

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมการเปิดปิดกล้องและการจับภาพ  
Camera switch controller and capturing picture



โดย

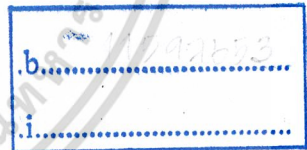
นางสาว พิชามณัฐ์ เศรษฐวิจิตร เลขประจำตัว 44010430

นางสาววิชุดา เศรษฐประเสริฐพร เลขประจำตัว 4401446

อาจารย์ที่ปรึกษา

รศ.ชนิษฐา แซ่ตั้ง

เลขหมู่.....  
เลขทะเบียน 61517  
วัน,เดือน,ปี.18 ก.ค. 2549



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิตสาขาวิชา

อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมการเปิดปิดกล้องและการจับภาพ  
Camera switch controller and capturing picture



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมการเปิดปิดกล้องและการจับภาพ

ผู้จัดทำ

1. นางสาว พิชามณัฐ เตชะวิจิตร
2. นางสาว วิชดา เตชะประเสริฐพร



อาจารย์ที่ปรึกษา

(รศ. วนิดา หงส์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ เรื่อง (ภาษาไทย) เครื่องควบคุมการเปิดปิดกล้องและการจับภาพ  
(ภาษาอังกฤษ) Camera switch controller and capturing picture

จัดทำโดย

1. นางสาว พิชามณัฐ์ เตชะวิจิตร
2. นางสาว วิชุดา เตชะประเสริฐพร

ปริญญานิพนธ์นี้ได้รับการตรวจสอบแล้ว พร้อมทั้งทำการสอบได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมการเปิดปิดกล้องและการจับภาพ  
Camera switch controller and capturing picture

นางสาวพิชามณูษ์ เศษะวิจิตร 44010430

นางสาววิชุดา เศษะประเสริฐพร 44010446

รศ.พนินฐา แซ่ตั้ง อาจารย์ที่ปรึกษา

ภาคการศึกษาที่ 2 ปีการศึกษา 2547

บทคัดย่อ

โครงการนี้เป็นเครื่องควบคุมกล้องในการจับภาพ ด้วยการปรับทิศทาง ด้วยสแต็ปปีงมอเตอร์ สองตัวทำหน้าที่ควบคุมการหมุนของกล้อง โดยที่ตัวหนึ่งควบคุมการเลื่อนตำแหน่งทางซ้ายหรือขวา ส่วนอีกตัวหนึ่งควบคุมการเลื่อนมุมเงยหรือมุมก้ม โดยสอดคล้องตามเมาส์ที่เลื่อนไปบนหน้าจอภาพ แล้วใช้ ไมโครคอนโทรลเลอร์รับข้อมูลจากพอร์ตอนุกรมของคอมพิวเตอร์ที่แสดงพิกัดการกวาดหน้ากล้องโดยเมาส์ มีตัวตรวจจับการบุกรุกเป็นวงจรรับส่งอินฟาเรดที่ทำหน้าที่ตรวจจับเพื่อที่จะเป็นตัวเปิดระบบ

Camera switch controller and capturing picture

Miss PICHAMON TEHAVIJIT

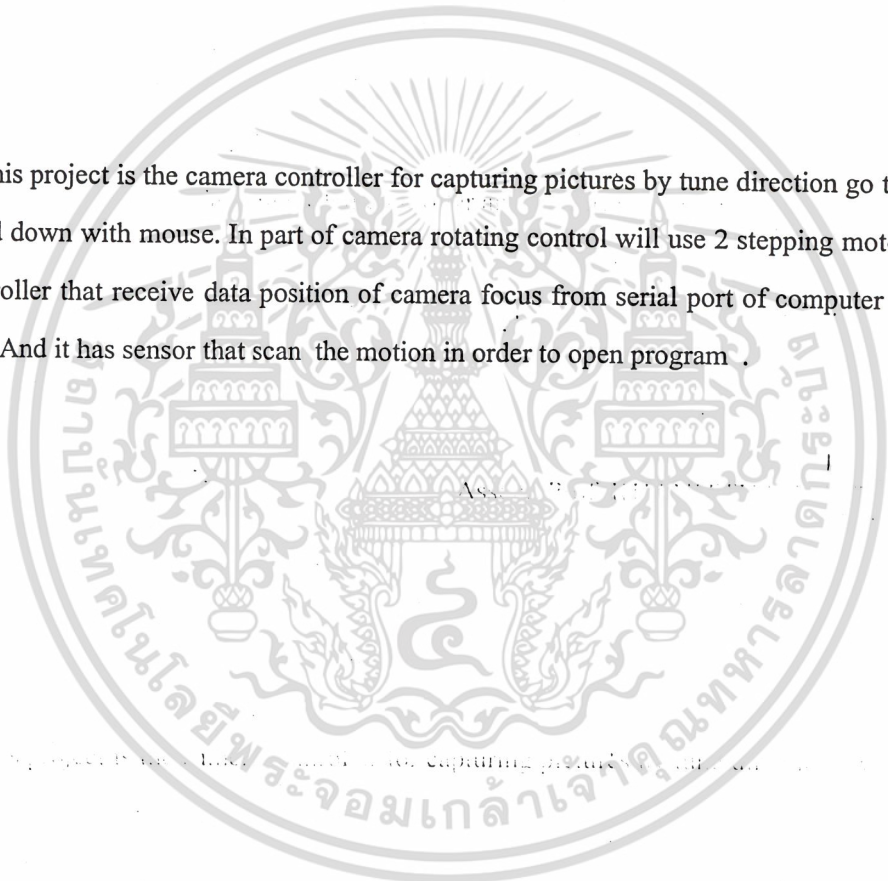
Miss WICHUDA TECHAPRESERTPORN

Associ. Prof. KHANITTHA SAETUNG (Adviser)

2<sup>nd</sup> Semester , Education Year 2004

Abstract

This project is the camera controller for capturing pictures by tune direction go to left or right and up and down with mouse. In part of camera rotating control will use 2 stepping motors with have microcontroller that receive data position of camera focus from serial port of computer commanded by mouse. And it has sensor that scan the motion in order to open program .



## สารบัญ

|   | หน้า |
|---|------|
| บทคัดย่อ  | I    |
| ABSTRACT  | II   |
| สารบัญ  | III  |
| สารบัญรูปภาพ  | IV   |
| บทที่ 1      บทนำ                                   | 1    |
| บทที่ 2      สตีปิ้งมอเตอร์                         |      |
| 2.1      ชนิดของสตีปิ้งมอเตอร์                      | 3    |
| 2.2      การทำงานของสตีปิ้งมอเตอร์                  | 3    |
| 2.3      การควบคุมการหมุนของสตีปิ้งมอเตอร์          | 6    |
| วงจรขับสตีปิ้งมอเตอร์                               | 8    |
| บทที่ 3      ไมโครคอนโทรลเลอร์และพอร์ตอนุกรม RS 232 |      |
| 3.1      ไมโครคอนโทรลเลอร์                          | 10   |
| 3.2      สื่อสารพอร์ตอนุกรม RS-232                  | 13   |
| บทที่ 4      Microsoft Visual C++ 6.0               | 14   |
| บทที่ 5      หลักการออกแบบและการสร้าง               | 24   |
| แผนภาพการทำงาน                                      | 28   |
| บทที่ 5      การทดสอบประสิทธิภาพ                    | 33   |
| บทที่ 6      สรุปผลการทดลอง                         | 36   |
| บรรณานุกรม  |      |
| กิตติกรรมประกาศ                                     |      |
| ภาคผนวก   |      |
| -      Program                                      |      |
| 5.1      Datasheet ไมโครคอนโทรลเลอร์ RS-232         |      |

## สารบัญรูปภาพ

|            | หน้า  |    |
|------------|---|----|
| รูปที่ 2.1 | ลักษณะการพันขดลวดบนสเตเตอร์   | 4  |
| รูปที่ 2.2 | ทิศทางการหมุน โรเตอร์ของสเต็ปป์มอเตอร์ 4 เฟส  | 4  |
| รูปที่ 2.3 | ไบโพลาร์สเต็ปมอเตอร์แบบ 2 เฟส สนามแม่เหล็กจะเปลี่ยน<br>เมื่อกลับทิศทางการไหลของกระแส                                    | 5  |
| รูปที่ 2.4 | ยูนิโพลาร์สเต็ปมอเตอร์การเปลี่ยนขั้วแม่เหล็กเกิดจากกระแส<br>ไหลต่างขดกัน กระแสจะ ไม่ไหลพร้อมกัน 2 ขด ในสเตเตอร์เดียวกัน | 6  |
| รูปที่ 2.5 | วงจรทดลองการทำงานของสเต็ปป์   | 6  |
| รูปที่ 2.6 | บล็อกไดอะแกรมควบคุมด้วยสเต็ปป์มอเตอร์   | 9  |
| รูปที่ 2.7 | การขับสเต็ปป์มอเตอร์แบบยูนิโพลาร์และไบโพลาร์  | 9  |
| รูปที่ 3.1 | แสดงสถาปัตยกรรมของ 89C51  | 10 |
| รูปที่ 3.2 | การต่อ Oscillator   | 13 |
| รูปที่ 3.3 | การส่งข้อมูลแบบอนุกรม   | 13 |
| รูปที่ 4.1 | ระดับชั้นคลาสของสถาปัตยกรรม   | 20 |
| รูปที่ 5.1 | แผนภาพการทำงาน  | 25 |
| รูปที่ 5.2 | ภาพของ โปรแกรมที่สร้างเพื่อใช้งาน   | 26 |
| รูปที่ 5.3 | วงจรชุดควบคุมกล้อง  | 30 |
| รูปที่ 5.4 | วงจรส่งสัญญาณ   | 32 |
| รูปที่ 5.5 | วงจรรับสัญญาณ   | 32 |
| รูปที่ 6.1 | โปรแกรมการจับภาพเปิด  | 33 |
| รูปที่ 6.2 | คลิกตำแหน่งที่ต้องการดู   | 33 |
| รูปที่ 6.3 | เลือกบันทึกภาพ  | 34 |
| รูปที่ 6.4 | คลิกที่ตำแหน่งใหม่  | 34 |
| รูปที่ 6.5 | สัญญาณในส่วนวงจรส่ง   | 35 |
| รูปที่ 6.6 | สัญญาณภาครับ ขณะไม่มีสิ่งกีดขวาง  | 35 |
| รูปที่ 6.7 | สัญญาณภาครับ ขณะมีสิ่งกีดขวาง   | 35 |

## บทที่ 1

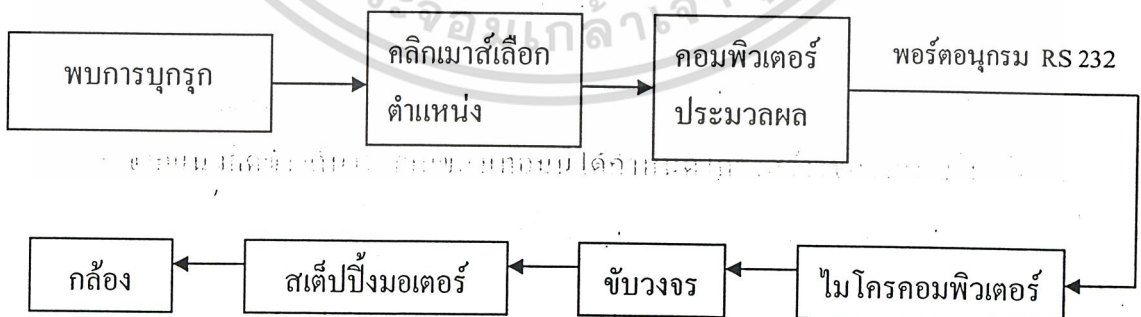
### บทนำ

ในยุคปัจจุบันนี้ ระบบรักษาความปลอดภัยเป็นสิ่งที่มีจำเป็นอย่างยิ่งสำหรับหน่วยงานต่างๆ เช่นการจดจำทะเบียนรถในลานจอดรถ การตรวจสอบอาวุธในการเข้าสถานที่ราชการ การสแกนลายนิ้วมือในการเข้าตึก รวมทั้งการรักษาเอกสาร และอื่นๆ อีกมากมาย จะเห็นได้ว่าทุกสิ่งทุกอย่างจำเป็นต้องมีระบบคอมพิวเตอร์เข้ามาเกี่ยวข้อง งานส่วนใหญ่ก็จะนำไมโครคอมพิวเตอร์เข้ามาเกี่ยวข้อง ณ ตอนนี้อุปกรณ์ราคาไมโครคอมพิวเตอร์ได้ลดลงอย่างมาก จนนำมาใช้กันอย่างแพร่หลาย นอกจากนี้อุปกรณ์ตรวจจับภาพก็ราคาไม่สูงเหมือนเมื่อก่อน ดังนั้นจึงนำมาสิ่งเหล่านี้มาประยุกต์ใช้เพื่อรักษาความปลอดภัย เนื่องจากในยุคนี้เรามีอุปกรณ์ตรวจจับความเคลื่อนไหว ซึ่งรูปแบบที่จะให้แจ้งว่ามีสิ่งผิดปกติก็มีได้หลายแบบเช่นส่งเสียงดัง เปิดไฟให้สว่าง การเฝ้าดูเราก็ไม่จำเป็นต้องนั่งเฝ้าตลอดทั้งวัน ดังนั้นเราจึงมีเวลาทำอะไรอย่างอื่น ได้อีกมากมาย

#### ลักษณะของโครงการ

จากแนวคิดข้างต้นในส่วนของเทอมนี้ได้กำหนดโครงสร้างของระบบในเบื้องต้นดังนี้ กล้องจะถูกติดตั้งอยู่บนสเต็ปปีงมอเตอร์ซึ่งมีอยู่ 2 ตัวในการควบคุมทิศทางขึ้น-ลง และซ้าย-ขวา โดยการสั่งงานจากการคลิกเมาส์ที่หน้าจอคอมพิวเตอร์ เมื่อตรวจจับแล้วพบว่ามีการบุกรุก โปรแกรมจะถูกเปิดขึ้นเพื่อจับภาพโดยอัตโนมัติ ซึ่งเราเลือกที่จะจับได้ และสามารถเลือกที่จะบันทึกภาพได้ด้วย

.....การทำงานจะแสดงดังแผนภาพอย่างคร่าวๆ ได้ดังนี้.....



แผนภาพการทำงาน

.....การทำงานจะแสดงดังแผนภาพอย่างคร่าวๆ ได้ดังนี้.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื้อหาในรายงานเล่มนี้จะแบ่งได้เป็นส่วน ๆ ดังนี้

- ส่วนที่ 1    ตัวตรวจจับความเคลื่อนไหว ซึ่งเป็นวงจรรับ-ส่ง อินฟราเรด ถ้าพบว่ามีผู้บุกรุกเกิดขึ้น  
เปิดโปรแกรมจับภาพโดยอัตโนมัติ
- ส่วนที่ 2    คอมพิวเตอร์ ซึ่งภายในจะบรรจุ โปรแกรมการหาพิกัดของเมาส์ที่ต้องการจับภาพโดย  
จะส่งค่าผ่านทางพอร์ตอนุกรม
- ส่วนที่ 3    ส่วนที่รับค่าจากพอร์ตมาคำนวณหาทิศทางที่เมาส์ต้องการเพื่อที่จะ ไปควบคุมทิศทาง  
ของกล้องว่าจะให้หมุนขึ้น-ลง และซ้าย-ขวา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2 สเต็ปป์มอเตอร์

สเต็ปป์มอเตอร์เป็นมอเตอร์ประเภทหนึ่งที่ถูกนำไปใช้งานมาก ไม่ว่าจะเป็นส่วนประกอบในหุ่นยนต์ เครื่องจักรกลที่ควบคุมด้วยคอมพิวเตอร์ ต่างก็ใช้สเต็ปมอเตอร์เป็นตัวขับเคลื่อน หัวข้อต่างๆ ที่จะกล่าวก็จะประกอบไปด้วยการทำงานของสเต็ปป์มอเตอร์ ตลอดจนการควบคุมการหมุนและวงจรอิเล็กทรอนิกส์ที่นำมาขับสเต็ปป์มอเตอร์

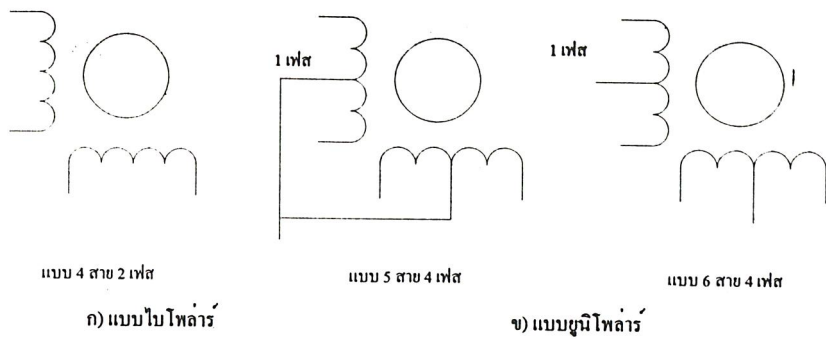
### 2.1 ชนิดของสเต็ปป์มอเตอร์

การแบ่งชนิดของสเต็ปป์มอเตอร์แบ่งได้เป็น 6 ชนิดตามโครงสร้างพื้นฐานหรือความแตกต่างของรูปแบบโรเตอร์ ดังนี้

1. ชนิดวาริเอเบิลรีลักแตนซ์ ( Variable Reluctance : VR)
2. ชนิดไฮบริด (Hybrid)
3. ชนิดเพอร์มาเนนต์แม็กเน็ต (Permanent)
4. ชนิดแรเอิร์ธเพอร์มาเนนต์แม็กเน็ต ( Rare Earth Permanent Magnet)
5. ชนิดไบโพลาร์
6. ชนิดยูนิโพลาร์

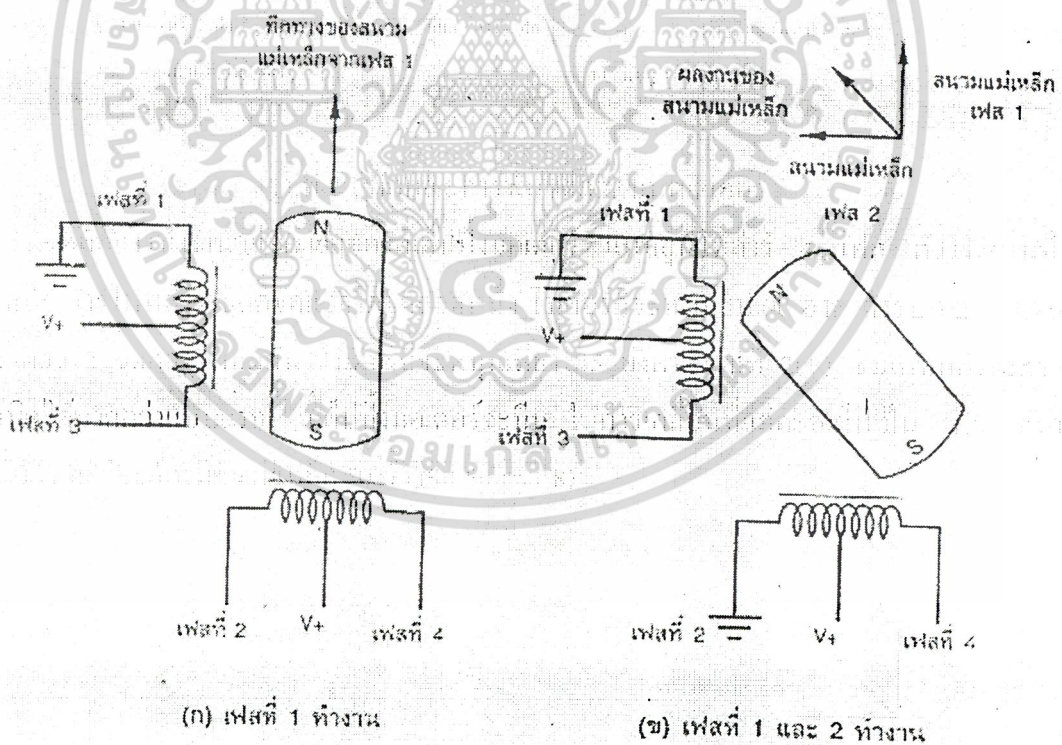
### 2.2 การทำงานของสเต็ปป์มอเตอร์

การทำงานของสเต็ปป์มอเตอร์จะมีความแตกต่างจากมอเตอร์ทั่วไป คือเมื่อป้อนกำลังไฟให้ตัวมัน ก็จะหมุนตามจังหวะของสัญญาณ ไฟฟ้าที่ป้อน สเต็ปป์มอเตอร์สามารถกำหนดตำแหน่งการหมุนได้อย่างละเอียด โดยใช้คอมพิวเตอร์เป็นตัวกำหนดและจัดเก็บตัวเลขของตำแหน่งเหล่านั้นไว้



รูปที่ 2.1 ลักษณะการพันขดลวดบนสเตเตอร์

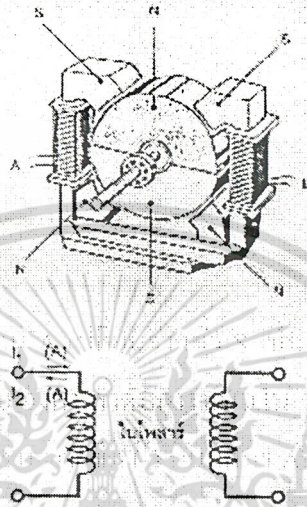
หลักการทํางานอย่างง่ายของสเต็ปปีงมอเตอร์ แบบยูนิโพลาร์ 4 เฟส ตัวโรเตอร์เป็นแม่เหล็ก การให้พลังงานแก่ขดลวดใดขดลวดหนึ่ง โรเตอร์ก็จะหมุนไป 90 องศา ดังรูปที่ 2(ก) แต่ถ้าให้ที่เดียว 2 ขดลวดพร้อมกัน โรเตอร์ก็จะหมุนเพียง 45 องศา ดังรูปที่ 2(ข) ซึ่งแบบหลังจะสร้างแรงบิดได้มากกว่าแบบแรก สเต็ปปีงมอเตอร์จะมีมุมในการเคลื่อนที่แต่ละสเต็ปเป็น 1.8 องศา ดังนั้น โรเตอร์จะต้องมีขั้วแม่เหล็ก 50 ขั้ว ( $90/50 = 1.8$ )



รูปที่ 2.2 ทิศทางการหมุนโรเตอร์ของสเต็ปปีงมอเตอร์ 4 เฟส

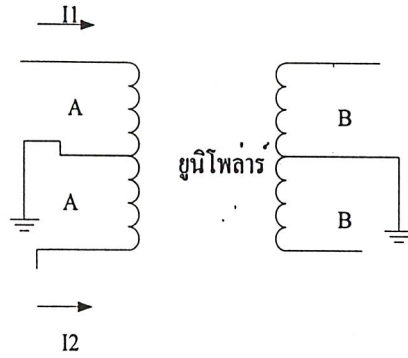
ส่วนรูปที่ 2.3 เป็นการพันขดลวดแบบไบโพลาร์ เมื่อขดลวด A และ B มีกระแสไหลผ่าน เป็นผลให้โรเตอร์ที่มีขั้วแม่เหล็กคํงกับสเตเตอร์ถูกดูดต่อมาเมื่อมีกระแสไหลในขดลวด A เปลี่ยนเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทิศทางการไหลของกระแสไฟฟ้าในขั้วแม่เหล็กที่แกน A เปลี่ยนจากขั้ว S เป็นขั้ว N จากขั้ว N เป็นขั้ว S สลับกัน  
 โรเตอร์จึงผลักให้หมุนทวนเข็มนาฬิกา 90 องศา ลำดับการหมุนใน 1 รอบเป็นสเต็ปดังนี้ AB ,AB ,AB ,  
 AB ,AB มี 4 สเต็ปๆละ 90 องศา เครื่องหมายขีด (-) บนอักษร A,B แทนการกลับขั้ว



รูปที่ 2.3 ไบโพลาร์สเต็ปมอเตอร์แบบ 2 เฟส สนามแม่เหล็กจะเปลี่ยนเมื่อกลับทิศทางการไหลของกระแส

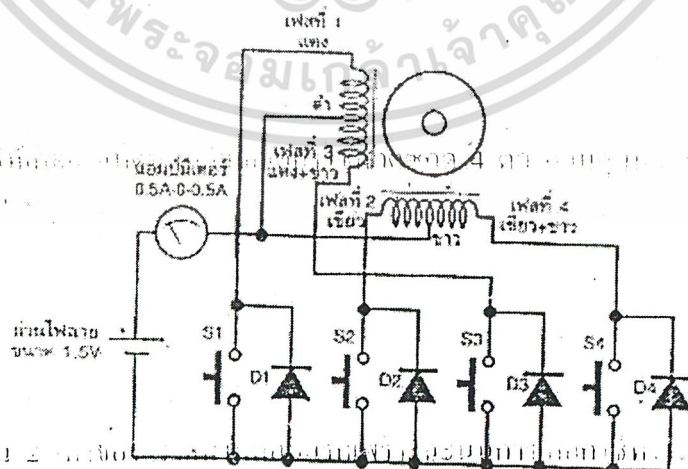
จะสังเกตได้ว่าเมื่อกลับขั้วแม่เหล็กในแต่ละเฟสจะต้องมีการหยุดกระแสก่อนแล้วกระแสจึงค่อยเปลี่ยนทิศทาง จึงสรุปเป็นสเต็ปได้ดังนี้ AB ,B ,AB ,A ,B ,AB ,A ,AB การทำงานเป็นแบบครึ่งสเต็ปนี้ เป็นผลให้ค่าโมเมนต์มีค่าน้อยกว่าปรกติเพราะมีช่วงเวลาที่กระแสไหลแค่เฟสเดียว ส่วนแบบยูนิโพลาร์ก็คล้ายกับแบบไบโพลาร์โดยคิดขดเดียวในแต่ละเฟสของยูนิโพลาร์จะมีแทปกกลาง ซึ่งจะแบ่งเป็น 2 ขด ดังรูปที่ 4 เป็นผลให้ค่าฟลักซ์แม่เหล็กมีค่าน้อยกว่าไบโพลาร์ ดังนั้นเมื่อสนามแม่เหล็กเปลี่ยนแปลงกระแสจะไม่เปลี่ยนทิศทางการไหล สนามแม่เหล็กที่ได้ก็น้อยตามแรงบิดที่ขึ้นกับสนามแม่เหล็กก็น้อยกว่าด้วย



รูปที่ 2.4 ยูนิโพลาร์สเต็ปมอเตอร์การเปลี่ยนขั้วแม่เหล็กเกิดจากกระแสไหลต่างขดกัน กระแสจะไม่ไหลพร้อมกัน 2 ขด ในสเตเตอร์เดียวกัน

2.3 การควบคุมการหมุนของสเต็ปมอเตอร์

วิธีศึกษาการทำงานของสเต็ปมอเตอร์ มีวิธีง่ายๆ ดังรูปที่ 5 โดยป้อนแรงดัน ขนาด 1.5 โวลต์ มาใช้ขับให้เกิดแรงบิดขึ้นที่ตัวมอเตอร์ใช้สวิทซ์กัก 4 ตัว ควบคุมมอเตอร์ให้ทำงานแบบเวฟหรือครึ่งสเต็ป ไดโอด D1-D4 ป้องกันการสปาร์คที่หน้าคอนแทคสวิทซ์และแรงดันย้อนกลับ สมมติว่าขดลวดเฟสที่ 1 สวิทซ์ S1 ปิดวงจร จะมีกระแสไหลผ่านขดลวดเฟสที่ 1 ลงกราวด์ และเมื่อสวิทซ์ S1 เปิดวงจรก็จะทำให้เกิดกระแสไหลผ่านทางขดลวดเฟสที่ 3 โดยผ่านทางไดโอด D3 ในขณะเดียวกันก็จะมีแรงดันตกคร่อมขดลวด 2 ขณะเปิดวงจร (เป็นแรงดันตกคร่อมไดโอด) ซึ่งจะมีขนาดแรงดันเป็น 2 เท่าของแรงดันในตอนแรกเพราะฉะนั้นการเลือกใช้ทรานซิสเตอร์ต้องคำนึงถึงข้อนี้ด้วยเมื่อนำมาใช้ในการขับมอเตอร์แบบยูนิโพลาร์



รูปที่ 2.5 วงจรทดลองการทำงานของสเต็ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกระตุ้นเพื่อควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปในการใช้งานจริงทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนแบบซีควเอนเชียลในรูปแบบที่ถูกต้อง แบ่งเป็น 3 รูปแบบ คือ

#### แบบเวฟ (Wave)

เป็นการกระตุ้นรูปแบบที่ง่ายที่สุด โดยกระตุ้นขดลวดที่ละขดในเวลาหนึ่งและเรียงถัดกันไป เช่น ขดที่ 1,2,3,4,1 หรือ 1,4,3,2,1 ขึ้นอยู่กับทิศทางที่ต้องการให้หมุน ดังนั้นจึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกระตุ้นเท่านั้น วงจรกระตุ้นแบบเวฟจึงมีราคาถูกและง่าย ขั้นตอนการทำงานต่างๆ แสดงดังตารางที่ 1

| สเต็ปที่ | เฟส 1 | เฟส 2 | เฟส 3 | เฟส 4 |
|----------|-------|-------|-------|-------|
| 1        | ทำงาน | -     | -     | -     |
| 2        | -     | ทำงาน | -     | -     |
| 3        | -     | -     | ทำงาน | -     |
| 4        | -     | -     | -     | ทำงาน |

ตารางที่ 1 แสดงการกระตุ้นขดลวดแต่ละเฟสแบบเวฟ

#### แบบ 2 เฟส (Two Phase)

เป็นการกระตุ้นอีกรูปแบบหนึ่งซึ่งคล้ายกับแบบเวฟ แต่จ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และถัดเรียงไปเช่นเดียวกับแบบเวฟคือขดลวดที่ 1,2,3,4,1,2 หรือ 1,4,3,2,1,4 ขึ้นอยู่กับทิศทางที่ต้องการให้หมุนการเพิ่มจำนวนของขดลวดที่ถูกกระตุ้นนี้ทำให้เพิ่มแรงบิดได้มากกว่าแบบเวฟ แต่ต้องใช้แหล่งจ่ายกำลังไฟฟ้ามากขึ้น การทำงานแสดงดังตารางที่ 2

| สเต็ปที่ | เฟส 1 | เฟส 2 | เฟส 3 | เฟส 4 |
|----------|-------|-------|-------|-------|
| 1        | ทำงาน | ทำงาน | -     | -     |
| 2        | -     | ทำงาน | ทำงาน | -     |
| 3        | -     | -     | ทำงาน | ทำงาน |
| 4        | ทำงาน | -     | -     | ทำงาน |

ตารางที่ 2 แสดงการกระตุ้นขดลวดแต่ละเฟสแบบ 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา หรือทำซ้ำหรือส่งต่อเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แบบครึ่งสเต็ป (Half Step)

เป็นการผสมผสานกันระหว่างแบบเวฟและแบบ 2 เฟสเพื่อเพิ่มจำนวนสเต็ปต่อรอบอีกหนึ่งเท่าตัว จะกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ ขดลวด 1,12,23,3,34,4,41,1 หรือในการหมุนอีกทิศทางหนึ่งจะได้เป็น 1,14,4,43,3,32,2,21,1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มขึ้นมากอีกเพราะช่วงสเต็ปมีระยะสั้นลงและแต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขด ที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องระวังไว้ว่าเมื่อกระตุ้นให้ทำงานในรูปแบบนี้ จะต้องทำการหมุนถึง 2 สเต็ป จึงจะได้เท่ากับ 1 สเต็ป เต็มเหมือนกับในการควบคุม 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้เทียบเท่ากับแบบ 2 เฟสแรก จึงจะเพียงพอ ขั้นตอนการทำงานต่างๆ แสดงดังตารางที่ 3

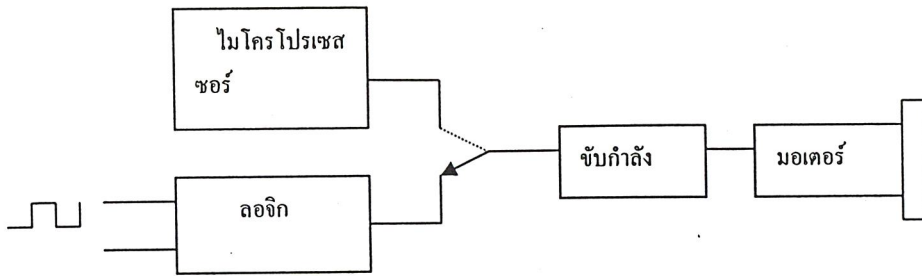
| สเต็ปที่ | เฟส 1 | เฟส 2 | เฟส 3 | เฟส 4 |
|----------|-------|-------|-------|-------|
| 1        | ทำงาน | -     | -     | -     |
| 2        | ทำงาน | ทำงาน | -     | -     |
| 3        | -     | ทำงาน | -     | -     |
| 4        | -     | ทำงาน | ทำงาน | -     |
| 5        | -     | -     | ทำงาน | -     |
| 6        | -     | -     | ทำงาน | ทำงาน |
| 7        | -     | -     | -     | ทำงาน |
| 8        | ทำงาน | -     | -     | ทำงาน |

ตารางที่ 3 แสดงการกระตุ้นขดลวดแต่ละเฟสแบบครึ่งสเต็ป

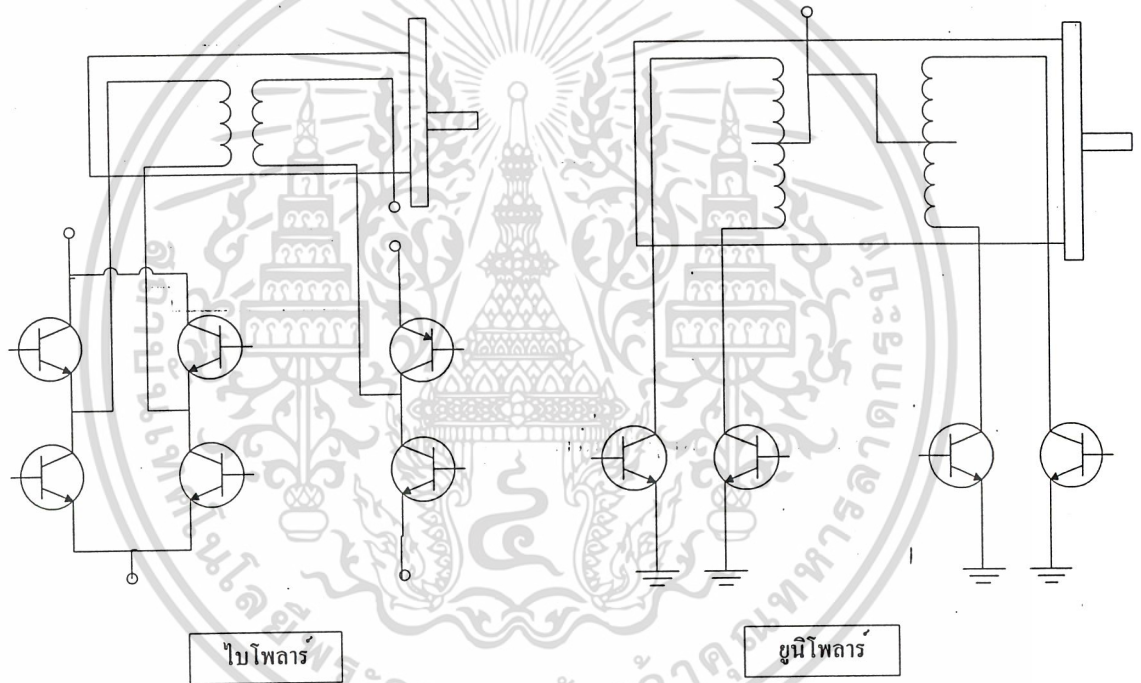
### วงจรขับสเต็ปปึงมอเตอร์

การนำเอาวงจรอิเล็กทรอนิกส์มาใช้กับสเต็ปปึงมอเตอร์แสดงดังบล็อกไดอะแกรมรูปที่ 6 การขับด้วยวงจรอิเล็กทรอนิกส์นี้ก็ต้องแยกด้วยว่าขับเคลื่อนแบบไบโพลาร์หรือยูนิโพลาร์ อีกทั้งยังขึ้นอยู่กับเฟสของมอเตอร์ด้วยว่ามีจำนวนเฟสมากน้อยแค่ไหนที่ต้องควบคุมวงจรขับแบบยูนิโพลาร์ นั้นแสดงไว้ในรูปที่ 7 จากวงจรจะเห็นได้ว่าจะใช้ทรานซิสเตอร์ขับขดลวดมอเตอร์ในลักษณะ 1 ขดต่อทรานซิสเตอร์ 1 ตัว ซึ่งจะแตกต่างจากการขับแบบไบโพลาร์ ขดลวดทางซ้ายมือต้องใช้ทรานซิสเตอร์ถึง 4 ตัว ต่อวงจรแบบบริดจ์มาขับขดลวด 1 ขด ส่วนขดลวดทางขวามือนั้นใช้ทรานซิสเตอร์ 2 ตัว มาขับขดลวด 1 ขด ดังรูปที่ 7 (ข) แต่แหล่งจ่ายไฟที่จ่ายให้กับทรานซิสเตอร์ 2 ชุดจะไม่เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 บล็อกไดอะแกรมควบคุมด้วยสแต็ปปีงมอเตอร์



รูปที่ 2.7 การขับสแต็ปปีงมอเตอร์แบบยูนิโพลาร์และไบโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ไมโครคอนโทรลเลอร์ และ พอร์ทอนุกรม RS -232

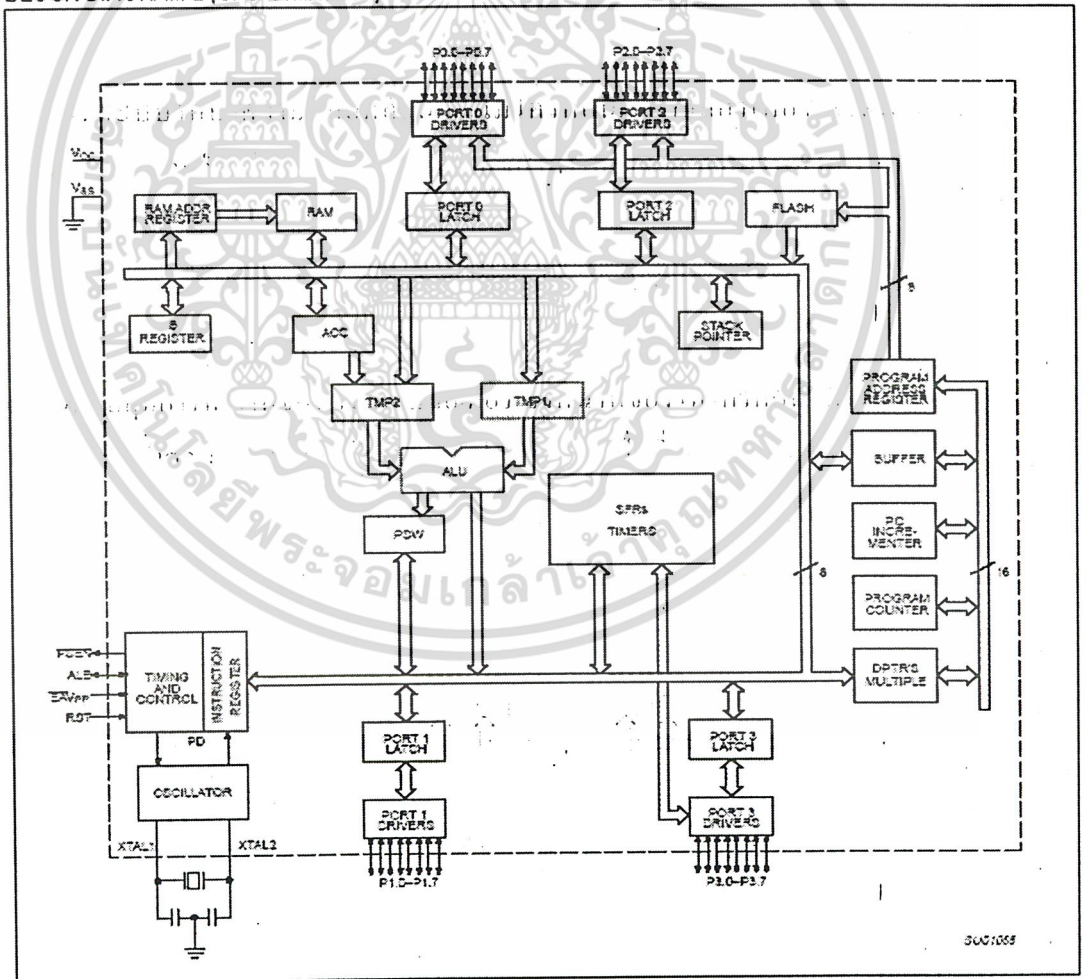
ในระบบรักษาความปลอดภัยนี้ ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51 เนื่องจากว่าภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานได้เป็นทั้งอินพุตและเอาต์พุต มีวงจรสื่อสารอนุกรมแบบฟูลดูเพลิกซ์ พอร์ตการสื่อสารอนุกรมสามารถส่งข้อมูลได้ระยะทางไกลมากที่สุด 1 กิโลเมตร

3.1 ไมโครคอนโทรลเลอร์

เป็นส่วนควบคุมการทำงานของ วงจรขับเคลื่อนมอเตอร์ ซึ่งหน้าที่ของไมโครคอนโทรลเลอร์

1. รับคำสั่งควบคุมจากคอมพิวเตอร์ มาประมวลผลเพื่อที่จะทำงานต่อไป
2. จัดเก็บข้อมูลตำแหน่งของมอเตอร์ทั้งสองตัวเพื่อที่จะส่งไปให้มอเตอร์

BLOCK DIAGRAM 2 (CPU ORIENTED)



รูป 3.1 แสดงสถาปัตยกรรมของ 89C2501

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

89C51 ไมโครคอนโทรลเลอร์ แต่ละข้างจะมีขา 20 ขารวมทั้งหมด 40 ขา โดย V ขาที่ 40 เป็นขาป้อนไฟเลี้ยง +5 โวลต์ เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับแรงดันของลอจิก 0 และ 1 ของ 89C51 จึงต่อเข้ากับอุปกรณ์ ลอจิกแบบ TTL โดยตรง พอร์ต 1 เป็นพอร์ตขนานขนาด 8 บิต อยู่ที่ขา 1 ถึง 8 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับ แต่ละขาจะเขียนว่า P1.0 , P1.1 , ..... P1.7 หมายถึง บิต 7 ของพอร์ต 1 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด และ P1.0 คือ บิต 0 ของพอร์ต 1 เป็นบิตที่มีนัยสำคัญน้อยที่สุด พอร์ต 0 นี้ ใช้ได้ทั้งการรับส่ง ตำแหน่งของข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ตรับส่งข้อมูลก็ได้ ข้อมูลที่ถูกส่งออกทางพอร์ต 1 จะถูกค้างค่าไว้ที่ขาพอร์ต

วงจรภายในส่วน Timing และ Control จะเป็นตัวสร้างสัญญาณมาควบคุมวงจร เพื่อให้การทำงานแต่ละอย่างข้างต้น เมื่อแต่ละบิตของพอร์ต 1 มีการทำงาน คือรับส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory และใช้รับส่งข้อมูลผ่านทางพอร์ตโดยตรง ในกรณีที่ไม่มี การใช้หน่วยความจำ Program Memory หรือ Data Memory ภายนอกแล้ว วงจร Timing And Control จะทำให้สถานะลอจิกของขา Control: เป็น 1 ซึ่งทำให้สวิตช์ MUX อยู่ในตำแหน่งข้างบน เมื่อพอร์ต 1 จะส่งข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำหรือข้อมูลที่เขียนออกไปยัง หน่วยความจำภายนอก ก็จะส่งค่าดังกล่าวมายัง ADDR / DATA ถ้าข้อมูลที่ส่งมาเป็น 1 ก็จะทำให้สัญญาณออกจาก AND GATE เป็น 1 และสัญญาณที่ออกจาก Inverter ดังนั้น FET ตัวบนเปิด (สถานะเปิดของ FET คือ ความต้านทานระหว่างขา D กับ S มีค่าต่ำมากเหมือนวงจรปิด) ส่วน FET ตัวล่างปิด (สถานะปิดของ FET คือ ความต้านทานระหว่างขา D กับ S มีค่าสูงมากเหมือนวงจรเปิด) สถานะลอจิกที่ขา P1.X PIN จะเป็น 1 แต่ถ้า ข้อมูลที่ส่งออกมายัง ADDR / DATA เป็น 1 ก็จะทำให้สัญญาณจาก AND GATE เป็น 0 และสัญญาณที่ออกจาก Inverter เป็น 1 ดังนั้น FET ตัวบนจะปิด ส่วน FET ตัวล่างจะเปิด ทำให้สถานะลอจิกที่ขา P0.X PIN เป็น 0 เมื่อ 89C51 ต้องการใช้พอร์ต 0 สำหรับ อ่านข้อมูลจากหน่วยความจำภายนอกก็จะทำได้โดยวงจร Timing and Control ทำให้สถานะลอจิกของสัญญาณ Control ในรูปเป็น 0 ทำให้เอาท์พุทจาก AND GATE เป็น 0 FET ตัวบนจะปิดและสวิตช์ MUX จะอยู่ในตำแหน่งข้างล่างดังนั้น FET ตัวล่างจะเปิดหรือปิดก็แล้วแต่ ข้อมูลที่ขา Q มีสถานะลอจิก 0 ทำให้ FET ตัวล่างปิดคั้งนั้น ขา P0.X PIN จะอยู่ในสถานะอิมพีแดนซ์สูง เพราะ FET ทั้ง 2 ตัวปิด

พอร์ต 1 เป็นพอร์ตขนานขนาด 8 บิต คือ ขา P1.0 หมายถึงบิต 0 ของพอร์ต 1 เป็นบิต นัยสำคัญต่ำสุดและ บิต P1.7 หมายถึง บิตที่ 7 ของพอร์ต 1 ซึ่งเป็นบิตนัยสำคัญสูงสุด

ส่วนที่ 1 พอร์ต 1 Latch ซึ่งจะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ต 0 ส่วนที่ 2 คือ พอร์ต 1 driver จะมีตัวต้านทานต่ออยู่เป็น Internal Pull Up พอร์ต 1 นี้จะใช้ทำหน้าที่เป็นตัวรับส่งข้อมูล เท่านั้น ข้อมูลที่ถูกส่งออกมาทางพอร์ต 1 จะถูก Latch ไว้ แล้วส่งออกไปทางแต่ละขาก่อนที่จะอ่าน ข้อมูลเข้าไปทางพอร์ต 1 จะต้องเขียน 1 ไปยังทุกบิตของพอร์ต 1 เสียก่อน เพื่อให้ FET อยู่ในสถานะ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารของบริษัทสงวนลิขสิทธิ์สงวนเนื้อหาทั้งหมด ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต

ปิดก่อน มิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งมาค้างอยู่ก็จะทำให้ FET อยู่ในสภาวะเปิด ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่พอร์ตนี้นี้มันก็จะถูกลัดวงจรโดยไม่ว่าลอจิกของสัญญาณที่เข้ามาจะเป็นอะไร ข้อมูลที่อ่านเข้าไปจะเป็น 0 เสมอ

พอร์ต 3 คือขา P3.0 ถึง P3.7 ตามลำดับ โดยข้อมูลที่เขียนมายังพอร์ต 3 ทาง internal bus เหมือนกับพอร์ตอื่นๆและ พอร์ต 3 จะมี Internal Pull Up อยู่ทุกบิต แต่พอร์ต 3 นี้ แต่ละบิตจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน และจะมีส่วนที่มีสัญญาณ alternative Output Function ที่สร้างมาจากส่วน Timing And Control สัญญาณ Alternative Output Function เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ต 3 ทำงานในฟังก์ชันอื่น และจุด Alternative Input Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่นตามการทำงานของบิตนั้น แต่ละบิตของพอร์ต 3 จะมีฟังก์ชันอื่นๆ ดังนี้

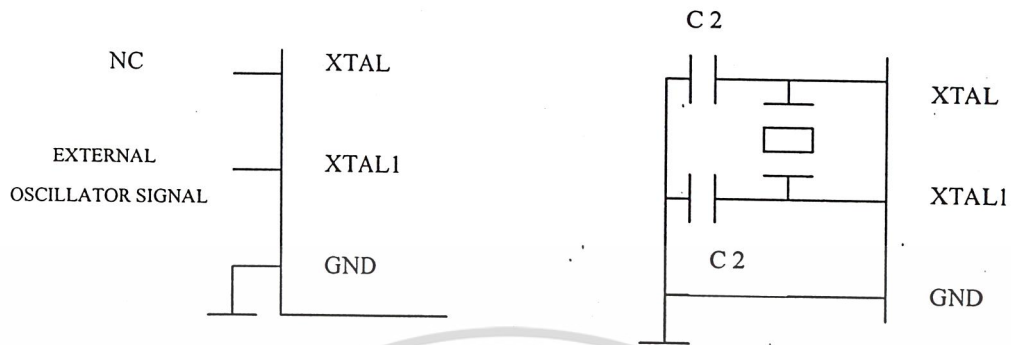
- P3.0 / RXD (Serial Input Port) เป็นขาที่ใช้รับส่งข้อมูล
- P3.1 / TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม
- P3.2 / INTO (External Interrupt) ใช้รับสัญญาณขัดจังหวะภายนอก
- P3.4 / TO (Timer / Counter 0 External Interrupt) ขารับสัญญาณเข้าไปในวงจร Timer / Counter ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ TO นี้หรือ สัญญาณนาฬิกาที่ได้
- P3.5 / T1 (Timer / Counter 1 External Input) ขารับสัญญาณเข้าไปยัง Timer / Counter 1 ซึ่งมีการทำงานเหมือนกับ TO
- P3.7 / RD ขาสัญญาณข้อมูล

RST ขารีเซ็ตขานี้จะใช้ทำการรีเซ็ตการทำงานของ 89C51 ที่ขา RST ภายใน 89C51 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ถ้าป้อนสัญญาณที่สภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ ดังนั้นจึงสามารถต่อตัวเก็บประจุภายนอกระหว่างขา RST กับไฟเลี้ยง +5 โวลต์ เพื่อให้เกิดการรีเซ็ต เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 89C51 ซึ่งเรียกว่า “Power On Reset”

XTAL 1 ขาที่ 5 ขานี้จะต่อเข้ากับขง Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ซึ่งจะเห็นว่าวงจรภายในออสซิลเลเตอร์ NAND GATE จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิลเลตหรือไม่ก็ขึ้นกับสัญญาณ PD ซึ่งต่อมาจากบิต PD ของรีจิสเตอร์ POON ถ้าต้องการใช้สัญญาณนาฬิกาจากภายนอกมาเป็นสัญญาณนาฬิกา ควบคุมการทำงานของ 89C51 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อ Crystal

XTAL 2 ขาที่ 4 ขานี้เป็นจุดเอาต์พุตของวงจรขยายแบบกลับเฟส สัญญาณที่ประกอบเป็นวงจรออสซิลเลเตอร์ซึ่งอินพุตคือขา XTAL 1 ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็น

สัญญาณนาฬิกาของ 89C2051 แล้ว ให้ปล่อยขานี้ลอยไว้ แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 3.2

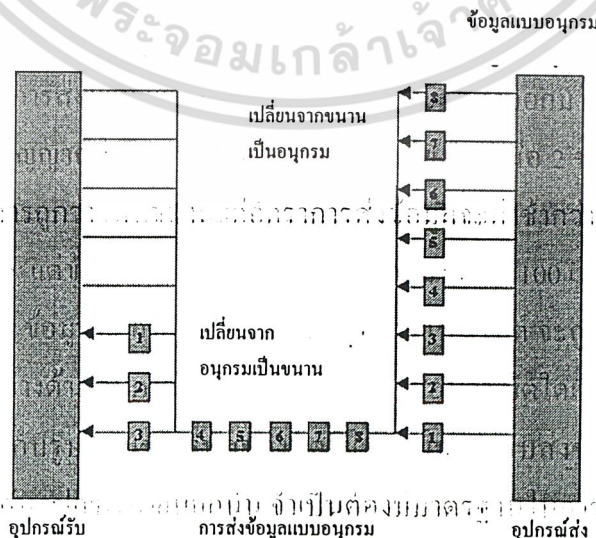


รูปที่ 3.2 การต่อ Oscillator

3.2 สื่อสารพอร์ตอนุกรม RS-232

ลักษณะของการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกส่งออกมาทีละบิต จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ช่องสัญญาณในการส่งข้อมูลอาจจะใช้เพียง 1 หรือ 2 ช่องสัญญาณเท่านั้น ทำให้ค่าใช้จ่ายในการสื่อสารถูกกว่าแบบขนานแต่อัตราการส่งข้อมูลจะล่าช้ากว่าการส่งแบบขนาน ในการส่งข้อมูลแบบอนุกรม แต่ข้อดีคือสามารถส่งข้อมูลได้ไกลมากกว่า 100 เมตร

จากรูปที่ 3.3 ข้อมูลที่ต้องการส่งจะอยู่ในลักษณะเป็นไบนารี จะถูกทยอยส่งทีละบิต แล้วมารวมกันเป็นไบนารี ซึ่งทางด้านอุปกรณ์รับจะต้องคอยตรวจสอบว่า บิตใดเป็นบิตเริ่มแรกของไบนารีนั้น การตรวจสอบขึ้นอยู่กับรูปแบบของรหัสของบิตที่ใช้ ซึ่งในการรับส่งข้อมูลแบบอนุกรมระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้น จำเป็นต้องมีมาตรฐานในการรับส่ง ซึ่งมาตรฐานที่นิยมมากที่สุดคือ มาตรฐาน RS-232



รูปที่ 3.3 การส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### Microsoft Visual C++ 6.0

โปรแกรม Microsoft Visual C++ เป็นโปรแกรมที่ใช้พัฒนาโปรแกรมบน Windows ซึ่งมีรูปแบบการพัฒนาเป็นแบบ Visual ก็คือ เราสามารถออกแบบหน้าต่างของโปรแกรมได้โดยง่าย และมีคลาสไลบรารีที่ชื่อ MFC ซึ่งจะช่วยในการเขียนโปรแกรมบน Windows ได้ง่ายมากยิ่งขึ้น

#### คอมโพเนนต์ที่มีอยู่ใน Visual C++

Visual C++ นั้นมีความสามารถและเครื่องมือให้นำมาใช้งานได้หลายอย่าง ซึ่งเราเรียกแต่ละความสามารถ หรือเครื่องมือว่า “คอมโพเนนต์” (Component) ซึ่งใน Visual C++ นั้นประกอบไปด้วยคอมโพเนนต์ต่างๆดังนี้

#### VC++ Developer Studio

เป็นแกนหลัก หรือหัวใจหลักโดยจะทำหน้าที่เป็นศูนย์กลางในการสร้างแอปพลิเคชัน ซึ่งจะประกอบไปด้วยองค์ประกอบย่อยๆข้างใน โดยจะทำหน้าที่ต่างๆ จัดการกับองค์ประกอบย่อยๆภายในดังนี้

- Project Manager ทำหน้าที่จัดการกับการสร้างแอปพลิเคชันในลักษณะของโปรเจกต์ (แอปพลิเคชันหนึ่งๆประกอบด้วยองค์ประกอบหลายๆส่วนซึ่งเราเรียกรวมๆกันว่าโปรเจกต์)
- Text editor ทำหน้าที่ในการเขียนโปรแกรม โดยเราจะเรียกโปรแกรมที่เขียนว่าซอร์ส โค้ด
- Resource editor ทำหน้าที่ออกแบบหน้าต่างโปรแกรม เช่น เมนู ไอคอนต่างๆ และไอคอน
- Wizard ต่างๆ เพื่อสร้างแอปพลิเคชันในลักษณะวิซาร์ดต่างๆเช่น AppWizard และ ClassWizard จะช่วยให้เราสร้างโค้ดพื้นฐานแก่แอปพลิเคชันอย่างรวดเร็ว โดยเราเพียงแต่กำหนดคลาส C++ จัดการกลวิธ โดวส์เมสเสจ และกระทำงานอื่นๆ อีก เราสามารถสร้างและรัน หรือ execute โปรแกรมจากภายใน
- Compiler จะทำการคอมไพล์อย่างอัตโนมัติ มีการลิงค์ (Link) ไฟล์ต่างๆ เข้าด้วยกัน และอื่นๆที่ถูกต้องโดยเครื่องมือ Build
- Debugger ดีบั๊ก เพื่อแก้ไขความผิดพลาดของแอปพลิเคชัน โดยผ่าน Debugger
- Online Help หากเราต้องการความช่วยเหลือ ( กรณีที่ได้ติดตั้ง MSDN แล้ว) โดยการเลือกคำสั่งบนเมนู Help ของ Developer Studio ก็จะได้รายละเอียดต่างๆ เพื่อคลิกขยายสงสัยต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### VC++ Runtime Libraries

คอมโพเนนต์นี้ทำหน้าที่เก็บฟังก์ชันมาตรฐานต่างๆ ของ ANSI C เช่น ฟังก์ชัน sin ซึ่งเราสามารถเรียกภายในโปรแกรม C หรือ C++

### VC++ MFC and Template Libraries

MFC เป็นไลบรารีคลาส C++ ที่ถูกสร้างมาโดยเฉพาะสำหรับการสร้างแอปพลิเคชันเพื่อใช้งานกับ Windows ซึ่งมีรูปแบบการติดต่อกับผู้ใช้แบบกราฟิก หรือที่เรียกว่า GUI (Graphic User Interface) ซึ่งคลาส MFC ทำให้การเขียนโปรแกรมง่ายขึ้นมาก แล้วยังช่วยประหยัดเวลาในการเขียนโค้ด แล้วยังแก้ไขได้มาก

นอกจากนี้แล้วยังมี ALE (Active Template Library) ซึ่งเป็นชุดของคลาส C++ ที่เป็นเทมเพลต (Template) หรือต้นแบบ ที่จะช่วยอำนวยความสะดวกในการสร้าง ActiveX Control และรวมทั้งรูปแบบการสร้างแอปพลิเคชันอื่นๆ ที่เป็นออบเจกต์ COM (Component Object Model)

### VC++ Build Tools

ประกอบด้วยคอมไพเลอร์ C\C++, Link , คอมไพเลอร์ Resource (สำหรับเตรียมการเกี่ยวกับ Resource โปรแกรมต่างๆ เช่น เมนู ไอคอน เป็นต้น) รวมทั้งเครื่องมืออื่นๆ ที่จำเป็นต่อการสร้างแอปพลิเคชันสำหรับ Windows หรือที่เรามักจะเรียกอีกชื่อว่า การสร้างแอปพลิเคชันแบบ Win 32

### ActiveX

เป็นซอฟต์แวร์ย่อยๆ (Software Component) หรือองค์ประกอบย่อยๆ ที่เราเพิ่มเข้าไปในแอปพลิเคชันที่สร้างขึ้น ซึ่ง ActiveX นั้นทำให้เราไม่จำเป็นต้องสร้างทุกส่วนของแอปพลิเคชันเอง เพียงแต่เลือกใช้องค์ประกอบย่อยๆ ที่เหมาะสมกับงานเพื่อสร้างเป็นแอปพลิเคชันที่สมบูรณ์

ActiveX นั้นมีข้อดีคือ การสร้างซอฟต์แวร์ที่สามารถนำกลับมาใช้งานใหม่ได้ (Reusable) ซึ่งเราจะสร้าง ActiveX ด้วยการ ใช้ไลบรารี MFC ActiveX Control ซึ่งผลลัพธ์ที่ได้จากการสร้างเราจะเรียกว่า ActiveX Control

### Data Access

เป็นคอมโพเนนต์ที่รวมไครเวอร์ฐานข้อมูลชนิดต่างๆ ไว้ให้เราสามารถสร้างแอปพลิเคชันที่เชื่อมต่อกับฐานข้อมูลได้อย่างสะดวก และนอกจากไครเวอร์แล้วยังประกอบไปด้วยคอนโทรล และเครื่องมืออื่นๆ ที่ช่วยในการสร้างแอปพลิเคชันกับฐานข้อมูลทำได้อย่างรวดเร็ว

### Enterprise Tools

เป็นคอมโพเนนต์ที่ประกอบด้วยคอมโพเนนต์ย่อยๆ อีกดังต่อไปนี้

- Microsoft Visual SourceSafe 6.0 Client
- Application Performance Explorer
- Repository
- Visual Component Manager
- Self-installing .exe redistributable files
- Visual Basic Enterprise Components
- VC++ Enterprise Tools
- Microsoft Visual Modeler
- Visual Studio Analyzer

### Graphics

เป็นคอมโพเนนต์เกี่ยวกับรูปภาพในรูปแบบต่างๆ เช่น metafile, bitmap, cursor และ icon รวมทั้ง video clip ซึ่งเราจะใช้คอมโพเนนต์เหล่านี้ในการเขียนโปรแกรมกราฟิกต่างๆ เช่น การนำรูปภาพที่มีนามสกุลเป็นจุด bmp การนำ cursor และอื่นๆ เข้ามาใส่ในแอปพลิเคชันของเรา

### Tools

เป็นคอมโพเนนต์ที่เป็นเครื่องมือเสริมการทำงานชนิดต่างๆ ของ Visual C++ ดังนี้

- API Text Viewer
- MS Info
- MFC Trace Utility
- Spy++
- Win 32 SDK Tools
- OLE/Com Object Viewer
- ActiveX Control Test Container
- VC Error Lookup

### การใช้สภาพแวดล้อมของ Microsoft Visual C++

วิธีการใช้สภาพแวดล้อมการพัฒนาที่รวมเป็นชุด หรือที่เรียกว่า IDE ( Integrated Development Environment) ของ Microsoft Visual C++ ซึ่งได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ชนิดโปรเจกต์ต่างๆ

โปรเจกต์ คือ การรวบรวมองค์ประกอบต่างๆ ของแอปพลิเคชันที่เราจะสร้างขึ้น ( เช่น ซอร์สโค้ด , Resource , ...) ซึ่ง Visual C++ นั้นสามารถสร้างแอปพลิเคชันได้หลายรูปแบบ โปรเจกต์แต่ละแบบที่จะพบจาก Visual C++ ได้แก่

- **ATL COM AppWizard**

ATL (Active Template Library ) ใช้สำหรับการสร้างโปรเจกต์ ATL ที่เป็น COM ( Component Object Model) เช่น ActiveX Control โดยปกติการสร้างโปรเจกต์ประเภทนี้จะถูกใช้โดยนักพัฒนาซอฟต์แวร์ที่ชำนาญ และเน้นหนักไปทางด้านซอฟต์แวร์คอมพิวเตอร์ ( Software Component) ที่สามารถนำกลับมาใช้อีกครั้ง และใช้ได้กับทุกภาษาโปรแกรมโดยการเปรียบเทียบแล้วการสร้างแอปพลิเคชันโดยใช้ MFC ผลลัพธ์คือการใช้ ATL แอปพลิเคชันที่ได้จะมีขนาดเล็กกะทัดรัดกว่าการใช้ MFC

- **Cluster Resource Type Wizard**

Cluster Resource Type Wizard ใช้สร้างโปรเจกต์สองชนิดที่เราสามารถกระจายบน Microsoft Cluster Server เพื่อจัดการและตรวจสอบซอร์สต่างๆ บน cluster การสร้างโปรเจกต์เหล่านี้จะทำได้เมื่อได้ติดตั้งบน Windows NT Server หรือ Windows 2000 server และต้องเป็น Visual c++ Enterprise Edition

- **Custom AppWizard**

ปกติเราจะใช้ MFC AppWizard เป็นเครื่องมือในการสร้างแอปพลิเคชันขึ้นมา ซึ่งสามารถสร้างไฟล์เริ่มต้นได้อย่างรวดเร็ว แต่ถ้าเป็นการสร้างแอปพลิเคชันที่มีความพิเศษในแบบที่ไม่เหมือนใคร จะทำให้ MFC AppWizard ไม่สามารถตอบสนองได้ จึงจำเป็นต้องใช้ Custom AppWizard ในการสร้างโปรแกรมนั้น

Custom AppWizard เป็นประโยชน์มากสำหรับการสร้าง Generic โปรแกรม ซึ่งสามารถสร้างฟังก์ชันทั่วไปอย่างซ้ำแล้วซ้ำเล่า (โปรแกรมนั้นสามารถถูกเรียกใช้แล้วถูกเรียกใช้อีก) Custom AppWizard มีลักษณะการใช้งานเหมือนกับ MFC AppWizard โดยมันจะเสนอหลายๆทางเลือก ติดตามการตัดสินใจของผู้ใช้ และใช้การตัดสินใจเหล่านั้นเพื่อสร้างซอร์สโค้ด , รีซอร์สต่างๆ (Resources) และไฟล์โปรเจกต์ที่จำเป็นต่อการแอปพลิเคชัน

- **Database Project ( Enterprise Edition only)**

ถ้าต้องการใช้การสืบค้นข้อมูลเพื่อทดสอบ SQL stored procedures เราก็ใช้ Database Project นี้

- **DevStudio Add-in Wizard**

ด้วยการ Add-ins หรือมาโคร VBScript เราสามารถทำงานโดยอัตโนมัติในสภาพแวดล้อม Visual C++ Developer Studio ตัวอย่างเช่น เราสามารถอัปเดตโปรเจกต์ด้วยการเปิดโปรเจกต์, การเปลี่ยนบางไฟล์, การบันทึกการเปลี่ยนแปลง และการคอมไพล์โปรเจกต์อีกครั้งอย่างอัตโนมัติ

- **Extended Stored Procedure AppWizard**

เราสามารถสร้าง SQL Server Extended Stored Procedures ( XPROC ) ได้อย่างง่ายด้วยการใช้ Extended Stored Procedures Appwizard ใน Visual C++ ที่เป็น Enterprise Edition

Extended Stored Procedure นั้นเป็นฟังก์ชันภาษาซี extern ที่ออกไปเป็นไฟล์ Win32 DLL ซึ่งเราสามารถเรียกจากภายใน SQL Server

วิธีการเรียกใช้ Extended Stored Procedures จะทำเช่นเดียวกับ Stored Procedure ปกติ แต่มันรันบน Server เหมือนกับโปรเซสที่ถูกคอมไพล์แล้ว ไม่เหมือนกับ SQL Server Stored Procedure ที่เวลารันต้องการคำสั่งในภาษา SQL (SQL Statement)

- **ISAPI Extension (Internet Server API) Wizard**

ISAPI มาจากคำเต็ม Internet Server Application Programming Interface ถ้าเราต้องการซอฟต์แวร์ที่เข้ากันได้กับ ISAPI แล้ว เราสามารถสร้าง Internet Server Extension And Filter ด้วย MFC Internet Server Extension เป็น DLL ที่ใช้เพื่อเสริมความสามารถของเว็บเซิร์ฟเวอร์ ISAPI

การใช้ ISAPI Extensions นี้เป็นทางเลือกหนึ่งในการเขียน CGI (Common Gateway Interface) แอปพลิเคชัน ส่วน Internet Server Filter คือ DLL ที่รันบนเซิร์ฟเวอร์ ISAPI ที่สามารถรันได้เพื่อกรองข้อมูลที่ผ่านไปยังเซิร์ฟเวอร์และจากเซิร์ฟเวอร์

- **MFC ActiveX Control Wizard**

เมื่อเราต้องการสร้าง ActiveX Control ในครั้งแรกเราอาจใช้ MFC ActiveX Control Wizard เพื่อเป็นแนวทางแก่เรา ซึ่งมันจะประกอบไปด้วยชุดของไดอะล็อกบ็อกซ์ต่างๆ ให้เราเลือกเอาคุณลักษณะ และฟังก์ชันต่าง ๆ ให้แก่ ActiveX Control

- **MFC AppWizard (DLL version)**

ถ้าเราต้องการรวบรวมฟังก์ชันต่างๆ เข้าไปใน DLL และฟังก์ชันเหล่านี้ใช้คลาส MFC เราต้องใช้ MFC AppWizard (DLL version) นี้ การสร้าง DLL นี้ทำได้ทั้งจากการใช้ ATL และ MFC แต่การใช้ MFC AppWizard (DLL version) จะง่ายกว่า

- **MFC AppWizard (.EXE version)**

เมื่อเราต้องสร้าง โปรแกรม MFC ที่สามารถรันได้เราต้องใช้ MFC AppWizard (.EXE version) เพื่อเป็นการแนะแนวทางแก่เราด้วยชุดของไดอะล็อกต่างๆ ที่เราเลือก Option ที่มีคุณลักษณะและฟังก์ชันต่างๆ

แต่ละชุดจะเป็นไดอะล็อกย่อยต่างๆ เราสามารถคลิกไปข้างหน้า ถอยหลัง และทำการเปลี่ยนแปลงได้ นอกจากนี้แล้ว Option ที่เราเลือกมาทุก ๆ Option นี้จะมี Help อยู่ใน MFC AppWizard โดยการคลิกขวาที่ตัวคอนโทรล ซึ่งจะมีรายละเอียดในแต่ละ Option นั้นๆ

- **Utility Project**

Utility Project ถูกใช้เหมือนกับเป็น Container ให้แก่ไฟล์ที่เราสามารถคอมไพล์โดยไม่มีภารกิจเข้าด้วยกัน

- **Win32 Application**

Win32 Application เป็นแอปพลิเคชันที่เขียนด้วยภาษา C หรือภาษา C++ โดยในการสร้างแอปพลิเคชันจะไม่มีการใช้คลาส MFC เลย ซึ่งเรามักจะได้ยินชื่อของเครื่องมือที่ใช้สร้างแอปพลิเคชันแบบนี้ว่า Win32 SDK ( Software Development Kit)

การสร้างแอปพลิเคชันชนิดนี้มีข้อดีคือ มีผลิตภัณฑ์การทำงานที่รวดเร็วมาก ( เพราะเรียกใช้ฟังก์ชันของ Windows โดยตรง) แต่มีข้อเสียสำคัญคือ การเขียนโปรแกรมมีความซับซ้อนกว่า เพราะต้องมีความรู้พื้นฐานเกี่ยวกับระบบปฏิบัติการทำงานภายใน Windows เป็นอย่างดี และเข้าใจฟังก์ชันต่างๆ ของ Windows ซึ่งมีหลายหมวดหมู่ และแต่ละหมวดหมู่ก็มีฟังก์ชันภายในเป็นจำนวนมาก

- **New Database Wizard**

New Database Wizard เป็นเครื่องมือวิซาร์ดที่ใช้สำหรับการสร้างฐานข้อมูล SQL ใหม่ ในการใช้เครื่องมือนี้เราต้องติดต่อกับฐานข้อมูล SQL Server ได้หรือต้องติดตั้ง MSDE 6.0 (Microsoft Data Engine) ซึ่งมีคุณลักษณะคล้ายคลึงกับโปรแกรม SQL Server 7

- **Win32 Console Application**

แอปพลิเคชันแบบ Console นั้นก็เหมือนกับแอปพลิเคชัน DOS ซึ่งการสร้างโปรแกรมนี้ใช้ไลบรารี Run-Time เช่น Cout , cin , math เป็นต้น

ข้อดีของการสร้างแอปพลิเคชันแบบนี้คือ โค้ดจากการสร้างแอปพลิเคชันสามารถนำไปรันในแพลตฟอร์มอื่น ๆ (DOS, Linux) ได้ด้วยเพราะใช้คอมไพเลอร์ตามมาตรฐาน ANSI C

- **Win32 Dynamic Link Library**

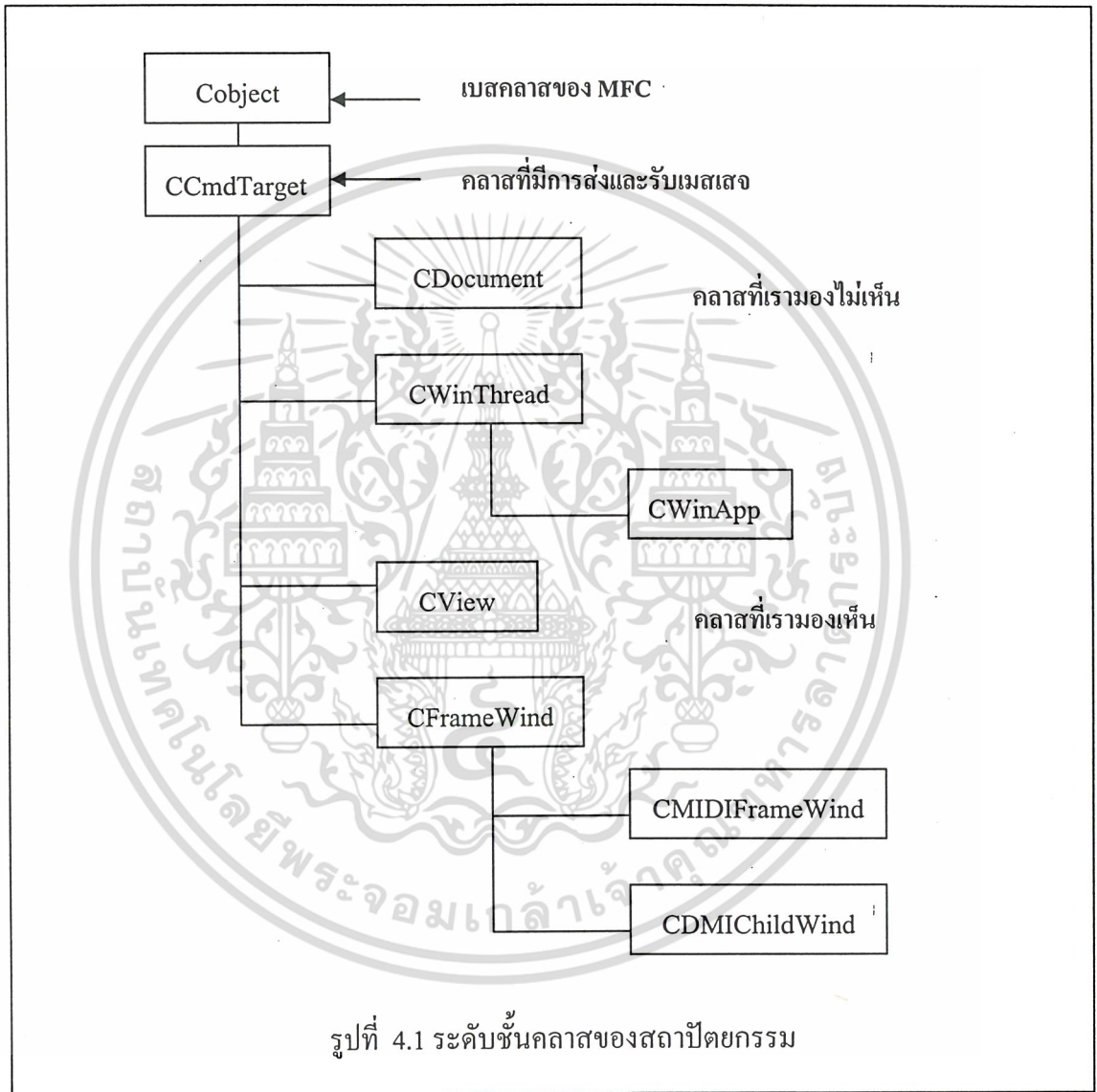
Win32 DLL เป็นไบนารีไฟล์ที่เขียนโดยภาษา C หรือ C++ ด้วยการ ใช้ Win32 API ซึ่งไม่ได้ใช้คลาส MFC เลย

Win32 DLL ทำงานเหมือนกับการแชร์ไลบรารีของฟังก์ชันต่างๆ คือ ถูกเรียกใช้งานจากแอปพลิเคชันต่าง

• Win32 Static Library

Static library เป็นไฟล์ที่ประกอบด้วยหลายๆ ออบเจกต์และฟังก์ชัน รวมทั้งข้อมูลที่เชื่อมโยงกับแอปพลิเคชันของเรา ซึ่งเมื่อแอปพลิเคชันนั้นถูกคอมไพล์ จะมีการเชื่อม Static Library มายังแอปพลิเคชันที่ถูกสร้างนั้นซึ่งส่วนที่เชื่อมนี้จะรวมเข้าไปกับตัวแอปพลิเคชัน

สถาปัตยกรรมของ MFC ( Microsoft Foundation Class) และข้อดีของการใช้ MFC



คลาส CObject

คลาส CObject ของ MFC เป็นคลาสแม่ หรืออาจจะเรียกว่า เบสคลาส (Base Class) ซึ่งมีคลาสต่างๆ ที่สืบทอดจากคลาสนี้มากกว่า 200 คลาส และมีอีก 30 กว่าคลาสที่ไม่ได้สืบทอดมาจากคลาส CObject

คลาส MFC เกือบทั้งหมด (รวมทั้งที่เราสร้างคลาสขึ้นมาใหม่ด้วย) จะต้องสนับสนุนกับการจัดการกับ Runtime type information และจัดการกับการ Persistence ออบเจกต์ หรือที่พูดว่าการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำซีเรียลไลเซชัน (Serialization) และกระทำเอาต์พุตที่วิเคราะห์ไว้สำหรับออบเจกต์ที่สืบทอดมาจากคลาสนี้คลาสนี้สืบทอดมาจาก CObject สามารถบรรจุอยู่ในคลาสนอนเทรนเนอร์ MFC ได้

คลาสนี้หลายที่สืบทอดมาจากคลาสนี้ CObject จะต้องมีคุณลักษณะอย่างน้อยหนึ่งในสามอย่างดังนี้

1. **การสนับสนุนการวิเคราะห์ทั่ว ๆ ไป** : คลาสนี้ที่สืบทอดมาจะต้องมีคุณลักษณะนี้เป็นส่วนใหญ่ ซึ่งถูกสร้างอยู่ในคลาสนี้ CObject ที่อนุญาตให้เรากระทำการตรวจสอบออบเจกต์ที่สืบทอดมาจาก CObject ใช้ได้หรือไม่ และการใส่ข้อมูลการตรวจสอบจากอินพุตที่อ่านได้ และเอาต์พุตที่แสดงในหน้าต่างเอาต์พุตของ Debugger ได้หรือไม่
2. **การสนับสนุน RunTime Type Information (RTTI)** : เมื่อคลาสนี้ MFC สนับสนุน RTTI มันสามารถดึงชื่อคลาสนี้ของออบเจกต์ และข้อมูลอื่นๆ เกี่ยวกับออบเจกต์ ณ เวลาทำงาน(Run Time)
3. **การสนับสนุน Serialization** : การ Serialization เป็นกระบวนการของการไหล (Stream) ข้อมูลที่คงอยู่เรื่อยๆ ของออบเจกต์ไปยังฮาร์ดดิสก์ (การบันทึก) หรือจากฮาร์ดดิสก์ออกไป (การอ่าน)

#### คลาสนี้ CCmdTarget

คลาสนี้สืบทอดมาจาก CObject มีหน้าที่รับผิดชอบเส้นทางของระบบ และอีเวนต์วินโดวส์เพื่อไปยังออบเจกต์ต่างๆ ที่สามารถตอบสนองต่ออีเวนต์เหล่านี้ คลาสนี้ที่รับระบบและอีเวนต์วินโดวส์จะเป็นคลาสนี้ CWnd (คลาสนี้วินโดวส์) คลาสนี้ CDocument (คลาสนี้คอกิวเมนต์) คลาสนี้ CWinThread (คลาสนี้ User Interface Thread) และคลาสนี้ CWinApp (คลาสนี้แอปพลิเคชัน)

นอกจากนี้คลาสนี้ CCmdTarget เป็นเบสคลาสนี้สำหรับสถาปัตยกรรมแม่พิมพ์เมสเสจ

#### คลาสนี้ CWinthread

คลาสนี้สืบทอดมาจากคลาสนี้ CCmdTarget คลาสนี้แสดงเธรด (Thread) ของการ Execution ภายในแอปพลิเคชัน MFC แอปพลิเคชัน MFC ทั้งหมดมีอย่างน้อยหนึ่งออบเจกต์ CWinThread ซึ่งจะเป็นออบเจกต์ CWinApp ของแอปพลิเคชันหลักโดยที่ CWinApp สืบทอดมาจาก CWinThread อีกทอดหนึ่ง ถ้าเราต้องการที่เพิ่มการ โพรเซส Asynchronous ภายในแอปพลิเคชันของเรา เราสามารถสร้างและรันออบเจกต์ CWinThread เพิ่มอีกเมื่อจำเป็น

โดยทั่วไปมีเธรดอยู่สองชนิดที่สนับสนุน คือเธรด Worker และเธรด user-interface เธรด Worker ไม่มีการลูปเมสเสจ ตัวอย่างเช่น เธรดที่กระทำการคำนวณที่เป็นฉากหลัง (Background) ในแอปพลิเคชัน spreadsheet ส่วนเธรด user-interface มีการลูปเมสเสจและโพรเซสเมสเสจที่ได้รับมาจากระบบ

### คลาส CWinApp

คลาสนี้สืบทอดจากคลาส CWinThread คลาสนี้ไม่เพียงแต่แสดงเชรคหลักของโปรแกรมเมื่อมันทำงาน แต่ยังแสดงตัวแอปพลิเคชันของมันเองด้วย ผลลัพธ์ที่ได้ก็คือมีเพียงออบเจกต์เดียวในแอปพลิเคชัน MFC ใดๆ

ออบเจกต์ได้ให้เมมเบอร์ฟังก์ชันสำหรับการเริ่มต้นแอปพลิเคชันของเรา (และอินสแตนซ์ของมัน) และสำหรับการรันแอปพลิเคชัน แต่ละแอปพลิเคชันซึ่งใช้คลาส MFC สามารถมีเพียงหนึ่งออบเจกต์เท่านั้นที่สืบทอดจาก CWinApp

### คลาส CDocument

คลาสนี้สืบทอดจากคลาส CCmdTarget คลาส CDocument ถูกสร้างโดยออบเจกต์ Document Template คลาสนี้ได้ให้ฟังก์ชันพื้นฐานสำหรับคลาสคือออบเจกต์ที่ผู้ใช้กำหนด คือออบเจกต์แสดงหน่วยของข้อมูลที่โดยปกติผู้ใช้เปิดด้วยคำสั่งไฟล์ Open และบันทึกด้วยคำสั่งไฟล์ Save

คลาสนี้สนับสนุนการทำงานมาตรฐาน เช่นการสร้างเอกสาร หรือคือออบเจกต์ การโหลดและการบันทึก คือออบเจกต์ คลาสที่เกี่ยวข้องกับคือออบเจกต์อีกคลาสหนึ่งคือ คลาส CArchive ที่อำนวยความสะดวกในการโอนย้ายข้อมูลของคือออบเจกต์กับฮาร์ดดิสก์

### คลาส CWnd

คลาสนี้สืบทอดจากคลาส CCmdTarget คลาสนี้เป็นออบเจกต์ GUI (Graphic User Interface) พื้นฐานมากที่สุดในคลาส MFC อินสแตนซ์ของคลาสนี้ และคลาสที่สืบทอดเป็นวินโดวส์ วินโดวส์ก็เป็นออบเจกต์ที่มี Procedure วินโดวส์ซึ่งถูกริจิสแล้ว และดังนั้นมันสามารถรับและจัดการกับระบบ และอีเวนต์วินโดวส์ วินโดวส์ส่วนใหญ่แล้วจะมีการแสดงกราฟิก และตอบสนองกับอินพุตผู้ใช้

### คลาส CFrameWnd

คลาสนี้สืบทอดจาก CWnd คลาส CFrameWnd เป็นวินโดวส์ที่ประกอบด้วย Title bar เมนู system border ป๊อปอัพ minimize ทูลบาร์ สเตตัสบาร์ และวินโดวส์ที่กำลังทำงานอยู่ คลาส CFrameWnd สนับสนุน SDI (Single Document Interface)

แต่ถ้าเป็น MDI (Multiple Document Interface) จะใช้คลาส CMDI FrameWnd สำหรับเฟรม workspace และคลาส CMDIChildWnd สำหรับวินโดวส์ลูก MDI ทั้งสองคลาสนี้สืบทอดจากคลาส CFrameWnd

คลาสที่เกี่ยวข้องกับคลาสนี้คือ คลาส CMenu ที่ได้ให้การอินเตอร์เฟซตลอดที่มีการใช้เมนูของแอปพลิเคชันจะมีประโยชน์มากสำหรับการจัดการเมนู ณ เวลาทำงาน

### คลาส CView

คลาสนี้สืบทอดจาก CWnd คลาส CView ได้ให้ฟังก์ชันพื้นฐานสำหรับคลาสวิวที่ผู้ใช้กำหนด วิวที่ติดกับค็อกคิวเมนต์ และกระทำเหมือนกับสื่อกลางระหว่างค็อกคิวเมนต์กับผู้ใช้ วิววาดภาพของค็อกคิวเมนต์บนหน้าจอหรือเครื่องพิมพ์ และตีความหมายจากอินพุตผู้ใช้ให้ปรากฏบนค็อกคิวเมนต์

วิวเป็นลูกของเฟรมวินโดวส์ วิวที่มากกว่าหนึ่งวิวสามารถใช้เฟรมวินโดวส์เหมือนกับในกรณีของการแยกวินโดวส์ (Splitter Window) ความสัมพันธ์ระหว่างคลาสวิว คลาสเฟรมวินโดวส์ และคลาสค็อกคิวเมนต์จะถูกสร้างโดยคลาส CDocTemplate เมื่อผู้ใช้เปิดวินโดวส์ หรือแยกวินโดวส์ เฟรมเวิร์คจะสร้างวิวใหม่ และยังคงยึดติดกับค็อกคิวเมนต์ วิวสามารถยึดติดกับค็อกคิวเมนต์เดียว แต่ค็อกคิวเมนต์หนึ่งสามารถมีได้หลาย ๆ วิว

### ประโยชน์จากการใช้คลาส MFC

คลาส MFC ได้ให้ข้อดีหลาย ๆ อย่างในการพัฒนาแอปพลิเคชันบนวินโดวส์ ถึงแม้ว่าคลาส MFC ไม่ได้ห่อหุ้มด้วยฟังก์ชัน Win32API ทั้งหมด แต่มันก็ได้รวบรวมสิ่งที่เป็นโครงสร้างเฟรมเวิร์คหลักให้มีผลให้การพัฒนาแอปพลิเคชันต่าง ๆ บนวินโดวส์ง่ายขึ้นไปมาก เช่น การสร้าง ActiveX Control และ Component การสร้างแอปพลิเคชันอินเทอร์เน็ต เป็นต้น เราสามารถสรุปข้อดีได้ต่อไปนี้

1. ความสามารถเข้ากันได้ดีระหว่างคลาส MFC ในเวอร์ชันที่แล้ว ๆ กับคลาส C++ ใหม่
2. เฟรมเวิร์ค MFC ใช้เทคนิคของ OOP มีผลให้โค้ดสามารถนำกลับมาใช้ใหม่ได้
3. การสร้างแอปพลิเคชันเฟรมเวิร์คหลักให้ ซึ่งมีผลให้เราลดการเขียนโค้ดลงไปมาก เมื่อเทียบกับการเขียนโดยใช้ SDK
4. เมื่อใช้คลาส MFC แล้ว ตัวคอมไพเลอร์ยังคงสนับสนุนไลบรารีมาตรฐาน ANSI C

ส่วนข้อเสียของการใช้ MFC เมื่อเทียบกับการใช้ SDK คือ ใช้โค้ดจำนวนมากกว่า เนื่องจากโค้ด MFC เมื่อคอมไพล์ จะมีการแปลงโค้ด MFC อีกครั้งหนึ่ง และมีผลให้การทำงานของโปรแกรมช้ากว่าเล็กน้อย อย่างไรก็ตามมันก็เร็วกว่าภาษาระดับสูง เช่น Pascal และ Visual Basic เป็นต้น

## บทที่ 5

### หลักการออกแบบและการสร้าง

ในส่วนนี้เราได้ทำการออกแบบและสร้างในส่วนของ ชุดควบคุมกลิ้ง โดยอธิบายได้ดังนี้ โดยใช้ไมโครคอนโทรลเลอร์ เป็นตัวควบคุมชุดขับ สเต็ปป์มอเตอร์โดยวงจรขับจะใช้ IC เบอร์ ULN 2803 เป็นตัวขับ ซึ่งภายในจะประกอบไปด้วยทรานซิสเตอร์ 8 ตัว ต่อแบบคาร์ลิงตัน ส่วนของ stepping motor ที่ใช้จะมีขนาดเล็ก ซึ่งมีคุณสมบัติคือ ใช้ไฟเลี้ยง 12 โวลต์ และมี 1.8 ดีกรี ต่อ 1 สเต็ป จำนวน 2 ตัว เพื่อควบคุมการหมุนขึ้น-ลง และ ซ้าย-ขวา

สำหรับ โปรแกรมที่ควบคุมการหมุนของสเต็ปป์มอเตอร์นี้ จะถูกเก็บไว้ที่ตัวไมโครคอนโทรลเลอร์ โดยใช้ภาษาแอสเซมบลีเป็นตัวเขียน แผนภาพในการทำงานจะอยู่ด้านหลัง

การควบคุมกลิ้งนี้เราจะใช้เมาส์เป็นตัวกำหนดทิศทางบนหน้าจอคอมพิวเตอร์ของเรา หลังจากมีการคลิกเมาส์ลากเส้นทางไปยังที่ที่เราต้องการ ค่าของเมาส์ (พิจารณาเป็นแกน X กับแกน Y) จะถูกส่งเข้ามาทาง RS 232 ซึ่งเป็นพอร์ตอนุกรมในการรับค่าจากคอมพิวเตอร์จะรอรับทีละไบต์ ( 8 บิต ) ซึ่งค่าที่ส่งเข้ามาจะเป็นรหัส ASCII ดังนั้นจึงต้องมีการแปลงค่าให้อยู่ในรูป BCD สำหรับค่าที่รับเข้ามาค่าแรกจะเป็นค่าของบิตล่างของค่า Y ค่าที่เข้ามาตัวที่สองจะเป็นบิตบนของแกน Y ค่าตัวที่สามจะเป็นค่าบิตล่างของแกน X ค่าสุดท้ายที่เข้ามาจะเป็นค่าของ X บิตบนนำค่าที่ได้มาคำนวณตามโปรแกรมที่เราได้เขียนไว้เพื่อหาว่าต้องการให้มอเตอร์มีทิศทางอย่างไร และต้องหมุนให้เค็ดมูกี่สเต็ป โดยการหมุนของมอเตอร์เราจะใช้การกระตุ้นแบบครึ่งสเต็ป โดยถ้าเกิดไม่มีการส่งค่าจากพอร์ตอนุกรมภายใน 20 นาที จะถือว่าจบการทำงาน

ในส่วนของวงจรตรวจจับความเคลื่อนไหว จะทำงานโดยค่านวจรส่ง(ไอซีเบอร์ 74HC00) จะส่งสัญญาณออกไปตลอดเวลา ลักษณะสัญญาณที่ได้จะเป็นพัลส์ของ 0 กับ 1 สลับกันไป ในส่วนของวงจรรับ (ไอซีเบอร์ TSOP 4838) เมื่อไม่มีอะไรมากระทบมันกับตัวส่งก็จะได้สัญญาณเป็น HIGH แต่ถ้ามีสิ่งกีดขวางมากระทบทำให้วงจรรับรับ สัญญาณจากวงจรส่งไม่ได้ วงจรรับจึงมีสถานะเป็น LOW

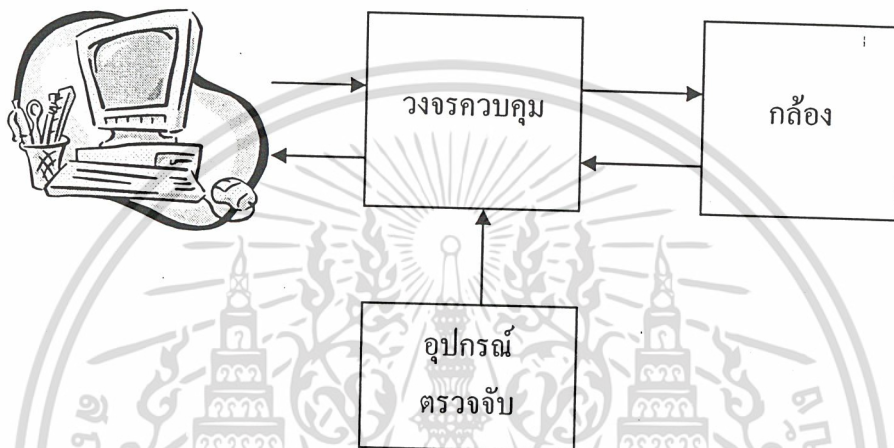
ในภาคของโปรแกรมบนคอมพิวเตอร์จะใช้โปรแกรม Microsoft Visual C++ โดยใช้ในส่วนของ win 32 application โดยเหตุที่ใช้ Visual C++ ก็เพราะเหมาะกับการนำมาดัดแปลงเพื่อใช้ประโยชน์ได้มาก โดย code ที่ใช้เขียนจะใช้ภาษาซี ในการเขียนโปรแกรม โดยแบ่งออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของการเขียน windows และส่วนของการส่งพิกัดไปทาง serial port โดยในส่วนของ windows เมื่อเราคลิกแล้วจะสามารถบอกพิกัดของหน้าจอได้ และจะส่งข้อมูลออกไปที่ serial port เพื่อไปขับมอเตอร์อีกครั้ง อธิบายโดยรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ส่วนของคอมพิวเตอร์

คอมพิวเตอร์จะเป็นตัวสร้างหน้าจอวินโดวส์เพื่อดึงภาพจากกล้องวงจรปิดมาปรากฏบนหน้าจอ และเมื่อมีการคลิกเมาส์บนหน้าจอเพื่อเลือกตำแหน่ง คอมพิวเตอร์ก็จะส่งพิกัดของหน้าจอ ตำแหน่งนั้นไปให้ไมโครคอนโทรลเลอร์เพื่อจะนำไปประมวลผลต่อไป โดยส่งค่าเข้ามาทางพอร์ตอนุกรมการสื่อสาร

ส่วนควบคุมการกวาดของกล้องวงจรปิดโดยคอมพิวเตอร์ การทำงานดังแผนภาพ ดังนี้



รูปที่ 5.1 แผนภาพการทำงาน

### ส่วนของโปรแกรม

โปรแกรมภาษาซีโดยใช้ Visual C++ 6.0 โดยใช้ในส่วนของ MFC ในการสร้างทั้งวินโดวส์ โดยในโปรแกรมจะแบ่งเป็น 4 ส่วนใหญ่ๆ ได้แก่

1. หน้าจอวินโดวส์ Dialog
2. ส่วนติดต่อกับพอร์ตอนุกรม
3. การดึงภาพขึ้นหน้าจอ
4. การรอข้อความ (Message) เพื่อทำงานเช่นการคลิกเมาส์ การลากหน้าจอ การกดคีย์บอร์ด เป็นต้น

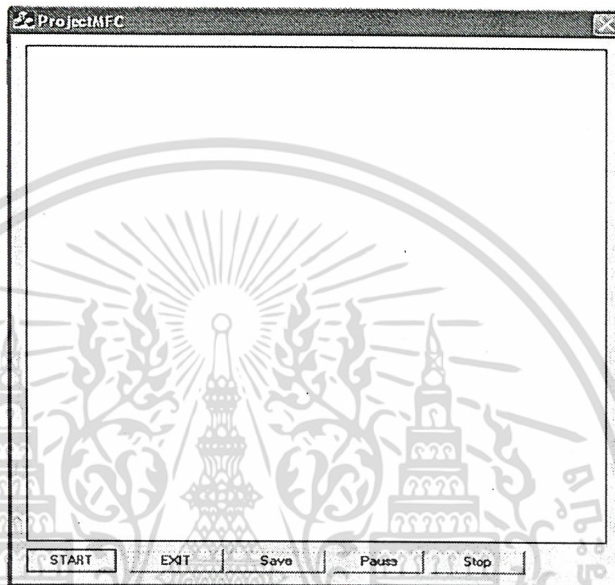
### รายละเอียดของโปรแกรม

ในส่วนโปรแกรมจะสร้างหน้าจอ Dialog มาตรฐานขึ้นมาก่อน โดยขั้นตอนการสร้างโปรแกรมสามารถสรุปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. หน้าจอวินโดวส์ Dialog

ในส่วนนี้จะสร้างเพียงหน้าจอ windows ที่เป็นสีเทา โดยในรายละเอียดโปรแกรมนั้น จะมีทั้งส่วนที่รวมถึงการกดคานาทัมขวาเพื่อปิดโปรแกรม เมื่อเลื่อนหน้าต่างจะมีการเลื่อนตามที่เมาส์คลิกลากไป การสร้างหน้าจอใหม่เมื่อมีโปรแกรมอื่นเปิดขึ้นมาทับและได้ทำการปิดไปแล้ว ฯลฯ และสร้างส่วนของปุ่ม control ต่างๆขึ้นมา โดยรายละเอียดของปุ่ม control แต่ละตัวจะแตกต่างกันออกไปแล้วแต่ประโยชน์การใช้งาน



รูปที่ 5.2 ภาพของโปรแกรมที่สร้างเพื่อใช้งาน

## 2. ส่วนติดต่อกับพอร์ตอนุกรม

โปรแกรมจะทำการเตรียมการการสื่อสารผ่านพอร์ตสื่อสารอนุกรม โดยใช้ class `serialcomm` ซึ่งเป็น class ที่อำนวยความสะดวกในการติดต่อกับพอร์ตอนุกรม ซึ่งสามารถหา download ได้ทางอินเทอร์เน็ตนำมาช่วยในการสื่อสาร ซึ่งมีวิธีนำมาใช้ 4 ขั้นตอนดังนี้

1. กำหนดตัวแปรสำหรับเก็บ file names , data read (Buffer), File handles, ค่าของการรับ/ส่งข้อมูล
2. ใช้คำสั่ง `CreateFile` ในการอ้างอิงไฟล์ข้อมูลขึ้นมาใหม่
3. กำหนดค่าสำหรับการติดต่อกับ Serial port โดยใช้ data structure ชื่อ `DCB`
4. ใช้คำสั่ง `WriteByte` เพื่ออ่าน หรือ `ReadByte` เพื่อเขียนตัวอักษรผ่านทาง SerialPort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

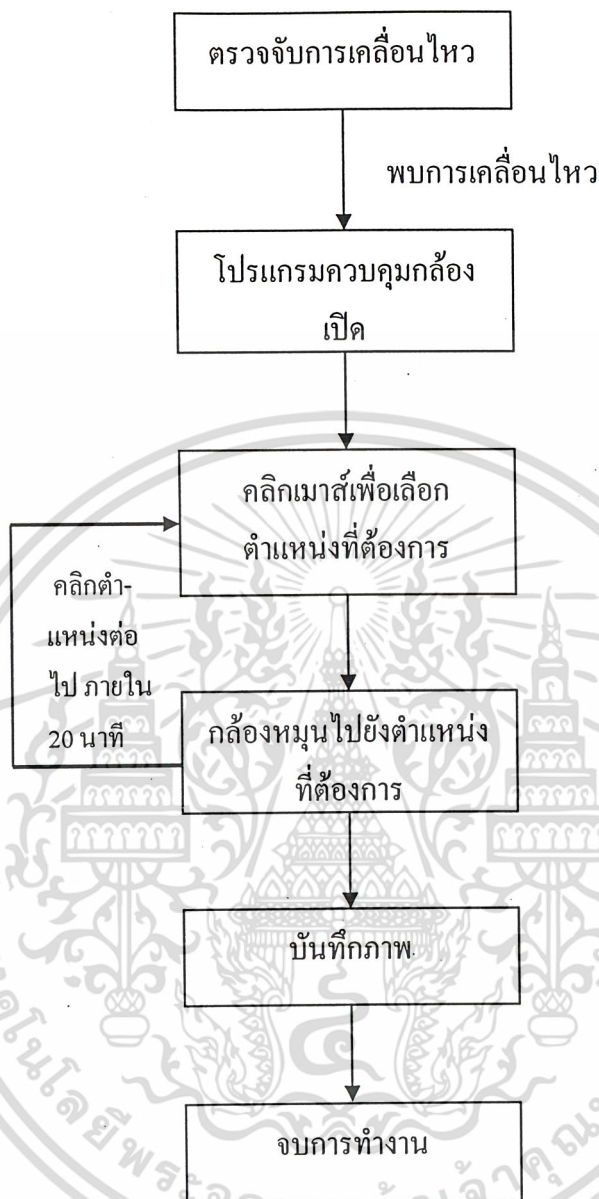
### 3. การดึงภาพขึ้นหน้าจอ

การดึงภาพหน้าจอจะใช้ร่วมกับ โปรแกรม VideoOCX ซึ่งเป็นซอฟต์แวร์คอมพิวเตอร์ของการดึงภาพจากกล้องโดยตรงอยู่แล้ว โดยการที่เราจะนำซอฟต์แวร์คอมพิวเตอร์อื่นมาใช้ร่วมกับโปรแกรมเรานั้น ต้องใช้ ActiveX ช่วยในการดึงฟังก์ชันต่างๆมาใช้

### 4. การรอข้อความ (Message) เพื่อทำงาน

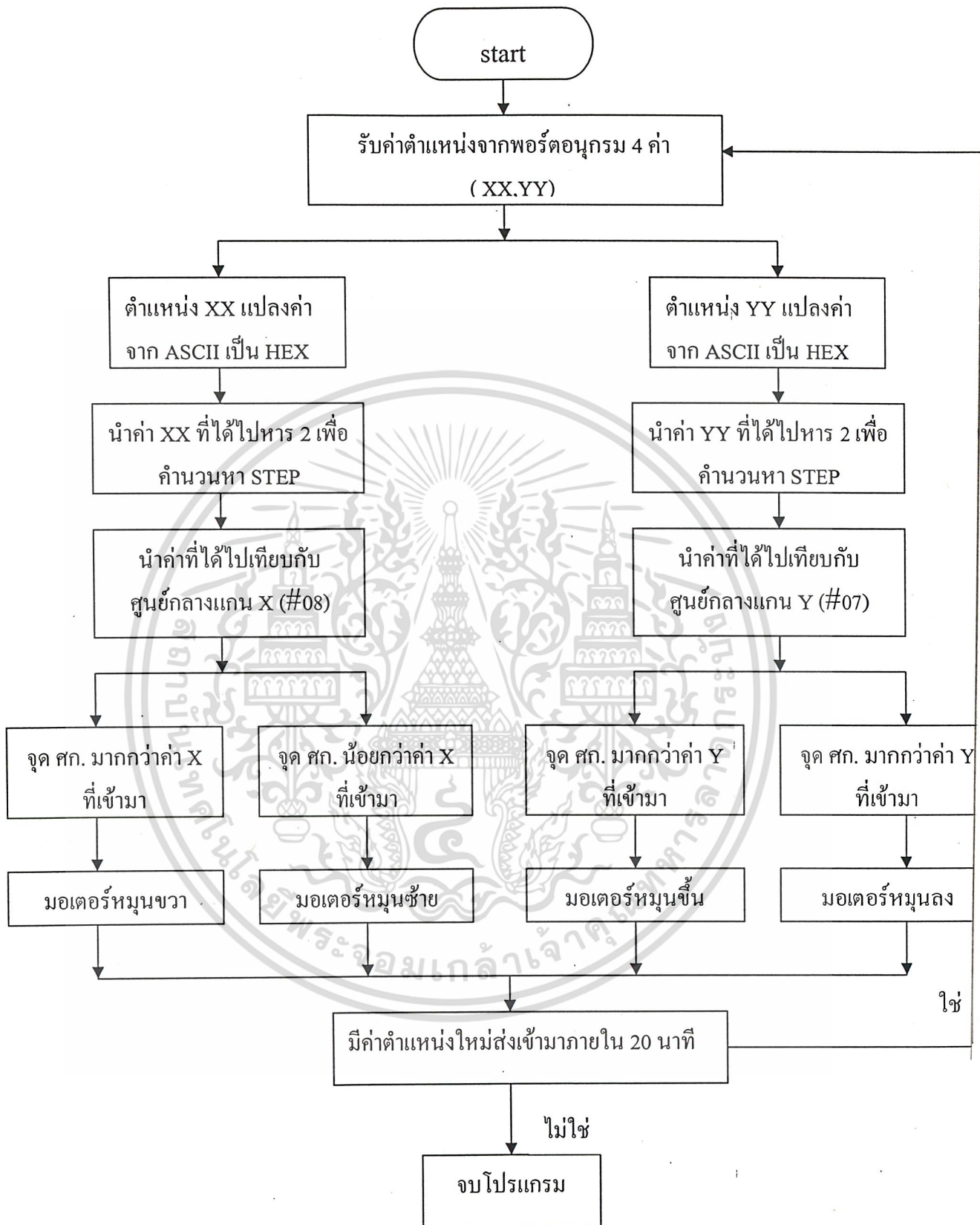
ในโปรแกรมนี้จะมีข้อความที่ต้องรออยู่ 7 Message คือ

1. Message Start ในส่วนนี้จะทำการสร้างรูปของการรอรับสัญญาณของเซ็นเซอร์ ที่ส่งมาโดยไมโครคอนโทรลเลอร์ เมื่อได้รับสัญญาณมา ก็จะทำการเรียกฟังก์ชันของ VideoOCX ในการดึงภาพขึ้นหน้าจอ
2. Message Exit จะเป็นการเรียกฟังก์ชันการปิดโปรแกรมซึ่งเป็นของ Windows Dialog มาตรฐาน
3. Message Save เป็นการเรียนฟังก์ชันการบันทึกภาพของ VideoOCX
4. Message Pause เป็นการเรียนฟังก์ชันการหยุดการบันทึกภาพของ VideoOCX
5. Message Stop เป็นการหยุดการดึงภาพของหน้าจอ
6. Message การจัดการเคลื่อนไหวของเมาส์ ซึ่งเราต้องใช้ฟังก์ชันใน VideoOCX ซึ่งเพราะบริเวณที่เราต้องการตำแหน่งของภาพนั้นจะอยู่บน VideoOCX แต่ในส่วนของ VideoOCX นั้นไม่ได้สร้างมาเพื่อบอกตำแหน่งเมื่อมีการดึงภาพขึ้นหน้าจอแล้ว จึงต้องมีการเลื่อนเมาส์ไปในส่วนของ VideoOCX ที่ไม่ใช่ส่วนของการแสดงภาพ เพื่อให้โปรแกรมได้รู้ว่าเราต้องการให้กล้องหมุน โปรแกรมจะทำการหยุดภาพเพื่อรอรับ Message การปล่อยเมาส์
7. Message การปล่อยเมาส์ ซึ่งเราจะใช้ก็ต่อเมื่อเราต้องการระบุตำแหน่งโดยเมื่อเรากดและปล่อยเมาส์แล้ว ก็จะทำการแปลงค่าพิกัดของตำแหน่งเมาส์และส่งผ่านทางพอร์ตอนุกรมที่เราทำการเปิดไว้ก่อนหน้าแล้ว และโปรแกรมจะทำการเรียกฟังก์ชันการดึงภาพขึ้นมาอีกครั้ง



Flowchart แสดงการทำงานของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### Flowchart แสดงการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### วงจรชุดควบคุมกล้อง

พอร์ตการสื่อสารอนุกรมจะถูกป้อนเข้าที่ขา TXD จะเป็นตัวส่งข้อมูลไปที่คอมพิวเตอร์ เพื่อให้โปรแกรมการจับภาพเปิดขึ้นมา ขา RXD เพื่อรับข้อมูลพิกัดจากคอมพิวเตอร์ เข้าสู่ AT89C51 แล้ว เมื่อประมวลผลแล้วจะต้องให้มอเตอร์หมุนที่สเต็ป ค่าที่ได้จะถูกส่งออกมาทางพอร์ต 2 โดย P 2.0 – P2.3 ควบคุมมอเตอร์หมุนซ้าย-ขวา P2.4-P2.7 ควบคุมมอเตอร์หมุนขึ้น-ลง

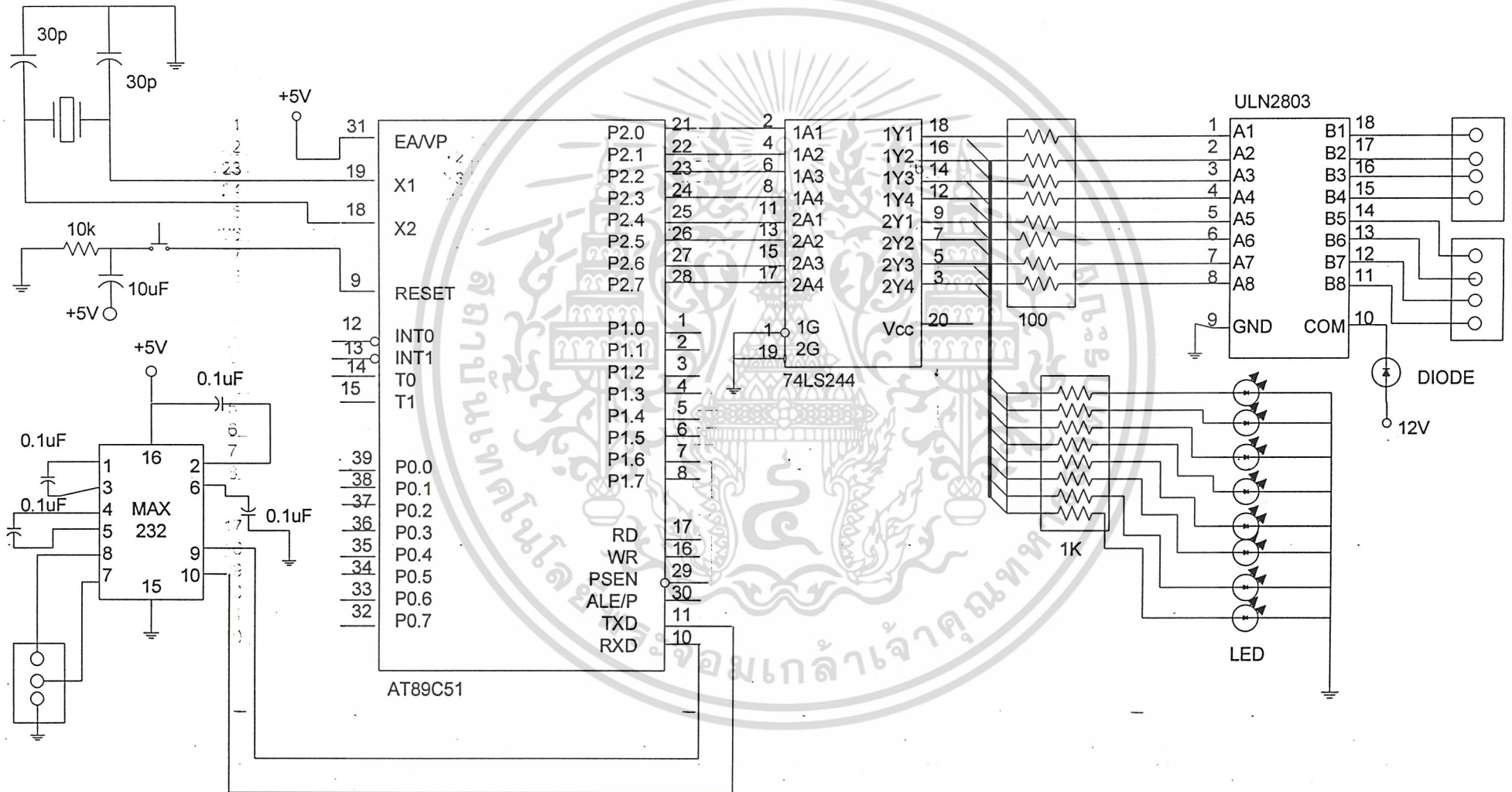
### วงจรตรวจจับสัญญาณ

วงจรถูกส่งจะใช้ไอซีเบอร์ 74HC00 ซึ่งโครงสร้างภายในเป็นแนบคู่เกตต่อกัน 4 ตัว สัญญาณเอาท์พุทที่ได้จะเป็นพัลส์สลับกันไปมาระหว่าง HIGH กับ LOW ต่อเข้ากับขา คอลเลคเตอร์ ของ PN 2222A เพื่อขับอินฟาเรด ในวงจรนี้เราต้องป้อนความถี่ที่ 38KHz เพื่อที่ว่า สามารถส่งสัญญาณไปที่วงจรรับได้

วงจรรับใช้ไอซีเบอร์ TSOP 4838 เป็นไอซีสำเร็จรูปที่ภายในมีตัวดีมอดูเลทอยู่ภายใน ถ้า มีสิ่งกีดขวางระหว่างวงจรรับกับวงจรส่ง สัญญาณเอาท์พุทที่ได้จะเป็น LOW ถ้าไม่มีสิ่งกีดขวาง ระหว่างวงจรรับกับวงจรส่ง เอาท์พุทที่วัดได้จะมีสถานะ

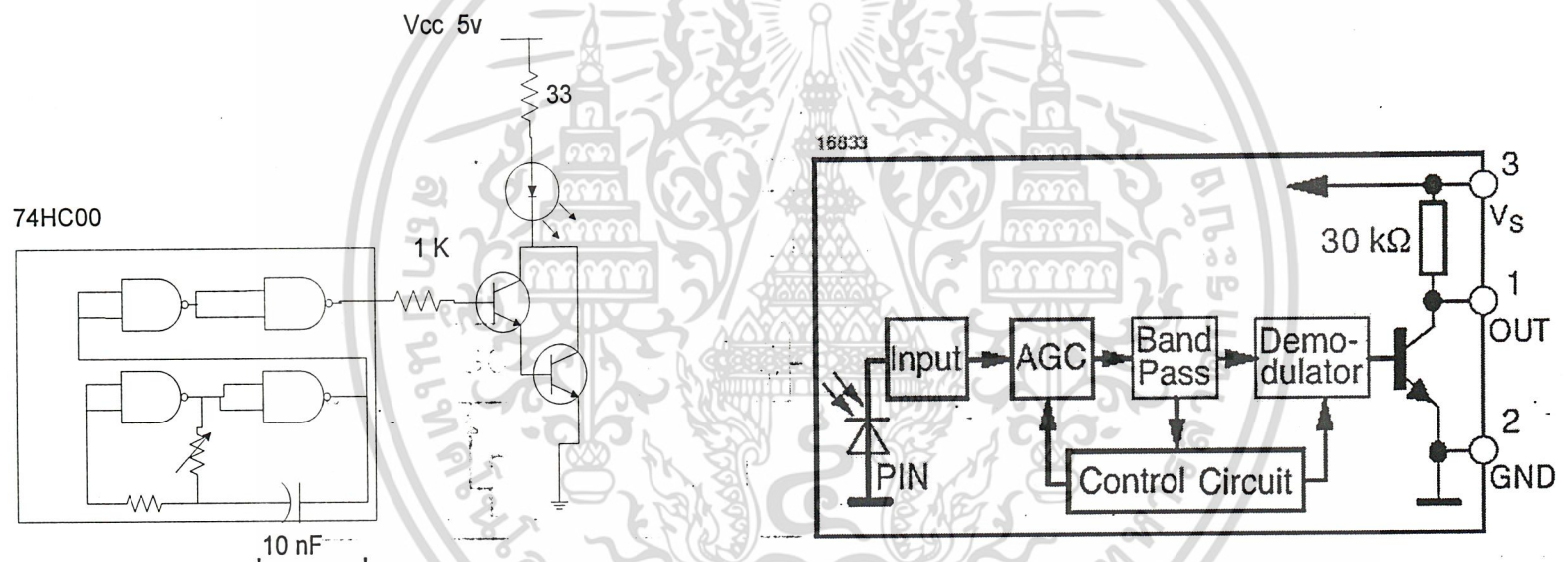
วงจรถูกส่งจะใช้ไอซีเบอร์ 74HC00 ซึ่งโครงสร้างภายในเป็นแนบคู่เกตต่อกัน 4 ตัว

ถ้ามีสิ่งกีดขวางระหว่างวงจรรับกับวงจรส่ง สัญญาณเอาท์พุทที่ได้จะเป็น LOW



รูปที่ 5.3 วงจรชุดควบคุมกล้อง

0.5v



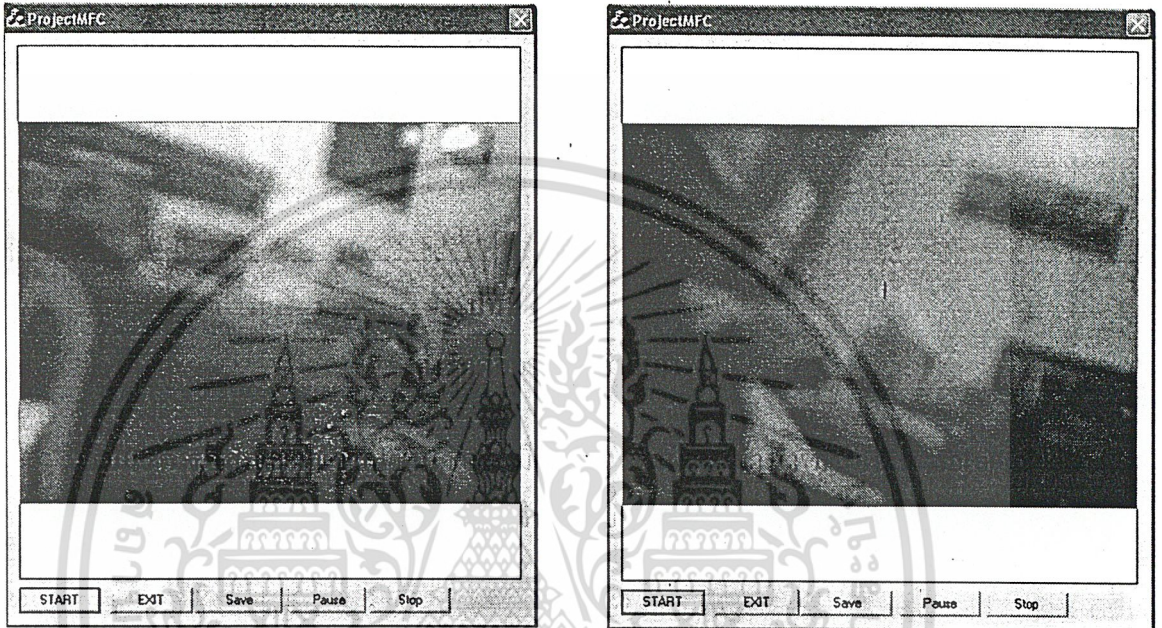
รูปที่ 5.4 วงจรส่งสัญญาณ (เซ็นเซอร์เพื่อเปิดโปรแกรม)

รูปที่ 5.5 วงจรรับสัญญาณ

## บทที่ 6

## การทดสอบประสิทธิภาพ

เริ่มแรก รอสัญญาณจากเซนเซอร์ เมื่อมีคนเดินผ่านเข้ามา มันจะส่งสัญญาณไปที่คอมพิวเตอร์ คอมพิวเตอร์ก็จะเปิดโปรแกรมการจับภาพขึ้นมา ดังรูป 6.1



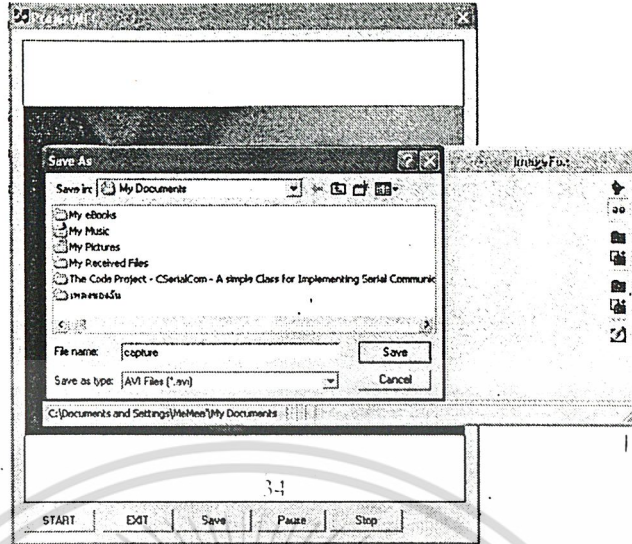
รูปที่ 6.1 โปรแกรมการจับภาพเปิด

รูปที่ 6.2 คลิกตำแหน่งที่ต้องการดู

เมื่อได้รูปปรากฏที่หน้าจอแล้ว หลังจากนั้นเราสามารถคลิกไปยังตำแหน่งที่เราต้องการดูภาพได้ ดังรูป 6.2

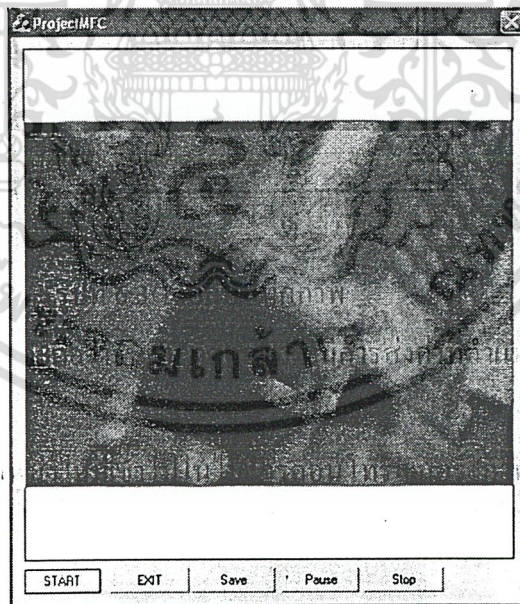
เราสามารถเลือกได้ว่าจะบันทึกภาพ ภาพนั้นหรือไม่

- กรณีเลือกบันทึกภาพ จะได้ดังรูป 6.3



รูปที่ 6.3 เลือกบันทึกภาพ

- กรณีเลือกไม่บันทึกภาพ จะถือว่าจบสิ้น ก็คือต่อไปจะมีการส่งค่าที่ตำแหน่งใหม่เข้ามาภายในเวลา 30 วินาทีหรือไม่
- ถ้าใช่ก็จะส่งค่าตำแหน่งใหม่เข้าไปในไมโครคอนโทรลเลอร์ ภาพที่หน้าจอก็จะเปลี่ยนตำแหน่ง ดังรูป 6.4

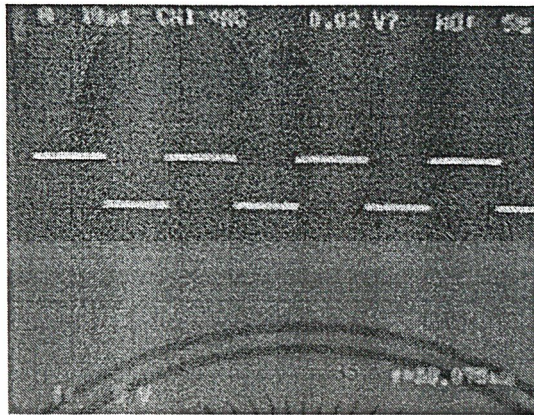


รูปที่ 6.4 คลิกที่ตำแหน่งใหม่

- ถ้าไม่มีการส่งข้อมูลตำแหน่งใหม่เข้ามา จะถือว่าเป็นการจบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของวงจรตรวจจับสัญญาณ  
ภาควงจรส่ง

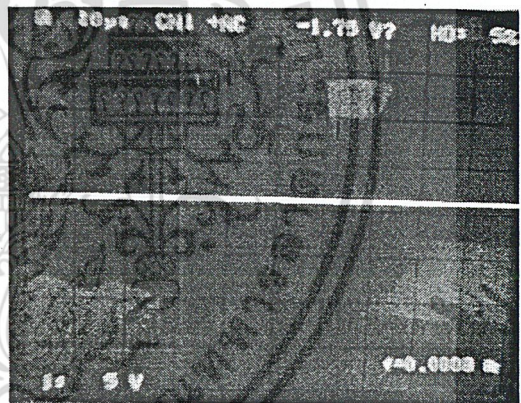
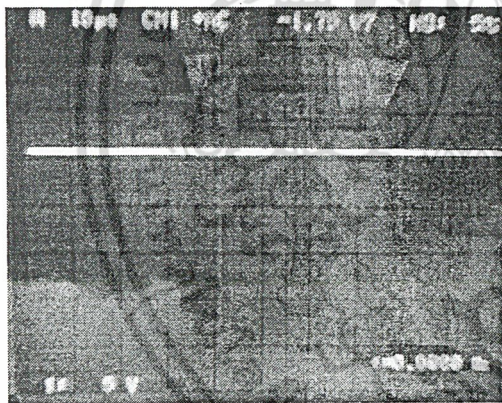


รูปที่ 6.5 สัญญาณในส่วนวงจรส่ง

ภาควงจรรับ

ขณะไม่มีสิ่งกีดขวาง

ขณะมีสิ่งกีดขวาง



รูปที่ 6.6 สัญญาณภาครับ

รูปที่ 6.7 สัญญาณภาครับ

ในส่วนการหมุนของมอเตอร์

ใช้ LED เป็นตัวแสดงผล เราทดสอบค่าได้โดยใช้ HyperTerminal ใส่ค่าพิกัด XY ต่างๆ เข้าไป แล้วดูว่าเกิดการหมุนไปที่สเต็ป กำหนดค่าจุดศูนย์กลาง (8,7) ค่าที่เข้ามาหาร 3 โดยมีสเต็ป ดังนี้ หมุนซ้าย-ขวา 09H,08H,0CH,04H,06H,02H,03H,01H

หมุนขึ้น-ลง 090H,080H,0C0H,040H,060H,020H,030H,010H

| พิกัด (X,Y) | ค่าที่หารแล้ว | ซ้าย-ขวา | ขึ้น-ลง   | สเต็ป |
|-------------|---------------|----------|-----------|-------|
| 5,12        | 1,4           | ซ้าย 03H | ขึ้น 040H | 7,3   |
| 35,24       | 11,8          | ขวา 06H  | ขึ้น 010H | 3,1   |
| 25,36       | 8,12          | -        | ลง 060H   | -5    |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### สรุปผลและวิจารณ์

จากการทดลองพบว่าตำแหน่งต่างๆที่ได้คลิกกับภาพที่ปรากฏอาจจะผิดพลาดกัน ไปถึง อาจเนื่องมาจากมุมมองของกล้อง

ส่วนโปรแกรมบนหน้าจอคอมพิวเตอร์เมื่อมีการดึงภาพขึ้นหน้าจอแล้ว ต้องมีการเลื่อนเมาส์ไปในส่วนของ VideoOCX ที่ไม่ใช่ส่วนของการแสดงภาพ เพื่อให้โปรแกรมได้รู้ว่าเราต้องการให้กล้องหมุน โปรแกรมจะทำการหยุดภาพเพื่อรอรับ Massage การปล่อยเมาส์

โดยภาพรวมถือว่ากล้องทำงานตามที่เราได้ออกแบบไว้ คือเมื่อมีคนเดินผ่านจะทำให้โปรแกรมเปิดอัตโนมัติ เราสามารถเพิ่มลูกเล่นตรงใส่เสียงไปด้วยก็ได้ เพื่อผู้ใช้จะได้ทราบว่ามีคนเดินเข้ามา เมื่อโปรแกรมเปิดเราก็คลิกไปยังตำแหน่ง เช่น ขึ้นบนกับหมุนซ้าย จะพบว่ากล้องจะหมุนไปทางนั้นเช่นกัน จุดที่เพิ่มเติมอาจจะทำโดยสร้างแผนผังห้องที่เราจะควบคุมไว้อีกบล็อกหนึ่ง เพื่อจะสะดวกต่อการที่เราคลิกว่าต้องการดู ณ ตำแหน่งไหน ก็คลิกไปที่ตำแหน่งนั้นกล้องก็ชี้ภาพไปยังตำแหน่งนั้นทันที



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

พีรภัทร์ สว่างเพ็ชร , เทคนิคการเขียน โปรแกรมและเกมด้วย Visual c++ , กรุงเทพฯ: ซีเอ็ดยูเคชั่น , 2545

นิรุช อำนวยศิลป์ ,เขียนเกมอย่างมืออาชีพ Visual c++ และ DirectX ,นนทบุรี : อินโฟเฟรต , 2545  
วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล ,เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ,อินโนเวทีฟ เอ็กซ์เพอริเมนต์

ยุทธนา ลีลาสุวรรณกุล ,คู่มือการเขียน โปรแกรมด้วย Visual C++ 6.0 , นนทบุรี : อินโฟเฟรต , 2544



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี ขอขอบพระคุณอาจารย์ขนิษฐา แซ่ตั้ง ที่แนะนำสิ่งต่าง ๆ ให้แก่พวกเรา รวมทั้งรุ่นพี่ปริญญาโทและพี่หมีที่ช่วยคิดปัญหา และก็เพื่อน ๆ ในห้องโปรเจ็ค ถ้าปริญญาโทรุ่นนี้มีสิ่งทีผิดพลาดประการใด ก็ขออภัยมา ณ ที่นี้ ด้วย

-----  
พิชามณูษ์ เตชะวิจิตร

-----  
วิชุดา เตชะประเสริฐพร  
คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรม ส่วนควบคุมมอเตอร์

```

;*****
;Define Port $ Pin name
;*****
POINT_LR EQU 61H
POINT_UD EQU 62H
MSEC EQU 63H
SEC EQU 64H
MIN EQU 65H
ABUF EQU 66H ;-69H
CUR_LR EQU 70H
CUR_UD EQU 71H
TAR_UD EQU 72H
TAR_LR EQU 73H
SENSORBIT P1.0
SOUND BIT P1.1
;*****
ORG 0000H
LJMP START
ORG 00BH
LJMP TIMER

START: MOV TMOD,#21H
MOV SCON,#52H
MOV TH1,#0FDH
MOV TL1,#0FDH
MOV TH0,#0DCH
MOV TL0,#00H
MOV PCON,#0
SETB TR1
SETB ET0
SETB EA
SETB TR0

MOV POINT_LR,#0
MOV POINT_UD,#0

MOV MSEC,#0
MOV SEC,#0
MOV MIN,#0
MOV CUR_LR,#08
MOV CUR_UD,#07

S1: LCALL CHECK
S2: LCALL SERIAL
LCALL COMPUTE
CJNE A,#0,S2
SJMP S1

;*****FUNCTION*****
;CHECK SENSOR
;*****
CHECK: JB SENSOR,$
SETB SOUND
MOV A,#20
LCALL DELAY_MSEC
MOV DPTR,#HELLO
LCALL TX_TEXT
RET

;*****
;COMPUTE STEP
;*****
COMPUTE: MOV A,ABUF ;U-D
ANL A,#0FH
MOV R3,A
MOV A,ABUF+1
RL A
RL A
RL A
RL A
ANL A,#0F0H
ORL A,R3
MOV R3,A
MOV R2,#0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV R1,#0                                CJNE A,SEC,SERIAL1
LCALL DTOH                                POP 1
LCALL DIV                                  POP 7
MOV TAR_UD,DPL                             MOV A,#1
LCALL MOTOR_UD                             RET

SERIAL2: CLR RI
MOV A,ABUF+2 ;L-R                          MOV A,SBUF
ANL A,#0FH                                  MOV @R1,A
MOV R3,A                                    INC R1
MOV A,ABUF+3                                DJNZ R7,SERIAL1
RL A                                         POP 1
RL A                                         POP 7
RL A                                         MOV A,#0
RL A                                         RET
ANL A,#0F0H                                ;*****
ORL A,R3                                    ;TX SERIAL TEXT
MOV R3,A                                    ;*****
MOV R2,#0                                  TX_TEXT: CLR TI
MOV R1,#0                                  TX_LOOP: CLR A
LCALL DTOH                                  MOV A,@A+DPTR
LCALL DIV                                  INC DPTR
MOV TAR_LR,DPL                              CJNE A,#0FFH,TX_CHAR
LCALL MOTOR_LR                              CLR SOUND
RET                                          RET
TX_CHAR: MOV SBUF,A
;*****
;* RECIEVE SERIAL
;*****
SERIAL: PUSH 7
PUSH 1
MOV R1,#ABUF
MOV R7,#4
MOV MSEC,#0
MOV SEC,#0

SERIAL1: JB RI,SERIAL2
MOV A,#15

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOTOR_R: INC CUR_LR MOV POINT_UD,#7
MOV A,POINT_LR SJMP MOTOR_UD2
CJNE A,#0,MOTOR_R1 MOTOR_U1: DEC POINT_UD
MOV POINT_LR,#7 SJMP MOTOR_UD2
SJMP MOTOR_LR2

MOTOR_R1: DEC POINT_LR MOTOR_D: DEC CUR_UD
SJMP MOTOR_LR2 MOV A,POINT_UD
CJNE A,#7,MOTOR_D1

MOTOR_L: DEC CUR_LR MOV POINT_UD,#0
MOV A,POINT_LR SJMP MOTOR_UD2
CJNE A,#7,MOTOR_L1 MOTOR_D1: INC POINT_UD
MOV POINT_LR,#0 MOTOR_UD2: MOV DPTR,#UD
SJMP MOTOR_LR2 MOV A,POINT_UD

MOTOR_L1: INC POINT_LR MOVC A,@A+DPTR
ANL P2,#0FH

MOTOR_LR2: MOV DPTR,#LR ORL P2,A
MOV A,POINT_LR MOV A,#7
MOVC A,@A+DPTR LCALL DELAY_MSEC
ANL P2,#0F0H SJMP MOTOR_UD
ORL P2,A
MOV A,#7 ;*****
LCALL DELAY_MSEC ;DELAY MSEC
SJMP MOTOR_LR ;*****

;*****
MOTOR_UD: MOV A,TAR_UD ;DELAY SEC
CJNE A,CUR_UD,MOTOR_UD1 ;*****
RET DELAY_SEC: MOV MSEC,#0
CJNE A,MSEC,$
RET

MOTOR_UD1: SUBB A,CUR_UD MOV SEC,#0
JC MOTOR_D CJNE A,SEC,$
RET

MOTOR_U: INC CUR_UD
MOV A,POINT_UD
CJNE A,#0,MOTOR_U1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\*\*\*\*\*

;\* DECIMAL TO HEX \*

\*\*\*\*\*

```
DTOH:  MOV R4,#16
DTOH1: MOV R5,#3
        MOV R0,#1
        CLR C
DTOH2: MOV A,@R0
        RRC A
        PUSH PSW
        JNB ACC.7,DTOH3
        CLR C
        SUBB A,#30h
DTOH3: JNB ACC.3,DTOH4
        CLR C
        SUBB A,#3
DTOH4: MOV @R0,A
        INC R0
        POP PSW
        DJNZ R5,DTOH2
        MOV A,DPH
        RRC A
        MOV DPH,A
        MOV A,DPL
        RRC A
        MOV DPL,A
        DJNZ R4,DTOH1
        RET
```

\*\*\*\*\*

;CAL STEP

\*\*\*\*\*

```
DIV:   MOV B,#03
        DIV AB
        MOV DPL,A
        RET
```

\*\*\*\*\*

;DELAY

\*\*\*\*\*

```
TIMER: PUSH PSW
        PUSH ACC
        INC MSEC ;10 MSEC
        MOV A,MSEC
        CJNE A,#100,TIMER_EXT
        MOV MSEC,#0
        INC SEC
        MOV A,SEC
        CJNE A,#60,TIMER_EXT
        MOV MSEC,#0
        INC MIN
TIMER_EXT: MOV TH0,#0DCH
          MOV TL0,#00H
          SETB TR0
          POP ACC
          POP PSW
          RETI
```

\*\*\*\*\*

;DEFINE CONSTANT

\*\*\*\*\*

```
HELLO: DB 'B'
        DB 0FFH
LR:     DB
        09H,08H,0CH,04H,06H,02H,03H,01H
UD:     DB
        090H,080H,0C0H,040H,060H,020H,030H,010H
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนของคอมพิวเตอร์

```
// ProjectMFCDlg.cpp : implementation file
//
#include "stdafx.h"
#include "ProjectMFC.h"
#include "ProjectMFCDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

class __declspec(dllexport) CSerialComm();

//////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
   //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA

// ClassWizard generated virtual function overrides
   //{{AFX_VIRTUAL(CAboutDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

    //}}AFX_VIRTUAL

// Implementation
protected:

    //{{AFX_MSG(CAboutDlg)
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
   //{{AFX_DATA_INIT(CAboutDlg)
   //{{AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CAboutDlg)
   //{{AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
   //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()
//////////////////////////////////////
// CProjectMFCDlg dialog
CProjectMFCDlg::CProjectMFCDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CProjectMFCDlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CProjectMFCDlg)
    m_ControlEvent = _T("");
   //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CProjectMFCDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CProjectMFCDlg)
    DDX_Control(pDX, IDC_VIDEOOXCCTRL1, m_Video);
    DDX_Text(pDX, IDC_EDIT1, m_ControlEvent);
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CProjectMFCDlg, CDialog)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//{{AFX_MSG_MAP(CProjectMFCDlg)
ON_WM_SYSCOMMAND()
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_WM_LBUTTONDOWN()
ON_BN_CLICKED(IDC_BUTTON1, OnRECORD)
ON_BN_CLICKED(IDC_BUTTON2, Onstop)
//}}AFX_MSG_MAP

END_MESSAGE_MAP()

////////////////////////////////////

// CProjectMFCDlg message handlers
BOOL CProjectMFCDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    // Add "About..." menu item to system menu.
    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);
    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);           // Set big icon
    SetIcon(m_hIcon, FALSE);        // Set small icon

    // TODO: Add extra initialization here

    m_Image = 0;
    m_ResultImage = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_HelperImage = 0;
m_Running = FALSE;
check_hide = false;
port.OpenPort("COM2");
port.ConfigurePort(CBR_9600,8,true,NOPARITY,ONESTOPBIT);
port.SetCommunicationTimeouts(0,500,0,0,0);
return TRUE; // return TRUE unless you set the focus to a control
}

void CProjectMFCDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.
void CProjectMFCDlg::OnPaint()
{
    if(IsIconic())
    {
        CPaintDC dc(this); // device context for painting
        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);
        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CProjectMFCDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

UINT CaptureThread(LPVOID pParam)
{
    CProjectMFCDlg *parent;
    //int r_value,g_value,b_value;
    int max_red = 0;
    int max_green = 0;
    int max_blue = 0;
    parent = (CProjectMFCDlg *)pParam;
    while (parent->m_Running){
        parent->m_Video.Capture(parent->m_Image);
        BYTE *data = (BYTE*)parent->m_Video.GetDataPointer(parent->m_Image);
        int size = parent->m_Video.GetImageDataSize(parent->m_Image);
        /*
        for (int i = 0; i < size; i++){
            switch (i % 3){
                case 0:
                    if (data[i] > max_blue)
                        max_blue = data[i];
                    break;
                case 1:
                    if (data[i] > max_green)
                        max_green = data[i];
                    break;
                case 2:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (data[i] > max_red)
            max_red = data[i];
        break;
    }
    if (i == (size - 1))
        //m_Red = ;
    }
}
/*
parent->m_Video.Show(parent->m_Image);
Sleep(5);
}
return(0);
}
void CProjectMFCDlg::OnOK()
{
/* // TODO: Add extra validation here
if (m_Image != 0)
    return;
// disable internal error message display
m_Video.SetErrorMessage(0);
if (!m_Video.Init())
{
    // Error while trying to init control
    MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error", MB_OK);
    return; // exit
}
// allocate memory for all images that will be needed to demonstrate VideoOCXTools - features
// Make sure you free the resources later using VideoOCX::ReleaseImageHandle(long).
m_Image = m_Video.GetColorImageHandle();
m_ResultImage = m_Video.GetColorImageHandle();
m_HelperImage = m_Video.GetColorImageHandle();
Sleep(100); // make sure VideoOCX has time to be properly initialized
if (m_Video.Start())
    m_Running = TRUE;
AfxBeginThread(CaptureThread, this);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//CDialog::OnOK();
}*/

if(!m_Video.Init()) // Init VideoOCX and
connect to the camera
{
// error in Init
MessageBox(m_Video.GetLastErrorString(),"VideoOCX Error", MB_OK |
MB_ICONEXCLAMATION);
return;
}

m_Video.SetPreview(TRUE); // Turn on Preview to see the
"live" image
}

void CProjectMFCDlg::OnCancel()
{
// TODO: Add extra cleanup here
m_Running = FALSE; // tell thread to end
Sleep(250); // wait for thread to end
// end capture mode
if(!m_Video.Stop())
{
// error
MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error", MB_OK);
}

// free memory
m_Video.ReleaseImageHandle(m_Image);
m_Video.ReleaseImageHandle(m_ResultImage);
m_Video.ReleaseImageHandle(m_HelperImage);
m_Image = 0;
m_ResultImage = 0;
m_HelperImage = 0;
check_hide = false;
// close video connection and free memory
if(!m_Video.Close())
{
// error while closing video connection

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error", MB_OK);
    }

    CDialog::OnCancel();
}

void CProjectMFCDlg::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CDialog::OnLButtonDown(nFlags, point);
}

void CProjectMFCDlg::OnRECORD()
{
    if(!m_Video.AVISaveMovieInit("")) // Init Movie mode and ask for a filename
    {
        // error in AVISaveMovieInit
        MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error", MB_OK |
MB_ICONEXCLAMATION);
        return;
    }

    m_Video.SetAudio(TRUE); // Also record audio
    // Call m_VideoOCX.AVISaveMovieShowAudioDlg() to set the audio compression
    // Call m_VideoOCX.AVISaveMovieShowCompressionDlg() if you want to specify a video
compressor
    m_Video.AVISaveMovieSetFrameRate(15); // Lets save 15 frames per second
    // Try this instead
    //m_VideoOCX.AVISaveMovieSetFrameRate(0.1); // 1 Frame every 10 seconds (e.g. a simple
time-lapse recorder)

    if(!m_Video.AVISaveMovieStart()) // Start recording
    {
        // error in AVISaveMovieStart
        MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error", MB_OK |
MB_ICONEXCLAMATION);
        return;
    }
}

void CProjectMFCDlg::Onstop()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(!m_Video.AVISaveMovieStop()) // end movie capture
{
    // error in Stop
    MessageBox(m_Video.GetLastErrorString(),"VideoOCX Error", MB_OK |
    MB_ICONEXCLAMATION);
    OnOK();
}
}
BEGIN_EVENTSINK_MAP(CProjectMFCDlg, CDialog)
//{{AFX_EVENTSINK_MAP(CProjectMFCDlg)
    ON_EVENT(CProjectMFCDlg, IDC_VIDEOOCXCTRL1, -605 /* MouseDown */,
    OnMouseDownVideooctrl1, VTS_I2 VTS_I2 VTS_I4 VTS_I4)
//}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()
/*void CProjectMFCDlg::OnMouseDownVideooctrl1(short Button, short Shift, long x, long y)
{
switch(Button)
{
case 1:
    m_ControlEvent.Format("Left mouse Button Down at (%d,%d)", x, y);
    break;
case 2:
    m_ControlEvent.Format("Right mouse Button Down at (%d,%d)", x, y);
    break;
default:
    m_ControlEvent.Format("Mouse Button Down at (%d,%d)", x, y);
    break;
}

int z,a,b,c,d,e,f,g;
// POINT ptCurrentCursor;
// GetCursorPos(&ptCurrentCursor);
// x=ptCurrentCursor.x;
// y=ptCurrentCursor.y;
z=x;
if (z>=1000)
{
a=z/1000;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        b=a*1000;
        z=z-b;
        a=a+48;
    }
else
        a=48;

if (z>=100)
{
    c=z/100;
    d=c*100;
    z=z-d;
    c=c+48;
}else
    c=48;
if (z>=10)
{
    e=z/10;
    f=e*10;
    z=z-f;
    e=e+48;
}else
    e=48;
    g=z;
    g=g+48;

//port.WriteByte(a);
port.WriteByte(c);
//port.WriteByte(e);
//port.WriteByte(g);

z=y;

if (z>=1000)
{
    a=z/1000;
    b=a*1000;
    z=z-b;
    a=a+48;
}
else
    a=48;

if (z>=100)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            c=z/100;

            d=c*100;

            z=z-d;

            c=c+48;

        }else

            c=48;

        if (z>=10)

        {
            e=z/10;

            f=e*10;

            z=z-f;

            e=e+48;

        }else

            e=48;

        g=z;

        g=g+48;

//port.WriteByte(a);
port.WriteByte(c);
//port.WriteByte(e);
//port.WriteByte(g);

// Update Dialog:
UpdateData(FALSE);
}*/

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS

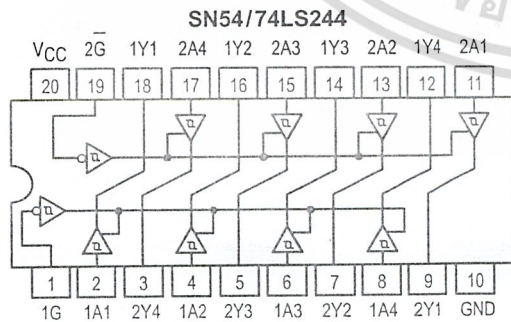
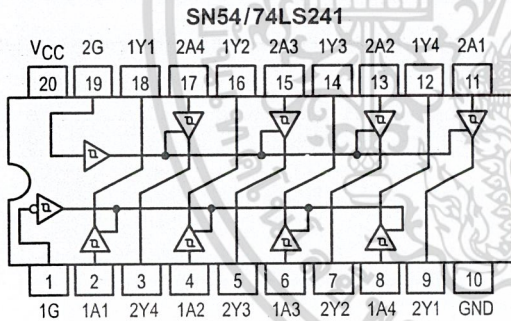
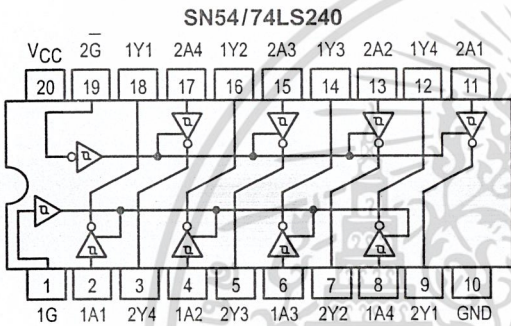
The SN54/74LS240, 241 and 244 are Octal Buffers and Line Drivers designed to be employed as memory address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC board density.

- Hysteresis at Inputs to Improve Noise Margins
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- Input Clamp Diodes Limit High-Speed Termination Effects

**SN54/74LS240**  
**SN54/74LS241**  
**SN54/74LS244**

**OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS**  
**LOW POWER SCHOTTKY**

## LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)



**J SUFFIX CERAMIC CASE 732-03**

**N SUFFIX PLASTIC CASE 738-03**

**DW SUFFIX SOIC CASE 751D-03**

**ORDERING INFORMATION**

|             |         |
|-------------|---------|
| SN54LSXXXJ  | Ceramic |
| SN74LSXXXN  | Plastic |
| SN74LSXXXDW | SOIC    |

FAST AND LS TTL DATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# SN54/74LS240 • SN54/74LS241 • SN54/74LS244

## TRUTH TABLES

SN54/74LS240

| INPUTS |   | OUTPUT |
|--------|---|--------|
| 1G, 2G | D |        |
| L      | L | H      |
| L      | H | L      |
| H      | X | (Z)    |

SN54/74LS244

| INPUTS |   | OUTPUT |
|--------|---|--------|
| 1G, 2G | D |        |
| L      | L | L      |
| L      | H | H      |
| H      | X | (Z)    |

SN54/74LS241

| INPUTS |   | OUTPUT | INPUTS |   | OUTPUT |
|--------|---|--------|--------|---|--------|
| 1G     | D |        | 2G     | D |        |
| L      | L | L      | H      | L | L      |
| L      | H | H      | H      | H | H      |
| H      | X | (Z)    | L      | X | (Z)    |

H = HIGH Voltage Level  
 L = LOW Voltage Level  
 X = Immaterial  
 Z = HIGH Impedance

## GUARANTEED OPERATING RANGES

| Symbol | Parameter                           | Min    | Typ  | Max        | Unit |
|--------|-------------------------------------|--------|------|------------|------|
| VCC    | Supply Voltage                      | 54     | 4.5  | 5.0        | 5.5  |
|        |                                     | 74     | 4.75 | 5.0        | 5.25 |
| TA     | Operating Ambient Temperature Range | 54     | -55  | 25         | 125  |
|        |                                     | 74     | 0    | 25         | 70   |
| IOH    | Output Current — High               | 54, 74 |      | -3.0       | mA   |
|        |                                     | 54, 74 |      | -12<br>-15 | mA   |
| IOL    | Output Current — Low                | 54     |      | 12         | mA   |
|        |                                     | 74     |      | 24         | mA   |

SN54/74LS240 • SN54/74LS241 • SN54/74LS244

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

| Symbol                           | Parameter                               | Limits    |       |      | Unit | Test Conditions   |
|----------------------------------|---|-----------|-------|------|------|---|
|                                  |   | Min       | Typ   | Max  |      |   |
| V <sub>IH</sub>                  | Input HIGH Voltage                      | 2.0       |       |      | V    | Guaranteed Input HIGH Voltage for All Inputs  |
| V <sub>IL</sub>                  | Input LOW Voltage                       | 54        |       | 0.7  | V    | Guaranteed Input LOW Voltage for All Inputs   |
|                                  |   | 74        |       | 0.8  |      |   |
| V <sub>T+</sub> -V <sub>T-</sub> | Hysteresis                              | 0.2       | 0.4   |      | V    | V <sub>CC</sub> = MIN   |
| V <sub>IK</sub>                  | Input Clamp Diode Voltage               |           | -0.65 | -1.5 | V    | V <sub>CC</sub> = MIN, I <sub>IN</sub> = -18 mA   |
| V <sub>OH</sub>                  | Output HIGH Voltage                     | 54, 74    | 2.4   | 3.4  | V    | V <sub>CC</sub> = MIN, I <sub>OH</sub> = -3.0 mA  |
|                                  |   | 54, 74    | 2.0   |      | V    | V <sub>CC</sub> = MIN, I <sub>OH</sub> = MAX  |
| V <sub>OL</sub>                  | Output LOW Voltage                      | 54, 74    | 0.25  | 0.4  | V    | I <sub>OL</sub> = 12 mA   |
|                                  |   | 74        | 0.35  | 0.5  | V    | I <sub>OL</sub> = 24 mA   |
|                                  |   |           |       |      |      | V <sub>CC</sub> = V <sub>CC</sub> MIN, V <sub>IN</sub> = V <sub>IL</sub> or V <sub>IH</sub> per Truth Table |
| I <sub>OZH</sub>                 | Output Off Current HIGH                 |           |       | 20   | μA   | V <sub>CC</sub> = MAX, V <sub>OUT</sub> = 2.7 V   |
| I <sub>OZL</sub>                 | Output Off Current LOW                  |           |       | -20  | μA   | V <sub>CC</sub> = MAX, V <sub>OUT</sub> = 0.4 V   |
| I <sub>IH</sub>                  | Input HIGH Current                      |           |       | 20   | μA   | V <sub>CC</sub> = MAX, V <sub>IN</sub> = 2.7 V  |
|                                  |   |           |       | 0.1  | mA   | V <sub>CC</sub> = MAX, V <sub>IN</sub> = 7.0 V  |
| I <sub>IL</sub>                  | Input LOW Current                       |           |       | -0.2 | mA   | V <sub>CC</sub> = MAX, V <sub>IN</sub> = 0.4 V  |
| I <sub>OS</sub>                  | Output Short Circuit Current (Note 1)   | -40       |       | -225 | mA   | V <sub>CC</sub> = MAX   |
| I <sub>CC</sub>                  | Power Supply Current Total, Output HIGH |           |       | 27   | mA   | V <sub>CC</sub> = MAX   |
|                                  | Total, Output LOW                       | LS240     |       | 44   |      |   |
|                                  |   | LS241/244 |       | 46   |      |   |
|                                  | Total at HIGH Z                         | LS240     |       | 50   |      |   |
|                                  |   | LS241/244 |       | 54   |      |   |

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T<sub>A</sub> = 25°C, V<sub>CC</sub> = 5.0 V)

| Symbol                               | Parameter                                   | Limits |           |          | Unit | Test Conditions                                    |
|--------------------------------------|---|--------|-----------|----------|------|--|
|                                      |   | Min    | Typ       | Max      |      |  |
| t <sub>PLH</sub><br>t <sub>PHL</sub> | Propagation Delay, Data to Output LS240     |        | 9.0<br>12 | 14<br>18 | ns   | C <sub>L</sub> = 45 pF,<br>R <sub>L</sub> = 667 Ω  |
| t <sub>PLH</sub><br>t <sub>PHL</sub> | Propagation Delay, Data to Output LS241/244 |        | 12<br>12  | 18<br>18 | ns   |  |
| t <sub>PZH</sub>                     | Output Enable Time to HIGH Level            |        | 15        | 23       | ns   |  |
| t <sub>PZL</sub>                     | Output Enable Time to LOW Level             |        | 20        | 30       | ns   |  |
| t <sub>PLZ</sub>                     | Output Disable Time from LOW Level          |        | 15        | 25       | ns   |  |
| t <sub>FHZ</sub>                     | Output Disable Time from HIGH Level         |        | 10        | 18       | ns   | C <sub>L</sub> = 5.0 pF,<br>R <sub>L</sub> = 667 Ω |

FAST AND I S TTL DATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC WAVEFORMS

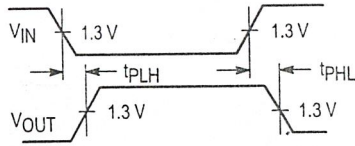


Figure 1

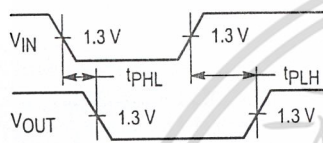


Figure 2

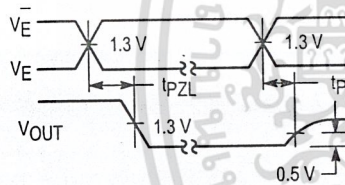


Figure 3

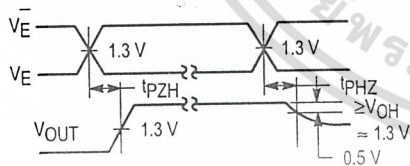
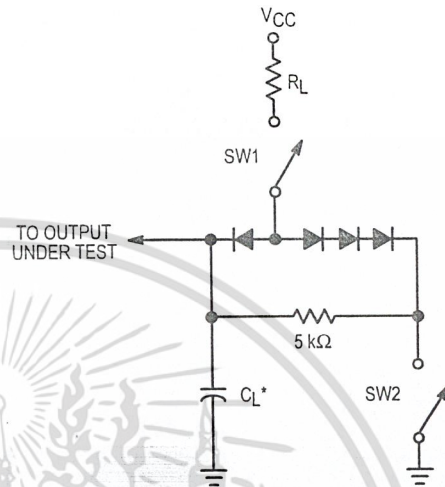


Figure 4



SWITCH POSITIONS

| SYMBOL           | SW1    | SW2    |
|------------------|--------|--------|
| t <sub>PZH</sub> | Open   | Closed |
| t <sub>PZL</sub> | Closed | Open   |
| t <sub>PLZ</sub> | Closed | Closed |
| t <sub>PHZ</sub> | Closed | Closed |

Figure 5



# Octal High Voltage, High Current Darlington Transistor Arrays

The eight NPN Darlington connected transistors in this family of arrays are ideally suited for interfacing between low logic level digital circuitry (such as TTL, CMOS or PMOS/NMOS) and the higher current/voltage requirements of lamps, relays, printer hammers or other similar loads for a broad range of computer, industrial, and consumer applications. All devices feature open-collector outputs and free wheeling clamp diodes for transient suppression.

The ULN2803 is designed to be compatible with standard TTL families while the ULN2804 is optimized for 6 to 15 volt high level CMOS or PMOS.

**MAXIMUM RATINGS** ( $T_A = 25^\circ\text{C}$  and rating apply to any one device in the package, unless otherwise noted.)

| Rating                              | Symbol    | Value       | Unit             |
|-------------------------------------|-----------|-------------|------------------|
| Output Voltage                      | $V_O$     | 50          | V                |
| Input Voltage (Except ULN2801)      | $V_I$     | 30          | V                |
| Collector Current – Continuous      | $I_C$     | 500         | mA               |
| Base Current – Continuous           | $I_B$     | 25          | mA               |
| Operating Ambient Temperature Range | $T_A$     | 0 to +70    | $^\circ\text{C}$ |
| Storage Temperature Range           | $T_{stg}$ | -55 to +150 | $^\circ\text{C}$ |
| Junction Temperature                | $T_J$     | 125         | $^\circ\text{C}$ |

$R_{\theta JA} = 55^\circ\text{C/W}$   
Do not exceed maximum current limit per driver.

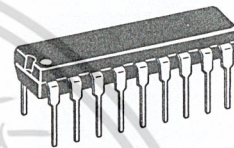
### ORDERING INFORMATION

| Device   | Characteristics      |                                      |   |
|----------|----------------------|--------------------------------------|---|
|          | Input Compatibility  | $V_{CE}(\text{Max})/I_C(\text{Max})$ | Operating Temperature Range             |
| ULN2803A | TTL, 5.0 V CMOS      | 50 V/500 mA                          | $T_A = 0 \text{ to } +70^\circ\text{C}$ |
| ULN2804A | 6 to 15 V CMOS, PMOS |                                      |   |

## ULN2803 ULN2804

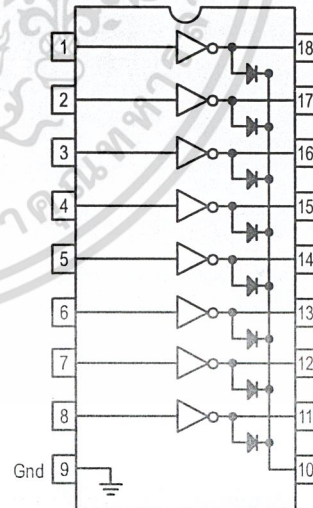
### OCTAL PERIPHERAL DRIVER ARRAYS

#### SEMICONDUCTOR TECHNICAL DATA



A SUFFIX  
PLASTIC PACKAGE  
CASE 707

### PIN CONNECTIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ULN2803 ULN2804

### ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ , unless otherwise noted)

| Characteristic  |  | Symbol        | Min                                  | Typ                                  | Max   | Unit          |
|---|--|---------------|--------------------------------------|--------------------------------------|---|---------------|
| Output Leakage Current (Figure 1)<br>( $V_O = 50\text{ V}$ , $T_A = +70^\circ\text{C}$ )<br>( $V_O = 50\text{ V}$ , $T_A = +25^\circ\text{C}$ )<br>( $V_O = 50\text{ V}$ , $T_A = +70^\circ\text{C}$ , $V_I = 6.0\text{ V}$ )<br>( $V_O = 50\text{ V}$ , $T_A = +70^\circ\text{C}$ , $V_I = 1.0\text{ V}$ )   | All Types<br>All Types<br>ULN2802<br>ULN2804   | $I_{CEX}$     | –<br>–<br>–<br>–                     | –<br>–<br>–<br>–                     | 100<br>50<br>500<br>500                             | $\mu\text{A}$ |
| Collector–Emitter Saturation Voltage (Figure 2)<br>( $I_C = 350\text{ mA}$ , $I_B = 500\text{ }\mu\text{A}$ )<br>( $I_C = 200\text{ mA}$ , $I_B = 350\text{ }\mu\text{A}$ )<br>( $I_C = 100\text{ mA}$ , $I_B = 250\text{ }\mu\text{A}$ )   | All Types<br>All Types<br>All Types  | $V_{CE(sat)}$ | –<br>–<br>–                          | 1.1<br>0.95<br>0.85                  | 1.6<br>1.3<br>1.1                                   | V             |
| Input Current – On Condition (Figure 4)<br>( $V_I = 17\text{ V}$ )<br>( $V_I = 3.85\text{ V}$ )<br>( $V_I = 5.0\text{ V}$ )<br>( $V_I = 12\text{ V}$ )  | ULN2802<br>ULN2803<br>ULN2804<br>ULN2804   | $I_{I(on)}$   | –<br>–<br>–<br>–                     | 0.82<br>0.93<br>0.35<br>1.0          | 1.25<br>1.35<br>0.5<br>1.45                         | mA            |
| Input Voltage – On Condition (Figure 5)<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 300\text{ mA}$ )<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 200\text{ mA}$ )<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 250\text{ mA}$ )<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 300\text{ mA}$ )<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 125\text{ mA}$ )<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 200\text{ mA}$ )<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 275\text{ mA}$ )<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 350\text{ mA}$ ) | ULN2802<br>ULN2803<br>ULN2803<br>ULN2803<br>ULN2804<br>ULN2804<br>ULN2804<br>ULN2804 | $V_{I(on)}$   | –<br>–<br>–<br>–<br>–<br>–<br>–<br>– | –<br>–<br>–<br>–<br>–<br>–<br>–<br>– | 13<br>2.4<br>2.7<br>3.0<br>5.0<br>6.0<br>7.0<br>8.0 | V             |
| Input Current – Off Condition (Figure 3)<br>( $I_C = 500\text{ }\mu\text{A}$ , $T_A = +70^\circ\text{C}$ )  | All Types  | $I_{I(off)}$  | 50                                   | 100                                  | –   | $\mu\text{A}$ |
| DC Current Gain (Figure 2)<br>( $V_{CE} = 2.0\text{ V}$ , $I_C = 350\text{ mA}$ )   | ULN2801  | $h_{FE}$      | 1000                                 | –                                    | –   | –             |
| Input Capacitance   |  | $C_I$         | –                                    | 15                                   | 25  | pF            |
| Turn–On Delay Time<br>(50% $E_I$ to 50% $E_O$ )   |  | $t_{on}$      | –                                    | 0.25                                 | 1.0   | $\mu\text{s}$ |
| Turn–Off Delay Time<br>(50% $E_I$ to 50% $E_O$ )  |  | $t_{off}$     | –                                    | 0.25                                 | 1.0   | $\mu\text{s}$ |
| Clamp Diode Leakage Current (Figure 6)<br>( $V_R = 50\text{ V}$ )   | $T_A = +25^\circ\text{C}$<br>$T_A = +70^\circ\text{C}$                               | $I_R$         | –<br>–                               | –<br>–                               | 50<br>100   | $\mu\text{A}$ |
| Clamp Diode Forward Voltage (Figure 7)<br>( $I_F = 350\text{ mA}$ )   |  | $V_F$         | –                                    | 1.5                                  | 2.0   | V             |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ULN2803 ULN2804

## TEST FIGURES

(See Figure Numbers in Electrical Characteristics Table)

Figure 1.

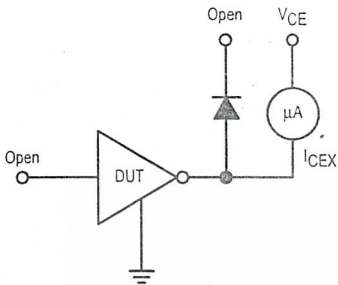


Figure 2.

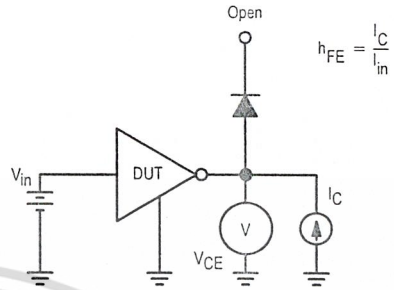


Figure 3.

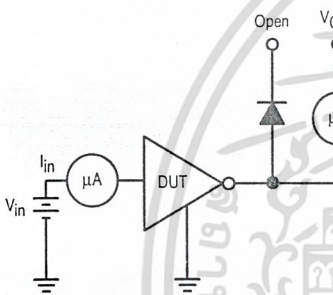


Figure 4.

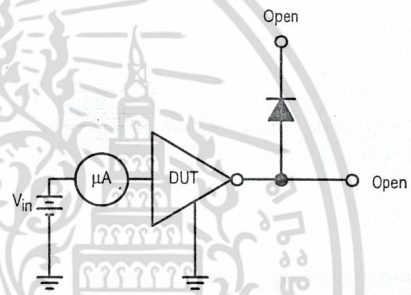


Figure 5.

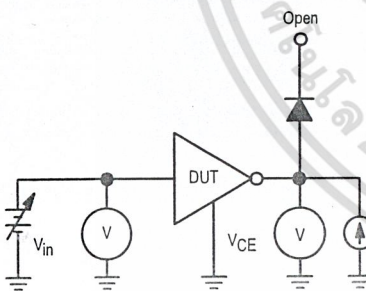


Figure 6.

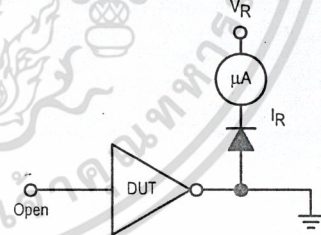
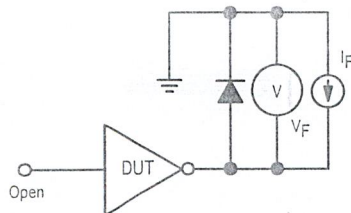


Figure 7.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ULN2803 ULN2804

TYPICAL CHARACTERISTIC CURVES –  $T_A = 25^\circ\text{C}$ , unless otherwise noted  
Output Characteristics

Figure 8. Output Current versus Saturation Voltage

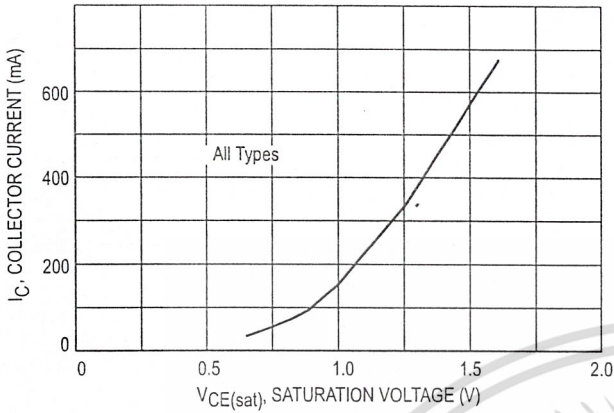
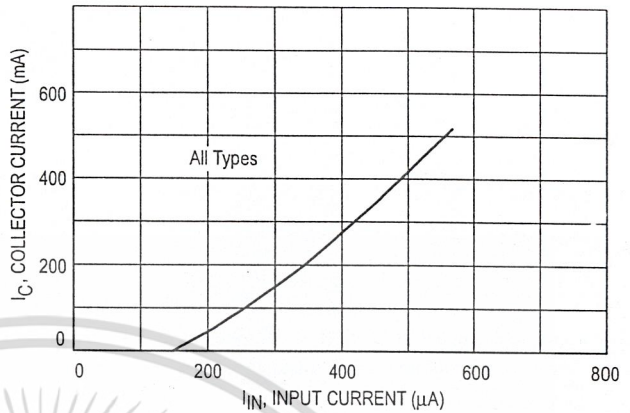


Figure 9. Output Current versus Input Current



## Input Characteristics

Figure 10. ULN2803 Input Current versus Input Voltage

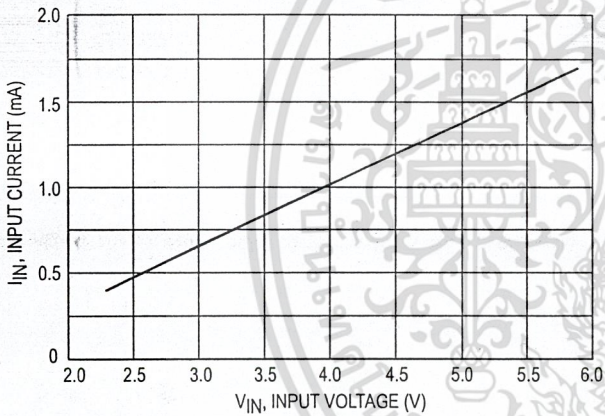


Figure 11. ULN2804 Input Current versus Input Voltage

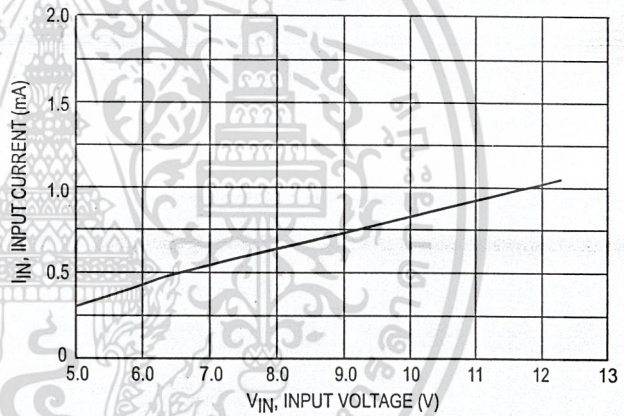
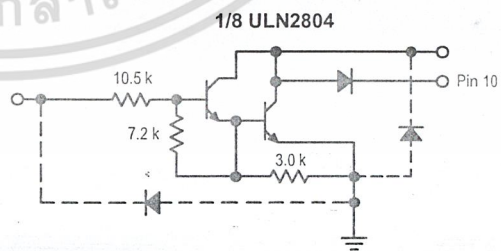
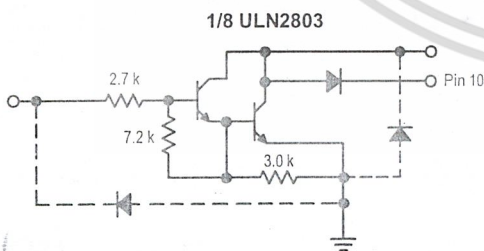


Figure 12. Representative Schematic Diagrams



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

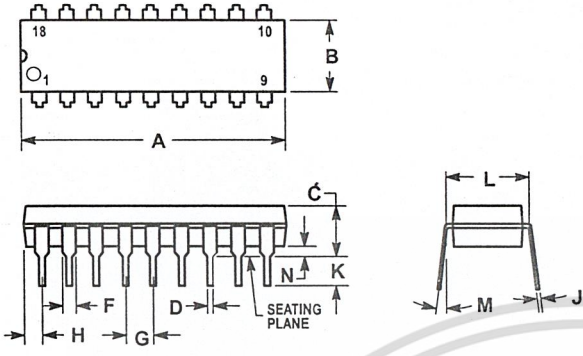
# ULN2803 ULN2804

## OUTLINE DIMENSIONS

A SUFFIX  
PLASTIC PACKAGE  
CASE 707-02  
ISSUE C

NOTES:

1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.



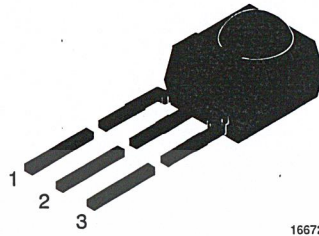
| DIM | MILLIMETERS |       | INCHES    |       |
|-----|-------------|-------|-----------|-------|
|     | MIN         | MAX   | MIN       | MAX   |
| A   | 22.22       | 23.24 | 0.875     | 0.915 |
| B   | 6.10        | 6.60  | 0.240     | 0.260 |
| C   | 3.56        | 4.57  | 0.140     | 0.180 |
| D   | 0.36        | 0.56  | 0.014     | 0.022 |
| F   | 1.27        | 1.78  | 0.050     | 0.070 |
| G   | 2.54 BSC    |       | 0.100 BSC |       |
| H   | 1.02        | 1.52  | 0.040     | 0.060 |
| J   | 0.20        | 0.30  | 0.008     | 0.012 |
| K   | 2.92        | 3.43  | 0.115     | 0.135 |
| L   | 7.62 BSC    |       | 0.300 BSC |       |
| M   | 0°          | 15°   | 0°        | 15°   |
| N   | 0.51        | 1.02  | 0.020     | 0.040 |

## IR Receiver Modules for Remote Control Systems

### Description

The TSOP48.. - series are miniaturized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter.

The demodulated output signal can directly be decoded by a microprocessor. TSOP48.. is the standard IR remote control receiver series, supporting all major transmission codes.



16672

### Features

- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Improved shielding against electrical field disturbance
- TTL and CMOS compatibility
- Output active low
- Low power consumption

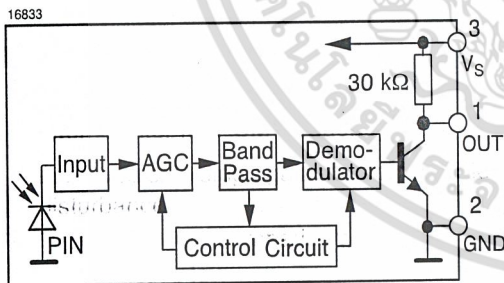
### Special Features

- Improved immunity against ambient light
- Suitable burst length  $\geq 10$  cycles/burst

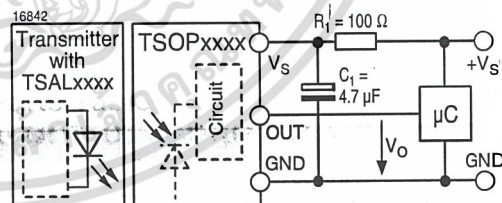
### Parts Table

|          |          |
|----------|----------|
| TSOP4830 | 30 kHz   |
| TSOP4833 | 33 kHz   |
| TSOP4836 | 36 kHz   |
| TSOP4837 | 36.7 kHz |
| TSOP4838 | 38 kHz   |
| TSOP4840 | 40 kHz   |
| TSOP4856 | 56 kHz   |

### Block Diagram



### Application Circuit



$R_1 + C_1$  recommended to suppress power supply disturbances.

The output voltage should not be hold continuously at a voltage below  $V_O = 3.3$  V by the external circuit.



### Absolute Maximum Ratings

$T_{amb} = 25\text{ }^{\circ}\text{C}$ , unless otherwise specified

|                             |   |           |                |                    |
|-----------------------------|---|-----------|----------------|--------------------|
| Supply Voltage              | (Pin 3)                                       | $V_S$     | - 0.3 to + 6.0 | V                  |
| Supply Current              | (Pin 3)                                       | $I_S$     | 5              | mA                 |
| Output Voltage              | (Pin 1)                                       | $V_O$     | - 0.3 to + 6.0 | V                  |
| Output Current              | (Pin 1)                                       | $I_O$     | 5              | mA                 |
| Junction Temperature        |   | $T_j$     | 100            | $^{\circ}\text{C}$ |
| Storage Temperature Range   |   | $T_{stg}$ | - 25 to + 85   | $^{\circ}\text{C}$ |
| Operating Temperature Range |   | $T_{amb}$ | - 25 to + 85   | $^{\circ}\text{C}$ |
| Power Consumption           | ( $T_{amb} \leq 85\text{ }^{\circ}\text{C}$ ) | $P_{tot}$ | 50             | mW                 |
| Soldering Temperature       | $t \leq 10\text{ s}$ , 1 mm from case         | $T_{sd}$  | 260            | $^{\circ}\text{C}$ |

### Electrical and Optical Characteristics

$T_{amb} = 25\text{ }^{\circ}\text{C}$ , unless otherwise specified

|                            |   |              |     |          |     |                 |
|----------------------------|---|--------------|-----|----------|-----|-----------------|
| Supply Current (Pin 3)     | $V_S = 5\text{ V}$ , $E_v = 0$  | $I_{SD}$     | 0.8 | 1.2      | 1.5 | mA              |
|                            | $V_S = 5\text{ V}$ , $E_v = 40\text{ klx}$ , sunlight                                     | $I_{SH}$     |     | 1.5      |     | mA              |
| Supply Voltage             |   | $V_S$        | 4.5 |          | 5.5 | V               |
| Transmission Distance      | $E_v = 0$ , test signal see fig.1, IR diode TSAL6200, $I_F = 250\text{ mA}$               | $d$          |     | 35       |     | m               |
| Output Voltage Low (Pin 1) | $I_{OSL} = 0.5\text{ mA}$ , $E_e = 0.7\text{ mW/m}^2$ , test signal see fig. 1            | $V_{OSL}$    |     |          | 250 | mV              |
| Irradiance (56 kHz)        | Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$ , test signal see fig.1 | $E_{e\ min}$ |     | 0.3      | 0.5 | $\text{mW/m}^2$ |
| Irradiance (30-40 kHz)     | Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$ , test signal see fig.1 | $E_{e\ min}$ |     | 0.2      | 0.4 | $\text{mW/m}^2$ |
| Irradiance                 | $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$ , test signal see fig. 1                       | $E_{e\ max}$ | 30  |          |     | $\text{W/m}^2$  |
| Directivity                | Angle of half transmission distance   | $\Phi_{1/2}$ |     | $\pm 45$ |     | deg             |

## Typical Characteristics ( $T_{amb} = 25\text{ }^{\circ}\text{C}$ unless otherwise specified)

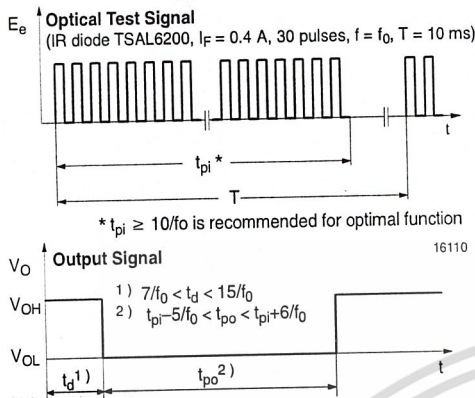


Figure 1. Output Function

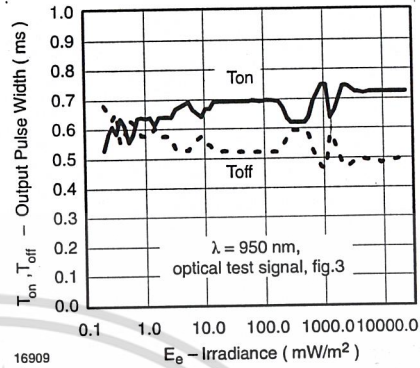


Figure 4. Output Pulse Diagram

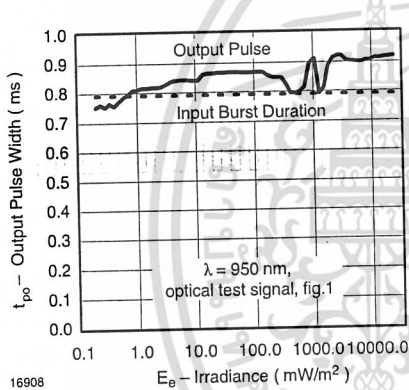


Figure 2. Pulse Length and Sensitivity in Dark Ambient

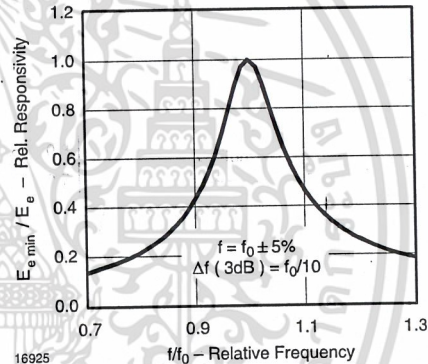


Figure 5. Frequency Dependence of Responsivity

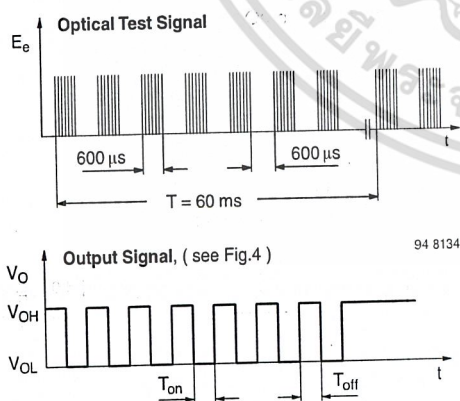


Figure 3. Output Function

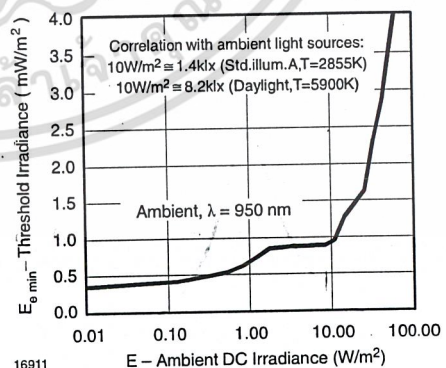


Figure 6. Sensitivity in Bright Ambient

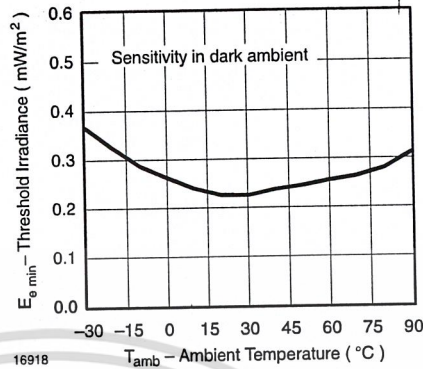
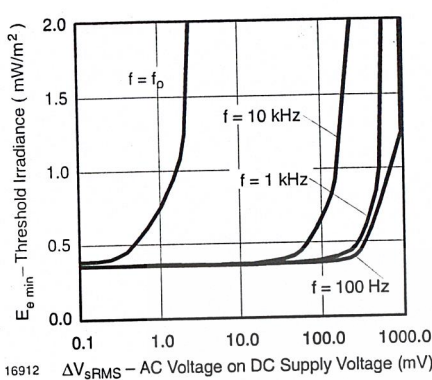


Figure 7. Sensitivity vs. Supply Voltage Disturbances

Figure 10. Sensitivity vs. Ambient Temperature

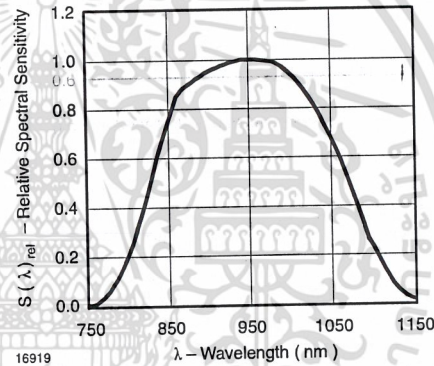
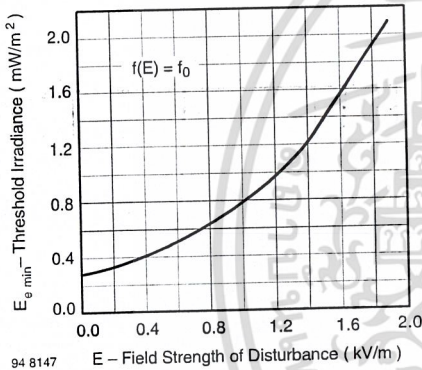


Figure 8. Sensitivity vs. Electric Field Disturbances

Figure 11. Relative Spectral Sensitivity vs. Wavelength

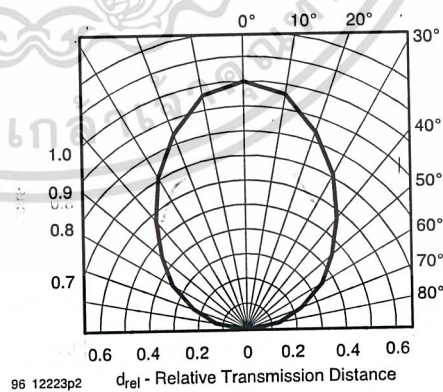
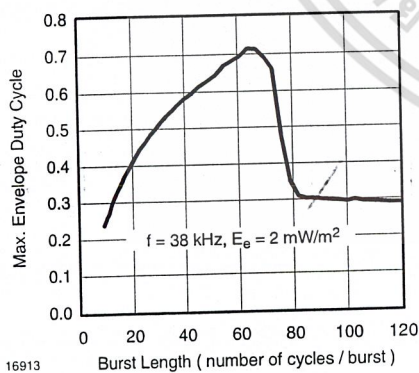


Figure 9. Max. Envelope Duty Cycle vs. Burstlength

Figure 12. Directivity

### Suitable Data Format

The circuit of the TSOP48.. is designed in that way that unexpected output pulses due to noise or disturbance signals are avoided. A bandpass filter, an integrator stage and an automatic gain control are used to suppress such disturbances.

The distinguishing mark between data signal and disturbance signal are carrier frequency, burst length and duty cycle.

The data signal should fulfill the following conditions:

- Carrier frequency should be close to center frequency of the bandpass (e.g. 38 kHz).
- Burst length should be 10 cycles/burst or longer.
- After each burst which is between 10 cycles and 70 cycles a gap time of at least 14 cycles is necessary.
- For each burst which is longer than 1.8 ms a corresponding gap time is necessary at some time in the data stream. This gap time should be at least 4 times longer than the burst.
- Up to 800 short bursts per second can be received continuously.

Some examples for suitable data format are: NEC Code (repetitive pulse), NEC Code (repetitive data), Toshiba Micom Format, Sharp Code, RC5 Code, RC6 Code, R-2000 Code, Sony Code.

When a disturbance signal is applied to the TSOP48.. it can still receive the data signal. However the sensitivity is reduced to that level that no unexpected pulses will occur.

Some examples for such disturbance signals which are suppressed by the TSOP48.. are:

- DC light (e.g. from tungsten bulb or sunlight)
- Continuous signal at 38 kHz or at any other frequency
- Signals from fluorescent lamps with electronic ballast with high or low modulation ( see Figure 13 or Figure 14 ).

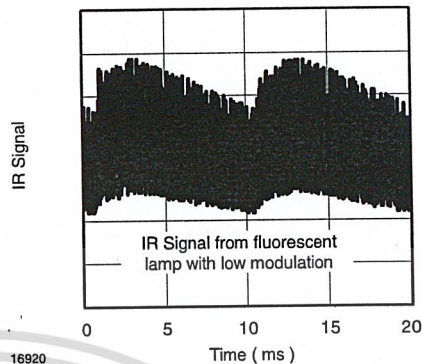


Figure 13. IR Signal from Fluorescent Lamp with low Modulation

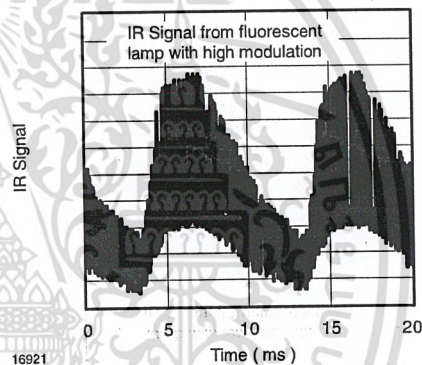


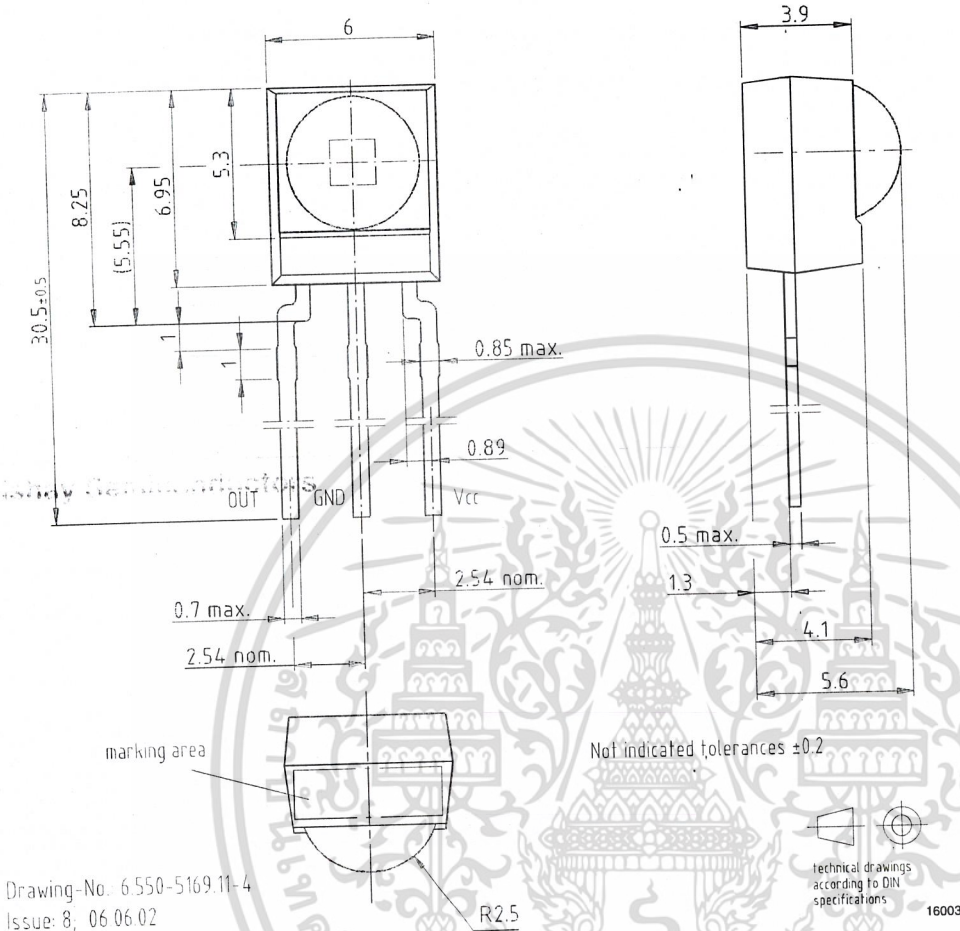
Figure 14. IR Signal from Fluorescent Lamp with high Modulation

# TSOP48..

Vishay Semiconductors



## Package Dimensions in mm



Drawing-No. 6.550-5169.11-4  
Issue: 8, 06.06.02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# DATA SHEET

For a complete data sheet, please also download:

- The IC06 74HC/HCT/HCU/HCMOS Logic Family Specifications
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Information
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Outlines

## 74HC/HCT00 Quad 2-input NAND gate

Product specification  
File under Integrated Circuits, IC06

December 1990

Philips  
Semiconductors



**PHILIPS**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Quad 2-input NAND gate

## 74HC/HCT00

## FEATURES

- Output capability: standard
- $I_{CC}$  category: SSI

## GENERAL DESCRIPTION

The 74HC/HCT00 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT00 provide the 2-input NAND function.

## QUICK REFERENCE DATA

GND = 0 V;  $T_{amb} = 25\text{ }^{\circ}\text{C}$ ;  $t_r = t_f = 6\text{ ns}$

| SYMBOL            | PARAMETER                              | CONDITIONS                                   | TYPICAL |     | UNIT |
|-------------------|--|--|---------|-----|------|
|                   |  |  | HC      | HCT |      |
| $t_{PHL}/t_{PLH}$ | propagation delay nA, nB to nY         | $C_L = 15\text{ pF}$ ; $V_{CC} = 5\text{ V}$ | 7       | 10  | ns   |
| $C_I$             | input capacitance                      |  | 3.5     | 3.5 | pF   |
| $C_{PD}$          | power dissipation capacitance per gate | notes 1 and 2                                | 22      | 22  | pF   |

## Notes

1.  $C_{PD}$  is used to determine the dynamic power dissipation ( $P_D$  in  $\mu\text{W}$ ):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

$f_i$  = input frequency in MHz

$f_o$  = output frequency in MHz

$C_L$  = output load capacitance in pF

$V_{CC}$  = supply voltage in V

$\sum (C_L \times V_{CC}^2 \times f_o)$  = sum of outputs

2. For HC the condition is  $V_I = \text{GND to } V_{CC}$   
For HCT the condition is  $V_I = \text{GND to } V_{CC} - 1.5\text{ V}$

## ORDERING INFORMATION

See "74HC/HCT/HCU/HCMOS Logic Package Information".

Quad 2-input NAND gate

74HC/HCT00

PIN DESCRIPTION

| PIN NO.      | SYMBOL          | NAME AND FUNCTION       |
|--------------|-----------------|-------------------------|
| 1, 4, 9, 12  | 1A to 4A        | data inputs             |
| 2, 5, 10, 13 | 1B to 4B        | data inputs             |
| 3, 6, 8, 11  | 1Y to 4Y        | data outputs            |
| 7            | GND             | ground (0 V)            |
| 14           | V <sub>CC</sub> | positive supply voltage |

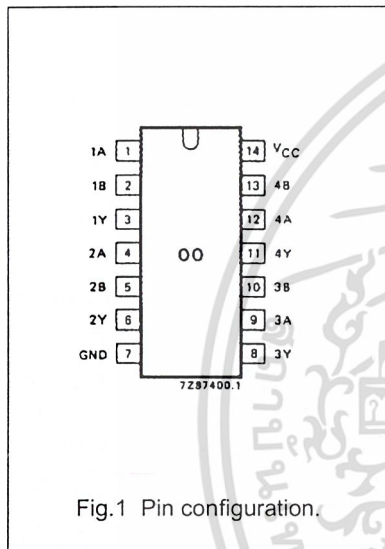


Fig.1 Pin configuration.

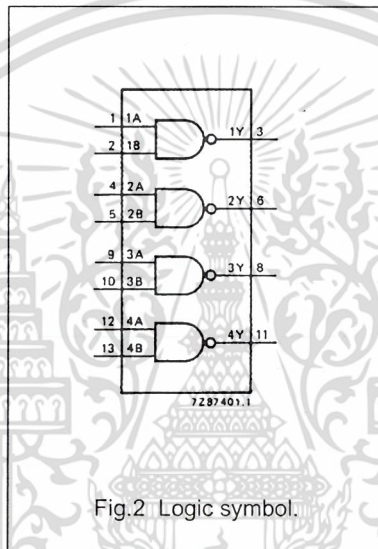


Fig.2 Logic symbol.

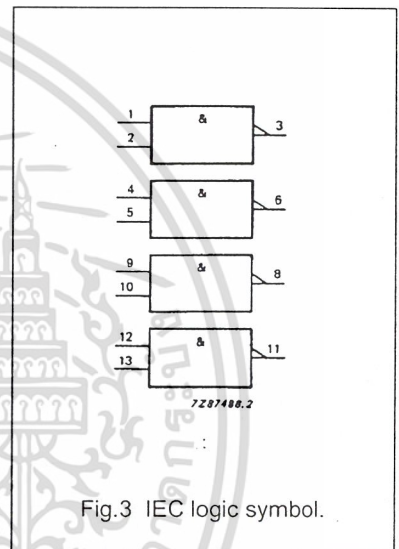


Fig.3 IEC logic symbol.

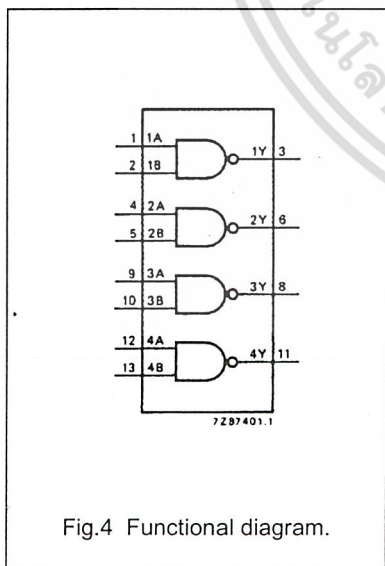


Fig.4 Functional diagram.

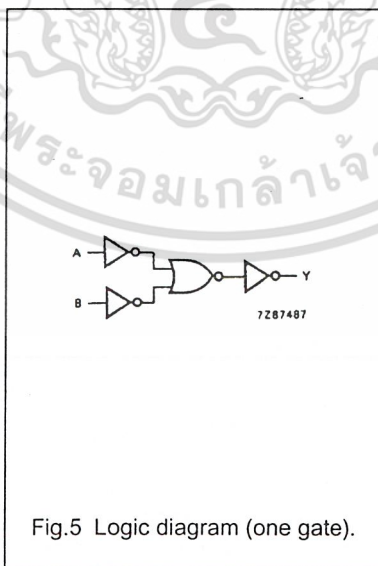


Fig.5 Logic diagram (one gate).

FUNCTION TABLE

| INPUTS |    | OUTPUT |
|--------|----|--------|
| nA     | nB | nY     |
| L      | L  | H      |
| L      | H  | H      |
| H      | L  | H      |
| H      | H  | L      |

Note

1. H = HIGH voltage level  
L = LOW voltage level

## Quad 2-input NAND gate

## 74HC/HCT00

## DC CHARACTERISTICS FOR 74HC

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications".

Output capability: standard

I<sub>CC</sub> category: SSI

## AC CHARACTERISTICS FOR 74HC

GND = 0 V; t<sub>r</sub> = t<sub>f</sub> = 6 ns; C<sub>L</sub> = 50 pF

| SYMBOL                              | PARAMETER                         | T <sub>amb</sub> (°C) |              |                |            |                 |             |                 |    | UNIT              | TEST CONDITIONS        |           |
|-------------------------------------|-----------------------------------|-----------------------|--------------|----------------|------------|-----------------|-------------|-----------------|----|-------------------|------------------------|-----------|
|                                     |                                   | 74HC                  |              |                |            |                 |             |                 |    |                   | V <sub>CC</sub><br>(V) | WAVEFORMS |
|                                     |                                   | +25                   |              |                | -40 to +85 |                 | -40 to +125 |                 |    |                   |                        |           |
|                                     |                                   | min.                  | typ.         | max.           | min.       | max.            | min.        | max.            |    |                   |                        |           |
| t <sub>PHL</sub> / t <sub>PLH</sub> | propagation delay<br>nA, nB to nY |                       | 25<br>9<br>7 | 90<br>18<br>15 |            | 115<br>23<br>20 |             | 135<br>27<br>23 | ns | 2.0<br>4.5<br>6.0 | Fig.6                  |           |
| t <sub>THL</sub> / t <sub>TLH</sub> | output transition<br>time         |                       | 19<br>7<br>6 | 75<br>15<br>13 |            | 95<br>19<br>16  |             | 110<br>22<br>19 | ns | 2.0<br>4.5<br>6.0 | Fig.6                  |           |

## DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications".

Output capability: standard

I<sub>CC</sub> category: SSI

## Note to HCT types

The value of additional quiescent supply current ( $\Delta I_{CC}$ ) for a unit load of 1 is given in the family specifications.

To determine  $\Delta I_{CC}$  per input, multiply this value by the unit load coefficient shown in the table below.

| INPUT  | UNIT LOAD COEFFICIENT |
|--------|-----------------------|
| nA, nB | 1.50                  |

## AC CHARACTERISTICS FOR 74HCT

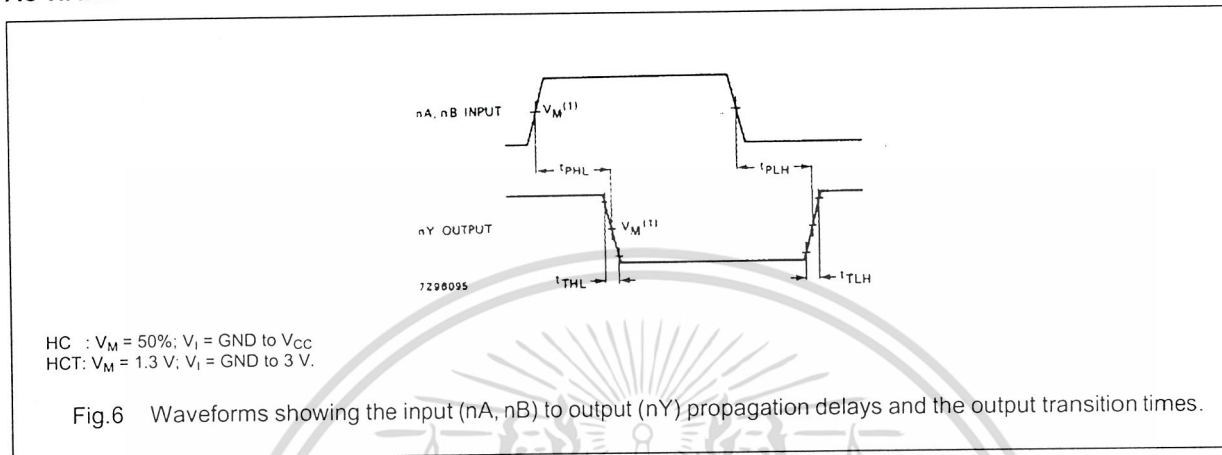
GND = 0 V; t<sub>r</sub> = t<sub>f</sub> = 6 ns; C<sub>L</sub> = 50 pF

| SYMBOL                              | PARAMETER                         | T <sub>amb</sub> (°C) |      |      |            |      |             |      |    | UNIT | TEST CONDITIONS        |           |
|-------------------------------------|-----------------------------------|-----------------------|------|------|------------|------|-------------|------|----|------|------------------------|-----------|
|                                     |                                   | 74HCT                 |      |      |            |      |             |      |    |      | V <sub>CC</sub><br>(V) | WAVEFORMS |
|                                     |                                   | +25                   |      |      | -40 to +85 |      | -40 to +125 |      |    |      |                        |           |
|                                     |                                   | min.                  | typ. | max. | min.       | max. | min.        | max. |    |      |                        |           |
| t <sub>PHL</sub> / t <sub>PLH</sub> | propagation delay<br>nA, nB to nY |                       | 12   | 19   |            | 24   |             | 29   | ns | 4.5  | Fig.6                  |           |
| t <sub>THL</sub> / t <sub>TLH</sub> | output transition<br>time         |                       | 7    | 15   |            | 19   |             | 22   | ns | 4.5  | Fig.6                  |           |

Quad 2-input NAND gate

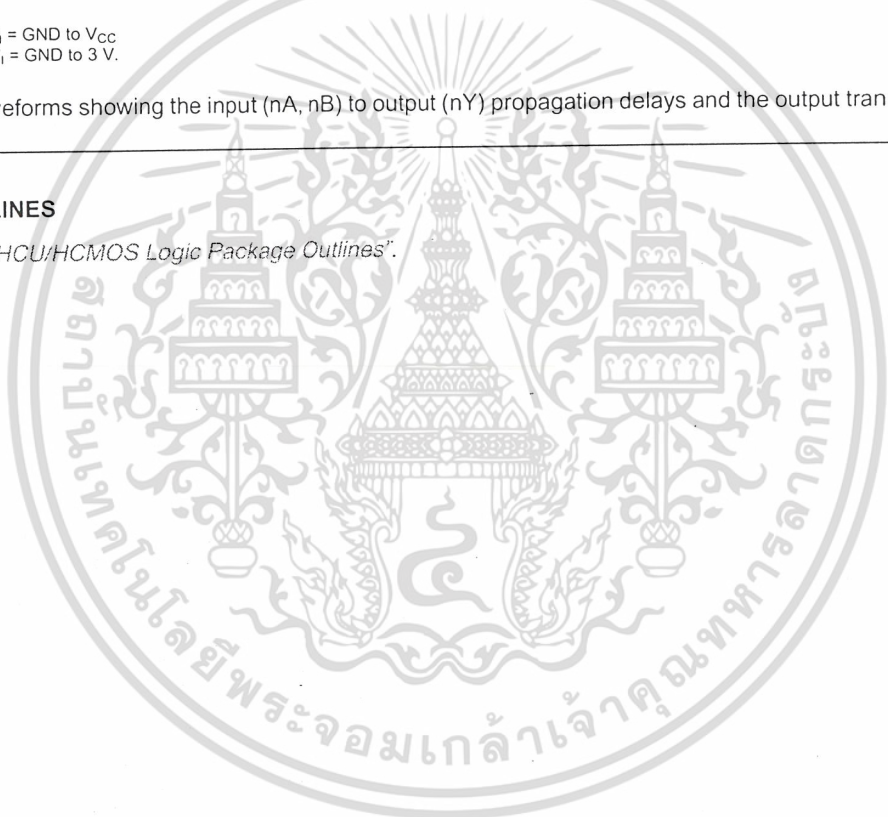
74HC/HCT00

AC WAVEFORMS



PACKAGE OUTLINES

See "74HC/HCT/HCU/HCMOS Logic Package Outlines".



This datasheet has been download from:

[www.datasheetcatalog.com](http://www.datasheetcatalog.com)

Datasheets for electronics components.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้