

ภาควิชาครุศาสตร์วิศวกรรม
 คณะครุศาสตร์อุตสาหกรรม
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ใบรับรองปริญญาโท

ปริญญาโท โปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80
 CAI MICROPROCESSOR Z80

- ชื่อนักศึกษา
- | | | | |
|----------------|-------------|--------------|----------|
| 1. นายไพศาล | ภานนท์ | รหัสประจำตัว | 39031423 |
| 2. นางสาวศศิธร | เกียรติจรูญ | รหัสประจำตัว | 39031432 |
| 3. นางสาวอัญญา | สีนวล | รหัสประจำตัว | 39031441 |
| 4. นายอานันต์ | สายสลาม | รหัสประจำตัว | 39031442 |

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์
 อาจารย์ควบคุมปริญญาโท

- | | |
|-----------------|--------------------|
| 1. คร.สุรสิทธิ์ | ราตรี |
| 2. ผศ.ดร.ธีระพล | เทพหัสดิน ณ อยุธยา |
| 3. อาจารย์สุชิน | อาจหาญ |



คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อาจารย์สุชิน อาจหาญ	
2. อาจารย์กิตติพงศ์ มะโน	
3. อาจารย์อำพล ทองระอา	
4. อาจารย์ไพบุลย์ พวงวงศ์ตระกูล	
5. อาจารย์ปิยะ จิตธรรมมาภิรมย์	

วันเดือนปีที่สอบวันที่ 11 ธันวาคม 2540 เวลา 11.00 น. ถึง 13.00 น.

สถานที่สอบ ห้อง 310 คณะครุศาสตร์อุตสาหกรรม

เลขหม.....
 เลขทะเบียน..... 30121
 วัน, เดือน, ปี..... ๑๑ ส.ย. 2541



ภาควิชารับรองแล้ว

 ธีระพล เทพหัสดิน ณ อยุธยา
 ภาควิชาครุศาสตร์วิศวกรรม
 เดือน..... พ.ศ.....

ปริญญานิพนธ์

โปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80

CAI MICROPROCESSOR Z80



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


ปริญญานิพนธ์

เรื่อง โปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80
CAI MICROPROCESSOR Z80


ผู้จัดทำ

1. นายไพศาล ภานนท์
2. นางสาวศศิธร เกียรติขจรบูรณ์
3. นางสาวอัจฉรา สีนวล
4. นายอาทิตย์ สายสลาม

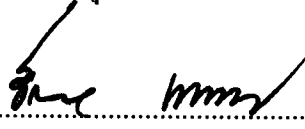
อาจารย์ที่ปรึกษา

ลงนาม.....
(ดร.สุรสิทธิ์ ราตรี)

ลงนาม.....
(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

ลงนาม.....
(อาจารย์สุชิน อจหาญ)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....
(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

ปริญญานิพนธ์

เรื่อง โปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80
CAI MICROPROCESSOR Z80

วัตถุประสงค์

1. เพื่อศึกษาโปรแกรมสำหรับสร้างสื่อประสม HURESS
2. เพื่อสร้างโปรแกรมช่วยสอน วิชาไมโครโปรเซสเซอร์ Z80 โดยใช้โปรแกรมสำหรับสร้างสื่อประสม HURESS ของโครงการวิจัยร่วมระหว่างไทยและญี่ปุ่นในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบสอนทางไกลแบบสองทาง พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ
3. เพื่อเป็นแนวทางการพัฒนา โปรแกรมช่วยสอนวิชาไมโครโปรเซสเซอร์ Z80 ให้เป็นส่วนหนึ่งของวิชา ประจำในโครงการวิจัยร่วมฯ และสามารถถูกนำไปใช้อย่างแพร่หลายต่อไป

ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อใช้เป็นสื่อช่วยการเรียนรู้ในเนื้อหาของไมโครโปรเซสเซอร์เบอร์ Z80 เบื้องต้น
2. เพื่อใช้เป็นต้นแบบของการศึกษาโปรแกรมสำหรับสร้างสื่อประสม HURESS ของโครงการวิจัยร่วมระหว่างไทยญี่ปุ่นในการพัฒนาทรัพยากรมนุษย์ด้วยระบบการสอนทางไกลแบบสองทางพร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ
3. เพื่อพัฒนาให้ ช่วยสอนวิชาไมโครโปรเซสเซอร์ Z80 ให้เป็นส่วนหนึ่งของวิชาประจำในโครงการวิจัยร่วมฯ และสามารถถูกนำไปใช้อย่างแพร่หลายต่อไป

โปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80

นายไพศาล	ภานนท์
นางสาวศศิธร	เกียรติจักรนุรณ
นางสาวอัจฉรา	สีนวล
นายอานัติ	สายสถาม

อาจารย์ที่ปรึกษา

ดร.สุรสิทธิ์	ราตรี
ผศ.ดร.ธีระพล	เทพหัสดิน ณ อยุธยา
อาจารย์สุชิน	อาจหาญ

ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอการสร้างบทเรียนโปรแกรมช่วยสอนวิชา "ไมโครโปรเซสเซอร์ Z80" ซึ่งมีเนื้อหาคำบรรยายเป็นภาษาอังกฤษ ทั้งนี้เพื่อพัฒนาเนื้อหาวิชาให้สอดคล้องกับระบบของโครงการวิจัยร่วมระหว่างไทย ญี่ปุ่น และจีน ในการพัฒนาทรัพยากรมนุษย์ด้วยระบบการสอนทางไกลแบบสองทาง พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ โดยใช้เครื่องมือการพัฒนา CAI (AUTHORING TOOLS) ชื่อโปรแกรม HURESS บทเรียนโปรแกรมคอมพิวเตอร์ช่วยสอนนี้ สามารถนำไปใช้เป็นบทเรียนจริงหรือนำไปใช้เป็นบทเรียนเสริมวิชา ไมโครโปรเซสเซอร์ ทำให้การเรียนมีประสิทธิภาพมากขึ้น เนื่องจากข้อได้เปรียบในเรื่องการให้ข้อมูลป้อนกลับ การเปิดโอกาสให้ผู้เรียนได้มีส่วนร่วมในการเรียนการสอน และยังลดปัญหาความแตกต่างระหว่างบุคคล

CAI MICROPROCESSOR Z-80

MR.PAISARN	PARNON
MISS SASITORN	KIATHKAJORNBOON
MISS AHTCHARA	SEENAL
MR.ARNUT	SAISLAM

ADVISORS

DR.SURASIT	RATREE
DR.TEERAPOL	TEPHATSADIN N AYUTAYA
MR.SUCHIN	ADHAN

1997

ABSTRACT

This thesis presents the computer Assisted Instruction program used in the course "MICROPROCESSOR Z-80" instructed in English so as to develop the course content with the serve of Collaborative Research Project on Multimedia Human Resource Development Network Technology among Thailand, Japan and China. To develop human resource in tele-education on media via satellite with CAI (Authoring Tool) by HURESS. This program could be applied more effectively which learners could be used individually. This brings about the more efficiency of learning by the advantages of information retrieval, helping learners participate in teaching and learning and also lessening individual difference.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ สำเร็จลุล่วงลงได้เพราะพระคุณของ คุณพ่อ คุณแม่ที่ช่วยเหลือทางด้านทุนการศึกษา ท่านอาจารย์ ดร.สุรสิทธิ์ ราตรี อาจารย์ที่ปรึกษาปริญญาบัตรนี้ ที่ให้ความช่วยเหลือทางด้านคำปรึกษา ข้อเสนอแนะแนวทางและวิธีแก้ไขปัญหาในการดำเนินงาน รวมถึงสถานที่ เครื่องมือและอุปกรณ์ต่างๆ รวมทั้งอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน และขอขอบคุณคุณสุวิทย์ ยิบมันตะสิริ เจ้าหน้าที่ควบคุมห้องสอนทางไกล ที่ได้ให้ความช่วยเหลือให้คำปรึกษาและอำนวยความสะดวกในการใช้ห้องสอนทางไกล รวมทั้งคุณอมรเทพ ทองเหลือง ที่ได้คำปรึกษาแนะนำเกี่ยวกับโปรแกรมต่างๆ ที่ใช้ในโครงการงาน และเพื่อนๆ ทุกคนที่ให้กำลังใจตลอดมา ซึ่งในโอกาสนี้คณะผู้จัดทำรู้สึกซาบซึ้งในพระคุณของทุก ๆ ท่าน คณะผู้จัดทำขอขอบพระคุณเป็นอย่างยิ่งมา ณ โอกาสนี้



IV

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญาโท	1
1.2 ขอบเขตของปริญญาโท	2
1.3 เนื้อหาโดยสังเขป	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 กล่าวนำ	4
2.2 ประวัติความเป็นมาของคอมพิวเตอร์ช่วยสอน	4
2.2.1 ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน	5
2.2.2 ลักษณะการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอน	6
2.2.3 ข้อได้เปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน	7
2.2.4 ข้อเสียเปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน	7
2.2.5 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน	8
2.2.6 ลักษณะโครงสร้างของบทเรียนคอมพิวเตอร์ช่วยสอน	8
2.2.7 การสอนทางไกลใน โครงการวิจัยร่วมญี่ปุ่นและจีนใน	8
การพัฒนาทรัพยากรมนุษย์ ด้วยระบบการสอนทางไกลแบบ	
สองทางพร้อมสื่อประสมดาวเทียมระหว่างประเทศ	
2.3 หลักการของโปรแกรม HURESS	11
2.4 ส่วนประกอบของโปรแกรม HURESS	12
2.4.1 Books	12
2.4.2 Page	12

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.3 Media	13
2.4.4 Desk	13
2.4.5 Bookshelf	13
2.4.6 Entity	13
2.4.7 Procedures	13
2.4.8 Object	13
2.5 หลักการใช้โปรแกรม HURESS	14
2.5.1 การยืมและเปิดหนังสือ(Borrowing and Opening a Book)	14
2.5.2 การเปิดหนังสือ Opening a Book	15
2.5.3 การทำงานเกี่ยวกับหน้าหนังสือ	16
2.5.4 การใช้งาน Desk Browser	17
2.5.5 การใช้หน้าสารบัญ	18
2.5.6 การใช้หน้าดัชนี	18
2.5.7 การคืนหนังสือ	19
2.6 การทำงานในส่วนของ Desk Main Menu	20
2.6.1 การทำงานในส่วนของเมนูย่อย [Desk]	21
2.6.2 การทำงานในส่วนของเมนูย่อย [Bookshelf]	22
2.6.3 การทำงานในส่วนของเมนูย่อย [Book]	22
2.6.4 การทำงานในส่วนของเมนูย่อย [Page]	23
2.6.5 การทำงานในส่วนของเมนูย่อย [Edit]	24
2.6.6 การทำงานในส่วนของ Dialog Box ต่างๆ ในโปรแกรม HURESS	25
2.6.7 Media Information Dialog Box	34
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	40
3.1 เครื่องมือที่เกี่ยวข้องในการออกแบบและการทำงาน	40
3.1.1 ทางด้านฮาร์ดแวร์ (Hardware)	40

สารบัญ (ต่อ)

เรื่อง	หน้า
3.1.2 ทางด้านซอฟต์แวร์ (Software)	40
3.2 ขั้นตอนในการออกแบบโปรแกรม	41
3.3 ขั้นตอนในการสร้างบทเรียน	44
3.4 ขั้นตอนในการสร้างแบบฝึกหัด	44
3.5 วิธีในการสร้างส่วนของบทเรียนและแบบฝึกหัดด้วยโปรแกรม HURESS	45
3.5.1 การสร้างหนังสือ	45
3.5.2 การสร้างข้อความของหนังสือ	48
3.5.3 การสร้างลำดับการทำงานของหนังสือ	50
3.5.4 การสร้างปุ่ม Next	51
3.5.5 การสร้าง Basic Page ของหน้า Background	52
3.5.6 การสร้างกรอบของภาพ	54
3.5.7 การสร้าง Button Next Page	55
3.5.8 การสร้าง Button Previous page	56
3.5.9 การสร้าง Button Return	57
3.5.10 การเขียนโปรแกรมย่อยควบคุมการทำงานของหน้าหนังสือ	59
บทที่ 4 การทดลองและผลการทดลอง	61
4.1 การเข้าสู่โปรแกรม HURESS	61
4.2 การใช้งานของโปรแกรม	62
4.2.1 ส่วนของเนื้อหา	62
4.2.2 ส่วนของแบบทดสอบ	67
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา	72
5.1 ปัญหาที่พบในการทำงาน	72
5.2 การแก้ปัญหา	73
5.3 ข้อเสนอแนะ	73
5.4 ประโยชน์ที่ได้รับจากโครงการ	73

VII

สารบัญ (ต่อ)

เรื่อง	หน้า
5.5 แนวทางพัฒนา	74
ภาคผนวก ก การติดตั้งโปรแกรม HURESS	75
ภาคผนวก ข การกำหนดรูปแบบของหนังสือด้วยโปรแกรมย่อย	85
ภาคผนวก ค รูปแบบการใช้งานของฟังก์ชันต่างๆ	89
บรรณานุกรม	146
ประวัติผู้แต่ง	147



VIII

สารบัญภาพ

รูปภาพ	หน้า
รูปที่ 2.1 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน	5
รูปที่ 2.2 ระบบสอนทางไกลแบบสองทางพร้อมสื่อประสมดาวเทียมระหว่างประเทศ	9
รูปที่ 2.3 เครื่องข่ายของระบบ	9
รูปที่ 2.4 การเขียนโปรแกรมแบบธรรมดาและการเขียนโปรแกรมแบบ Event-driven	12
รูปที่ 2.5 การขีมนหนังสือผ่าน Desk Main Menu	14
รูปที่ 2.6 หนังสือที่ถูกขีมน	14
รูปที่ 2.7 การขีมนหนังสือผ่าน Book Shelf Browser	15
รูปที่ 2.8 การเปิดหนังสือ	16
รูปที่ 2.9 หนังสือที่ทำการเปิดได้แล้ว	16
รูปที่ 2.10 การทำงานเกี่ยวกับหน้าหนังสือ	17
รูปที่ 2.11 การใช้ Desk Browser	17
รูปที่ 2.12 การใช้หน้าสารบัญ	18
รูปที่ 2.13 การใช้หน้าดัชนี	19
รูปที่ 2.14 การคืนหนังสือโดยปุ่ม Close Box	19
รูปที่ 2.15 การทำงานในส่วนของ Desk Main Menu	20
รูปที่ 2.16 การเข้าโปรแกรม HURESS	25
รูปที่ 2.17 การเปิดหนังสือ	26
รูปที่ 2.18 การขีมนหนังสือ	26
รูปที่ 2.19 การกำหนดตำแหน่ง Selection bar	26
รูปที่ 2.20 การ Set การแสดงของ Item ใน Section	27
รูปที่ 2.21 การกำหนดแบ็คกราวด์	27
รูปที่ 2.22 การเพิ่มชั้นวางหนังสือ	28
รูปที่ 2.23 การเพิ่มชั้นวางหนังสือ	28
รูปที่ 2.24 การสร้างหนังสือใหม่	29

สารบัญญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 2.25 การกำหนดขนาดของหนังสือ	29
รูปที่ 2.26 การกำหนดรายละเอียดในส่วนต่างๆของหนังสือ	30
รูปที่ 2.27 การยืมหนังสือ	30
รูปที่ 2.28 การลบหนังสือ	31
รูปที่ 2.29 การย้ายหนังสือ	31
รูปที่ 2.30 การคัดลอกหนังสือ	32
รูปที่ 2.31 การกำหนดรายละเอียดหน้าของเนื้อหา	32
รูปที่ 2.32 การกำหนดหน้าสารบัญ	33
รูปที่ 2.33 การกำหนดขนาดของหน้า	33
รูปที่ 2.34 การกำหนดรายละเอียดของหน้า	34
รูปที่ 2.35 การกำหนดดัชนี	34
รูปที่ 2.36 การทำรูปเคลื่อนไหว	35
รูปที่ 2.37 การเลือกคำตอบในแบบฝึกหัด	35
รูปที่ 2.38 การกำหนดรายละเอียดของหน้าดัชนี	36
รูปที่ 2.39 การเรียกภาพมาประกอบโปรแกรม	36
รูปที่ 2.40 การตั้งเวลา	37
รูปที่ 2.41 การสร้างข้อความ	37
รูปที่ 2.42 การสร้างปุ่ม	38
รูปที่ 2.43 การดึงรูปมาประกอบโปรแกรม	38
รูปที่ 2.44 การสร้างปุ่มทำงานต่างๆ	39
รูปที่ 3.1 ผังการทำงานของ โปรแกรมช่วยสอนวิชาดิจิทัล	41
รูปที่ 3.2 ผังขั้นตอนการออกแบบโปรแกรม	42
รูปที่ 3.2 ผังขั้นตอนการออกแบบโปรแกรม (ต่อ)	43
รูปที่ 3.3 แผนผังการสร้างบทเรียน	44

สารบัญญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.4 แผนผังการสร้างแบบฝึกหัด	45
รูปที่ 3.5 เมนู Desk ที่ Create and Borrow a New Book	45
รูปที่ 3.6 Dialog Box ในการกำหนดขนาดของหนังสือ	46
รูปที่ 3.7 วินโดว์ของหนังสือเมื่อสร้างเสร็จแล้ว	46
รูปที่ 3.8 Operation of Media Selection	47
รูปที่ 3.9 Desk Browser	47
รูปที่ 3.10 การสร้างข้อความของหนังสือ	48
รูปที่ 3.11 การเพิ่มของ Media ใน Desk Browser	48
รูปที่ 3.12 การกำหนดรายละเอียดของมีเดียที่สร้างขึ้น	49
รูปที่ 3.13 การกำหนดรายละเอียดของข้อความ	49
รูปที่ 3.14 ผลที่ได้จากการสร้างมีเดีย	50
รูปที่ 3.15 การสร้างโปรแกรมย่อย	50
รูปที่ 3.16 การเขียนโปรแกรมย่อยของหนังสือ	51
รูปที่ 3.17 การกำหนดรายละเอียดของปุ่ม	51
รูปที่ 3.18 การเขียนโปรแกรมย่อยกำหนดการทำงานของปุ่ม	52
รูปที่ 3.19 ตัวอย่าง Front Cover ที่สร้าง	52
รูปที่ 3.20 การเพิ่ม Basic page ใน Desk Browser	53
รูปที่ 3.21 การแสดงผลของหน้า Basic Page	53
รูปที่ 3.22 การกำหนดรายละเอียดของ Panel Media	54
รูปที่ 3.23 Panel หลังจากกำหนดรายละเอียดแล้ว	54
รูปที่ 3.24 การกำหนดรายละเอียดของ Button	55
รูปที่ 3.25 การเขียนโปรแกรมย่อยให้กับ Button	55
รูปที่ 3.26 การแสดงผลของ Button	56
รูปที่ 3.27 การกำหนดรายละเอียดของ Previous Button	56

สารบัญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.28 การเขียนโปรแกรมย่อยของ Previous Button	57
รูปที่ 3.29 การแสดงผลของ Previous Button	57
รูปที่ 3.30 การกำหนดรายละเอียดของ Button	58
รูปที่ 3.31 การเขียนโปรแกรมย่อยให้กับ Button	58
รูปที่ 3.32 การแสดงผล Button	59
รูปที่ 3.33 การเขียนโปรแกรมย่อยของ Next Page	59
รูปที่ 3.34 การเขียนโปรแกรมย่อยของ Previous Page	60
รูปที่ 4.1 การเข้าสู่โปรแกรม	61
รูปที่ 4.2 การป้อน User name	62
รูปที่ 4.3 การเลือกวิชาเรียน	62
รูปที่ 4.4 Front Cover	63
รูปที่ 4.5 How to Read This matterial	63
รูปที่ 4.6 Unit Menu	64
รูปที่ 4.7 เนื้อหาแต่ละบทเรียน	64
รูปที่ 4.8 ปุ่ม Click here to show figure	65
รูปที่ 4.9 การใช้งานหน้าสารบัญ	66
รูปที่ 4.10 การใช้งานหน้าดัชนี	66
รูปที่ 4.11 ปุ่ม Close	67
รูปที่ 4.12 หน้าจอหลักของส่วนแบบทดสอบ	67
รูปที่ 4.13 หน้าจอหลักของหัวข้อแบบทดสอบ	68
รูปที่ 4.14 หน้าจอแบบทดสอบหลักของบทเรียน	68
รูปที่ 4.15 คำถามหัวข้อที่ต้องการทดสอบ	69
รูปที่ 4.16 แสดงข้อความคำตอบถูกต้อง	69
รูปที่ 4.17 แสดงข้อความคำตอบผิด	70

สารบัญญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 4.18 แสดงคำอธิบายคำตอบเพิ่มเติม	70
รูปที่ ก.1 การ Download software	76
รูปที่ ก.2 ตำแหน่งที่ต้องการเก็บไฟล์	76
รูปที่ ก.3 การบันทึกโปรแกรม	77
รูปที่ ก.4 ตำแหน่งที่จะเก็บไฟล์	77
รูปที่ ก.5 การExtract ไฟล์ Oracle95.exe	78
รูปที่ ก.6 ทำการ Extract เสร็จสมบูรณ์	78
รูปที่ ก.7 การ Extract ไฟล์ Huresscai.exe	78
รูปที่ ก.8 ทำการ Extract เสร็จสมบูรณ์	79
รูปที่ ก.9 ไฟล์ใน Sub directory temp	79
รูปที่ ก.10 การติดตั้ง ORACLE	79
รูปที่ ก.11 การติดตั้ง ORACLE เสร็จสมบูรณ์	80
รูปที่ ก.12 การติดตั้ง CAI Setup เสร็จสมบูรณ์	81
รูปที่ ก.13 Short cut ของ HURESS	81
รูปที่ ก.14 Configuration network	82
รูปที่ ก.15 Domain Sever	82
รูปที่ ก.16 Protocol TCP/IP	83
รูปที่ ก.17 Properties ของ Protocol TCP/IP	83

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญาโท

จากโครงการวิจัยร่วมระหว่างไทยและญี่ปุ่นในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบสอนทางไกลแบบสองทาง พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ ซึ่งมีหลักการและเหตุผลดังต่อไปนี้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง ได้ร่วมมือกับมหาวิทยาลัยฮอกไกโด ประเทศญี่ปุ่น ทำการศึกษาค้นคว้าวิจัยด้านการพัฒนาทรัพยากรมนุษย์ ด้วยระบบการสอนทางไกลแบบสองทาง พร้อมสื่อประสม โดยติดต่อสื่อสารผ่านดาวเทียมระหว่างประเทศ(JC SAT-3)ของประเทศญี่ปุ่น โครงการนี้ได้รับพระมหากรุณาธิคุณจากสมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี เสด็จเปิดโครงการเมื่อวันที่ 4 กรกฎาคม พ.ศ.2539 ในงานนิทรรศการพระจอมเกล้าลาดกระบังนิทรรศน์เฉลิมพระเกียรติเทคโนโลยีเพื่อปวงชนที่ผ่านมา โครงการนี้มีวัตถุประสงค์เพื่อ

1. พัฒนาทรัพยากรมนุษย์ด้านโทรคมนาคมในภูมิภาคเอเชียแปซิฟิก
2. เพื่อพัฒนาระบบเครือข่ายการสอนทางไกลด้วยสื่อระบบสองทาง ผ่านดาวเทียมที่มีประสิทธิภาพมากกว่าวิธีปกติ ที่ใช้สอนระหว่างมนุษย์ต่อมนุษย์โดยตรง
3. เพื่อประเมินผลของระบบดังกล่าวที่ถูกนำมาใช้ระหว่างประเทศญี่ปุ่นและประเทศในภูมิภาคเอเชียแปซิฟิก

ระยะเวลาของการดำเนินโครงการวิจัย 2 ปี โครงการได้เริ่มดำเนินการวิจัย โดยได้ทำการสอนครั้งแรกพร้อมกัน 3 ประเทศ คือระหว่างมหาวิทยาลัยฮอกไกโด ประเทศญี่ปุ่น มหาวิทยาลัยนานกิง ประเทศจีน และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประเทศไทย ตั้งแต่วันที่ 6 พฤศจิกายน 2539 เป็นต้นมาโดยทำการสอนประมาณ สัปดาห์ละ 3 วัน วันละ 1 ชั่วโมงประกอบด้วยจำนวน 3 วิชาคือ

1. วิชา Telecommunication Network ตั้งแต่วันที่ 6-27 พฤศจิกายน 2539 จำนวน 10 บท
2. วิชา Online System Design ตั้งแต่วันที่ 3-25 ธันวาคม 2539 จำนวน 13 บท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. วิชา Network Theory ตั้งแต่วันที่ 9-30 มิถุนายน 2540 จำนวน 10 บท

นอกจากการศึกษาวิจัยถึงประสิทธิภาพการเรียนการสอนด้วยระบบวิธีการดังกล่าวแล้วโครงการวิจัยนี้ ยังมุ่งพัฒนาสื่อการเรียนการสอน ในลักษณะสื่อประสม ของเนื้อหาวิชาอื่นๆที่เกี่ยวข้องกับวิทยาศาสตร์และเทคโนโลยี เมื่อพัฒนาได้คุณภาพตามต้องการแล้ว จะสามารถนำไปใช้ในการเรียนการสอนร่วมกับมหาวิทยาลัยของประเทศอื่นๆในเอเชียแปซิฟิกอีกด้วย

โครงการนี้ได้รับความสนใจจากนักศึกษาและประชาชนทั่วไป และจะเป็นประโยชน์อย่างยิ่งต่อการพัฒนาการศึกษา และช่วยแก้ปัญหาการขาดแคลนอาจารย์ สร้างความเสมอภาคทางการศึกษา สามารถติดต่อกับมหาวิทยาลัยในต่างประเทศทั่วโลกได้ อันจะนำไปสู่การพัฒนาทางวิชาการต่อไป

ดังนั้นเพื่อให้สอดคล้องต่อโครงการดังกล่าว กลุ่มผู้เสนอหัวข้อปริญญานิพนธ์จึงได้มีแนวความคิดจัดสร้างวิชาใหม่ขึ้นมาอีกหนึ่งวิชา จากเดิมที่มีอยู่เพียง 3 วิชาโดยได้เลือกเนื้อหาวิชาที่ผู้ศึกษาคอมพิวเตอร์โดยทั่วไปจะต้องมีความรู้เหล่านี้เป็นพื้นฐาน ดังนั้นจึงได้เลือกวิชา ไมโครโปรเซสเซอร์ Z80 ในการจัดทำโปรแกรมช่วยสอนในโครงการนี้

1.2 ขอบเขตของปริญญานิพนธ์

โปรแกรมช่วยสอนวิชาไมโครโปรเซสเซอร์ Z80 สามารถแสดงการทำงานในโหมดกราฟฟิก (Graphics Mode) ในบทเรียนจะแบ่งเนื้อหาออกเป็น 10 บท แต่ละบทจะมีแบบทดสอบท้ายบทเพื่อเป็นการทดสอบความเข้าใจ โดยที่ผลการทดสอบของนักเรียนแต่ละคนจะปรากฏบนจอคอมพิวเตอร์ของผู้สอนด้วย ซึ่งผู้สอนจะเป็นคนเช็คได้ว่า นักเรียนคนไหนตอบผิดหรือตอบถูก

ในส่วนของบทเรียนโปรแกรมนี้จะมีรูปแบบการนำเสนอในรูปแบบเป็นคู่มือของอาจารย์ผู้สอน โดยแต่ละส่วนของบทเรียนจะเป็นหัวข้อหลักๆ ที่อาจารย์ใช้สอน และในแต่ละหัวข้อหลักๆ จะเป็นเนื้อหาที่เป็นใจความสำคัญของหัวข้อนั้นๆ มีรูปภาพประกอบคำอธิบาย และมีให้เลือกเสียงอธิบายได้ในแต่ละหัวข้อ ในแต่ละบทเรียนจะมีคู่มือที่เป็นเนื้อหาอธิบายหัวข้อต่างๆ ให้กับอาจารย์ผู้สอนใช้ประกอบการบรรยาย

โครงการนี้จะใช้โปรแกรมสำหรับสร้างสื่อประสม HURESS ของโครงการวิจัยร่วมระหว่างไทยญี่ปุ่นในการพัฒนาทรัพยากรมนุษย์ด้วยระบบการสอนทางไกลแบบสองทาง

พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ ในการสร้างโปรแกรมช่วยสอนวิชา ไมโคร
โพรเซสเซอร์ Z80

1.3 เนื้อหาโดยสังเขป

เนื้อหาทั้งหมดของปฏิญานิพนธ์ฉบับนี้ ประกอบด้วย

บทที่ 2 กล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องกับโครงการ ประกอบด้วยประวัติ
ความเป็นมา ความหมาย ประเภท การสร้างและประโยชน์ของคอมพิวเตอร์ช่วยสอน รวมถึง
หลักการและรายละเอียดการใช้งานในโปรแกรม HURESS

บทที่ 3 การออกแบบ การสร้าง และการทำงาน โดยกล่าวถึงการอธิบาย ขั้นตอนการ
ออกแบบบทเรียนคอมพิวเตอร์ช่วยสอน วิธีการสร้างบทเรียนคอมพิวเตอร์ช่วยสอนในแต่ละ
ส่วน ด้วยโปรแกรม HURESS และการทำงานในแต่ละส่วนที่สร้างขึ้น

บทที่ 4 การทดลองและผลการทดลอง เป็นการกล่าวนำเนื้อหาในบทเรียนแต่ละส่วน
และผลการทดสอบประสิทธิภาพของบทเรียนคอมพิวเตอร์ช่วยสอน

บทที่ 5 กล่าวสรุปการทำปฏิญานิพนธ์ ปัญหาที่เกิดขึ้นตั้งแต่เริ่มทำปฏิญานิพนธ์
วิธีการแก้ปัญหาที่เกิดขึ้น ข้อเสนอแนะและแนวทางในการนำปฏิญานิพนธ์ไปพัฒนาต่อไป

ภาคผนวก ก กล่าวถึงการติดตั้งโปรแกรม HURESS ในส่วนนี้จะประกอบด้วยขั้น
ตอนวิธีการติดตั้งโปรแกรม HURESS ตามลำดับขั้นตอน

ภาคผนวก ข กล่าวถึงการใช้โปรแกรมย่อยในการกำหนดการทำงานของหนังสือ ซึ่ง
เป็นส่วนที่อธิบายหลักการการเขียนคำสั่งควบคุมการทำงานต่างๆ ของหนังสือ

บทที่ 2

ทฤษฎี และหลักการ

2.1 กล่าวนำ

เนื้อหาของปริญญาพันธึนบถนี้เป็นทฤษฎีและหลักการที่นำมาใช้ประกอบการสร้างครงงาน โดยประกอบด้วย ประวัตึความเป็นมาและหลักการของบถเรียนคอมพิวเตอร์ช่วยสอน และหลักการรายละเอียดการใช้งานในแต่ละส่วนของโปรแกรม HURESS ซึ่งเนื้อหาในบถนี้จะใช้เป็นพื้นฐานในการนำไปสร้างและออกแบบโปรแกรมช่วยสอนซึ่งมีเนื้อหาในบถต่อไป

2.2 ประวัตึความเป็นมาของคอมพิวเตอร์ช่วยสอน

วิวัฒนาการของบถเรียนคอมพิวเตอร์ช่วยสอนมีลำดับดังนี้

ปี ค.ศ. 1950 ศูนย์วิจัยของ IBM ได้ริเริ่มนำบถเรียนคอมพิวเตอร์ช่วยสอนเพื่อใช้กับงานวิจัยด้านจิตวิทยา นับว่าเป็นการบุกเบิกด้านนี้เป็นแห่งแรก

ต่อมา ปี ค.ศ. 1958 มหาวิทยาลัย Florida ได้พัฒนาบถเรียนคอมพิวเตอร์ช่วยสอนเพื่อใช้ทบทวนวิชาฟิสิกส์และสถิติ พร้อมๆ กับมหาวิทยาลัย Stanford ได้นำคอมพิวเตอร์มาช่วยสอนวิชาภาษาอังกฤษและคณิตศาสตร์ในระดับมัธยมศึกษา

ปี ค.ศ. 1960 มหาวิทยาลัย Illinois ได้จัดทำบถเรียนคอมพิวเตอร์ช่วยสอนในสาขาวิชาจิตวิทยาการศึกษา และด้านวิศวกรรมศาสตร์โดยตั้งชื่อว่า PLATO CAI (Programmed Learning for Automated Teaching Operations CAI)

ต่อมาเมื่อคอมพิวเตอร์แพร่หลายมากขึ้น ได้มีการส่งเสริมพัฒนาบถเรียนคอมพิวเตอร์ช่วยสอนมากขึ้น ในปี ค.ศ. 1967 มีบถเรียนคอมพิวเตอร์ช่วยสอนจำหน่ายมากกว่า 1500 เรื่อง

ในปี ค.ศ. 1970 บถเรียนคอมพิวเตอร์ช่วยสอนได้เริ่มแพร่หลายในทวีปยุโรป องค์กรการ French National Experiment in Educational Computing ของฝรั่งเศส ได้เริ่มพัฒนาบถเรียนคอมพิวเตอร์ช่วยสอนสำหรับผู้สอนในระดับมัธยมศึกษาพร้อมๆ กับประเทศอังกฤษได้ริเริ่มโครงการพัฒนาบถเรียนคอมพิวเตอร์ช่วยสอนมากกว่า 35 โครงการ

ปี ค.ศ. 1971 มหาวิทยาลัย Brigcam Young ร่วมกับมหาวิทยาลัย Texas ได้พัฒนาบถเรียนคอมพิวเตอร์ช่วยสอนสำหรับกับมินิคอมพิวเตอร์โดยผสมผสานคอมพิวเตอร์กับ

เอกสารนเป็นเอกสารทสงวนไว้สําหรับการแข่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้มาไปเซประโยชนด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทรทัศน์เพื่อช่วยสอนวิชาภาษาอังกฤษและคณิตศาสตร์ ตั้งชื่อว่า TICCIT (Time Shared Interactive Computer Controlled Information Television)

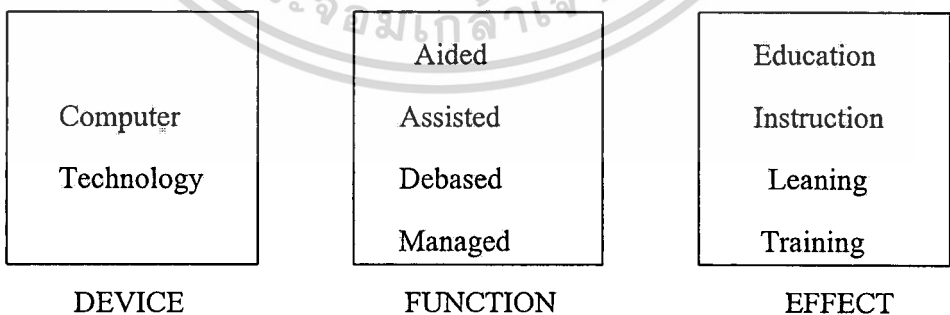
พัฒนาการของบทเรียนคอมพิวเตอร์ช่วยสอนกับเครื่องคอมพิวเตอร์ขนาดใหญ่ นับตั้งแต่ต้นมา จะประสบความสำเร็จในระดับหนึ่งเท่านั้น ต่อมาเมื่อไมโครคอมพิวเตอร์เริ่มเข้ามา มีบทบาทในสถานศึกษามากขึ้น จึงมีการใช้บทเรียนคอมพิวเตอร์ช่วยสอนกันอย่างจริงจัง เช่น มหาวิทยาลัย Osaka ในประเทศญี่ปุ่น เป็นต้น

ปัจจุบันบทเรียนคอมพิวเตอร์ช่วยสอนได้มีบทบาทสำคัญต่อการเรียนการสอนทั้งในห้องเรียน และการฝึกอบรมในสถานประกอบการ โดยเฉพาะเทคโนโลยีคอมพิวเตอร์ในปัจจุบันสามารถประยุกต์ใช้งานได้ทั้งภาพ เสียง และการโต้ตอบ ในลักษณะสื่อประสม (Multi Media) ทำให้แนวโน้มของบทเรียนคอมพิวเตอร์ช่วยสอน ได้แสดงบทบาทในการประยุกต์ใช้งานได้อย่างเต็มที่

2.2.1 ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน

(Computer Assisted Instruction : CAI) เป็นคำศัพท์เดิมนิยมใช้ในสหรัฐอเมริกา มีความหมายว่า การสอนโดยใช้คอมพิวเตอร์เป็นเครื่องช่วยสอน แต่ปัจจุบันมีผู้นิยมใช้ คำว่า CBT (Computer Based Teaching หรือ Computer Based Training) มากกว่า ความหมายก็คือ การสอนหรือการฝึกอบรมโดยใช้คอมพิวเตอร์เป็นหลัก

นอกจากนี้ยังมีคำอื่นๆ ที่กำหนดขึ้นมาอีก ซึ่งส่วนใหญ่แล้วจะประกอบด้วย 3 ส่วน ได้แก่ เครื่องมือ (Device) หน้าที่ (Function) และผลที่เกิด (Effect) ซึ่งสรุปได้ดังรูปที่ 2.1



รูปที่ 2.1 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน

ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน จึงตรงกับคำต่อไปนี้

(Computer Assisted Instruction หรือ Computer Aided Instruction : CAI)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Computer Based Teaching หรือ Computer Based Training : CBT)

(Computer Based Education : CBE)

(Computer Managed Instruction : CMI)

(Computer Managed Learning : CML)

สำหรับประเทศไทยนั้นจะคุ้นเคยกับคำว่า CAI มากกว่าคำอื่นๆ ส่วนภาษาไทยนั้น จะใช้คำว่าบทเรียนคอมพิวเตอร์ช่วยสอน บทเรียนช่วยสอนด้วยคอมพิวเตอร์ บทเรียน โปรแกรมคอมพิวเตอร์ บทเรียนสำเร็จรูปด้วยคอมพิวเตอร์หรืออื่นๆ

ความหมายของบทเรียนคอมพิวเตอร์ช่วยสอน คือ บทเรียนสำเร็จรูปที่นำเสนอเนื้อหา สื่อ กิจกรรม การตรวจปรับการประเมินผล และกระบวนการเกี่ยวกับการเรียนรู้อื่นๆ ด้วยคอมพิวเตอร์

2.2.2 ลักษณะการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอน

บทเรียนคอมพิวเตอร์ช่วยสอน เป็นการประยุกต์ใช้คอมพิวเตอร์เพื่อการศึกษาในรูปแบบของบทเรียนสำเร็จรูป เพื่อสนับสนุนการเรียนการสอนรายบุคคลทั้งในห้องเรียนของสถาบัน การศึกษาและการฝึกอบรมในสถานประกอบการ ลักษณะการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอนจึงเหมาะสมกับสถานการณ์ต่อไปนี้

1. ใช้เพื่อสอนแทนผู้สอนทั้งในและนอกห้องเรียน เช่น การสอนแทนผู้สอนโดยตรง การสอนทบทวน การสอนเสริม เป็นต้น
2. ใช้เพื่อการศึกษาทางไกลผ่านสื่อโทรคมนาคม เช่น การเรียนการสอนทางไกล (Distance Learning)
3. ใช้กับเนื้อหาการสอนที่ซับซ้อน ไม่สามารถศึกษาได้จากของจริงโดยตรง เช่นการเปลี่ยนแปลงภายในโมเลกุล
4. ใช้กับลักษณะงานที่อันตราย มีความเสี่ยงต่อความเสียหายสูง โดยเฉพาะอย่างยิ่ง การฝึกอบรม เช่น การจำลองระบบการบิน (Flight Simulator)
5. ใช้กับสาระเนื้อหาที่ต้องการแสดง ให้เห็นลำดับขั้นตอนการเปลี่ยนแปลงทีละขั้นๆ โดยการจำลองจากเหตุการณ์จริงที่เกิดขึ้นเร็วไปหรือช้าเกินไป
6. ใช้ในการฝึกอบรมพนักงานใหม่ โดยไม่ต้องเสียเวลาเริ่มงานเหมือนกับการฝึกอบรมแบบปกติ

7. ใช้เพื่อคงความเป็นมาตรฐาน ทั้งหลักสูตรการสอนและการฝึกอบรมให้เหมือนกันทุกแห่งที่ใช้หลักสูตรเดียวกัน

2.2.3 ข้อได้เปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน

จากผลการวิจัยในการใช้งานของบทเรียนคอมพิวเตอร์ช่วยสอน สรุปได้ว่า

1. บทเรียนคอมพิวเตอร์ช่วยสอน ทำให้ผลสัมฤทธิ์ทางการเรียนสูงขึ้นเมื่อเปรียบเทียบกับการเรียนการสอนในห้องเรียน
2. ลดเวลาเรียนลง เมื่อเปรียบเทียบกับการเรียนการสอนในห้องเรียน
3. ผู้เรียนสนใจการเรียนมากขึ้นเมื่อเรียนด้วยบทเรียนคอมพิวเตอร์ช่วยสอน
4. ผู้เรียนจะมีปฏิสัมพันธ์ (Interactive) กับบทเรียนอย่างแท้จริงโดยมีการโต้ตอบซึ่งกันและกัน ทำให้เกิดการเรียนรู้อย่างลึกซึ้ง
5. ผู้เรียนเป็นผู้ควบคุมบทเรียนด้วยตนเอง นับตั้งแต่การจัดการบทเรียนเลือกเรียนกิจกรรมที่ตนถนัด จนถึงประเมินผลการเรียนด้วยตนเอง
6. บทเรียนคอมพิวเตอร์ช่วยสอนเสนอเนื้อหาได้รวดเร็วฉับไว เก็บเนื้อหาไว้ได้มากกว่า เสนอรูปภาพที่เคลื่อนไหวซับซ้อนและมีเสียงประกอบได้
7. สามารถนำติดตัวไปเรียนในสถานที่ต่างๆ โดยไม่มีข้อจำกัดด้านเวลาและสถานที่

2.2.4 ข้อเสียเปรียบของบทเรียนคอมพิวเตอร์ช่วยสอน

ข้อเสียเปรียบของบทเรียนคอมพิวเตอร์ช่วยสอนเมื่อเปรียบเทียบกับการเรียนการสอนปกติ จำแนกออกเป็น 4 ประเด็นได้ดังนี้

1. ค่าใช้จ่ายการพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอน จำเป็นต้องลงทุนค่าใช้จ่ายค่อนข้างสูง ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์
2. ผู้เชี่ยวชาญ ต้องจัดเตรียมผู้เชี่ยวชาญหลายด้านมาระดมความคิดเพื่อพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอน ทั้งด้านหลักสูตร การเรียนการสอน สื่อการสอน การวัดและประเมินผล และด้านการโปรแกรมคอมพิวเตอร์
3. ระยะเวลาในการพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอน จะต้องใช้เวลามากสำหรับการพัฒนา การทดสอบ และการปรับปรุงบทเรียน
4. ความยากในการออกแบบ เนื่องจากบทเรียนคอมพิวเตอร์ช่วยสอนจะต้องออกแบบให้ยืดหยุ่นต่อการใช้งาน มีทางเลือกที่เหมาะสมกับผู้เรียนที่มีความถนัดแตกต่างกันจึงเป็นการยากที่จะออกแบบเนื้อหาให้สอดคล้องกับกลุ่มเป้าหมายที่มีความแตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 ส่วนประกอบของบทเรียนคอมพิวเตอร์ช่วยสอน ประกอบด้วย

1. ฮาร์ดแวร์ ได้แก่ ตัวเครื่องและอุปกรณ์ที่ใช้เป็นทางผ่านของบทเรียนประกอบด้วย จอภาพ แป้นพิมพ์ เครื่องขับแผ่นดิสก์ เครื่องอ่าน CD-ROM ลำโพง และอื่นๆ

2. ซอฟต์แวร์ ได้แก่ โปรแกรมที่ใช้จัดการและนำเสนอบทเรียน เรียกว่า Authoring System เป็นโปรแกรมสำเร็จรูปที่ออกแบบมาเพื่อการออกแบบบทเรียนคอมพิวเตอร์ช่วยสอน โดยตรง เช่น Authorware, Icon Author, Tencor หรืออาจใช้ภาษาคอมพิวเตอร์พัฒนาบทเรียน ก็ได้

2.2.6 ลักษณะโครงสร้างของบทเรียนคอมพิวเตอร์ช่วยสอน

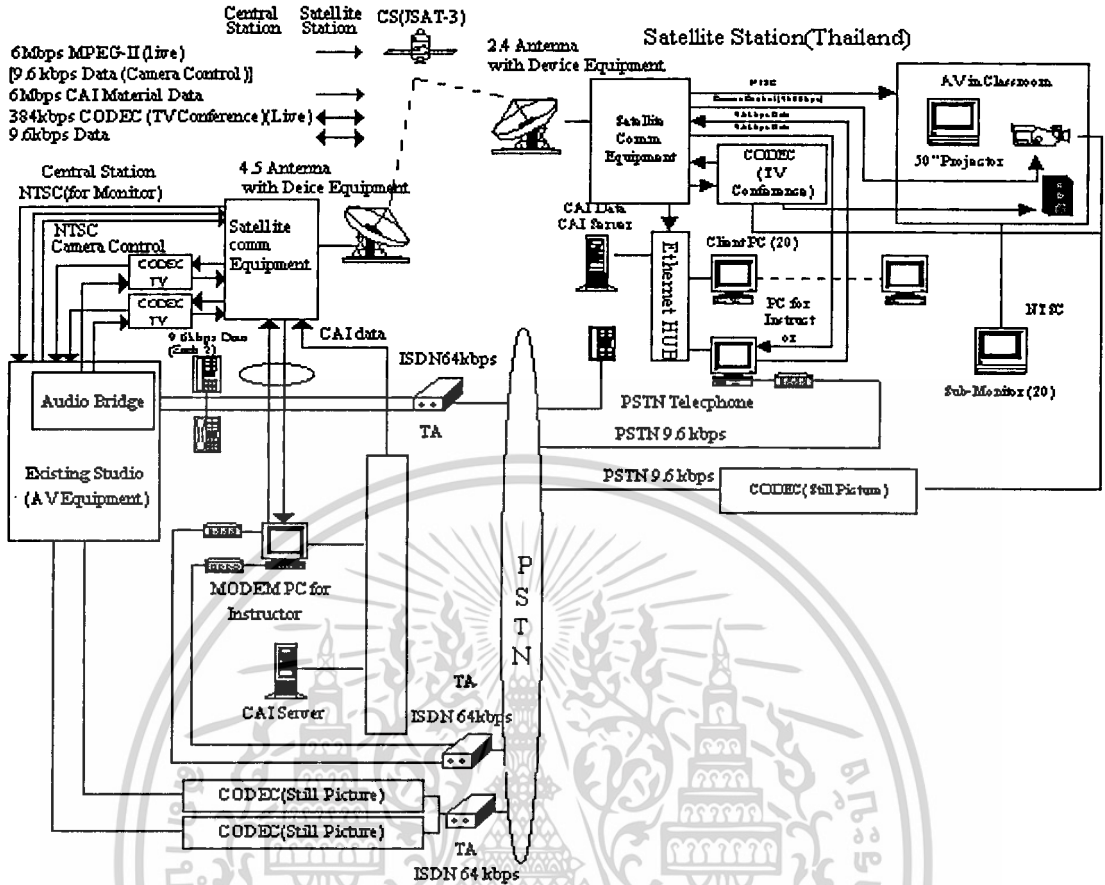
บทเรียนคอมพิวเตอร์ช่วยสอนเป็นการนำคอมพิวเตอร์มาใช้ในการเรียนการสอนการ ทบทวนการทำแบบฝึกหัดหรือการวัดผล ผู้เรียนแต่ละคนจะนั่งอยู่หน้าเครื่องคอมพิวเตอร์ เรียกโปรแกรมบทเรียนที่เตรียมไว้สำหรับการสอนในวิชานั้นๆ ขึ้นมาแสดงบนหน้าจอภาพ และอ่านทำความเข้าใจได้ตรงกับบทเรียน ตามการจัดการของบทเรียน จนจบบทเรียน

ลักษณะสำคัญของบทเรียนคอมพิวเตอร์ช่วยสอนบทเรียนหนึ่งๆ จะประกอบด้วย สาระสำคัญๆ 5 ประการดังนี้

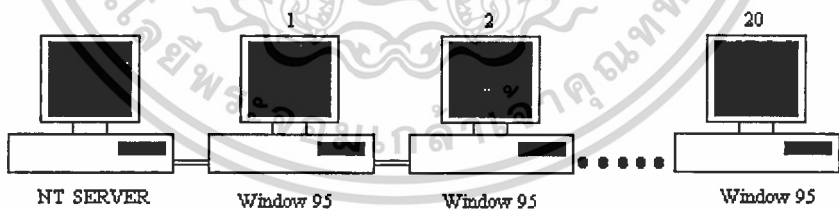
1. การสนับสนุนการเรียนการสอนรายบุคคล
2. การโต้ตอบระหว่างผู้เรียนกับบทเรียน
3. รูปแบบการนำเสนอบทเรียน
4. การจัดการบทเรียน
5. ประสิทธิภาพในการใช้งานของบทเรียน

2.2.7 การสอนทางไกลใน โครงการวิจัยร่วมญี่ปุ่นและจีน ในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบการสอนทางไกลแบบสองทางพร้อมสื่อประสมดาวเทียม ระหว่างประเทศ

เทคโนโลยีทางการสื่อสาร โทรคมนาคมและคอมพิวเตอร์ได้ถูกพัฒนาขึ้นเป็น อย่างมากในช่วง 10 ปีที่ผ่านมา ไม่ว่าจะเป็นโทรศัพท์มือถือ เครื่องเล่นวีดีโอแบบเลเซอร์ โทรทัศน์ระบบดิจิตอล หรือ HDTV (High Definition TeleVision) ระบบดาวเทียมแบบ ดิจิตอล



รูปที่ 2.2 ระบบสอนทางไกลแบบสองทางพร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ



รูปที่ 2.3 เครื่องข่ายของระบบ

หรือ DSS (Digital Satellite System) คอมพิวเตอร์แบบสื่อประสม (Multimedia Computer) ตลอดจนระบบเครือข่ายคอมพิวเตอร์ขนาดใหญ่ที่รู้จักกันดีในนามของ อินเทอร์เน็ต (Internet) เทคโนโลยีเหล่านี้ต่างๆ ได้รับการได้รับการพัฒนาขึ้นเพื่อทำให้บริการและตอบสนองความต้องการของมนุษย์ในรูปแบบต่างๆ รวมทั้งเพื่อการแข่งขันกันทางการตลาด แห่งยุค-โลกาภิวัตน์ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน่วยงานทั้งรัฐบาลและเอกชน ต่างก็พยายามที่จะปรับปรุงรูปแบบการทำงานและการให้บริการของตน ให้มีลักษณะของการนำเทคโนโลยีสารสนเทศ (Information technology) เข้ามาใช้ในองค์กรกันมากขึ้น โดยพยายามนำเทคโนโลยีสื่อสารและคอมพิวเตอร์เข้ามาประยุกต์ เพื่อให้เกิดประสิทธิภาพการทำงานในองค์กร โดยเฉพาะอย่างยิ่งหน่วยงานทางการศึกษา ที่ต่างก็พยายามพัฒนารูปแบบการเรียนการสอนของตน ให้สามารถบริการนักศึกษา และลดปัญหาต่างๆ ที่เป็นอุปสรรคต่อการเรียนการสอนลง เช่น ปัญหาการขาดแคลนอาจารย์ ปัญหาเรื่องระยะทาง เป็นต้น รวมทั้งเพื่อตอบสนองนโยบายของรัฐบาลในการที่กระจายโอกาสทางการศึกษาไปสู่ชนบทได้อย่างทั่วถึง โดยอาศัยเทคโนโลยีเข้ามาช่วยขยายบริการเพื่อให้ผู้เข้าเรียนเข้าถึงบริการการศึกษามากที่สุด

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นหนึ่งในหน่วยงานการศึกษาของรัฐที่ได้พยายามพัฒนารูปแบบการเรียนการสอนให้สอดคล้องกับการเปลี่ยนแปลงของเทคโนโลยีเพื่อตอบสนองนโยบายของรัฐบาล โดยทำการวิจัยเรื่องระบบการเรียนการสอนทางไกลในโครงการต่างๆ ด้วยเทคโนโลยีที่แตกต่างกันไป เช่น การใช้ระบบการประชุมแบบเห็นภาพทางเครื่องคอมพิวเตอร์ (Desktop Video Conferencing) บนเครือข่ายผ่านดาวเทียมในโครงการ POST - Partners การใช้ระบบการสอนทางไกลผ่านเครือข่ายใยแก้วนำแสงในโครงการวิทยาเขตสารสนเทศของทบวงมหาวิทยาลัย และการใช้ระบบการสอนทางไกลแบบสื่อประสมผ่านดาวเทียมในโครงการวิจัยร่วมระหว่างไทย-ญี่ปุ่น เป็นต้น นอกจากนี้สถาบันฯ ยังได้นำเทคโนโลยีเครือข่ายคอมพิวเตอร์เข้ามาใช้เพื่อพัฒนารูปแบบการศึกษาและสร้างงานวิจัยใหม่ๆ ขึ้นในสถาบัน อันได้แก่ เทคโนโลยีเครือข่ายระบบเปิด (Open System) และระบบเครือข่ายภายในสถาบันฯ (Campus Network) เป็นต้น เทคโนโลยีดังกล่าวนี้เป็นโครงสร้างพื้นฐานสำหรับการเชื่อมโยงเครือข่ายคอมพิวเตอร์ภายในสถาบันฯ ให้สามารถติดต่อสื่อสารกับอินเทอร์เน็ต (Internet) และโลกภายนอกโดยตรง

โครงการสำนักนวัตกรรมและเทคโนโลยีการศึกษา เป็นหน่วยงานใหม่ของสถาบันที่ถูกจัดตั้งขึ้นเพื่อตอบสนองนโยบายและแนวทางดังกล่าว โดยใช้เป็นศูนย์รวมในการให้บริการผลิตโปรแกรมบทเรียน เพื่อสนับสนุนการสอนแก่คณาจารย์ในสถาบันฯ รวมทั้งเป็นศูนย์วิจัยพัฒนาและให้บริการทางด้านเทคโนโลยีทางการศึกษา เพื่อพัฒนาศักยภาพและคุณภาพทรัพยากรมนุษย์ให้เกิดประโยชน์ ในการถ่ายทอดความรู้ทางวิชาการสูงสุด ปัจจุบัน

โครงการสำนักนวัตกรรมฯ รับผิดชอบในการพัฒนาทรัพยากรมนุษย์ด้วยสื่อประสมผ่านดาวเทียม ไทย-ญี่ปุ่น ซึ่งโครงการทั้งสองเป็นโครงการที่จะนำเทคโนโลยีเข้ามาใช้เพื่อประสิทธิภาพทางการศึกษาเป็นหลัก และพัฒนาการเรียนการสอนของสถาบันฯ

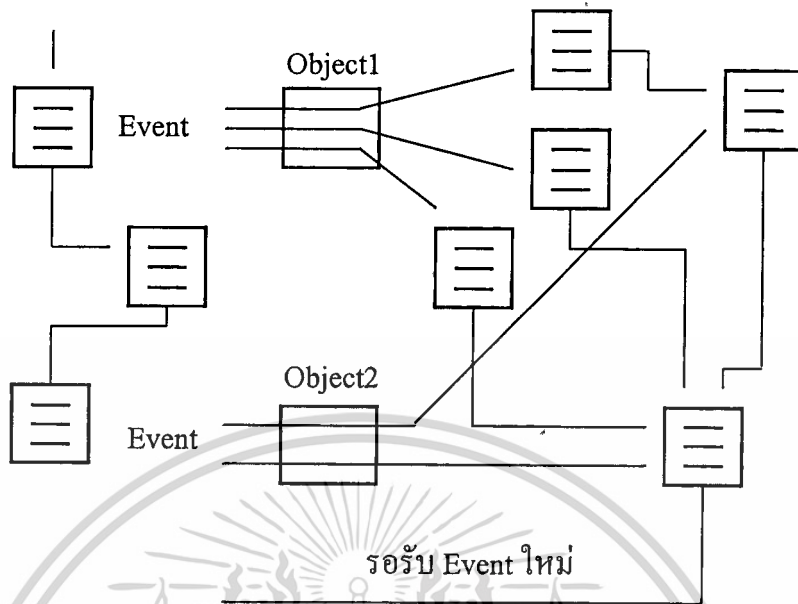
จากโครงสร้างของแต่ละระบบเดิม ทำให้การใช้งานถูกจำกัดอยู่เฉพาะในแต่ละส่วน โดยไม่สามารถกระจายโอกาสไปยังกลุ่มเป้าหมายได้เท่าเทียมกันทุกภาควิชาและคณะในสถาบันฯ เนื่องจากขาดการรวมหรือเชื่อมต่อกับเทคโนโลยีต่างๆ ที่มีอยู่เข้าด้วยกันเพื่อให้เกิดการใช้งานได้สูงสุด ดังนั้นโครงการสำนักนวัตกรรมฯ จึงได้มีแนวคิดที่จะสร้างระบบเครือข่ายสำนักนวัตกรรมฯ ขึ้น โดยรวมเทคโนโลยีต่างๆ ที่มีอยู่เดิมด้วยการเชื่อมต่อเครือข่ายของแต่ละระบบในแต่ละโครงการเข้าด้วยกัน พร้อมนำเสนอการใช้งานและการให้บริการทางการศึกษาในรูปแบบต่างๆ ที่จะเกิดขึ้นบนเครือข่ายใหม่ เช่น บริการด้านการเรียนการสอนทางไกลผ่านเครือข่ายในรายวิชาต่างๆ บริการสื่อการสอน บริการสอนทางไกลผ่านอินเทอร์เน็ต บริการการประชุมแบบเห็นภาพบนเครื่องคอมพิวเตอร์ บริการวีดิโอตามคำขอและบริการมหาวิทยาลัยเสมือน เป็นต้น

2.3 หลักการของโปรแกรม HURESS

โปรแกรม HURESS เป็นโปรแกรมเพื่อพัฒนาสื่อมัลติมีเดีย ถูกออกแบบมาเป็นสื่อของโครงการวิจัยร่วมญี่ปุ่นและจีน ในการพัฒนาทรัพยากรมนุษย์ ด้วยระบบการสอนทางไกลแบบสองทางพร้อมสื่อประสมดาวเทียมระหว่างประเทศโดยเฉพาะ โปรแกรม HURESS ใช้หลักของภาพและการมองเห็น โดยเริ่มจากออกแบบวินโดว์ย่อย วินโดว์ย่อยจะประกอบด้วยสิ่งต่างๆ ที่เราจะทำงานด้วยหรือเรียกว่าเป็น วัตถุ (Object) เช่น ข้อความ ช่องรับข้อความ Scroll Bar หรือปุ่ม (Button) เมื่อกำหนดสิ่งเหล่านี้ครบตามความต้องการแล้วจึงระบุว่าจะประกอบแต่ละอย่างจะทำงานอย่างไร โดยเขียนโปรแกรมย่อยๆ ปะเข้าไปที่วัตถุเหล่านี้ที่ต้องทำแบบนี้เพราะว่าการทำงานในวินโดว์เป็นแบบที่เรียกว่า อีเวนต์ไดรฟ์ (Event-Driven) คือ ขึ้นกับเหตุการณ์

จากแนวทางนี้จะเห็นว่ามัลติมีเดียเป็นธรรมชาติมาก คือ กำหนดหน้าตาของจอภาพหรือส่วนติดต่อกับผู้ใช้ และระบุว่าถ้าเกิดเหตุการณ์อย่างนี้กับสิ่งนี้จะต้องทำอย่างไร ซึ่งโปรแกรมที่เราจะเขียนก็คือส่วนที่จะต้องทำอย่างไรนั่นเอง ดังรูปที่ 2.4

เริ่มต้น



รูปที่ 2.4 การเขียนโปรแกรมแบบธรรมดาและ การเขียนโปรแกรมแบบ Event-driven

โดยสรุปแล้วรูปแบบของหลักการใน HURESS คือ เริ่มจากออกแบบจอภาพและเขียนโปรแกรมสำหรับแต่ละอีเวนต์ ปะเข้าไปยัง ออปเจ็ค ต่างๆ ให้ทำงานตามเหตุการณ์ที่เกิดขึ้น

2.4 ส่วนประกอบของโปรแกรม HURESS

2.4.1 Books

ส่วนที่ถือว่าเป็นส่วนสำคัญหลักในโปรแกรม HURESS ก็คือ หนังสือ (Book) หนังสือจะถูกเก็บไว้ในชั้นหนังสือ ที่เรียกว่า Bookshelf หนังสือในโปรแกรมนี้อถือว่าเป็นหนังสืออิเล็กทรอนิกส์ ซึ่งในแต่ละหน้าจะสามารถเรียกใช้อุปกรณ์ต่างๆ เพื่อสร้าง ตัวหนังสือ รูปภาพ เสียง รวมถึงการเรียกใช้ Background ต่างๆ ได้

2.4.2 Page

ในส่วนของหน้าหนังสือ จะเป็นส่วนที่แสดงเนื้อหาต่างๆ ของหนังสือ ในแต่ละหน้าหนังสือ สามารถที่จะเรียกใช้เครื่องมือในการสร้าง ตัวหนังสือ ภาพ เสียง ปุ่มต่างๆ ได้ตามแต่ต้องการ

ในส่วนของประเภทหน้าหนังสือในโปรแกรม HURESS นั้น จะประกอบไปด้วย 5 ประเภทด้วยกัน คือ

1. หน้าปกหนังสือ (Cover)
2. หน้าสารบัญ (Table of Contents)
3. เนื้อหาหลัก (Main Text)
4. หน้าดัชนี (Index)
5. หน้าที่ท้ายของหนังสือ (Back Cover)

2.4.3 Media

ในส่วนนี้จะเป็นส่วนของเครื่องมืออำนวยความสะดวก ในการสร้าง Media ใช้งาน ต่างๆ ในโปรแกรม เช่น ตัวอักษร รูปภาพ เสียง ปุ่มการใช้งาน ฯลฯ

2.4.4 Desk

ในส่วนนี้จะเป็นส่วนหลักในการใช้งานโปรแกรมในลักษณะ โครงสร้างโปรแกรม เช่น การยืมหนังสือ การสร้างชั้นหนังสือ การปิดชั้นหนังสือ การเช็คหน้าจอ การเช็คเมนูบาร์ ต่างๆ ฯลฯ

2.4.5 Bookshelf

เป็นส่วนในการใช้เก็บหนังสือ ชั้นหนังสือสามารถสร้างขึ้นกี่ชั้นก็ได้ แล้วแต่เรา ต้องการเก็บหนังสืออย่างไร

2.4.6 Entity

เป็นส่วนของรูปแบบฟังก์ชันการใช้งานในลักษณะต่างๆ ตามเมนูเลือกใช้งานหรือใน รูปแบบของไอคอน ซึ่งการใช้งานจะทำงานตามฟังก์ชัน โดยการใช้เมาส์คลิกที่ฟังก์ชันนั้น

2.4.7 Procedures

เป็นส่วนการลำดับการทำงานของแต่ละฟังก์ชัน โดยในส่วนนี้จะมีการเขียนลำดับการทำงานในรูปของการเขียนโปรแกรมการทำงาน ให้กับฟังก์ชันที่เราต้องจะให้ทำ

2.4.8 Objects

โปรแกรม HURESS จะสนับสนุน กับ OLE2, ซึ่ง OLE2 เป็นลักษณะฟังก์ชันการแลกเปลี่ยนข้อมูลข่าวสารต่างๆ ใน Windows อื่นๆ ได้

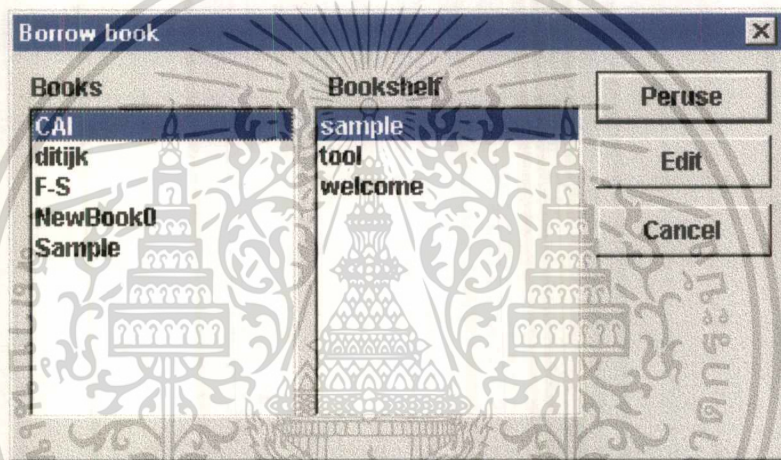
2.5 หลักการใช้โปรแกรม HURESS

ในหัวข้อนี้จะกล่าวถึงการใช้หนังสือ ในโปรแกรม HURESS ซึ่งจะมีลำดับการใช้เป็นลำดับตามหัวข้อต่อไปนี้

2.5.1 การยืมและเปิดหนังสือ (Borrowing and Opening a Book)

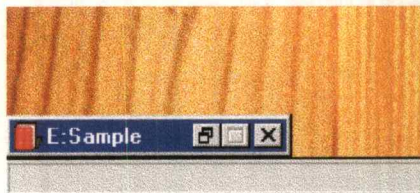
1. การยืมหนังสือผ่าน Desk Main Menu

การยืมในส่วนนี้จะยืมจาก ชั้นวางหนังสือ (Bookshelf) บน Desk Main Menu เมื่อทำการยืมจะปรากฏ Dialog Box ดังรูปที่ 2.5



รูปที่ 2.5 การยืมหนังสือผ่าน Desk Main Menu

จากชั้นวางหนังสือ (Bookshelf) หากยืมหนังสือ Sample ในชั้นวางหนังสือ Sample หนังสือที่ถูกยืมก็จะปรากฏเป็น Dialog Box ตรงมุมล่างซ้ายของ Desk Screen ในลักษณะ Minimize Form ดังรูปที่ 2.6

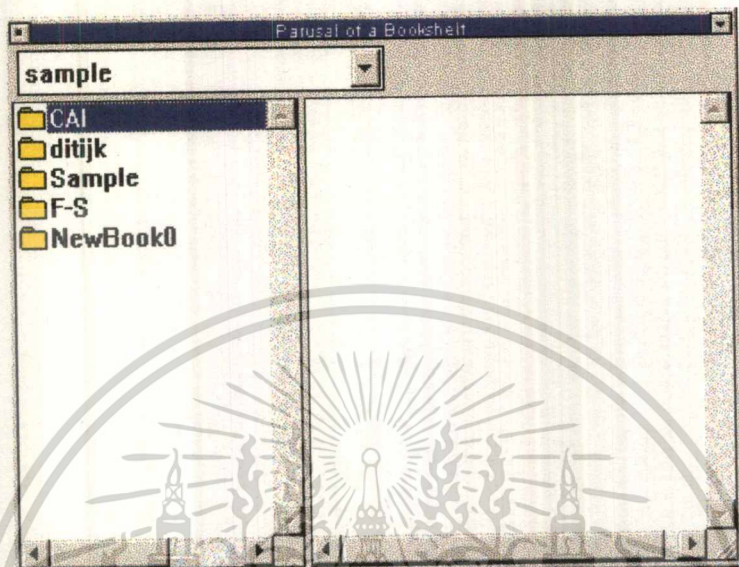


รูปที่ 2.6 หนังสือที่ถูกยืม

2. การยืมหนังสือ โดยผ่าน Bookshelf Browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Desk Main Menu นำเมาส์ไปคลิกที่ Bookshelf แล้วคลิกเลือกที่ Peruse Bookshelf ก็จะปรากฏหนังสือที่เราต้องการยืมดังรูปที่ 2.7



รูปที่ 2.7 การยืมหนังสือผ่าน Bookshelf Browser

3. การยืมโดยใช้ เมนู Selection Bar

นำเมาส์ไปคลิกที่ ปุ่มดังรูป  ก็จะเข้าสู่การยืมหนังสืออีกแบบหนึ่ง

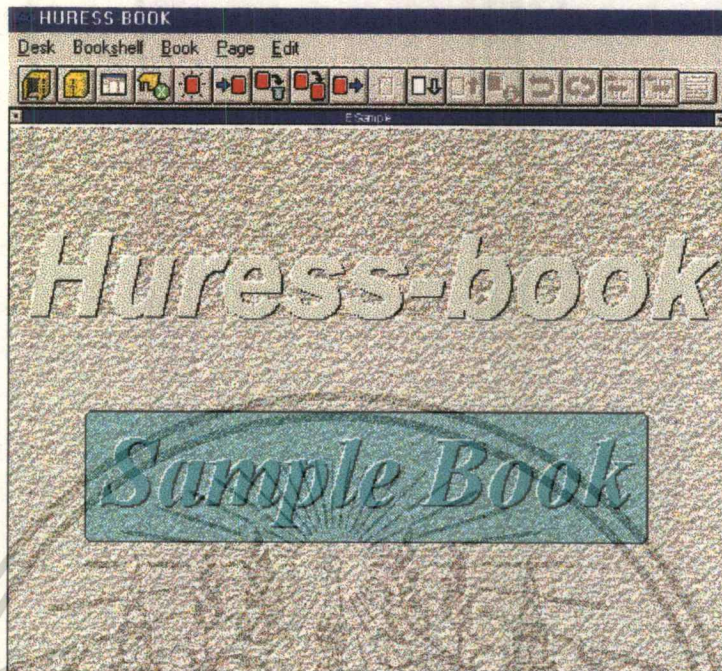
2.5.2. การเปิดหนังสือ Opening a Book

นำเมาส์ไปคลิกตรง Minimize Book มุมล่างด้านซ้ายของ Windows Desktop แล้วทำการคลิกปุ่ม Restore ดังรูปที่ 2.8 และหนังสือที่ถูกยืมก็จะปรากฏบนหน้าจอดังรูปที่ 2.9



รูปที่ 2.8 การเปิดหนังสือ

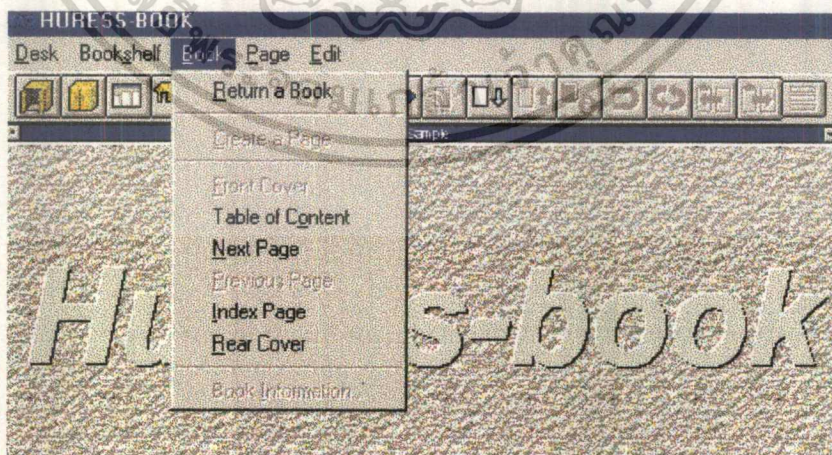
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 หนังสือที่ทำการเปิดได้แล้ว

2.5.3 การทำงานเกี่ยวกับหน้าหนังสือ

คลิกเมาส์ไปที่ Book บน Desk Main Menu ก็จะปรากฏหน้าจอดังรูปที่ 2.10



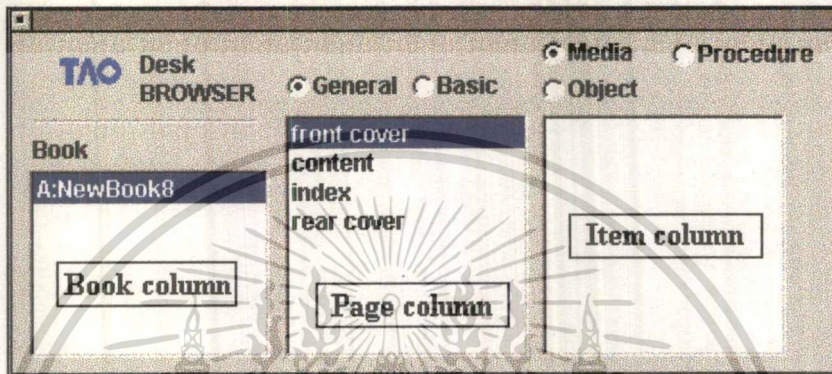
รูปที่ 2.10 การทำงานเกี่ยวกับหน้าหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการคลิกที่ Next Page หรือ Previous Page ส่วนของ Next Page จะเป็นการเลื่อนหน้าต่อไป ส่วน Previous Page จะเป็นการเลื่อนไปยังหน้าที่ผ่านมา

2.5.4 การใช้งาน Desk Browser

Desk Browser จะเป็นตัวแสดงรายละเอียดของหนังสือที่เราพิมพ์ ดังรูปที่ 2.11



รูปที่ 2.11 การใช้ Desk Browser

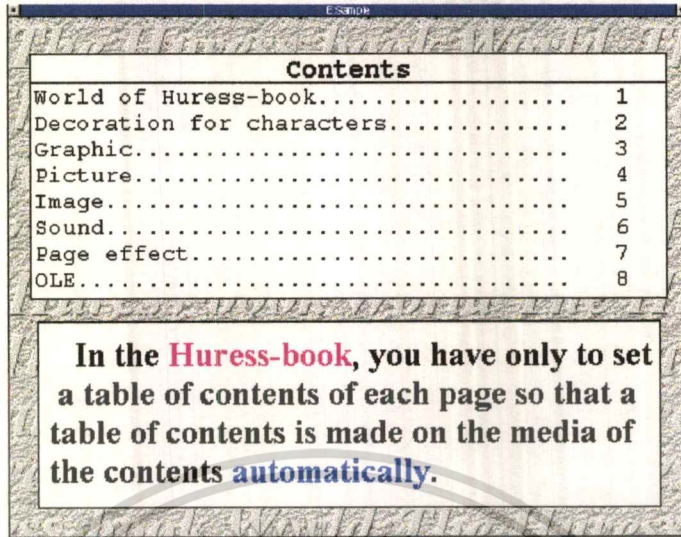
Book Column จะแสดงชื่อหนังสือที่เราเปิดอ่าน

Page Column จะแสดงหน้าต่างๆ ของหนังสือ

Item Column จะแสดงสื่อต่างๆ ที่เราใช้ประกอบในหนังสือเล่มนี้

2.5.5 การใช้หน้าสารบัญ

คลิกเมาส์ไปหน้าสารบัญ (Contents) ก็จะปรากฏหน้าจอดังรูปที่ 2.12

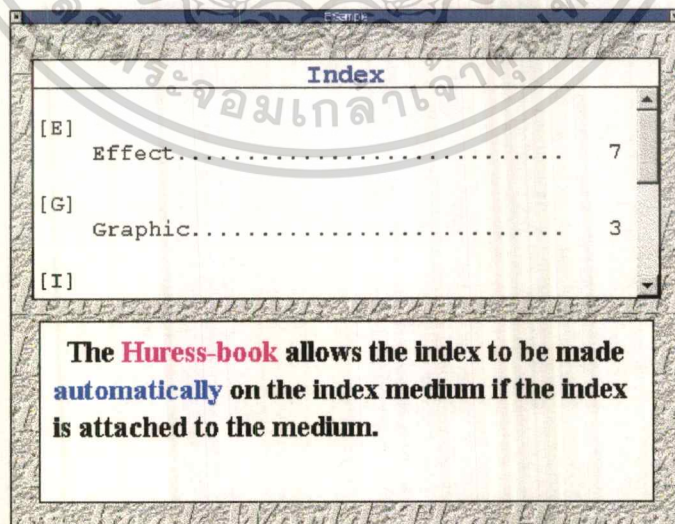


รูปที่ 2.12 การใช้หน้าสารบัญ

หน้าสารบัญนี้หากต้องการไปที่หน้าไหนก็ทำการ Double Click ที่หน้าที่ต้องการในตาราง Contents ได้เลย

2.5.6 การใช้งานหน้าดัชนี

คลิกเมาส์ไปที่หน้าดัชนี (Index) เมื่อต้องการจะไปหัวข้อใดในดัชนีก็ทำการ Double Click ที่หัวข้อในตารางดัชนีได้ หน้าดัชนีแสดงดังรูปที่ 2.13



รูปที่ 2.13 การใช้งานหน้าดัชนี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 การคืนหนังสือ

ให้ทำการคลิกเมาส์ไปยัง Return a Book บน Desk Main Menu หรือ คลิกเมาส์ไปที่



Menu Selection Bar หรือจะทำการคลิกซ้ายที่ Close Box บนมุมซ้ายบนของหนังสือ ดัง

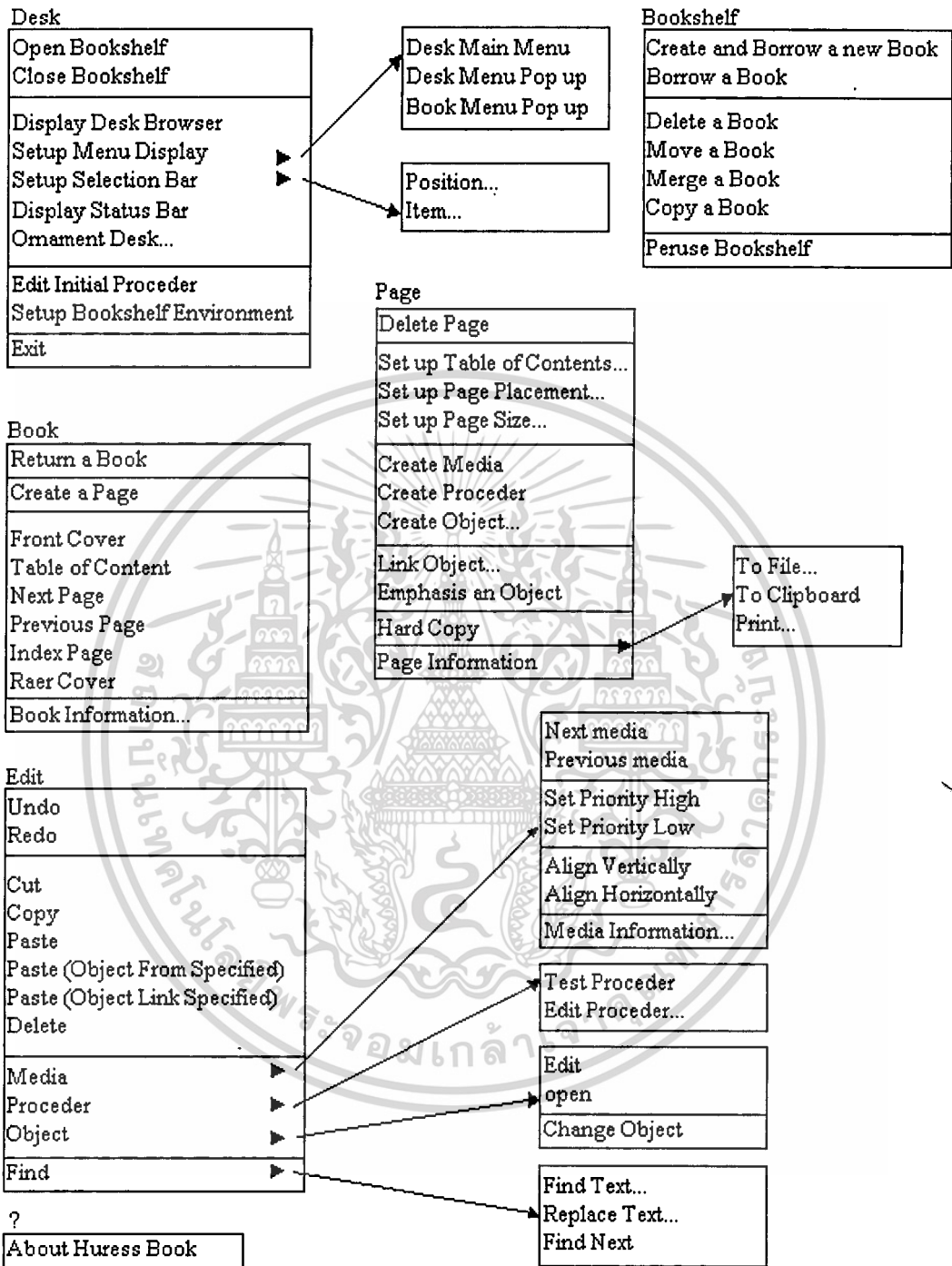
รูป 2.14



รูปที่ 2.14 การคืนหนังสือ โดย ปุ่ม Close Box

2.6 การทำงานในส่วนของ Desk Main Menu

การทำงานของโปรแกรม HURESS โดยผ่าน Desk Main Menu แสดงดังรูปที่ 2.15



รูปที่ 2.15 การทำงานในส่วนของ Desk Main Menu

2.6.1 การทำงานในส่วนของเมนูย่อย [Desk]

1. Open Bookshelf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเมนูที่ใช้สำหรับการเปิดชั้นวางหนังสือก่อนที่จะเปิดหนังสือที่เราต้องการในชั้นนั้นๆ

2. Close Bookshelf

เป็นเมนูที่ใช้สำหรับการปิดชั้นหนังสือ ก่อนการปิด โปรแกรม HURESS ทุกครั้ง

3. Display Desk Browser

เมื่อทำการเลือกเมนูนี้ ส่วนของ Desk Browser ก็จะปรากฏ

4. Setup Menu Display - Desk Main Menu

เป็นส่วนที่ใช้ Setup Desk Main Menu ที่จะให้แสดงผลส่วนไหนของจอภาพ

5. Setup Menu Display - Desk Menu Pop up

เป็นส่วนที่ใช้ Setup Desk Main Menu ที่จะให้ปรากฏบนจอภาพหรือไม่

6. Setup Menu Display - Book Menu Pop up

เป็นเมนูที่ใช้ Setup Book Menu Pop Up ที่จะให้ปรากฏบนจอภาพหรือไม่

7. Setup Selection Bar - Position

เป็นเมนูส่วนที่ใช้ Setup ตำแหน่งของ Selection Bar ที่จะให้ปรากฏบนหรือล่างของจอภาพ

8. Setup Selection Bar - Item

เป็นเมนูส่วนที่ใช้ Set Buttons ที่จะให้ปรากฏบน Menu Selection Bar หรือไม่

9. Display Status Bar

เป็นเมนูที่ใช้ Setup Bar แสดงสถานะของเครื่องมือ

10. Ornament Desk

เป็นเมนูที่ใช้ Setup Background ของ โปรแกรม HURESS

11. Edit Initial Procedure

เป็นเมนูส่วนที่ใช้เขียนลำดับการทำงานของหนังสือเล่มนั้นๆ ที่กำลังใช้อยู่ ตั้งแต่เริ่มเปิดหนังสืออ่านจนกว่าจะคืนหนังสือ

12. Set Up Bookshelf Environment

เป็นเมนูส่วนที่ใช้ในการสร้างชั้นหนังสือ

13. Exit

เป็นเมนูส่วนที่ใช้ในปิดโปรแกรม HURESS

2.6.2 การทำงานในส่วนของเมนูย่อย [Bookshelf]

1. Create and Borrow a New Book

เป็นเมนูส่วนที่ใช้ในการสร้างหนังสือขึ้นใหม่ และถือว่าการยืมหนังสือเล่มนั้น
ด้วย

2. Borrow a Book

เป็นเมนูส่วนที่ใช้ในการยืมหนังสือ ในชั้นหนังสือนั้นๆ

3. Delete a Book

เป็นเมนูส่วนที่ใช้ในการลบหนังสือเล่มนั้นๆ ที่เราต้องการจะลบ

4. Move a Book

เป็นเมนูในส่วนที่ต้องการจะย้ายหนังสือ ไปไว้ยังชั้นหนังสืออื่นที่เราต้องการจะนำ
หนังสือเล่มนั้นไปเก็บไว้

5. Merge a Book

เป็นเมนูที่ใช้ในการรวมหนังสือ 2 เล่มเข้าด้วยกัน เพื่อสร้างเป็นหนังสือเล่มใหม่ขึ้นมา

6. Copy a Book

เป็นส่วนของเมนูที่ใช้ในการ Copy หนังสือที่เราต้องการ

7. Peruse Bookshelf

เป็นเมนูแสดงจำนวน Bookshelf ว่ามีกี่เล่ม

2.6.3 การทำงานในส่วนของเมนูย่อย [Book]

1. Return a book

เป็นเมนูที่ใช้ในการคืนหนังสือ

2. Create a Page

เป็นเมนูที่ใช้ในการสร้างหน้าที่เราต้องการ

3. Front Cover, Table of Contents, Next Page, Previous Page, Index Page, Real Cover

เป็นเมนูแสดงสถานะของหน้าหนังสือที่กำลังอ่านอยู่ ถ้าหากต้องการจะไปหน้าใด ก็คลิกเมาส์มายังเมนูนี้เลือกหน้าที่ต้องการได้เลย

4. Book Information

เป็นเมนูส่วนที่ใช้เช็คแก้ไขข้อมูลเกี่ยวกับหนังสือ เช่น ขนาดหนังสือ เป็นต้น

2.6.4 การทำงานในส่วนของเมนูย่อย [Page]

1. Delete Page

เป็นเมนูส่วนที่ใช้ในการลบหน้าที่ต้องการจะลบ ในหนังสือเล่มนั้นๆ ที่กำลังเปิดอ่านอยู่

2. Set up Table of Contents

เป็นเมนูส่วนที่ใช้ในการเช็คหน้าสารบัญ (Contents) ของหนังสือเล่มนั้นๆ

3. Set up Page Placement

เป็นส่วนของเมนูที่ใช้ในการเช็ควางตำแหน่งของหน้ากระดาษ

4. Set up Page Size

เป็นส่วนที่ใช้ในการเช็คขนาดของหน้ากระดาษ

5. Create Media

เป็นส่วนในการสร้าง Media ที่ต้องการจะสร้างในโปรแกรม

6. Create Procedure

เป็นส่วนที่ใช้ในการสร้างลำดับคำสั่งในการทำงานของ Media ที่เราต้องการ

7. Create Object

เป็นส่วนที่แสดงการเพิ่ม Dialog Box ของ Object ในการสร้าง Objects

8. Link Object

เป็นส่วนที่ใช้ในการเชื่อมโยงส่วนของ Dialog Box เข้าด้วยกัน

9. Emphasis and Object

เป็นส่วนที่ใช้ในการสร้างวงกลมในหน้านั้นๆ

10. Hard Copy to File

เป็นส่วนในการใช้ในการ Copy รูปภาพ ให้เป็นไฟล์

11. Hard Copy to Clipboard

เป็นส่วนที่ใช้ Copy รูปภาพลง Clipboard

12. Hard Copy to Print

เป็นส่วนที่ใช้ในการนำหน้าปัจจุบันพิมพ์ออกทางเครื่องพิมพ์

13. Page Information

เป็นส่วนที่ใช้เช็คเกี่ยวกับข้อมูลต่างๆ ของหน้ากระดาษ

2.6.5 การทำงานในส่วนของเมนูย่อย [Edit]**1. Undo**

เป็นเมนูที่ยกเลิกการกระทำครั้งสุดท้าย

2. Redo

เป็นส่วนที่เรียกใช้ข้อมูลที่ทำการ Undo ไป ให้ปรากฏขึ้นมาใหม่

3. Cut

เป็นส่วนที่ลบ Media ที่ต้องการ

4. Copy

เป็นส่วนที่ใช้ในการคัดลอก Media ที่เลือก

5. Paste

เป็นส่วนที่ใช้ในการวาง Media ที่คัดลอกมาในครั้งล่าสุด

6. Paste (Object Link Specified)

เป็นส่วนที่ใช้ในการวาง Object ที่คัดลอกมาในครั้งล่าสุด

7. Paste (Object Form Specified)

เป็นส่วนที่ใช้ในการวาง Form ที่คัดลอกมาในครั้งล่าสุด

8. Delete

เป็นส่วนที่ใช้ในการลบ Media และอื่นๆ ที่ต้องการ

9. Media - Next Media

เป็นการเลือก Media ถัดไป

10. Media - Previous Media

เป็นการเลือกใช้ Media ก่อน Media ปัจจุบัน

11. Media - Set Priority High

เป็นการเพิ่มลำดับความสำคัญของ Media ขึ้น 1 ลำดับ

12. Media - Set Priority Low

เป็นการลดลำดับความสำคัญของ Media ลง 1 ลำดับ

13. Media - Information

เป็นการเช็กรายละเอียดเกี่ยวกับ Media

14. Media - Procedure Test Procedure

เป็นการทดสอบการทำงานของโปรแกรมย่อย

15. Media - Procedure Edit

เป็นการแก้ไข โปรแกรมย่อย

2.6.6 การทำงานในส่วนของ Dialog Box ต่างๆ ในโปรแกรม HURESS**1. HURESS-Book User Validation Dialog Box**

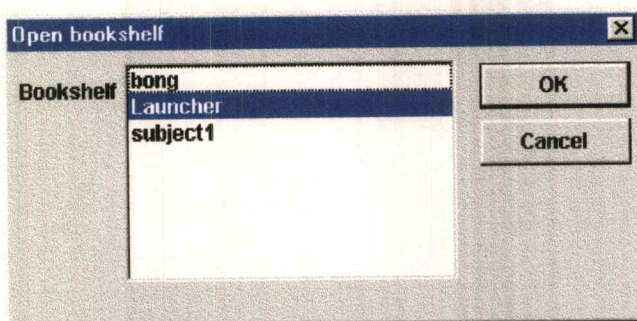
Dialog Box นี้ใช้เพื่อป้อน User name และ Password เพื่อที่จะเข้าสู่โปรแกรม HURESS ดังรูปที่ 2.16



รูปที่ 2.16 การเข้าโปรแกรม HURESS

2. Dialog Box to open a Bookshelf

Dialog Box จะใช้เพื่อเลือกชั้นวางหนังสือ เมื่อเราต้องการเลือก กด OK เพื่อใช้ชั้นหนังสือ นั้น ดังรูปที่ 2.17



รูปที่ 2.17 การเปิดหนังสือ

3. Dialog Box to Close a Bookshelf

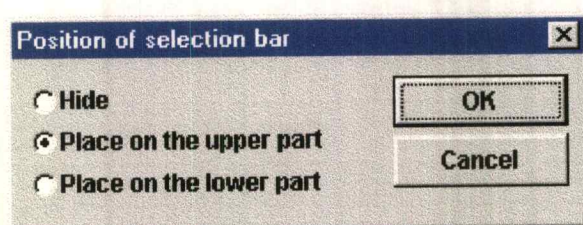
Dialog Box ใช้เพื่อปิดชั้นหนังสือ กด OK เพื่อทำการปิดหนังสือต่างๆ ดังรูปที่ 2.18



รูปที่ 2.18 การขีหนังสือ

4. Dialog Box to Set up Position of Selection Bar

Dialog Box นี้ ใช้ในการกำหนดตำแหน่งของ Selection Bar แสดงดังรูปที่ 2.19

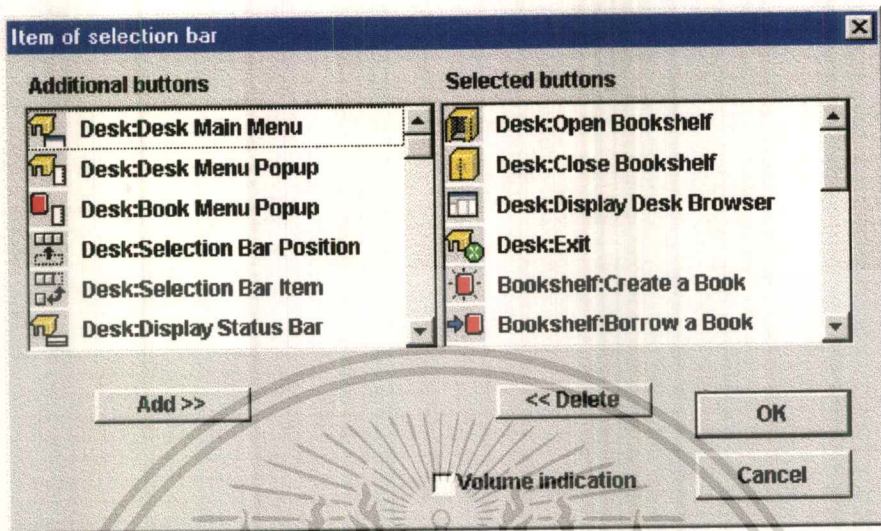


รูปที่ 2.19 การกำหนดตำแหน่ง Selection Bar

5. Dialog Box to Set up Item of Selection Bar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

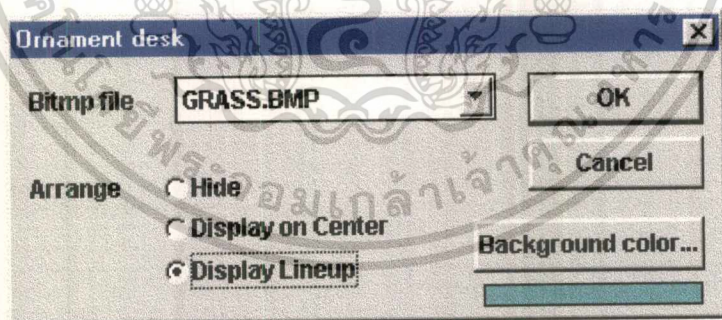
Dialog Box นี้ใช้เมื่อต้องการ Set การแสดงของ Item ใน Selection Bar ดังรูปที่ 2.20



รูปที่ 2.20 การ Set การแสดงของ Item ใน Selection Bar

6. Dialog Box for Ornament Desk

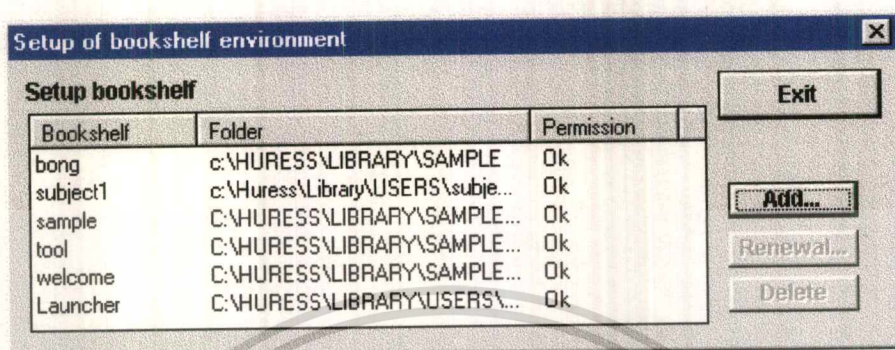
Dialog Box นี้ใช้เพื่อกำหนดการแสดงผลของ Background ในรูปแบบต่างๆ ดังรูปที่ 2.21



รูปที่ 2.21 การกำหนด Background

7. Dialog Box to Set up Bookshelf Environment

Dialog Box นี้ใช้เพื่อต้องการเพิ่มชั้นวางหนังสือ ดังรูปที่ 2.22



รูปที่ 2.22 การเพิ่มชั้นวางหนังสือ

8. Additional Dialog box for Bookshelf

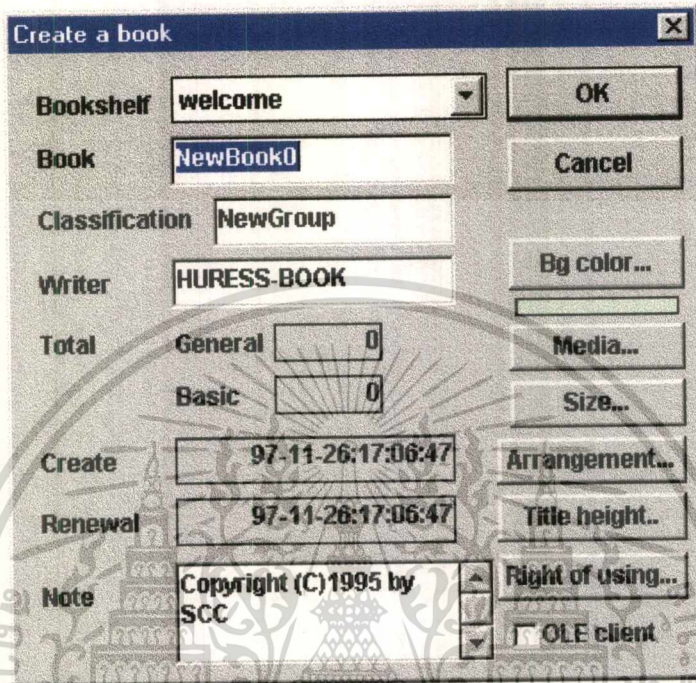
Dialog Box นี้ใช้เมื่อต้องการจะเพิ่มชั้นวางหนังสือ ดังรูปที่ 2.23



รูปที่ 2.23 การเพิ่มชั้นวางหนังสือ

9. Dialog Box for Creating a Book

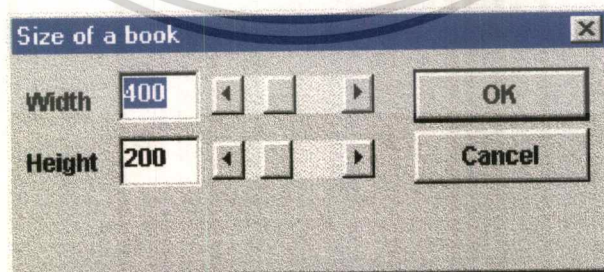
Dialog box นี้ใช้เพื่อสร้างหนังสือเล่มใหม่ ดังรูปที่ 2.24



รูปที่ 2.24 การสร้างหนังสือเล่มใหม่

10. Dialog Box to Set up the Size of a Book

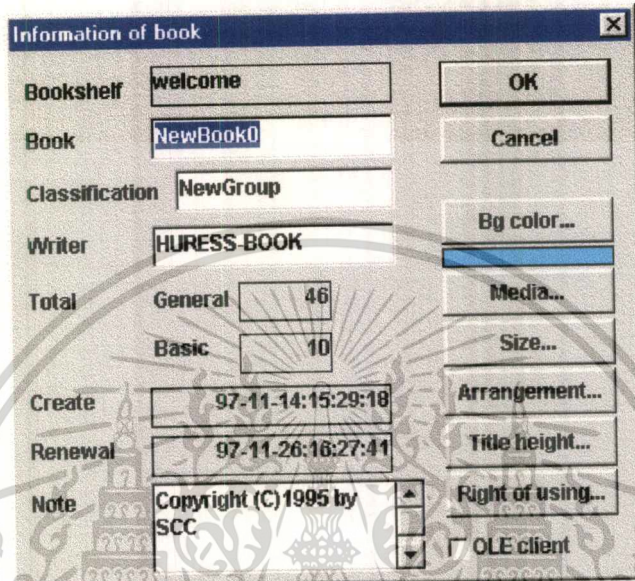
Dialog Box นี้ ใช้เพื่อกำหนดขนาดของหนังสือ ดังรูปที่ 2.25



รูปที่ 2.25 การกำหนดขนาดของหนังสือ

11. Information Dialog Box for Books

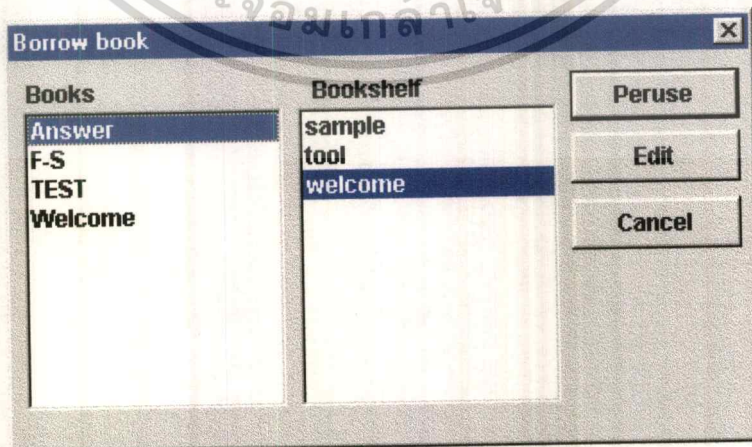
Dialog Box นี้ใช้เพื่อกำหนดรายละเอียดในส่วนต่างๆ ของหนังสือที่สร้างขึ้นใหม่ ดังรูปที่ 2.26



รูปที่ 2.26 การกำหนดรายละเอียดในส่วนต่างๆ ของหนังสือ

12. Dialog Box for Book Borrowing

Dialog Box นี้ ใช้ในการยืมหนังสือ ดังรูปที่ 2.27

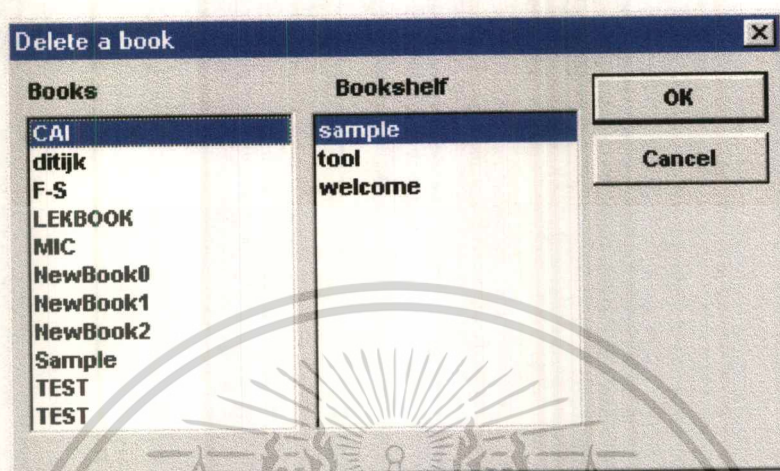


รูปที่ 2.27 การยืมหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. Dialog Box to Delete a Book

Dialog Box นี้ ใช้เพื่อต้องการจะลบหนังสือ ดังรูปที่ 2.28



รูปที่ 2.28 การลบหนังสือ

14. Dialog Box to move a Book

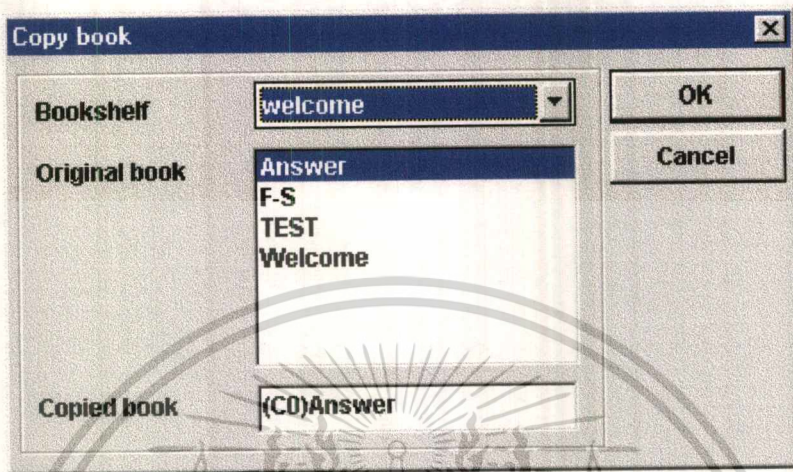
Dialog Box นี้ ใช้เพื่อทำการย้ายหนังสือ ดังรูปที่ 2.29



รูปที่ 2.29 การย้ายหนังสือ

15. Dialog Box to Copy a Book

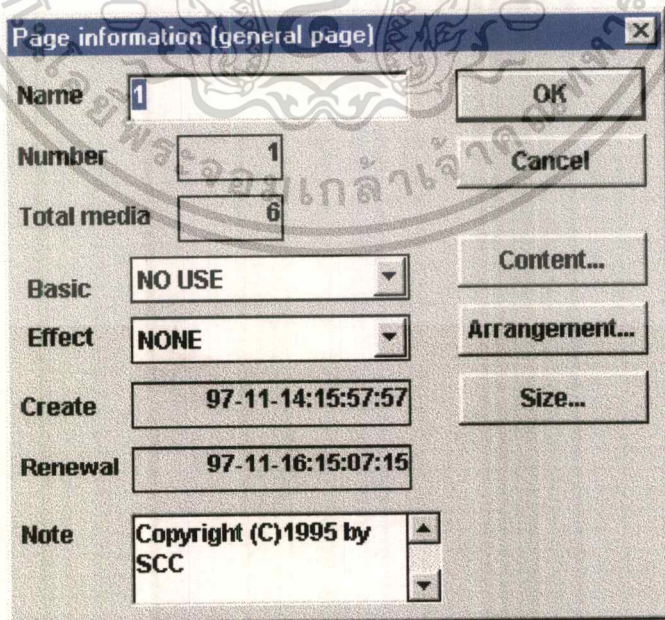
Dialog Box นี้ ใช้เพื่อทำการ Copy หนังสือ ดังรูปที่ 2.30



รูปที่ 2.30 การคัดลอกหนังสือ

16. Page Information (General Page) Dialog Box

Dialog Box นี้ ใช้เพื่อกำหนดรายละเอียดหน้าของเนื้อหา ดังรูปที่ 2.31

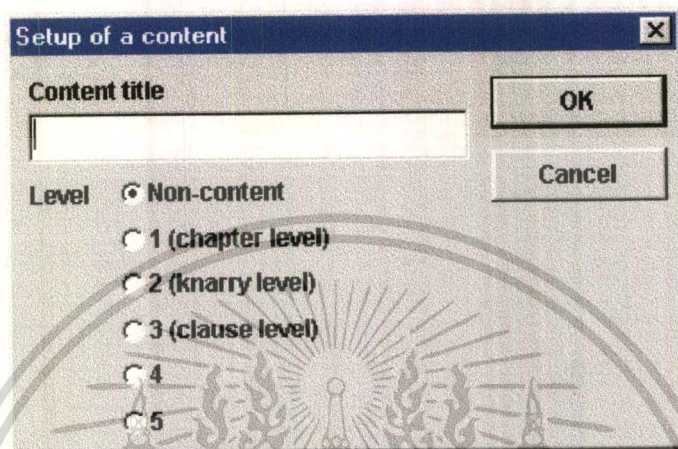


รูปที่ 2.31 การกำหนดรายละเอียดหน้าของเนื้อหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. Dialog Box to Set up a Table of Contents

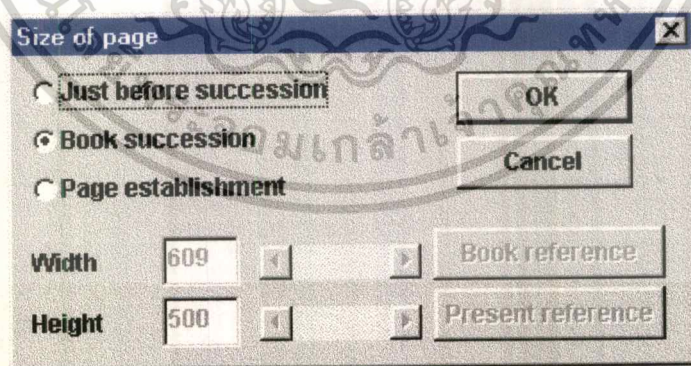
Dialog Box นี้ ใช้เพื่อต้องการ Set หน้าต่างๆ ให้อยู่ในรายการของหน้าสารบัญ ดังรูปที่ 2.32



รูปที่ 2.32 การกำหนดหน้าสารบัญ

18. Dialog Box to Set up Page Size

Dialog Box นี้ ใช้เพื่อต้องการกำหนดขนาดของหน้าต่างๆ ดังรูปที่ 2.33



รูปที่ 2.33 การกำหนดขนาดของหน้า

19. Page Information (Basic Page) Dialog Box

Dialog Box นี้ ใช้เพื่อกำหนดรายละเอียดของหน้า Background ดังแสดงในรูปที่ 2.34

The dialog box titled "Page information (basic page)" has the following fields and values:

- Name: [.....]
- Number: 3
- Total media: 5
- Create: 97-11-15:18:30:39
- Renewal: 97-11-22:21:01:28
- Note: Copyright (C)1995 by SCC

Buttons: OK, Cancel

รูปที่ 2.34 การกำหนดรายละเอียดของหน้า

2.6.7 Media Information Dialog Box

1. Index Registration Dialog Box

Dialog Box นี้ ใช้เพื่อการทำดัชนีของหนังสือ ดังรูปที่ 2.35

The dialog box titled "Registration of index" has the following fields and buttons:

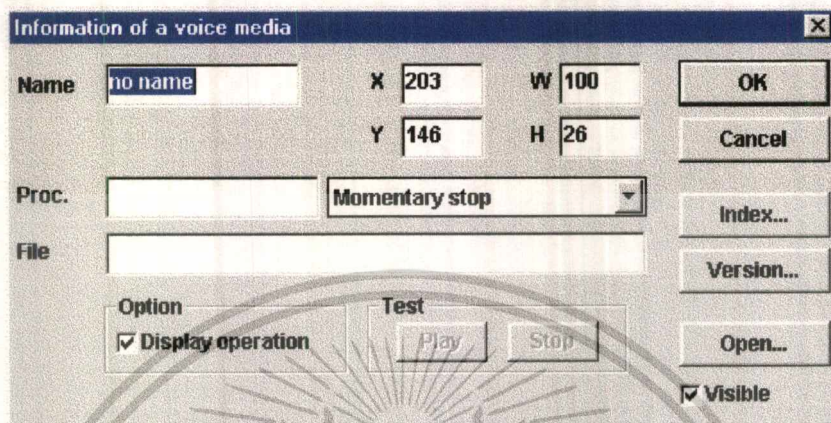
- Reading: []
- Index: []

Buttons: OK, Cancel

รูปที่ 2.35 การกำหนดดัชนี

2. Information Dialog Box for Voice media

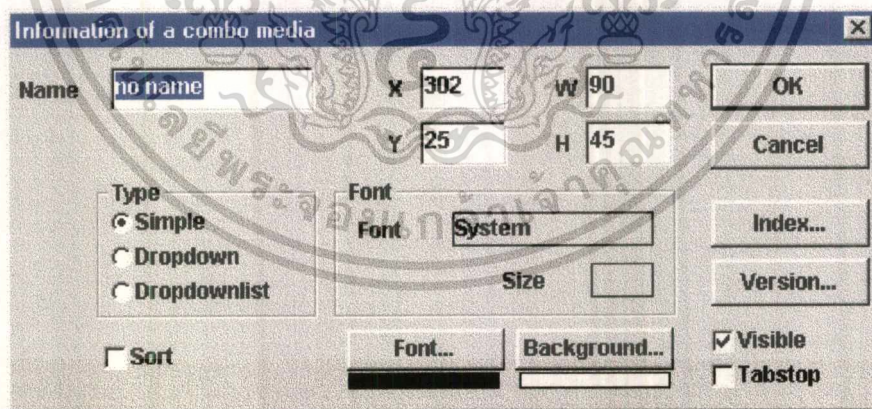
Dialog Box นี้ ใช้เพื่อทำรูปภาพเคลื่อนไหว ดังรูปที่ 2.36



รูปที่ 2.36 การทำรูปเคลื่อนไหว

3. Information Dialog Box for Combo Media

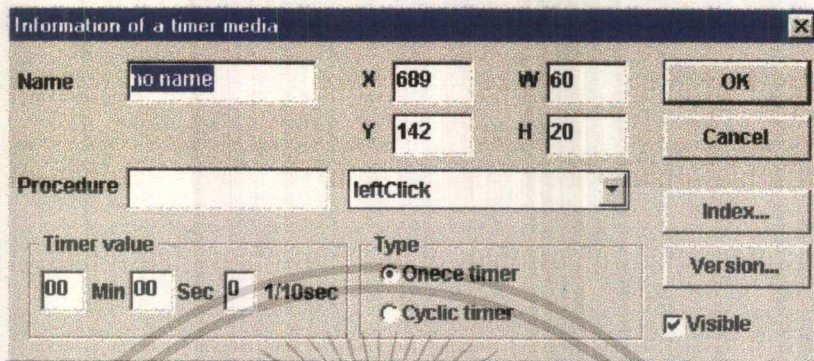
Dialog Box นี้ใช้เพื่อใช้ในการเลือกคำตอบของแบบทดสอบ ดังแสดงในรูปที่ 2.37



รูปที่ 2.37 การเลือกคำตอบในแบบทดสอบ

6. Information Dialog Box for timer media

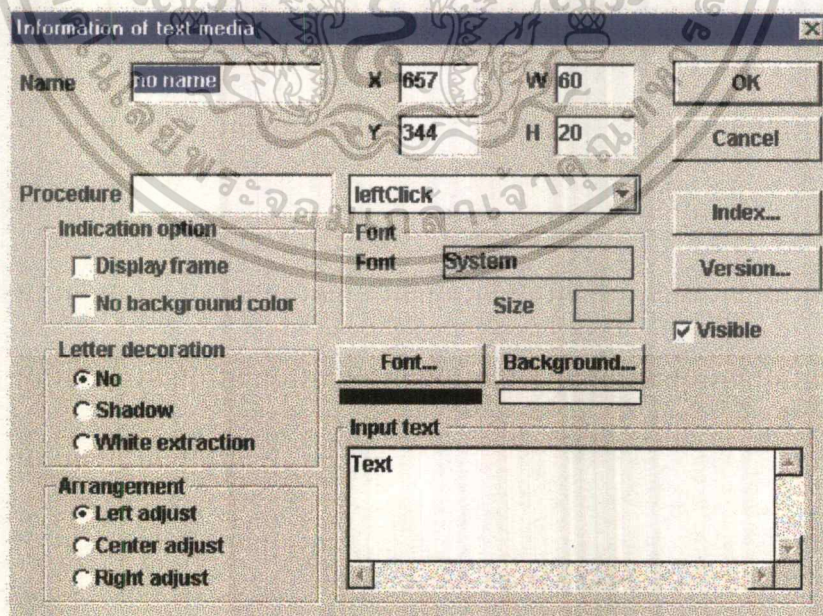
Dialog Box นี้ใช้เพื่อการตั้งเวลา จะปรากฏ Dialog Box ดังรูปที่ 2.40



รูปที่ 2.40 การตั้งเวลา

7. Information Dialog Box for Text Media

Dialog Box นี้ใช้เพื่อการสร้างข้อความที่ต้องการ และสามารถกำหนดรูปแบบของตัวอักษร สี ขนาด ดังรูปที่ 2.41

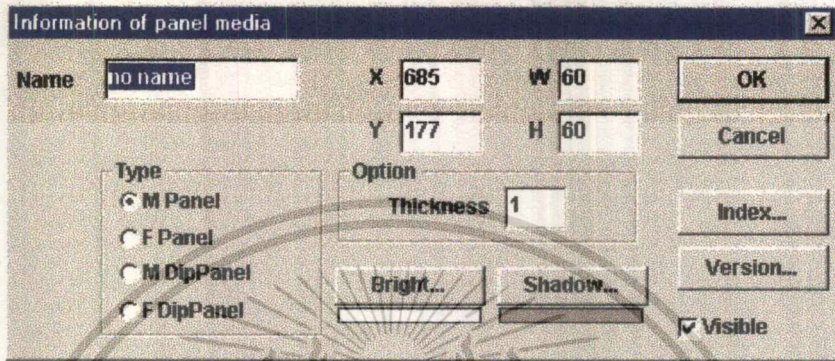


รูปที่ 2.41 การสร้างข้อความ

8. Information Dialog Box for Panel Media

Dialog Box นี้ ใช้เพื่อการสร้างปุ่ม ที่สามารถกำหนดรูปแบบของปุ่มนั้นๆ ดังรูปที่

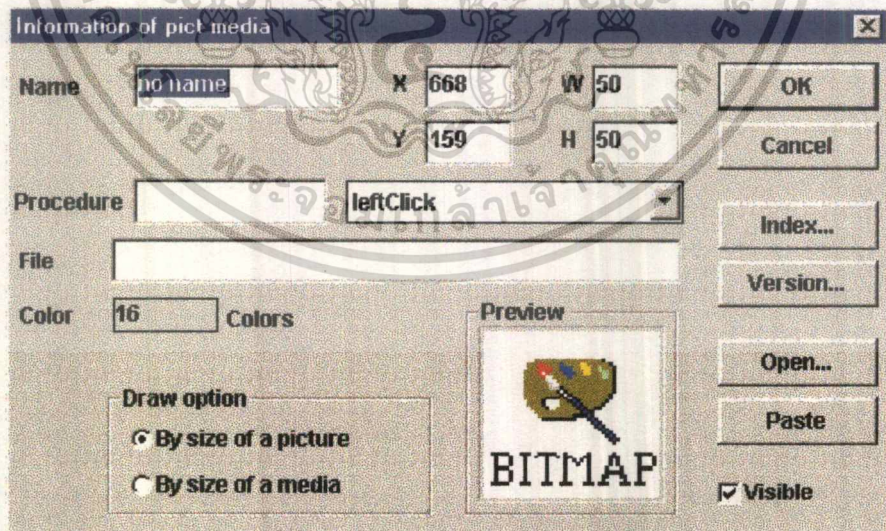
2.42



รูปที่ 2.42 การสร้างปุ่ม

9. Information Dialog Box for Pict Media

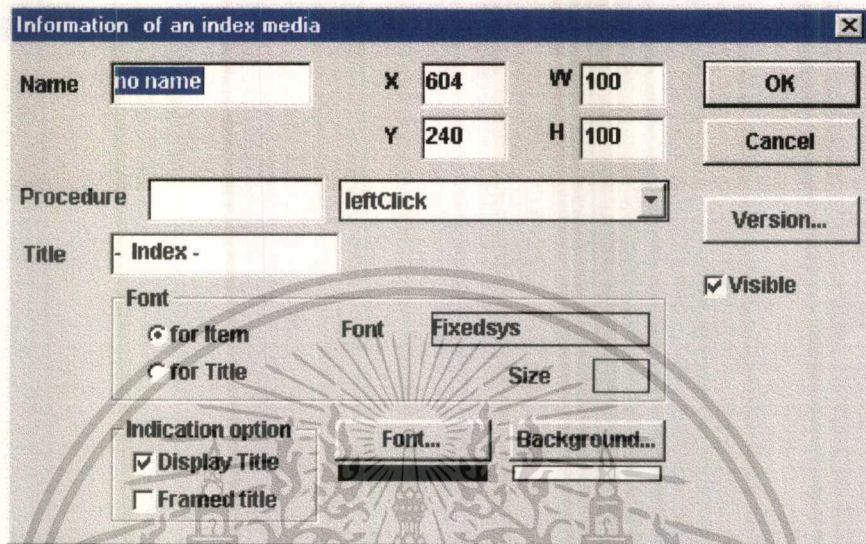
Dialog Box นี้ ใช้เพื่อดีงรูปภาพ ที่ต้องการมาประกอบหนังสือ ดังรูปที่ 2.43



รูปที่ 2.43 การดีงรูปมาประกอบโปรแกรม

4. Information Dialog Box for Index Media

Dialog Box นี้ใช้เพื่อกำหนดรายละเอียดของ Index ดังรูปที่ 2.38

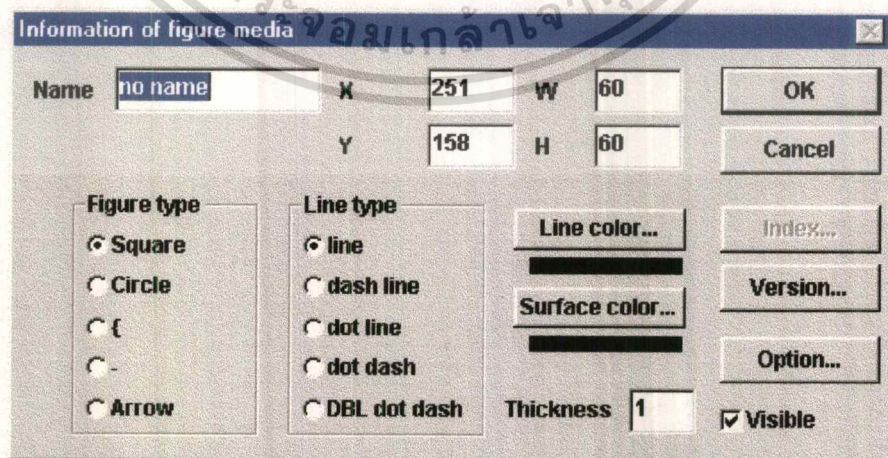


รูปที่ 2.38 การกำหนดรายละเอียดของหน้าดัชนี

5. Information Dialog Box for Figure media

Dialog Box นี้ใช้เพื่อการวาดเกี่ยวกับเส้นต่างๆ รูปร่างกลม วงรี ลูกศร แสดงในรูปที่

2.39

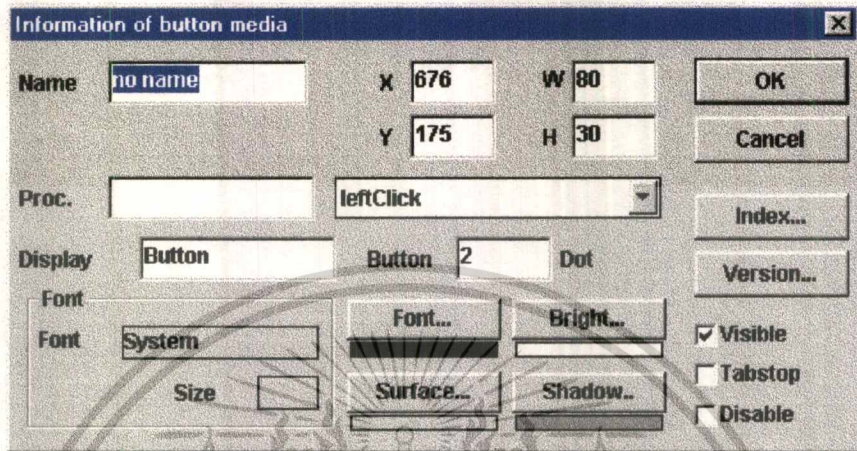


รูปที่ 2.39 การเรียกภาพมาประกอบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Information Dialog Box for Button Media

Dialog Box นี้ ใช้เพื่อสร้างปุ่มที่สามารถกำหนดรูปแบบได้ ดังรูปที่ 2.44



รูปที่ 2.44 การสร้างปุ่มทำงานต่างๆ

บทที่ 3

การออกแบบ การสร้าง และการทำงาน

การออกแบบและการสร้างโปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80 ซึ่งเป็นสื่อประกอบการเรียนการสอนในโครงการสอนระบบทางไกลแบบ 2 ทาง ซึ่งเป็นบทเรียนช่วยสอนที่สร้างขึ้นโดยโปรแกรม HURESS เน้นสื่อความหมายรูปแบบที่อ่านง่าย และในทุกๆบทจะประกอบด้วยคำอธิบายที่สำคัญหลักๆของแต่ละหัวข้อและรูปภาพที่เกี่ยวข้องในการอธิบาย ส่วนแบบฝึกหัดก็จะมีคำถามตอบและมีการเช็คคำตอบระหว่างผู้สอนและผู้เรียนได้ โดยผลการทำแบบฝึกหัดจะตรวจได้ด้วยอาจารย์ผู้สอน เทคนิคและวิธีการที่ใช้สร้างบทเรียนและแบบฝึกหัด โดยโปรแกรม HURESS ได้นำเสนอรวมในส่วนของการสร้างบทเรียนและแบบฝึกหัด

3.1 เครื่องมือที่เกี่ยวข้องในการออกแบบและการสร้าง

3.1.1 ทางด้านฮาร์ดแวร์ (Hardware)

1. เครื่องคอมพิวเตอร์ Pentium-100 NEC PowerMate V
2. หน่วยความจำหลัก 8 MB
3. ฮาร์ดดิสก์ 1.2 GB
4. Mouse
5. Sound Card

3.1.2 ทางด้านซอฟต์แวร์ (Software)

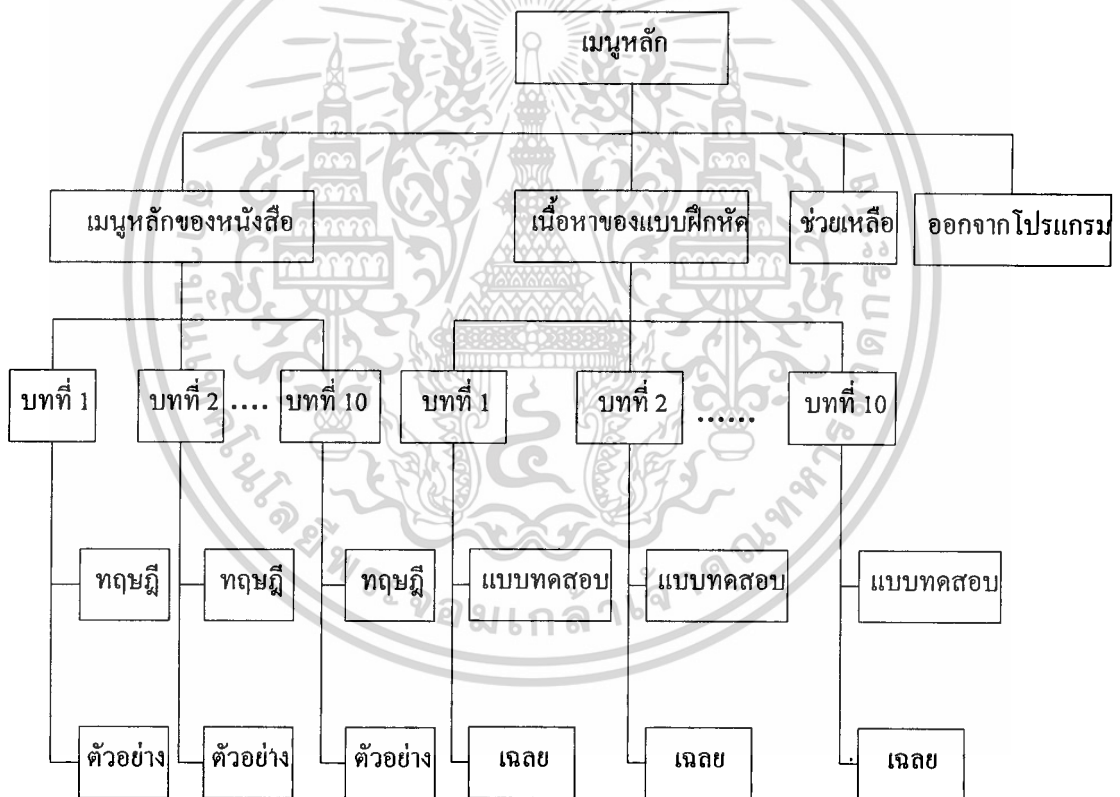
1. โปรแกรม MS-WINDOWS 95
2. โปรแกรม HURESS
3. โปรแกรม GIFCON
4. โปรแกรม ACDSsee 95

3.2 ขั้นตอนในการออกแบบโปรแกรม

โปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80 ประกอบด้วยบทเรียนทั้งหมด 10 บทเรียน ซึ่งในแต่ละบทเรียนจะประกอบไปด้วย 2 ส่วนหลักๆ คือ

ส่วนที่หนึ่ง เป็นทฤษฎีของบทเรียนในแต่ละบท ซึ่งจะเป็เนื้อหาหลักๆ ของแต่ละหัวข้อและมีรูปภาพประกอบคำอธิบาย

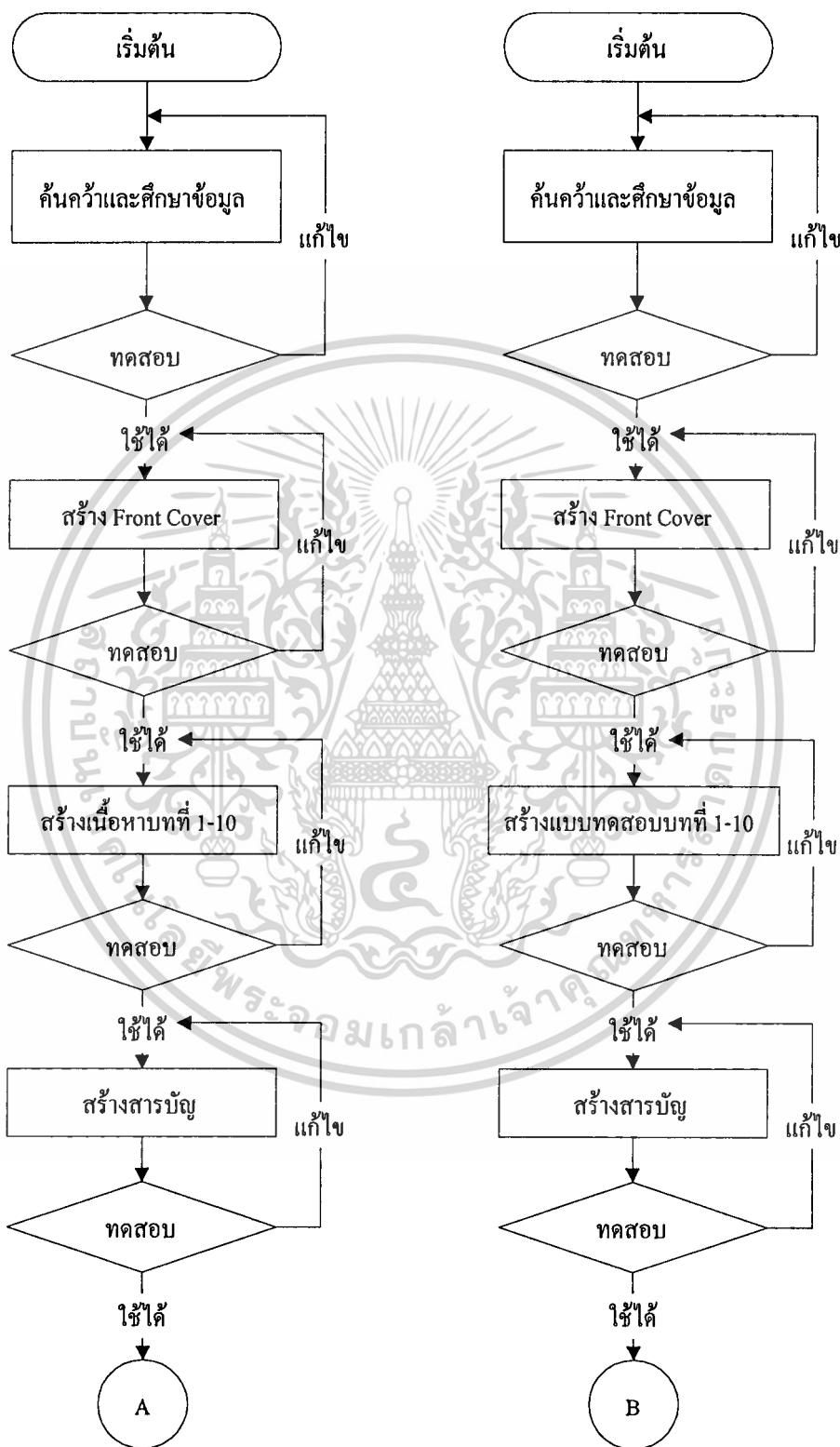
ส่วนที่สอง เป็นส่วนของแบบฝึกหัด ซึ่งผู้เรียนในแต่ละประเทศและอาจารย์ผู้สอนสามารถทำแบบฝึกหัดและตรวจคำตอบร่วมกันได้



รูปที่ 3.1 ผังการทำงานของโปรแกรมช่วยสอนวิชาไมโครโปรเซสเซอร์ Z80

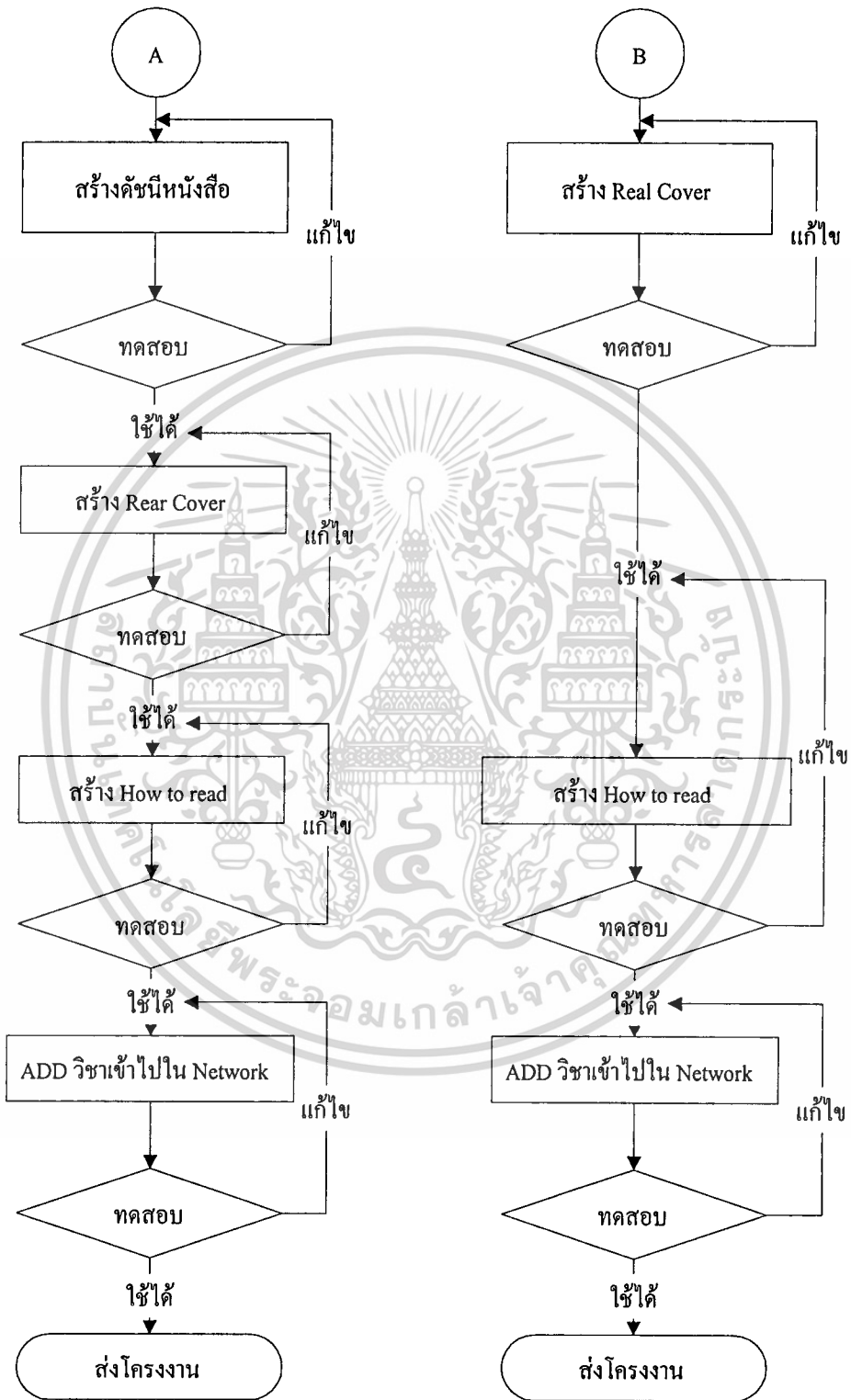
จากผังการทำงานของโปรแกรมที่กำหนดขึ้นในรูปที่ 3.1 ได้ออกแบบและสร้างส่วนต่างๆของผังการทำงาน โดยมีลำดับขั้นตอนการออกแบบและสร้างโปรแกรม ดังผังการทำงานในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ผังขั้นตอนการออกแบบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

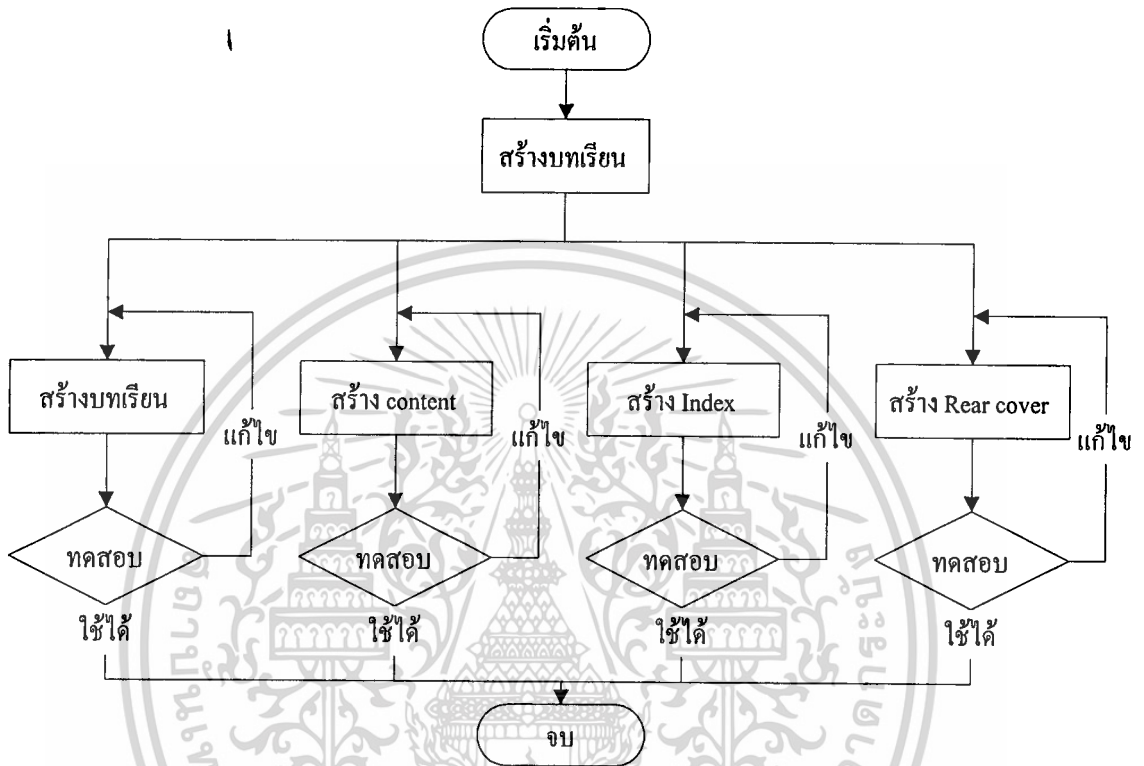


รูปที่ 3.2 (ต่อ) ผังขั้นตอนการออกแบบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการสร้างบทเรียน

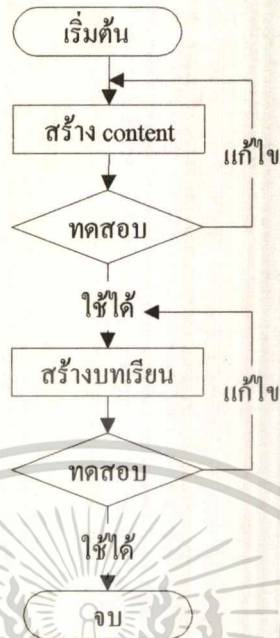
การสร้างบทเรียนด้วยโปรแกรม HURESS มีขั้นตอนการสร้างดังผังรูปที่ 3.3



รูปที่ 3.3 ผังการสร้างบทเรียน

3.4 ขั้นตอนการสร้างแบบฝึกหัด

การสร้างแบบฝึกหัดด้วยโปรแกรม HURESS มีขั้นตอนการสร้างดังผังรูปที่ 3.4

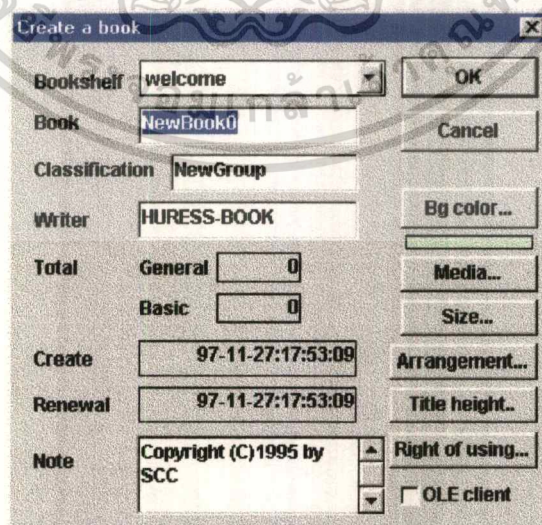


รูปที่ 3.4 ผังการสร้างแบบฝึกหัด

3.5 วิธีการสร้างส่วนของบทเรียนและแบบฝึกหัดด้วยโปรแกรม HURESS

3.5.1 การสร้างหนังสือ

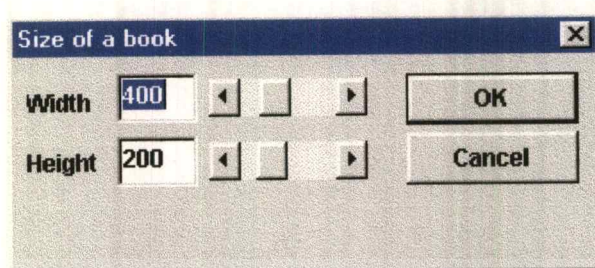
นำมาส์ไปคลิกที่ Create and Borrow a New Book จะปรากฏ Dialog Box ดังรูปที่ 3.5



รูปที่ 3.5 เมนู Desk ที่ Create and Borrow a New book

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการกำหนดรายละเอียดส่วนต่างๆ ของหนังสือ เช่น ขนาดของหนังสือ ดังรูปที่ 3.6



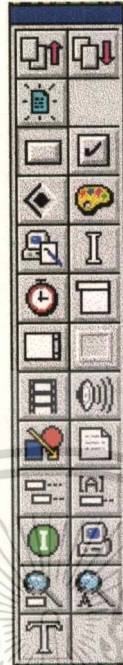
รูปที่ 3.6 Dialog Box ในการกำหนดขนาดของหนังสือ

เมื่อทำการกำหนดรายละเอียดต่างๆ เสร็จแล้ว หนังสือที่ถูกสร้างจะปรากฏที่มุมล่างด้านซ้ายของจอภาพ ดังรูปที่ 3.7



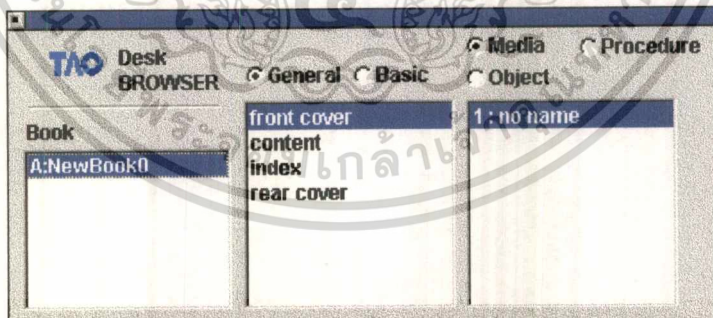
รูปที่ 3.7 วินโดว์ของหนังสือเมื่อสร้างเสร็จแล้ว

ทำการเปิดหนังสือโดยดับเบิลคลิกที่วินโดว์ของหนังสือ จะปรากฏหนังสือขึ้นมาพร้อมกับ Operation of Media Selection Bar ซึ่งเป็น Icon เครื่องมือที่ใช้ในการสร้างหนังสือ ดังรูปที่ 3.8



รูปที่ 3.8 Operation of Media Selection Bar

หนังสือที่ถูกสร้างขึ้นมา จะมี Desk Browser ขึ้นมาด้วย ซึ่ง Desk Browser จะแสดงถึงส่วนประกอบต่างๆ ของหนังสือ ดังรูป 3.9



รูปที่ 3.9 Desk Browser

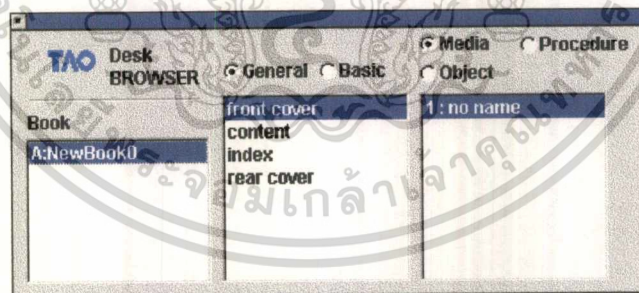
3.5.2 การสร้างข้อความของหนังสือ

การสร้างข้อความของหนังสือ จะทำได้โดยใช้ Text Media ในการสร้าง ดังรูปที่ 3.10



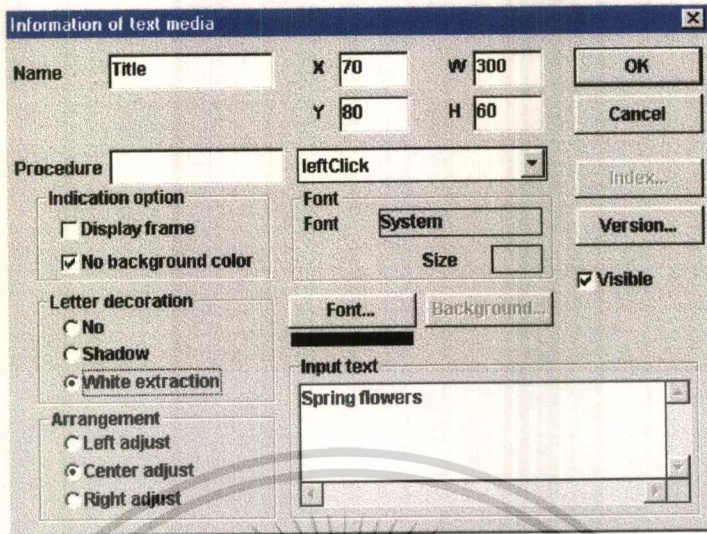
รูปที่ 3.10 การสร้างข้อความของหนังสือ

เมื่อทำการสร้างข้อความแล้ว ใน Desk Browser ก็จะมี Media เพิ่มขึ้น ดังรูปที่ 3.11



รูปที่ 3.11 การเพิ่มของ Media ใน Desk Browser

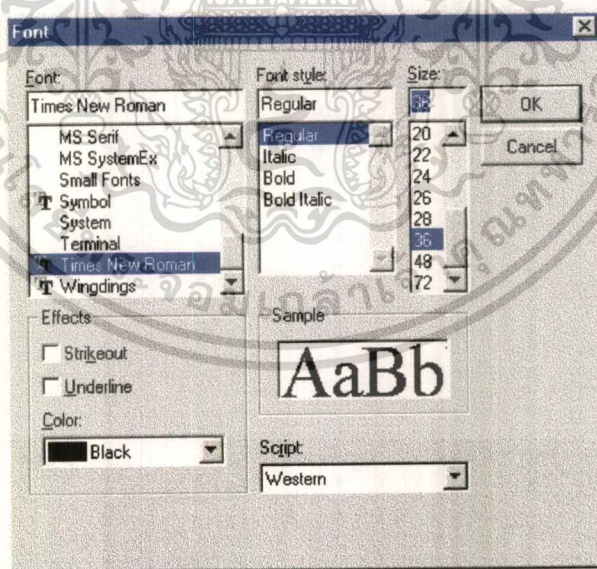
เมื่อมีมีเดียเพิ่ม ก็จะต้องมีการกำหนดรายละเอียดต่าง ๆ ของมีเดียที่เราสร้างขึ้น ดังรูปที่ 3.12



รูปที่ 3.12 การกำหนดรายละเอียดของมีเดียที่สร้างขึ้น

มีเดียที่เป็นข้อความสามารถกำหนดรูปแบบของตัวหนังสือได้ในรูปแบบที่ต้องการ ดัง

รูปที่ 3.13

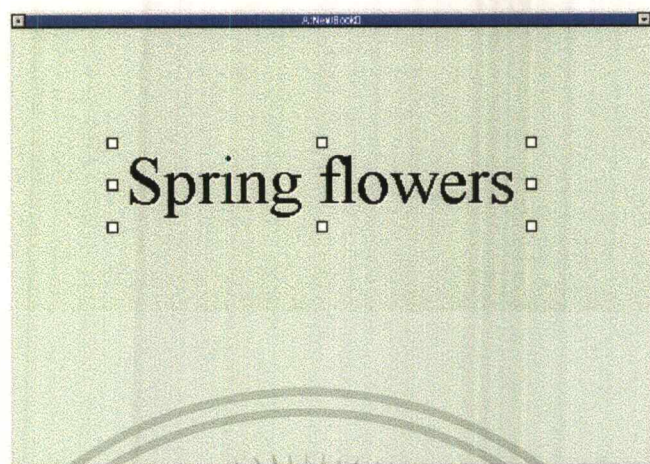


รูปที่ 3.13 การกำหนดรายละเอียดของข้อความ

เมื่อมีการกำหนดรายละเอียดของมีเดียเรียบร้อยแล้ว คลิก OK ข้อความที่กำหนดแล้ว

จะปรากฏดังรูปที่ 3.14

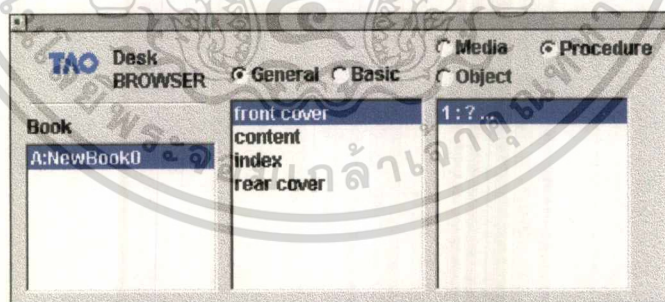
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ผลที่ได้จากการสร้างมีเดีย

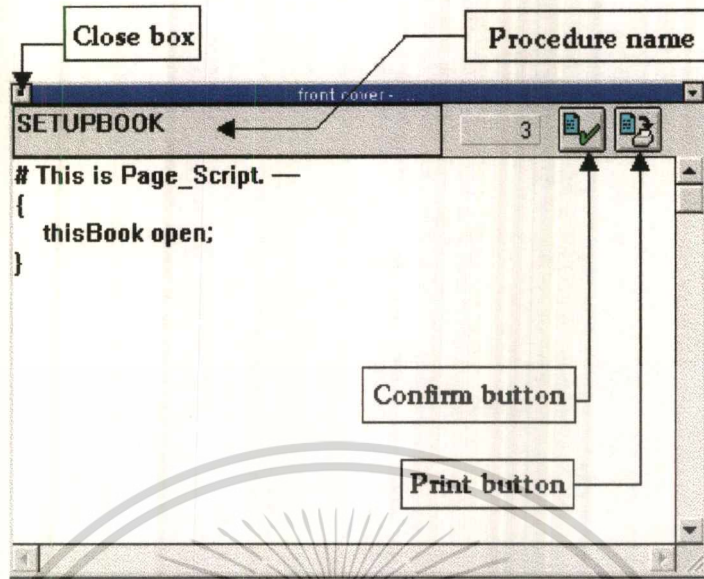
3.5.3 การสร้างลำดับการทำงานของหนังสือ

ในการสร้างลำดับการทำงานของหนังสือ จะเป็นในลักษณะการเขียนโปรแกรมย่อยให้กับหนังสือ ในการที่จะเขียนโปรแกรม ให้คลิกซ้ายที่ [Create Procedure] บน Media Selection Bar จากนั้น Desk Browser จะปรากฏดังรูปที่ 3.15



รูปที่ 3.15 การสร้างโปรแกรมย่อย

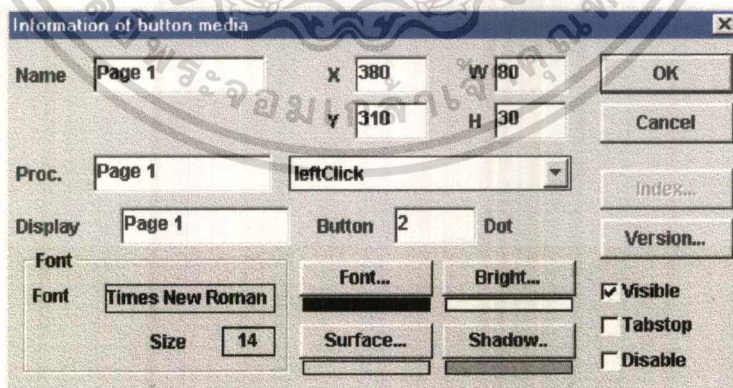
ในการสร้างโปรแกรมย่อย ให้ดับเบิลคลิกที่ "1:?..." ใน Desk Browser เพื่อทำการเขียนโปรแกรมย่อยให้กับหนังสือ ดังรูป 3.16



รูปที่ 3.16 การเขียน โปรแกรมย่อยของหนังสือ

3.5.4 การสร้างปุ่ม Next

ในการสร้างปุ่มทำงานในการเปิดหน้าหนังสือถัดไป สามารถทำได้โดยสร้างปุ่มขึ้นมา โดยคลิกที่ [Button Media] บน Media Selection Bar แล้วคลิกขวาเพื่อวางปุ่ม จากนั้นก็ดับเบิลคลิกเข้าไปที่ปุ่ม เพื่อทำการกำหนดรายละเอียดต่างๆ ของปุ่ม ดังรูปที่ 3.17

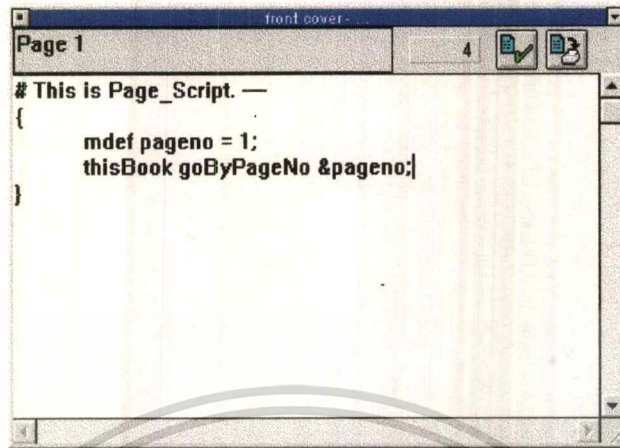


รูปที่ 3.17 การกำหนดรายละเอียดของปุ่ม

การที่จะให้ปุ่มทำงานได้ตามที่เราต้องการ ต้องเขียน โปรแกรมย่อยให้กับปุ่มด้วย ดัง

รูปที่ 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

Page 1
# This is Page_Script. —
{
  mdef pageno = 1;
  thisBook goByPageNo &pageno;
}

```

รูปที่ 3.18 การเขียน โปรแกรมย่อยกำหนดการทำงานของปุ่ม

หลังจากนำสิ่งที่สร้างตามหัวข้อที่ผ่านมารวมกัน จะได้เป็น Front Cover ดังรูปที่ 3.19

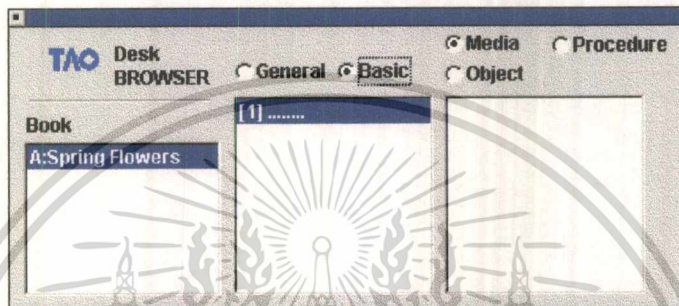


รูปที่ 3.19 ตัวอย่าง Front Cover ที่สร้าง

3.5.5 การสร้าง Basic Page ของหน้า Background

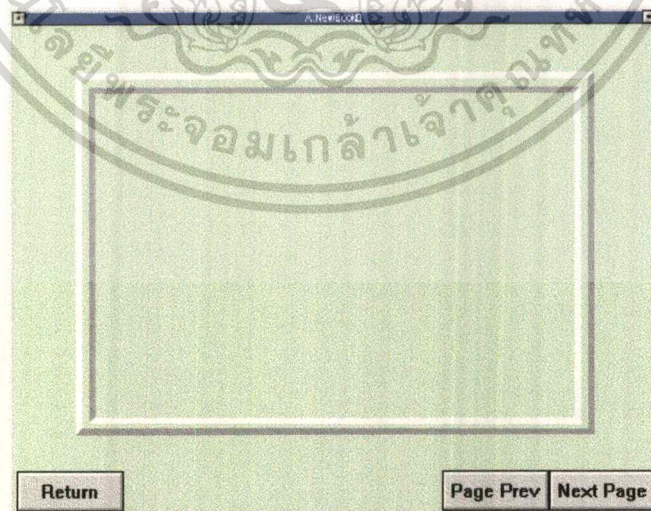
การสร้าง Background ทำได้ตามขั้นตอนต่อไปนี้

1. คลิกซ้ายที่ "Basic" บน Desk Browsers จะแสดงคอลัมน์ของ Basic Page
2. คลิกซ้ายที่ "Create a Page" บนเมนู Book ใน Desk Main Menu จากนั้น Desk Browser จะปรากฏหน้าของ Basic Page ที่เราสร้างขึ้นใหม่ ดังรูปที่ 3.20



รูปที่ 3.20 การเพิ่ม Basic Page ใน Desk Browsers

3. ดับเบิลคลิกซ้ายที่ "([1].....)" ของ Desk Browser จอภาพจะปรากฏ Dialog Box เพื่อทำการกำหนดชื่อของหน้า Basic Page ดังตัวอย่างในรูปที่ 3.21

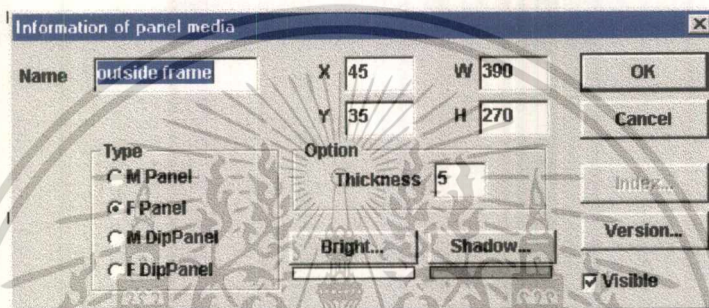


รูปที่ 3.21 การแสดงผลของหน้า Basic Page

3.5.6 การสร้างกรอบของรูปภาพ

1. การสร้างกรอบของรูปภาพทำได้โดยการคลิกซ้ายที่ Panel Media ใน Media Selection Bar หลังจากนั้นคลิกขวาบนพื้นที่ของหน้าที่เราต้องการสร้างกรอบให้กับรูปภาพมีเดียที่ใช้สร้างกรอบจะปรากฏขึ้นมา

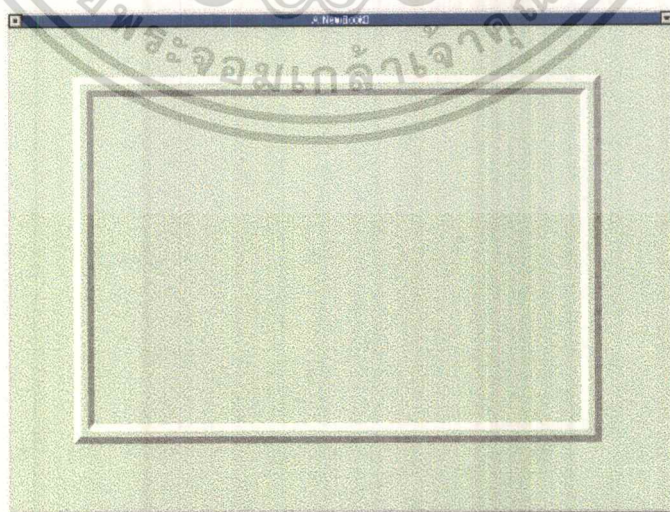
2. ดับเบิลคลิกซ้ายที่ Media จอภาพจะปรากฏ Dialog Box เพื่อทำการกำหนดรายละเอียดต่างๆของ Panel Media ทำการกำหนดรายละเอียดดังรูปที่ 3.22



รูปที่ 3.22 การกำหนดรายละเอียดของ Panel Media

เมื่อทำการกำหนดรายละเอียดของ Panel แล้ว จะปรากฏ Panel ขึ้นมาที่หน้าหนังสือ ดัง

รูปที่ 3.23

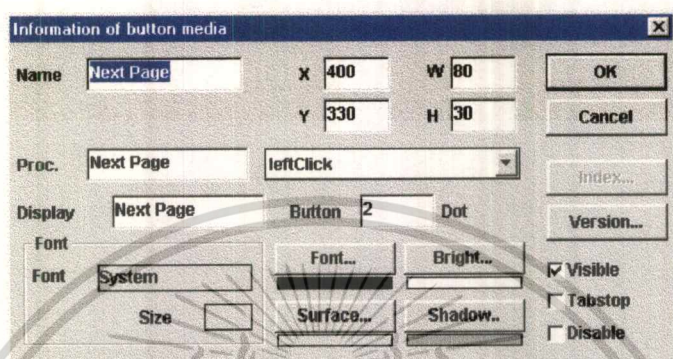


รูปที่ 3.23 Panel หลังจากกำหนดรายละเอียดแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

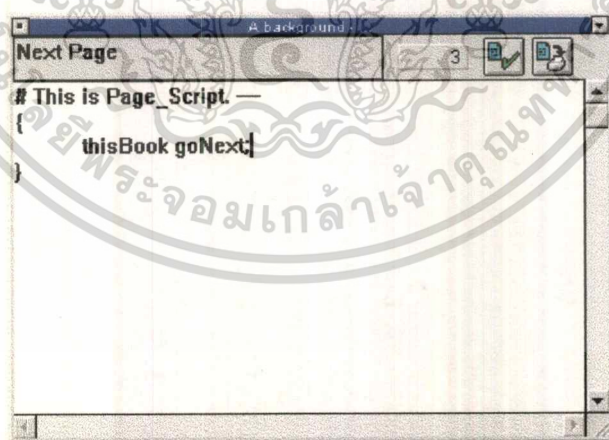
3.5.7 การสร้าง Button Next Page

ในการสร้าง Button Next Page ทำการเลือกมีเดียและกำหนดรายละเอียดของปุ่ม ดังรูปที่ 3.24



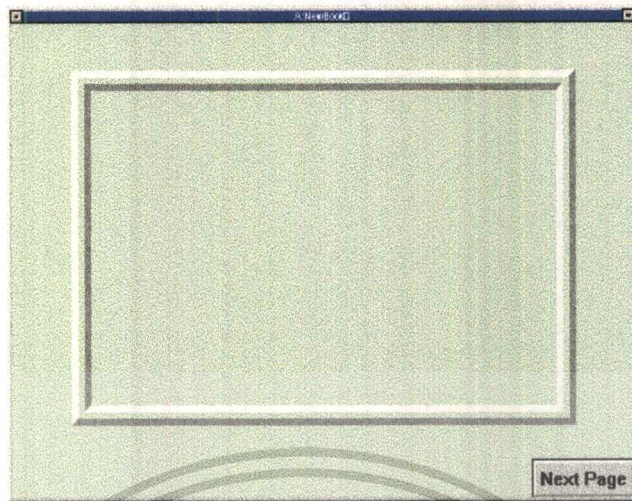
รูปที่ 3.24 การกำหนดรายละเอียดของ Button

จากนั้นทำการเขียนโปรแกรมย่อยให้กับ Button เพื่อกำหนดการทำงานของ Button ดังรูปที่ 3.25



รูปที่ 3.25 การเขียนโปรแกรมย่อยให้กับ Button

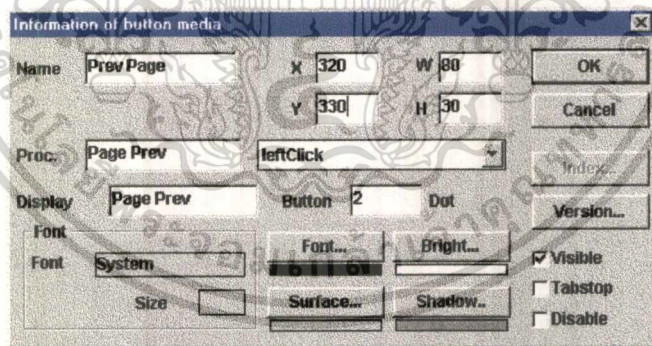
เมื่อมีการกำหนดการทำงานให้กับ Button แล้ว จะปรากฏหน้าจอดังรูปที่ 3.26



รูปที่ 3.26 การแสดงผลของ Button

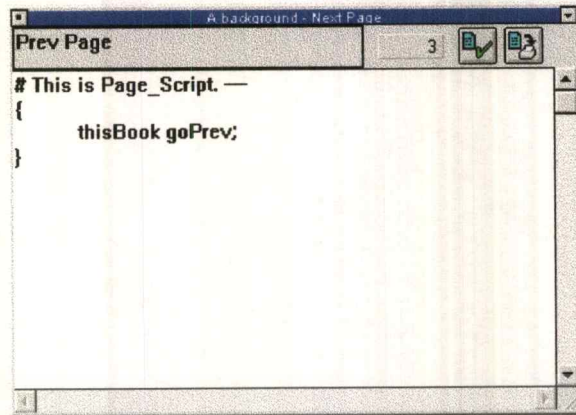
3.5.8 การสร้าง Button Previous Page

ในการสร้างปุ่ม Previous ต้องเลือกมีเดียในการสร้างปุ่ม และกำหนดรายละเอียดของ Button ดังรูปที่ 3.27



รูปที่ 3.27 การกำหนดรายละเอียดของ Previous Button

จากนั้นทำการเขียน โปรแกรมย่อย เพื่อให้ Button ทำงานตามที่ต้องการ ดังรูปที่ 3.28



รูปที่ 3.28 การเขียน โปรแกรมย่อยของ Previous Button

เมื่อทำการกำหนดการทำงานของ Button เรียบร้อยแล้ว จะปรากฏหน้าจอดังรูปที่ 3.29

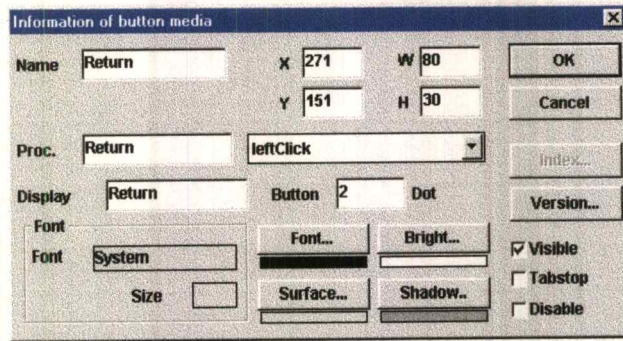


รูปที่ 3.29 การแสดงผลของ Previous Button

3.5.9 การสร้าง Button Return

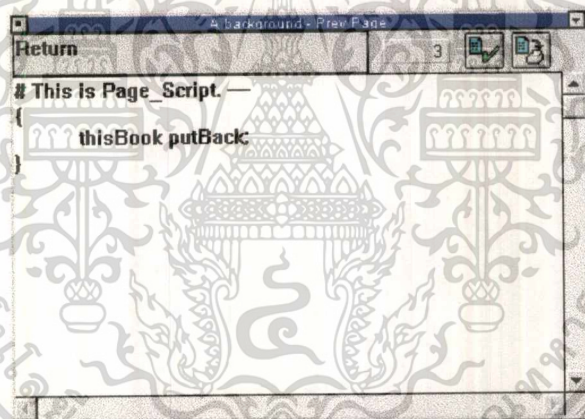
ในการสร้าง Button Return ทำการเลือกมีเดียและกำหนดรายละเอียดของปุ่ม ดังรูปที่

3.30



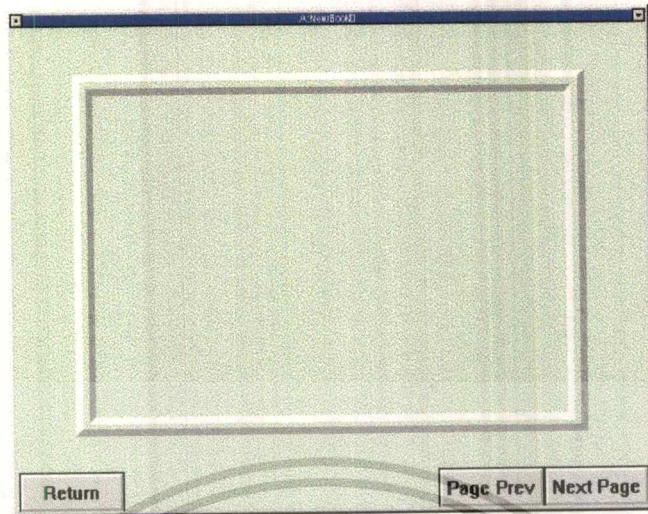
รูปที่ 3.30 การกำหนดรายละเอียดของ Button

จากนั้นทำการเขียนโปรแกรมย่อยให้กับ Button เพื่อกำหนดการทำงานของ Button ดังรูปที่ 3.31



รูปที่ 3.31 การเขียน โปรแกรมย่อยให้กับ Button

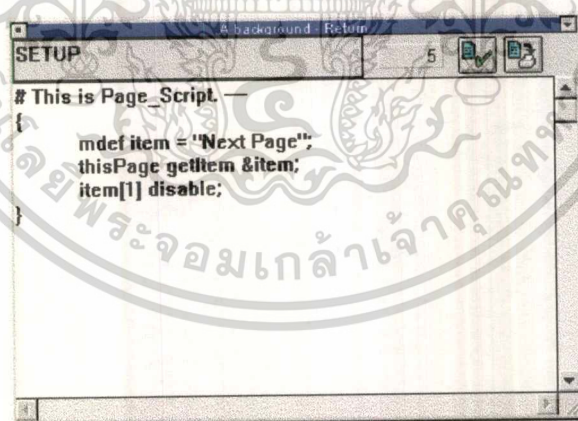
เมื่อมีการกำหนดการทำงานให้กับ Button แล้ว จะปรากฏหน้าจอดังรูปที่ 3.32



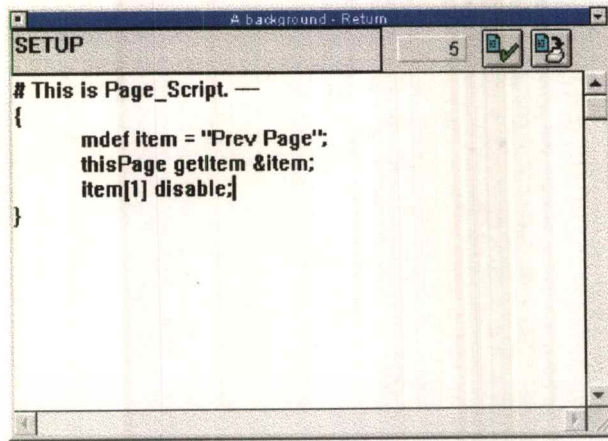
รูปที่ 3.32 การแสดงผลของ Button

3.5.10 การเขียนโปรแกรมย่อยควบคุมการทำงานของหน้าหนังสือ

การเขียนโปรแกรมย่อยควบคุมการทำงานของหน้าหนังสือ แบบ Next Page และ Previous Page แสดงดังรูปที่ 3.33 และ รูปที่ 3.34 ตามลำดับ



รูปที่ 3.33 การเขียนโปรแกรมย่อยของ Next Page



```

SETUP
# This is Page_Script. —
{
  mdef item = "Prev Page";
  thisPage getItem &item;
  item[1] disable;
}

```

รูปที่ 3.34 การเขียนโปรแกรมย่อยของ Previous Page



บทที่ 4

การทดลองและผลการทดลอง

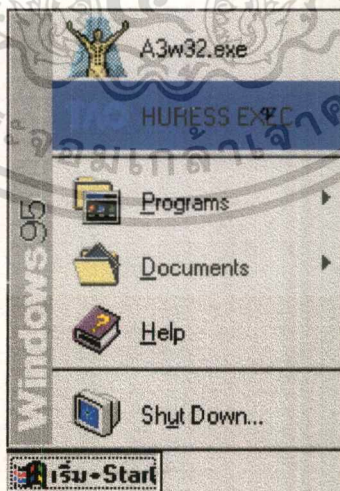
การใช้งานและทดสอบโปรแกรมช่วยสอน วิชา ไมโครโปรเซสเซอร์ Z-80 นั้น จะทำการทดสอบการใช้โปรแกรมในแต่ละส่วนตามเมนูหลักของหนังสือ โดยในส่วนนี้จะแบ่งการทดสอบโปรแกรมออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของเนื้อหา และ ส่วนของแบบฝึกหัด

จากนั้น ได้มีการทดสอบการใช้งาน ได้จริงในรูปแบบการเชื่อมโยงในระบบเครือข่าย ในส่วนนี้ได้ทดลองนำโปรแกรม HURESS ทำการติดตั้ง ในระบบ Network อื่น ๆ และเพิ่มวิชาที่สร้างเข้าไปในระบบ เพื่อทดสอบโปรแกรม

4.1 การเข้าสู่โปรแกรม HURESS

1. เปิดเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server
2. เปิดเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client
3. นำเมาส์ไปคลิกที่ปุ่ม Start เลือกเมนู Programs แล้วรันโปรแกรม HURESS BOOK

จะปรากฏหน้าจอ ดังรูปที่ 4.1



รูปที่ 4.1 การเข้าสู่โปรแกรม HURESS

4. ป้อน User name เพื่อเข้าสู่โปรแกรม ดังรูปที่ 4.2

รูปที่ 4.2 การป้อน User name

5. เมื่อเข้าสู่โปรแกรม HURESS แล้วจะปรากฏรายชื่อของวิชา ในแต่ละวิชาจะมีทั้งส่วนของบทเรียนและแบบทดสอบ ซึ่งเราสามารถเลือกได้ ดังรูปที่ 4.3

รูปที่ 4.3 การเลือกวิชาเรียน

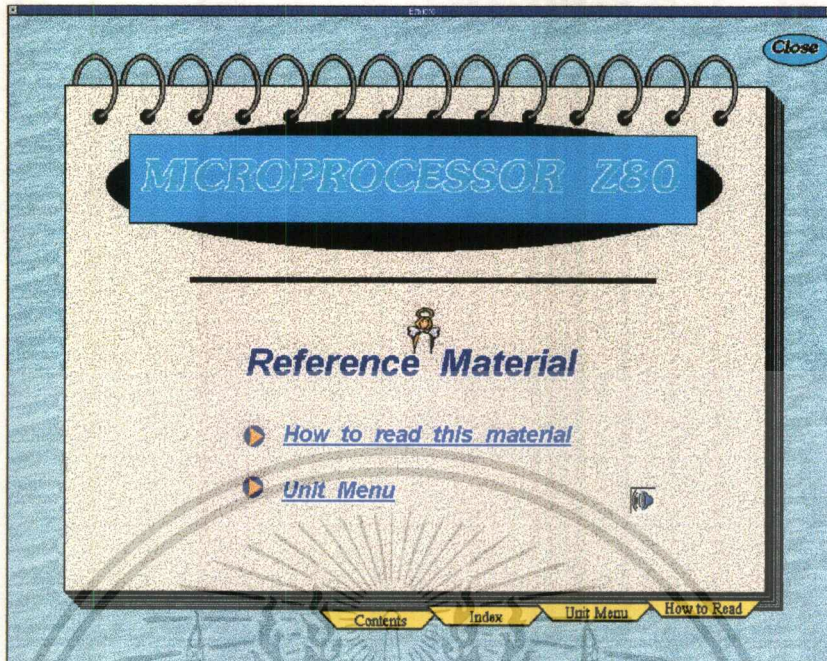
4.2 การใช้งานของโปรแกรม

4.2.1 ส่วนของเนื้อหา

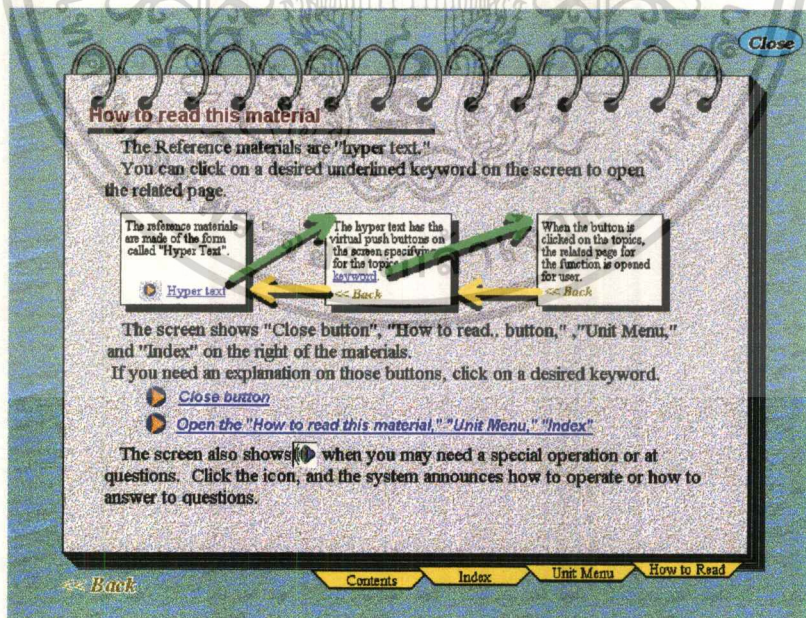
1. เมื่อเข้าสู่เนื้อหาจะปรากฏส่วนของ Front Cover ดังรูปที่ 4.4

2. เลื่อนเมาส์ไปคลิกที่ How to Read (ซึ่งเป็นข้อความแบบ Hypertext) จะปรากฏหน้าจอ ดังรูปที่ 4.5 ซึ่งเป็นคำอธิบายการใช้งาน โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Front Cover



รูปที่ 4.5 How to Read

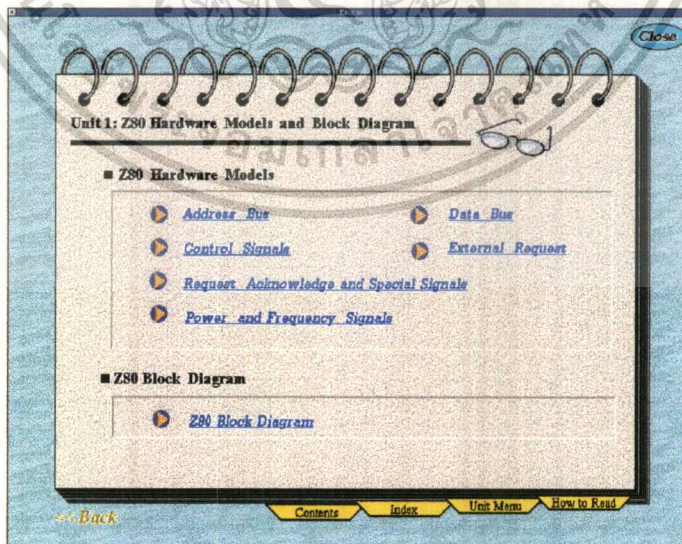
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลื่อนเมาส์ไปคลิกที่ Unit Menu จะแสดงรายชื่อของบทเรียนขึ้นมาดังรูปที่ 4.6



รูปที่ 4.6 Unit Menu

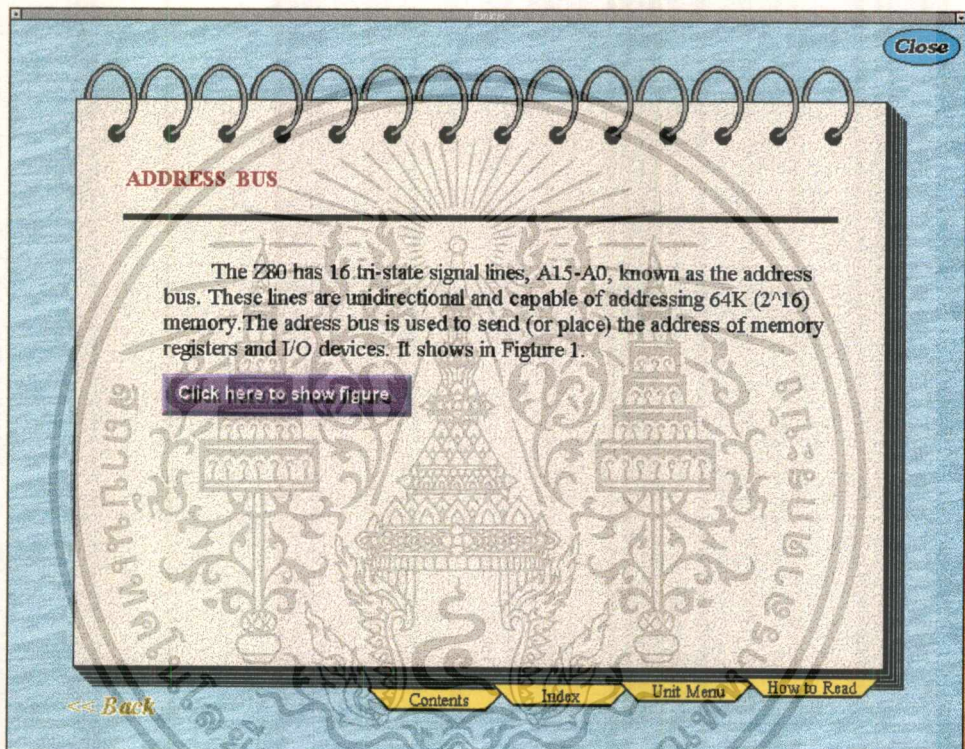
4. เลื่อนเมาส์ไปคลิกยังบทเรียนที่ต้องการจะแสดงเนื้อหาของบทเรียนนั้นๆ ดังปรากฏหัวข้อบทเรียนให้เลือกดังรูปที่ 4.7



รูปที่ 4.7 หัวข้อเนื้อหาของแต่ละบทเรียน

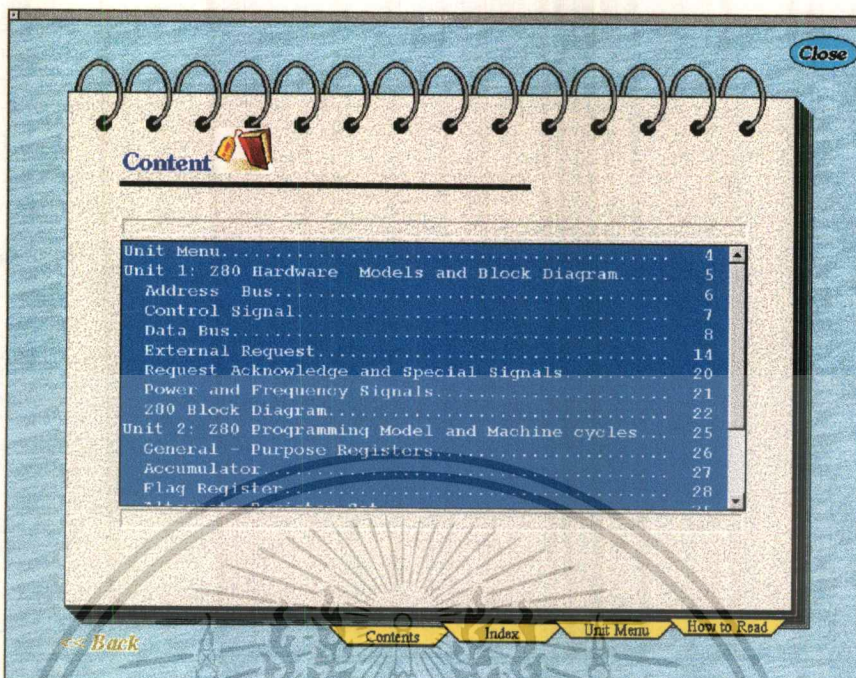
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากรูปที่ 4.7 สามารถกลับไปยังหน้า Unit Menu, Contents, Index และHow to Read ได้โดยเลื่อนเมาส์ไปคลิกที่ปุ่มนั้นๆ
6. การกลับไปยังหน้าเดิมได้โดยเลื่อนเมาส์ไปคลิกที่ปุ่ม Back
7. ในส่วนของการแสดงรูปภาพ สามารถทำได้โดยการเลื่อนเมาส์ไปคลิกที่ปุ่ม Click here to show figure ดังรูปที่ 4.8

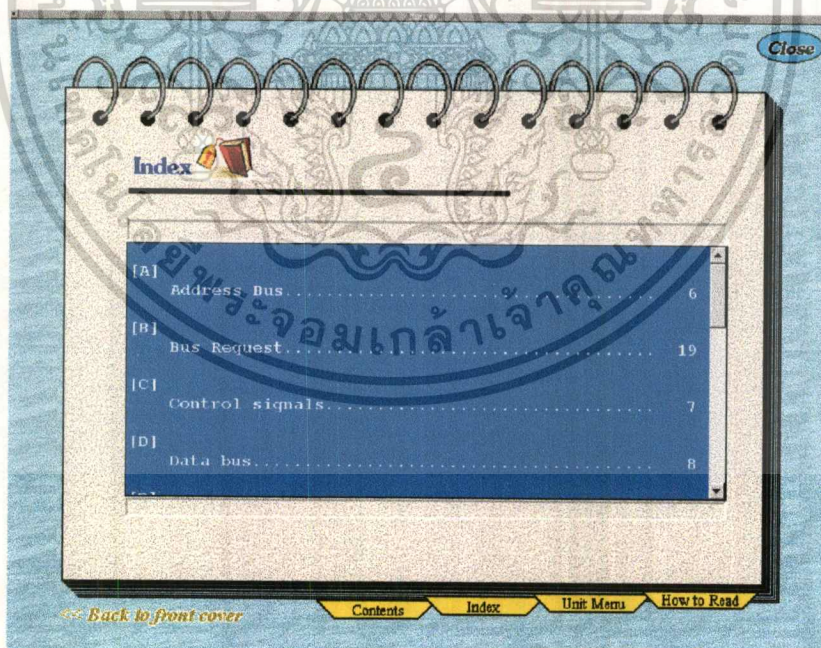


รูปที่ 4.8 ปุ่ม Click here to show figure

8. เมื่อทำการเลื่อนเมาส์ไปคลิกที่ "Contents" จะปรากฏหน้าจอจดังรูปที่ 4.9 และการใช้งานของหน้าสารบัญ (Contents) นั้น ทำการดับเบิลคลิกที่หน้าที่ต้องการไป
9. เมื่อทำการเลื่อนเมาส์ไปคลิกที่ "Index" จะปรากฏหน้าจอจดังรูปที่ 4.10 และการใช้งานของหน้าดัชนี (Index) นั้น ทำการดับเบิลคลิกคำที่ต้องการเพื่อที่จะไปหน้าที่มีรายละเอียดของคำนั้น



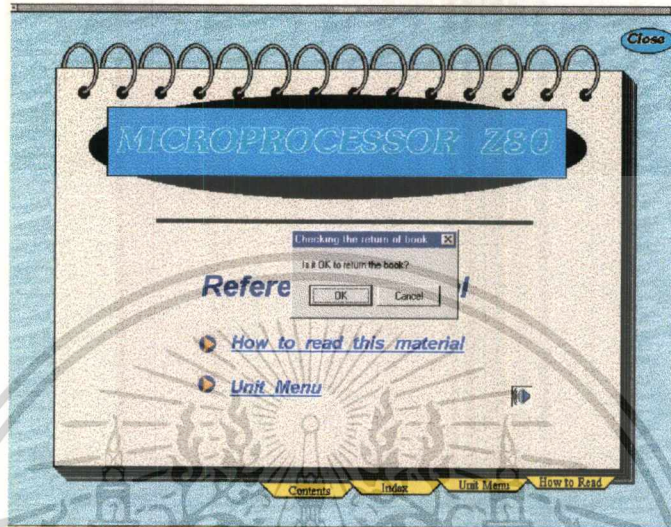
รูปที่ 4.9 การใช้งานหน้าสารบัญ



รูปที่ 4.10 การใช้งานหน้าดัชนี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

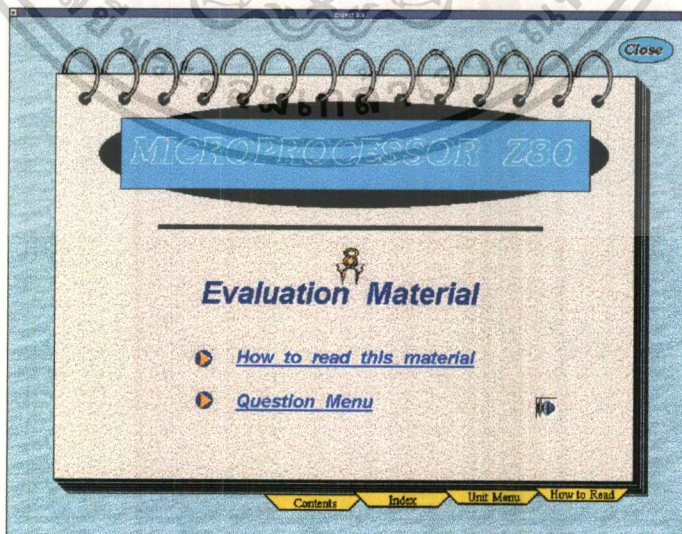
10. การออกจากโปรแกรมทำได้โดยการเลื่อนเมาส์ไปคลิกที่ "Close" ที่อยู่ตรงมุมบนด้านขวาดังรูปที่ 4.11



รูปที่ 4.11 ปุ่ม Close

4.2.2 ส่วนของแบบทดสอบ

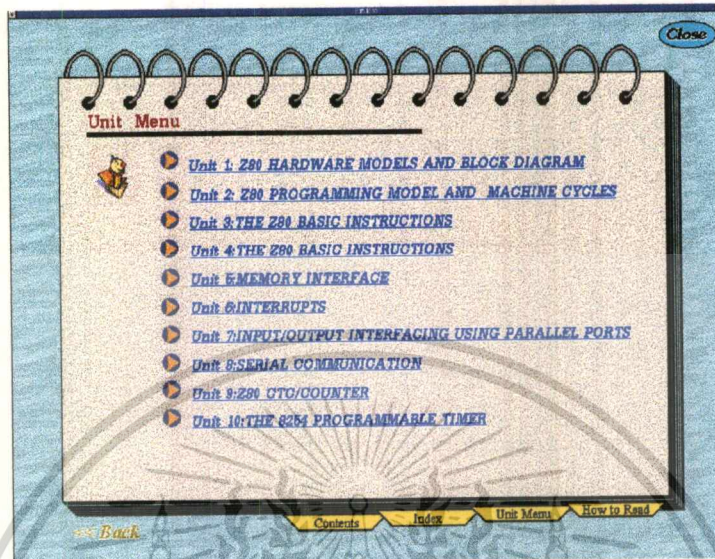
1. เมื่อเข้าสู่แบบทดสอบจะปรากฏหน้าจอหลักดังรูป 4.12



รูปที่ 4.12 หน้าจอหลักของส่วนแบบทดสอบ

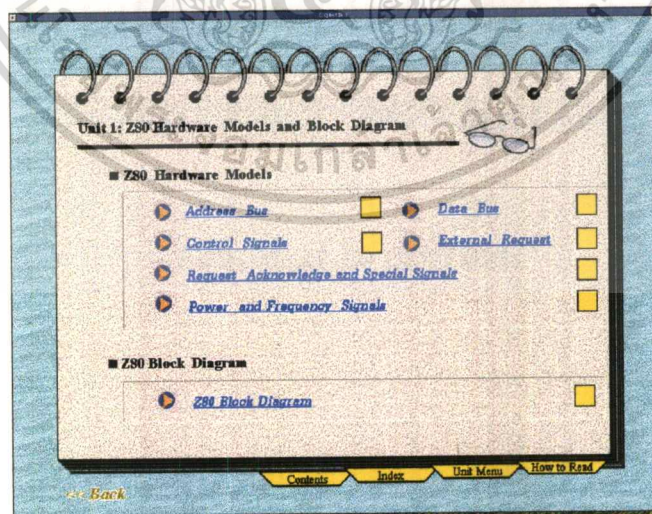
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คลิกที่ปุ่ม Question Menu เพื่อเข้าสู่หน้าจอหลักหัวข้อแบบทดสอบ ดังรูป 4.13



รูปที่ 4.13 หน้าจอหลักของหัวข้อแบบทดสอบ

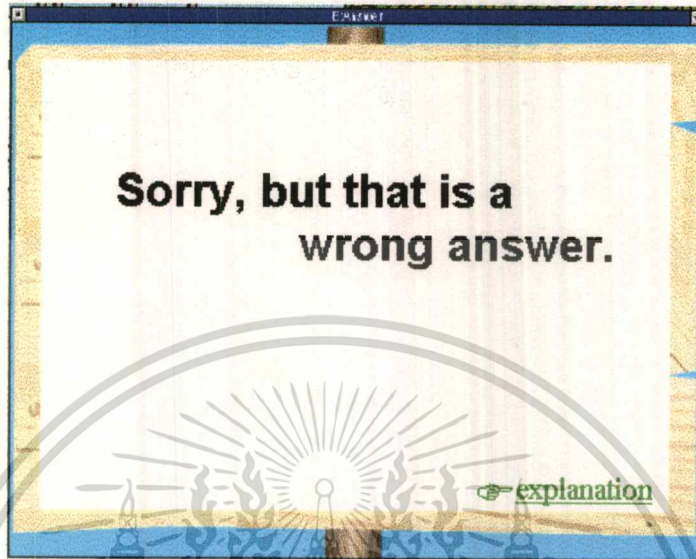
3. เลือกบทเรียนที่จะทำการทดสอบจะเข้าสู่หน้าจอแบบทดสอบหลักของบทเรียนนั้นๆ ดังรูปที่ 4.14



รูปที่ 4.14 หน้าจอแบบทดสอบหลักของบทเรียน

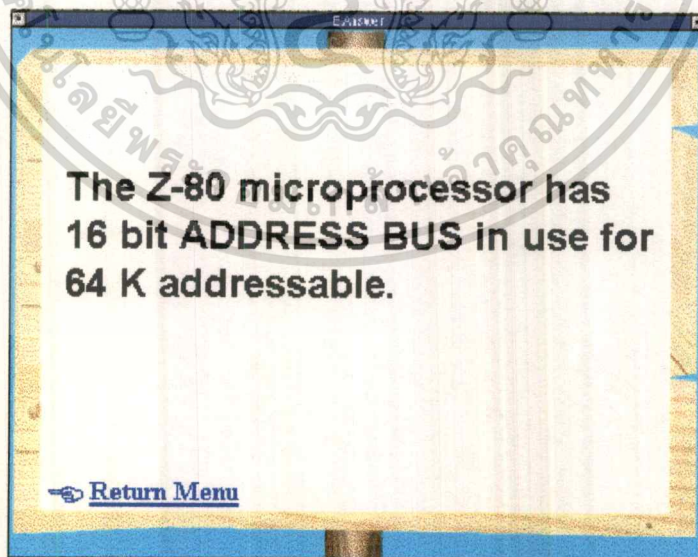
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. หากตอบผิดจะปรากฏหน้าจอดังรูปที่ 4.17



รูปที่ 4.17 คำตอบผิด

8. กดปุ่ม explanation เพื่อดูคำอธิบายเพิ่มเติม รูปที่ 4.18 แสดงคำอธิบายเพิ่มเติม



รูปที่ 4.18 อธิบายคำตอบเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เมื่อคูปอง Return จะกลับไปสู่น้ำจอบทดสอบหลักของบทเรียนพร้อมทั้งทำการ Check (ทำเครื่องหมาย) ว่าได้ทำข้อนั้นๆไปแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปปัญหา แนวทางแก้ไข และพัฒนา

ปริญญานิพนธ์ฉบับนี้ เสนอการสร้างบทเรียนโปรแกรมช่วยสอนวิชา ไมโครโปรเซสเซอร์ Z80 ซึ่งมีเนื้อหาคำบรรยายเป็นภาษาอังกฤษ ทั้งนี้เพื่อพัฒนาเนื้อหาวิชาให้สอดคล้องกับระบบของโครงการวิจัยร่วมระหว่างไทย ญี่ปุ่นและจีน ในการพัฒนาทรัพยากรมนุษย์ด้วยระบบการสอนทางไกลแบบสองทาง พร้อมสื่อประสมผ่านดาวเทียมระหว่างประเทศ โดยใช้เครื่องมือการพัฒนา CAI (AUTHORING TOOLS) ชื่อโปรแกรม HURESS ซึ่งบทเรียนโปรแกรมคอมพิวเตอร์ช่วยสอนนี้สามารถนำไปใช้เป็นบทเรียนจริงหรือนำไปใช้เป็นบทเรียนเสริมวิชา ไมโครโปรเซสเซอร์ Z80 ซึ่งผู้เรียนสามารถเรียนรู้ได้ตามความเข้าใจของตนเอง ทำให้การเรียนมีประสิทธิภาพมากขึ้นเนื่องจากข้อได้เปรียบในเรื่องของการให้ข้อมูลป้อนกลับ โดยในบทเรียนจะมีแบบทดสอบเพื่อทดสอบความรู้ความเข้าใจของผู้เรียน การเปิดโอกาสให้ผู้เรียนได้มีส่วนร่วมในการเรียนการสอนและยังลดปัญหาความแตกต่างระหว่างบุคคล และการสร้างบทเรียนคอมพิวเตอร์ช่วยสอนนั้นนอกจากจะสร้างขึ้นโดยโปรแกรม HURESS แล้วยังใช้โปรแกรมช่วยสร้างภาพอื่นๆ ทำให้สามารถช่วยออกแบบบทเรียนโปรแกรมคอมพิวเตอร์ช่วยสอนได้เป็นอย่างดี

5.1 ปัญหาที่พบในการทำงาน

5.1.1 ในระยะแรกต้องเสียเวลากับการศึกษาการใช้โปรแกรม HURESS เนื่องจากคู่มือการใช้โปรแกรมเป็นภาษาอังกฤษทั้งหมด และไม่มีพื้นฐานในการใช้โปรแกรมมาก่อน

5.1.2 โปรแกรม HURESS มีข้อจำกัดในการทำงาน ทำให้เกิดความล่าช้าในการทำงานเช่น การทำ edit แต่ละตัวต้องทำงานเสร็จสิ้นก่อนแล้วจึงจะสามารถทำกับ Media ตัวอื่นได้ไม่เช่นนั้นจะทำให้ข้อมูลถูกทำลายได้

5.1.3 เนื่องจากข้อจำกัดทางด้านฮาร์ดแวร์ ทำให้ไม่สามารถปฏิบัติงานนอกสถานที่ที่จัดไว้ได้

5.1.4 Option การทำงานของ Media บางตัว ยังมีขีดความสามารถในการทำงานน้อย ซึ่งทำให้ไม่สามารถสร้างหน้าตาของโปรแกรมออกมาตามที่ต้องการ เช่น Text Media เป็นต้น

5.1.5 การสร้างแบบทดสอบไม่สามารถที่จะกระทำได้เนื่องจากไม่มีตัวอย่างการสร้างและการใช้ Media ที่จะนำมาประกอบกันเป็นแบบทดสอบ

5.2 การแก้ปัญหา

5.2.1 ศึกษาการใช้โปรแกรมอย่างต่อเนื่องและจริงจัง

5.2.2 สร้าง Media ที่ละตัว

5.2.3 นำโปรแกรมไปติดตั้งกับระบบเครือข่ายภายนอก ทำได้โดยการเข้าไปในเซิร์ฟเวอร์และนำเอาไฟล์ที่เป็นแบบฝึกหัดของบทเรียนที่สร้างไว้แล้วเช่นวิชา Communication Network มาเป็นตัวอย่างของการสร้างแบบทดสอบโดยเข้าไปในเซิร์ฟเวอร์ที่ C:\library\Huress\CommunicationNetwork\ComnetKR และทำการ Copy ไฟล์ที่ชื่อ Data000 ของแบบทดสอบออกมา เมื่อได้มาแล้ว เราจะยังไม่สามารถที่จะเข้าไปทำการศึกษาในส่วนของโปรแกรมได้จริงเราต้องทำการแก้ไขไฟล์นั้นก่อนโดยใช้โปรแกรม Disk edit ของ Norton utilities เพื่อที่จะทำการแก้ไขให้เรามีสิทธิ์ที่จะแก้ไขข้อมูลในไฟล์นั้นได้ โดยการ Serch หาคำว่า SUPERVISOR แล้วทำการเปลี่ยนเป็นคำว่า GUEST แทน เราจึงจะทำการแก้ไขและศึกษาเนื้อหาอื่นๆได้

5.3 ข้อเสนอแนะ

5.3.1 นำโปรแกรมที่ใช้สร้าง CAI อื่นๆ มาใช้ร่วมกับโครงการฯ เช่นเดียวกับโปรแกรม HURESS ได้

5.3.2 นำโปรแกรม HURESS มา INSTALL ใช้กับเครื่อง PC และสามารถใช้งานจริงได้

5.3.4 ควรพัฒนาโปรแกรมให้เป็นระบบเปิด

5.4 ประโยชน์ที่ได้รับจากโครงการ

5.4.1 ทำให้มีความสามารถในการใช้โปรแกรมสร้างสื่อประสม HURESS

5.4.2 ทำให้การเรียนการสอนสะดวกยิ่งขึ้น

5.4.3 ใช้เป็นแบบของการศึกษาโปรแกรมสร้างสื่อประสม HURESS

5.4.4 ได้แนวทางสำหรับการพัฒนาวิชาอื่นๆ เพื่อร่วมโครงการวิจัยร่วมระหว่างไทย

และญี่ปุ่นต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 แนวทางพัฒนา

- 5.5.1 เพิ่มเทคนิคให้สวยงามเพื่อดึงดูดความสนใจของผู้ใช้
- 5.5.2 เพิ่ม Media ในการนำเสนอให้ผู้เรียนสนใจและใช้สะดวกมากขึ้น
- 5.5.3 เพิ่มบทเรียนในวิชาอื่นๆได้
- 5.5.4 สามารถนำโปรแกรมมาใช้สอนจริงได้



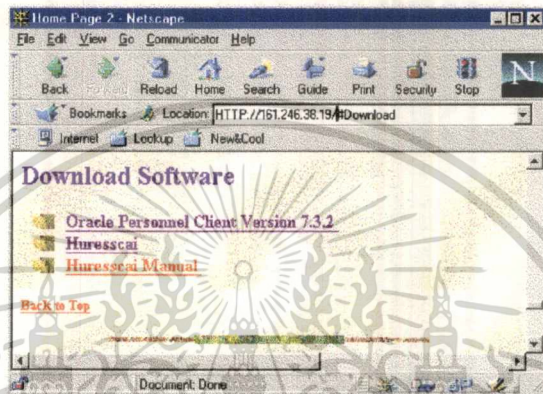


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งโปรแกรม HURESS

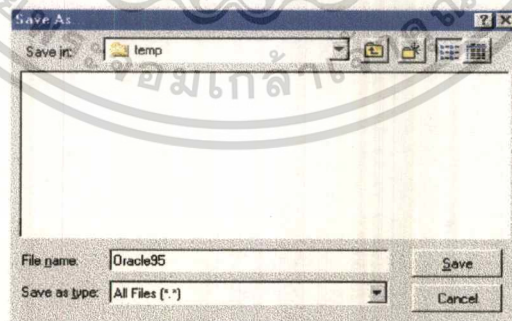
1. การ Download Software

1.1 เรียกโปรแกรม Netscape หรือ Explorer แล้ว get URL <http://161.246.38.19> ดังรูป ก.1



รูปที่ ก.1 การ Download Software

1.2 เลือก Oracle Personnel Client Version 7.3.2 เป็น Software ที่ใช้ติดตั้ง Oracle database ในฝั่ง Client ดังรูปที่ ก.2

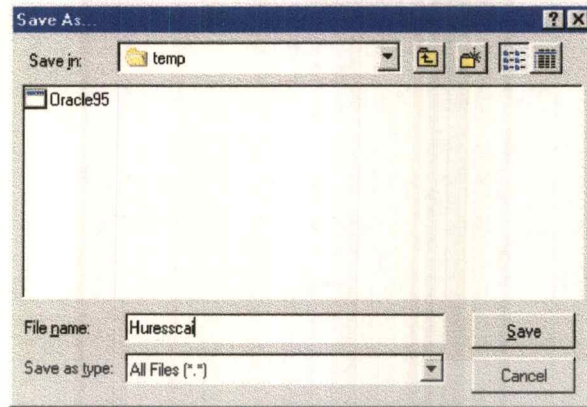


รูปที่ ก.2 ตำแหน่งที่ต้องการเก็บไฟล์

1.3 เลือก Save และรอนจนกระทั่ง Windows Dialog ด้านล่างหายไปแสดงว่าเสร็จ ดังรูปที่

ก.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 การบันทึกโปรแกรม

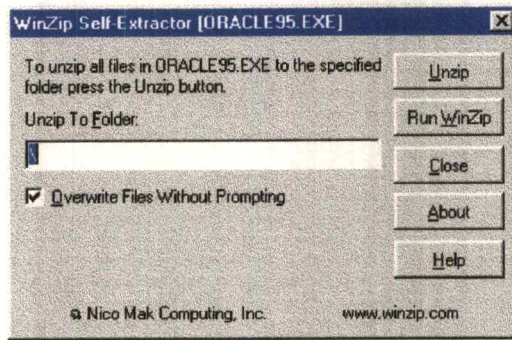
1.4 เลือก Huresscai เป็น Software ที่ใช้ติดตั้ง Huress link และการ Setup ค่า parameter ที่สำคัญในการติดต่อกับ Database Server และ Software CAI ดังรูป ก.4



รูปที่ ก.4 ตำแหน่งที่จะเก็บไฟล์

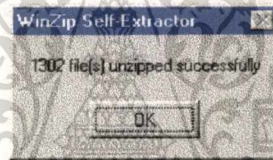
1.5 เลือก Save และรอนจนกระทั่งเสร็จ

1.6 Extract ไฟล์ Orawin95.exe ซึ่งเป็นกลุ่มของไฟล์ที่ถูก compression โดยใช้ WinZip การ Extract กระทำโดย Double Click ไฟล์ Orawin95.exe ดังรูปที่ ก.5



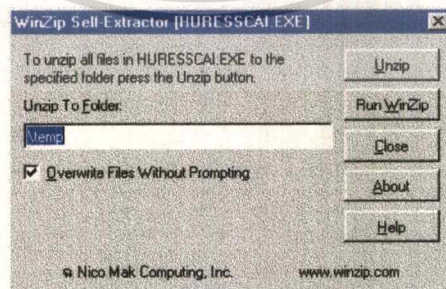
รูปที่ ก.5 การ Extract ไฟล์ Oracle95.exe

จากหน้าจอ Click Unzip รอจนกระทั่งหน้าจอจะปรากฏ Windows successfully แล้ว Click OK และ Click Close บน Windows Self-Extractor ดังรูป ก.6



รูปที่ ก.6 ทำการ Extract เสร็จสมบูรณ์

Extract ไฟล์ Huresscai.exe ซึ่งเป็นกลุ่มของไฟล์ที่ถูก compression โดยใช้ WinZip การ Extract กระทำโดย Double Click ไฟล์ Huresscai.exe ดังรูป ก.7



รูปที่ ก.7 การ Extract file Huresscai.exe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหน้าจอ Click Unzip รอจนกระทั่งหน้าจอจะปรากฏ Windows Dialog ดังรูป
Click OK แล้ว Click Close บน Self-Extractor Windows ดังรูป ก.8



รูปที่ ก.8 การ Extract เสร็จสมบูรณ์

Subdirectory \temp จะปรากฏไฟล์ต่างๆดังรูปที่ ก.9



รูปที่ ก.9 ไฟล์ใน Sub Directory temp

2. การติดตั้ง Software

เลือกติดตั้ง Software Oracle ก่อน โดย Double Click ไฟล์ Oraclesetup จะปรากฏหน้า
จอดังรูป ก.10



รูปที่ ก.10 การติดตั้ง ORACLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Language : English

Company Name : CAI

Oracle Home : C:\ORAWIN95

Installation Options

Application Developer (Complete)

Runtime (Database Only)

Custom

Software Asset Manager

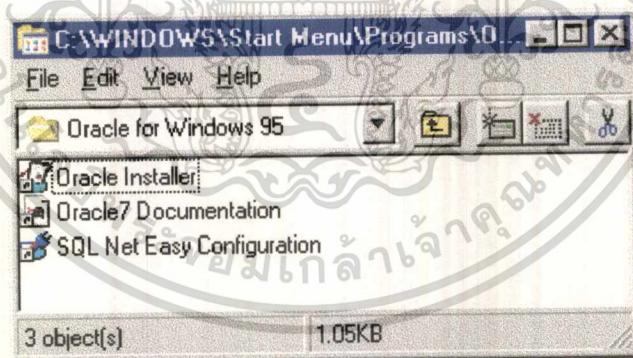
Products available

Oracle Call Interface 7.2.2.3.1

Oracle Installer 3.1.2.1.3

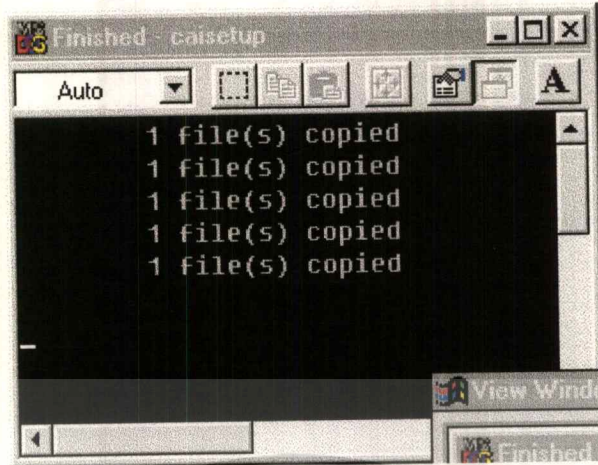
SQL*Net Client 2.2.2.1.0

หลังจากติดตั้ง Oracle เรียบร้อยแล้วจะปรากฏหน้าจอ Windows ดังรูปที่ ก.11



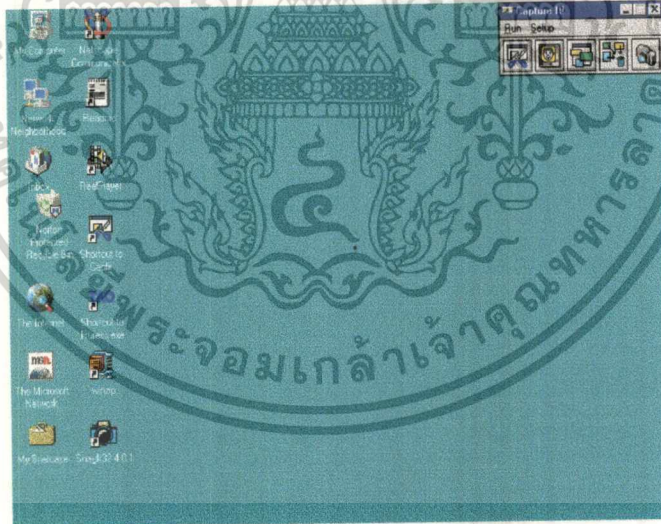
รูปที่ ก.11 การติดตั้ง ORACLE เสร็จสมบูรณ์

การติดตั้ง Software HuressCAI โดย Double Click ชื่อไฟล์ caisetup.bat จะปรากฏ หน้าจอดังรูปที่ ก.12



รูปที่ ก.12 การติดตั้ง CAI setup เสร็จสมบูรณ์

การ Move Shortcut to Huress.exe ไปไว้ยังหน้าจอ Desktop ของ Windows จะแสดง
ผลดังรูปที่ ก.13

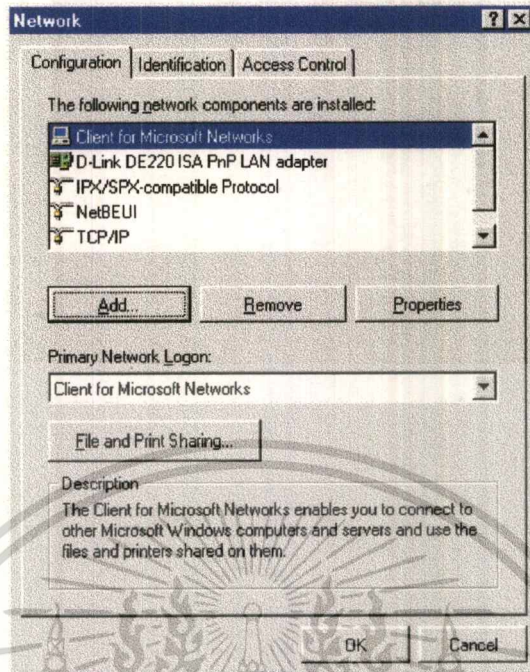


รูปที่ ก.13 Short cut ของ Huress

2.1 การ Configuration Network Protocol

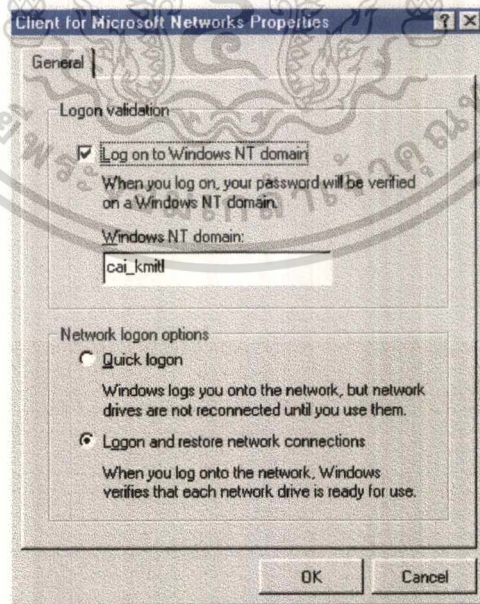
1. กำหนด Domain login จาก Client For Microsoft Networks ดังรูป ก.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.14 Configuration Network

หน้าจะปรากฏ ดังรูป ก.15 แล้วกำหนดค่าตามรูปข้างล่างนี้

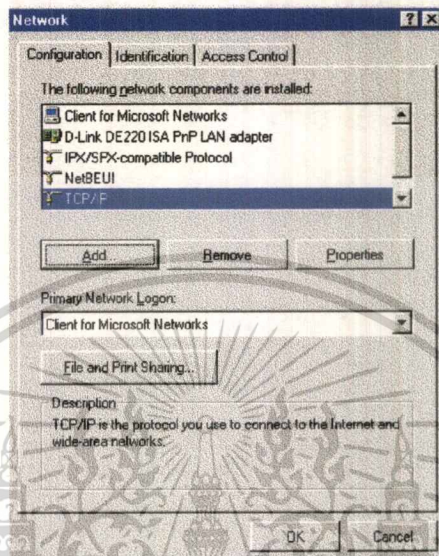


รูปที่ ก.15 Domain Sever

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

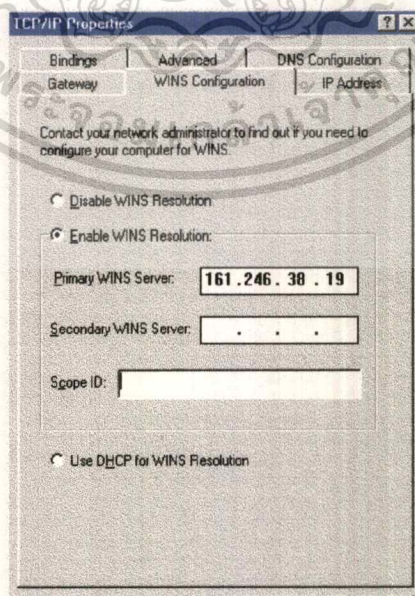
ต่อจากนั้น เลือกรายการไปที่ Protocol TCP/IP แล้ว Click บน Properties ดังรูปที่

ก.16



รูปที่ ก.16 Protocol TCP/IP

หน้าจอจะปรากฏปุ่มให้คลิก Click Mouse ไปที่ WINS ดังหน้าจอรูปที่ ก.17



รูปที่ ก.17 Properties ของ Protocol TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการกรอกข้อมูล ตามรูป โดยให้ใส่ IP Address ของเครื่อง Server แล้ว ทำการ Restart เครื่องคอมพิวเตอร์ ต่อจากนั้น Login เข้า User ชื่อ student และ Password ชื่อ student ชื่อ Domain Name ชื่อ cai_kmitl



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดรูปแบบของหนังสือด้วย โปรแกรมย่อย

ในการสร้างหนังสือที่ต้องการขึ้นมาแต่ละเล่มนั้นจะต้องมีการกำหนดรูปแบบการทำงานของหนังสือเล่มนั้นๆ ซึ่งจะกำหนดได้ด้วยโปรแกรมย่อยดังนี้

โปรแกรมย่อยชื่อ "SETUPBOOK"

ข้อกำหนดของโปรแกรมย่อย

1. จะถูกเรียกเพียงครั้งเดียวเมื่อตอนเริ่มเปิดหนังสือครั้งแรกเท่านั้น
2. จะต้องสร้างในหน้าแรกเท่านั้น

เครื่องหมาย "#" เป็นการกำหนดให้โปรแกรมรู้ว่าข้อความหลังจากเครื่องหมายนั้นจะไม่นำมาทำงานเนื่องจากเป็นเครื่องหมาย COMMENT

```
# === SETUPBOOK ===
```

```
# Procedure for initialization of the book
```

```
{
```

```
# เป็นการกำหนดตัวแปรสำหรับกำหนดค่ารูปภาพที่จะใช้ทำเป็นพื้น(Background)
```

```
  mdef pic;
```

```
# Message for checking the return of the book
```

```
# เป็นการกำหนดข้อความในกรอบแสดงข้อความเมื่อต้องการปิดโปรแกรม
```

```
  bdef CloseBookMes="Checking the return of book","Is it OK to return the book?";
```

```
  bdef Back="front cover";
```

```
# Variable for recording the return page
```

```
  bdef RetPage;
```

```
  bdef RetPage2;
```

```
  RetPage = (0-1);
```

```
  RetPage2 = (0-1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่งที่ใช้ในการตรวจสอบสถานะของหนังสือ

```
thisBook getInformation &pic;
```

```
pic[0] = pic[4];
```

```
pic = "c:\\micro\\picture1\\background1.bmp";
```

เป็นการกำหนดรูปภาพที่เป็นพื้นของหนังสือ

```
thisBook setPicture &pic;
```

Open book

เป็นคำสั่งที่ทำให้เปิดหนังสือและแสดง โดยอัตโนมัติเมื่อเรียกยืมหนังสือ

```
thisBook open;
```

```
}
```



โปรแกรมย่อยชื่อ "SETUP"

ข้อกำหนดของโปรแกรมย่อย

1. หากจำเป็นต้องมีทุกหน้า
2. เป็นการกำหนดค่าเริ่มต้นให้กับหน้านั้นๆ
3. จะถูกเรียกใช้เมื่อเวลาเปิดหน้านั้นๆขึ้นมาทุกๆครั้ง

```
# === SETUP ===
```

```
# Procedure for initialization of the book
```

```
{
```

```
# เป็นการกำหนดค่าตัวแปรชื่อ Back ให้เป็นข้อความคำว่า "front cover"
```

```
Back="front cover";
```

```
}
```





ภาคผนวก ค
รูปแบบการใช้งานของฟังก์ชันต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการใช้งานของฟังก์ชันต่าง ๆ

This manual deal with MEDIA-ADK which operates on Borland C++ ver4.52 object Windows (OWL)2.5 and provides media development kit involving C++ class library designed to develop Huress-Book AS-IN media.

In case you desire to develop Huress-Book applications but if you are not satisfied with the functions offered by the standard media, application developers can create new media programs with unique functions. In order to create new media programs, you need have a basic knowledge on:

1. Borland C++ ver 4.52 OWL 2.5 Programming
2. Outline of Huress-Book and its operation
3. Outline of media on Huress-Book

This manual discuss media programming methods assuming that users are familiar with the above-mentioned software. Besides, it is essential that user of MEDIA-SDK have proper hardware and software operating on Huress-Book and Borland C++ ver4.52 OWL 2.5

1. The terms and the description format of MEDIA-SDK

In this manual, special terms of the MEDIA-SDK are used beside Huress-Book terms.

DW_OBJECT	The object to be managed by Huress-Book
DW_MESSAGE	The message to be processed by Huress-Book
DW_METHOD	The procedure to be managed by Huress-Book
DW_SELECTOR	The string recognizing DW_MESSAGE
DW_CLASS	The DW_OBJECT of classes to be managed by Huress-Book
Media Class	The DW_OBJECT to manage each media (same type of media)
Media Instance	The DW_OBJECT to create each media
Media Method	The DW_OBJECT included in the media class and media instance
Media Message	The messages for the media class and media instance
Media View	The contents displayed on the screen of the media instance. It is a GUI of the media instance.
Book View	The contents displayed on the screen of the book (pages).
Identifier	The file open identifier for the media, etc.
Identification Number	The unique number of media, etc.

Media Information Dialog	The modal dialog to update media information included in the media instance
Media file	The file to hold media information created for each media instance.
Huress-Book Class Name	The name of media class managed in Huress_-Book.
Huress-Book Class Number	The class number of media class managed in Huress-Book.

In this manual, functions and DW_METHOD included in the DW_OBJECT are represented as follows.

[Format] Calling format
 goByPageNo DW_SELECTOR
 stage Return value
 arg A variable of the arguments
 An argument to represent input
 An argument to represent output
 [Comment] Explanation of METHOD functions

Sample 1 DW_METHOD

```

goByPageNo
[format] stage=thisBook goByPageNo arg;
        arg[0] Page Number (An integer DW_OBJECT)
        arg[1] The DW_OBJECT of the page
stage    The DW_OBJECT for true or false
  
```

[Comment] This message closes the currently opened page of the book, then open the page specified by the page number. If the message is issued from and to the current page, the message return no value since it quits the current page immediately.

Sample 2 Function (A function of the C++ language))

```

HPK_newBasicObject
[format] HPK_ObjectId HPK_newBasicObject(
        HPK_classId xclass    //A type of the basic DW_OBJECT
);
[Comment] This function creates the basic DW_OBJECT specified by xclass.
        HPK_Integer    an integer DW_OBJECT
        HPK_Float      A floating point DW_OBJECT
  
```

HPK_Character	A character DW_OBJECT
HPK_String	A string DW_OBJECT
HPK_Point	A position DW_OBJECT
HPK_Rect	A rectangle DW_OBJECT
HPK_Boolean	A logical value DW_OBJECT
HPK_Atom	An atom DW_OBJECT

[Example] r = HPK_newBasicObjec(HPK_Integer);

An integer DW_OBJECT is created into r.

2. Installation of the MEDIA_SDK

Let's create a parent folder to be used for developing media. This folder is called development root folder temporarily. This folder creates the following folder:

- A folder storing MEDIS-SDK files (\SDK)
- A folder storing sample programs (\SMP)

Copy the files from the Huress-Book folder in to the development folder.

\MEDLASDK\SDK Copy all the files of the file extension. 'H' and '.LIB' to the development folder \SDK.

\MEDLASDK\SMP Copy all the files to the development folder \SMP.

The list of MEDIA-SDK files are:

MEDLASDK\SDK\ SDK.H	The main header of the MEDIA-SDK
SDK_HPK.H	The header of the Huress-Book interface library
SDK_REC.H	The header of the media message dispatch in the MEDIA
SDK_CLS.H	The header of the media class in MEDIA-SDK
SDK_INS.H	The header of the media instance in MEDIA-SDK
SDK.CPP	The main source of the media in MEDIA-SDK
SDK_REC.CPP	The source of the media message dispatch in MEDIA-SDK
SDK_CLS.CPP	The class source of the media in MEDIA-SDK
SDK_INS.CPP	The source of the media instance in MEDIA-SDK
SDK_ETC.CPP	The source of the function library in MEDIA-SDK
SDK_LIB	The MEDIA-SDK importing library
HPK.LIB	The Huress-Book interface importing library

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEDLASDK\SMP\

SMP.IDE	A project of a basic sample program
SMP.H	Header of basic sample program
SMP.CPP	Source of a basic sample program
SMP.RC	A resource of a basic sample program
SMP.RES	A complied source of a basic sample program
SMP.DEF	A definition file of a basic sample program
SMP.A.H	Header of a Type A media sample program
SMP.A.CPP	Source of Type A media sample program
SMP.A.RC	A resource source of a Type A media sample program
SMP.A.RES	A complied resource of a Type A media sample program
SMP.A.DEF	A definition file of a Type A media sample program
SMP.B.H	Header of a Type B media sample program
SMP.B.CPP	Source of Type B media sample program
SMP.B.RC	A resource source of a Type B media sample program
SMP.B.RES	A complied resource of a Type B media sample program
SMP.B.DEF	A definition file of a Type B media sample program

3. Procedure for media programming

The media program is created according to the following procedures.

3.1 Creation of media development folders

Base on the procedure set forth in Section [MEDIA-SDK Installation] , a media development folder id created , which can be named at the option of a developer. Files to be created hereafter should be stored in the particular folder.

3.2 Creation of project files (IDE)

Develop project files by referencing SMP.IDE. Refer to OPTION assigned to SMP.IDE to assign PROJECTOPTION for project files, where you be required to adjust assignment of folders and files dependent on new media programs.

3.3 Creation of module definition file

Develop module definition file by referencing SMP.DEF.

Contents of <SMP.DEF>

LIBRARY SAMPLE

DESCRIPTION 'Huress-Book Media SAMPLE'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EXETYPE WINDOWS

CODE PRELODE MOVEABLE DISCRARBLE

DATA PELOAD MOVEABLE SINGLE

3.4 Creation of media program source files

Create the following resources by referencing Section 4 (Sources Programming Procedures).

3.5 Creation of resource definition file

Create the following resource by referencing SMP.RC.

-Icon for media program identification

Create Icon to be used for media program assignment in the [Book Information]0.

Huress-Book uses the first Icon in the resource a media program identifier Icon.

-Media class identifier bit map

Create the bit map with a specification of 22*2016 color. Where you desire to create multiple media classes with the media program,create a bit map for each media class.

Upon creation of the multiple class, this bitmap should be specified according to Section 4

-Media information dialog

Create a dialogue capable of concurrently updating common media information and individual media information. Section 8 and Section 8.9 explain common media information. Individual media information should be designed by a media developer. Where multiple media class are generated using the same media program dialogue should be creates for each class.

3.6 Compiling

After confirming that the following items have been assigned in the project file,proceed to the compile process.

Source programs	(.CPP.H)
resource file	(.RES)
module definition files	(.DEF)
Huress-Book interface import library	(HPK.LIB)
MEDLASDK import library	(SDK.LIB)

After normal termination of compilation, a new media program. DLL is generated in the media development folder. Copy this in the folder\EXEMEDIA in which Huress-Book is installed

3.7 Introduction

The media program can be used if the BOOK utility is used for editing and the media program id installed in the utility.

4. How to Make Source Programs.

In this chapter, using a simple sample we will explain the basic of creating a media program. For creating higher level media programs, refer to other sample programs and the explanation Chapter 9.

4.1 Definition of the "Include Files" Used

In the source file of the media program, define in order the following items. The file to perform "include" on is <sdk.h>. This contains all the files which need to be "included" for MEDIA-SDK.

4.2 Definitions of the Global Variable

TModule	*the Module = NULL;	A media program,(.DLL)
		A pointer to module
Tmedia Class	*mySMPclass;	A pointer to media class
HPK_ObjectId	nullObject = {NULL}	An underfined DW_OBJECT
HPK_ObjectId	trueObject;	A true logic value DW_OBJECT
HPK_ObjectId	falseObject;	A false logical value DW_OBJECT

4.3 Definitions of the DllEntryPoint

```
bool WINAPI DllEntryPoint
{
    if(reason == DLL_PROCESS_ATTACH){
        HBITMAP bitmap;
        trueObject    =b2oB(TRUE);
        falseObject   =b2oB(False);

        TIndexArray(0,0,0);

        The Module = new Tmodule("SMP",hInstance);
        bitmap = LoadBitmap(hInstance, "SMP");
        mySMPClass = new TSMPClass("SMPPRG" , "SMP",93,bitmap);
    }
    return TRUE;
}
```

"SMPPRG" is the name of the media program which will be created here.

"SMP" is the Huress-Book class name of the "media class" which we define here.

"93" is the Huress-Book class number of the "media class" which we here.

4.4 Definition of the Call Back Function for Handling Completion

```
extern "C" {
void FAR PASCAL_expor DetachMediaClass()
{
    delete mySMPCClass;
}
}
```

At the time of completing the media program, this function is used by Huress-Book.

4.5 Definition of TMediaClass, a Derivative Class

\from Huress-Book the following BW_MESSAGE (requests) is communicated to "media class."

The requests to "media class" are:

- new Create a new media "instance."
- load Load media information from the media file and create a media "instance."
- pate Load media information from the clipboard and create a media "instance."
- getError Return media class error information.

In "media class," the capability to respond to each of these requests is needed, and TmediaClass of Media-SDK handles this processing on behalf of the "Constructor" and the make Instance function.

For Details on the DW_MESSAGE to "media class," refer to 7, "Desk DW_OBJECT Reference."

] Here-below is a sample definition of the derivative class.

```
<Definition of class>
class TSMPClass :publicClass
{
<Definition of member function >
//constructor (override is necessary)
TSMPClass::TSMPClass(char*progName,char*className,
                    HPK_ClassId no,HBITMAP paletteBmp)
                    :TMediaClass(progName,className,no paletteBmp)
{
}
//create a media instance (override is necessary)
Tmedia Instance *TSMPClass::makeInstance() // creation of a media instance
{
```

```

//create a media instance and return the pointer
return new TSMPInstance();
}

```

4.6 Definition of TmediaInstance, a Derivative Class

From Huress-Book the following DW_MESSAGE (requests) is communicated to "media instance," and within this class, the capability to respond to each of the requests is needed.

For details about the DW_MESSAGE to "media instance", refer to "7.Desk DW_ONJECT Reference."

Requests to "media instance":

save Save the media information.

remove Delete the information (media file).

copy Copy the media information into the clipboard.

point Move the media instance.

size Change the size of the media instance.

draw Enable the media view.

invalid Temporarily select the media.

show show the media.

hide hide the media.

setFlage Set execute mode.

getError Return media error information.

dispose Delete media instance.

info Renew the media information by means if the media information dialog.

asString Return the name of the media instance.

move change the (rectangle) coordinate of the media instance.

setFocus Set focus on media instance.

getPoint Return the coordinates (position) of the media instance.

getSize Return the Size of the media instance.

getRect Return the (rectangle) coordinate of the media instance.

TMediaInstance of MEDIA-SDK handle almost all the above processing on be half of the developer , but it is necessary to override the "member " function in the following sample. Also, in media instance, by defining it own reception-capable DW_METHOD , it can be used from page procedure.

<Definition of a class>

```
class TSMPInstance: public TMediaInstance
{
protected:
    BOOL load();
    BOOL save();
    BOOL information();

public:
    TSMPInstance():TMediaInstance(){}
    BOOL create(short mode ,HPK_ObjectId class,HPK_ObjectId*arg);
    void DMTest1 (TMessageInfo&mi); //original DW_METHOD
    //next line id necessary when original receivable DW_METHOD is offer to media instance
    DECLARE_METHODODE_TABLE(TSMPInstance);
};
```

<Definition of member functions>

```
// creation of a media view
BOOL TSMPInstance::create(short mode ,HPK_objectId classId,
                           HPK_ObjectId *arg)
{
    //common process to create view
    if(TMediaInstance::create(mode,classId,arg) ==False)
        return FALSE;
    //create a media view here
    //this sample create a button view
    mediaWin = new Tbutton (
        book win,
        -1,
        header. name,
        header.x, headr.y. header.h
    );
    media win->Create();
    return TRUE;
}

//load media information (override is necessary)
```

```

BOOL TMP Instance ::load ()
{
    if(TMediaInstance::load()==FALSE) // loading a media common information
        return FALSE;

    // load individual information from media information here
    return TRUE;
}

//save media information (override is necessary)
BOLL TMPInstance::save()
}

// update media informations (override I necessary)
BOOL TSMPInstance::information()
{
    // execute a dialog box to update media informations (common information and
    //individual information)here
    // use media Win or bookWin for parent window
    Return FALSE; // no information updated
}

// definition of original receivable method
void TSMPInstance::DMTest1(TMessageInfo&mi)
{
    static HPK_ClasId ct[] = {HPK_String};

    //test of parameters
    if(mi.testBasicObject) (ct,1)==False){
        mi.result = null object; // stop the procedure
        return;
    }

    ::Message Box(media Win->HWindow, "", "test1",MB_OK);
    mi.result = trueObject;
}

/dispatch table of original receivable W_METHOD
DEFINE_METHOD (TSMPInstance,TMediaInstance)
    METHODENTRY("test1" ,DMTest1 ),
END_METHOD;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 Definition of Media View Class (Derivative class of Window)

In general, with media instance the media view must be displayed in a window (HWND) specified by Huress-Book. If this display is done in a derivative class of Twindow, it can be done very simply.

for this reason, define derivative class of Twindow which have the functions set by the developer. Among these derivative classes , you can be Tcontrol, Tbutton ,TFrameWindow, ect. but Tdialog is not appropriate.

4.8 Notes on Creating Media programs

4.8.1 About the media view class

with media instance the content (method) of processing must be changed according to the mode communicated from desk DW_OBJECT.

Do the processing in accordance with the selected mode.

(When using Browser mode)

Prohibit renewal of media information.

(When using Edit mode)

Allow renewal of media information.

Ignore all mouse events with media view or alternatively, pass events such as mouse movements or clicks received by the media to the master window (Book View.)

In the basic sample, inside the DW_MESSAGE response function, DMDraw, we switch black and forth between enabling and disabling the window.

4.8.2. About dialog box

In the media program, When using dialog box in cases other than the media information dialog box, use the usage start statement (startDialog) and the usage end statement (endDialog) in respect to Desk DW_OBJECT. (Refer to 9.Type A Media Sample.)

4.8.3. When creating dialog box or window

With a constructor such as TWindow, TDialog, and TWindowFrame, always specify its own DLL module (Tmodule).

4.8.4. About window messages

With media view, when using a derivative class of Twindow, there are limitations on window messages sent to Book view, the master window.

WM_COMMAND and WM_SYSCOMMAND may not be sent.

4.8.5. About assignment of Huress-Book class names and Huress-Book class numbers be unique within Huress-Book. These are administered through our development group, so kindly inquire as necessary.

4.8.6. About the MEDIA-SDK source code

The goal of the SDK source code is to provide a hint as to how to create a media program with Huress-Book. Review it thoroughly and understand its outline, including the class layers (overall -- specialized construction) and object organization (overall -- component construction.)

Also, when creating a media program, refer to detailed functional capabilities from the parent (super-) class down through successive member functions and data members.

By becoming well versed in MEDIA-SDK, it will be possible to create media programs with programming language environments other than Borland C++ ver4.52 OWL2.5.

However note that in order to guarantee operation of standard media, you may not modify the source code or re-do the SDK-DLL in any way.

4.8.7. About accelerators

When browsing books, media involving input require the use of an accelerator. However, in the book window, a dedicated accelerator is provided, so the (external) accelerator must be reconfigured.

Also, for media, WM_COMMANDs with ID numbers less than 2000 will not be communicated. Use ID numbers than 2000.

4.8.8 About waiving the execute right

A media program may not generate a command to waive the execute right while DW_MESSAGE is being executed. For example, the GetMessage of Windows API would be such a command.

4.8.9 About changing the palette

In cases when the media program changes the system palette of Windows, by all means handle it by using the Huress-Book Interface Library is not used and processing is done only with Windows API, the media will not run normally.

5. Huress-Book Interface Library Reference

Huress-Book Interface Library provides the execution environment for various objects of Huress-Book. In this execution environment, it performs DW_OBJECT management, Variable management and Message management.

In DW_OBJECT management, various load modules on Huress-Book do the direct management (Creation and selection) for related DW_CLASS or DW_INSTANCE. This is just a closed loop management in

side of the module, therefore it is necessary to create it can notify the DW_OBJECT ,that was created in the load module , to Huress-Book that recognizes the object.

Afterward,with the aid of DW_OBJECT, Huress-Book provides by specifying the object.

Variable management provides the variables for procedure. The variables are able to use for information transfer between DW_OBJECT.

MESSAGE management has the function to send DW_MESSAGE in order to transmit information between DW_OBJECT.

In Huress-Book interface library , there are many C language functions to make possible to create new media programs. MEDIA-SDK class library ,that is description in [8. MEDIA -SDK class library reference] uses this Huress-Book interface library. C language function using in Huress-Book interface library are implement in HPK.DLL.

Types of DW_OBJECT

-Desk DW_OBJECT

Desk DW_OBJECT creates the desk and only one.

-Bookshelf DW_OBJECT

Bookshelf DW_OBJECT create the bookshelf. It is create for each bookshelf connected to the desk.

-Book DW_OBJECT

Book DW_OBJECT create the book. It is when the book is brought in on the desk.

-Page DW_OBJECT

Page DW_OBJECT create the page. It is created for the open page while the book is on the desk.

-Basic DW_OBJECT

Basic DW_OBJECT is needed for the information transfer between DW_OBJECT's and it is divided into the 9 type of DW_OBJECT as follows. The creation of these DW_OBJECT can be handled by services (function) provided of these DW_OBJECT is done automatically.

integer DW_OBJECT

real number DW_OBJECT

Character DW_OBJECT

Text DW_OBJECT

Placement DW_OBJECT

Rectangular DW_OBJECT

Logic DW_OBJECT

Atom DW_OBJECT

For assigning and general purpose dataa domain.

Variable DW_OBJECT For creating variable

-Media DW_OBJECT

Media DW_OBJECT creates the media. There are media DW_OBJECT for each media. This is the MEDIA-SDK to support for create these DW_OBJECT programs.

Text DW_OBJECT

Button DW_OBJECT

HPK Constant

HPK_MAXARG	20	//max. number of the transferring in //and out DW- OBJECT
HPK_SELLEN	32	//Max. bytes of the DW_SELECTOR
HPK_CLSLEN	32	//32 Max.bytes for Huress-Book class name
HPK_MAXINDEX	500	//Max. number of elements of variable
HPK_MEDIAPAL		Register partition for palette for media
HPK_Integer	1	//integer
HPK_Float	2	//real number
HPK_Character	3	//character
HPK_String	4	//text
HPK_Point	5	//placement
HPK_Rect	6	//rectangular
HPK_Boolean	7	//logic
HPK_Atom	8	//atom
HPK_Variable	19	//variable(Special Object)
HPK_NULL	0	//NULL object
MBONO	20	// number o class no. o max. basic
ENOCCLASS		undefined class
ECLASSID		illegal class ID
EDUPCLASS		duplicate class
EINSTAANCE		illegal instance
EMESSID		illegal message ID
EDUPVALI		duplicate variable
ENOVAL		undefined variable
EVALI		illegal variable
ENOMEM		lack of memory space
EOTHER		illegal argument etc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EZERODEV	zero devide
ENOMES	NO Message
EMAXINDEX	exceed max. number of subscript character
typedef short	HPK_Class; // Class number
typedef union HPKOBJ	HPK-objectId; //DW_OBJECT identifier

□ □ HPK_is Null

```
[Form] BOOL HPK_isNull (
    HPK_ObjectId object;
);
```

[Comment] It tests if object is valid.

```
[Example] if(HPK_isNull(result)==TRUE)
    ::MessageBox(NULL,"result
error","MEDIA",MB_OK|MB_SYSTEMMODAL);
```

□ □ HPK_add class

```
[Form]HPK_ObjectId HPK_addClass(
    NULL, reservation
    void*object, Internal identified value of DW_CLASS
    char*name, Huress-Book class name
    HPK_Class xclass Huress -Book class number
    FARPROC callBack Call back function at receiving
    DW_MESSAGE
);
```

[Comment] It registers DW_CLASS. Upon completion of registration,a wide area variable is created, and the DW_CLASS id stored in the 1t element <index 0>. Do specify normal null for the object.

```
[Example] HPK_addClass(NULL,NULL, "MYCLASS", 300, callBack);
```

□ □ HPK_rmClass

```
[Form] HPK_ObjectId HPK_rmClas(
    HPK_ObjectId classObject
);
```

[Comment] it deletes DW_CLASS specified by class Object.

```
[Example] HPK_rmClass(SMPCIAS);
```

□ □ HPK_addInstance

[Form] HPK_HPObjectId HPK_addInstance(
 HPK_ObjectId xclass,
 void*object
);

[Common] It creates HPK_OBJECT and register into Huress-Book .

[Example] HPK_addInstance (300,myInstance);

□ □ HPK_getObject Info

[Form] void HPK_getObjectInfo(
 HPK_ObjectId instance,
 HPK_ClassId*xclass,
 void**object
);

[Comment] It obtains information of instance.

[Examp!] HPK_getObjectInfo (myInstance,&myClassId,&myObj);

□ □ HPK_sendMessage

[Form] HPK_ObjectId HPK_sendMessage(
 HPK_objectId self, MESSAGE sends out DW_OBJECT
 HPK_objectId target, MESSAGE receivers DW_OBJECT
 char *message, MESSAGE name
 short count, number of arguments
 HPK_ObjectId obj[] a row argument DW_OBJECT
);

[Comment] It sends DW_MESSAGE from self to target.

Return value is DW_OBJECT from target.

This function is to send is to send out DW_MESSAGE of Huress from C++ language.

It is the function as Huress-Book sending out message from user

[Example] HPK_sendingMessage (mediaObj,ntioObj, "write", arg);

□ □ HPK_defgdef

[From] void HPK_degdef{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char*name
);
```

[Comment] It creates the variable specified by the variable name(name) and register it into the wide area scope. Once the variable is registered, it stays in Huress-Book until HPK_rmgdef() is called.

[Example] HPK_degdef("my Huress-Book").

□ □ HPK_rmgdef

```
[From] void HPK_rmgdef(
char*name
);
```

[Comment] It creates the variable specified by the variable name(name) and register it into the book scope. The variable stays until the Book is returned.

[Example] HPK_degdef("myBook").

□ □ HPK_findVObj

```
[From] HPK_ObjectId HPK_findVObj(
char*name
);
```

[Comment] It fetches the variable DW_PBJECT specified by the variable name (name). The variable DW_PBJECT that has an object name is retrieved in order of procedure, page, page,book and wide area scope. The first DW_PBJECT found id returned.

[Example] desk = HPK_VObject("DESK");

□ □ HPK_setObjectTo var

```
[form] void HPK_ObjectToVar(
HPK_ObjectId variable,
HPK_ObjectId Object,
short index
);
```

[Comment] It substitutes DW_OBJECT for the subscript (index)specified by variable
DW_OBJECT (variable).

[Example] HPK_setObjefToVar(variable,object,10);

□ □ HPK_gettObjectForm Var

```
[Form] HPK_Object HPK_gettObjectFormVar(
    HPK_ObjectId variable ,
    short index
);
```

[Comment] it fetches DW_OBJECT of subscript (index) specified by variable DW_OBJECT (variable).

{Notice} There is a maximum value to specify the subscript.

[Example] r = HPK_gettObjectFromVar(variable,10);

□ □ HPK_newBasicObject

```
[From] HPK_ObjectID HPK_newBasicObject(
    HPK_ClassID xclass    types of basic DW_OBJECT
```

[Comment] It creates the basic DW_OBJECT specified by xclass.

HPK_Integer	Integer DW_OBJECT
HPK_Float	Real Number DW_OBJECT
HPK_Character	Character DW_OBJECT
HPK_String	Text DW_OBJECT
HPK_Point	Placement DW_OBJECT
HPK_Rect	Rectangle DW_OBJECT
HPK_Boolean	Logic DW_OBJECT
HPK_Atom	Atom DW_OBJECT

[Example] r = HPK_newBasicObject(HPK_Integer);

□ □ HPK_getIValue

```
[From] long HPK_getIValue(
    HPK_ObjectId &object
);
```

[Comment] It fetches the value of Integer DW_OBJECT(object).

[Example] i = HPK_getIValue(object);

□ □ HPK_getFValue

```
[From] double HPK_getFValue(
    HPK_ObjectId &object
```

[Comment] It fetches the value of real DW_OBJECT(object).

[Example] f = HPK_getFValue(object);

□ □ HPK_getCValue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[From] short HPK_getCValue(
        HPK_ObjectId &object
```

[Comment] It fetches the value of character DW_OBJECT(object).

[Example] c = HPK_getCValue(object);

□ □ HPK_getSValue

```
[From] char HPK_getSValue(
        HPK_ObjectId &object
```

[Comment] It fetches the value of text DW_OBJECT(object).

{Notice} The return value is a pointer to text. In case the object is deleted, the entity of text dose not exist anymore once the object has been deleted. Normally make sure to copy the text into your own domain after executing this function.

[Example] s = HPK_getSValue(object);

□ □ HPK_getPValue

```
[From] void HPK_getPValue(
        HPK_ObjectId &object
        short *x,
        short *y
    );
```

[Comment] It fetches the value of placement DW_OBJECT(object).

[Example] HPK_getPValue(object,&x,&y);

□ □ HPK_getRValue

```
[From] void HPK_getRValue(
        HPK_ObjectId &object
        short *x1,
        short *y1,
        short.*x2,
        short *y2
    );
```

[Comment] It fetches the value of rectangle DW_OBJECT(object).

[Example] HPK_getRValue(object,&x1,&y1,&x2,&y2);

□ □ HPK_getBValue

```
[From] BOOL HPK_getBValue(
        HPK_ObjectId &object
        BOOL b
```

```
);
```

[Comment] It fetches the value of logic DW_OBJECT(object).

{Notice} BOOL is short.

[Example] r = HPK_getBValue(object);

□ □ HPK_getAValue

```
[From] void HPK_getAValue(
        HPK_ObjectId &object
        long *l,
        void **buffer
    );
```

[Comment] It fetches the value of atom DW_OBJECT(object).

The value obtained is memory domain address and its size.

[Example] HPK_getAValue(object,&len,&buffer);

□ □ HPK_setIValue

```
[From] void HPK_setIValue(
        HPK_ObjectId &object
        long l
    );
```

[Comment] It initializes DW_OBJECT(object) with the integer i.

[Example] HPK_setIValue(object,10L);

□ □ HPK_setFValue

```
[From] void HPK_setFValue(
        HPK_ObjectId &object
        double d
    );
```

[Comment] It initializes DW_OBJECT(object) with the real number d.

[Example] HPK_setFValue(object,10.5L);

□ □ HPK_setCValue

```
[From] void HPK_setCValue(
        HPK_ObjectId &object
        short c
    );
```

[Comment] It initializes DW_OBJECT(object) with the character c.

[Example] HPK_setCValue(object,'C');

□ □ HPK_setsValue

```
[From] void HPK_setSValue(
        HPK_ObjectId &object
        char *s
    );
```

[Comment] It initializes DW_OBJECT(object) with the string.

{Notice} It sets a copied text of s.

[Example] HPK_setSValue(object,"string");

□ □ HPK_setPValue

```
[From] void HPK_setPValue(
        HPK_ObjectId &object
        short x,
        short y
    );
```

[Comment] It initializes DW_OBJECT(object) with the placement (x,y).

[Example] HPK_setPValue(object,10,10);

□ □ HPK_setRValue

```
[From] void HPK_setRValue(
        HPK_ObjectId &object
        short x1,
        short y1,
        short x2,
        short y2
    );
```

[Comment] It initializes DW_OBJECT(object) with the rectangle coordinates (x1,y1,x2,y2).

[Example] HPK_setRValue(object,10,10,200,200);

□ □ HPK_setBValue

```
[From] void HPK_setBValue(
        HPK_ObjectId &object
        BOOL b
    );
```

[Comment] It initializes DW_OBJECT(object) with the boolean b.

{Notice} BOOL is short.

[Example] HPK_setBValue(object,TRUE);

HPK_setAValue

```
[Form] void HPK_setAValue(
        HPK_ObjectId &object
        long l
    );
```

[Comment] It initializes DW_OBJECT(object) with the long l.

As the results,the data domain of this value is secured.

[Example] HPK_setAValue(object,1000);

HPK_addPalett

```
[Form] void HPK_addPalette(
        HWND hWnd,           windows handle of media view
        HPALETTE hPalette,   Palette of media instance
        int kind             DW_OBJECT partition
    );
```

[Comment] It registers the media view that changes the palette.If there is no palette in the media instance,the palette can be initialized with NULL.

[Example] HPK_addPalette(HWindow,NULL,HPK_MEDIAPAL);

HPK_detachPalette

```
[Form] void HPK_detachPalette(
        HWND hWnd,           windows handle of media view
    );
```

[Comment] It deletes the registration of media view that changes the palette.

The registered palette is not deleted by this function.

[Example] HPK_detachPalette(HWindow);

HPK_realizePalette

```
[Form] BOOL HPK_realizePalette(
        HWND hWnd,           windows handle of media view
        HDC hDc              devic context of media view
    );
```

[Comment] It changes the palette. To use this function, it is necessary to register previously the media view and palette in the HPK_addPalette. The return value is true when changing the entry of palette and false otherwise.

[Example] HPK_realizePalette(HWindow,hDC);

HPK_isRelizeBkgnd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[Form] BOOL HPK_isRealizeBkgnd(
    HWND hWnd windows handle of media view
);
```

[Comment] To change the background when changing the palette. This function is used when changing the palette without using `HPK_realizePalette`. To use this function, it is necessary to register previously the media view by `HPK_addPalette`. The return value is true when the palette must be changed as background, false otherwise.

```
[Example] HPK_isRealizeBkgnd(HWindow);
```

`HPK_error`

```
[Form] short HPK_error();
```

[Comment] It obtains the execution condition of the other HPK functions. It can be used after executing the other HPK functions.

0	Normal
ENOCLASS	Unidentified class
ECLASSID	Illegal class ID
EDUPCLASS	Duplicated class
EINSTANCE	Illegal instance
EMESSID	Illegal message ID
EDUPVALI	Duplicated variable
ENOVALI	Undefined variable
EVALI	Illegal variable
ENOMEM	Lack of memory
EOTHER	Illegal argument
EZERODEV	Zero devide
ENOMES	No Message
EMAXINDEX	Exceed MAX.number of Subscripts

```
[Example] r = HPK_setError();
```

```
if(r != 0)
```

```
    ::MessageBox(NULL, "HPKERROR", "HURESS-Book", MB_OK | MB_SYSTEMMODAL);
```

6. Bookshelf DW_OBJECT Reference

The Bookshelf `DW_OBJECT` provides media program with variable length or fixed length file ad media file for each instance. It can hole the media information in this file. The structure of the file can be

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

constructed freely but with information common with SDK. The Bookshelf DW_OBJECT provides various services in regard of indexing. In this section, users as follows. The ntioObj found in the [Form] is the bookshelf DW_OBJECT. This is mentioned by the desk DW_OBJECT when creating media instance.

openItemId

[Form] state = ntioObj openItemId arg;

- arg[0] page Identifier (Integer DW_OBJECT)
- arg[1] media Identifier No (Integer DW_OBJECT)
- arg[2] media Identifier (Integer DW_OBJECT)
- state Execution state (Logic DW_OBJECT)

[Comment] It opens the media file.

[Example] send2Com(mediaObj,ntioObj, "openItemId", 3,arg):

newItem

[Form] state = ntioObj newItemId arg;

- arg[0] page Identifier (Integer DW_OBJECT)
- arg[1] media attribute No (Integer DW_OBJECT)
- arg[2] media name (Text DW_OBJECT)
- arg[3] media Identifier No (Integer DW_OBJECT)
- arg[4] media Identifier (Integer DW_OBJECT)
- state Execution state (Logic DW_OBJECT)

[Comment] It create the media file.

[Example] send2Com(mediaObj,ntioObj, "newItemId", 5,arg):

seek

[Form] state = ntioObj seek arg;

- arg[0] 0 (Integer DW_OBJECT)
- arg[1] media Identifier (Integer DW_OBJECT)
- arg[2] seek Identifier (Integer DW_OBJECT)
- state Execution state (Logic DW_OBJECT)

[Comment] It initializes the placement of sataa to read or write.

[Example] send2Com(mediaObj,ntioObj, "seek", 3,arg):

read

[Form] state = ntioObj read arg;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

arg[0] 0 (Integer DW_OBJECT)
 arg[1] media Identifier (Integer DW_OBJECT)
 arg[2] Atom Identifier (Atom DW_OBJECT)
 arg[3] Data length to read out (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It reads out data.

[Example] send2Com(mediaObj,ntioObj, "read", 4,arg):

write

[Form] state = ntioObj write arg;

arg[0] 0 (Integer DW_OBJECT)
 arg[1] media Identifier (Integer DW_OBJECT)
 arg[2] Atom for read out (Atom DW_OBJECT)
 arg[3] Data length to read out (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It write data.

[Example] send2Com(mediaObj,ntioObj, "read", 4,arg):

removeItem

[Form] state = ntioObj removeItemId arg;

arg[0] media Identifier (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It deletes the media file.

[Example] send2Com(mediaObj,ntioObj, "removeItem", 1,arg):

closeItem

[Form] state = ntioObj closetemId arg;

arg[0] media Identifier (Integer DW_OBJECT)
 state Execution state (Logic DW_OBJECT)

[Comment] It closes the media file.

[Example] send2Com(mediaObj,ntioObj, "closeItem", 1,arg):

newIndex

[Form] state = newIndex arg;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

arg[0] Book Identifier (Integer DW_OBJECT)
 arg[1] Index name (Text DW_OBJECT)
 arg[2] Key Text (Text DW_OBJECT)
 arg[3] Page Identifier No (Integer DW_OBJECT)
 arg[4] Media Identifier (Integer DW_OBJECT)
state Execution state (Logic DW_OBJECT)

[Comment] It create the media file.

[Example] send2Com(mesiaObj,ntioObj, "newIndex", 5,arg);

removeIndex

[Form] state = removeIndex arg;

arg[0] Book Identifier (Integer DW_OBJECT)
 arg[1] Index name (Text DW_OBJECT)
 arg[2] Key Text (Text DW_OBJECT)
 arg[3] Media Identifier (Integer DW_OBJECT)
state Execution state (Logic DW_OBJECT)

[Comment] It deletes the index.

[Example] send2Com(mesiaObj,ntioObj, "removeIndex", 4,arg);

renameItem

[Form] state = renameItem arg;

arg[0] Media Identifier (Integer DW_OBJECT)
 arg[1] New media name (Text DW_OBJECT)
state Execution state (Logic DW_OBJECT)

[Comment] It changes the media name.

[Example] send2Com(mesiaObj,ntioObj, "renameItem", 2,arg);

get Error

[Form] state = getError arg;

arg[0] Error Information1 (Integer DW_OBJECT)
 arg[1] Error Information1 (Integer DW_OBJECT)
state Execution state (Logic DW_OBJECT)

[Comment] It gets the error information of bookshelf.

[Example] send2Com(mesiaObj,ntioObj, "getError", 2,arg);

7. Desk DW_OBJECT Reference

There are many function to send an and receive DW_MESSAGE in the Desk DW_OBJECT. This section describes the typical message receiving function that become necessary for making the media program. Also this section describes how to send messages to the media class and the indicated in [Form] in this reference shows the Desk DW_OBJECT.

receiving of DW_MESSAGE

setItem

[Form] state = Desk setItem arg;

arg[0] Media program name DDL name except ".DLL"
(Text DW_OBJECT)

arg[1] Huress -Book class name of media (Text DW_OBJECT)

arg[2] Media class

arg[3] HBITMAP (Integer DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] It notifies the add-in media information to Huress-Book.

[Example] arg[0] = b20("samp"); //DDL name Except "DLL"
arg[1] = b20("samp");
arg[2] = HPK_addInstance(mediaId,NULL);
arg[3] = b20((long) paletteBmp);
send2Com(mediaId,deskId "setItem", 4,arg);

startDialog

[Form] state = Desk startDialog arg;

state Execution state (Logic DW_OBJECT)

[Comment] It notifies the use of modal diagram to Huress-Book.

[Example] send2Com(mediaId,deskId "startDialog", -1,arg);

endDialog

[Form] state = Desk endDialog arg;

state Execution state (Logic DW_OBJECT)

[Comment] It notifies the use of modal dialog box to Huress-Book.

[Example] send2Com(mediaId,deskId "endDialog", -1,arg);

sent MESSAGE to the media class

new

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Form] state = media Class new arg;

- arg[0] Book shelf (Bookshelf DW_OBJECT)
- arg[1] HWND of book(page)in which media view is placed.
(Integer DW_OBJECT)
- arg[2] Book Identifier (Integer DW_OBJECT)
- arg[3] Page Identifier (Integer DW_OBJECT)
- arg[4] Page (Page DW_OBJECT)
- arg[5] -
- arg[6] Media (Media DW_OBJECT)
- arg[7] Media name (Media DW_OBJECT)
- arg[8] Media rectangular (Media DW_OBJECT)
- state Execution state (Logic DW_OBJECT)

[Comment] create the new media instance.

load

[Form] state = media Class new arg;

- arg[0] Book shelf (Bookshelf DW_OBJECT)
- arg[1] HWND of book(page)in which media view is placed.
(Integer DW_OBJECT)
- arg[2] Book Identifier (Integer DW_OBJECT)
- arg[3] Page Identifier (Integer DW_OBJECT)
- arg[4] Page (Page DW_OBJECT)
- arg[5] Media Identifier No. (Integer DW_OBJECT)
- arg[6] Media (Media DW_OBJECT)
- arg[7] Media name (Text DW_OBJECT)
- arg[8] Rectangular of Media (Rectangular DW_OBJECT)
- state Execution state (Logic DW_OBJECT)

[Comment] Create media instance by loading the media information from media file.

paste

[Form] state = media Class new arg;

- arg[0] Book shelf (Bookshelf DW_OBJECT)
- arg[1] HWND of book(page)in which media view is placed.
(Integer DW_OBJECT)
- arg[2] Book Identifier (Integer DW_OBJECT)
- arg[3] Page Identifier (Integer DW_OBJECT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

arg[4] Page (Page DW_OBJECT)

arg[5] -

arg[6] Media (Media DW_OBJECT)

arg[7] Media name (Text DW_OBJECT)

arg[8] Rectangular of Media (Rectangular DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] Create media instance by loading the media information from the clipboard.

systemMessage

[Form] state = media Class systemMessage arg;

arg[0] WM_XXX prefix message no. of Windows (Integer DW_OBJECT)

arg[1] WPARAM for prefix message of Windows (Integer DW_OBJECT)

arg[2] LPARAM for prefix message of Windows (Integer DW_OBJECT)

stage Execution stage (Login DW_OBJECT)

[Comment] Process WM_XXX prefix message sending to an application from Windows.

[Remark] At present there are two WM_XXX prefix message that are sent by this send.

MESSAGE as flows:

WM_PALLETTECHANGED

WM_PALETTEISCHANGING

getError

[From] state = media Class getError arg;

arg[0] Information 1 (Integer DW_OBJECT)

arg[1] Information 2 (Integer DW_OBJECT)

arg[2] Information 3 (Integer DW_OBJECT)

stage Execution stage (Login DW_OBJECT)

[Comment] Return error information of media class.

Send out message to media instance

save

[Form] state = media Instance save arg;

arg[0] Media Identifier No. (Integer DW_OBJECT)

state Execution state (Logic DW_OBJECT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Comment] Store the media information.

Remove

[Form] state = media Instance remove;

state Execution state (Logic DW_OBJECT)

[Comment] Delete the media information.

Copy

[Form] state = mediaInstance copy;

state Execution state (Logic DW_OBJECT)

[Comment] Copy the media information into the clipborad.

Point

[Form] state = mediaInstance point arg;

arg[0] Destination point coordinates (Placement DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] Move the media instance.

Size

[Form] state = mediaInstance size arg;

arg[0] Width (Integer DW_OBJECT)

arg[1] Height (Integer DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[comment] Change the size of media instance.

Draw

[Form] state = mediaInstance draw arg;

arg[0] Rectangle of draw domain(Rectangle DW_OBJECT)

arg[1] Hight (Integer DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] Draw the media by canceling a temporary delete.

Invalidate

[Form] state = mediaInstance invalidate;

state Execution state (Logic DW_OBJECT)

[Comment] Delete temporary the media.

Show

[Form] state = mediaInstance show;

state Execution state (Logic DW_OBJECT)

[Comment] Display the media.

Hide

[Form] state = mediaInstance hide;

state Execution state (Logic DW_OBJECT)

[Comment] Hide the media.

Info

[Form] state = mediaInstance info arg;

arg[0] -

arg[1] Need to change (Logic DW_OBJECT)

arg[2] Name after renewal (Text DW_OBJECT)

arg[3] Rectangle after renewal (Rectangle DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] Renew the media information

setFlag

[Form] state = mediaInstance setFlag arg;

arg[0] Flag (Integer DW_OBJECT)

0x00000001 Execution mode

0x00000002 Index permission mode

state Execution state (Index permission mode)

[Comment] Set Execution mode and Index mode

the 0th bit on edit
off peruse

the 1st bit on Index creation permission
off Index creation inhibit

getError

[Form] state = mediaInstance getError arg;

arg[0] Information1 (Integer DW_OBJECT)

arg[1] Information2 (Integer DW_OBJECT)

arg[2] Information3 (Integer DW_OBJECT)

state Execution state (Logic DW_OBJECT)

[Comment] Return error information of media class.

asString

[Form] state = mediaInstance asString;

state Media name (Text DW_OBJECT)

[Comment] Return name of media instance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

□ □ Move

[Form] state = mediaInstance move arg;

arg[0] Coordinates of media (Rectangle DW_OBJECT)
(Rectangle)

state Execution state (Logic DW_OBJECT)

[Comment] Change the coordinates of the media instance

□ □ setFocus

[Form] state = mediaInstance setFocus;

state Executionstate (Logic DW_OBJECT)

[Comment] Set focus to the media instance.

□ □ getPoint

[Form] state = mediaInstance getPoint;

state coordinates (Placement) (Placement DW_OBJECT)

[Comment] Return the coordinates of media instance.

□ □ getSize

[Form] state = mediaInstance getSize;

state size (Placement DW_OBJECT)

[Comment] Return the size of media instance.

□ □ getRect

[Form] state = mediaInstance getRect;

state Coordinate (Rectangle) (Rectangular DW_OBJECT)

[Comment] Return the coordinates of media instance.

8. MEDIA-SDK Class Library Reference

The MEDIA-SDK class library provide the developers with the common functions required to make the media program by making the class library in C++ language (Borland C++).

The class library consists of the following classes and are used to define the sub-classes.

class for common processing for media class

class for common processing for media instance

class for index registration

class for procedure defining for event and page

class for dispatching DW_MESSAGE to response function specified for each

DW_MESSAGE.

Since the source program is available to the developers, try to refer to the function.

8.1 SDK Macro

There are several macros defined for message response function.

```
DECLARE_METHOD_TABLE(TMediaClass);
DECLARE_METHOD_TABLE(cls);
DEFINE_METHOD(cls)\
DEFINE_METHODS1(cls,base)\
METHODDENTRY(message,method)\
```

8.2 SDK Constant

The following media creation codes are defined.

```
PROCNEW      0 It creates or is created the new media.
PROCLOAD    1 It creates or is created the media from the media file.
PROCPASTE   2 It creates or is created the media from the clipboard.
```

8.3 TMessage Info

this is the class structure that contains the information regarding DW_MESSAGE generated from Huress-Book. The object is created in TMediaClass::execMessage and transfers to the message response function.

from (public)

[form] HPK_Object from;

[Comment] It is the DW_OBJECT as the source of DW_MESSAGE.

self (public)

[Form] HPK_Object self;

[Comment] It is the DW_OBJECT as the destination of DW_MESSAGE.

method (public)

[form] char*method;

[Comment] It is the DW_OBJECT as the destination of DW_MESSAGE.

argCount (public)

[Form] short argCount;

[Comment] This is the number of arguments to be attached to the DW_MESSAGE.

arg (public)

[Form] HPK_Object *arg;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Comment] This is a row of arguments to be attached to the DW_MESSAGE.

The entity of argument is in the memory space in the source DW_MESSAGE.

result (public)

[Form] HPK_ObjectId result;

[Comment] This DW_OBJECT is reinitialized with false.

dispatch Object (public)

[From] TMethodHandler *dispatch Object;

[Comment] The object pointer that has message response function is stored.

That is Find(TMessageInfo&info) function of TMethodHandler.

Entry (public)

[Form] TGenericMethodTableEntry *Entry;

[Comment] This is a table that converts the DW_SELECTOR to the Message response function.

Constructor (public)

[From] TMessageInfo(
 HPK_ObjectId f,
 HPK_ObjectId s,
 char *m, short c,
 HPK_ObjectId *a
);

[Comment] It is used in the TmediaClass::execMessage.

TestVariable (public)

[Form] BOOL testVariable(
 HPK_ClassId *xclass,

short count Number of arguments
);

[Comment] It test arguments to be attached the DW_MESSAGE.

The arguments are defined as variables DW_OBJECT.

The xclass is a row of class number of DW_OBJECT. It skips the test if-1 is included in the row element.

TestBasicObject (public)

[Form] BOOL testBasicObject(
 HPK_ClassId *xclass,

```

short count          number of arguments
);

```

[Comment] It tests argument to be attached to the DW_MESSAGE.

The arguments are defined as basic DW_OBJECT.

The xclass is a row of class numbers of W_OBJECT. It skips the test if-1 is included in the row element.

Is VariableType (public)

[Form] BOOL is VariableType();

[Comment] It tests whether or the argument to be attached are the DW_OBJECT.

8.4 TMethodTapEntry

```

template <class T> class_SDKCLASS TMethodTableEntry {
public:
    typedef void (T::*XPMF)(TMessageInfo&);
    char *message;
    XPMF Pmf;

```

8.5 TMethodHandler

This is the class to manage the response function of Hures-Book. It has a function to change from the text to the message response function

Distracter (public)

[Form] virtual TMethodHandler(){}

[Comment] This is a virtual function.

Find (public)

[Form] virtual BOOL find (TMessageInfo&info) = 0;

[Comment] This is a genuine virtual function. It tests whether DW_MESSAGE is receivable

or not

dispatch (public)

[Form] virtual void dispatch (TMessageInfo&mi);

[Comment] It dispatches to the message response function.

SearchEntries (protected)

[Form] BOOL searchEntries(TGenericMethodTableEntry *entries,
TMessageInfo&info);

[Comment] It test whether DW_MESSAGE is receivable or not.

8.6 TMediaClass

This class is to manage the media. The object is made with one for each media. Users can define the sub-class for each of media by using this class as the parent class. The member functions to override for the sub-class are the constructor and makeInstance function. It is not necessary to modify the other function and member variables.

<<Data Member>>

Desk (protected)

[Form] HPK_ObjectId deskId;

[Comment] It stores the desk the DW_OBJECT.

MediaId (protected)

[Form] HPK_ObjectId mediaId;

[Comment] It stores the DW_OBJECT of this media class.

<<Member Function>>

Consturctor (public)

```
[Form] TMediaClass(
    char *progName, //Name of media program except .DLL
    char *className, //Huress-Book class name
    HPK_ClassId classId //Huress-Book class Number
    HBITMAP paletteBmp //bit map
);
```

[Comment] It creates the media class by Huress-Book class name className and Huress-Book class number classId. It also sets up bit map (22*20) to be used for media palette. The deletion of bit map is done by Huress-Book

Destructor (public)

[Form] ~TMediaClass();

[Comment] It delete the media class

makeInstance (public)

```
[Form] virtual TMediaUnstance *makeInstance(
    short mode,
    HPK_ObjectId classId,
```

```
HPK_ObjectId *arg
```

```
);
```

[Comment] It creates the media Instance(TMEDIAInstance) calling out form DMNew(), DMLoad() and DMPaste(). This function is a virtual function.

```
□ □ DMNew (public)
```

```
[Form] void DMNew (
```

```
    TMessageInfo& mi
```

```
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It creates the new media instance.

```
□ □ DWLoad (public)
```

```
[Form] void DMLoad (
```

```
    TMessageInfo& mi
```

```
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It creates the media information of media instance by loading from the media file.

```
□ □ DMPaste (public)
```

```
[Form] void DMPaste(
```

```
    TMessageInfo&mi
```

```
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It creates the media instance from the clipboard.

```
□ □ DMSystemMeassage (public)
```

```
[Form] Void DMSystemMessage(
```

```
    TMessageInfo& mi
```

```
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It processes Window's WM_XXX prefix messages.

```
□ □ DmGetError (public)
```

```
[Form] void DMGetError(
```

```
    TmessageInfo& mi
```

```
);
```

[Comment] This is the response function of DW_MESSAGE from the Desk DW_OBJECT.

It return the error information of media class.

ExecMessage (public)

```
[Form] HPK_ObjectId execMessage(
    HPK_ObjectId from,
    HPK_ObjectId self,
    char *message,
    short count,
    Hpk_ObjectId obj[]
);
```

[Comment] It dispatches the message from DW_SELECTOR of the Receive DW_MWSSAGE to the response function.

GetClassId (public)

```
[form] HPK_ClassId getClass();
```

[Comment] It returns class number of the media DW_OBJECT.

==<(public)

```
[Form] int operator == (const TMediaClass& target) const;
```

```
int operator< (const Tmedia class& target) const;
```

[Comment] It returns the class number of the media DW_OBJECT.

IsValidInstance (protected)

```
[Form] virtual BOOL is ValidInstance (void *instance);
```

[Comment] It tests whether the media instance is valid or not.

8.7 Tmedia Index

This is the class to store the index information of media.

<<Data Member>>

keyName (private)

```
[Form] char keyName[64];
```

[Comment] This is a text for the Index key. Categorization takes place by this text.

IndexName (private)

```
[Form] char indexName [64];
```

[Comment] Index name

saveFlag (private)

```
[Form] BOOL saveFlag;
```

[Comment] This is a flag to indicate whether or not index is saved. Normally, this flag is changed automatically by the other classes.

<<Member Function>>

constructor (public)

```
[Form] TMediaIndex(
    char *key,
    char *index
);
```

[Comment] It creates the index object from the index key name and the index name.

getKey (public)

```
[Form] char *getKey();
```

[Comment] It returns the index key name.

getIndex (public)

```
[Form] char *getIndex();
```

[Comment] It returns the index name.

setSave (public)

```
[Form] void setSave();
```

[Comment] It notifies the necessity of saving the index.

IsSave (public)

```
[Form] BOOL issave();
```

[Comment] It returns TRUE when it is necessary to save the index, FALSE the otherwise.

operator == (<public)

```
[Form] int operator == (const TMediaIndex& record) const;
```

```
int operator <(const TMediaIndex& record) const;
```

[Comment] It conducts the comparison arithmetic by the address value of the object.

8.8 TMediaEvent

This is a class to hold the proceder name of the page corresponded to the event and the event name.

This is a sub-class of the string class that holds the events name.

<<Data Member>>

scriptName (private)

```
[Form] char scriptName[32];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Comment] This is an array to store the procedure name to be executed.

<<Member Function>>

Consturctor (public)

```
[Form] TMediaEvent(
    char *event,
    char *script = 0
);
```

[Comment] It creates the object that stores the event name corpeponded to the procedure name.

getScriptName (public)

```
[Form] char *getScriptName();
```

[Comment] It returns the procedure name.

setScriptName (public)

```
[Form] void setscriptName(
    char* str
);
```

[Comment] It sets up the procedure name.

setscriptName (public)

```
[Form] void setScriptName(
    string&str
);
```

[Comment] It sets up the procedure name.

8.9 TMediaInstance

This is the class to create the media instance. The object is created for each media instance. Users can define sub-classes for each for media by using this class as the parent class. Indefining the sub-classes, the constructor, load function, save function and information function are overridden. Users can override the other member function and data member as needed.

<<Data Member>>

header (protected)

```
[Form] MediaHeader header;
```

[Comment] It is the common information of media DW_OBJECT.

```
Struct MediaHeader {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

long	filler1;	reserved)
short	version	Version of media
char	filler2[7];	(reserved)
char	name[32];	Media name
BOOL	showFlag;	Initial display on/off
short	x;	X coordinate
short	y;	Y coordinate
short	w;	Width
short	h;	Height
char	filler3[31];	(reserved)
short	indexCount	Index number

DWORDDDlength; Fixed length of the media

}

This information is set with the default value in the class.

The x, y, w h are reinitialized in this class but can be reinitialized in the sub-classes.

Make sure to reflect this change to the media view. The indexCount is managed in this class

indexArray (protected)

[Form] TIndexArray *indexArray;

[Comment] This is array of the index information. Users can only do reference.

To add an index information into the array, use the addIndex function.

To delete an index information in the array, use the delIndex function.

EventArray (protected)

[Form] TEvenArray *evenArray;

[Comment] This is an array to store the procedure name at the occurrence of event.

Users can only do reference.

To add a procedure name into the array, use the setScriptName function.

To obtain a procedure name, use the getscriptName function.

ExeMode (protected)

[Form] BOOL exeMode;

[Comment] It shows the peruse mode if this value is TRUE

It shows the edit mode if it is FALSE.

Users can only do reference.

IndexMode (protected)

[Form] BOOL indenMode;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Comment] It shows the permission to create the index if the value is TRUE.

It the value is FALSE, it is not allowed to create the index.

BookWin (protected)

[Form] Twindow *bookWin;

[Comment] This is a window of OWL to create the in which the media is stored.

PageObj (protected)

[Form] HPK_Object pageObj;

[Comment] This is DW_OBJECT of page in which the media is stored.

NtioObj (protected)

[Form] HPK_ObjectId ntioObj;

[Comment] This is DW_OBJECT of the bookshelf in which the media is stored

mediaObj(protected)

[Form] HPK_BbjectId mediaObj;

[Comment] This is the DW_OBJECT (self Object) of this media.

bookshelf (protected)

[Form] HPK_Object bookID;

[Comment] This is the book identifier of the book in which the media is stored.

pageId (protected)

[Form] HPK_Object pageId;

[Comment] This is the page identifier of the page in the which the media is stored.

pageNo (protected)

[Form] HPK_ObjectId pageNo;

[Comment] This is the identifier number of the page in which the media is stored.

mediaId (protected)

[Form] HPK_ObjectTd mediaId;

[Comment] This is the media identifier of the mrdia.

MediaNo (protected)

[Form] HPK_ObjectId mediaNo;

[Comment] This is the media number of the media.

ClassId (protected)

[Form] HPK_ObjectId classId;

[Comment] This is the DW_OBJECT of the media class.

MediaWin (protected)

[Form] Twindow *mediaWin;

[Comment] This is the OWL window of the media. The value is set by the users.

However, it is not necessary to set up the value when the media does not use OWL window. In this case, users can draw the image of the media directly to the bookWin.

Invalid (protected)

[Form] BOOL invalid;

[Comment] Ti shows the media to be invalid.

<<Member function>>

seek (protected)

[Form] BOOL seek(

long point

);

[Comment] It renews the write pointer by reading the media information.

Read (protected)

[Form] long read(

void *buffer,

long size

);

[Comment] It reads the data of the media information.

Write (protected)

[Form] long write(

void *buffer;

long size

);

[Comment] It writes the media information data into the media file.

addIndex (protected)

[Form] void addIndex(

TMediaIndex *mediaIndex

);

[Comment] It registers the index.

delIndex(protected)

[Form] delIndex(

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TmediaIndex *mediaIndex

);

[comment] It delete the index.

get Error (protected)

[Form] void gteError();

[Comment] It obtains the error information of the bookshelf.

RunMethod(protected)

[Form] virtual BOOL runMethod(

long status = 0; This is the status of media itself;

);

[Comment] It demand the execution of the procedure to the page.

The status is the status of the media it self

It sets up the following DW_OBJECT with argc and argv for the procedure read out.

argc

2(number of arguments)

argv[0]

Sourec media DW_OBJECT

argv[1]

Text DW_OBJECT of destination media name.

load (protected)

[Form] virtual BOOL Load();

[Comment] It loads the media information from the media file.

It is necessary to cverride these member function in the sub-classes.

In this case make sure to call initially TMediaInstance::Load().

save (protected)

[Form] virtual BOOL save();

[Comment] It saves the media information to the media file.

information (protected)

[Form] virtual BOOL information();

[Comment] It renews the information of the media.

It returns TURE when it is renewed and FALSE when it is abandoned.

constructor (public)

[Form] virtual ~TMediaInstance();

[Comment] It creates the media instance.

destructor (public)

);

[Comment] This is the response function of DW_MESSAGE from the desk.

It renews the media information.

DMSetFlag(public)

[Form] virtual void DMSetFlag (
 TMessageInfo& mi
);

[Comment] This is the response function of DW_MESSAGE from the desk.

It sets up the execution mode and index permit mode.

DMSetFocus(public)

[Form] virtual void DMSetFocus (
 TMessageInfo& mi
);

[Comment] This is the response function of DW_MESSAGE from the desk.

It sets the focus to the media.

DMGetError(public)

[Form] virtual void DMGetError (
 TMessageInfo& mi
);

[Comment] This is the response function of DW_MESSAGE from the desk.

It returns the media error information.

DMDispose(public)

[Form] virtual void DMDispose (
 TMessageInfo& mi
);

[Comment] This is the response function of DW_MESSAGE from the desk.

It deletes the media instance.

8. 10. Static Data

This is the static data in the MEDIA-SDK.

HPK_ObjectId	nullObject = {NULL};	undefined DW_OBJECT
HPK_ObjectId	trueObject;	TRUE DW_OBJECT
HPK_ObjectId	falseObject;	FALSE DW_OBJECT
TClassArray	*classArray;	management row of media class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BOOL	rmMediaFlag = FALSE;	felete flag of media instance
long	error1;	error infomation 1
long	error2;	error infomation 2
long	error3;	error infomation 3

8. 11. SDK function Library

Huress-Book provides users with the following function that make easier to use the Huress-Book interface library for creating and setting up DW_OBJECT.

b20 (No.1)

[Form] HPK_ObjectId b20(long i);

[Comment] It creates the integer DW_Object that has integer i.

b20 (No.2)

[Form] HPK_ObjectId b20(double d);

[Comment] It creates the real number DW_Object that has floating point d.

b20 (No.3)

[Form] HPK_ObjectId b20(short c);

[Comment] It creates the character DW_Object that has character c.

b20 (No.4)

[Form] HPK_ObjectId b20(char *s);

[Comment] It creates the text DW_Object that has text s.

b20 (No.5)

[Form] HPK_ObjectId b20(short x,short y);

[Comment] It creates the Placement DW_Object that has placement (x,y).

b20 (No.6)

[Form] HPK_ObjectId b20(POINT p);

[Comment] It creates the Placement DW_Object that has placement p.

b20 (No.7)

[Form] HPK_ObjectId b20(short x1,short y1,short x2,short y2);

[Comment] It creates the rectangle DW_Object that has rectangle (x1,y1,x2,y2).

b20 (No.8)

[Form] HPK_ObjectId b20(RECT r);

[Comment] It creates the rectangle DW_Object that has rectangle r.

b20 (No.9)

[Form] HPK_ObjectId b20(BOOL b);

[Comment] It creates the logic DW_Object that has the logic value b.

b20 (No.10)

[Form] HPK_ObjectId b20(void *buf,long l);

[Comment] It creates the atom DW_Object of length l and buffer buf.

02Rect

[Form] RECT 02Rect(HPK_ObjectId object);

[Comment] It reads the value from the object of the rectangle DW_OBJECT.

02Point

[Form] POINT 02Point(HPK_ObjectId object);

[Comment] It reads the value from the object of the placement DW_OBJECT.

send2Com

[From] BOOL send2Com(

 HPK_ObjectId self, SelfDW_OBJECT

 HPK_ObjectId target, Opponent DW_OBJECT

 char *msg, DW_SELECTOR

 short count, number of arguments

 HPK_ObjectId args[], an array of arguments

 BOOL f = TRUE message flag

[Comment] DW_MESSAGE is sent out by assigning variables and substituting the arguments for the variable. The functions are called when f is TRUE. If any error is detected, the error messages are displayed in the system modal dialog box.

 Make sure that the number of elements stored in "count" are in the array.

 These functions handle the transfer arguments to variable. To send out DW_MESSAGE by adding arguments directly, use the HPK_SindMessage() function of the Huress-Book Interface library.

9. How to create Media Using Sample Programs

There are three types of media program on the MEDIA-SDK as follows.

Basic Sample media (Media class name:SMP)

Type A Media sample (Media class name:SMP_A)

Type B Media sample (Media class name:SMP_B)

This section given an overview of the sample program and some advice to follow when creating. Readthrough this section when using the sample programs.

It is possible to compile the sample of MEDIA_SDK by boland C++ ver.4.52 and execute them by the following procedure.

Create the folder for the media development as described in [2.MEDIA_SDK installation] and copy the necessary files.

Load the Borland C++ ver.4.52 and open SMP.IDE of \SMP folder.

Execute compile link. three sets of DLL's become available. (SMP.DLL,SMP_A.DLL,SMP_B.DLL)

Hurress-book copies these DLL's to the folder \EXECYMEDIA.

Load Huress-Book and borrow the book in edit mode.

Set up media program by book information set up.

9.1. Basic Sample Media

this the media program to make to the media view borrowing entirely from the button class (TButton) of Boland C++ ver.4.52 ObjectWindows2.5. This is a sample program to explain the basic creation of media program, therefore it is not necessarily true to cover the perfect functions. Refer to the other sample for creating the actual media program.

Function Overview

Look like button

Does not have the button functions

Information dialog box and Index dialog box are only forms.

Main class

Tmedia class sub class

TSMPClass

TMedia Instance sub class

TSMPIInstance

Media view

Tbutton

Information Dialog

TInfordlg

Index Dialog

TIndexDlg

File Construction

Source program

SMP.CPP

Header file

SMP.H

Resource source

SMP.RC

Definition file

SMP.DEF

The matter to be referenced

Refer to the [4. Creating method of source program] for more details.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In this example, the location of paste is shifted below to the right.(The location is controlled by the band variable paste Count)

6 . Avoiding the influence of the background transparent media

There are case when the background disturb the display of type A media if users call the setWindowsPos function of Windows API for Type A media display.

7 . Get the page color

If it is necessary to have a color on the page when transparent media, get the color from the bookWin (protect member of TMediaInstance that has a window for the page).

```
(example)COLORREFbookColor=(COLORREF)bookWin>GetWindowLong(0);
```

8. Utilization of notice message to parent windows

As the event to the media, there is a case when users want to users the notice message from control to the parent windows. an example is the case of button control, where users want to pick up the notice of BN_CLICKED. In this case, the media view are created as follows:

- Create the media view as a sub-class of TFrameWindows.
- Paste the control that is to be used as client of the media view.

9.3. Type B Media Sample

The "Type B media" is the media that does not have its own windows but draws on the windows of the book. It does not accept the event such as mouse click but it uses smoothly superimpose function for multiple media. Refer to section 2 for type A media and type B media.

Function Overview

- Draw the picture in some random from in the random color.
- Background of the media is always transparent.
- There is an individual media information for own media (Brightness of RGB and number of apex).
- It treats DB_MESSAGE of show and hide
- It can set up the index
- It is possible to copy and paste the media.

Main class

TMedia Class sub class

TSMPClass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMedia Instance sub class	TSMPInstrance
Media view	none
Information Dialog	TInfodlg
Index Dialog	TIndexDlg

File Construction

Source program	SMP_B.CPP
Header file	SMP_B.H
Resource source	SMP_B.RC
Definition file	SMP_B.DEF

The matters to be referenced

The creation of type B media is basically the same as type A. Here are the description of major differences from type A media sample.

1 . Treatment of media Win

Since type B media does not create its own Windows, make sure to set NULL into media Win When creating media.

(refer to TSMPInstance::create)

2. Display of media

The display of media is done by draw message (DW_MESSAGEAGE).

It is show by rectangle and HDC that are transferred by arguments of draw message.

(refer to TSMPInstance::DMDraw)

3 . Response to show and hide

It is not necessary to describe the most of time for type A media since MEDIA-SDK processes show and hide. However, it need to be overridden for type B media. Asdescribed in the example, the re-write of type B media is always done by draw message. This ir true when changing the display contents by own DW_MESSAGE at execution.

Override DMshow and DMHide of TMediaInstance

Set the show Flag of header in DMShow and DMHide, and make the windows portion of to be invalid. (make a state that the rewrite is necessary.)

(refer to TSMPInstant::DMShow, TSMPInstance::DMHine)

4 . Superimpose of media

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

To display correctly the superimposed media, the display and deletion of the media should be managed by Huress-Book. When redrawing become necessary for the media , the user should write the program so that rewrite do not take place but instead informs necessity of redrawing the book Windows.

(refer to TSMPIInstance::information. It redraws since the display contents and location of media may change up on the execution of the information dialog box.)

5 . Load and Save pf individual mrdia information

The media have individual media information other than the header of TMediaInstance. (MediaHeader2 struction) To load and save thesmedia, it is necessary to load and save the individual media information as well.

(refer to TSMPIInstance::load, TSMPIInstance::saveRefer to)

6 . Copy and Paste of media

The media have individual media information other than the header of TMediaInstance. (MediaHeader2 struction) to copy and paste these media, it is necessary to copy and paste the individual media information as well.

(refer to TSMPIInstance::dmCopy, TSMPIInstance::create)

บรรณานุกรม

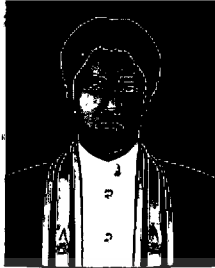
Barry B. Brey "The Z80 Microprocessor" Prentice-Hall, inc. New Jersey. 1988.

Ramesh Gaonkar. "The Z80 Microprocessor" Macmillan Publishing Company, a Division of Macmillan, Inc. USA. 1993.

Telecommunications Advancement Organization of Japan (TAO) "CAI Materials Making Manual" Software Corporation (SCC). Japan. 1996.



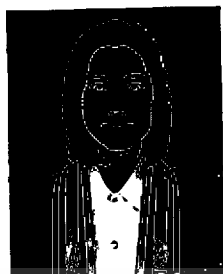
ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นายไพศาล พานนท์
วันเดือนปีเกิด	3 เมษายน 2518
สถานที่เกิด	จังหวัด กรุงเทพมหานคร
ภูมิลำเนาเดิม	จังหวัด กรุงเทพมหานคร
ที่อยู่ปัจจุบัน	19/70 หมู่ 6 หมู่บ้านมโนรมย์ 2 มีนบุรี กรุงเทพฯ 10510
โทรศัพท์	(02) 9158142
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนดลวิทยา
มัธยมศึกษาตอนต้น	โรงเรียนดลวิทยา
ประกาศนียบัตรวิชาชีพ(ปวช.)	วิทยาลัยเทคนิคสมุทรปราการ
ประกาศนียบัตรวิชาชีพชั้นสูง	สถาบันเทคโนโลยีราชมงคล วิทยาเขตเทคนิคกรุงเทพ
ปริญญาตรี	สาขาวิชา อิเล็กทรอนิกส์และ คอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล
ทุนการศึกษา
คดีพจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาวศศิธร เกียรติจรบูรณ์
วันเดือนปีเกิด	22 พฤษภาคม 2519
สถานที่เกิด โรงพยาบาลลำปาง	จังหวัดลำปาง
ภูมิลำเนาเดิม	จังหวัดลำปาง
ที่อยู่ปัจจุบัน	111 หมู่ 8 ต.แจ้ห่ม อ.แจ้ห่ม จ.ลำปาง 25120
โทรศัพท์	(054) 369529
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนบ้านหนองนา
มัธยมศึกษาตอนต้น	โรงเรียนแจ้ห่มวิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคลำปาง
ประกาศนียบัตรวิชาชีพชั้นสูง	สถาบันเทคโนโลยีราชมงคล วิทยาเขต ภาควิชาชีพ
ปริญญาตรี	สาขาวิชา อิเล็กทรอนิกส์และ คอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล
ทุนการศึกษา
คดีพจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาวอัจฉรา สีนวล
วันเดือนปีเกิด	26 สิงหาคม 2517
สถานที่เกิด โรงพยาบาลระยอง	จังหวัดระยอง
ภูมิลำเนาเดิม	จังหวัดระยอง
ที่อยู่ปัจจุบัน	42/3 หมู่ 5 ต.มาบข่า อ.บ้านค่าย จ.ระยอง 21180
โทรศัพท์	(038) 636540
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนวัดมาบข่า
มัธยมศึกษาตอนต้น	โรงเรียนระยองวิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคระยอง
ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)	วิทยาลัยเทคนิคระยอง
ปริญญาตรี	สาขาวิชา อิเล็กทรอนิกส์และ คอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล
ทุนการศึกษา
คตินิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญยานิพนธ์	นายอานัติ สายสกลาม
วันเดือนปีเกิด	21 พฤษภาคม 2516
สถานที่เกิด โรงพยาบาลแมคคอมมิค	จังหวัด เชียงใหม่
ภูมิลำเนาเดิม	จังหวัด ลำปาง
ที่อยู่ปัจจุบัน	259/47 ซ.ดับเพลิง ลาดกระบัง กรุงเทพ 10520
โทรศัพท์	(02) 7380820
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียน อัสสัมชัญ ลำปาง
มัธยมศึกษาตอนต้น	โรงเรียน บุญญวาทย์วิทยาลัย ลำปาง
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคลำปาง
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขต ภาคพายัพ
ปริญญาตรี	สาขาวิชา อิเล็กทรอนิกส์และ คอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล
ทุนการศึกษา
คติพจน์	ทำวันนี้ให้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้