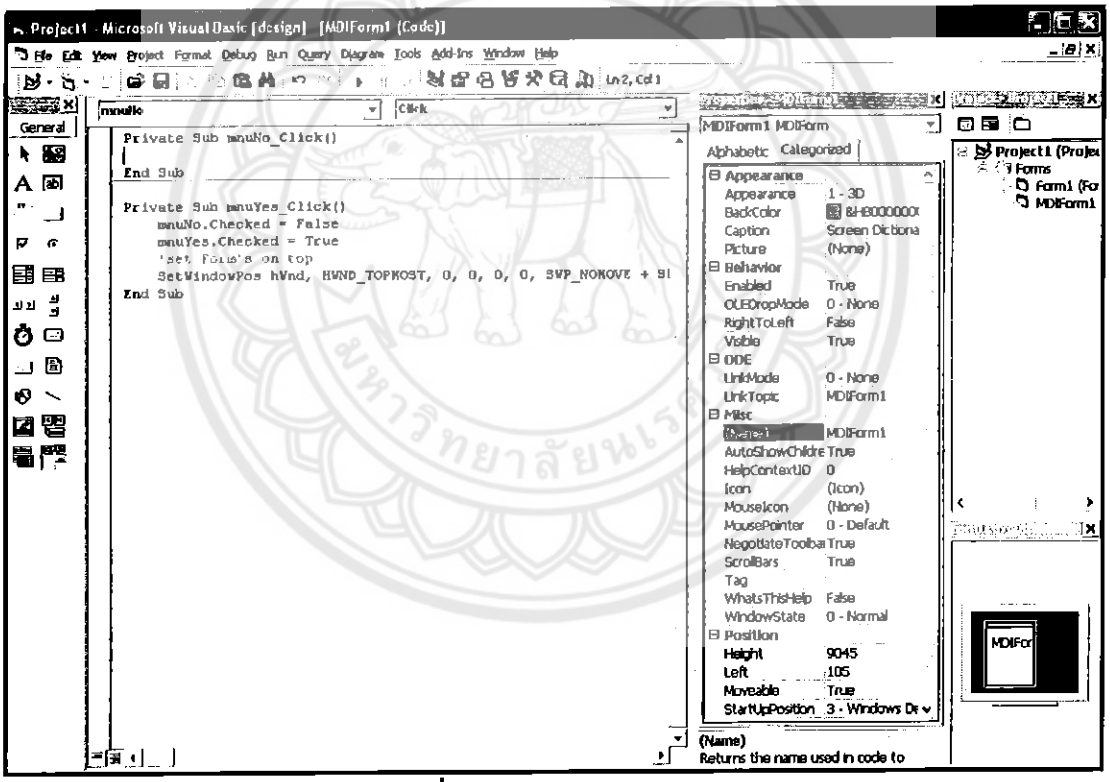
จากนั้นให้เพิ่ม Code ลงในชุดคำสั่งของ Private Sub mnuYes_Click() ดังนี้

```
Private Sub mnuYes_Click()
        mnuNo.Checked = False
        mnuYes.Checked = True 'แสดงเครื่องหมาย ✔ หน้าคำสั่ง Yes
        'set Form's on top
        SetWindowPos hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE +
    SWP_NOSIZE
End Sub
```

ต่อไปจะเป็นการเขียน Code ให้คำสั่ง No ซึ่งวิธีการนั้นจะเป็นเช่นเดียวกับคำสั่ง Yes โดยที่เริ่มจากคลิกเลือกที่คำสั่ง No โปรแกรม Visual Basic 6.0 จะทำการ generate ฟังก์ชันขึ้นมาให้ในส่วนของ View Code ดังรูปที่ ข.24 และรูปที่ ข.25 ตามลำดับ



รูปที่ ข.24 แสดงการเลือกคำสั่ง Yes เพื่อเขียน Code



รูปที่ ข.25 แสดงหน้า View Code

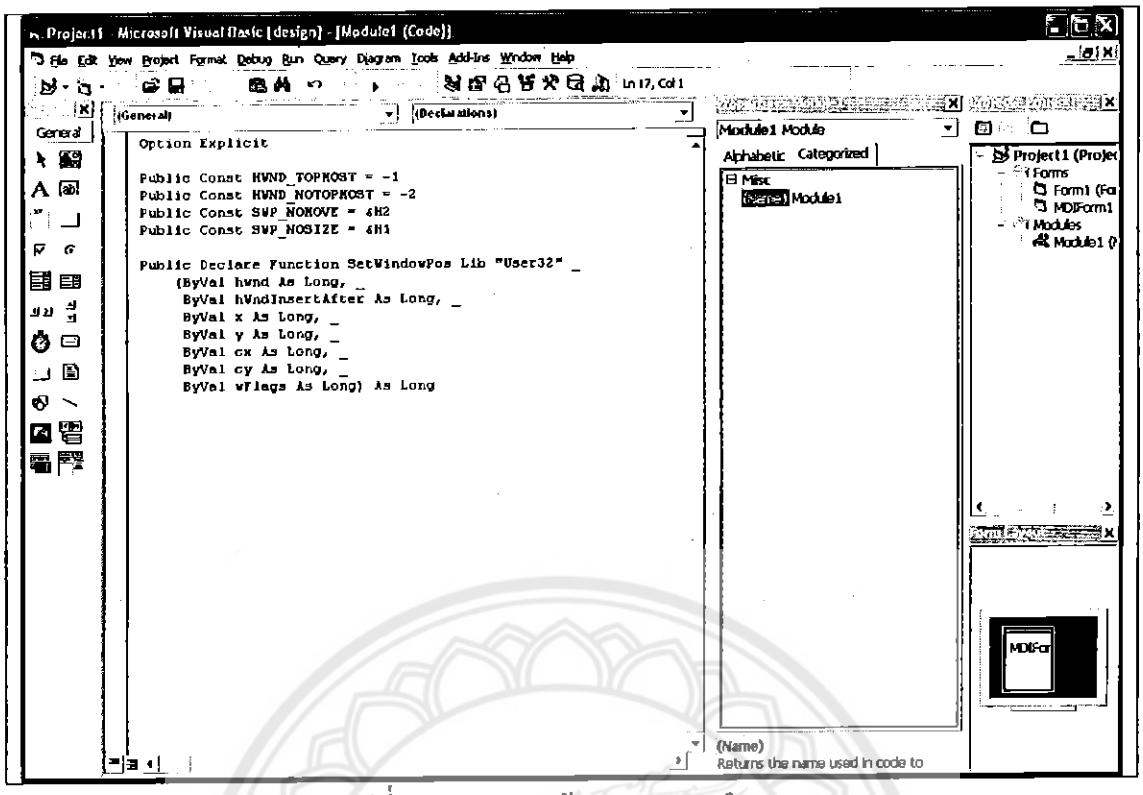
จากนั้นให้เพิ่ม Code ลงในชุดคำสั่งของ Private Sub mnuNo_Click() ดังนี้

```
Private Sub mnuNo_Click()
        mnuYes.Checked = False
        mnuNo.Checked = True 'แสดงเครื่องหมาย ✔ หน้าคำสั่ง No
        'Unset Form's on top
```

```
                SetWindowPos hwnd, HWND_NOTOPMOST, 0, 0, 0, 0, SWP_NOMOVE +
            SWP_NOSIZE
End Sub
```

ทดลอง Run โปรแกรมเมื่อเลือกคำสั่ง Yes หรือ No จะยังไม่สามารถใช้ได้เนื่องจากยังไม่ได้ประกาศฟังก์ชัน SetWindowPos จึงต้องประกาศฟังก์ชันก่อนโดยที่จะนำฟังก์ชันนี้ไปประกาศไว้ในส่วนของ Module โดยที่ต้องสร้าง Module ขึ้นมาก่อนเลือก >Project >> Add Module เมื่อสร้างเสร็จก็นำฟังก์ชัน SetWindowPos ไปว่างไว้ใน View Code ของ Module ดังรูปที่ ข.25 ทดลอง Run โปรแกรมอีกครั้งก็จะสามารถใช้ชุดคำสั่งได้ทั้ง 2 คำสั่ง

```
Public Const HWND_TOPMOST = -1
Public Const HWND_NOTOPMOST = -2
Public Const SWP_NOMOVE = &H2
Public Const SWP_NOSIZE = &H1
Public Declare Function SetWindowPos Lib "User32" _
                (ByVal hwnd As Long, _
                 ByVal hWndInsertAfter As Long, _
                 ByVal x As Long, _
                 ByVal y As Long, _
                 ByVal cx As Long, _
                 ByVal cy As Long, _
                 ByVal wFlags As Long) As Long
```



รูปที่ ข.26 แสดงหน้า View Code ใน Module

16. คำสั่งต่อไปที่จะเขียน Code ลงไปคือคำสั่ง Help

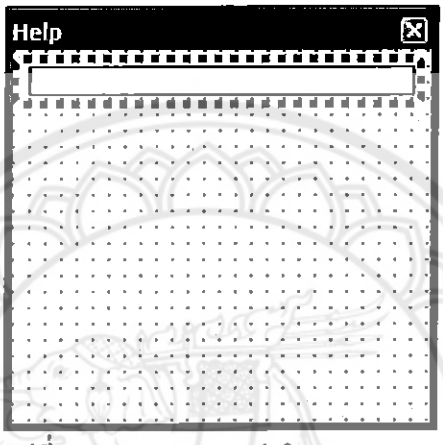
คำสั่ง Help

เป็นคำสั่งที่มีไว้บอกถึงข้อมูลของคณะผู้จัดทำ เมื่อเลือกคำสั่งนี้จะปรากฏ Form2 ขึ้นมาซึ่งข้างใน Form2 จะเก็บข้อมูลของคณะผู้พัฒนา สามารถสร้าง Form2 ได้โดยการที่เลือก > Project >>Add Form เมื่อได้ Form2 มาแล้วให้วาด Tool Box ชนิด Label A ลงบน Form2 ดังรูปที่ ข.26 จากนั้นให้เขียน Code ลงใน From1 ดังต่อไปนี้

```
Private Sub mnuHelp_Click()
        Form2.Visible = True
        Form2.Label1 = "        " & vbCrLf & _
                "        " & vbCrLf & _
                "    Get Text Project by" & vbCrLf & _
                "        " & vbCrLf & _
                "          ID    :46380033" & vbCrLf & _
                "          Name  :Mr.Pichit Thammawong " & vbCrLf & _
                "          E-mail:u46380033@nu.com " & vbCrLf & _
```

```
                    "      " & vbCrLf & _
                    "        ID    :46380191" & vbCrLf & _
                    "        Name  :Mr.Peerapol Nokphuak " & vbCrLf & _
                    "        E-mail:u46380191@nu.com " & vbCrLf & _
                    "      " & vbCrLf & _
                    "      "

End Sub
```



รูปที่ ข.27 แสดงตำแหน่งในการวาด Label1

กำหนดค่าใน Properties ของ Form2 และ Label ได้ดังนี้

Form2 ใน Tab Alphabetic

- AutoRedraw : True
- BorderStyle : 5 – Sizeble ToolWindow
- Caption : Help
- Height : 3180
- Left : 3225
- ScaleHeight : 2790
- ScaleWidth : 3150
- Top : 3210
- Width : 3270

Label ใน Tab Alphabetic

- Appearance : 0 – Flat
- AutoSize : True

- BorderStyle : 1 – Fixed Single
- Caption : ลบ Label1 ออก
- Width : 2895
- WordWarp : True

ทดลอง Run โปรแกรมแล้วเลือกคำสั่ง Help ซึ่งอยู่ใน Menu ก็จะปรากฏ Form2 ที่ชื่อ Help ขึ้นมาดังรูปที่ ข.28

รูปที่ ข.28 แสดง Form2 เมื่อเลือกใช้คำสั่ง Help

17. สั่งต่อไปที่จะเขียน Code ลงไปคือ คำสั่ง Exit เป็นคำสั่งหยุดการทำงานของโปรแกรมโดยที่สองวิธีในการหยุดคือ คลิก  กับเลือกที่ Menu >> Exit

หยุดโปรแกรมโดยวิธีคลิก 

ในการเขียน Code จะเขียนลงในโปรแกรมหลัก (MDI Form) สามารถเขียนได้ดังนี้

```
Private Sub MDIForm_Unload(Cancel As Integer)
    Unload Form1
    Unload Form2
    Unload Me 'หมายถึงโปรแกรมหลัก(MDI Form)
End Sub
```

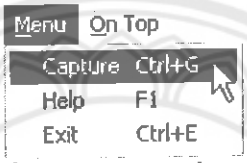
หยุดโปรแกรมโดยเลือกที่ Menu >> Exit

ลักษณะในการเขียน Code ลงไปในคำสั่ง Exit จะใช้วิธีเดียวกับคำสั่ง On Top จากนั้นก็เขียน Code ต่อไปนี้ลงไป

```
Private Sub mnuExit_Click()
    Unload Form1
    Unload Form2
    Unload Me
End Sub
```

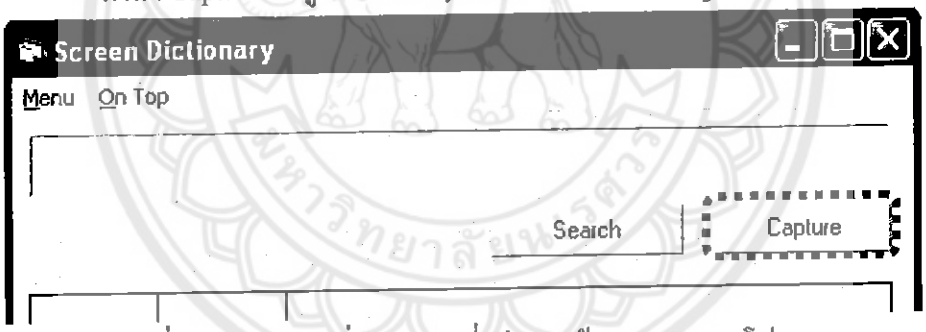
18. คำสั่งต่อไปจะเป็นคำสั่งที่ถือว่าเป็นหัวใจหลักของโปรแกรม คือคำสั่ง Capture คำสั่งนี้สามารถเรียนกใช้ได้ 2 ทางคือ

- คำสั่ง Capture ที่อยู่ใน Menu ดังรูปที่ ข.29



รูปที่ ข.29 แสดงคำสั่ง Capture ใน Menu

- คำสั่ง Capture ที่อยู่บนหน้า Layout ของโปรแกรม ดังรูปที่29



รูปที่ ข.30 แสดงคำสั่ง Captureที่อยู่บนหน้า Layout ของโปรแกรม

ทั้งนี้ทั้งนั้นทั้งสองคำสั่งจะเรียกใช้ฟังก์ชัน StartGet เหมือนกันทั้ง 2 คำสั่งซึ่งสามารถเขียน Code ได้ดังนี้

Code ที่อยู่ในคำสั่งCapture ใน Menu bar เขียนได้ดังนี้

```
Private Sub mnuGrabText_Click()
    StartGet
End Sub
```

Code ที่อยู่ในคำสั่ง Capture บน Layout ของโปรแกรม

```
Private Sub capture_Click()
    StartGet
End Sub
```

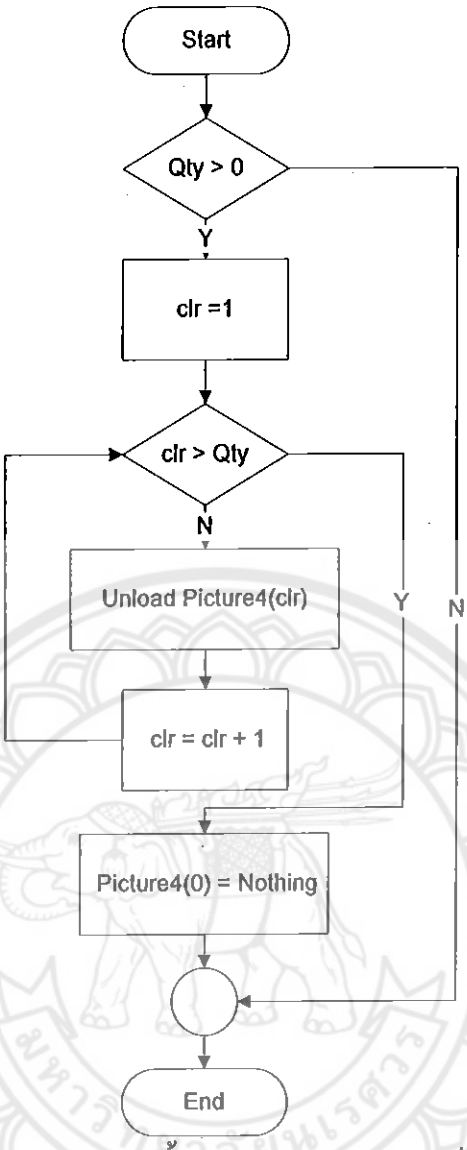
ต่อไปจะเป็นการอธิบายขั้นตอนการทำงานของคำสั่งนี้จากข้างต้นคำสั่งจะมีการเรียกใช้ฟังก์ชัน StartGet ในฟังก์ชันนี้จะมี Process ซึ่งเขียนเป็นฟังก์ชันได้ดังนี้

```
Private Sub StartGet()
    ClearIndexPic  'เรียกใช้ฟังก์ชัน ClearIndexPic
    Text1.Text = " "  'ลบตัวอักษรใน Text1
    txtSearch.Text = " "  'ลบตัวอักษรใน txtSearch
    Text3.Text = " "  'ลบตัวอักษรใน Text3
    Picture3(0) = Nothing  'Clear ค่าใน Array Picture3(0)
    Picture4(0) = Nothing  ' Clear ค่าใน Array Picture4(0)
    MDIForm1.Visible = False  ' Unload MDIFrom1
    Form1.Visible = False  'Unload Form1
    Timer1.Enabled = True  'Start Timer1
End Sub
```

จากฟังก์ชันข้องต้นจะเห็นว่าตอนนี้โปรแกรมเรายังไม่ได้สร้าง Tool box ต่างๆดังนี้ Text3, Picture3(0), Timer1 ซึ่ง Tool Box เหล่าผู้พัฒนาจะกล่าวถึงวิธีการสร้างในครั้งต่อไป

ฟังก์ชัน ClearIndexPic

ในฟังก์ชัน StartGet จะเริ่มต้นโดยการเรียกใช้ฟังก์ชัน ClearIndexPic ฟังก์ชันนี้จะเป็นการ Clear ค่าใน Picture4(0) ซึ่งเป็น Tool Box ชนิด Arrayสามารถเขียนเป็น Algorithm และฟังก์ชันได้ดังนี้



รูปที่ ข.31 Algorithm แสดงขั้นตอนการทำงานของฟังก์ชัน ClearIndexPic

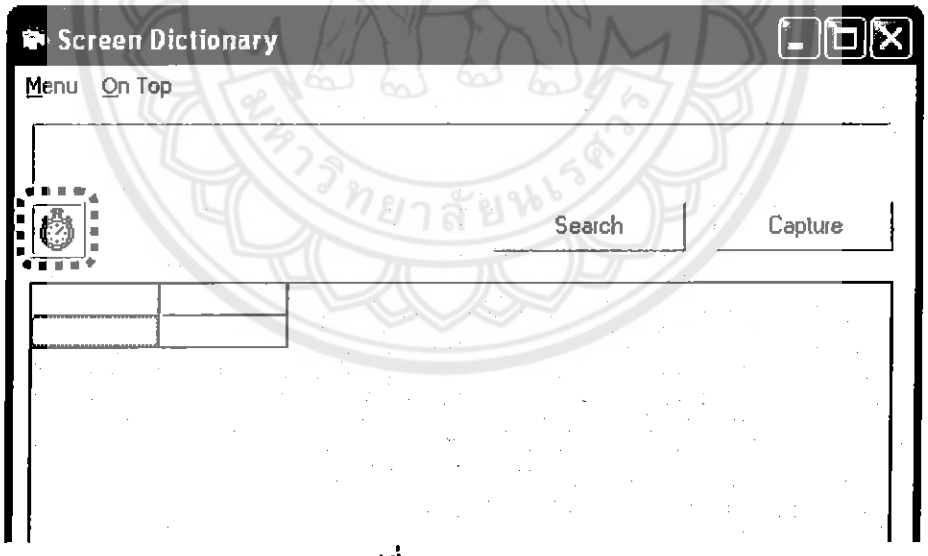
```
Public Sub ClearIndexPic()
  If Qty > 0 Then
  Dim clr As Integer
    For clr = 1 To Qty
      Unload Picture4(clr)
    Next
  Picture4(0) = Nothing
  End If
End Sub
```

เมื่อออกจากฟังก์ชัน ClearIndexPic ลำดับต่อไปจะกำหนดค่าใน Properties ของแต่ละ Tool Box คือ

- Text1.Text = " " 'ลบตัวอักษรใน Text1
- txtSearch.Text = " " 'ลบตัวอักษรใน txtSearch
- Text3.Text = " " 'ลบตัวอักษรใน Text3
- Picture3(0) = Nothing 'Clear ค่าใน Array Picture3(0)
- Picture4(0) = Nothing ' Clear ค่าใน Array Picture4(0)
- MDIForm1.Visible = False ' Unload MDIFrom1
- Form1.Visible = False 'Unload Form1

ฟังก์ชัน Timer1

ขั้นตอนต่อมาจะเรียกใช้ฟังก์ชัน Timer1 ฟังก์ชันนี้จะเป็นฟังก์ชันหน่วงเวลาเพื่อที่จะ Get หน้าจอให้กลายเป็นรูปภาพเหตุที่ต้องหน่วงเวลาเนื่องจากต้องรอให้ Form ต่างๆ Unload ให้หมดซะก่อนแล้วจึงทำการ Get หน้าจอในฟังก์ชันของ Timer1 นี้สามารถสร้างได้โดยที่เลือก  จากนั้นนำมาวาดลงในโปรแกรมหลักจะได้ดังรูปที่31



รูปที่ ข.32 แสดง Timer1

สามารถกำหนดค่าใน Properties ของ Timer1 ได้ดังนี้

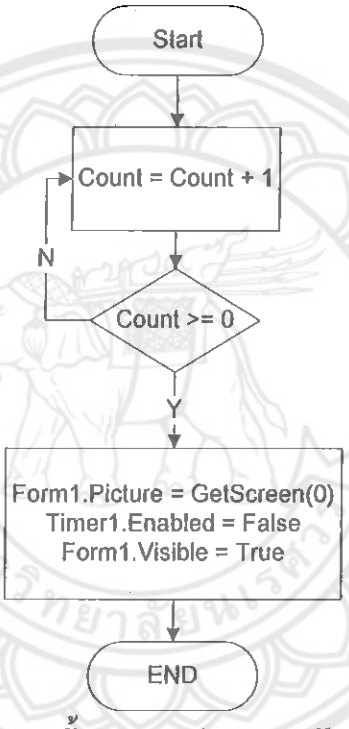
ใน Tab Alphabetic

- Enabled : False
- Interval : 500

ขั้นตอนการทำงานของฟังก์ชัน Timer1 คือเริ่มแรกจะเริ่มนับ โดยที่กำหนดให้ Static Count จากนั้นจะเช็คว่า Count >= 0 หรือไม่ถ้าไม่ก็ให้ทำการ Count ต่อไปแต่ถ้าใช่ก็ให้กำหนดค่าใน Prpperties ของแต่ละ Tool Box ดังนี้

- Form1.Picture = GetScreen(0) 'ให้ Background มีค่าเท่ากับค่าในฟังก์ชัน GetScreen โดยที่ Return ค่า 0 กับไปที่ฟังก์ชัน GetScreen
- Timer1.Enabled = False 'Unload ฟังก์ชัน Timer1
- Form1.Visible = True 'Load Form1 ขึ้นมา

สามารถเขียนเป็น Algorithm และเขียนเป็นฟังก์ชันได้ดังนี้



รูปที่ ข.33 แสดงขั้นตอนการทำงานของฟังก์ชัน Timer1

```
Private Sub Timer1_Timer()
   Static count
   If count >= 0 Then
      Form1.Picture = GetScreen(0)
      Timer1.Enabled = False
      Form1.Visible = True
   End If
End Sub
```

ฟังก์ชัน GetScreen

ฟังก์ชันนี้จะเป็นขั้นตอนในการแปลงค่าบนหน้าจอปัจจุบันให้กลายเป็นรูปภาพหรือพูดอีกในหนึ่งก็คือการทำ Cap Screen ขั้นตอนในการทำงานนั้นจะเริ่มจากเรียกใช้ฟังก์ชัน GetDesktopWindow เพื่อที่จะดึงค่า handle บน Desktop ของวินโดว์มาใช้เมื่อได้ค่า handle แล้วก็จะเรียกใช้ฟังก์ชัน GetWindowRect เพื่อที่จะนำค่าคู่ลำดับใน Desktop คือ ซ้ายบน, ซ้ายล่าง, ขวาบน, ขวาล่าง ออกมาเพื่อจะหาขนาดของ Desktop เมื่อได้ขนาดของ Desktop แล้วก็รับค่า device context ของวินโดว์ว่ามีขนาดเท่าไหร่ จากนั้นทำการสร้าง device context โดยการใช้ฟังก์ชัน CreateCompatibleDC เมื่อได้ device context ทำการ Copy ค่าใน device context แล้วแปลงให้เป็น bitmap แล้วนำไปเก็บไว้ใน Temp Picture โดยใช้ฟังก์ชัน CreateCompatibleBitmap จากนั้นทำการ คืนค่าที่จอง device context ไว้โดยใช่คำสั่ง DeleteDC และ ReleaseDC ตามลำดับเมื่อได้ค่าของ bitmap มาแล้วต่อไปจะเป็นการวาดรูปลงบน Form1 โดยใช้คำสั่ง OleCreatePictureIndirect สามารถเขียน Code ได้ดังนี้

```
Function GetScreen(Optional ByVal hwnd As Long) As IPictureDisp
        Dim targetDC As Long
        Dim hdc As Long
        Dim tempPict As Long
        Dim oldPict As Long
        Dim wndWidth As Long
        Dim wndHeight As Long
        Dim Pic As PICTDESC
        Dim rcWindow As RECT
        Dim guid(3) As Long
        If hwnd = 0 Then hwnd = GetDesktopWindow
    ' get window's size
```

```
    GetWindowRect hwnd, rcWindow
    wndWidth = rcWindow.Right - rcWindow.Left
    wndHeight = rcWindow.Bottom - rcWindow.Top
' get window's device context
    targetDC = GetWindowDC(hwnd)
' create a compatible DC
    hdc = CreateCompatibleDC(targetDC)

'create a memory bitmap in the DC just created
' the has the size of the window we're capturing
    tempPict = CreateCompatibleBitmap(targetDC, wndWidth, wndHeight)
    oldPict = SelectObject(hdc, tempPict)
' copy the screen image into the DC
    BitBlt hdc, 0, 0, wndWidth, wndHeight, targetDC, 0, 0, vbSrcCopy
' set the old DC image and release the DC
    tempPict = SelectObject(hdc, oldPict)
    DeleteDC hdc
    ReleaseDC GetDesktopWindow, targetDC
' fill the ScreenPic structure
    With Pic
      .cbSize = Len(Pic)
      .pictType = 1          ' means picture
      .hIcon = tempPict
      .hPal = 0          ' (you can omit this of course)
    End With
' convert the image to a IpictureDisp object
' this is the IPicture GUID {7BF80980-BF32-101A-8BBB-00AA00300CAB}
' we use an array of Long to initialize it faster
    guid(0) = &H7BF80980
    guid(1) = &H101ABF32
    guid(2) = &HAA00BB8B
    guid(3) = &HAB0C3000
```

```
        ' create the picture
        ' return an object reference right into the function result
          OleCreatePictureIndirect Pic, guid(0), True, GetScreen
End Function
```

เมื่อทำฟังก์ชัน GetScreen () เสร็จแล้วฟังก์ชัน Timer1 จะหยุดการทำงานและเปิด Form1 ขึ้นมาแล้วก็นำภาพของ Desktop Windows มาวางไว้ที่ Form1 ในการกำหนดค่า Properties ให้กับ Form1 มีรายละเอียดดังนี้

ใน Tab Alphabetic

- AutoRedraw : True
- BorderStyle : 0 – None
- DrawMode : 6 – Invert
- MaxButton : False
- MinButtin : False
- ScaleMode : 3 – Pixel
- WindowState : 2 – Maximized

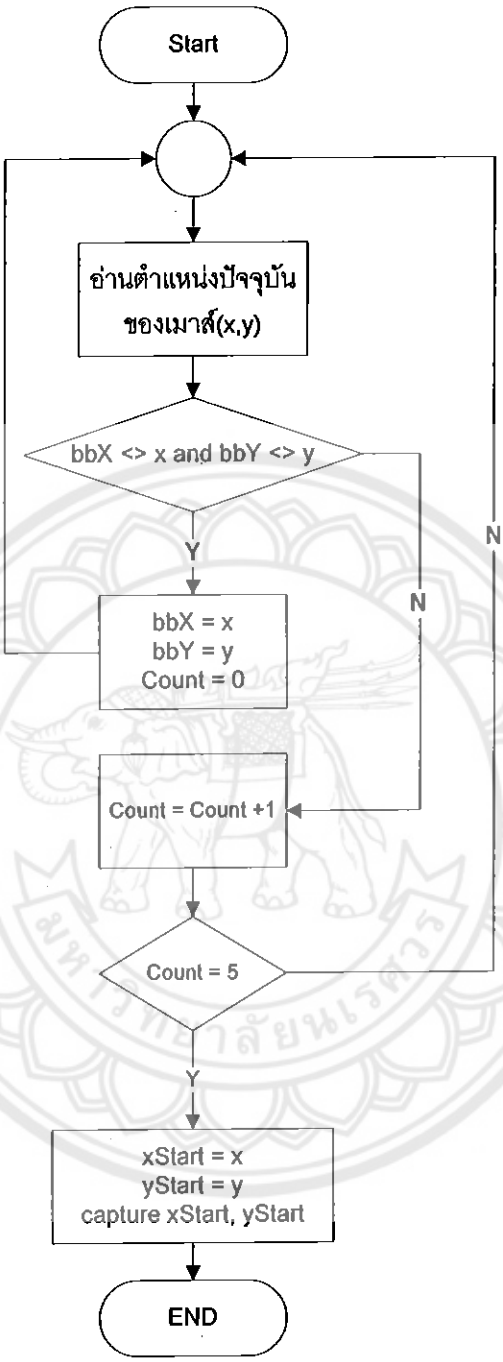
ฟังก์ชัน Timer2

ทันทีที่ Form2 แสดงฟังก์ชัน Timer2 จะทำงานทันที ฟังก์ชัน Timer2 นี้ทำหน้าที่ในการตรวจจับ Event ของ Mouse ว่ามีการหยุดอยู่กับที่หรือไม่ถ้าเมาส์หยุดอยู่กับที่นานเป็นเวลา 5 วินาทีให้เก็บค่าลำดับ x,y ที่ Mouse หยุดอยู่กับที่ส่งเข้าไปในฟังก์ชัน Capture และออกจากฟังก์ชัน Timer2 ในการสร้างTimer2 สามารถสร้างได้จาก  จากนั้นนำมาวาดลงใน Form1 การกำหนดค่าของ Timer2 ใน Properties สามารถกำหนดได้ดังนี้

ใน Tab Alphabetic

- Name : Timer2
- Enabled : True
- Interval : 1000

สามารถเขียน Algorithm และเขียน Code ได้ดังนี้



รูปที่ ข.34 แสดงการทำงานของ Timer2

```
Private Sub Timer2_Timer()
Static bbX As Long
Static bbY As Long
Dim pt As POINTAPI
Call GetCursorPos(pt)
        If ((bbX <> pt.x) And (bbY <> pt.y)) Then
            CheckEnd = 0
        Else
            CheckEnd = (CheckEnd + 1)
            If (CheckEnd > 0) Then
            xStart = pt.x: yStart = pt.y
            capture xStart, yStart
            End If
        End If
    bbX = pt.x
    bbY = pt.y
End Sub
```

ฟังก์ชัน Capture

เมื่อได้ค่าคู่ลำดับ x,y มาแล้วฟังก์ชัน Capture จะกำหนดให้ Timer2 หยุดการทำงานและจะเรียก Form Child ขึ้นมาโดยที่ภายใน Form Child นั้นจะมี Picture Box ไว้เพื่อเก็บพักรูปภาพที่ได้จากการตัด Form Child สามารถสร้างได้จาก > Project > Add Form > Form เมื่อได้ Form มาแล้วต่อไปจะกำหนดคุณสมบัติให้กับ Form Child และ Picture Box สามารถกำหนดได้ดังนี้

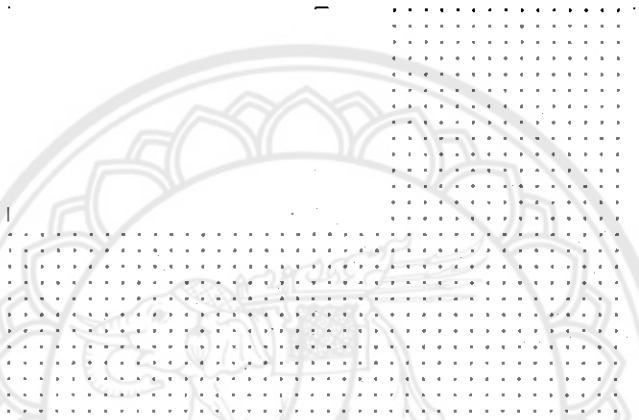
Form Child ใน Tab Alphabetic

- Name : frmChild
- AutoRedraw : Ture
- BorderStyle : 0 – None
- Caption : frmChild
- LinkTopic : Form2
- MaxButton : False
- MDIChild : True
- MinButton : False

- ScaleMode : 3 – Pixel

Picture Box ใน Tab Alphabetic

- AutoRedraw : Ture
- AutoSize : Ture
- BorderStyle : 0 – None
- Left : 0
- ScaleMode : 3 – Pixel
- Top : 0



รูปที่ ข.35 แสดงหน้าของ From Child เมื่อทำการกำหนดคุณสมบัติเสร็จแล้ว

หลักการในการตัดรูปภาพนั้นสามารถกลับไปดูได้ในบทที่ 2 เมื่อทำการตัดรูปภาพได้แล้วเพื่อเป็นง่ายในการวิเคราะห์จึงได้นำรูปจาก Form Child มาเก็บไว้ใน Picture2 ซึ่งอยู่ใน MDI Form สามารถกำหนดคุณสมบัติของ Picture2 ได้ดังนี้

Picture2 ใน Tab Alphabetic

- Appearance : 0 – Flat
- AutoRedraw : Ture
- AutoSize : Ture
- ScaleMode : 3 – Pixel

ในการทดสอบโปรแกรมจำเป็นต้องเห็นขั้นตอนในการ Run โปรแกรมผู้พัฒนาจึงกำหนดให้ Picture2 มีคุณสมบัติ Visible เป็น True ก่อนแต่พัฒนาโปรแกรมจนเสร็จผู้พัฒนาถึงจะกำหนด Visible ให้เป็น False

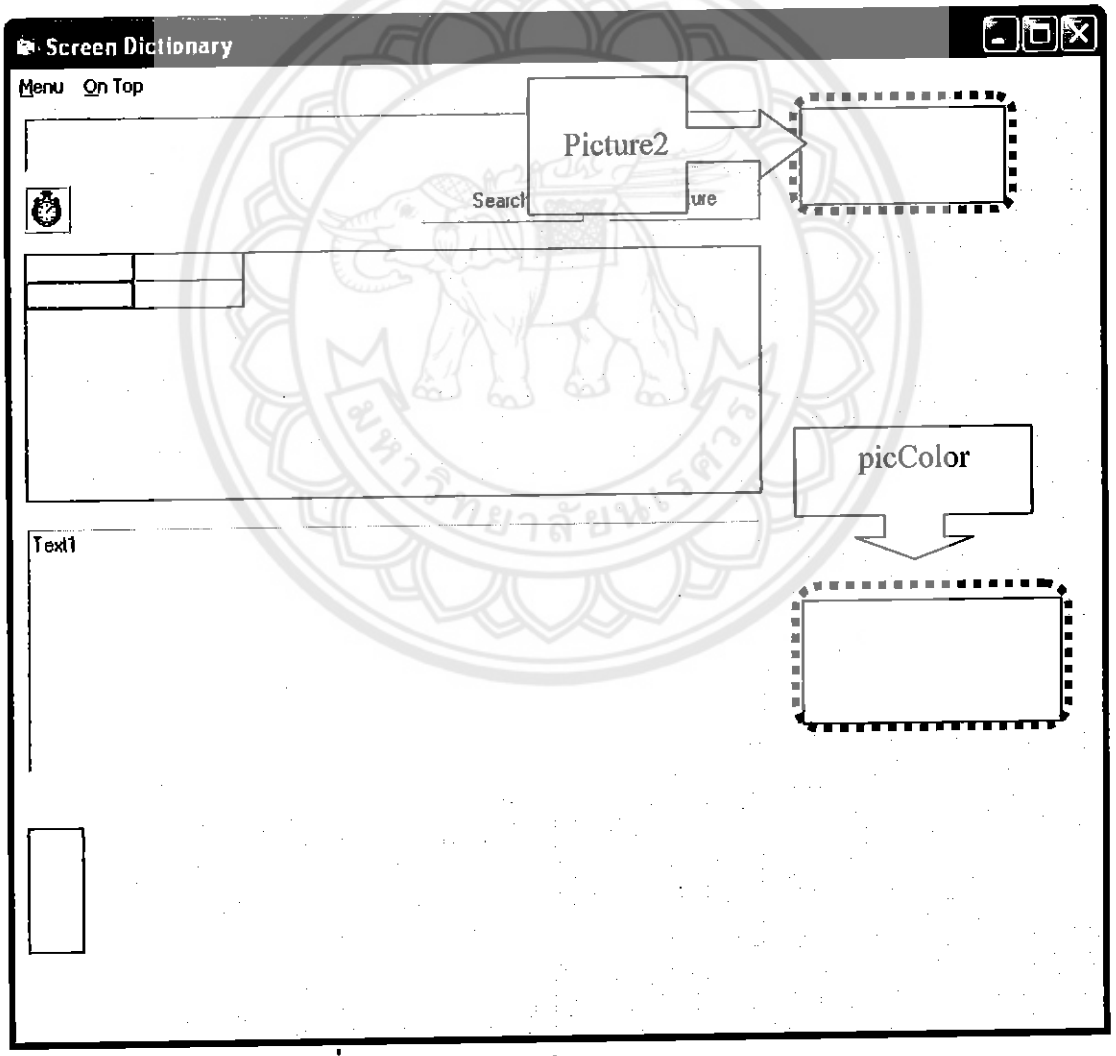
เมื่อนำภาพมาเก็บไว้ใน Picture2 แล้วจะทำการเซตภาพโดยใช้ชื่อว่า pic.bmp เพื่อให้เห็นขั้นตอนชัดเจนมากยิ่งขึ้นหลังจากนั้นจะทำการโหลดรูปจาก Path ที่เก็บรูปภาพมาเพื่อทำภาพขาวดำ (Monochrome) โดยใช้ฟังก์ชัน SetBitMap กับ MakeToGray เมื่อได้รูปขาวดำแล้วจะทำการบันทึกรูปอีกครั้ง ให้เป็น

pic2.bmp และจะนำไปเก็บไว้ใน Picture Box ที่ชื่อ picColor ซึ่งอยู่ใน MDI Form สามารถกำหนดคุณสมบัติของ picColor ได้ดังนี้

picColor ใน Tab Alphabetic

- Name : piccolo
- Appearance : 0 – Flat
- AutoRedraw : Ture
- AutoSize : Ture
- ScaleMode : 3 – Pixel

ในกรณีของ picColor เป็นกรณีเดียวกับ Picture2 คือผู้พัฒนาจำเป็นต้องเห็นขั้นตอนการ Run โปรแกรมจึงต้องคุณสมบัติ Visible เป็น True



รูปที่ ข.36 แสดงตำแหน่งของ Picture2 และ picColor

หลังจากที่ได้ภาพขาวดำแล้วโปรแกรมจะนำภาพที่ได้ไปวิเคราะห์เพื่อตัดคำโดยเริ่มจากฟังก์ชัน StartAnalystPicX, ClearIndexPic, StartVertical ตามลำดับ จากนั้นจะนำคำที่ต้องการไปเข้าสู่กระบวนการ OCR โดยใช้ฟังก์ชัน ODRFuntion เมื่อได้ข้อความมาแล้วจึงนำเอาข้อความนั้นไปหาความหมายใน Data Base โดยใช้ฟังก์ชัน SearchDic เสร็จแล้วจึงทำการ Unload Form Child และ Form1 เป็นอันเสร็จสิ้นฟังก์ชัน Capture สามารถเขียน Code ได้ดังนี้

```
Private Sub capture(xStart As Single, yStart As Single)
Timer2.Enabled = False
' สร้าง Form Child
   Dim frmChild As New frmChild
   frmChild.Show
   frmChild.Picture1.Visible = False
' กำหนดขนาดขอบเขตของการตัดภาพ
   With MDIForm1.ActiveForm.Picture1
      .BackColor = &HFFFFFF
      .Cls
      .Width = 300
      .Height = 200
      MDIForm1.ActiveForm.Width = Screen.TwipsPerPixelX * (xStart)
      MDIForm1.ActiveForm.Height = Screen.TwipsPerPixelY * (yStart)
' คำสั่งวาดรูป
      MDIForm1.ActiveForm.Picture1.PaintPicture Form1.Picture, 0, 0, , , xStart - 150, yStart -
100, 300, 200
      .Visible = True
   End With
' ส่งรูปจาก Form Chlid ไปยัง Picture2 ที่อยู่ใน MDI Form
   MDIForm1.Picture2.Picture = MDIForm1.ActiveForm.Picture1.Image
' สร้าง Path ในการเก็บรูป
   Path = App.Path & "\TempPic\pic.bmp"
' บันทึกรูปลงใน Path ที่สร้างไว้โดยใช้ชื่อ pic.jpg
   SavePicture MDIForm1.Picture2, Path
' ทำการโหลดภาพขึ้นมาใหม่เพื่อที่จะนำไปแปลงเป็นภาพขาวดำ
```

```
    MDIForm1.picColor.Picture = LoadPicture(App.Path & "\TempPic\pic.bmp")
' ฟังก์ชันทำภาพขาวดำ
    SetBitMap MDIForm1.picColor
    MakeToGray MDIForm1.picColor
' ฟังก์ชันในการวิเคราะห์ตัวอักษร
    MDIForm1.StartAnalystPicX
    MDIForm1.ClearIndexPic
    MDIForm1.StartVertical
' ฟังก์ชันการทำ OCR
    MDIForm1.ODRFuntion
' ฟังก์ชันในการค้นหาคำศัพท์
    MDIForm1.SearchDic
    Unload frmChild
    Unload Me
    Exit Sub
errMouseUp:
    MsgBox Err.Description & ": Error number " & Err.Number
End Sub
```

ฟังก์ชัน SetBitMap

เป็นฟังก์ชันในการ Scan รูปภาพไปเก็บไว้ใน Bitmap เพื่อรอที่จะนำไปแปลงให้กลายเป็นภาพขาวดำอีกทีหนึ่งสามารถเขียน Code ได้ดังนี้

```
Private Sub SetBitMap(ByVal picColor As PictureBox)
' Prepare the bitmap description.
    With bitmap_info.bmiHeader
        .biSize = 40
        .biWidth = picColor.ScaleWidth
' Use negative height to scan top-down.
        .biHeight = -picColor.ScaleHeight
        .biPlanes = 1
        .biBitCount = 32
        .biCompression = BI_RGB
```

```
    bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
    pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
    .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
  End With
End Sub
```

ฟังก์ชัน MakeToGray

ฟังก์ชันนี้จะทำหน้าที่ตรวจสอบ ฟังก์ค่าสีไหนควรจะเป็นสีขาวค่าสีไหนควรจะเป็นสีดำซึ่งผู้พัฒนาได้กำหนดให้ค่าสีที่น้อยกว่า 192 ให้เป็นสีดำส่วนค่าสีที่มีค่าตั้งแต่ 192 ขึ้นไปจะกำหนดให้เป็นสีขาวสามารถเขียน Code ได้ดังนี้

```
Private Sub MakeToGray(ByVal picColor As PictureBox)
' Load the bitmap's data.
  ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
  GetDIBits picColor.hdc, picColor.Image, _
    0, picColor.ScaleHeight, pixels(1, 1, 1), _
    bitmap_info, DIB_RGB_COLORS
' Modify the pixels.
  For y = 1 To picColor.ScaleHeight
    For x = 1 To picColor.ScaleWidth
      ave_color = CByte((CInt(pixels(pixR, x, y)) + _
        pixels(pixG, x, y) + _
        pixels(pixB, x, y)) \ 3)
      If ave_color < 192 Then
        ave_color = 0
      Else
        ave_color = 255
      End If
      pixels(pixR, x, y) = ave_color
      pixels(pixG, x, y) = ave_color
      pixels(pixB, x, y) = ave_color
    Next x
  Next y
```

```
' Display the result.
    SetDIBits picColor.hdc, picColor.Image, _
        0, picColor.ScaleHeight, pixels(1, 1, 1), _
        bitmap_info, DIB_RGB_COLORS
    picColor.Picture = picColor.Image
' เมื่อได้ภาพขาวดำแล้วก็นำไปเก็บไว้ใน Path ที่กำหนดโดยใช้ชื่อว่า pic2.bmp
    Path = App.Path & "\TempPic\pic2.bmp"
    SavePicture MDIForm1.picColor, Path
End Sub
```

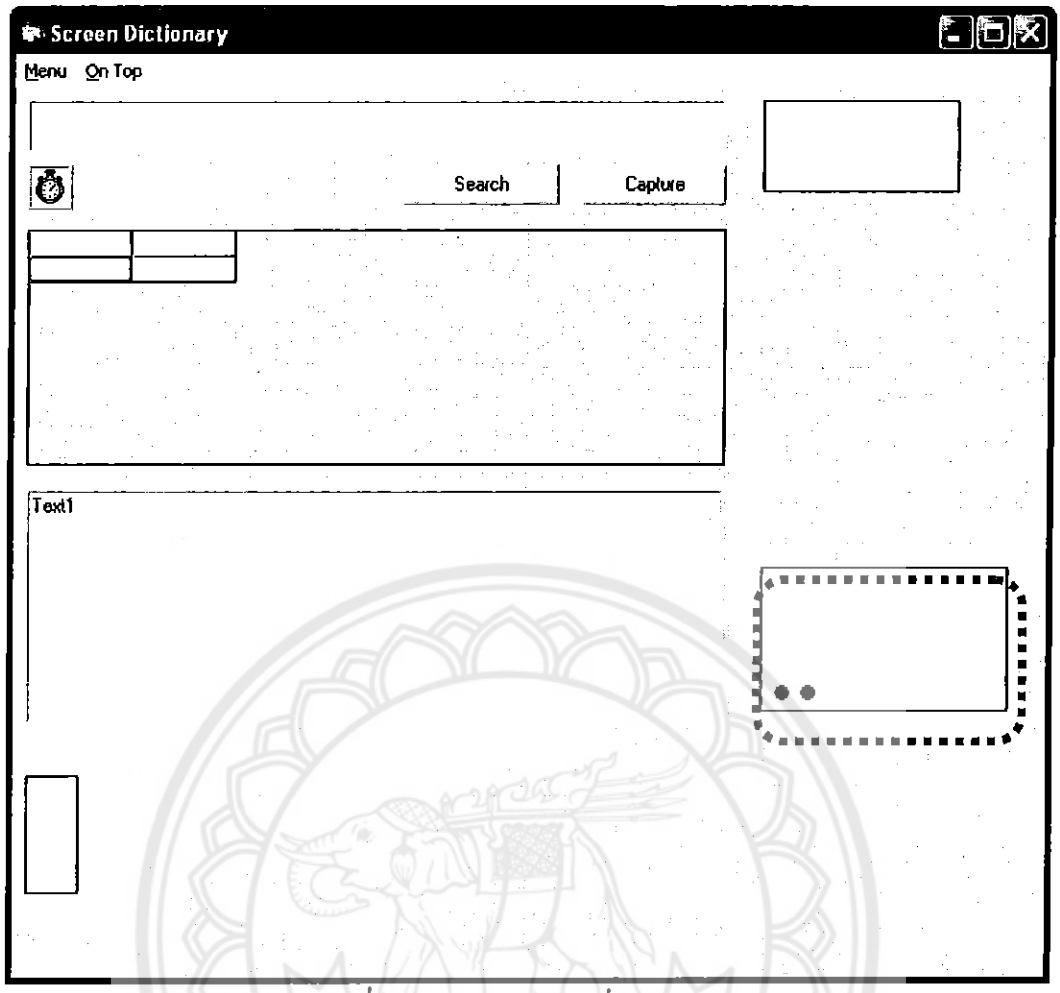
ฟังก์ชัน StartAnalystPicX

เป็นฟังก์ชันหาตำแหน่งจุดเริ่มต้นของการสแกนภาพของคู่ลำดับ x ซึ่งมีฟังก์ชันที่น่าสนใจคือ GetPixel เป็นฟังก์ชันที่รับค่าของ Pixel ในคู่ลำดับนั้นๆ และมี Tool Box ใหม่เพิ่มขึ้นมานั้นก็คือ Shape ทำไว้เพื่อตรวจสอบขั้นตอนการทำงานของฟังก์ชันว่าฟังก์ชันได้ทำตามความต้องการของผู้พัฒนาหรือไม่ Shape สามารถทำได้โดยเลือก  แล้วนำไปวาดใน Picture Box ที่ชื่อ picColor และกำหนดคุณสมบัติดังต่อไปนี้

ใน Tab Alphabetic

- BackColor : กำหนดสีอะไรก็ได้ ควรเลือกที่เป็นสีต่างกันเพื่อง่ายต่อการพิจารณา
- BackStyle : 1 – Opaque
- Index : 0 ' เพื่อกำหนดให้เป็น Array
- Shape : 3 – Circle

ในส่วนของฟังก์ชันนี้จะใช้ Shape 2 ตัว



รูปที่ ข.37 แสดง Shape ที่อยู่ใน picColor

ฟังก์ชันนี้สามารถเขียน Code ได้เป็น

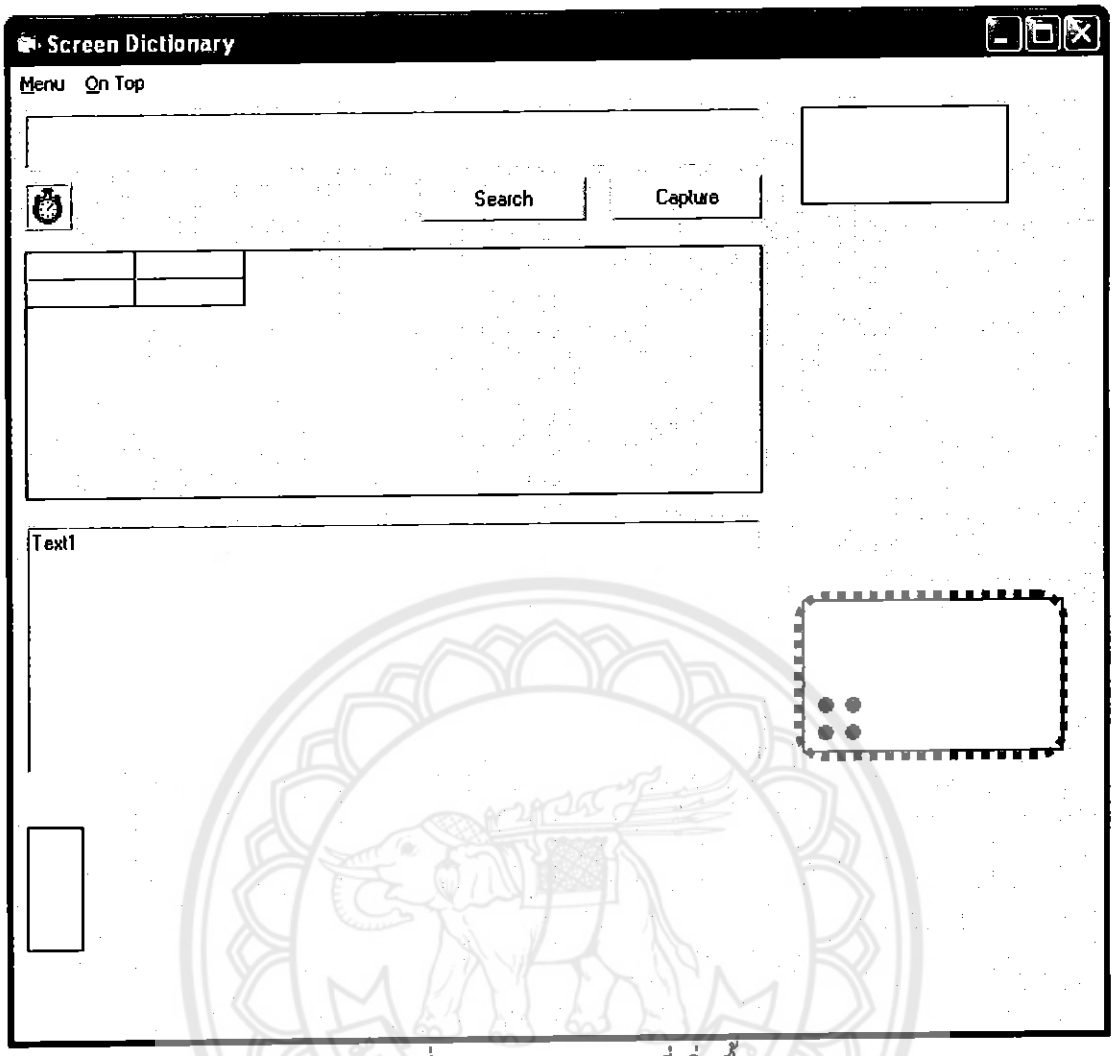
```
Public Sub StartAnalystPicX()
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2 As Integer
Dim Object1_Y1, Object1_Y2 As Integer
Dim cont1, cont2 As Integer
Qty = -1
object = False
picColor.Picture = LoadPicture(App.Path & "\TempPic\pic2.bmp")
  For i = picColor.ScaleWidth / 2 To 0 Step -1
    For j = picColor.ScaleHeight / 2 To picColor.ScaleHeight
```

```
Colour = GetPixel(picColor.hdc, i, j)
If Colour <> &HFFFFFF And Colour <> -1 Then
    cont1 = 0
    cont2 = 0
    If object = False Then
        Object1_X1 = i
        Object1_Y1 = j
        object = True
        Qty = Qty + 1
        If Qty = 0 Then
         Shape1(Qty).Visible = True
         Shape1(Qty).Left = Object1_X1 - 5
         Shape1(Qty).Top = Object1_Y1
        Else
         Load Shape1(Qty)
         Shape1(Qty).Visible = True
         Shape1(Qty).Left = Object1_X1 - 5
         Shape1(Qty).Top = Object1_Y1
        End If
    End If
  Exit For
End If
  If object = True Then
    If j = picColor.ScaleHeight Then
      cont1 = cont1 + 1
      If cont1 = 5 Then
        Object1_X2 = i
        Object1_Y2 = j
        object = False
       If Qty = 0 Then
        Shape2(Qty).Visible = True
        Shape2(Qty).Left = Object1_X2 - 5
```

```
            Shape2(Qty).Top = Object1_Y1
            StartAnalystPicY i, 0
            Dim Z As Integer
            Z = 1
            Exit For
          End If
        End If
      End If
    End If
  Next j
  If Z = 1 Then Exit For
 Next i
End Sub
```

ฟังก์ชัน StartAnalystPicY

เป็นที่รับค่า x มาจากฟังก์ชัน StartAnalystPicX อีกทีหนึ่งฟังก์ชันเป็นฟังก์ชัน**หาจุดเริ่มต้นของการสแกนภาพ**ของคู่ลำดับ y ซึ่งจะใช้ฟังก์ชัน GetPixel เช่นเดียวกับฟังก์ชัน StartAnalystPicX **และฟังก์ชันก็จะใช้** Shap อีก 2 ตัวเพื่อแสดงตำแหน่งเริ่มต้นสแกนของคู่ลำดับ y



รูปที่ ข. 38 แสดง Shape ที่เพิ่มขึ้นมา

ฟังก์ชันนี้สามารถเขียน Code ได้เป็น

```
Private Sub StartAnalystPicY(ii As Long, jj As Long)
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2 As Integer
Dim Object1_Y1, Object1_Y2 As Integer
Qty = -1
object = False

 For j = picColor.ScaleHeight / 2 To 0 Step -1
  For i = ii To picColor.ScaleWidth
```

```
Colour = GetPixel(picColor.hdc, i, j)
If Colour <> &HFFFFFF And Colour <> -1 Then
 If object = False Then
    Object1_X1 = i
    Object1_Y1 = j
    object = True
    Qty = Qty + 1
    If Qty = 0 Then
    Shape3(Qty).Visible = True
    Shape3(Qty).Left = Object1_X1 - 3
    Shape3(Qty).Top = Object1_Y1 - 3
    Else
    Load Shape3(Qty)
    Shape3(Qty).Visible = True
    Shape3(Qty).Left = Object1_X1 - 3
    Shape3(Qty).Top = Object1_Y1 - 3
    End If
 End If
 Exit For
End If
 If object = True Then
   If i = picColor.ScaleWidth Then
     Object1_X2 = i
     Object1_Y2 = j
     object = False

      If Qty = 0 Then
    Shape4(Qty).Visible = True
    Shape4(Qty).Left = Object1_X1 - 3
    Shape4(Qty).Top = Object1_Y2 - 3
     StepHorizontal ii, j
     Dim Z As Integer
```

```
            Z = 1
            Exit For
          End If
        End If
      End If
    Next i
      If Z = 1 Then Exit For
  Next j
End Sub
```

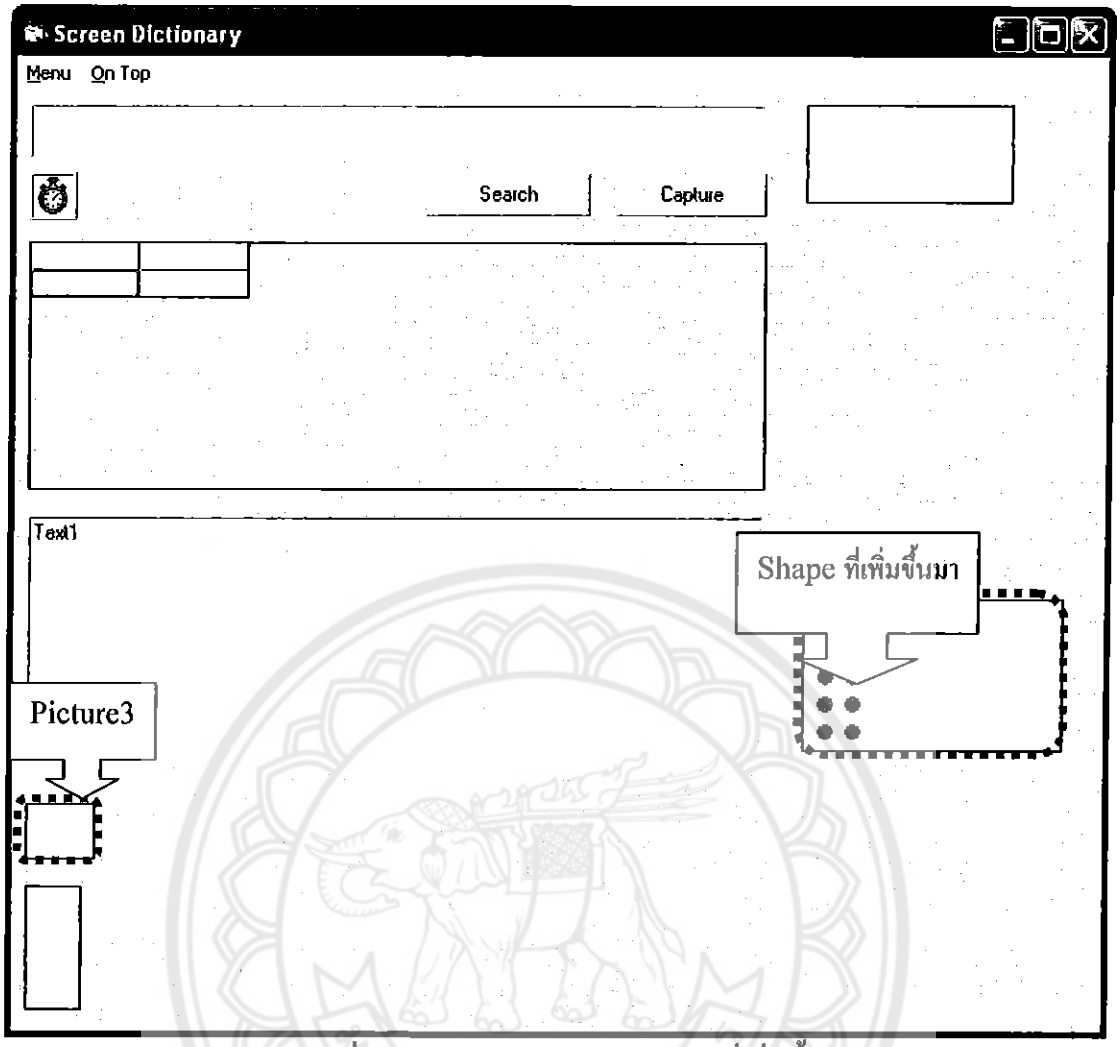
ฟังก์ชัน StepHorizontal

หลังจากได้คู่ลำดับเริ่มต้นสแกน (x,y) จากฟังก์ชัน StartAnalystPicX และ StartAnalystPicY แล้วในฟังก์ชันนี้จะรับค่า (x,y) มาสแกนหาแถวของคำที่ต้องการแล้วตัดออกในฟังก์ชันนี้นอกจากฟังก์ชัน GetPixel แล้วยังมีฟังก์ชัน ResizePic2, showFovea2 เพิ่มเข้ามา โดยที่ฟังก์ชัน ResizePic2 จะเป็นการกำหนดขนาดของกล่องให้พอดีกบรูปภาพที่ส่งมา ส่วนฟังก์ชัน showFovea2 เป็นการวาดรูปลงบนกล่องนั้นๆซึ่งเมื่อได้แถวที่ต้องการแล้วจะนำแถวที่ได้ไปเก็บไว้ใน Picture3 และทำการบันทึกรูปภาพนั้นลงบน Path ที่กำหนดโดยใช้ชื่อว่า pic3.bmp การกำหนดคุณสมบัติของ Picture3 สามารถกำหนดได้ดังนี้

ใน Tab Alphabetic

- Appearance : 0 – Flat
- AutoRedraw : True
- AutoSize : True
- ClipControls : False
- Index : 0

และในฟังก์ชันนี้จะใช้อีก 2 ตัวดังรูปที่ ข.38



รูปที่ ข.39 แสดง Picture3 กับ Shape ที่เพิ่มขึ้นมา

ฟังก์ชันนี้สามารถเขียน Code ได้เป็น

```
Private Sub StepHorizontal(ii As Long, jj As Long)
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2 As Integer
Dim Object1_Y1, Object1_Y2 As Integer
Dim ObjectHeight As Integer
Dim Convert As Integer
Qty = -1
object = False
```

```
For j = jj To picColor.ScaleHeight
 For i = ii To picColor.ScaleWidth


  Colour = GetPixel(picColor.hdc, i, j)
  If Colour <> &HFFFFFF And Colour <> -1 Then
   If object = False Then
       Object1_X1 = i
       Object1_Y1 = j
       object = True
       Qty = Qty + 1
       If Qty = 0 Then
       Shape5(Qty).Visible = True
       Shape5(Qty).Left = Object1_X1 - 3
       Shape5(Qty).Top = Object1_Y1 - 3
       Else
       Load Shape5(Qty)
       Shape5(Qty).Visible = True
       Shape5(Qty).Left = Object1_X1 - 3
       Shape5(Qty).Top = Object1_Y1 - 3
       End If
   End If
   Exit For
  End If
   If object = True Then
    If i = picColor.ScaleWidth Then
       Object1_X2 = i
       Object1_Y2 = j
       object = False


      If Qty = 0 Then
      Shape6(Qty).Visible = True
      Shape6(Qty).Left = Object1_X1 - 3
```

```
            Shape6(Qty).Top = Object1_Y2 - 3
            ResizePic2 Picture3(Qty), ObjectHeight, Convert, Object1_Y1, Object1_Y2
            showFovea2 picColor, Picture3(Qty), Object1_Y1, Object1_Y2
            Picture3(0).Refresh
            Path = App.Path & "\TempPic\pic3.bmp"
            SavePicture Picture3(Qty).Image, Path
            Dim x As Integer
            x = 1
            Exit For
            End If
          End If
        End If
      Next i
      If x = 1 Then Exit For
    Next j
End Sub
```

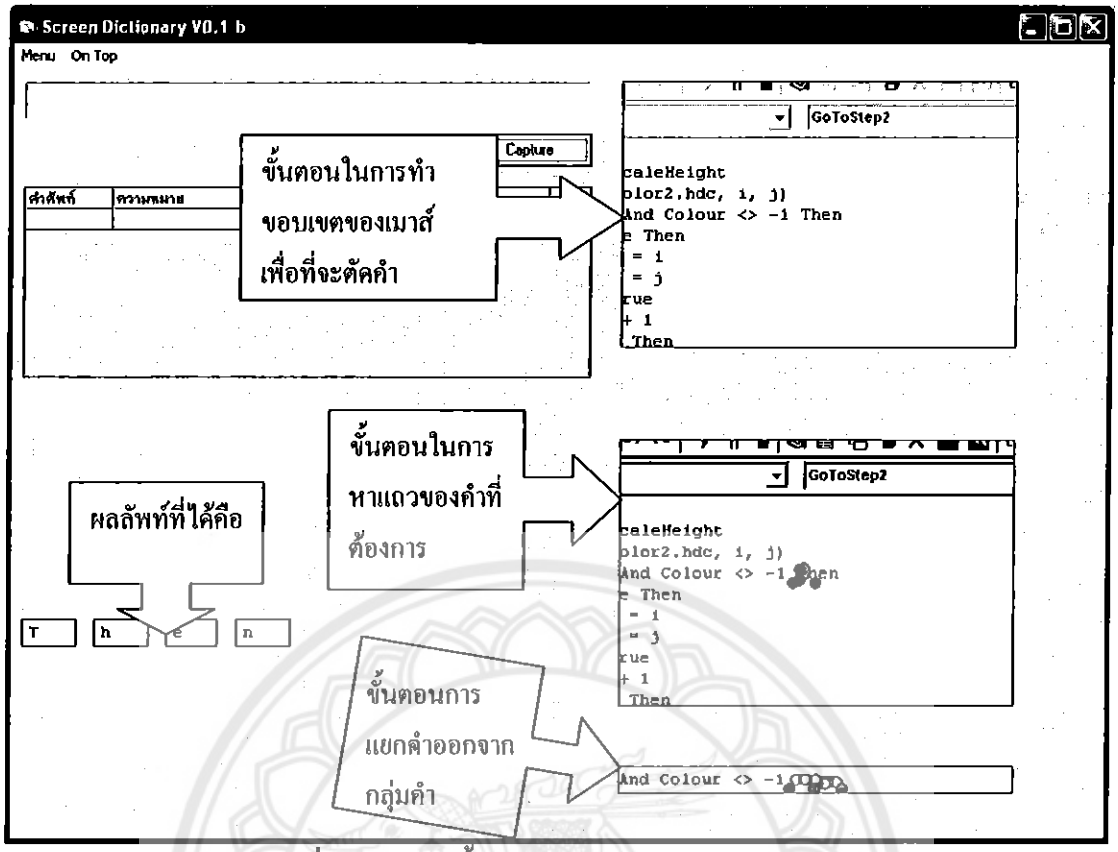
จากนั้นจะทำการ Clear ค่า Index ใน Piture และ Shape โดยใช้ฟังก์ชัน ClearIndexPic

ฟังก์ชัน StartVertical

เป็นฟังก์ชันที่ทำหน้าที่ตัดแยกกลุ่มคำที่ต้องการออกจากประโยค จากนั้นก็แยกคำออกจากกลุ่มคำซึ่งมี 3 ฟังก์ชันด้วยกันคือ

- StartVertical ขั้นตอนนี้จะได้ค่าขอบซ้ายมา
- GoToStep2 ขั้นตอนนี้จะได้ค่าขอบขวามา
- GoToStep3 ขั้นตอนนี้จะทำการสแกนจากขอบซ้ายไปสิ้นสุดที่ขอบขวา

เมื่อเสร็จ 3 ขั้นตอนนี้จะได้คำที่แยกออกจากกันโดยเอาคำนี้ไปเก็บไว้ใน Picture4 ซึ่งเป็น Picture ชนิด Array หลังจากนั้นก็ทำการบันทึกรูปทั้งหมดไว้ใน Path ที่กำหนด



รูปที่ ข.40 แสดงขั้นตอนการทำงานของฟังก์ชัน Capture

ฟังก์ชัน OCRFuntion

เมื่อได้คำที่ต้องการแล้วต่อไปจะนำคำที่ได้ไปเข้ากระบวนการ OCR โดยที่จะนำรู้ที่อยู่ใน Picture Array ออกมาทำ OCR ทีละตัวจากนั้นจะคำที่ได้ไปวางไว้ใน txtSearch เพื่อทำการค้นหาความหมายต่อไป สามารถเขียน Code ได้ดังนี้

```
Public Sub ODRFuntion()
Dim test As Integer
Dim MyLen
For test = 0 To Qty
' ดึงรูปออกมาทีละภาพ
    Path = App.Path & "\TempPic\OCRPic" & test & ".bmp"
' เมื่อได้คำมาแล้วให้ตัดช่องว่างออกด้วยแล้วไปเก็บไว้ใน Text2
    Text2.Text = Trim(OCR(Path, -1))
    Text2.Text = Left$(Text2, 1)
' กำหนดให้ Cursor อยู่ท้ายสุดของประโยคเสมอ
    Text3.SelStart = Len(Text3.Text)
```

```
' ป้อนคำจาก Text2 ไปยัง Text3
    Text3.SelText = Trim(Text2.Text)
' แปลงให้เป็นตัวเล็กทั้งหมด
    Text3 = StrConv(Text3.Text, vbLowerCase)
' ตัดช่องว่างทั้งซ้ายและขวา
    MyLen = Left$(Text3, Len(Trim(Text3)) + 1)
    MyLen = Right$(MyLen, Len(Trim(MyLen)))
' นำกลุ่มคำที่ได้ไปใส่ไว้ใน txtSearch
    MDIForm1.txtSearch.Text = MyLen
Next
End Sub
```

ฟังก์ชัน SearchDic

เมื่อเสร็จสินกระบวนการ OCR แล้วโปแกรมจะทำการ Search เพื่อหาความหมายของคำนั้น โดยการเข้าสู่ฟังก์ชัน SearchDic ในฟังก์ชันนี้จะทำการตรวจสอบก่อนว่ามีกลุ่มคำอยู่ใน txtSearch หรือยังหากยังก็จะไม่ทำการค้นหาแต่ถ้าหากมีก็จะเรียกใช้ฟังก์ชัน FilePath เพื่อบอกที่อยู่ของ Database เมื่อรู้ที่อยู่ของ Database แล้วก็จำทำการ Select ข้อมูลจาก Database ทั้งหมดมาก่อนโดยใช้ฟังก์ชัน ShowData จากนั้นทำการกำหนดตั้งชื่อของ คอลัมน์ ขนาดของตาราง และจำนวนอักษรที่แสดงในตารางโดยใช้ฟังก์ชัน OrderGrid จากนั้นทำคำที่อยู่ใน txtSesrch มา select หาอีกครั้งก็จะได้ความหมายของคำนั้นๆ

ภาคผนวก ค

All source code of program to following :

```
 ///Form1(ScreenGrab).frm///
'From1 is create for Screen Grab in computer
'Define image size and convert image from color to gray scale
'And detect sub image character become to text character
Option Explicit
Dim xStart As Single, yStart As Single
Dim CheckEnd As Integer
Dim pixels() As Byte
Dim bytes_per_scanLine As Integer
Dim pad_per_scanLine As Integer
Dim x As Integer
Dim y As Integer
Dim cont, test As Integer
Dim ave_color2 As Byte
Dim Path, Text As String
Public Qty, Qty2, Qty3, Qty4 As Integer
Public Colour As Double
Private Sub Form_Unload(Cancel As Integer)
    MDIForm1.Visible = True
End Sub
Private Sub capture(xStart As Single, yStart As Single)
Timer2.Enabled = False
 '   On Error GoTo errMouseUp
    'create new child forms of MDI
    Dim frmChild As New frmChild
    frmChild.Show
    frmChild.Picture1.Visible = False
    With MDIForm1.ActiveForm.Picture1
        .BackColor = &HFFFFFF
        .Cls
        .Width = 300
        .Height = 200
     'systemmetrics 26= caption and menubar;8= 3d borders of form
       MDIForm1.ActiveForm.Width =Screen.TwipsPerPixelX*(xStart)
       MDIForm1.ActiveForm.Height=Screen.TwipsPerPixelY*(yStart)
       MDIForm1.ActiveForm.Picture1.PaintPicture Form1.Picture,
          0, 0, , , xStart - 150, yStart - 100, 300, 200
        .Visible = True
    End With
    'unload me?
    'convert picture
    MDIForm1.Picture2.Picture =
MDIForm1.ActiveForm.Picture1.Image
    Path = App.Path & "\TempPic\pic.bmp"
    SavePicture MDIForm1.Picture2, Path
    MDIForm1.picColor.Picture = LoadPicture(App.Path &
"\TempPic\pic.bmp")
    SetBitMap MDIForm1.picColor
    MakeToGray MDIForm1.picColor
```

```
    MDIForm1.StartAnalystPicX
    MDIForm1.ClearIndexPic
    MDIForm1.StartVertical
    MDIForm1.ODRFuntion
    MDIForm1.SearchDic
    Unload frmChild
    Unload Me
    Exit Sub
errMouseUp:
    MsgBox Err.Description & ": Error number " & Err.Number
End Sub

Private Sub Timer2_Timer()
Static bbX As Long
Static bbY As Long
Dim pt As POINTAPI
Call GetCursorPos(pt)
If ((bbX <> pt.x) And (bbY <> pt.y)) Then
        CheckEnd = 0
Else
        CheckEnd = (CheckEnd + 1)
        If (CheckEnd > 0) Then
        xStart = pt.x: yStart = pt.y
        capture xStart, yStart
        End If
End If
        bbX = pt.x
        bbY = pt.y
End Sub

' Convert a color image to gray scale.
Private Sub SetBitMap(ByVal picColor As PictureBox)
    ' Prepare the bitmap description.
    With bitmap_info.bmiHeader
        .biSize = 40
        .biWidth = picColor.ScaleWidth
        ' Use negative height to scan top-down.
        .biHeight = -picColor.ScaleHeight
        .biPlanes = 1
        .biBitCount = 32
        .biCompression = BI_RGB
        bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \
32) * 4)
        pad_per_scanLine = bytes_per_scanLine - (((.biWidth *
.biBitCount) + 7) \ 8)
        .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
    End With
End Sub

Private Sub MakeToGray(ByVal picColor As PictureBox)

        ' Load the bitmap's data.
    ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To
          picColor.ScaleHeight)
    GetDIBits picColor.hdc, picColor.Image, _
        0, picColor.ScaleHeight, pixels(1, 1, 1), _
```

```
        bitmap_info, DIB_RGB_COLORS


    ' Modify the pixels.
    For y = 1 To picColor.ScaleHeight
        For x = 1 To picColor.ScaleWidth
            ave_color = CByte((CInt(pixels(pixR, x, y)) + _
                pixels(pixG, x, y) + _
                pixels(pixB, x, y)) \ 3)
            If ave_color < 192 Then
                ave_color = 0
            Else
                ave_color = 255
            End If
            pixels(pixR, x, y) = ave_color
            pixels(pixG, x, y) = ave_color
            pixels(pixB, x, y) = ave_color
        Next x
    Next y

    ' Display the result.
    SetDIBits picColor.hdc, picColor.Image, _
        0, picColor.ScaleHeight, pixels(1, 1, 1), _
        bitmap_info, DIB_RGB_COLORS
    picColor.Picture = picColor.Image
    Path = App.Path & "\TempPic\pic2.bmp"
    SavePicture MDIForm1.picColor, Path
End Sub
```

```
///MDIFORM.frm///
'This form will show application in user interface
'include of another button and feather
Option Explicit
Dim Conn As New ADODB.Connection
Dim rsCustomers As New ADODB.Recordset, rsClone As New
ADODB.Recordset
Dim strConn As String
Dim num As Long
Dim Path, Text As String
Public Qty, Qty2, Qty3, Qty4, Qty5, Qty6, Qty7, Qty8, Qty9,
           Qty10, Qty11, Qty12 As Integer
Public Colour As Double

Function GetScreen(Optional ByVal hwnd As Long) As IPictureDisp
    Dim targetDC As Long
    Dim hdc As Long
    Dim tempPict As Long
    Dim oldPict As Long
    Dim wndWidth As Long
    Dim wndHeight As Long
    Dim Pic As PICTDESC
    Dim rcWindow As RECT
    Dim guid(3) As Long
    If hwnd = 0 Then hwnd = GetDesktopWindow

    ' get window's size
    GetWindowRect hwnd, rcWindow
    wndWidth = rcWindow.Right - rcWindow.Left
    wndHeight = rcWindow.Bottom - rcWindow.Top

    ' get window's device context
    targetDC = GetWindowDC(hwnd)

    ' create a compatible DC
    hdc = CreateCompatibleDC(targetDC)


    ' create a memory bitmap in the DC just created
    ' the has the size of the window we're capturing
    tempPict = CreateCompatibleBitmap(targetDC, wndWidth,
                wndHeight)
    oldPict = SelectObject(hdc, tempPict)

    ' copy the screen image into the DC
    BitBlt hdc, 0, 0, wndWidth, wndHeight, targetDC, 0, 0,
vbSrcCopy

    ' set the old DC image and release the DC
    tempPict = SelectObject(hdc, oldPict)
    DeleteDC hdc
    ReleaseDC GetDesktopWindow, targetDC
```

```
    ' fill the ScreenPic structure
    With Pic
        .cbSize = Len(Pic)
        .pictType = 1           ' means picture
        .hIcon = tempPict
        .hPal = 0           ' (you can omit this of course)
    End With
    ' convert the image to a IpictureDisp object
    ' this is the IPicture GUID {7BF80980-BF32-101A-8BBB-
00AA00300CAB}
    ' we use an array of Long to initialize it faster
    guid(0) = &H7BF80980
    guid(1) = &H101ABF32
    guid(2) = &HAA00BB8B
    guid(3) = &HAB0C3000
    ' create the picture,
    ' return an object reference right into the function result
    OleCreatePictureIndirect Pic, guid(0), True, GetScreen
End Function

Private Sub capture_Click()
    ClearIndexPic
    txtSearch.Text = " "
    Text3.Text = " "
    Picture3(0) = Nothing
    Picture4(0) = Nothing
    MDIForm1.Visible = False
    Form1.Visible = False

    'start main grab sequence
    Timer1.Enabled = True
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
    Call SetWindowLong(hwnd, GWL_WNDPROC, g_nProcOld)
        If debugme = True Then MsgBox "unload event"
        If debugme = True Then MsgBox MDIForm1.ActiveForm Is
Nothing
        Unload Form1
        Unload Form2
        Unload Me
End Sub

Private Sub MDIForm_Load()
  If txtSearch.Text <> "" Then
        FilePath
        ShowData
        OrderGrid
            Set rsClone = rsCustomers.Clone
        With rsClone
            .Filter = "Eentry LIKE '" & txtSearch.Text & "%'"
            Set mfgCustomers.DataSource = rsClone
        End With
    OrderGrid
    Set rsClone = Nothing
    End If
```

```
    OrderGrid
End Sub

Private Sub mnuExit_Click()
    'child forms take care of themselves
    Unload Form1
    Unload Form2
    Unload Me
End Sub

Private Sub mnuGrabText_Click()
    MDIForm1.Visible = False
    Form1.Visible = False
    'start main grab sequence
    Timer1.Enabled = True
End Sub

Private Sub mnuHelp_Click()
Form2.Visible = True
Form2.Label1 = "            " & vbCrLf & _
               "            " & vbCrLf & _
               "      Get Text Project by" & vbCrLf & _
               "           " & vbCrLf & _
               "              ID    :46380033" & vbCrLf & _
               "              Name  :Mr.Pichit Thammawong " &
vbCrLf & _
               "              E-mail:u46380033@nu.com " & vbCrLf &
_
               "           " & vbCrLf & _
               "              ID    :46380191" & vbCrLf & _
               "              Name  :Mr.Peerapol Nokphuak " &
vbCrLf & _
               "              E-mail:u46380191@nu.com " & vbCrLf &
_
               "           " & vbCrLf & _
               "           "
End Sub

Private Sub mnuNo_Click()
    mnuYes.Checked = False
    mnuNo.Checked = True

    'Unset Form's on top
    SetWindowPos hwnd, HWND_NOTOPMOST, 0, 0, 0, 0, SWP_NOMOVE +
SWP_NOSIZE
End Sub

Private Sub mnuYes_Click()
    mnuNo.Checked = False
    mnuYes.Checked = True

    'set Form's on top
    SetWindowPos hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE +
SWP_NOSIZE
End Sub
```

```
Private Sub Timer1_Timer()
    Static count
    'timer on 500 interval
    If count >= 0 Then
        Form1.Picture = GetScreen(0)
        Timer1.Enabled = False
        Form1.Visible = True
    End If
End Sub
Private Sub ShowData()
    With Conn
        If .State = adStateOpen Then .Close
        .ConnectionString = strConn
        .ConnectionTimeout = 90
        .Open
    End With
    Dim sqlCustomers As String
    sqlCustomers = "SELECT Eentry, Tentry"
    sqlCustomers = sqlCustomers & " FROM DIC_DB "
    sqlCustomers = sqlCustomers & " ORDER BY Eentry "
    With rsCustomers
        If .State = adStateOpen Then .Close
        .ActiveConnection = Conn
        .CursorType = adOpenForwardOnly
        .CursorLocation = adUseClient
        .Open sqlCustomers
           If .RecordCount <> 0 Then
                Set mfgCustomers.DataSource = rsCustomers
           End If
    End With
End Sub

Private Sub OrderGrid()
    With mfgCustomers
        .TextMatrix(0, 0) = "คำศัพท์"
        .TextMatrix(0, 1) = "ความหมาย"
        .ColWidth(0) = 1000
        .ColWidth(1) = 5000
    End With
End Sub

Private Sub mfgCustomers_Click()
    Text1.Visible = True
    Text1.Text = mfgCustomers.TextMatrix(mfgCustomers.RowSel,
mfgCustomers.ColSel)
End Sub

Private Sub cmdSearch_Click()
Dim rsClone As New ADODB.Recordset
        If txtSearch.Text = "" Then Exit Sub
        FilePath
        ShowData
        OrderGrid
        Set rsClone = rsCustomers.Clone
        Dim Tdata  As String
```

```
        Tdata = Trim(txtSearch.Text)
        With rsClone
            .Filter = "Eentry LIKE '" & Tdata & "%'"
            Set mfgCustomers.DataSource = rsClone
        End With
    OrderGrid
    Set rsClone = Nothing
End Sub

Private Sub FilePath()
   Dim DB_File As String
        DB_File = App.Path
        If Right$(DB_File, 1) <> "\" Then DB_File = DB_File & "\"
        DB_File = DB_File & "DIC_DB.MDB"
        strConn = "Provider = Microsoft.Jet.OLEDB.4.0;" & _
                          "Data Source = " & DB_File & ";" & _
                          "Persist Security Info = False"
End Sub

Public Sub SearchDic()
Dim rsClone As New ADODB.Recordset
        If txtSearch.Text = "" Then Exit Sub
        FilePath
        ShowData
        OrderGrid
        Set rsClone = rsCustomers.Clone
        Dim Tdata  As String
        Tdata = Trim(txtSearch.Text)
        With rsClone
            .Filter = "Eentry LIKE '" & Tdata & "%'"
            Set mfgCustomers.DataSource = rsClone
        End With
    OrderGrid
    Set rsClone = Nothing
End Sub

Public Sub StartAnalystPicX()
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2, Object1_X3, Object1_X4 As Integer
Dim Object1_Y1, Object1_Y2, Object1_Y3, Object1_Y4 As Integer
Dim ObjectWidth As Integer
Dim Convert As Integer
Dim cont1, cont2 As Integer
Qty = -1
Qty2 = -1
object = False

picColor.Picture = LoadPicture(App.Path & "\TempPic\pic2.bmp")
  For i = picColor.ScaleWidth / 2 To 0 Step -1
    For j = picColor.ScaleHeight / 2 To picColor.ScaleHeight
      Colour = GetPixel(picColor.hdc, i, j)
      If Colour <> &HFFFFFF And Colour <> -1 Then
            cont1 = 0
            cont2 = 0
```

```
            If object = False Then
                  Object1_X1 = i
                  Object1_Y1 = j
                  object = True
                  Qty = Qty + 1

            End If
          Exit For
        End If
           If object = True Then
              If j = picColor.ScaleHeight Then
                  cont1 = cont1 + 1
                  If cont1 = 5 Then
                      Object1_X2 = i
                      Object1_Y2 = j
                      object = False

                    Qty2 = Qty2 + 1
                    If Qty = 0 Then
                      StartAnalystPicY i, 0
                      Dim Z As Integer
                      Z = 1
                      Exit For
                    End If
                  End If
              End If
           End If
      Next j
      If Z = 1 Then Exit For
    Next i
End Sub
Private Sub StartAnalystPicY(ii As Long, jj As Long)
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2, Object1_X3, Object1_X4 As Integer
Dim Object1_Y1, Object1_Y2, Object1_Y3, Object1_Y4 As Integer
Dim ObjectHeight As Integer
Dim Convert As Integer

Qty3 = -1
Qty4 = -1
object = False
  For j = picColor.ScaleHeight / 2 To 0 Step -1
    For i = ii To picColor.ScaleWidth

      Colour = GetPixel(picColor.hdc, i, j)
      If Colour <> &HFFFFFF And Colour <> -1 Then
        If object = False Then
              Object1_X1 = i
              Object1_Y1 = j
              object = True
              Qty3 = Qty3 + 1
        End If
        Exit For
      End If
```

```
            If object = True Then
               If i = picColor.ScaleWidth Then
                   Object1_X2 = i
                   Object1_Y2 = j
                   object = False
                     Qty4 = Qty4 + 1
                     If Qty3 = 0 Then

                   Dim Z As Integer
                   Z = 1
                   Exit For
                End If
               End If
            End If
      Next i

          If Z = 1 Then Exit For
    Next j

End Sub

Private Sub StepHorizontal(ii As Long, jj As Long)
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2, Object1_X3, Object1_X4 As Integer
Dim Object1_Y1, Object1_Y2, Object1_Y3, Object1_Y4 As Integer
Dim ObjectHeight As Integer
Dim Convert As Integer

Qty5 = -1
Qty6 = -1
object = False

  For j = jj To picColor.ScaleHeight
    For i = ii To picColor.ScaleWidth

      Colour = GetPixel(picColor.hdc, i, j)
      If Colour <> &HFFFFFF And Colour <> -1 Then
        If object = False Then
              Object1_X1 = i
              Object1_Y1 = j
              object = True
              Qty5 = Qty5 + 1
        End If
        Exit For
      End If
         If object = True Then
           If i = picColor.ScaleWidth Then
               Object1_X2 = i
                Object1_Y2 = j
                object = False
                Qty6 = Qty6 + 1
              If Qty5 = 0 Then
```

```
                ResizePic2 Picture3(Qty5), ObjectHeight, Convert,
Object1_Y1, Object1_Y2
                showFovea2 picColor, Picture3(Qty5), Object1_Y1,
Object1_Y2
                Picture3(0).Refresh
                Path = App.Path & "\TempPic\pic3.bmp"
                SavePicture Picture3(Qty5).Image, Path
                Dim x As Integer
                x = 1
                Exit For
                End If
              End If
            End If
        Next i
        If x = 1 Then Exit For
    Next j

End Sub

Public Sub StartVertical()
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2, Object1_X3, Object1_X4 As Integer
Dim Object1_Y1, Object1_Y2, Object1_Y3, Object1_Y4 As Integer
Dim ObjectWidth As Integer
Dim Convert As Integer
Dim cont1, cont2 As Integer
Qty = -1
Qty2 = -1
object = False

picColor2.Picture = LoadPicture(App.Path & "\TempPic\pic3.bmp")
  For i = picColor2.ScaleWidth / 2 To 0 Step -1
    For j = 0 To picColor2.ScaleHeight
      Colour = GetPixel(picColor2.hdc, i, j)
      If Colour <> &HFFFFFF And Colour <> -1 Then
            cont1 = 0
            cont2 = 0
            If object = False Then
                  Object1_X1 = i
                  Object1_Y1 = j
                  object = True
                  Qty = Qty + 1
            End If
        Exit For
      End If
         If object = True Then
            If j = picColor2.ScaleHeight Then
                cont1 = cont1 + 1
                If cont1 = 5 Then
                    Object1_X2 = i
                    Object1_Y2 = j
                    object = False
                    Qty2 = Qty2 + 1
                  If Qty = 0 Then
```

```
                    GoToStep2 i, 0
                    Dim Z As Integer
                    Z = 1
                    Exit For
                  End If
                End If
              End If
            End If
        Next j
        If Z = 1 Then Exit For
    Next i
End Sub

Private Sub GoToStep2(ii As Long, jj As Long)
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2, Object1_X3, Object1_X4 As Integer
Dim Object1_Y1, Object1_Y2, Object1_Y3, Object1_Y4 As Integer
Dim ObjectWidth As Integer
Dim Convert As Integer
Dim cont1, cont2 As Integer
Qty = -1
Qty2 = -1
object = False
  For i = picColor2.ScaleWidth / 2 To picColor2.ScaleWidth
    For j = 0 To picColor2.ScaleHeight
      Colour = GetPixel(picColor2.hdc, i, j)
      If Colour <> &HFFFFFF And Colour <> -1 Then
            cont1 = 0
            cont2 = 0
            If object = False Then
                Object1_X1 = i
                Object1_Y1 = j
                object = True
                Qty = Qty + 1
            End If
        Exit For
      End If
         If object = True Then
            If j = picColor2.ScaleHeight Then
                cont1 = cont1 + 1
                If cont1 = 5 Then
                    Object1_X2 = i
                    Object1_Y2 = j
                    object = False

                  Qty2 = Qty2 + 1
                  If Qty = 0 Then
                    GoToStep3 ii, i
                    Dim Z As Integer
                    Z = 1
                    Exit For
                  End If
                End If
            End If
```

```
          End If
      Next j
      If Z = 1 Then Exit For
    Next i
End Sub
Private Sub GoToStep3(iii As Long, jjj As Long)
Dim i As Long
Dim j As Long
Dim object As Boolean
Dim Object1_X1, Object1_X2, Object1_X3, Object1_X4 As Integer
Dim Object1_Y1, Object1_Y2, Object1_Y3, Object1_Y4 As Integer
Dim ObjectWidth As Integer
Dim Convert As Integer
Dim cont1, cont2 As Integer
Qty = -1
Qty2 = -1
object = False

  For i = iii To jjj
    For j = 0 To picColor2.ScaleHeight
      Colour = GetPixel(picColor2.hdc, i, j)
      If Colour <> &HFFFFFF And Colour <> -1 Then
            If object = False Then
                  Object1_X1 = i
                  Object1_Y1 = j
                  object = True
                  Qty = Qty + 1
            End If
        Exit For
      End If
         If object = True Then
            If j = picColor2.ScaleHeight Then
               Object1_X2 = i
               Object1_Y2 = j
               object = False
              Qty2 = Qty2 + 1
              If Qty = 0 Then
              ResizePic Picture4(Qty), ObjectWidth, Convert,
Object1_X1, Object1_X2
              showFovea picColor2, Picture4(Qty), Object1_X1,
Object1_X2
              Picture5.Picture = Picture4(Qty).Image
              Path = App.Path & "\TempPic\OCRPic" & Qty & ".bmp"
              SavePicture Picture5, Path
              Else
              Load Picture4(Qty)
              Picture4(Qty).Visible = True
              ResizePic Picture4(Qty), ObjectWidth, Convert,
                Object1_X1, Object1_X2

              Picture4(Qty).Height = picColor2.Height
              Picture4(Qty).Left = Picture4(Qty - 1).Left +
                Picture4(Qty - 1).Width + 100

              showFovea picColor2, Picture4(Qty), Object1_X1,
                Object1_X2
```

```
                Picture4(0).Refresh
                Picture5.Picture = Picture4(Qty).Image
                Path = App.Path & "\TempPic\OCRPic" & Qty & ".bmp"
                SavePicture Picture5, Path
                End If
              End If
            End If
        Next j
    Next i
End Sub

Sub showFovea(inputImage As PictureBox, foveaImage As PictureBox,
ByVal X1 As Integer, ByVal X2 As Integer)
Dim Cal As Integer
Dim rc As Integer

Cal = X2 - X1
rc = BitBlt(foveaImage.hdc, 5, 0, Cal, 500, inputImage.hdc, X1,
0, SRCCOPY)
End Sub
Sub ResizePic(Picture As PictureBox, _
              ObjectWidth As Integer, _
              Convert As Integer, _
              ByVal Object1_X1 As Integer, _
              ByVal Object1_X2 As Integer)

     ObjectWidth = (Object1_X2 - Object1_X1) + 20
     Convert = ObjectWidth * Screen.TwipsPerPixelX
     Picture.Width = Convert - 100
     Picture.Height = picColor2.Height
End Sub

Sub showFovea2(inputImage As PictureBox, foveaImage As
PictureBox, ByVal Y1 As Integer, ByVal Y2 As Integer)
Dim Cal As Integer
Dim rc As Integer

Cal = Y2 - Y1
rc = BitBlt(foveaImage.hdc, 0, 5, 500, Cal, inputImage.hdc, 0,
Y1, SRCCOPY)
End Sub
Sub ResizePic2(Picture As PictureBox, _
               ObjectHeight As Integer, _
               Convert As Integer, _
               ByVal Object1_Y1 As Integer, _
               ByVal Object1_Y2 As Integer)

     ObjectHeight = (Object1_Y2 - Object1_Y1) + 20
     Convert = ObjectHeight * Screen.TwipsPerPixelY
     Picture.Width = picColor.Width
     Picture.Height = Convert - 100
End Sub

Public Sub ClearIndexPic()
    If Qty > 0 Then
    Dim CIP As Integer
```

```
    For CIP = 1 To Qty

        Unload Picture4(CIP)
        Next CIP
        Picture4(0) = Nothing
    End If
End Sub
Public Sub ODRFuntion()
Dim test As Integer
Dim MyLen
For test = 0 To Qty
        Path = App.Path & "\TempPic\OCRPic" & test & ".bmp"
        Text2.Text = Trim(OCR(Path, -1))
        Text2.Text = Left$(Text2, 1)
        Text3.SelStart = Len(Text3.Text)
        Text3.SelText = Trim(Text2.Text)
        Text3 = StrConv(Text3.Text, vbLowerCase)
        MyLen = Left$(Text3, Len(Trim(Text3)) + 1)
        MyLen = Right$(MyLen, Len(Trim(MyLen)))
        MDIForm1.txtSearch.Text = MyLen
Next
End Sub
```

```
///Modules.bas///
'All Function is often called and function API
Option Explicit

  Public Declare Function BitBlt Lib "gdi32" _
    (ByVal hDCDest As Long, _
     ByVal XDest As Long, _
     ByVal YDest As Long, _
     ByVal nWidth As Long, _
     ByVal nHeight As Long, _
     ByVal lScreenDC As Long, _
     ByVal xSrc As Long, _
     ByVal ySrc As Long, _
     ByVal dwRop As Long) As Long

Public Declare Function CallWindowProc& Lib "User32" Alias
"CallWindowProcA" _
    (ByVal lpPrevWndFunc&, _
     ByVal hwnd&, _
     ByVal Msg&, _
     ByVal wParam&, _
     ByVal lParam&)

  ' declarations of the API functions used
Public Declare Sub CopyMemory Lib "kernel32" Alias
"RtlMoveMemory" _
    (lpDest As Any, _
     lpSource As Any, _
     ByVal cBytes&)

  Public Declare Function CreateCompatibleDC Lib "gdi32" _
    (ByVal hdc As Long) As Long

Public Declare Function CreateCompatibleBitmap Lib "gdi32" _
    (ByVal hdc As Long, _
     ByVal nWidth As Long, _
     ByVal nHeight As Long) As Long

Public Declare Function DeleteDC Lib "gdi32" _
    (ByVal hdc As Long) As Long

Public Declare Function GetCursorPos Lib "User32" (lpPoint As
POINTAPI) As Long

Public Declare Function GetDesktopWindow Lib "User32" () As Long

Public Declare Function GetDIBits Lib "gdi32" _
    (ByVal aHDC As Long, _
     ByVal hBitmap As Long, _
     ByVal nStartScan As Long, _
     ByVal nNumScans As Long, _
     lpBits As Any, _
     lpBI As BITMAPINFO, _
     ByVal wUsage As Long) As Long
```

```
Public Declare Function GetPixel Lib "gdi32" _
    (ByVal hdc As Long, _
     ByVal x As Long, _
     ByVal y As Long) As Long

Public Declare Function GetObject Lib "gdi32" Alias "GetObjectA"
_(ByVal hObject As Long, _
  ByVal nCount As Long, _
  lpObject As Any) As Long

Public Declare Function GetWindowDC Lib "User32" _
    (ByVal hwnd As Long) As Long

Public Declare Function GetWindowRect Lib "User32" _
    (ByVal hwnd As Long, _
     lpRect As RECT) As Long

Public Declare Function LineTo& Lib "gdi32" _
    (ByVal hdc As Long, _
     ByVal x As Long, _
     ByVal y As Long)

Public Declare Function MoveToEx& Lib "gdi32" _
    (ByVal hdc As Long, _
     ByVal x As Long, _
     ByVal y As Long, _
     ByVal lp As Long)

Public Declare Function OCR Lib "OCR.dll" _
    (ByVal file As String, _
     ByVal imageType As Long) As String

Public Declare Function OleCreatePictureIndirect Lib
"olepro32.dll"(lpPictDesc As PICTDESC, _
     riid As Any, _
     ByVal fOwn As Long, _
     ipic As IPicture) As Long

Public Declare Function SelectObject Lib "gdi32" _
    (ByVal hdc As Long, _
     ByVal hObject As Long) As Long

Public Declare Function SetDIBits Lib "gdi32" _
    (ByVal hdc As Long, _
    ByVal hBitmap As Long, _
    ByVal nStartScan As Long, _
    ByVal nNumScans As Long, _
    lpBits As Any, _
    lpBI As BITMAPINFO, _
    ByVal wUsage As Long) As Long

  ' API call to alter the class data for a window
Public Declare Function SetWindowLong Lib "User32" Alias
"SetWindowLongA" _
    (ByVal hwnd&, _
     ByVal nIndex&, _
```

```
     ByVal dwNewLong&) As Long

Public Declare Function SetWindowPos Lib "User32" _
    (ByVal hwnd As Long, _
     ByVal hWndInsertAfter As Long, _
     ByVal x As Long, _
     ByVal y As Long, _
     ByVal cx As Long, _
     ByVal cy As Long, _
     ByVal wFlags As Long) As Long

Public Declare Function ReleaseDC Lib "User32" _
    (ByVal hwnd As Long, _
     ByVal hdc As Long) As Long

  ' the message we will subclass
Public Const WM_GETMINMAXINFO As Long = &H24&
Public Const SWP_NOMOVE = &H2
Public Const SWP_NOSIZE = &H1
Public Const HWND_TOPMOST = -1
Public Const HWND_NOTOPMOST = -2
Public Const GWL_WNDPROC As Long = (-4&)
Public Const SRCCOPY = &HCC0020

Public Const debugme As Boolean = False
Public g_nProcOld As Long
Public bitmap_info As BITMAPINFO
Public ave_color As Byte

Public Const DIB_RGB_COLORS = 0&
Public Const BI_RGB = 0&

Public Const pixR As Integer = 3
Public Const pixG As Integer = 2
Public Const pixB As Integer = 1

Public Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Public Type PICTDESC
    cbSize As Long
    pictType As Long
    hIcon As Long
    hPal As Long
End Type

Public Type POINTAPI
    x As Long
    y As Long
End Type
```

```
Public Type MINMAXINFO
    ptReserved As POINTAPI
    ptMaxSize As POINTAPI
    ptMaxPosition As POINTAPI
    ptMinTrackSize As POINTAPI
    ptMaxTrackSize As POINTAPI
End Type

Public Type BITMAP '14 bytes
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type

Public Type BITMAPINFOHEADER '40 bytes
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As Long
    biSizeImage As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type

Public Type RGBQUAD
    rgbBlue As Byte
    rgbGreen As Byte
    rgbRed As Byte
    rgbReserved As Byte
End Type

Public Type BITMAPINFO
    bmiHeader As BITMAPINFOHEADER
    bmiColors As RGBQUAD
End Type
Public Function RGB_get(ByVal CVal As Long, r As Long, B As Long,
G As Long)
G = Int(CVal / 65536)
B = Int((CVal - (65536 * G)) / 256)
r = CVal - (65536 * G + 256 * B)
End Function
```

ประวัติผู้เขียนโครงงาน

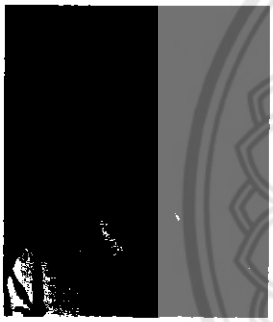
ชื่อ นายพิชิต ธรรมวงศ์

ภูมิลำเนา บ้านเลขที่ 68 หมู่ 5 ตำบล แม่ยวม อำเภอแม่สะเรียง จังหวัดแม่ฮ่องสอน 58110

ประวัติการศึกษา

- จบมัธยมศึกษาตอนปลายจาก โรงเรียนแม่สะเรียง "บริพัตร" ศึกษา จังหวัดแม่ฮ่องสอน
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร พิษณุโลก

E-mail: skn_yao@hotmail.com



ชื่อ นายพีระพล นกเผือก

ภูมิลำเนา บ้านเลขที่ 50 หมู่ 6 ตำบล วงฆ้อง อำเภอพรหมพิราม จังหวัดพิษณุโลก 65180

ประวัติการศึกษา

- จบมัธยมศึกษาตอนปลาย จากโรงเรียนอินทุภูติพิยา อำเภอพรหมพิราม จังหวัดพิษณุโลก
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร พิษณุโลก

E-mail: p_pee_nk@hotmail.com / Peerapol_Nokphuak@yahoo.com