

สารบัญ

บทที่	หน้าที่
1. บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์โครงการ	1
1.3 ขอบเขตโครงการ	1
1.4 วิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ	3
2. ทฤษฎีและเอกสารที่เกี่ยวข้อง	
2.1 บทนำ	4
2.2 ทฤษฎีเกี่ยวกับการประมวลผลภาพ	4
2.3 ทฤษฎีเกี่ยวกับการเขียน โปรแกรม	6
2.4 มาตรฐานพอร์ตขนาน (Parallel Port)	12
2.5 โหมดการทำงานของพอร์ตขนาน	13
2.6 ชนิดของรีจิสเตอร์	15
2.7 การทำงานของพอร์ตเครื่องพิมพ์	16
2.8 ส่วนของการสร้างเว็บไซต์	17
3. การออกแบบ	
3.1 บทนำ	41
3.2 การออกแบบชุดเลือกสัญญาณภาพ	42
3.3 การออกแบบโปรแกรมระบบรักษาความปลอดภัย	43
3.4 การออกแบบเว็บเพจ	47
4. การทดลองและผลการทดลอง	
4.1 บทนำ	48
4.2 การทดสอบการทำงานของแต่ละ โมดูล	48
4.3 การทดสอบการทำงานของระบบโดยรวม	49
5. สรุปผลของโครงการ	
5.1 บทนำ	52
5.2 สรุปผลการทำงานของอุปกรณ์ชุดเลือกสัญญาณภาพ	52

สารบัญ(ต่อ)

บทที่		หน้าที่
	5.3 สรุปผลการทำงาน โปรแกรมตรวจจับสัญญาณภาพ	53
	5.4 สรุปผลการทำงาน ส่วนแสดงผลเว็บเพจ	53
	5.5 สรุปปัญหาที่เกิดขึ้น	53
	5.6 แนวทางการพัฒนา	54



รายงานวิจัย



ระบบรักษาความปลอดภัยเคลื่อนที่
Security On Mobile System

คณะผู้วิจัย

นายสันติ สติตววรรณะ
นายพิทักษ์ สติตววรรณะ



058945

ว. 004.678

ก. 579

จ. 2549

มหาวิทยาลัย เทคโนโลยีราชมงคลศรีวิชัย

วิทยาเขตภาคใต้

พ.ศ. 2549

ได้รับทุนอุดหนุนจากเงินงบประมาณผลประโยชน์ ประจำปีงบประมาณ 2548

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอุปกรณ์ไร้สายทั้งระบบเครือข่ายและโทรศัพท์มือถือได้มีการพัฒนาอย่างแพร่หลายและราคาไม่แพงทำให้เกิดการนำระบบมาประยุกต์ในงานต่างๆ เพิ่มมากขึ้น

จากความต้องการในเรื่องการรักษาความปลอดภัยปัจจุบันได้เพิ่มมากขึ้น แต่ระบบรักษาความปลอดภัยโดยทั่วไปยังคงเป็นแบบทางเดียว คือผู้ดูแลต้องคอยควบคุมและตรวจสอบด้วยสายตา จากจุดนี้เห็นได้ว่าต้องสิ้นเปลืองแรงงานคนและประสิทธิภาพการทำงานไม่คงที่ เนื่องจากคนมีการอ่อนล้าในการทำงาน ผู้จัดทำเห็นว่าหากนำระบบเครือข่ายไร้สายมารวมกับระบบรักษาความปลอดภัยและอาศัยโครงข่ายอินเทอร์เน็ตเป็นตัวเชื่อมระบบให้เกิดการตรวจสอบข้อมูลได้สะดวก โดยสามารถใช้คอมพิวเตอร์และโทรศัพท์มือถือ สำหรับระบบจัดเก็บข้อมูลใช้การประมวลผลภาพให้เกิดการจัดเก็บข้อมูลแบบอัตโนมัติได้

1.2 วัตถุประสงค์โครงการ

1.2.1 เพื่อให้เกิดการนำองค์ความรู้หลายด้านมาประยุกต์ให้เกิดนวัตกรรมใหม่ในการรักษาความปลอดภัยแบบเคลื่อนที่

1.2.2. เพื่อใช้ทรัพยากรเครือข่ายอินเทอร์เน็ตได้อย่างคุ้มค่าและหลากหลาย

1.2.3. เพื่อนำหลักการประมวลผลภาพมาใช้ในการรักษาความปลอดภัย

1.2.4. เพื่อลดต้นทุนในระบบรักษาความปลอดภัยโดยการประยุกต์ใช้อุปกรณ์ที่ไม่ได้จำกัดในเรื่องของคุณลักษณะของอุปกรณ์แต่ละตัว

1.3 ขอบเขตโครงการ

1.3.1 ส่วนของชุดเชื่อมต่อเลือกสัญญาณภาพวิดีโอ

1) รับสัญญาณภาพอินพุตได้ 4 ช่อง และมีเอาต์พุต 1 ช่อง

2) เลือกสัญญาณภาพได้อัตโนมัติ โดยควบคุมจากโปรแกรมควบคุมที่ได้สร้างไว้อยู่ในไมโครคอมพิวเตอร์

1.3.2 ส่วนถ่ายภาพเคลื่อนไหว

- 1) ใช้กล้อง CCD เป็นตัวถ่ายภาพ ติดตั้งบริเวณพื้นที่ที่ต้องการ
- 2) กล้องจะถูกกำหนดตำแหน่งที่ถาวรไม่สามารถหมุนได้
- 3) ส่งสัญญาณไปให้ส่วนแสดงผลผ่านทางสายสัญญาณและแบบไร้สายได้
- 4) ระยะเวลามองเห็น 1 เมตร ขึ้นอยู่กับความสว่างของสภาพพื้นที่

1.3.3 ส่วนควบคุมและแสดงผล

- 1) เป็นละมุนภัณฑ์ (Software) สามารถทำงานบน Windows
- 2) แสดงภาพจากกล้องที่อยู่ตามบริเวณพื้นที่
- 3) แสดงผลการตรวจจับภาพรอบๆที่กล้องวงจรปิดรับภาพได้
- 4) แสดงผลผ่านระบบเครือข่ายอินเทอร์เน็ต ในรูปแบบเว็บไซต์

1.3.7 ส่วนภาคจ่ายไฟ

แหล่งจ่ายไฟของระบบมีอยู่ 3 ส่วน

- 1) ส่วนของกล้องวงจรปิด
- 2) ส่วนของตัวควบคุมเลือกสัญญาณภาพวิดีโอ
- 3) ส่วนของไมโครคอมพิวเตอร์

1.4 วิธีการดำเนินงาน

1.4.1 ศึกษาหลักการการทำงานของระบบวงจรปิด การบันทึกภาพและการเขียนเว็บไซต์ ร่วมกับฐานข้อมูล

1.4.2 ศึกษาหลักการการทำงานของอุปกรณ์ควบคุมการเลือกสัญญาณภาพวงจรปิด และชิ้นส่วนต่างๆ ที่มีลักษณะการทำงานเสมือนระบบรักษาความปลอดภัยราคาสูงในเว็บไซต์ต่างๆ

1.4.3 ศึกษาหลักการการทำงานการส่งข้อมูลระหว่างตัวควบคุมร่วมกับโปรแกรม Microsoft Visual Basic 6.0 ที่ทำงานกับคอมพิวเตอร์ โดยผ่าน Printer Port หรือ Parallel Port

1.4.4 ศึกษาการเขียนโปรแกรมติดต่อกับพอร์ตมาตรฐาน Parallel Port โดยใช้ร่วมกับไมโครคอมพิวเตอร์

1.4.5 ออกแบบส่วนแสดงผลของโปรแกรมควบคุมระบบวงจรปิด และส่วนการแสดงผลบนเว็บไซต์

1.4.6 ออกแบบลักษณะคำสั่งที่ใช้ในการติดต่อรับส่งข้อมูลระหว่างตัวอุปกรณ์ควบคุม การเลือกสัญญาณภาพและไมโครคอมพิวเตอร์

1.4.7 สร้างส่วนของกล่องควบคุมการเลือกสัญญาณภาพวิดีโอ

1.4.8 เขียน โปรแกรมส่วนควบคุมและแสดงผลข้อมูลทั้งหมดของระบบรักษาความปลอดภัย

1.4.9 ทดสอบการทำงานของระบบวงจรปิดและเว็บไซต์นี้ร่วมกับ โปรแกรมควบคุม และปรับปรุงแก้ไข

1.4.10 รวบรวมข้อมูลจัดทำเอกสารประกอบ

1.4.11 ทดสอบระบบทั้งหมดของระบบรักษาความปลอดภัยเคลื่อนที่

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

1.5.1 สามารถนำระบบรักษาความปลอดภัยที่ได้มาใช้แทนคนในจุดย่อยๆ ที่สำคัญ เช่น ภายในตัวอาคารและสถานที่ที่ต้องการสังเกตความเคลื่อนไหว

1.5.2 สามารถนำไปใช้งานได้หลากหลายองค์กร เนื่องจากมีงบประมาณค่าใช้จ่ายที่มีราคาไม่สูง

1.5.3 ลดความยุ่งยากในการตรวจสอบข้อมูลย้อนหลังเกี่ยวกับการทำงานและการรักษาความปลอดภัยเนื่องจากระบบทำการเก็บข้อมูลภาพไว้อัตโนมัติ

1.5.4 สามารถตรวจสอบข้อมูลในระบบรักษาความปลอดภัย ได้ทั้งที่เป็นรูปภาพที่เป็นไฟล์ข้อมูลในเครื่องคอมพิวเตอร์และสามารถตรวจสอบผ่านระบบอินเทอร์เน็ตได้

บทที่ 2

ทฤษฎีและเอกสารที่เกี่ยวข้อง

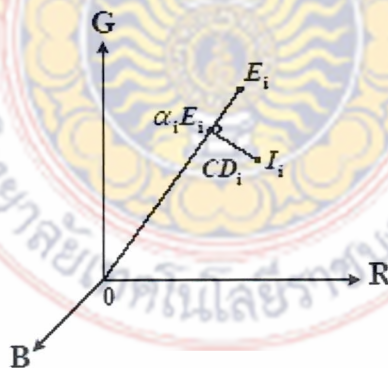
2.1 บทนำ

ในการพัฒนาโปรแกรมระบบรักษาความปลอดภัยนี้ได้นำเอาทฤษฎีและหลักการต่างๆเข้ามาประยุกต์รวมกันเพื่อให้ระบบสามารถทำงานได้ตามเป้าหมายที่ได้กำหนดไว้ ซึ่งส่วนหลักๆจะประกอบไปด้วย 3 ส่วน คือส่วนของการเลือกสัญญาณภาพ ส่วนของการประมวลผลภาพ และส่วนของ Web Programming ในบทนี้จะอธิบายถึงทฤษฎีต่างๆ ที่ใช้อ้างอิงในการสร้างโครงงานนี้ขึ้นมา

2.2 ทฤษฎีเกี่ยวกับการประมวลผลภาพ

การประมวลผลภาพ (Image Processing) นั้นสามารถทำได้หลายวิธี ในโครงงานนี้ได้เลือกเอาการประมวลผลโดยวิธีการคำนวณความแตกต่างของระดับความสว่าง (Brightness Component) และระดับสี (Color Component) ของจุดภาพ โดยใช้ทฤษฎีของแบบโครงสร้างสี (HSI color model) เป็นหลักการเบื้องต้น

แนวคิดที่ได้จากทฤษฎีของแบบโครงสร้างสี (HSI color model) คือ การพิจารณาแยกองค์ประกอบของจุดสีออกเป็น 2 องค์ประกอบ คือ ระดับความสว่าง (Brightness Component) และระดับสี (Color Component)



RGB color model

รูปที่ 2.4 แสดง Object Encapsulation

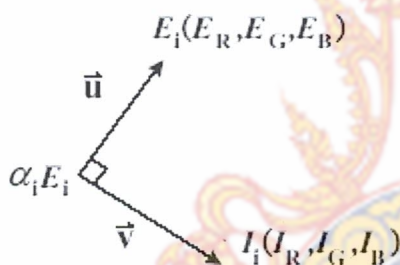
จากรูปจะแสดงแบบโครงสร้างสี (RGB color model) ในลักษณะ 3 มิติ พิจารณาจุด I ในภาพที่มีค่าของสี (R,G,B) ซึ่งมีพิกัดอยู่ในแบบโครงสร้าง

$E_i = [ER(i), EG(i), EB(i)]$ คือ ค่าของสีที่จุดภาพที่ i ในรูปภาพอ้างอิง

$I_i = [IR(i), IG(i), IB(i)]$ คือ ค่าของสีที่จุดภาพที่ i ในรูปภาพที่นำมาเปรียบเทียบ
 OE_i คือ เส้นตรงที่ลากจากจุดกำเนิด(0,0,0) ไปยังจุด E_i โดยใช้เป็นเส้นสังเกต
 CD_i หรือ การบิดเบือนของสี (Color Distortion) คือ ระยะห่างในแนวตั้งฉากระหว่างค่า
 ของสี(R,G,B) ของรูปภาพหรือเฟรมที่ต้องการ นำมาเปรียบเทียบกับเส้นสังเกต OE_i

α_i หรือการบิดเบือนของความสว่าง (Brightness Distortion) คือ ค่าคงที่ที่ทำให้ค่าของสี
 (R,G,B) ของรูปภาพหรือเฟรมที่ต้องการนำมาเปรียบเทียบกับค่าเข้าใกล้เส้นสังเกต OE_i หลักการพื้น
 ฐาน คือ การคำนวณการบิดเบือน หรือการเปลี่ยนแปลงขององค์ประกอบสีที่ตำแหน่งจุดภาพเดียว
 กัน (จุดภาพที่ i) ระหว่างรูปภาพหรือเฟรมที่ต้องการนำมาเปรียบเทียบ โดยใช้ตัวแปรในการเปรียบ
 เทียบ คือ CD_i และ α_i

ความสัมพันธ์ระหว่าง E_i และ I_i โดยใช้ทฤษฎีของเวกเตอร์ดังแสดงในภาพ



Vector

การหาค่าตัวแปรทั้งสองจะใช้คุณสมบัติของเวกเตอร์ในการคำนวณโดยที่

$$\begin{aligned} \vec{u} &= E_i - \alpha_i E_i \\ &= E_i(1 - \alpha_i) \end{aligned} \quad (1)$$

$$\vec{v} = I_i - \alpha_i E_i \quad (2)$$

$$\vec{u} \cdot \vec{v} = E_i(1 - \alpha_i) \cdot (I_i - \alpha_i E_i) \quad (3)$$

จากนั้นทำการหาผลลัพท์ (Dot product)

$$(1 - \alpha_i)[E_R(I_R - \alpha_i E_R) + E_G(I_G - \alpha_i E_G) + E_B(I_B - \alpha_i E_B)] = 0 \quad (4)$$

เนื่องจากเวกเตอร์ทั้งสองด้านตั้งฉากกัน ทำให้ผลลัพท์ (Dot product) มีค่าเป็น 0 จะได้ค่า α_i เป็น

$$\alpha_i = \frac{E_R I_R + E_G I_G - E_B I_B}{E_R^2 + E_G^2 + E_B^2} \quad (5)$$

จากนั้นแทนค่า α_i ที่คำนวณได้ในสมการที่ (4) จะได้ค่า CD_i เป็น

$$CD_i = \sqrt{(I_R - \alpha_i E_R)^2 + (I_G - \alpha_i E_G)^2 + (I_B - \alpha_i E_B)^2} \quad (6)$$

การกำหนดเงื่อนไข 3 ข้อ เพื่อใช้ในการตรวจจับการเปลี่ยนแปลงของจุดภาพคือ

1. $CD_i > k_1$
2. $k_1 < \alpha_i < k_2$
3. $DEI > k_3$, $DEI = E_i - I_i$

โดยที่ k_1, k_2, k_3 คือ ค่าคงที่ที่ได้จากการทดลองหาค่าที่ทำให้โปรแกรมสามารถตรวจจับได้อย่างถูกต้องมากที่สุด

2.3 ทฤษฎีเกี่ยวกับการเขียนโปรแกรม

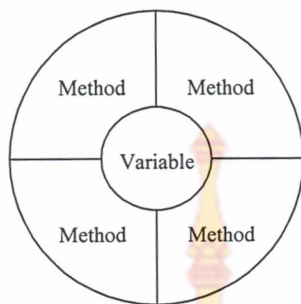
2.3.1 แนวคิดการเขียนโปรแกรมแบบติดต่อกับผู้ใช้ในการควบคุม

แนวคิดการสร้างโปรแกรมที่ประกอบขึ้นจากวัตถุเป็นหลักนี้เรียกว่า การโปรแกรมเชิงวัตถุ (Object Oriented Programming: OOP) และภาษาที่สนับสนุนการสร้างวัตถุจะเรียกว่า ภาษาเชิงวัตถุ (Object Oriented Languages: OOL)

วัตถุจริง (Real-World Object) จะมีคุณสมบัติ 2 อย่างร่วมกัน คือ สถานะ (State) และพฤติกรรม (Behavior) ยกตัวอย่าง รถจักรยานยนต์มีสถานะเช่น มี 2 ล้อ, มีตำแหน่งเกียร์ปัจจุบัน, มีจำนวนเกียร์ทั้งหมด และมีเบรก เป็นต้น ส่วนพฤติกรรมที่จะเกิดขึ้น เช่น การเบรก และการเร่งความเร็ว เป็นต้น

วัตถุเสมือน (Software Object) เป็นแบบจำลองของวัตถุจริง ซึ่งจะมีสถานะ (State) และพฤติกรรม (Behavior) เช่นเดียวกัน ในวัตถุเสมือนจะเรียกสถานะว่า “ตัวแปร” และเรียกพฤติกรรมว่า “วิธีการ” ดังนั้นเราสามารถที่จะให้คำจำกัดความได้ว่าวัตถุเสมือนก็คือ “ละมุนกันท์ที่รวบรวมตัวแปร และ วิธีการ เข้าไว้ด้วยกัน”

ทุกตัวแปรและวิธีการของวัตถุเสมือนจะเก็บไว้ในวัตถุนั้น ๆ เช่น จักรยานยนต์ที่จำลองมาจากจักรยานยนต์จริง จะต้องมีตัวแปรที่จะแสดงสถานะปัจจุบัน เช่น มีความเร็ว 100 กิโลเมตร/ชั่วโมง ความเร็วรอบในการหมุนของล้อ 900 รอบ/นาที ตัวแปรและวิธีการเหล่านี้เรียกว่า ตัวแปรตัวอย่าง และ วิธีการตัวอย่าง



รูปที่ 2.2 แสดง Object Encapsulation

ตัวแปรจะเป็นศูนย์กลางวัตถุโดยมีวิธีการล้อมรอบภายในวัตถุ และซ่อนตัวแปรเอาไว้จากวัตถุอื่น ๆ การรวมตัวแปร และ วิธีการ จะเรียกว่า Object Encapsulation ดังนั้นเมื่อเราต้องการใช้งานวัตถุใด ๆ ก็จะไม่รู้ว่าวัตถุนั้นทำงานอย่างไร เช่น เมื่อเราต้องการเปลี่ยนเกียร์รถจักรยานยนต์ ก็ไม่ต้องรู้ว่าระบบเกียร์ทำงานอย่างไร แต่ต้องรู้ว่าจะเปลี่ยนไปที่เกียร์ไหน

ข้อดีของ Encapsulation

1) Modularity - แหล่งรหัส (Source Code) สามารถที่จะเขียนและจัดการได้อย่างเป็นอิสระ จากแหล่งรหัสอื่น ๆ และวัตถุนั้นก็ยังใช้งานในระบบได้อย่างสมบูรณ์

2) Information Hiding Object จะมีการติดต่อแบบสาธารณะ (Public Interface) เพื่อที่จะทำให้วัตถุอื่น ๆ สามารถเข้ามาติดต่อกับวัตถุนั้นได้ โดยจะมีตัวแปร และ วิธีการที่จะมีความเป็นส่วนตัว (Private) ซึ่งสามารถที่จะเปลี่ยนแปลงการทำงานของวัตถุโดยไม่มีผลกระทบต่อวัตถุอื่น ๆ ที่เข้ามาติดต่อ

การแลกเปลี่ยนข้อมูลระหว่างวัตถุจะใช้ข้อความในการติดต่อ เนื่องจากวัตถุเพียง วัตถุเดียวทำงานได้ไม่มากนัก โดยข้อความจะประกอบด้วยวัตถุปลายทางที่ต้องการรับข้อความ, ชื่อของวิธีการของวัตถุที่ต้องใช้ในการติดต่อ และเงื่อนไขของแต่ละวิธีการที่ต้องการ เช่น คนสั่งให้เปลี่ยนเกียร์จักรยานยนต์ก็จะได้ข้อความ คือ ChangeGears (LowGears) เป็นต้น

2.3.2 การทำงานของวัตถุใน Visual Basic

ในที่นี้จะขอกล่าวถึงการทำงานของโปรแกรมวัตถุที่ใช้ใน Visual Basic เท่านั้น วัตถุเป็นเรื่องที่สำคัญที่สุดของ Visual Basic เนื่องจากทุก ๆ ส่วนประกอบใน Visual Basic นั้นจะเป็นวัตถุทั้งหมด ไม่ว่าจะเป็นหน้าต่าง ส่วนควบคุม หรือแม้เป็นการติดต่อกับฐานข้อมูลหรือการติดต่อ

กับระบบเครือข่ายก็ตาม ดังนั้นหากต้องการใช้ Visual Basic ได้อย่างเต็มประสิทธิภาพแล้วก็เป็นที่จะต้องเข้าใจการทำงานของวัตถุ

1) วัตถุใน Visual Basic จะประกอบไปด้วย 2 ส่วนคือ

ก) ข้อมูล (Data) จะเก็บข้อมูลต่าง ๆ ของวัตถุไว้ ดังนั้นข้อมูล ก็คือ คุณสมบัติ (Properties)

ข) รหัส (Code) จะเก็บคำสั่งต่าง ๆ ที่จะให้วัตถุนั้นทำงานให้กับเรา เช่น วิธีการเคลื่อนที่ (Move) ของปุ่มคำสั่ง จะเป็นการควบคุมไปยังตำแหน่งอื่นบนหน้าต่างโปรแกรม

ดังนั้นสรุปได้ว่าวัตถุเป็นการรวบรวมคำสั่งและข้อมูลเข้าไว้ด้วยกัน และ วัตถุไม่ใช่การควบคุม แต่การควบคุมเป็นเพียงส่วนหนึ่งของวัตถุเท่านั้น วัตถุใน Visual Basic จะสร้างมาจากห้อง (Class) ซึ่งห้องก็คือ การสร้างรายละเอียดต่าง ๆ ของวัตถุที่จะสร้าง เช่น กำหนดว่าจะมีคุณสมบัติอะไรบ้าง มีวิธีการที่จะทำอะไรได้บ้าง และมีเหตุการณ์ (Event) ที่จะเกิดขึ้นในขณะใด ๆ ดังนั้นก็คือเป็นการกำหนดพฤติกรรมของวัตถุที่จะสร้างขึ้นมานั่นเอง ตัวอย่างของความสัมพันธ์ระหว่างวัตถุและห้องคือ การควบคุมที่อยู่บนกล่องเครื่องมือ (Tools Box) นั้นจะแทนห้อง เมื่อนำการควบคุมมาสร้างบนหน้าต่างก็จะเป็นการสร้างแบบตัวอย่าง (Instance) ของห้องซึ่งมันก็คือ วัตถุนั่นเอง และวัตถุที่สร้างมาจากห้องเดียวกันจะเป็นวัตถุที่ไม่เกี่ยวข้องกัน แต่จะมีคุณสมบัติ วิธีการ และเหตุการณ์เหมือนกันเพราะมาจากห้องเดียวกัน

2) พื้นฐานในการทำงานกับวัตถุในการทำงานกับวัตถุเราจะทำงานกับ คุณสมบัติ วิธีการและเหตุการณ์ โดยคุณสมบัติจะบรรยายลักษณะของวัตถุ วิธีการจะเป็นการสั่งให้วัตถุทำงานให้ ส่วนเหตุการณ์จะเป็นเหตุการณ์ต่าง ๆ ที่เราต้องใส่คำสั่งตอบสนอง ตัวอย่างคุณสมบัติ เช่น จักรยานยนต์ยี่ห้อ 2 ล้อ มีเบรก และมีวิธีการเคลื่อนที่ เพื่อให้จักรยานยนต์ไปยังตำแหน่งที่กำหนดและเหตุการณ์ Crash ที่เป็นเหตุการณ์ที่เกิดขึ้นเมื่อจักรยานยนต์ถูกชน การทำงานกับวัตถุนั้นมีดังต่อไปนี้

ก) การกำหนดค่าให้คุณสมบัติ จะใช้รูปแบบคือ

<ชื่อวัตถุ>.<ชื่อคุณสมบัติ> = ค่าที่กำหนดให้ เช่น Form1.Caption = "Security Online"

ข) การอ่านค่าคุณสมบัติของวัตถุ จะใช้รูปแบบคือ

ข) ทำให้การทำงานของโปรแกรมเร็วขึ้น เนื่องจากไม่ต้องบรรจุ (Load) เพิ่มข้อมูลทำงานที่มีขนาดใหญ่ แต่บรรจุเพิ่มข้อมูล DLL ที่ต้องใช้งานเท่านั้น

ค) ทำให้โปรแกรมทำงานบางอย่าง อย่างที่ VB ไม่สามารถทำได้หรือทำได้แต่ไม่มีประสิทธิภาพเท่าที่ควร

3) การเรียกใช้ Windows API การเรียกใช้ Windows API มีขั้นตอนคือ ใช้คำสั่งในการประกาศหน้าที่และโปรแกรมย่อยที่จะใช้ในโปรแกรมของเรา

ก) การเรียกใช้หน้าที่หรือโปรแกรมย่อยที่ได้ประกาศเอาไว้แล้วคำสั่งที่ใช้ในการประกาศหน้าที่หรือโปรแกรมย่อยนั้นจะบอกว่าหน้าที่หรือโปรแกรมย่อยนั้นหาได้จากเพิ่มข้อมูลชื่ออะไรโดยคำสั่งที่ใช้คือ ประกาศตัว (Declare) โดยมีรูปแบบของการประกาศคือ

ข) การประกาศโปรแกรมย่อย

รูปแบบ : `Declare Sub <Name> Lib "LibName" [Alias "AliasName"] (<Parameter>)`

โดยที่ Name คือ ชื่อของโปรแกรมที่เรียกใช้ภายในโปรแกรมของเรา
 LibName คือ ชื่อของเพิ่มข้อมูล DLL ที่มีโปรแกรมย่อยนั้นอยู่
 Alias คือ การบอกว่าสามารถใช้ชื่อสมมุติ (Alias Name) แทนชื่อของโปรแกรมย่อยนั้นได้ ส่วนมากใช้เมื่อ ชื่อของโปรแกรมย่อยซ้ำกับคำสั่งวงของโปรแกรม (Visual Basic)

ค) การประกาศหน้าที่

รูปแบบ :

`Declare Function <Name> Lib "LibName" [Alias "AliasName"](<Parameter>)[As_`

`Type] โดยที่`

Name คือ ชื่อของโปรแกรมที่เรียกใช้ภายในโปรแกรมของเรา

LibName คือ ชื่อของเพิ่มข้อมูล DLL ที่มีหน้าที่นั้นอยู่

Alias เป็นการบอกว่าสามารถใช้ชื่อสมมุติ แทนชื่อของหน้าที่นั้นได้ ส่วนมากจะใช้เมื่อชื่อของหน้าที่ซ้ำกับคำสั่งวงของโปรแกรม (Visual Basic) ตัวอย่างการใช้งาน เช่น

`Declare Function SetWindowsPos Lib "User32" (ByVal h As Long, ByVal hb_ As Long, ByVal x As Long, ByVal y As Long, ByVal cy As Long, ByVal f As_ Long) As Long`

4) เงื่อนไขหรือรูปแบบใน Windows API

ในการส่งค่าของการใช้งาน DLL เราจำเป็นต้องทราบว่าต้องการส่งเงื่อนไขแบบส่งค่าไปอย่างเดียว (By Value) หรือว่าจะส่งค่าแบบอ้างอิง (By Reference)

ก) การส่งค่าแบบไปอย่างเดียว เป็นการส่งค่าของเงื่อนไขไปยังหน้าที่ หรือโปรแกรมย่อยที่เรียกใช้เท่านั้น โดยที่หน้าที่หรือโปรแกรมย่อยไม่สามารถเปลี่ยนแปลงค่าของตัวแปรที่ส่งเป็นเงื่อนไขนั้นได้เลย ตัวอย่างเช่น

```
Declare Function MyFunction Lib "MyDLL" (ByVal MyVar As Long) As Long
```

ข) การส่งค่าแบบอ้างอิง จะเป็นการส่งตำแหน่งที่อยู่ของหน่วยความจำ (Address) ของตัวแปรที่เป็นเงื่อนไขไปแทนทำให้หน้าที่หรือโปรแกรมย่อยที่เรียกใช้เงื่อนไข สามารถแก้ไขตัวแปรนั้นได้ ตัวอย่างเช่น

```
Declare Function MyFunction Lib "MyDLL" (ByVal MyVar As String) As Long
```

5) คุณสมบัติที่สำคัญของส่วนควบคุมที่ต้องการในการเรียกใช้ Windows API

ก) คุณสมบัติของ hWnd อธิบายได้ดังนี้ โดยทั่วไปถ้าเราต้องการแก้ไขหน้าต่าง หรือส่วนควบคุมด้วย Windows API จะต้องสร้างค่าของหน้าต่างปฏิบัติการ (Windows Handle) ที่มีอยู่ในแต่ละหน้าต่างและส่วนควบคุมนั้นเป็นเงื่อนไขไปด้วย โดยใช้คุณสมบัติ hWnd ของส่วนควบคุมตัวนั้นส่งค่าเงื่อนไขที่เป็นการควบคุมไป

ข) คุณสมบัติของ hDC อธิบายได้ดังนี้ เป็นคุณสมบัติที่เป็นการควบคุมของสิ่งแวดล้อมของอุปกรณ์ (Device Context) ของวัตถุต่าง ๆ เช่น หน้าต่างและส่วนควบคุมต่าง ๆ ใช้เพื่อวาดรูปภาพ (Graphic)

6) การเปลี่ยนชนิดของข้อมูลเมื่อใช้ Windows API ในการเรียกใช้หน้าที่ หรือโปรแกรมย่อยในเพิ่มข้อมูล DLL ของหน้าต่างจะเหมือนกับการเรียกใช้หน้าที่หรือโปรแกรมย่อย โดยทั่วไปยกเว้นการส่งเงื่อนไข เนื่องจากเพิ่มข้อมูล DLL ไม่ได้สร้างมาจาก Visual Basic แต่ถูกสร้างขึ้นมาโดยภาษา C, C++ ซึ่งมีชนิดของข้อมูลแตกต่างจาก Visual Basic ดังนั้นการใช้งานจึงจำเป็นต้องระวังในเรื่องชนิดของข้อมูลที่ใช้ในการส่งเงื่อนไขซึ่งเราจะเปลี่ยนชนิดของข้อมูลที่ใช้ในภาษา C ใน Windows API ไปเป็นชนิดของข้อมูลใน Visual Basic โดยการใช้อารางที่ 2.1 ตารางที่ 2.1 แสดงการเปลี่ยนชนิดของข้อมูลใน Windows API ไปเป็นชนิดของข้อมูลใน VB

ชนิดของข้อมูลใน Windows API	ชนิดของข้อมูลใน VB
Int	ByVal <ชื่อตัวแปร> As Long
Float	ByVal <ชื่อตัวแปร> As Single
Double	ByVal <ชื่อตัวแปร> As Double
Pointer (ชี้ไปยัง String)	ByVal <ชื่อตัวแปร> As String
Pointer 32 bit	ByVal <ชื่อตัวแปร> As Long
Null Pointer	VbNullString
Void Pointer	<ชื่อตัวแปร> As Any
BOOL	ByVal <ชื่อตัวแปร> As Boolean

2.4 มาตรฐานพอร์ตขนาน (Parallel Port)

Parallel Port เป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับสื่อสารกับอุปกรณ์ต่อพ่วงชนิดอื่น ๆ เช่นเครื่องพิมพ์ (Printer) หรืออุปกรณ์บันทึกข้อมูลที่ต้องการความเร็วในการส่งข้อมูลที่สูงกว่าพอร์ตอนุกรม (Serial Port) โดยเฉพาะอย่างยิ่งเมื่อมีการนำพอร์ตนี้ไปใช้งานอย่างแพร่หลายกับอุปกรณ์ต่อพ่วงชนิดเครื่องพิมพ์ทั้งเครื่องพิมพ์ชนิดกระแทก (Dot matrix) เครื่องพิมพ์ชนิดพ่นหมึก (Ink Jet Printer) และเครื่องพิมพ์ชนิดเลเซอร์ (Laser Printer) ซึ่งเป็นที่มาของชื่อ "พริ้นเตอร์พอร์ต"

ประกอบไปด้วยขาสัญญาณเอาต์พุตที่เป็นบิตข้อมูล (Data Bit) จำนวน 8 ขาสัญญาณ ขาสัญญาณเอาต์พุตที่เป็นบิตควบคุม (Control Bit) จำนวน 4 ขาสัญญาณ และขาสัญญาณอินพุตที่เป็นบิตสถานะ (Status Bit) จำนวน 5 ขาสัญญาณ ดังนี้

ขาสัญญาณ	คำอธิบาย	ขาสัญญาณ	คำอธิบาย
1	ขา Strobe (Control Bit0)	10	ขา ACK (Status Bit6)
2	ขา Data 0 (Data Bit0)	11	ขา Busy (Status Bit7)
3	ขา Data 1 (Data Bit1)	12	ขา Paper Empty (Status Bit5)
4	ขา Data 2 (Data Bit2)	13	ขา Select (Status Bit4)
5	ขา Data 3 (Data Bit3)	14	ขา Auto Feed (Control Bit1)
6	ขา Data 4 (Data Bit4)	15	ขา Error (Status Bit3)
7	ขา Data 5 (Data Bit5)	16	ขา Initialize Printer (Control Bit2)
8	ขา Data 6 (Data Bit6)	17	ขา Select Input (Control Bit3)
9	ขา Data 7 (Data Bit7)	18-25	ขากราวด์ (Ground Signal)

2.5 โหมดการทำงานของพอร์ตขนาน

1.Normal Mode หรือ SPP Mode (Standard Parallel Port) เป็นชื่อระบุของการส่งข้อมูลจากคอมพิวเตอร์ไปเครื่องพิมพ์ตามปกติ หรือเรียกว่าเป็นโหมดปกติ ซึ่งเป็นพอร์ตขนานดั้งเดิม ที่ถูกพัฒนามาพร้อมกับเครื่อง IBM นั้นเอง เพียงแต่ได้รับการปรับปรุงเพิ่มเติมโดยคงความเข้ากันได้กับโหมดพอร์ตขนานแบบเดิมไว้ ดังนั้นจึงเรียกกันโดยทั่วไปว่า Compatibility Mode หรือ Centronics Mode ข้อสังเกตสำหรับการทำงาน ของโหมดปกติก็คือมีการส่งข้อมูลในทิศทางเดียว จากคอมพิวเตอร์ไปยังอุปกรณ์ปลายทาง เครื่องพิมพ์ SPP Mode ถูกพัฒนาให้มีการส่งข้อมูลความเร็วสูงขึ้น ราวประมาณ 500 กิโลไบต์ต่อวินาที ซึ่งเห็นได้ว่าเพียงพอสำหรับเครื่องพิมพ์แบบเข็ม (Dot Matrix) และเครื่องพิมพ์แบบเลเซอร์รุ่นเก่า แต่ไม่เพียงพอสำหรับการ์ดเครือข่าย หรือสำหรับไคร์ที่แบบเคลื่อนย้ายได้ และเครื่องพิมพ์เลเซอร์รุ่นใหม่

2.EPP Mode (Enhanced Parallel Port) พัฒนาโดย Intel, Xircom and Zenith Data Systems เมื่อปี 1994 โดยมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพของพอร์ตขนาน พอร์ตขนานแบบ EPP ยังเข้ากันได้กับ SPP ส่วนที่ถูกพัฒนาขึ้นคือ พัฒนาให้มีการรับส่งข้อมูลสองทิศทางที่ความเร็วสูงขึ้น 500 กิโลไบต์ถึง 2 เมกะไบต์ต่อวินาที ทั้งนี้เพื่อใช้งานพอร์ตขนานกับอุปกรณ์อื่นๆนอกเหนือไปจากเครื่องพิมพ์ อุปกรณ์ที่นำมาใช้กับพอร์ตขนานแบบ EPP คือ ฮาร์ดดิสก์ การ์ดเครือข่าย คุณ

สมบัติของ EPP คือ สามารถส่งข้อมูลได้ 2 ทิศทาง (Bi-directional) และความเร็วในการส่งข้อมูลมากขึ้น ประโยชน์ของการส่งข้อมูลแบบสองทิศทาง คือ ทำให้การสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ต่อพ่วงที่นำมาใช้งานกับพอร์ตขนานดังกล่าวได้ ทำให้สามารถตรวจสอบสถานะและรายงานสถานะ การทำงานไปยังคอมพิวเตอร์ได้ จากการพัฒนาดังกล่าวส่งผลให้ผู้ผลิตเครื่องพิมพ์หลายราย โดยเฉพาะเครื่องพิมพ์ประเภทเลเซอร์และเครื่องพิมพ์ประเภทจีดหมึกต่างก็พัฒนาเครื่องพิมพ์ของตนให้ใช้ประโยชน์จากพอร์ตขนานแบบใหม่ได้ เช่น พิมพ์งานเร็วมากขึ้นและมีการตรวจสอบสถานะการทำงานของเครื่องพิมพ์ทุกๆระยะ เช่น ตรวจสอบว่าเครื่องพิมพ์อยู่ในสภาพพร้อมใช้งานหรือไม่มีกระดาษสำหรับพิมพ์งานหรือไม่ รวมไปถึงการตรวจสอบปัญหาความผิดพลาดต่างๆ และแสดงข้อความบอกสถานะนั้นบนหน้าจอคอมพิวเตอร์ เช่น Out of Paper , Cover Open ON ซอฟต์แวร์ควบคุมการพิมพ์บางรายถึงขนาดแจ้งวิธีแก้ปัญหาแบบทีละขั้นตอน (Step by Step) บนหน้าจอคอมพิวเตอร์ ทำให้ผู้ใช้งานแก้ปัญหาได้ด้วยตนเองได้

3.ECP Mode (Extended Capabilities Port) ECP ถูกเสนอ โดย Hewlett Packard และ Microsoft เพื่อนำมาใช้งานกับเครื่องพิมพ์และสแกนเนอร์รวมถึงอุปกรณ์ต่อพ่วง (Peripheral) อื่นๆ ECP สามารถใช้ DMA Channel (Direct Memory Access Channel) ได้ทำให้ความเร็วการส่งข้อมูลเพิ่มขึ้นเพราะไม่ต้องรอเวลาเมื่ออุปกรณ์ที่นำมาต่อพอร์ตขนานต้องการเข้าถึงหน่วยความจำ ในขณะที่เดียวกันก็ลดการขัดจังหวะการทำงาน ของซีพียู ความเร็วในการส่งข้อมูลสูงถึง 2 เมกะไบต์ ต่อวินาที เครื่องพิมพ์รุ่นใหม่ ๆ หลายรุ่นปัจจุบันสามารถใช้งานโหมด ECP ซึ่งทำให้สามารถส่งข้อมูลสื่อสารกับคอมพิวเตอร์ได้มากขึ้น แสดงสถานะ การทำงานและทำการตรวจสอบการทำงานของตัวเองได้

2.6 ชนิดของรีจิสเตอร์

รีจิสเตอร์ชนิดเดต้า (Data Register) มีตำแหน่งพอร์ตอยู่ที่ IOBase + 0 มีความกว้างของข้อมูลเป็นขนาด 1 ไบท์ (8 บิต) สามารถทำการอ่านหรือเขียนข้อมูลได้ (Read/Write Access) โดยข้อมูลแต่ละบิตจะเชื่อมโยงกับขาสัญญาณของคอนเน็กเตอร์ (DB25) ที่ขาสัญญาณ 2-9 คือ บิตที่ 0 เชื่อมโยงกับขาสัญญาณที่ 2 , บิตที่ 1 เชื่อมโยงกับขาสัญญาณที่ 3 และบิตที่ 7 เชื่อมโยงกับขาสัญญาณที่ 9 ตามลำดับ

จิสเตอร์ชนิดสถานะ (Status Register) มีตำแหน่งพอร์ตอยู่ที่ IOBase + 1 มีความกว้างของข้อมูลเป็นขนาด 1 ไบท์ (ใช้จริง 5 บิต) สามารถทำการอ่านข้อมูล (Read Access Only) จากรีจิสเตอร์นี้ได้โดยตรง โดยข้อมูลแต่ละบิตจะเชื่อมโยงกับขาสัญญาณของคอนเน็กเตอร์ (DB25) ดังนี้

058945
 2004.678
 ศ 579
 2549

ตำแหน่งบิต	ขาสัญญาณ	ความหมายของสัญญาณ
7	11	BUSY
6	10	-ACK
5	12	NO PAPER
4	13	SELECTED
3	15	-ERROR
2-0	N/A	N/A

รีจิสเตอร์สำหรับควบคุม (Control Register) มีตำแหน่งพอร์ตอยู่ที่ IOBase + 2 มีความกว้างขนาดข้อมูล 1 ไบท์ (ใช้จริง 4 บิต) สามารถทำการอ่านหรือเขียนข้อมูลได้ (Read/Write Access) แต่โดยปกติจะใช้เขียนข้อมูลอย่างเดียว ข้อมูลแต่ละบิตจะเชื่อมโยงกับขาสัญญาณของคอนเน็กเตอร์ (DB25) โดยมี 3 บิตเป็นชนิด Reversed กล่าวคือตำแหน่งบิตที่ 0,1 และ 3 จะมีสถานะปกติของลอจิกเป็น 1 (สถานะปกติของบิตทั่วไปจะเป็น 0) ดังนี้

ตำแหน่งบิต	ขาสัญญาณ	ชนิดบิต	ความหมายของสัญญาณ
0	1	Reversed Bit	-STROBE
1	14	Reversed Bit	-AUTOFEED
2	16	Normal	-INITIALIZE
3	17	Reversed Bit	-SELECT
4-7	N/A	N/A	N/A

2.7 การทำงานของพอร์ตเครื่องพิมพ์

เราสามารถแบ่งกลุ่มของพอร์ตเครื่องพิมพ์เพื่อศึกษาหลักการทำงานออกได้เป็น 2 กลุ่ม คือ กลุ่มอินพุต (Input) และ กลุ่มเอาต์พุต (Output) และมี 2 สภาวะ คือ High และ Low

ในสภาวะ High เป็นสภาวะที่ขา (Pin) บนพอร์ตเครื่องพิมพ์มีแรงดันอยู่ระหว่าง +2.4 ถึง +5 โวลต์ (Volt) ซึ่งกำหนดให้ลอจิก (Logic) เป็น 1

ในสถานะ Low เป็นสถานะที่ขา (Pin) บนพอร์ตเครื่องพิมพ์มีแรงดันอยู่ระหว่าง +0 ถึง +0.8 โวลต์ (Volt) ซึ่งกำหนดให้ลอจิก (Logic) เป็น 0

กลุ่มเอาต์พุต (Output)

จะพบว่ามี 8 เอาต์พุต อยู่บนพอร์ตข้อมูล (Data Port) และ 4 เอาต์พุต อยู่บนพอร์ตควบคุม (Control Port) โดยเครื่องคอมพิวเตอร์จะใช้ลอจิก (Logic) 0 เป็นตัวตรวจสอบเครื่องพิมพ์ว่ามีสถานะการเลือกเครื่องพิมพ์บนขา Select Input (Input คือ เครื่องพิมพ์), การกำหนดค่าเริ่มแรกบนขา Initialize Printer, การเลื่อนกระดาษขึ้นบรรทัดใหม่บนขา Auto Feed และการส่งข้อมูลออกจากเครื่องพิมพ์บนขา Strobe หรือ ไม่

อาจมีข้อสงสัยอีกว่าทำไมผู้ออกแบบพอร์ตเครื่องพิมพ์ต้องออกแบบให้ยุ่งยากแบบนี้ด้วย ทำไมไม่ออกแบบให้ลอจิกที่อ่านได้บนวงจร Input / Output กับพอร์ตเครื่องพิมพ์ มีค่าเหมือนกัน โดยปกติถ้าเชื่อมต่อเครื่องพิมพ์อยู่กับคอมพิวเตอร์ก็ไม่มีปัญหา แต่ถ้าไม่ได้เชื่อมต่อเครื่องพิมพ์กับคอมพิวเตอร์จะทำให้ขา Strobe, Auto Feed, Select Input มีสถานะเป็นลอจิก 1 จะทำให้คอมพิวเตอร์มองเหมือนกับว่าต่ออยู่กับเครื่องพิมพ์ทำให้มีการเลือกเครื่องพิมพ์, การกำหนดให้เลื่อนกระดาษขึ้นบรรทัดใหม่, การเตรียมพร้อมในการส่งข้อมูลออกไปยังเครื่องพิมพ์ ทำให้เกิดความสับสนเป็นอย่างมาก ผู้ออกแบบจึงให้อินเวอร์เตอร์ (Inverter) เป็นตัวปรับสถานะทางลอจิก (logic) ให้เหมาะสมกับความเป็นจริง

กลุ่มอินพุต (Input)

จะพบว่ามี 5 ขา อินพุต อยู่บนพอร์ตสถานะ (status port) โดยทำหน้าที่ตรวจสอบสถานะที่ถูกส่งมาจากเครื่องพิมพ์ให้สถานะ High ขา Select จะตรวจสอบว่าเครื่องพิมพ์เชื่อมต่ออยู่หรือไม่ ส่วนสถานะ High ของขา Busy และ PE จะให้ตรวจสอบว่าเครื่องพิมพ์กำลังทำงานอยู่และกระดาษบนเครื่องพิมพ์หมด ส่วนในสถานะ Low ของขา ใช้ตรวจสอบว่าเครื่องพิมพ์ได้รับข้อมูลที่ส่งไปให้ ส่วนสถานะ Low ของขา Error ตรวจสอบข้อผิดพลาดที่เกิดขึ้นบนเครื่องพิมพ์

อาจสงสัยว่าทำไมขา Busy มีการกำหนดสถานะไม่ตรงกัน เพราะเป็นค่าสถานะที่แสดงอยู่บน CPU และวงจร Input / Output ไม่ใช่ค่าที่แสดงอยู่บนพอร์ตเครื่องพิมพ์ชนิด D แบบ 25 ขา โดยที่ผู้ออกแบบจะใช้อินเวอร์เตอร์ (inverter) ในการแปลงสัญญาณลอจิกที่ส่งเข้า CPU และวงจร Input / Output ที่ต้องมีการแปลงสัญญาณ เนื่องจากขณะที่เครื่องพิมพ์ หรืออุปกรณ์ต่อพ่วง (peripheral) ต่าง ๆ กำลังทำงานอยู่นั้น อุปกรณ์เหล่านี้จะส่งลอจิก 1 มาที่พอร์ตเครื่องพิมพ์ชนิด D แบบ 25 ขา ถ้าเราไม่ทำการแปลงสัญญาณก่อนที่ส่งเข้าไปในวงจร Input / Output จะทำให้คอมพิวเตอร์มองว่า

เครื่องพิมพ์ หรืออุปกรณ์ต่อพ่วง (peripheral) อยู่ในสภาวะปกติ (การทำงานเสร็จเรียบร้อยแล้ว) แทนที่จะมองว่าขณะนี้อุปกรณ์กำลังทำงานอยู่

2.8 ส่วนของการสร้างเว็บไซต์

โปรแกรม Dreamweaver เป็นโปรแกรมช่วยออกแบบเว็บเพจ โปรแกรมหนึ่งที่มีความสามารถในการออกแบบเว็บเพจและการจัดการเว็บไซต์ได้อย่างมีประสิทธิภาพและเป็นที่ยอมรับสำหรับเว็บมาสเตอร์ และผู้สนใจทั่วไป

ส่วนประกอบของ Dream weaver

1. ไตเติลบาร์ (Title Bar) ส่วนที่อยู่บนสุดของหน้าต่างโปรแกรม เป็นส่วนที่แสดงชื่อไฟล์และหัวข้อของเว็บเพจ (ปกติจะเป็น Untitled จนกว่าจะมีการเปลี่ยนชื่อไฟล์) และกลุ่มของปุ่มทางขวามือซึ่งจะใช้ควบคุมการย่อ/ขยาย/ปิดหน้าต่างที่แสดงอยู่

2. เมนูบาร์ (Menu Bar) แถบเมนูคำสั่งที่ใช้จัดการกับเว็บเพจในหน้าต่างเว็บเพจของหน้าต่างโปรแกรม Dream weaver ซึ่งคำสั่งเหล่านี้จะถูกจัดกลุ่มเป็นหมวดเป็นหมู่ของคำสั่งที่เกี่ยวข้อง เช่น File, Edit, View, Window และ Help เป็นต้น

3. หน้าต่างเว็บเพจ (Webpage Window) เป็นพื้นที่ทำงานสำหรับการตกแต่งและใส่ออบเจกต์ต่างๆ ลงไปในเว็บเพจ หน้าตาของเว็บเพจเมื่อดูจากบราวเซอร์หรือพิมพ์ออกเครื่องพิมพ์จะมีลักษณะเดียวกับการตกแต่งในพื้นที่ทำงานนี้ด้วย จึงเป็นที่มาของคำว่า WYSIWYG (What-You-See-Is-What-You-Get) สีฉากพื้นหลัง (Background Color) ของพื้นที่ทำงานปกติจะเป็นสีขาวสามารถเปลี่ยนได้ภายหลังที่หน้าต่าง Properties

4. หน้าต่างออบเจกต์ (Objects Palette) Dream weaver ได้เตรียมพร้อมหน้าต่างออบเจกต์ที่มีลักษณะเป็นปุ่ม ซึ่งรวบรวมเอาไว้เป็นหมวดหมู่ที่เกี่ยวข้องกัน สำหรับใส่ออบเจกต์ในเว็บเพจ เช่น ตาราง รูปภาพ ActiveX เลเซอร์ เป็นต้น เพื่อการตกแต่งและเพิ่มความสามารถให้กับเว็บเพจ โดยการคลิกที่ปุ่มในหน้าต่างออบเจกต์ที่ต้องการ จากนั้นคลิกแล้วลากเมาส์บน เว็บเพจ เพื่อนำไปตกแต่งเว็บเพจได้ตามต้องการ การเปิดหน้าต่างออบเจกต์โดยคลิกเมนู Window > Objects หรือกด <Ctrl+F2>

5. หน้าต่างพร็อพเพอร์ตี้ (Properties Window) สำหรับกำหนดลักษณะรูปแบบตัวอักษร ขนาด การจัดวาง สี สันของข้อความหรือขนาดของออบเจกต์ ที่ใส่ลงไปในเว็บเพจ โดยสามารถ

สร้างลิงค์เว็บเพจหรือไปยังเว็บไซต์อื่นๆ ได้อีกด้วย การเปิดหน้าต่างพร็อพเพอร์ตี้โดยคลิกเมนู Window > properties

6.Assets palette หน้าต่างเก็บรายชื่อของรูปภาพและส่วนประกอบอื่นๆ ทั้งหมดในเว็บที่เปิดดู การเปิด หน้าต่าง Assets โดยคลิกเมนู window > Library หรือกด <F11>

7.Behavior inspector หน้าต่างสำหรับเพิ่มลูกเล่นให้กับเว็บเพจด้วยการใส่ JavaScript เพื่อควบคุมการทำงานของออบเจกต์ต่างๆ ตามเหตุการณ์ที่เกิดขึ้น เช่น เมื่อคลิกเมาส์ให้เกิดการสลับภาพเป็นภาพอื่นๆ เมื่อเริ่มเปิดหน้าเว็บเพจด้วยบราวเซอร์ให้ปรากฏกรอบข่าวสารได้ เป็นต้น การเปิดหน้าต่าง Behaviors โดยการคลิกที่ปุ่ม Behaviors หรือ คลิกเมนู Window>Behaviors หรือ กด <Shift+F3>

8.Timeline inspector สำหรับการสร้างภาพเคลื่อนไหว โดยการกำหนดเส้นทางเดินของเลเยอร์ที่บรรจุออบเจกต์ได้ตามเวลาที่กำหนดไว้ภายในเว็บเพจเดียวกัน การเปิดหน้าต่าง Timeline โดยคลิกที่ปุ่ม Timeline หรือคลิกเมนู Window > Timelines หรือกด <Shift+F9>

9.Code inspector สำหรับตรวจสอบโค้ดภาษา HTML ที่ควบคุมการแสดงผลข้อมูลของเว็บเพจในเบื้องหลังนั่นเอง การเปิดหน้าต่าง HTML โดยคลิกที่ปุ่ม Code inspector หรือคลิกเมนู Window > หรือกด <F10> คุณจะสังเกตเห็น โค้ดภาษา HTML ทั้งหมดเพื่อศึกษาถึงการเขียน โค้ด HTML ดังกล่าว ออบเจกต์ใดที่กำลังถูกเลือกอยู่จะปรากฏเป็นแถบสีเลือกที่โค้ดในหน้าต่างโค้ด HTML นี้ด้วยเช่นกัน คุณสามารถใช้คำสั่ง Replace เพื่อแก้ไขโค้ดเหล่านี้ใหม่ได้ตามหลักการเขียนโค้ดภาษา HTML

การให้บริการอินเทอร์เน็ตไม่ว่าจะเป็น E-mail,FTP,Gopher,Telnet หรือบริการอื่นๆ ต้องใช้อุปกรณ์เชื่อมต่อภายในอันซบซ้อของฮาร์ดแวร์ ที่สามารถทำงานได้ด้วยโปรแกรมเฉพาะที่ทำงานบนอินเทอร์เน็ตเท่านั้น

WWW แบ่งออกเป็น 2 ส่วนคือ ส่วนที่เป็น Client และส่วนที่เป็น Server เหมือนกับที่มีแบ่งกันในระบบเครือข่ายทั่วไป ทั้งสองส่วนจะถูกเชื่อมโยงถึงกันผ่านทางอินเทอร์เน็ต โดยมี HTML เป็นส่วนขยายข้อมูลสำคัญ เมื่อเว็บบราวเซอร์ส่งข้อความร้องขอข้อมูลที่อยู่ในรูปแบบของไฟล์ HTML จากเครื่องคอมพิวเตอร์ที่ใช้งานอยู่ผ่าน โมเด็มหรืออุปกรณ์สื่อสารข้อมูลอื่นไปยังศูนย์บริการอินเทอร์เน็ต (ISP) ตามโพรโตคอล (Protocol) ที่กำหนดไว้ผ่านทาง URL (Uniform Resource Locators) และเมื่อข้อมูลเดินทางผ่านมาถึงเว็บเซิร์ฟเวอร์ ศูนย์บริการปลายทางที่ผู้ใช้ต้องการ ณ ที่นี้ เครื่องเว็บเซิร์ฟเวอร์ของศูนย์ จะทำการอ่านข้อมูลที่ถูกส่งมาและจะทำงานตามคำสั่ง

ที่กำหนดโดยอาจมีการเชื่อมโยงไปยังเว็บเบราว์เซอร์อื่นอีก หลังจบสิ้นกระบวนการแล้วจะทำการจัดส่งข้อมูลคำตอบย้อนกลับมายังเครื่องคอมพิวเตอร์ที่ใช้งานอยู่ โปรแกรมเว็บเบราว์เซอร์ที่เครื่องคอมพิวเตอร์จะแปลงสัญญาณคำสั่งและแสดงผลเป็นข้อความรูปภาพ และเสียง

ประเภทของการเชื่อมโยง HTML

แบ่งออกเป็น 2 ประเภท คือ

- 1) การเชื่อมโยงภายในเว็บไซต์
- 2) การเชื่อมโยงภายนอกเว็บไซต์

การเชื่อมโยงทั้ง 2 ประเภท ยังแบ่งออกเป็น 2 ชนิด คือ การเชื่อมโยงข้อมูลภายในแฟ้มเอกสารและการเชื่อมโยงต่างแฟ้มเอกสาร

การเชื่อมโยงภาพเคลื่อนไหวและเสียง ไฟล์รูปภาพแบบ GIF และ JPEG นั้นเมื่อปรากฏบนจอภาพจะแสดงผลเป็นเพียงภาพนิ่งเท่านั้น ถ้ามีไฟล์ข้อมูลที่มีคุณสมบัติแบบอื่น เช่น ไฟล์ภาพเคลื่อนไหว หรือไฟล์เสียง เมื่อต้องการนำมาประกอบการแสดงผลจะต้องทำอะไร

วิธีการนำมาใช้งานไม่แตกต่างไปจากการเชื่อมโยงรูปภาพเท่าใดนัก เพียงแต่เปลี่ยนจากไฟล์ GIF หรือ JPEG ไปเป็นไฟล์ภาพเคลื่อนไหวหรือไฟล์เสียงเท่านั้น สามารถเชื่อมโยงเว็บเพจกับไฟล์เหล่านี้ได้ แต่ทั้งนี้จะต้องขึ้นอยู่กับภายในเครื่องคอมพิวเตอร์ที่ใช้งานอยู่นั้น ได้ติดตั้งอุปกรณ์สำหรับการแสดงผลไฟล์แบบนั้นหรือไม่ เช่น ต้องการแสดงผลไฟล์ชนิด .MOV จะต้องติดตั้งไดรเวอร์สำหรับ Quick Time Movie หรือไฟล์ชนิด .MPEG จะติดตั้งไดรเวอร์สำหรับ .MPEG หรือมีการ์ด MPEG Decoder เพื่อช่วยการแสดงผล

ไฟล์ที่สามารถนำมาใช้งานกับเว็บเบราว์เซอร์

1) ไฟล์ประเภทที่สามารถใช้งานกับเว็บเบราว์เซอร์ได้โดยตรง ได้แก่ ไฟล์ที่ส่วนขยายเป็น TXT, HTML (HTM), GIF, XBM, JPEG (JPG) และ TIFF (TIF) ขึ้นอยู่กับชนิดของเว็บเบราว์เซอร์

2) ไฟล์ประเภทที่จะต้องมีอุปกรณ์เพิ่มเติมจึงสามารถทำงานได้ ได้แก่ ไฟล์ที่ส่วนขยายเป็น PS, AIFF, AU, WAV, MOV, MPEG, AVI และอื่นๆ

ข้อควรระวังในการสร้างเว็บเพจ (Web Pages)

1) เว็บเพจไม่ควรมีขนาดใหญ่เกินไปเนื่องจากเมื่อเว็บเพจมีขนาดใหญ่แล้วความเร็วในการเข้าถึงจะช้าตามไปด้วย ซึ่งสามารถแก้ไขดังนี้

- 1.1 ภาพกราฟฟิกขนาดเล็กแต่อยู่ในคุณภาพที่ยอมรับได้ เช่นการสร้างปุ่มบน Page อาจใช้สี 16 สีแทน 16 ล้านสี
- 1.2 เลือกใช้ไฟล์ภาพชนิดที่มีขนาดเล็กที่สุด เช่น .GIF หรือ .JPEG
- 1.3 ถ้าจำเป็นต้องใช้ภาพขนาดใหญ่ควรทำเป็น Link Pages แล้วเตือนผู้ใช้ก่อน ว่าต้องใช้เวลาโหลดนาน
- 1.4 หลีกเลี่ยงการทำ Image map (พื้นที่ภาพสัมพันธ์คำสั่ง) กับภาพขนาดใหญ่
- 1.5 ใช้ฟอร์แมตไฟล์ที่มีขนาดเล็กที่สุด (เช่น ใช้ไฟล์ MIDI แทน Waves)
ให้พอดีกับ Pages หากไม่พอควรแบ่งเป็น Pages ใหม่

2) ควรมีการทดลองใช้ก่อนเผยแพร่ทุกครั้ง โดยจะต้องทดลองใช้ดูก่อนหลายๆ ระบบใช้มอนิเตอร์หลาย ๆ ชนิด ความละเอียดสีหลาย ๆ ระดับ รวมทั้งการเข้าถึงผ่านโมเด็มที่มีความเร็วหลาย ๆ ระดับ เพื่อให้แน่ใจว่าไม่มีปัญหาในทุกเรื่อง

Personnel Home Page (PHP)

Personnal Home Page (PHP) เป็นโปรแกรมประเภท Server Side Scripting ประเภทหนึ่ง คล้ายๆกับ SSI และ ASP ซึ่งจะทำให้สร้างเว็บไซต์แบบไดนามิกได้ สามารถดึง ข้อมูลจากฐานข้อมูลได้ และการทำงานของ PHP จะเหมือนกับการรวมเอาความสามารถของภาษา C, Java, Perl เข้าไว้ด้วยกัน หรือกล่าวอีกนัยหนึ่ง PHP คือการนำเอาข้อดีของภาษา C, Java และ Perl มารวมกันนั่นเอง

ก. ความแตกต่างของ HTML, CGI และ PHP

PHP เป็นโปรแกรมประเภทหนึ่งทำงานบน Web Server (Server Side Script) คือ Server ให้บริการเว็บไซต์ มีหน้าที่ส่งข้อมูลเว็บไซต์ให้กับผู้เยี่ยมชมและใช้เป็นเครื่องคอมพิวเตอร์ที่ไשר์น Server Side Script เพื่อใช้ในการจัดการข้อมูล, ประมวลผลข้อมูลสำหรับเว็บไซต์ เหมือนกับ CGI (Common Gateway Interface) แต่การใช้ PHP จะมีความสามารถบางอย่างสูงกว่า เช่น สามารถใช้งานร่วมกับ HTML ได้ PHP คล้ายกับ ASP ที่เป็น Server Side Script ที่ทำงานบน Web Server ของ Window NT นั่นเอง

HTML คือไฟล์ .htm หรือ .html เก็บข้อมูลในโฮมเพจ เอกสาร HTML เป็นเอกสารที่ไม่มีข้อมูล Dynamic อะไร และใน HTML นี้เอง สามารถแทรกโค้ด Client Side Script เช่น

JavaScript ลงไปได้ ทำให้เกิดลูกเล่นต่างๆ บนเว็บ เช่น การแสดงรูปภาพสลับไปมา เป็นต้น
ตัว HTML นี้ปกติจะเก็บในเครื่อง Server รอเวลามีคนมาเปิดเว็บไซต์ Web Server จะส่งไฟล์นี้
ไปแสดงผลที่เบราว์เซอร์

CGI (Common Gateway Interface) เป็นโปรแกรมประเภทหนึ่งที่ทำหน้าที่ประมวลผลข้อมูลบนเว็บไซต์ ไม่ว่าจะเป็นการคำนวณ,การบันทึกจำนวนผู้เข้าชมเว็บไซต์ โดยมากจะใช้ CGI ในการเขียน Counter, Guestbook หรือ WebBoard ซึ่งการเขียน CGI นั้นเขียนได้หลายภาษา เช่น C , Delphi , Perl และอื่นๆ อีกมากมาย นามสกุล CGI นั้นขึ้นอยู่กับข้อกำหนดของ Web Server และตัวโปรแกรมที่ใช้พัฒนาเช่น .pl (Perl) , .asp (Active Server Pages) , .cgi , .exe เป็นต้น

PHP (Personnel Home Page) เป็นไฟล์นามสกุล .php3 หรือ .php ที่ทำหน้าที่เป็น CGI ตัวหนึ่งภายใน Web Server ภายในไฟล์นี้ส่วนมากจะเหมือนกับเอกสาร HTML เพราะสามารถเขียนแท็ก HTML ลงในโปรแกรม PHP ได้ และสามารถบรรจุแท็กซึ่งเป็นคำสั่งของ PHP ลงไปได้ โดยใส่เอาไว้ภายในเครื่องหมาย <?...?>

จากความหมายและหน้าที่ของ HTML , CGI และ PHP ในข้างต้น จะเห็นว่า PHP จะเป็นการรวมกันระหว่าง HTML และ Server Side Script ดังนั้น PHP คือ HTML ที่บรรจุคำสั่ง PHP ลงไปให้สามารถประมวลผลที่ Server ได้นั่นเอง เพราะฉะนั้น ถ้าดูภาพรวมๆ ของ PHP แล้ว PHP มีลักษณะของภาษาคู่กับ Perl และสามารถเขียนเป็น CGI ที่ใช้ได้ พร้อมกับความสามารถทางด้านการจัดการโปรแกรม โดยให้อยู่ภายในขอบเขตที่กำหนด เพราะฉะนั้นอาจจะกล่าวได้ว่า PHP จะเป็นการรวมความสามารถต่างๆ ดังนี้ PHP=(ข้อดีของภาษา Perl , C , Java)+ (ขอบเขตคำสั่งของ PHP)+ (HTML , JavaScript) จะเห็นว่าในส่วนของ PHP ได้รวมรูปแบบที่ดีทางไวยากรณ์ของภาษา Perl , C , Java ไว้ภายในขอบเขตของสัญลักษณ์ <?...?> เพื่อแบ่งแยกขอบเขตระหว่าง HTML หรือตัวโปรแกรมเลย ด้วยเหตุนี้เอง จึงสามารถแทรกโค้ด JavaScript หรือแทรกแท็ก HTML มาตรฐานได้ ในขณะเดียวกันสามารถสั่งให้โปรแกรมรันงานบางอย่างได้อีกด้วย

ข. Basic syntax

รูปแบบการใช้คำสั่ง PHP จะแตกต่างจาก CGI ตัวอื่นๆ เช่น Perl หรือ C เพราะว่า PHP ยอมให้แทรก คำสั่งต่างๆ ลงใน File html ได้เลย โดยสามารถแทรกเป็นระยะๆ ได้ และแปลง

นามสกุล File html จาก *.html เป็น *.php3 รูปแบบการแทรกคำสั่งดังกล่าวจะคล้ายคลึงกับ java script หรือ ASP

ตัวอย่างการแทรก script php

```
<HTML>
<HEAD><TITLE>การทดสอบการแทรก script php</TITLE></HEAD>
<BODY>
<H1><? echo "This is test script PHP" ?></H1>
</BODY>
</HTML>
```

ผลลัพธ์ที่ได้

This is test script PHP

เนื่องจาก PHP เป็นภาษาที่พัฒนาภายหลังภาษาอื่นๆ จึงพยายามเอาจุดเด่นของแต่ละภาษามาพัฒนาปรับปรุงใช้ใน PHP ดังนั้น PHP จึงมีความยืดหยุ่นสูง สามารถใช้รูปแบบการแทรกคำสั่ง ได้หลายแบบ เช่น

SGML Style

```
<? echo ("This is the PHP Command\n"); ?>
```

XML Style

```
<?php echo("This is the PHP Command\n"); ?>
```

Java Script Style

```
<script language="php">
echo ("This is the PHP Command\n");
</script>
```

ASP Style

```
<% echo ("This is the PHP Command\n"); %>
```

ก. Comments

การเขียนโปรแกรมที่มีความยาวมาก มีข้อมูลคำสั่งเป็นพันๆบรรทัด Programmer ไม่สามารถจำคำสั่งการทำงานของ Program ได้ทุกบรรทัด ดังนั้นจำเป็นต้องมีการ Comments เตือนความจำในบรรทัดที่ต้องการ และยังเป็นประโยชน์สำหรับ Programmer คนอื่นที่จะเข้ามาแก้ไขด้วย PHP มีได้รูปแบบการ Comments ไว้ให้เลือกใช้ตามความถนัดหลายรูปแบบ ผู้ใช้สามารถเลือกเอาแบบที่ตนเองถนัดได้ดังต่อไปนี้

ตัวอย่างการแทรก Comments

```
<?php
echo "Hi Hello Word"; // การ Comments แบบ C++
echo "Hi Hello Word"; /* การ Comments แบบ C */
echo "Hi Hello Word"; # การ Comments แบบ Perl
?>
```

ระบบฐานข้อมูลบนเว็บไซต์

ในการเขียนโปรแกรมบนเว็บแน่นอนจะต้องมีการเก็บข้อมูลบางอย่างเอาไว้ เพื่อนำไปใช้ต่อไปซึ่งการเขียนระบบฐานข้อมูลด้วยตนเองนั้น จะต้องออกแบบรูปแบบของการเก็บเอง และในการนำข้อมูลจากฐานข้อมูลไปใช้นั้นย่อมเกิดความผิดพลาดได้ ถ้าเขียนโปรแกรมไม่รัดกุมพอในการเขียนโปรแกรมบนเว็บในยุคแรก ๆ การเก็บข้อมูลนั้น โดยมากจะใช้ text file เช่น Editor หรือโปรแกรมประมวลผลข้อมูลในไฟล์ ที่ใช้ text file เป็นฐานข้อมูล การควบคุม text file นั้นลำบากกว่าการควบคุม Binary file ที่มี field และ record เข้ามาช่วยควบคุม และโอกาสในการเกิดข้อผิดพลาดในการควบคุม text file นั้นมากกว่า เมื่อเว็บไซต์เป็นแหล่งข้อมูลที่มีคุณค่าการใช้ระบบฐานข้อมูลเข้ามาช่วยจัดการจึงเป็นอีกทางเลือกหนึ่งที่ทำให้การบริการข้อมูลบนเว็บไซต์มีความสะดวกมากขึ้น และโอกาสผิดพลาดมีน้อยลง

โปรแกรมที่จะทำหน้าที่เป็นตัวกลางในการเรียกใช้ข้อมูลจากฐานข้อมูล และนำข้อมูลมาแสดงบนเว็บไซต์นั้น คือโปรแกรมประเภท CGI (Common Gateway Interface) โดยโปรแกรม

CGI นี้จะเป็นตัวที่คอยสื่อสารกับบราวเซอร์ ในการแสดงผลข้อมูลที่ได้รับมาจาก ฐานข้อมูลผ่านทาง API (Application Programming Interface) ของระบบฐานข้อมูลนั้นๆ ในการเขียนโปรแกรมให้เชื่อมต่อไปยัง API ของระบบฐานข้อมูลนั้น โดยจะต้องดูถึงระบบฐานข้อมูลด้วยว่าทำงานอยู่ในระบบใดถ้าเป็นฐานข้อมูลทำงานบนระบบ Windows NT จะต้องใช้โปรแกรมที่ชื่อ ASP (Active Server Page) ทำหน้าที่เป็น CGI ที่ใช้ในการเชื่อมต่อไปยังฐานข้อมูล แต่เป็นฐานข้อมูลทำงานบน UNIX เช่นพวก MYSQL , MSQL สามารถใช้โปรแกรมภาษา PHP หรือ Perl ในการเรียกใช้ข้อมูลจากฐานข้อมูลได้

MYSQL

MYSQL เป็นระบบฐานข้อมูลตัวหนึ่งที่มีประสิทธิภาพสูง ซึ่งนอกจาก MYSQL แล้ว PHP ยังสามารถเชื่อมต่อกับฐานข้อมูลได้หลายตัว เช่น Oracle , Sybase แต่ MYSQL นั้นเล็กและง่ายในการจัดการข้อมูลมาก เหมาะสำหรับการทำ Database Server ที่สามารถทำงานได้ในระดับหนึ่ง และรองรับทั้งบน Windows และ UNIX ไม่ว่าจะเป็ MySQL บน Linux ที่ติดตั้งได้ไม่ยาก ถึงแม้ว่า MYSQL จะทำงานได้ดีในระดับหนึ่งอาจเปรียบเทียบไม่ได้กับฐานข้อมูลที่ราคาแพงและมีขนาดโตกว่า

สิ่งที่จะต้องมีการติดตั้ง MYSQL ดังนี้

1) สำหรับ UNIX / Linux MYSQL Server มีทั้งแบบ GZ และแบบ RPM (Redhat Package Manager)

2) สำหรับ Windows NT / Me / 95 / 98 / 2000 จะต้องใช้โปรแกรม MYSQL for Win32

ก. การจัดการฐานข้อมูลโดยใช้ SQL

ใน 1 ฐานข้อมูล จะประกอบไปด้วยตารางได้หลายตาราง และภายในตารางจะประกอบด้วย field หลายๆ field ด้วยกัน เมื่อมีการตีตารางแล้ว จะเก็บข้อมูลลงไปได้ซึ่งขึ้นอยู่กับว่าออกแบบตารางเพื่อเก็บข้อมูลอะไร

สามารถเก็บข้อมูลลงใน MYSQL ได้โดยใช้ฐานข้อมูลตัวเดียวกันแต่เก็บอยู่คนละตาราง คือ อาจจะมีฐานข้อมูลชื่อ website data และภายในฐานข้อมูลนี้มีตาราง search_data เพื่อเก็บข้อมูลคนละอย่าง เพื่อใช้ในคนละงานกันได้การเข้าถึงข้อมูลในตารางนั้น จะทำได้โดยใช้ภาษา

SQL (Structured Query Language) โดยผ่านทาง MySQL shell หรือ API ของโปรแกรมที่รองรับกับฐานข้อมูลตัวนี้ เช่น

- (ก) ชุดคำสั่ง MySQL ของ PHP
- (ข) ชุดคำสั่ง MySQL สำหรับภาษา Perl
- (ค) MySQL API สำหรับ Delphi
- (ง) MySQL API สำหรับ C/C++

เพราะฉะนั้นการที่โปรแกรมภาษาต่าง ๆ จะเข้าถึงฐานข้อมูล MySQL ได้ นั้นจะต้องมี API หรือชุดคำสั่งที่รองรับ เช่น PHP มีฟังก์ชันที่ใช้ในการเข้าถึง MySQL อยู่แล้ว

ข. การ Create database

หลังจากที่เราได้ติดตั้ง MySQL บนระบบแล้ว และได้ทดสอบการเข้าและออกโปรแกรมแล้วว่า Program สามารถทำงานได้ปกติ

จะเห็นได้ว่าโครงสร้างของ Database ของ MySQL จะไม่ต่างจากโครงสร้าง database ของโปรแกรม database ตระกูลอื่น โดยเริ่มจาก สร้าง file database ขึ้นมาก่อน แล้วจึง สร้าง table โดย design ตามที่ต้องการ จากนั้นจะได้ field ต่างๆ เมื่อแต่ละ field มีข้อมูลจะถูกรวมเป็น record ต่อไป ดังนั้นจึงต้องสร้าง file database ก่อน

รูปแบบ

รูปแบบคำสั่ง การสร้าง file database CREATE

DATABASE databasename คำอธิบาย CREATE

DATABASE เป็นคำสั่งที่ใช้สร้าง file database

databasename เป็นชื่อ file database ที่เรากำหนดขึ้น

mysql>CREATE DATABASE datatest; จากคำสั่งเรา

ต้องการสร้าง file database ที่ชื่อว่า test

จากคำสั่งข้างต้น โปรดสังเกตเมื่อสิ้นสุดคำสั่งจะต้องมีเครื่องหมาย ";" ทุกครั้ง ถ้าไม่มีเครื่องหมาย ";" โปรแกรมจะตีความหมายว่ายังไม่รับคำสั่งจนกว่าจะพบเครื่องหมาย ";"

ค. ชนิดของข้อมูล

ตารางที่ 2.6 ชนิดของข้อมูล

ชนิดของข้อมูล(Data type) ได้แก่

Datatype	คำอธิบาย	ตัวอย่าง
CHAR(M)	จะเอาไว้เก็บข้อมูลที่เป็น string(สายอักขระ) โดยที่ขนาดของการเก็บมีความคงที่	firstname CHAR(25);
VARCHAR(M)	ข้อมูลที่เป็น string(สายอักขระ) โดยที่ขนาดของการเก็บมีความคงที่	firstname CHAR(25);
INT(M) Unsigned	INT เก็บค่าจำนวนเต็ม มีค่าตั้งแต่ -2147483648 ถึง +2147483647 แต่ถ้าใส่ Unsigned จะมีค่าได้ตั้งแต่ 0 ถึง 4294967295	light_year INT; electron INT unsigned;
FLOAT[(M,D)]	ใช้เก็บเลขทศนิยม เลข 4 และ 2 บอกว่าตัวแปรนี้เก็บค่าได้ 4 ตัว และเป็นเลขทศนิยม 2 ตำแหน่ง Note: 42.35 เก็บค่าได้ถูกต้อง 324.56 เก็บค่าเป็น 324.5 2.2 เก็บค่าได้ถูกต้อง	rainfall FLOAT(4,2);
DATE	เก็บข้อมูลในรูปแบบ "YYYY-MM-DD"	today DATE;
TEXT/BLOB	เก็บข้อมูลตั้งแต่ 255-65535 ตัวอักษร ข้อแตกต่างระหว่าง TEXT กับ BLOB คือ BLOB จะถือ cases sensitivity	comment BLOB;
SET	เป็นกลุ่มของข้อมูลที่ยอมให้เลือกได้ 1 ค่าหรือหลายๆ ค่า สามารถกำหนดได้ถึง 64 ค่า เราสามารถเลือกได้ เป็น "", "SUT", "MIT" หรือ "MIT,RMUT";	university SET("SUT", "MIT", "AIT", " RMUT");

ง. โครงสร้าง table

Table ทุก Table จำเป็นต้องมีโครงสร้าง โครงสร้าง Table จะบ่งบอกถึงลักษณะของการ Design และ ยังช่วยให้การนำเข้าข้อมูลถูกชนิดตามที่ Design เอาไว้ ถ้านำเข้าข้อมูล input ผิดแล้วจะส่งผลกระทบต่อ Database design รูปแบบโครงสร้างของ Table phonebook

รูปแบบ

รูปแบบคำสั่งการขอ ดู โครงสร้าง table

DESC tablename

คำอธิบาย

DESC tablename

DESC เป็นคำสั่งสร้างดูโครงสร้าง table

tablename เป็นชื่อ table ที่ต้องการดู

การติดต่อ file database ที่มี table อยู่ในนั้น ต้องขอ ดู table เป็นดังนี้

รูปแบบ

รูปแบบคำสั่งการขอ ดู table ใน file database

SHOW TABLES

คำอธิบาย

SHOW TABLES

SHOW เป็นคำสั่งของดู

TABLES เป็น table ที่ต้องการดูใน file database

จ. การ สร้าง table

ก่อนที่เราจะสร้าง table จะต้องสร้าง file database เสียก่อน จึงนำ file "datatest" มาใช้
อีกครั้งโดย ใช้คำสั่ง USE database

การสร้าง table มีรูปแบบคำสั่งดังนี้

รูปแบบ

รูปแบบคำสั่งสร้าง table CREATE TABLE data1 (field1, field2, field3,
fieldn) หรือ create_definition: col_name type [NOT NULL |
NULL] [DEFAULT default_value] [AUTO_INCREMENT][PRIMARY
KEY] [reference_definition] or PRIMARY KEY (index_col_name,...)
or KEY [index_name] (index_col_name,...) or INDEX [index_name]
(index_col_name,...) or UNIQUE [INDEX] [index_name]
(index_col_name,...) คำอธิบาย CREATE TABLE data1

CREATE TABLE เป็นคำสั่งสร้าง table data1 เป็นชื่อ table ที่ต้องการสร้าง
field1 เป็นชนิดของ column

ฉ. การแก้ไขข้อมูล

ในการดำเนินการจัดการข้อมูลจำเป็นจะต้องมีการแก้ไขข้อมูลให้ทันเหตุการณ์ (Up Date) อยู่เสมอ เช่น บริษัทมีการปรับอัตราเงินเดือนของบุคลากร เราในฐานะผู้ดูแลจัดการข้อมูล ต้องปรับข้อมูลให้ตรงกับความเป็นจริงในขณะนั้น ว่ามีใครได้ปรับเงินเดือนไปบ้าง การแก้ไขข้อมูล มีหลักการอยู่ว่า จะต้องค้นหาข้อมูลเก่าให้พบก่อนจากนั้นนำข้อมูลใหม่เข้าไปแก้ไข (Up Date) การแก้ไขข้อมูล Update มีรูปแบบคำสั่งดังต่อไปนี้

รูปแบบ

รูปแบบคำสั่งการแก้ไขข้อมูล

```
UPDATE tablename SET fieldname='datanew' WHERE fieldname='dataold';
```

คำอธิบาย

```
UPDATE tablename SET fieldname='datanew' WHERE fieldname='dataold';
```

UPDATE เป็นคำสั่งแก้ไขข้อมูล

tablename เป็นชื่อ table ที่ต้องการแก้ไขข้อมูลภายใน table

fieldname='datanew' เป็นชื่อ field และข้อมูลใหม่ที่ต้องการแก้ไขข้อมูล

WHERE คำสั่งค้นหาข้อมูล

fieldname='dataold; เป็นชื่อ field และข้อมูลเก่าที่ต้องถูกแก้ไข

ข. การลบข้อมูล

เมื่อแก้ไขข้อมูลได้แล้ว ทีนี้จะมาทำการลบข้อมูล การลบข้อมูลจำเป็นต้องใช้ความระมัดระวังอย่างสูง เนื่องจาก MYSQL จะไม่ทวนคำถามก่อนว่าจะยืนยันการลบหรือไม่ ข้อมูลที่ลบไปแล้วไม่สามารถ Undo กลับมาได้ ถ้าต้องการเรียกคืนจะต้องสร้างขึ้นใหม่

รูปแบบ

รูปแบบคำสั่งการลบข้อมูล

```
DELETE FROM tablename WHERE (fieldname='datadel');
```

คำอธิบาย

```
DELETE FROM tablename WHERE (fieldname='datadel');
```

DELETE FROM เป็นคำสั่งลบข้อมูล

tablename เป็นชื่อ table ที่ต้องการลบข้อมูลภายใน table

WHERE คำสั่งค้นหาข้อมูล

fieldname='datadel' เป็นชื่อ field และข้อมูลที่ต้องการลบ

ค. เรียงลำดับข้อมูล

การจัดการข้อมูลจำเป็นต้องมีการจัดเรียงลำดับข้อมูล เพื่อง่ายแก่การดูและตัดสินใจ เพื่อจะได้นำไปใช้ประโยชน์ ได้ตรงตามความต้องการ เช่น ฝ่ายบุคลากรต้องการทราบข้อมูลของพนักงานที่มีเงินเดือนเรียงลำดับจากมากไปหาน้อย เพื่อนำไปคำนวณหาฐานเงินเดือน การจัดลำดับข้อมูล Sort Data

รูปแบบ

รูปแบบคำสั่งจัดลำดับข้อมูล

```
SELECT fieldname FROM tablename ORDER BY fieldname;
```

คำอธิบาย

```
SELECT fieldname FROM tablename ORDER BY fieldname;
```

SELECT เป็นคำสั่งขอข้อมูล

fieldname เป็นคำสั่งเลือกดูข้อมูลใน field (ถ้าต้องการเลือกทุก field ให้ใช้เครื่องหมาย *)

ORDER BY คำสั่งจัดเรียงลำดับข้อมูล

fieldname เป็นชื่อ field ที่ต้องการเรียง

ฉ. การเพิ่มข้อมูล

จากตอนที่ผ่านมามีสามารถสร้าง File Database และ Table ตลอดจนแก้ไขและลบได้ แล้ว การนำข้อมูลเข้าสู่ Table โดยมีวิธีการ ดังนี้

รูปแบบ

รูปแบบคำสั่งการเพิ่มข้อมูลลงใน table

```
INSERT INTO tablename (field1,field2,fieldn) VALUES ('field1', 'field2', 'field2');
```

คำอธิบาย

```
INSERT INTO tablename ('field1', 'field2', 'fieldn');
```

INSERT INTO เป็นคำสั่งเพิ่มข้อมูลเข้าสู่ table

tablename เป็นชื่อ table ที่ต้องการเพิ่มข้อมูล

(field1,field2,fieldn) เป็นชื่อ field ที่ต้องการเพิ่มข้อมูล

VALUES กลุ่มของข้อมูลที่ต้องการเพิ่ม

('field1', 'field2', 'fieldn') เป็นข้อมูลของแต่ละ field ที่ต้องการเพิ่ม

field 1 เป็น field ที่ 1 ของข้อมูลที่ต้องการเพิ่ม (จากที่ได้ออกแบบ Table ไว้)

field 2 เป็น field ที่ 2 ของข้อมูลที่ต้องการเพิ่ม

field n เป็น field ที่ n ของข้อมูลที่ต้องการเพิ่ม

ญ. การสืบค้นข้อมูล

เมื่อเพิ่มข้อมูลเข้าไปเรื่อยๆ แล้วถ้าข้อมูลไม่มากนัก จะสามารถเรียกดูได้ไม่ยาก โดยการเรียกดูทั้งหมดและใช้ตาเลือกดูข้อมูลที่ต้องการ แต่ถ้าข้อมูลเพิ่มขึ้นเรื่อยๆ จาก 10 กว่า Record จนเป็น 100 และ เป็น 10000 ในที่สุด การสืบค้นข้อมูลโดยวิธีดังกล่าวไม่สามารถทำได้ MYSQL มีระบบการสืบค้นข้อมูลมาให้แล้ว สามารถสืบค้นโดยการสร้างเงื่อนไขให้ตรง ความต้องการ และได้ข้อมูลจากการสืบค้นตรงต่อความต้องการโดยไม่ยากเลย โดยมีวิธีการดังนี้

ก) การสืบค้นข้อมูลโดยไม่มีเงื่อนไข

- 1) ขอข้อมูลจากตารางทั้งหมด
- 2) ขอข้อมูลบางส่วนในตารางทุก Record

รูปแบบ

รูปแบบคำสั่งสืบค้นข้อมูลโดยขอข้อมูลทั้งหมด table

SELECT * FROM tablename;

รูปแบบคำสั่งสืบค้นข้อมูลโดยขอข้อมูลบางส่วน(บาง field)ทั้งหมด table

SELECT fieldname FROM tablename;

คำอธิบาย

SELECT fieldname FROM tablename;

SELECT คำสั่งเลือกข้อมูล

fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

tablename table ที่ต้องการสืบค้น

ข) การสืบค้นข้อมูลโดยมีเงื่อนไข (Operator)

การสืบค้นข้อมูลโดยมีเงื่อนไขจำเป็นต้องใช้ Operator เข้าช่วย การใช้ Operator สามารถใช้ได้ ในโอกาสและสถานการณ์ที่แตกต่างกัน สามารถแยกการใช้ได้ดังนี้

(ก) Operator

การจัดการข้อมูลจำเป็นต้องใช้ตัว Operator เช่นต้องการดูคนชื่อ ภาสกร ถ้าเป็นฐานข้อมูลขนาดใหญ่สัก 1 ล้านคน โปรแกรมจะค้นคนชื่อ ภาสกร ออกมาเต็มไปหมดเป็น 100 ชื่อ ข้อมูลนี้จึงถือได้ว่ามีความหมายน้อยมาก แต่ถ้ามีการกำหนดลงไปด้วยว่า คนชื่อภาสกร อยู่จังหวัดกาญจนบุรี การศึกษาปริญญาโท เงินเดือนมากกว่า 10000 บาท จะได้ข้อมูลที่มี ความหมายและตรงกับความต้องการมากขึ้น การกำหนดข้อสืบค้นต่างๆเหล่านี้ เรียกว่าเป็น การกำหนดอย่างมีเงื่อนไข และสิ่งที่ขาดไม่ได้ที่จะต้องนำมาใช้ประกอบการกำหนดการสืบค้นอย่างมีเงื่อนไข นี้คือตัว Operator นั่นเอง

ตัว Operator สามารถแยกได้ 2 ประเภทดังนี้

- (1) Relational operator
- (2) Bulletin operator

(1) Relational operator ได้แก่

ตารางที่ 2.7 Relational operator

Operator	ความหมาย
= > < >= <= <>	เท่ากับ มากกว่า น้อยกว่า มากกว่า หรือเท่ากับ น้อยกว่า หรือเท่ากับ ไม่เท่ากับ

(2) Bulletin operator ได้แก่

ตารางที่ 2.8 Bulletin operator

Operator	ความหมาย
AND OR NOT	และ หรือ เป็นเท็จ

การใช้ตัว Operator ช่วยในการจัดการข้อมูล (เพิ่ม สืบค้น แก้ไข ลบ) อย่างมีประสิทธิภาพถูกต้องแม่นยำและรวดเร็ว จำเป็นต้องกำหนดเงื่อนไขลงไปด้วย สมมุติว่าต้องการสืบค้นคนชื่อ ภาสกร การศึกษาปริญญาโท เงินเดือนมากกว่า 10000 บาท จากสถานการณ่เงินเดือนที่กำหนด ถ้าต้องการให้เป็นจริง ก็จะต้องมีข้อมูลทุกเงื่อนไข จะขาดเงื่อนไขใดเงื่อนไขหนึ่งไม่ได้ ตัว Operator Bulletin ที่ใช้คือ AND แต่ถ้าเรายอมให้เงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง ไม่จำเป็นต้องเป็นจริงทุกเงื่อนไข ต้องใช้ตัว Operator Bulletin "OR" โดยสรุปเป็นตารางดังนี้

ตารางที่ 2.9 Operator Bulletin "OR"

ความต้องการข้อมูล	สถานการณ์	การใช้ Operator	ผลลัพธ์	เงื่อนไข
ต้องการข้อมูล	สืบค้น ชื่อนายภาสกร	สืบค้น ชื่อนายภาสกร	นายภาสกร อยู่	ต้องเป็น

ครบ	อยู่จังหวัดกาญจนบุรี เงินเดือนมากกว่า 10000	อยู่จังหวัดกาญจนบุรี AND (เงินเดือน > 10000)	จังหวัดกาญจนบุรี เงินเดือนมากกว่า 10000	จริง ทั้ง สองอย่าง
ไม่ต้องครบก็ได้	สืบค้น ชื่อนายภาสกร อยู่จังหวัดกาญจนบุรี เงินเดือนมากกว่า 10000	สืบค้น ชื่อนายภาสกร อยู่ จังหวัดกาญจนบุรี OR (เงินเดือน > 10000)	นายภาสกร อาจจะอยู่ จังหวัดกาญจนบุรีก็ได้ เงินเดือนอาจจะ มากกว่า10000 ก็ได้	ต้องเป็น จริง อย่าง ใดอย่าง หนึ่ง

เงื่อนไขการใช้ Bulletin operator การค้นข้อมูลให้ได้ข้อมูลที่ถูกต้องและประสิทธิภาพ จำเป็นต้องทำความเข้าใจตรรกะเสียก่อน การอธิบายหลักการตรรกะ logic ในรูปตารางดังนี้

ตารางที่ 2.10 เงื่อนไขการใช้ Bulletin operator

Operator	เงื่อนไข	ตัวอย่าง	ผลลัพธ์
AND	T AND T	(1=1) AND (2=2)	T
	T AND F	(1>1) AND (2=2)	F
	F AND T	(2=2) AND (1=1)	F
	F AND F	(2<2) AND (1<>1)	F
OR	T OR T	(1=1) OR (2=2)	T
	T OR F	(1>1) OR (2=2)	T
	F OR F	(2=2) OR (1>1)	F
NOT	T F	1=1 2<>2	F T

F = เท็จ

T = จริง

Operator นอกจากจะเป็นตัวช่วยในการสืบค้นข้อมูลแล้ว ยังนำมาช่วยในการแก้ไขและลบข้อมูลได้อีกด้วย การนำตัว Operator ไปช่วยในงานจัดการข้อมูล

ก. การสืบค้นโดยใช้ Operator WHERE

รูปแบบ

รูปแบบคำสั่งสืบค้นข้อมูล โดยใช้ Operator WHERE

```
SELECT fieldname FROM tablename WHERE (fieldname='namerecord');
```

คำอธิบาย

```
SELECT fieldname FROM tablename WHERE (fieldname='namerecord');
```

SELECT คำสั่งเลือกข้อมูล

fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table ใดๆ

WHERE ข้อมูลที่ต้องการสืบค้น

(fieldname=namerecord) field และ Record ที่ต้องการ

ข. การสืบค้นโดยใช้ Operator LIKE

รูปแบบ

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator LIKE%

```
SELECT fieldname FROM tablename WHERE (fieldname LIKE 'namerecord%');
```

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator %LIKE%

```
SELECT fieldname FROM tablename WHERE (fieldname LIKE '%namerecord%');
```

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator %LIKE

```
SELECT fieldname FROM tablename WHERE (fieldname LIKE '%namerecord');
```

คำอธิบาย

SELECT คำสั่งเลือกข้อมูล

fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table ใดๆ

LIKE ข้อมูลคล้ายที่ต้องการสืบค้น

namerecord% คือ record ที่ต้องการสืบค้นขึ้นต้นด้วยตัวอักษรที่กำหนดและตามด้วยอักษรอะไรก็ได้ %namerecord% คือ record ที่ต้องการสืบค้นประกอบด้วยตัวอักษรอะไรก็ได้ ตาม

ด้วยที่กำหนด และตามด้วยอักขระอะไรก็ได้ %namerecord คือ record ที่ต้องการสืบค้นลงท้ายด้วยตัวอักขระอะไรก็ได้ และตามด้วยอักขระที่กำหนด

ค. การสืบค้นโดยใช้ Operator AND

รูปแบบ

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator AND

```
SELECT fieldname FROM tablename WHERE
(fieldname1='namerecord1')AND(fieldname3='namerecord2');
```

คำอธิบาย

```
SELECT fieldname FROM tablename WHERE
(fieldname1='namerecord1')AND(fieldname3='namerecord2');
```

SELECT คำสั่งเลือกข้อมูล

fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

WHERE ที่ที่ต้องการสืบค้นข้อมูล

(fieldname1=namerecord1) field และ Record ที่ต้องการสืบค้น

AND เงื่อนไขและ หมายความว่าต้องเป็นจริงทั้งสองเหตุการณ์จึงจะแสดงผล

(fieldname2=namerecord2) field และ Record ที่ต้องการสืบค้น

ง. การสืบค้นโดยใช้ Operator OR

รูปแบบ

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator OR

```
SELECT fieldname FROM tablename WHERE
(fieldname1='namerecord1') OR (fieldname3='namerecord2');
```

คำอธิบาย

```
SELECT fieldname FROM tablename WHERE
```

```
(fieldname1='namerecord1') OR (fieldname3='namerecord2');
```

SELECT คำสั่งเลือกข้อมูล

fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

WHERE ที่ที่ต้องการสืบค้นข้อมูล

(fieldname1=namerecord1) field และ Record ที่ต้องการสืบค้น

OR เงื่อนไขหรือ หมายความว่าต้องเป็นจริงเหตุการณ์ใดเหตุการณ์จึงจะแสดงผล

(fieldname2=namerecord2) field และ Record ที่ต้องการสืบค้น

จ. การสืบค้นโดยใช้ Operator >

ฉ. การสืบค้นโดยใช้ Operator >=

ช. การสืบค้นโดยใช้ Operator <

ซ. การสืบค้นโดยใช้ Operator <=

ฅ. การสืบค้นโดยใช้ Operator <>

ในส่วนนี้จะขอกล่าวพร้อมกันทีเดียว 5 Operator เพราะใช้หลักการเดียวกันหมด

รูปแบบ

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator >

SELECT fieldname FROM tablename WHERE fieldname > data;

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator > =

SELECT fieldname FROM tablename WHERE fieldname > = data;

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator <

SELECT fieldname FROM tablename WHERE fieldname < data;

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator < =

SELECT fieldname FROM tablename WHERE fieldname <= data;

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator >

SELECT fieldname FROM tablename WHERE fieldname <> data;

คำอธิบาย

SELECT fieldname FROM tablename WHERE fieldname > data;

SELECT คำสั่งเลือกข้อมูล

fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

WHERE ที่ที่ต้องการสืบค้นข้อมูล

fieldname > data field ที่ต้องการให้แสดงผลว่า มากกว่า ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

fieldname >= data field ที่ต้องการให้แสดงผลว่า มากกว่าเท่ากับ ข้อมูลที่กำหนด data คือข้อมูลที่ กำหนด

fieldname < data field ที่ต้องการให้แสดงผลว่า น้อยกว่า ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

fieldname <= data field ที่ต้องการให้แสดงผลว่า น้อยกว่าเท่ากับ ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

fieldname <> data field ที่ต้องการให้แสดงผลว่า ไม่เท่ากับ ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

ฎ. การ Import และ Export ข้อมูล

การจัดการข้อมูลจำเป็นจะต้องมีการโอนถ่ายและเชื่อมต่อฐานข้อมูลจากแหล่ง ข้อมูลอื่นๆ ในการโอนถ่ายการเชื่อมต่อนั้นมีหลายวิธี สำหรับวิธีที่เป็นกลางและสามารถใช้โอนถ่ายเชื่อมต่อฐานข้อมูลได้ทุกตระกูล โดยไม่มีข้อจำกัด คือการ Import และ Export ข้อมูลนั่นเอง และยังเป็นการ Backup ข้อมูลไปในตัวด้วย หลักการคือว่า Export เป็น text file จากนั้น Import เข้าฐานข้อมูลที่อื่น หรือ Import ข้อมูลจาก text file ซึ่งมาจากแหล่งฐานข้อมูลอื่นเข้ามาเป็นชนิดของ MySQL

ก) การ Export Data

รูปแบบ

รูปแบบคำสั่งการ Export ข้อมูล

```
fieldname FROM tablename INTO OUTFILE '/directory/outfile.txt';
```

คำอธิบาย

```
SELECT fieldname FROM tablename INTO OUTFILE '/directory/outfile.txt';
```

SELECT คำสั่งเลือกข้อมูลfieldname ชื่อ field ที่ต้องการ Export Data ถ้าต้องการ Export ทุก field ให้ใช้เครื่องหมาย "*"

FROM จากที่ไหน tablename ชื่อ table ที่ต้องการ Export

INTO OUTFILE คำสั่ง Export

/directory/ ชื่อ Directory ที่ต้องการ Export ข้อมูลไปไว้
outfile.txt ชื่อ file ที่จะ Export ออกไป

ข) การ Import Data

รูปแบบ

รูปแบบคำสั่งการ Import ข้อมูล

```
LOAD DATA INFILE "/directory/inputdata.txt"REPLACE/IGNORE INTO
TABLE tablename;
```

คำอธิบาย

```
LOAD DATA INFILE "/directory/inputdata.txt"REPLACE/IGNORE INTO
TABLE tablename;
```

LOAD DATA INFILE คำสั่ง Import ข้อมูล

/directory/inputdata.txt directory และ file ที่ต้องการ Import

REPLACE ให้แทนที่ข้อมูลเก่า

IGNORE ให้ข้ามข้อมูลเก่าโดยไม่แทนที่

INTO TABLE ให้ Import ไปไว้ที่ table

tablename ชื่อ table ที่ต้องการ Import ไปไว้

ฐ. การกำหนดผู้ใช้ฐานข้อมูล

เนื่องจาก MYSQL ทำงานภายใต้ Client Server สามารถทำงานได้ทั้งเครือข่าย Internet และ Intranet นั้นหมายความว่าผู้ใช้สามารถใช้ MYSQL จากที่ใดก็ได้ในโลกที่มี Internet อย่างไรก็ตามถ้ามีผู้ใช้ผู้อื่นที่ไม่พึงประสงค์ เข้ามาใช้ข้อมูลของเราหรือผู้บุกรุกระบบ (Hacker) จะสร้างความเสียหายแก่ข้อมูลของเราได้ ความจำเป็นของระบบป้องกันความปลอดภัย จึงต้องเข้ามาเกี่ยวข้องด้วย MYSQL ได้มีระบบดังกล่าวมาให้เรียบร้อยแล้ว มันสามารถกำหนดผู้ใช้ฐานข้อมูลตลอดจนสิทธิในการใช้ เช่นสามารถดูข้อมูลได้อย่างเดียว หรือสามารถดูแลและลบข้อมูลได้ เป็นต้น การกำหนดสิทธิ(ระดับการใช้ข้อมูล) และผู้ที่มีรูปแบบคำสั่งดังต่อไปนี้

รูปแบบ

รูปแบบคำสั่งกำหนดผู้ใช้ข้อมูล

```
GRANT all/select,insert,update,delete ON tablename TO username;
```

คำอธิบาย

```
GRANT all/select,insert,update,delete ON tablename TO username;
```

GRANT คำสั่งกำหนดสิทธิและผู้ใช้ข้อมูล

all ให้สิทธิทั้งหมด insert, select, update, delete

select,insert,update,delete สิทธิการใช้งานไม่ต้องกำหนดให้ทั้งหมดก็ได้ สามารถเลือกได้ตามความต้องการ

ON เป็นการกำหนดให้ table ชื่ออะไร

TO ให้แก่ใคร

username ชื่อของ user ที่ต้องการกำหนด

MYSQL ยังสามารถขอดู user ที่เราได้กำหนดไปแล้วได้ด้วย

รูปแบบ

รูปแบบคำสั่งขอข้อมูลผู้ใช้ข้อมูล

```
SELECT user();
```

คำอธิบาย

```
SELECT user();
```

SELECT ขอข้อมูล

user() ข้อมูล user ที่เรากำหนด

ท. การยกเลิกผู้ใช้ฐานข้อมูล

เมื่อเราสามารถกำหนดผู้ใช้ฐานข้อมูลได้แล้วเราต้องสามารถยกเลิกได้ด้วยการยกเลิกสิทธิ เราสามารถยกเลิกทั้งหมด หรือยกเลิกสิทธิเฉพาะอย่างได้ เช่น user ชื่อ jib มีสิทธิสามารถทำได้ทุกอย่าง select, insert, update, delete data ได้ ในขณะที่เดียวกันเราสามารถยกเลิกสิทธิ บางอย่างได้ เช่น ยกเลิก การ update และ delete data ได้เป็นต้น

รูปแบบ

รูปแบบคำสั่งจัดการกำหนดผู้ใช้ข้อมูล

REVOKE all/select,insert,update,delete ON tablename From username;

คำอธิบาย

REVOKE all/select,insert,update,delete ON tablename From username;

REVOKE คำสั่งยกเลิกการกำหนดสิทธิและผู้ใช้ข้อมูล

all ยกเลิกทั้งหมด

select,insert,update,delete ยกเลิกบางส่วน สามารถเลือกได้ตามความต้องการ

ON ยกเลิกที่ table ชื่ออะไร

From ยกเลิกผู้ใด

username ชื่อของ user ที่ต้องการยกเลิก

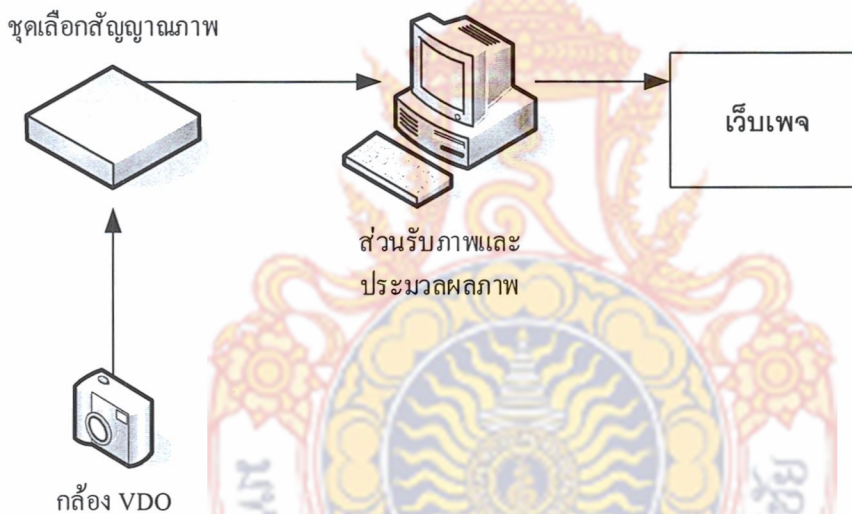


บทที่ 3

การออกแบบ

3.1 บทนำ

ในการสร้างระบบรักษาความปลอดภัยนั้น จุดที่สำคัญที่สุดจะอยู่ที่การตรวจจับการเปลี่ยนแปลงที่เกิดขึ้นในจุดที่ทำการตรวจสอบได้อย่างถูกต้องและแม่นยำ ซึ่งในจุดนี้จะต้องอาศัยการทำงานร่วมกันของส่วนต่าง ๆ หลายส่วน ได้แก่ กล้อง ชุดเลือกสัญญาณภาพ อุปกรณ์รับสัญญาณภาพ ชุดโปรแกรมสำหรับประมวลผลภาพ และส่วนของเว็บเพจ สำหรับรายงานผลให้กับผู้ใช้



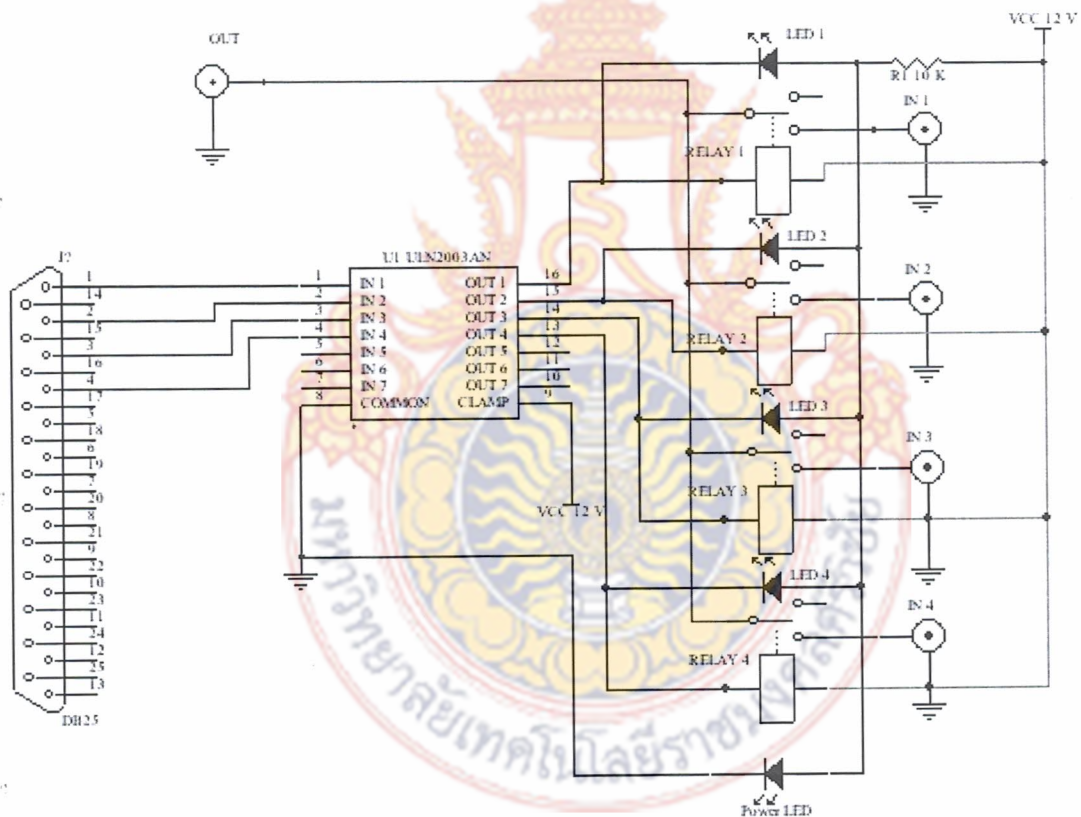
รูปที่ 3.1 แสดงการทำงานของระบบ

จากรูปที่ 3.1 จะเห็นได้ว่าการทำงานของระบบนั้นแบ่งออกได้เป็น 2 ส่วน คือ ส่วนของฮาร์ดแวร์ ซึ่งจะประกอบไปด้วย กล้อง ชุดเลือกสัญญาณ และส่วนของ Capture card และอีกส่วนก็จะเป็นส่วนของโปรแกรมซึ่งประกอบไปด้วย โปรแกรมประมวลผลภาพ และส่วนของเว็บเพจ บทนี้ก็จะอธิบายถึงการออกแบบในส่วนต่าง ๆ โดยอาศัยทฤษฎีในการออกแบบจากบทที่แล้ว ซึ่งมารายละเอียดดังนี้

3.2 การออกแบบชุดเลือกสัญญาณภาพ

3.2.1 แนวคิดและการออกแบบ

ชุดเลือกสัญญาณภาพ เป็นส่วนที่มีความสำคัญมากอีกส่วนหนึ่งซึ่งออกแบบมาเพื่อเพิ่มความสามารถของระบบรักษาความปลอดภัย ซึ่งปกติจะสามารถใช้งานได้เพียงจุดเดียวเนื่องจากข้อจำกัดของตัว Capture card ที่มีขายอยู่ทั่วไปสามารถรับสัญญาณภาพได้เพียงช่องเดียวเท่านั้น หากเปรียบเทียบกับการ์ดที่มีช่องรับสัญญาณหลายโมดูล ซึ่งมีราคาสูงกว่ามาก ดังนั้นจึงได้พัฒนาชุดเลือกสัญญาณขึ้นมา โดยใช้อุปกรณ์ที่สามารถหาได้ง่ายและวงจรที่ใช้ก็ไม่ซับซ้อนแต่สามารถทำงานได้ตามที่ต้องการ



รูปที่ 3.2 แสดงวงจรของชุดเลือกสัญญาณ 4 Input

จากรูปที่ 3.2 จะเห็นได้ว่า ภายในวงจรจะมีอุปกรณ์หลัก ๆ อยู่เพียงไม่กี่ชิ้น โดยตัวหลักของชุดสัญญาณจะอยู่ที่ IC เบอร์ ULN2003AN ซึ่งทำหน้าที่เป็น Driver เพื่อควบคุมการทำงานของชุด Relay ที่ทำหน้าที่เลือกสัญญาณภาพจากกล้องวิดีโอ และยังป้องกันความเสียหายของพอร์ตพรีนเตอร์ จากการดึงกระแสของอุปกรณ์ที่นำมาต่อ

3.2.2 การควบคุมการทำงาน

การควบคุมการทำงานของชุดเลือกสัญญาณภาพจะถูกควบคุมผ่านทางพอร์ตพรีนเตอร์ ส่วนชุดคำสั่งที่ใช้ในการควบคุมการทำงานของพอร์ตพรีนเตอร์นั้นจะใช้ไฟล์ไลบรารีที่ชื่อว่า inpout32.dll และ โค้ดที่เขียนด้วย Visual Basic 6

การทำงานของชุดเลือกสัญญาณจะต้องทำงานสัมพันธ์กับการแสดงผลบนหน้าจอของโปรแกรมรักษาความปลอดภัยซึ่งจะมีจอของช่องสัญญาณแต่ละช่องอยู่ซึ่งโค้ดที่ใช้ในการควบคุมการทำงานมีดังนี้

```
Private Sub Timer2_Timer()
    Image1(Ch).Visible = True
    Image1(Ch + 4).Visible = False
    Ch = Ch + 1
    If Ch > (Camera - 1) Then
        Ch = 0
        Och = 1
    Else
        Och = Och * 2
    End If
    Image1(Ch).Visible = False
    Image1(Ch + 4).Visible = True
    Lcount = 0
    mdTriger = 3000
    Out &H378, Och
    Debug.Print Ch & " - " & Och
End Sub
```

3.3 การออกแบบโปรแกรมระบบรักษาความปลอดภัย

การออกแบบในส่วนของโปรแกรมรักษาความปลอดภัยนั้นจะประกอบไปด้วยโปรแกรมส่วนหลักหลายๆส่วนรวมกัน ได้แก่

3.3.1 การรับภาพ

เป็นการทำงานในส่วนแรกของระบบ โดยในส่วนนี้จะเป็นการรับภาพจากชุดเลือกสัญญาณภาพเข้ามาทาง Capture card แล้วนำภาพที่ได้มาเก็บไว้ในหน่วยความจำของระบบ ก่อนที่จะนำไปประมวลผลต่อไป ในการติดต่อกับตัว Capture card จะติดต่อผ่านทางคอมโพเนนต์ที่ชื่อว่า ezVidc60.ocx ซึ่งเป็นคอมโพเนนต์ สำหรับติดต่อกับอุปกรณ์รับสัญญาณภาพ

```
If VidCap.CapSingleFrame Then
```

```
    MotionPic(Ch).Cls
```

```
    VidCap.SaveDIB VidCap.CaptureFile
```

```
    Set MotionPic(Ch).Picture = LoadPicture(VidCap.CaptureFile)
```

```
    .
```

```
    .
```

```
    .
```

```
Kill VidCap.CaptureFile
```

จากโค้ดจะเป็นการรับภาพเพื่อนำมาเก็บไว้ในหน่วยความจำเพื่อจะนำไปประมวลผลต่อไป หลังจากที่ได้นำภาพไปประมวลผลและแสดงผลแล้วก็จะทำการคืนค่าหน่วยความจำให้กับระบบ ในส่วนของขั้นตอนการเชื่อมต่อกับตัวการ์ดจะเป็นไปโดยอัตโนมัติ หากมีการติดตั้งตัวการ์ดแล้ว

3.3.2 การตรวจจับการเคลื่อนไหว

ส่วนนี้ถือเป็นส่วนหลักของโปรแกรมเลยทีเดียวเพราะจะเป็นส่วนที่ทำหน้าที่ในการประมวลผลภาพที่รับเข้ามาเพื่อทำการเปรียบเทียบ โดยการเปรียบเทียบนั้นจะทำการเปรียบเทียบค่าของสีในแต่ละจุดของภาพว่ามีส่วนใดที่เปลี่ยนแปลงไปจากเดิม จากนั้นก็เมื่อพบจุดที่มีการเปลี่ยนแปลงก็จะสร้างกรอบสี่เหลี่ยมขนาด 16 x 16 pixel ตรงบริเวณนั้น โดยโค้ดที่ใช้ในการตรวจจับการเปลี่ยนแปลงจะอยู่ในฟังก์ชัน GetMotion ในโปรแกรม

3.2.3 ส่วนของการเก็บข้อมูล

ในส่วนนี้จะเป็นการจัดเก็บข้อมูลและรูป เมื่อ โปรแกรมทำการตรวจจับได้แล้วเพื่อนำไปใช้ในการแสดงผลทางหน้าเว็บเพจ ในการจัดเก็บข้อมูลจะแบ่งย่อยได้อีก 2 ส่วน คือ การจัดเก็บรูป และการจัดเก็บข้อมูลของรูป โดยทั้ง 2 ส่วนจะทำงานต่อเนื่องกันตลอด จะทำแค่ส่วนใดส่วนหนึ่งไม่ได้ โดยแต่ละส่วนจะมีหลักการดังนี้

3.2.3.1 การจัดเก็บรูปภาพ

ในส่วนนี้จะเป็นส่วนที่นำเอาข้อมูลของภาพที่ผ่านการประมวลผลและตรวจสอบแล้วว่ามี ความแตกต่างมาจัดเก็บให้อยู่ในรูปของไฟล์ เพื่อนำไปใช้ในการแสดงผลต่อไป โดยในการจัดเก็บรูปภาพจะใช้ฟังก์ชัน savepicture ของ Visual Basic 6 ในการจัดเก็บ

```
Sub SavePic(chanel As Integer)
    On Error GoTo Cpath
    Dim PictureName As String
    Dim SavePath As String
    If Con.State Then
        PictureName = ImgName(chanel)
        SavePath = Mpath & Day(Date) & "_" & Month(Date) & "_" & Year(Date) &
            "\Camera_" & (chanel + 1) & "\"
        SavePicture MotionPic(chanel), SavePath & "\" & PictureName & ".jpg"
        SaveImg PictureName, chanel, ConvertPath(SavePath)
    Exit Sub
Cpath:
    MakeSureDirectoryPathExists (SavePath)
    If GetSetting(App.EXENAME, "Setting", "BkMode", "1") = 1 Then
        AutoBackup
    End If
    Resume
    End If
End Sub
```


จากโค้ดข้างต้นจะเป็นฟังก์ชันสำหรับบันทึกภาพที่ใช้จริงในโปรแกรม ซึ่งการทำงานโดยรวมมีดังนี้ ในขั้นแรกโปรแกรมจะทำการตรวจสอบการเชื่อมต่อกับฐานข้อมูลถ้าไม่สามารถเชื่อมต่อได้ก็จะหลุดออกจากฟังก์ชันทันที จากนั้นก็จะเป็นการกำหนดชื่อของไฟล์ และปลายทางที่ทำการจัดเก็บไฟล์รูป จากนั้นก็จะทำการบันทึกรูป ซึ่งในจุดนี้หาก path ที่กำหนดไม่มีอยู่ก็จะทำให้โปรแกรมเกิด Error ขึ้นมา จึงได้ใช้ฟังก์ชัน MakeSureDirectoryPathExists มาใช้แก้ปัญหาตรงจุดนี้ โดยหน้าที่ของฟังก์ชันนี้คือ สามารถสร้าง Directory และ sub directory ได้ในครั้งเดียว จากนั้นก็จะเก็บข้อมูลของรูปภาพลงในฐานข้อมูล.

3.2.3.1 การจัดเก็บข้อมูลลงในฐานข้อมูล

โดยขั้นตอนในการจัดเก็บจะเริ่มจาก ทำการกำหนดค่าต่างๆที่จำเป็นให้กับ Provider โดยการติดต่อกับฐานข้อมูลที่เป็น MySQL นั้นจะต้องใช้ Provider เฉพาะในการติดต่อ ซึ่งในที่นี้จะใช้ MySQL ODBC 3.51 Driver โดยมีลักษณะการใช้งานดังนี้

```
StrConn = "DRIVER={MySQL ODBC 3.51 Driver};" _
          & "SERVER=" & Fserver & ";" _
          & " DATABASE=" & Fdb & ";" _
          & "UID=" & Fuser & ";PWD=" & Fpasswd & "; OPTION=3;"
```

โดยโค้ดด้านบนจะเป็นโค้ดที่ใช้งานจริงโดยจะอยู่ใน โมดูลที่ชื่อ M_DataBase.bas โดยในการใช้งานจะต้องระบุ ชื่อของ Driver, Server ,Database,User และ password จากนั้นจึงเริ่มทำการเชื่อมต่อโดยใช้คำสั่งดังนี้

```
With Con
    If .State Then .Close
    .ConnectionString = StrConn
    .CursorLocation = adUseClient
    .CommandTimeout = 90
    .Open
End With
```

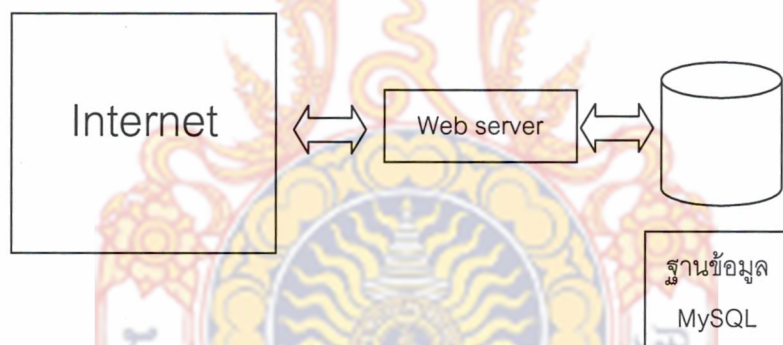
เมื่อทำการเชื่อมต่อได้แล้ว จึงจะสามารถทำงานต่างๆกับฐานข้อมูลได้ เมื่อใช้งานเสร็จแล้วก็ต้องทำการยกเลิกการเชื่อมต่อเพื่อคืนหน่วยความจำให้กับระบบโดยคำสั่งในการยกเลิกการเชื่อมต่อจะใช้ con.close

3.4 การออกแบบเว็บเพจ

สำหรับการออกแบบ เว็บจะแบ่งออกเป็นส่วนๆ ดังนี้

- ส่วนหน้าแรก homepage จะต้องน่าสนใจ
- ระบบต้องมีการล็อกอินเข้าสู่ระบบ เพื่อความปลอดภัย
- สามารถเลือกการแสดงผลในการดูแบบต่างๆ ได้เช่น รูปภาพที่บันทึกในแต่ละวันหรือดูสรุปเป็นรายเดือน
- สามารถค้นหาช่วงวันที่ต้องการได้

เนื่องจากระบบมีการบันทึกข้อมูลเป็นแบบรายวัน ฉะนั้นจึงจะต้องทำการออกแบบ โดยใช้ฐานข้อมูลเป็นตัวจัดเก็บข้อมูล โดยการเก็บแยกระหว่างรูปภาพกับรายละเอียดการบันทึกโดยใช้เก็บรายละเอียดการบันทึกในฐานข้อมูล



รูปที่ 3.3 โครงระบบการต่อเชื่อมเว็บเพจเข้าสู่ระบบ Internet

ลักษณะดังรูปที่ 3.3 เป็นวิธีการต่อเชื่อม ระบบการบันทึกข้อมูลระบบรักษาความปลอดภัย online เข้าสู่ระบบ internet โดยใช้ระบบฐานข้อมูล MySQL ในการบันทึกภาพใช้โปรแกรมที่เขียนขึ้นโดยโปรแกรม Visual Basic ติดต่อกับยังอุปกรณ์เลือกสัญญาณภาพ ผ่านทางพอร์ตขนาน(ปริ้นท์เตอร์)