ห้องสมุดคณะวิศวกรรมศาสตร์ พระจอมเกล้าลาดกระบัง

ระบบบริหารการจอดรถ

CAR PARKING MANAGEMENT SYSTEM

โดย

นางสาวกุลธิดา ณ พัทลุง

นางสาวฐิติพร ปริทัศน์ปรีดา

นางสาวธัญญามาศ นันทนาณัฐกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เลขหมู่.....

เลขทะเบียน.....

วัน,เดือน,ปี.....

.b.....

.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบบริหารการจอดรถ

CAR PARKING MANAGEMENT SYSTEM

โดย

นางสาวกุลธิดา	ณ พัทลุง	47012005
นางสาวฐิติพร	ปริทัศน์ปรีดา	47012009
นางสาวธัญญามาศ	นันทนาณัฐกุล	47012014

อาจารย์ที่ปรึกษา

ผศ.ดร.สุทธิชัย นพนาคีพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบบริหารการจอดรถ**

CAR PARKING MANAGEMENT SYSTEM

ผู้จัดทำ

นางสาวกุลธิดา ณ พัทลุง 47012005

นางสาวฐิติพร ปริทัศน์ปรีดา 47012009

นางสาวธัญญามาศ นันทนาณัฐกุล 47012014

..... อาจารย์ที่ปรึกษา

(ผศ.ดร.สุทธิชัย นพนาคีพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบบริหารการจอดรถ

CAR PARKING MANAGEMENT SYSTEM

โดย นางสาวกุลธิดา ณ พัทลุง 47012005

นางสาวฐิติพร ปริทัศน์ปรีดา 47012009

นางสาวธัญญามาศ นันทนาณัฐกุล 47012014

อาจารย์ที่ปรึกษา ผศ.ดร.สุทธิชัย นพนาคีพงษ์

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอระบบบริหารการจอดรถ โดยนำเอาไมโครคอนโทลเลอร์ใช้ในการควบคุมและแสดงผลเมื่อที่จอดรถเต็ม อีกทั้งยังอำนวยความสะดวกแก่ผู้ใช้ เนื่องจากปัจจุบันปัญหาในการจอดรถทำให้เกิดการสูญเสียทรัพยากรเวลาและด้านอื่นๆอีกมากมาย

ABSTRACT

This project presents the car parking management system. By using microcontroller for control and display status of space. This project help accommodation safe time and others for user.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะทำได้โดยการเขียนข้อมูลให้เป็น ลอจิก "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการจะติดต่อด้วย นอกจากนั้นขาของ พอร์ต 3 จะยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ซึ่งจะมีรายละเอียดดังต่อไปนี้

-P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรมหรือขา RxD

-P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

-P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพท์จากภายนอกช่อง 0 หรือขา INT0

-P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพท์จากภายนอกช่อง 1 หรือขา INT1

-P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

-P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 1 หรือขา T1

-P3.7 ใช้เป็นขาอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป

*** P3.6 อยู่ภายในไอซีไม่ได้ต่อออกมาภายนอก แต่ใช้เป็นขารับสถานะของการเปรียบเทียบสัญญาณ Analog Comparator Input ระหว่างพอร์ต P1.0 และ P1.1 จากภายนอก

- รีเซต (Reset) เป็นขาที่ใช้รับสัญญาณในการรีเซต โดยจะรีเซตระบบการทำงานของไมโครคอนโทรลเลอร์ ในการป้อนสัญญาณนั้นจะต้องทำให้สถานะที่ขานี้อยู่ในระดับลอจิก"1" (high) อย่างน้อย 2 แมทชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

- XTAL 1 และ XTAL 2 เป็นขาที่ใช้สำหรับต่อกับตัวคริสตอล เพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะ การทำงานของไมโครคอนโทรลเลอร์

2.1.1.5 วงจรนับ/วงจรตั้งเวลา

จากตารางที่ 2.1 ไมโครคอนโทรเลอร์เบอร์ 8052 มีวงจรนับและเป็นวงจรตั้งเวลาชนิด 16 บิต มากกว่าไมโครคอนโทรเลอร์เบอร์ 8051 อยู่หนึ่งตัว ในการทำงานของวงจรวงจรนับและเป็นวงจรตั้งเวลาเป็นดังนี้เมื่อทำงานเป็นวงจรตั้งเวลา รีจีสเตอร์วงจรนับจะเพิ่มขึ้นหนึ่งเมื่อมีสัญญาณป้อนให้ทางอินพุท อัตราการนับสัญญาณสูงสุดคือ 1/24 ของความเร็วสัญญาณนาฬิกาของไมโครคอนโทรเลอร์ วงจรนับและวงจรตั้งเวลา 0 และ 1 มีวิธีทำโปรแกรมให้ทำงานต่างกันถึง 4 แบบ ซึ่งรวมทั้งการทำงานเป็น 8 บิต หรือ 16 บิต และการบรรจุค่าพรีเซตหนึ่งค่าได้อย่างอัตโนมัติ วงจรนับและเป็นวงจรตั้งเวลาที่ 1 ซึ่งเลือกโปรแกรมให้ทำหน้าที่เป็นตัวกำเนิดสัญญาณของอัตราการส่งบิตออกไปยังพอร์ตอนุกรม สำหรับใช้เชื่อมต่อวงจรนับและเป็นวงจรตั้งเวลาที่ 2 เฉพาะไมโครคอนโทรเลอร์เบอร์ 8052 เท่านั้น และมีการทำงานย่อยๆ อีก 3 ชนิด ดังนี้

1. วงจรนับ 16 บิตที่สามารถโหลดค่ากลับคืนเองอย่างอัตโนมัติ
2. วงนับที่จองไว้ 16 บิต
3. วงจรกำเนิดสัญญาณของการส่งบิต เพื่อใช้ในการเชื่อมโยงข้อมูล

2.1.1.6 พอร์ตชนิดอนุกรมอยู่ภายใน

ไมโครคอนโทรเลอร์ทั้งหมดในตระกูล MCS-51 เป็นไมโครคอนโทรเลอร์ ที่มีการเชื่อมโยงข้อมูลชนิดสองทาง ทำให้รับและส่งข้อมูลพร้อมกัน ตัวรับส่งข้อมูลชนิดอะซิงโครนัส (Asynchronous

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Receiver) มีบัฟเฟอร์สำหรับข้อมูลเป็นพิเศษเพื่อเพิ่มความเร็วในการสื่อสารชนิดพอร์ตอนุกรมนั้นสามารถเลือกโปรแกรม เพื่อเลือกใช้การทำงานแบบใดแบบหนึ่งใน 4 แบบด้วยการใช้โปรแกรมควบคุมอัตราการส่งข้อมูลและรูปแบบของข้อมูล อัตราการส่งข้อมูลที่เลือกใช้ได้สูงสุดถึง 19,200 บิต/วินาที ด้วยความเร็วของสัญญาณนาฬิกา 1 เมกะเฮิรตซ์ สำหรับใช้ในระบบเครือ (Networks) และระบบการสื่อสารของไมโครคอนโทรลเลอร์ หลายตัวร่วมกันจะเลือกความเร็วของสัญญาณนาฬิกาด้วยวงจรนับและวงจรตั้งเวลา

2.1.1.7 การขัดจังหวะและชุดคำสั่ง

ไมโครคอนโทรลเลอร์ 8051 รับการขัดจังหวะได้ 5 แหล่ง ส่วนไมโครคอนโทรลเลอร์ 8052 รับการขัดจังหวะจากอุปกรณ์อื่นๆได้ถึง 6 แหล่ง คือ ขา INTO และขา INT1 ซึ่งกำหนดให้ใช้ระดับพัลส์หรือขอบขาพัลส์ก็ได้ วงจรนับและวงจรตั้งเวลาที่ 0 และ 1 ซึ่งสำหรับการใช้งานไมโครคอนโทรลเลอร์ 8051 จะเพิ่มวงจรตั้งเวลาหรือ วงจรนับตัวที่ 2 และตัวสุดท้ายจากพอร์ตอนุกรมสามารถกำหนดลำดับความสำคัญของการขัดจังหวะได้ 2 ระดับไม่ต้องอาศัยวงจรภายนอกเข้ามาช่วยแต่ละแหล่งการขัดจังหวะ 5 หรือ 6 แหล่งนั้น สามารถกำหนดให้เป็นเวคเตอร์เฉพาะ ตัวชี้ตำแหน่ง ดังนั้นเมื่อมีขัดจังหวะเข้ามาตัวไมโครโปรเซสเซอร์จะกระโดดไปที่ส่วนของโปรแกรมที่ทำงานตามวัตถุประสงค์ของการขัดจังหวะนั้นและหลังจากเก็บข้อมูลต่างของโปรแกรมการนับลงสแตก

2.1.1.8 AT89C51/52

AT89C51/52 พลิกรูปแบบการใช้งานไมโครคอนโทรลเลอร์ขนาด 8 บิต (Microcontroller Unit :MCU) ที่สามารถใช้เครื่องคอมพิวเตอร์ PC ทำการลบและทำการเขียนโปรแกรมได้โดยตรง ไม่ต้องถอด MCU ออกจากการ์ดหรือแผงวงจรในลักษณะที่เรียกว่า in system programming หรือจะใช้เครื่องโปรแกรม (Universal programmer)โดยตรงก็ได้ไม่ต้องใช้การลบด้วยแสงอัตราไวโอเลต (UV eraser) ทำให้ใช้เวลาลบได้เพียงเสี้ยววินาที และนั่นคือ AT89C51/52 ประดิษฐ์กรรมอุปกรณ์ ไมโครคอนโทรลเลอร์ที่มีหน่วยความจำแบบแฟลชอยู่ภายในตัวขนาดตั้งแต่ 1 กิโลไบต์ถึง 8 กิโลไบต์ที่สามารถโปรแกรมทับลงไปได้อีกนับพันครั้งเหมือนกับการโปรแกรมในลักษณะเดียวกันกับ อีพรอม (EPROM)

ไมโครคอนโทรลเลอร์ขนาด 8 บิต (MUC) โดยทั่วไปมักมีหน่วยความจำภายในตัวเป็นแบบแมสค์รอม (MASK ROM), วันไทม์โปรแกรมมิ่ง (one time programming:OTP) หรืออีพรอม ซึ่งค่อนข้างยุ่งยากในการพัฒนาผลิตภัณฑ์ และพัฒนาโปรแกรมตามคำสั่งต่างๆที่ต้องการโดยเฉพาะในแบบ วันไทม์โปรแกรมมิ่ง ซึ่งเป็นแบบโปรแกรมได้เพียงครั้งเดียวถ้ามีการเขียนโปรแกรมผิดพลาดเพียงนิดเดียวก็ต้องทิ้งไมโครคอนโทรลเลอร์ขนาด 8 บิต ตัวนั้นไปเลย เป็นเพราะเป็นแบบโปรแกรมได้เพียงครั้งเดียว ถ้าเป็นแบบแมสค์รอมถึงแม้ว่าราคาถูกกว่า แต่มีความยุ่งยากมากกว่าเนื่องจากผู้ผลิต ไมโครคอนโทรลเลอร์ขนาด 8 บิต จะต้องเป็นผู้โปรแกรมให้ เป็นเหตุทำให้ต้องสั่งผลิตจำนวนคราวละมากๆและค่อนข้างจะเสียเวลา ทำให้การวางแผนที่จะส่งผลิตภัณฑ์สู่ตลาดต้องใช้เวลา ส่วนแบบอีพรอมนั้นมีราคาแพงและต้องใช้วิธีการลบด้วยแสงอัลตร้าไวโอเลตด้วยเครื่องยูวีอิเรเซอร์ ในกรณีที่ต้องแก้ไขหรือโปรแกรมข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจำนวนครั้งในการลบก็จำกัดดังนั้นการเลือกใช้ MUC ที่ผ่านมามักเกิดมีปัญหาในการเลือกเบอร์หรือเลือกชนิดที่ไว้ใจได้มาตลอด ไมโครคอนโทรลเลอร์ตระกูล AT89C51 สามารถใช้งานร่วม และแทนกันได้กับไมโครคอนโทรลเลอร์ตระกูล MCS-51ได้แก่เบอร์ 80C31, 80C51, 80C52 หรือ 87C51/52 เป็นต้น ซึ่งAT89C51 เหมือนตระกูลเหล่านี้ของ INTEL ทั้งในด้านชุดคำสั่ง และการจัดเรียงขา นั่นคือเราสามารถนำ AT89C51 ของ ATMEL มาใช้งานแทนไมโครคอนโทรลเลอร์ของ INTEL ได้เลย

2.1.1.9 โครงสร้างโปรแกรมภาษา C51

โครงสร้างโปรแกรมภาษาซี C51 แบ่งส่วนประกอบเป็น 2 ส่วนคือ ส่วนหัวของโปรแกรม (Head) และส่วนตัวของโปรแกรม (Body) ตัวแสดงต่อไปนี้

```
#pragma SMALL
#include <reg51.h>
sbit light = P2^0;
void delay(void)
{
int i,j;
for(i=0;i<1000;i++)
{
   for(j=0;j<1000;i++)
   {
     i=i+0;
   }
}
}

void main(void)
 while(1)
 {
    light=0;
    delay();
    light=1;
    delay();
  }
}
```

ส่วนหัวโปรแกรม

ส่วนของตัวโปรแกรม

1. ส่วนหัวของโปรแกรม

ส่วนหัวของโปรแกรมจะเริ่มต้นตั้งแต่บรรทัดแรกจนถึงบรรทัดก่อน main() มีไว้สำหรับกำหนดพรีโพรเซสเซอร์ไดเร็คทีฟ, กำหนดตัวแปร, ค่าเริ่มต้น และฟังก์ชันย่อย

- พรีโพรเซสเซอร์ไดเร็คทีฟ (Preprocessor Directive)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พรีโพรเซสเซอร์ไดเร็คทีฟ (Preprocessor Directive)

พรีโพรเซสเซอร์ไดเร็คทีฟคือคำสั่งพิเศษที่ช่วยในการเขียนโปรแกรม C51 ให้ง่ายและมีประสิทธิภาพ คำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟจะถูกใช้ในขั้นตอนการแปลความหมายโปรแกรม โดยจะถูกแปลความหมายเป็นคำสั่งแรก

คำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟจะเริ่มต้นด้วยเครื่องหมาย # เช่น #include #define ซึ่งหมายความว่า ถ้าตัวแปลภาษาตรวจพบเครื่องหมาย # ที่บรรทัดใด ก็จะถือว่าข้อความทั้งหมดในบรรทัดนั้นเป็นคำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟ และในหนึ่งบรรทัดจะเขียนคำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟได้เพียงคำสั่งเดียวเท่านั้น

ตัวอย่างเช่น

```
#pragma SMALL    #include <reg51.h>
```

ถือว่าผิดเพราะอยู่บรรทัดเดียวกับมีคำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟสองคำสั่ง

คำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟที่ใช้สำหรับการเขียนโปรแกรมด้วยภาษา C51 แสดงไว้ดังนี้

- #if
- #ifdef
- #ifndef
- #include
- #define
- #undef
- #else
- #elif
- #endif
- #line
- #error
- #pragma

ตัวอย่างการใช้งานคำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟที่สำคัญที่ใช้สำหรับโครงงานนี้ แสดงดังต่อไปนี้

1. #pragma

ไดเร็คทีฟ #pragma ใช้สำหรับกำหนดว่าโปรแกรม C51 จะใช้โมเดลหน่วยความจำแบบไหน โดยจะกำหนดชื่อโมเดลต่อจาก #pragma ซึ่งเมื่อตัวแปลภาษาแปลความหมาย ก็จะจัดการหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเก็บโปรแกรมและข้อมูลตามที่กำหนดไว้ การเลือกโมเดลหน่วยความจำให้ถูกต้อง เพื่อโปรแกรมจะได้ทำงานอย่างมีประสิทธิภาพ

ตัวอย่างเช่น

```
#pragma SMALL
```

เป็นการกำหนดให้ใช้หน่วยความจำโปรแกรมและข้อมูลภายในชิปไมโครคอนโทรลเลอร์ MCS – 51

2. #include

ไดเร็คทีฟ #include ใช้สำหรับสั่งให้ตัวแปลภาษานำไฟล์ที่กำหนดไว้หลัง #include มารวมเข้ากับโปรแกรมก่อนแปลความหมาย ในกรณีที่ต้องการใช้งานคำสั่งหรือฟังก์ชันย่อยจากไฟล์ที่กำหนดจะเขียนอยู่ภายในเครื่องหมาย < >

- การประกาศตัวแปร

การประกาศตัวแปรคือ การจองพื้นที่หน่วยความจำสำหรับเก็บข้อมูล พร้อมทั้งกำหนดชื่อเรียกแทนหน่วยความจำในตำแหน่งนั้น ซึ่งการใช้งานจะเรียกผ่านชื่อตัวแปรที่ประกาศไว้ และขนาดของพื้นที่การจองจะขึ้นอยู่กับชนิดของข้อมูลด้วยว่ามีขนาดกี่ไบต์

ตัวอย่างเช่น

```
char c;
```

เป็นการประกาศค่าตัวแปรชื่อ C เป็นตัวแปรชนิด char ใช้เก็บข้อมูลประเภทอักขระ โดยใช้พื้นที่หน่วยความจำขนาด 1 ไบต์

นอกจากนี้ยังสามารถกำหนดค่าเริ่มต้นให้กับตัวแปรนั้นภายในบรรทัดเดียวได้เลย

ตัวอย่างเช่น

```
int x = 10
```

เป็นการประกาศค่าตัวแปร x เป็นตัวแปรชนิด Integer สำหรับเก็บข้อมูลประเภทจำนวนเต็ม ใช้พื้นที่หน่วยความจำขนาด 4 ไบต์ และกำหนดค่าเริ่มต้นให้เป็น 10

นอกจากำหนดเป็นค่าเริ่มต้นแล้ว ยังสามารถกำหนดตัวแปรแทนรีจิสเตอร์หรือพอร์ตของไมโครคอนโทรลเลอร์ได้ด้วย เพื่อช่วยในการเขียนโปรแกรมให้ง่ายขึ้น

ตัวอย่างเช่น

```
Sbit right = P0^2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการประกาศตัวแปรชื่อ right เป็นตัวแปรชนิดบิต ใช้พื้นที่หน่วยความจำขนาด 1 บิต และกำหนดให้ตัวแปรนี้แทนที่ 2 ของพอร์ต P0

คำสั่งในการประกาศตัวแปรจะต้องปิดด้วยเครื่องหมาย ; เสมอ

- ประกาศฟังก์ชัน

ฟังก์ชันคือ การนำคำสั่งหลายๆ คำสั่งมาเขียนรวมกันเป็นบล็อกของคำสั่ง ส่วนของการประกาศฟังก์ชันเป็นส่วนที่เราเขียนฟังก์ชันที่สร้างขึ้นเอง เหมือนเป็นการประกาศว่าฟังก์ชันที่สร้างขึ้นมีรูปแบบและการทำงานอย่างไร จึงจำเป็นต้องประกาศไว้ที่ส่วนหัวของโปรแกรม สำหรับการเรียกใช้งานนั้นจะเรียกในส่วนตัวของโปรแกรม ดังตัวอย่างต่อไปนี้

```
#pragma SMALL
#include "\C51P\INC\stdio.h"
void serial0_init_T1(void)
{
TH1 = 0x0f3;
  TMOD = 0x20;
  S0CON = 0x52;
  PCON = 0x80;
  TR1 = 1;
}

void main(void)
{
   serial0_init_T1()
   while(FOREVER)
   {
  printf("hello world")
}
}
```

ประกาศฟังก์ชันย่อยชื่อ serial0_init_T1

เรียกใช้งานฟังก์ชันย่อย

การประกาศฟังก์ชันนอกจากที่กล่าวมาแล้ว ยังมีอีกวิธีที่นิยมคือ ประกาศรูปแบบฟังก์ชันไว้ส่วนที่หัวของโปรแกรม แต่เขียนฟังก์ชันกำหนดไว้ที่ส่วนตัวของโปรแกรม โดยกำหนดไว้หลังจากฟังก์ชัน main แสดงดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#pragma SMALL
#include "\C51P\INC\stdio.h"

void serial0_init_T1(void)

void main(void)
 {
    serial0_init_T1()
    while(FOREVER)
    {
       printf("hello world")
    }
}
void serial0_init_T1(void)
{
    TH1 = 0x0f3;
    TMOD = 0x20;
    S0CON = 0x52;
    PCON = 0x80;
    TR1 = 1;
}
```

การประกาศรูปแบบฟังก์ชัน

เรียกใช้งานฟังก์ชันย่อย serial0_init_T1

กำหนดค่าทำงานของฟังก์ชันย่อย serial0_init_T1

การประกาศฟังก์ชันเพื่อให้ตัวแปลภาษารับรู้ว่ามีฟังก์ชันอะไรบ้าง รูปแบบของฟังก์ชันนั้นเป็นอย่างไร และในส่วนของตัวโปรแกรม การกำหนดฟังก์ชันต้องใช้รูปแบบเดียวกับที่ประกาศไว้ที่ส่วนหัวของโปรแกรม ไม่เช่นนั้นจะเกิดข้อผิดพลาด (Error) เวลาคอมไพล์

2. ส่วนหัวของโปรแกรม

โปรแกรมภาษา C51 จะเริ่มต้นทำงานที่ main () ซึ่งเป็นฟังก์ชันหลักของโปรแกรมภาษา C51 ไปจนจบโปรแกรมโดยภายในตัวโปรแกรมประกอบด้วยคำสั่งต่างๆ คำสั่งทั้งหมดในภาษา C51 สามารถแบ่งได้เป็น 2 ประเภทคือ

- คำสั่งเดี่ยว
- ฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะวิศวกรรมศาสตร์ พระจอมเกล้าลาดกระบัง

- คำสั่งเดี่ยว

เป็นคำสั่งที่สั่งให้คอมพิวเตอร์ทำงานอย่างใดอย่างหนึ่ง โดยคำสั่งในภาษา C51 มีจำนวนมาก เวลาใช้อาจจะต้องเปิดดูจากคู่มือการใช้คำสั่ง ตัวอย่างคำสั่งเดี่ยวแสดงไว้ดังนี้

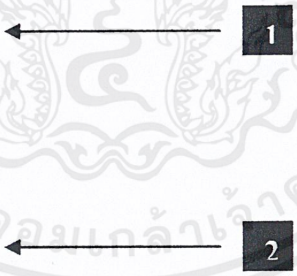
```
printf("hello world")      ; เป็นคำสั่งส่งข้อความจาก MCS – 51 ออกทางพอร์ตอนุกรม
T1 = 0                     ; เป็นคำสั่งเคลียร์ค่าของแฟล็ก T1
i++;                       ; เป็นคำสั่งเพิ่มค่าของตัวแปร i ขึ้นอีกหนึ่ง
x = a + b                  ; เป็นคำสั่งบวกค่าของตัวแปร a กับ b ผลลัพธ์เก็บไว้ที่ตัวแปร x
```

บางคำสั่งเป็นคำสั่งที่ใช้ในการติดต่อกับรีจีสเตอร์ หน่วยความจำ และพอร์ตต่างๆ ของ MCS – 51 โดยตรง ซึ่งคำสั่งเหล่านี้เก็บอยู่ในไฟล์ไลบรารี ดังนั้น เพื่อให้ตัวแปลภาษาแปลความหมายของโปรแกรมได้ถูกต้อง จึงต้องมีการใช้คำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟ #include ตามด้วยชื่อไฟล์ไลบรารีที่ส่วนหัวโปรแกรม

- ฟังก์ชัน

การเรียกใช้ฟังก์ชันแบ่งออกเป็น 2 แบบตามประเภทของฟังก์ชันนั้นคือ การเรียกใช้ฟังก์ชันมาตรฐานหรือฟังก์ชันที่สร้างไว้แล้ว สามารถเรียกใช้ได้เลยแต่ต้องกำหนดในพรีโพรเซสเซอร์ไดเร็คทีฟ #include ว่าฟังก์ชันที่ต้องเรียกใช้อยู่ไหน ดังตัวอย่างต่อไปนี้

```
#include "stdio.h"
void main(void)
 {
char  i;
for(1 = 0;< 100;++)
        printf("hello world")
}
```



(1) ประกาศชื่อไฟล์ไลบรารี stdio.h ไว้ที่ส่วนหัวของโปรแกรม โดยใช้คำสั่งพรีโพรเซสเซอร์ไดเร็คทีฟ #include เพื่อบอกให้ตัวแปลภาษารู้ว่า ในโปรแกรมมีฟังก์ชันจากไลบรารี stdio.h

(2) เรียกใช้งานฟังก์ชัน printf() ซึ่งฟังก์ชันนี้อยู่ในไฟล์ไลบรารี ถ้าไม่ประกาศชื่อไฟล์นี้ไว้ที่ส่วนหัวของโปรแกรม เวลาเรียกใช้งานฟังก์ชันจะเกิด Error ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. รูปแบบการเขียนคำสั่งภาษา C51

การเขียนคำสั่งในภาษา C51 จะยึดข้อกำหนดตามเงื่อนไขของภาษาซี โดยทุกคำสั่งลงท้ายด้วย (;) ซึ่งการเขียนหนึ่งคำสั่งในหนึ่งบรรทัดสามารถเขียนได้มากกว่าหนึ่งคำสั่ง แต่ทุกคำสั่งต้องปิดท้ายด้วย (;)

- คำบรรยายแทรก

คำบรรยายแทรกหรือที่เรียกว่า Comment มีไว้เพื่ออธิบายการทำงานของโปรแกรม โดยจะแทรกไว้ที่ตำแหน่งไหนของโปรแกรมก็ได้ ซึ่งจะช่วยให้คนที่ต้องการศึกษาโปรแกรมเข้าใจมากขึ้น

- ตัวแปรและค่าคงที่

เนื่องจากภาษา C51 มีโครงสร้างเหมือนภาษาซี การประกาศตัวแปรและค่าคงที่จึงคล้ายกัน นั่นคือ ตัวแปรและค่าคงที่ที่ซึ่งประกาศไว้จะถูกจองพื้นที่ไว้ในหน่วยความจำ โดยจะจองหน่วยความจำในส่วนใดนั้น ขึ้นอยู่กับการกำหนดโมเดลหน่วยความจำ

- ตัวแปรในภาษา C51

ตัวแปร คือการจองพื้นที่หน่วยความจำสำหรับเก็บข้อมูล พร้อมทั้งกำหนดชื่อเรียกแทนหน่วยความจำในตำแหน่งนั้น

- ชนิดของตัวแปร

ตัวแปรที่ใช้งานภายในซีพียูมีหลากหลายชนิด และขนาดของตัวแปรแต่ละชนิด จะขึ้นอยู่กับชนิดข้อมูลที่เราต้องการใช้ ซึ่งจะแสดงได้ดังตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ชนิดของตัวแปรในภาษา C51

ชนิดตัวแปร	ขนาดข้อมูล	ขอบเขตข้อมูล	คำอธิบาย
Bit	1 bit	0 ถึง 1	เก็บข้อมูลขนาด 1บิต
Char	8 bits	-128 ถึง 127	เก็บข้อมูลชนิดตัวอักษร โดยตัวอักษรมีพื้นที่เก็บ 8 bits
Unsigned Char	8 bits	0 ถึง 255	เก็บข้อมูลชนิดตัวอักษร แบบไม่คิดเครื่องหมาย
Int	14 bits	-32,76832,767	เก็บข้อมูลชนิด จำนวนเต็มแบบยาวใช้ พื้นที่หน่วยความจำ 16 bits
Unsigned Int	16 bits	0 ถึง 65535	เก็บข้อมูลชนิด จำนวนเต็มแบบไม่คิด เครื่องหมาย
Long	32 bits	-2,174,483,648 ถึง 2,147,438,649	เก็บข้อมูลชนิด จำนวนเต็มแบบยาว หน่วยความจำ 32 bits
Unsigned Long	32 bits	0 ถึง 4,294,967,296	เก็บข้อมูลชนิด จำนวนเต็มแบบยาว แบบไม่คิดเครื่องหมาย
Float	32 bits	3.4x10-38 ถึง 3.4x1038	เก็บข้อมูลชนิดตัวเลข ทศนิยมใช้พื้นที่ หน่วยความจำ 32 bits
Pointer	24/16/8 bits		ตัวชี้แอดเดรส
Sbit	1 bits	0 ถึง 1	สำหรับการเข้าถึงแบบ บิต
Sfr	8 bits	0 ถึง 255	สำหรับการเข้าถึง รีจิสเตอร์ภายในซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่งเงื่อนไขละการวนซ้ำ

1. คำสั่ง if

เป็นคำสั่งที่ทำงานแบบทางเลือกเดียวนั่นคือ จะทำงานเฉพาะกรณีที่เงื่อนไขเป็นจริงเท่านั้น รูปแบบการเขียนคำสั่งเป็นดังนี้

```
if(condition) {
   /*code*/ ;
            }
```

- ขึ้นต้นด้วยคำว่า if ตามด้วยเครื่องหมาย () ซึ่งภายในเป็นเงื่อนไข โดยปกติจะเขียนเป็นนิพจน์เปรียบเทียบที่ให้ผลออกมาว่าเป็นจริงหรือเท็จ

- ถ้าเงื่อนไขเป็นจริง โปรแกรมก็จะทำตามคำสั่งที่อยู่ตรง /* code */ ระหว่างเครื่องหมาย { } แต่ถ้าเงื่อนไขเป็นเท็จ ไม่ต้องทำอะไรจะไม่ต้องทำอะไรออกจากคำสั่ง if ไปเลย

2. คำสั่ง while

เป็นการวนซ้ำแบบ while จะทำตามคำสั่งไปเรื่อยๆ ตราบใดที่เงื่อนไขคงยังเป็นจริง

```
while (condition is true) {
/*Executable code*/ ;
            }
```

- คำสั่ง while จะเริ่มต้นด้วยคำว่า while แล้วตามด้วยการตรวจสอบเงื่อนไขการวนซ้ำที่อยู่ภายในเครื่องหมาย ()

- ถ้าเงื่อนไขเป็นจริงก็จะทำตามคำสั่งที่อยู่ตรง /* Executable code */ แต่ถ้าเงื่อนไขเป็นเท็จก็จะสิ้นสุดคำสั่ง while

- เครื่องหมายและตัวดำเนินการ

ในการเขียนภาษา C51 นั้น รูปแบบการเขียนคำสั่งดำเนินการจะเหมือนกับภาษาซี นั่นคือเขียนให้อยู่ในรูปนิพจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ตารางชนิดเครื่องหมายในภาษา C51

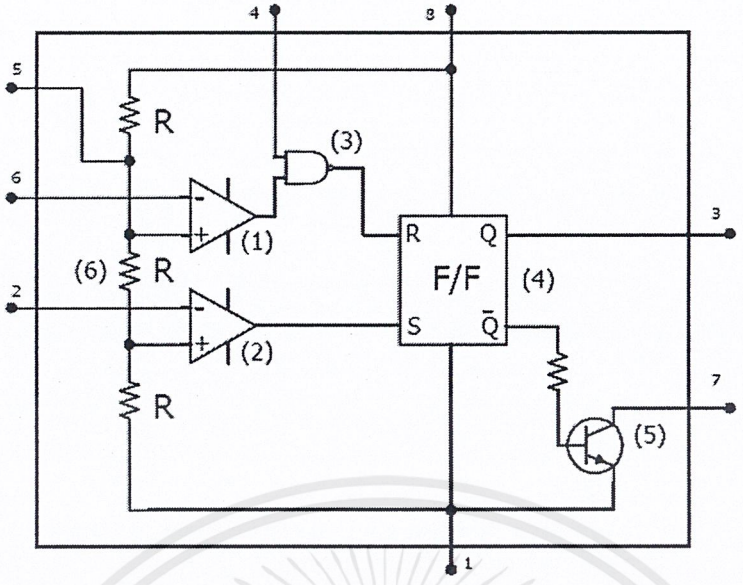
เครื่องหมาย	การดำเนินการ	ตัวอย่างการใช้งาน	ความหมาย
<	มากกว่า	X< Y	ถ้าค่าในตัวแปรเป็น X มากกว่าตัวแปร Y ผลลัพธ์เป็นจริง
>	น้อยกว่า	X>Y	ถ้าค่าในตัวแปรเป็น X น้อยกว่าตัวแปร Y ผลลัพธ์เป็นจริง
<=	มากกว่าหรือเท่ากับ	X<=Y	ถ้าค่าในตัวแปรเป็น X มากกว่าหรือเท่ากับตัวแปร Y ผลลัพธ์เป็นจริง
>=	น้อยกว่าหรือเท่ากับ	X>=Y	ถ้าค่าในตัวแปรเป็น Xน้อยกว่าหรือเท่ากับตัวแปร Y ผลลัพธ์เป็นจริง
&&	And	X&&Y	ถ้านำค่า X และ Y มาทำการ AND กัน ผลลัพธ์จะให้ค่าเป็นจริงหรือเท็จขึ้นอยู่กับค่า X และ Y โดยผลลัพธ์เป็นจริงกรณีเดียวคือ คือค่าของ X และ Y เป็นจริง
||	OR	X||Y	ถ้านำค่า X และ Y มาทำการ OR กัน ผลลัพธ์จะให้ค่าเป็นจริงหรือเท็จขึ้นอยู่กับค่า X และ Y โดยผลลัพธ์เป็นเท็จกรณีเดียวคือ คือค่าของ X และ Y เป็นเท็จ
!	NOT	!Y	ถ้านำค่า Y มา NOT กัน ผลลัพธ์ที่ได้ขึ้นอยู่กับค่า Y ว่าเป็นจริงหรือเท็จ ถ้า Y = เท็จ ผลลัพธ์จะเป็นจริงถ้า ถ้า Y = จริง โดยผลลัพธ์เป็นเท็จ

2.1.2 ทฤษฎี IC – 555

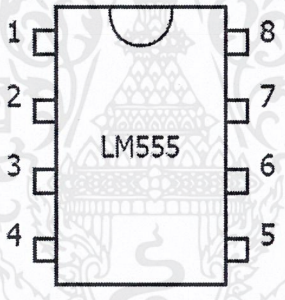
ไอซีเบอร์ 555 เป็นไอซีที่นิยมใช้กันมากในการนำไปสร้างสัญญาณลูกคลื่นแบบต่างๆ เช่น สัญญาณ Square wave สัญญาณพัลส์ สัญญาณ Ramp และวงจรตั้งเวลาไอซีเบอร์ 555 เป็นอุปกรณ์วงจรรวมที่มีอุปกรณ์อิเล็กทรอนิกส์อื่นๆ อยู่ภายใน และมีส่วนที่ต้องต่อภายนอกเพื่อควบคุมการทำงาน และใช้งานเป็นลักษณะต่างๆ ซึ่งง่ายต่อการออกแบบและง่ายในการสร้างสัญญาณพัลส์ความถี่ต่างๆ อีกทั้งสามารถเข้าใจในการทำงานได้ง่าย

นอกจากไอซีเบอร์ 555 และยังมีไอซีเบอร์ 556 ที่เป็นแบบ Dual timer ประกอบด้วยไอซีเบอร์ 555 จำนวน 2 ตัว อยู่ภายในตัวเดียวกันเพื่อใช้เป็นวงจรตั้งเวลาและสะดวกในการออกแบบวงจรที่ต้องใช้ไอซีเบอร์ 555 หลายๆตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 โครงสร้างภายในไอซีเบอร์ 555



รูปที่ 2.7 การจัดวางขา ไอซีเบอร์ 555

1. Gnd
2. Trigger
3. Output
4. Reset
5. Control Voltage
6. Threshold
7. Discharge
8. Vcc

2.1.2.1 หน้าที่ต่างๆ ของไอซีเบอร์ 555

ขา (1),(2) คือวงจรคอมพาราเตอร์ Comparator เป็นตัวที่ทำหน้าที่เปรียบที่สัญญาณอินพุททั้ง 2 ขา ถ้าศักดาไฟฟ้าที่ขั้วบวกมากกว่าศักดาไฟฟ้าที่ขั้วลบเอาท์พุตจะมีค่า “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ Vcc ถ้าศักดาที่ขั้วบวกน้อยกว่าขั้วลบ เอาท์พุท จะมีค่า “0” เท่ากับศูนย์โวลต์

ขา (3) คือวงจรแนนเกท (NAND Gate) มีคุณสมบัติว่า ถ้าอินพุตทั้งสองขาเป็น “1” เอาท์พุทจะมีค่าเป็น “0” แต่ถ้าอินพุทขาใดขาหนึ่งเป็น “0” ก็จะทำให้เอาท์พุท จะมีค่าเป็น “1” ทันที

ตารางที่ 2.4 Truth table NAND Gate

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

ขา (4) เป็นวงจร ฟลิปฟลอป (Flip-Flop) แบบ RS-F/F มีคุณสมบัติดังนี้

ตารางที่ 2.5 ตาราง Flip-Flop แบบ RS-F/F

R	S	Q_{n+1}	$\overline{Q}_n$
0	0	Q_n	$\overline{Q}_n$
0	1	1	0
1	0	0	1
1	1	ไม่ใช้	ไม่ใช้

Q และ $\overline{Q}$ จะตรงข้ามกันเสมอ คือ ถ้า Q = “1” , $\overline{Q}$ = “0”

Q = “0” , $\overline{Q}$ = “1”

ขา (5) เป็นวงจรทรานซิเลเตอร์ที่ทำหน้าที่ เป็นสวิตช์ คือถ้า Q เป็น “1” ทรานซิเลเตอร์จะทำงาน “on” นั่นคือจะ Short วงจร ระหว่างขา 7 และ 1 ถ้า $\overline{Q}$ เป็น “0” ทรานซิเลเตอร์จะทำงาน “off” นั่นคือจะ Open ขา 7 และ 1 ของไอซีเบอร์ 555

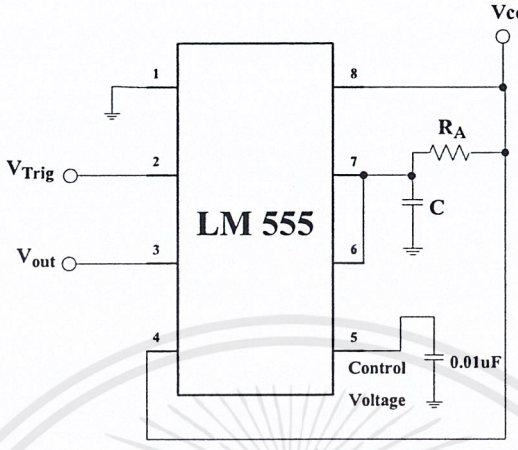
ขา (6) เป็นตัวต้านทาน 3 ตัว ที่มีขนาดเท่ากันทั้ง 3 ตัว มีหน้าที่แบ่งแรงดัน Vcc ออกเป็น 3 ส่วนเท่าๆ กัน ตกคร่อม R แต่ละตัว มีค่าเท่ากับ Vcc/3 ค่า R นี้จะมีขนาดความต้านทานเป็น กิโลโอห์ม

โดยอาศัยการทำงานเป็น 2 แบบคือ เป็นตัวผลิตสัญญาณความถี่ต่างๆ และเป็นวงจรนับสัญญาณจึงแบ่งการทำงานเป็นวงจรที่สำคัญได้ 2 ชนิดคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.1.1 วงจรโมโนสเตเบิล (Monostable Multivibator)

คือวงจรที่สร้างสัญญาณพัลส์ขึ้นมา 1 ลูก หลังจากมีการทริก (กระตุ้น) ให้วงจรเกิดสัญญาณ มีความกว้างที่สามารถกำหนดได้ด้วยค่า RC



รูปที่ 2.8 แสดงวงจร วงจรโมโนสเตเบิล ไอซีเบอร์ 555

หลักการทำงานวงจรโมโนสเตเบิล

ขณะที่ไม่มีสัญญาณทริก ที่ขา 2 V_{Trig} = Vcc แรงดันคร่อมคาปาซิสเตอร์จะเป็นศูนย์ เพราะว่า Vcc จะมีกระแสไหลผ่าน R_A ผ่าน ทรานซิสเตอร์ภายในลงกราวด์ได้ Vo เท่ากับศูนย์

เมื่อมีสัญญาณ Trig ที่ขา 2 คือ V_{Trig} = 0, ทรานซิสเตอร์ภายในจะ "off" Vcc จะทำการเก็บประจุ (Charge C) จนกระทั่งเกิดแรงดันตกคร่อมคาปาซิสเตอร์ประมาณ 2 Vcc/3 คาปาซิสเตอร์จะคายประจุผ่านทรานซิสเตอร์ภายในขณะที่คาปาซิสเตอร์คายประจุเอาท์พุทจะเป็น "1" ได้ Vo = Vcc

2.1.2.1.2 วงจรอะสเตเบิล (Astable Multivibator)

คือวงจรสร้างสัญญาณ Square wave มีความถี่ที่สามารถกำหนดได้ตามความต้องการด้วยค่าคาปาซิสเตอร์ C และค่าความต้านทาน R_A และ R_B

โดยในวงจรนี้นอกจากจะสามรถกำหนดความถี่ของสัญญาณ Square wave ด้วยค่า C และค่า $R_T = R_A + R_B$ แล้วยังสามารถกำหนดช่วง on และ off ของสัญญาณด้วยค่าที่แตกต่างกันของ R_A และ R_B ได้อีกด้วย แบ่งการทำงานออกเป็น 4 ช่วง

ช่วงที่ 1 ขณะที่ V_{Trig} = Vcc คอมพาราเตอร์ (2) ตัวล่างทำให้เอาท์พุทเป็น "0" (ขาลบมีศักดามากกว่าขาบวก) คอมพาราเตอร์ (1) ตัวบนทำให้เอาท์พุทเป็น "1" เพราะว่าขา 6 ต่อกับขา 7 ผ่านทรานซิสเตอร์ภายในลงกราวด์ทำให้ ทรานซิสเตอร์ภายใน "on" ได้เอาท์พุทของแนนเกทเป็น "0" เพราะอินพุททั้งคู่เป็น "1" ดังนั้น R = "0" และ S = "0" จะทำให้ $\overline{Q}$ = "1" ถ้า $\overline{Q}$ = "1" ทรานซิสเตอร์ภายในจะ "on" คาปาซิสเตอร์จะไม่มีการเก็บประจุ เนื่องจากกระแสจาก Vcc จะไหลผ่าน R และ ทรานซิสเตอร์ภายในลงกราวด์

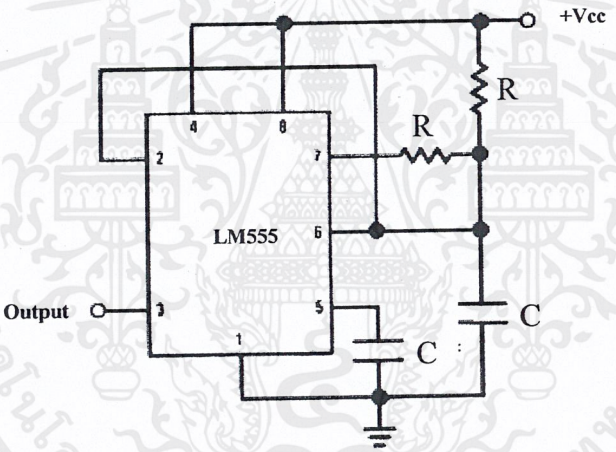
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"on" คาปาซิสเตอร์จะไม่มีการเก็บประจุ เนื่องจากกระแสจาก Vcc จะไหลผ่าน R และ ทรานซิสเตอร์ภายในลงกราวด์

ช่วงที่ 2 ขณะที่ trig ด้วย Vtrig = 0 V เอาท์พุทของคอมพาราเตอร์ (2) ตัวล่าง จะเปลี่ยนจาก "0" เป็น "1" ส่วนเอาท์พุทตัวคอมพาราเตอร์ (1) ตัวบนยังคงเหมือนเดิม ดังนั้นที่วงจร F/F ขา R = "0" และ S = "1" จะเป็นการเซต F/F ได้ Q = 1 และ $\overline{Q}$ = "0" เมื่อ $\overline{Q}$ = "0" ทำให้ทรานซิสเตอร์ภายในเกิดการ "off" ส่งผลให้คาปาซิสเตอร์จะทำการเก็บประจุ

ช่วงที่ 3 เมื่อสัญญาณ Trig กลับเป็น Vcc อีกครั้ง เอาท์พุทของคอมพาราเตอร์ (2) ตัวล่าง จะเปลี่ยนจาก "1" เป็น "0" ทำให้เอาท์พุทของคอมพาราเตอร์ที่ส่งไปให้วงจร F/F ไม่เปลี่ยนแปลง

ช่วงที่ 4 เมื่อคาปาซิสเตอร์ถูกเก็บประจุจนกระทั่ง Vc มีค่าเท่ากับ 2/3 (Vcc) หรือมากกว่านั้น ส่งผลให้เอาท์พุทของคอมพาราเตอร์ (1) ตัวบน เปลี่ยนจาก "1" เป็น "0" ทำให้เอาท์พุทของแนนเกทเท่ากับหนึ่ง ที่วงจร F/F ขา R = "1" และ S = "0" ดังนั้น Q = "0", $\overline{Q}$ = "1" ทำให้ ทรานซิสเตอร์ภายใน "on" อีกครั้ง คาปาซิสเตอร์จะทำคายประจุผ่านทรานซิสเตอร์ภายในลงกราวด์ กลับสู่สภาพเดิม



รูปที่ 2.9 แสดงวงจรอะสเตเบิล ไอซีเบอร์ 555

หลักการทำงานอะสเตเบิล

ใช้หลักการเก็บประจุ และคายประจุของคาปาซิสเตอร์โดยจะทำการเก็บประจุผ่าน R_A, R_B และจะคายประจุผ่าน R_B และ ทรานซิสเตอร์ภายในลงกราวด์

- ขณะที่คาปาซิสเตอร์เก็บประจุจนถึงระดับแรงดัน 2/3(Vcc) ขณะนั้นแรงดันของ Vo = Vcc
- ขณะที่คาปาซิสเตอร์คายประจุออก เกิดแรงดันตกคร่อมมีช่วงจาก 2/3(Vcc/3) ขณะนั้นแรงดันของ Vo = Vcc แบ่งการทำงานออกเป็น 5 ช่วง

ช่วงที่ 1 เป็นช่วงที่ทำการเปิดวงจร คาปาซิสเตอร์จะทำการเก็บประจุจาก Vcc ผ่าน R_A, R_B เอาท์พุทของคอมพาราเตอร์ตัวบนและตัวล่างจะเป็น "1" ทำให้เอาท์พุทของแนนเกท จะเป็น "0" ส่งผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ F/F ขา R = "0" และ S = "1" เป็นการเซต F/F เอาท์พุต Q = "1", $\overline{Q}$ = "0" คาปาซิสเตอร์จะทำการเก็บประจุไปเรื่อยจนมีแรงดันตกคร่อม เท่ากับ Vcc/3

ช่วงที่ 2 ช่วงที่คาปาซิสเตอร์ ทำการเก็บประจุเกิน Vcc/3 จะทำให้คอมพาราเตอร์ (2) ตัวล่างเป็น "0" F/F ขา R = "0" และ S = "0" ได้เอาท์พุท Vo เหมือนเดิม

ช่วงที่ 3 เมื่อคาปาซิสเตอร์ถูกเก็บประจุจนกระทั่งมีค่ามากกว่า 2(Vcc/3) เอาท์พุทของคอมพาราเตอร์ (1) ตัวบน จะเปลี่ยนเป็น "0" เอาท์พุทของแนนเกทจะเป็น "1" ดังนั้น F/F ขา R = "1" และ S = "0" เป็นการทำให้ F/F ขา Q = "0" และ $\overline{Q}$ = "1" ทรานซิสเตอร์จะ on

ช่วงที่ 4 คาปาซิสเตอร์จะทำคายประจุผ่าน R_B และทรานซิสเตอร์ภายในลงกราวด์ ทำให้เอาท์พุทของคอมพาราเตอร์ (1) ตัวบน เป็น "1" คอมพาราเตอร์ (2) ตัวล่าง เป็น "0" ทำให้ F/F ขา R = "0" และ S = "0" เอาท์พุทของ F/F จะคงเดิม Q = "0", $\overline{Q}$ = "1"

ช่วงที่ 5 เมื่อคาปาซิสเตอร์ทำคายประจุ จนแรงดันตกคร่อม C = Vcc/3 ทำให้คอมพาราเตอร์ (1) ตัวบน เป็น "1" คอมพาราเตอร์ (2) ตัวล่าง เป็น "1" ทำให้ F/F ขา R = "0" และ S = "1" เป็นการทำให้ F/F ได้ Q = "1", $\overline{Q}$ = "0" ทรานซิสเตอร์ภายในจะ off คาปาซิสเตอร์จะทำการเก็บประจุใหม่อีกครั้ง ในขั้นตอน 2 และเป็นเช่นนี้เรื่อยๆ

2.1.3 อินฟราเรด

อินฟราเรด กล่าวคือ คลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นอยู่ระหว่าง 10^{-4} m ถึง $7x10^{-7}$ m คือ แสงสีแดง และความยาวคลื่นจะแปรผกผันกับความถี่ ถ้าความยาวคลื่นมาก ความถี่จะน้อย ในทางกลับกันความยาวคลื่นน้อย ความถี่จะมากเป็นคลื่นที่มีความถี่ถัดจากความถี่ของสีแดงลงมา (infa = ใต้, red = แดง) เราจึงมองไม่เห็น กล่าวคืออยู่ในช่วง (สีรุ้ง, ม่วง, คราม, น้ำเงิน, เขียว, เหลือง, แสด, แดง) เนื่องจากความถี่เป็นสัดส่วนกับพลังงาน ถ้าความถี่มาก พลังงานจะมากตามความถี่ โดยคลื่น Infrared จึงเป็นคลื่นที่มีพลังงานต่ำ ใช้ได้ในระยะไม่ไกลนัก ช่วงที่มองเห็นต้องมีความยาวคลื่นในช่วง $7x10^{-7}$ ลำแสงสีแดง ถึง $4x10^{-7}$ ลำแสงสีม่วง ซึ่งภายในโครงงานนี้จะอินฟราเรดเป็นตัวรับตัวส่งอธิบายดังนี้

2.1.3.1 ตัวส่งอินฟราเรด

ตัวส่งไดโอดเปล่งแสงชนิดอินฟราเรด วัสดุสำหรับ LED ที่เปล่งแสงอินฟราเรดที่นิยมใช้มี 3 ชนิด คือ GaAs, GaAlAs และ GaInAsP โดยมีรายละเอียดดังนี้

-ไดโอดเปล่งแสงชนิด GaAs

<u>กรณีโดปด้วย Si</u>

GaAs เป็นสารประกอบกึ่งตัวนำตระกูล III-V เนื่องจาก Si เป็นธาตุตระกูล IV ดังนั้น Si ที่ถูกโดปเข้าไปใน GaAs จึงสามารถเป็นได้ทั้งโดเนอร์และแอกเซ็ปเตอร์ โดยขึ้นอยู่กับเงื่อนไขในการผลิต ทำให้ได้รอยต่อ p-n ได้ง่ายและเป็น LED ที่ให้แสงอิฟราเรดที่มีเอาต์พุตสูงมาก ในการปลูกผลึก GaAs ด้วยวิธี LPE นั้นจะเติมธาตุ Si เข้าไปในเบ้าของ Ga และเมื่อลดอุณหภูมิลงจาก 900 °c จะได้ผลึกเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิด n และเมื่ออุณหภูมิต่ำลงกว่า 850 °c ก็จะได้ผลึกเป็นชนิด p LED ชนิดนี้มีประสิทธิภาพสูงมาก ประมาณ 6% จัดว่าเป็น LED ที่ผลิตแบบจำนวนมากๆที่ได้คุณภาพสูง จึงมีการนำไปใช้งานกันอย่างแพร่หลายในรีโมตคอนโทรล

<u>กรณีโดปด้วย Zn</u>

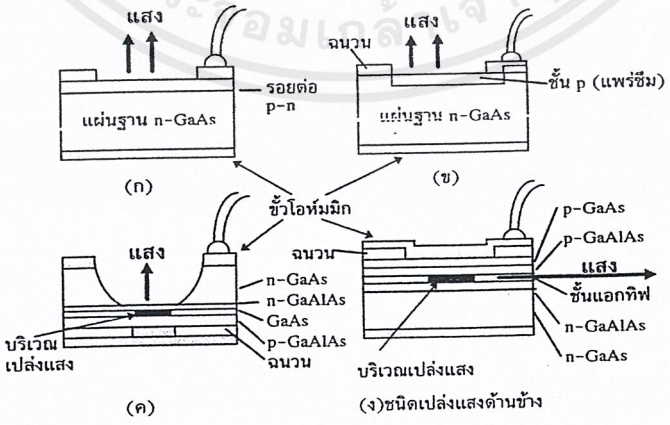
ใช้ n-GaAs เป็นแผ่นฐาน จากนั้นจึงแพร่ซึมธาตุ Zn เข้าไปเพื่อให้เกิดรอยต่อ p-n เมื่อฉีดพาหะเข้าสู่ Led ชนิดนี้ อิเล็กตรอนจากแถบคอนดักชันจะรวมตัวกับโฮลในแถบเวเลนซ์โดยตรง ดังนั้นผลตอบสนอง ต่อเวลาจะรวดเร็วมากเป็นหน่วยของนาโนวินาที ให้แสงความยาวคลื่น 910 nm แต่มีข้อเสียว่าเนื่องจากความยาวคลื่นแสงเท่ากับช่องว่างพลังงานจึงทำให้แสงถูกดูดกลืนภายในด้วยแผ่นฐานมาก ทำให้เอาต์พุตแสงมีค่าต่ำ

- ไดโอดเปล่งแสงชนิด GaAlAs

LED ชนิด GaAlAs มีโครงสร้างแบบรอยต่อดับเบิลเฮเทโร (Double Heterojunction) คล้ายเลเซอร์ไดโอด แต่ชั้นเปล่งแสงตรงกลางหนากว่าชั้นเปล่งแสงในเลเซอร์ไดโอด และมีการออกแบบให้แสงเปล่งออกจากระนาบได้ดีดังโครงสร้างในรูป 7.4.16 เอาต์พุตของแสงมีค่าสูงมาก สามารถใช้ในการสื่อสารด้วยแสงได้ดี นอกจากนี้การออกแบบให้ LED มีรูปเป็นโดมจะยิ่งทำให้แสงวิ่งออกมาจาก LED ได้ดียิ่งขึ้น บางรุ่นมีประสิทธิภาพควอนตัมสูงกว่า 50% ก็มี

- ไดโอดเปล่งแสงชนิด GaInAsP

LED ชนิดนี้เปล่งแสงอินฟราเรดที่มีความยาวคลื่นย่าวกว่าชนิดอื่นที่กล่าวมาแล้วข้างต้น ถูกออกแบบขึ้นมาเพื่อใช้กับเส้นใยแก้วนำแสงโดยเฉพาะ ซึ่งต้องการแสงความยาวคลื่น 1.3 และ 1.55 μm เนื่องจากการสูญเสียของแสงภายในเส้นใยแก้วนำแสงมีค่าน้อยที่สุดที่ความยาวคลื่น 1.3 และ 1.55 μm ดังนั้น LED ชนิดนี้ จึงเหมาะสมมากที่สุดในการใช้งานในระบบการสื่อสารด้วยแสง



รูปที่ 2.10 โครงสร้างของ LED ชนิดแสงอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 ตัวรับอินฟราเรด

ตัวรับอินฟราเรดนี้ ขาที่ยาวเป็นขาอาโนด ขาที่สั้นเป็นขาคาโทด มันจะมีสัญญาณก็ต่อเมื่อมีแสงอินฟาเรดเข้ามา และตัวรับอินฟราเรดตัวนี้จะเรโซแนนซ์ที่ความถี่ 38 KHz ซึ่งวงจรภายในจะตรวจจับเอาเฉพาะสัญญาณที่มีความแรงสูงสุด สามารถทำงานกลางแจ้ง แสงแดดแรงได้โดยไม่โดนรบกวนจากบรรยากาศโดยรอบ โดยปกติแล้วแสงแดดก็เป็นตัวรบกวนตัวรับของเรา เพราะแสงแดดมีช่วงความถี่คลื่นที่กว้าง ในช่วงความถี่คลื่นที่กว้างนี้ มีความถี่บางความถี่ที่มีความยาวคลื่นที่มีผลต่อตัวรับอินฟราเรดแต่ที่ยังทำงานได้เนื่องจากกำลังสัญญาณรบกวนของแสงในย่านนี้ น้อยกว่ากำลังของสัญญาณที่ตัวส่ง ซึ่งเราจะสังเกตเห็นว่า ตัวถังของตัวรับอินฟราเรดต่างๆ มีสีดำทึบ วัสดุชนิดนี้ที่จะช่วยกรองแสงในย่านที่เราไม่ต้องการออก เพราะฉะนั้นตัวรับนี้ใช้ตรวจจับแสงอินฟราเรดที่ผสมสัญญาณพาห์ความถี่ที่ 38 KHz ซึ่งให้ผลเป็นลอจิก “1” เมื่อตรวจจับสัญญาณไม่ได้ ให้ผลเป็นลอจิก “0” เมื่อตรวจจับสัญญาณแสงได้

2.1.4 ทฤษฎีการขับโมดูลแสดงผลแบบผลึกเหลว

ตัวแสดงผล (DISPLAY) ภายในผลึกเหลวสามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอแอลซีดี ตัวคอนโทรลเลอร์ (CONTROLLER) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกควบคุมการทำงานโมดูลแอลซีดี เช่น จอลบภาพ แสดงตัวอักษร หรือที่เลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะที่นิยม คือเบอร์ HD44780 จะใช้ควบคุมแอลซีดีแบบอักขระ

ตัวขับ (DRIVER) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลข้อมูลตามที่กำหนดชิปที่ใช้ทำหน้าที่นี้ได้แก่เบอร์ HD44100H และ MSM5259 เป็นต้น

2.1.5 โครงสร้างภายในตัวควบคุมโมดูลแอลซีดี

ในการใช้งานโมดูล (MODULE) แอลซีดีจำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน ในที่นี้ขอยกตัวอย่างโมดูลแอลซีดี แบบอักขระเพราะสามารถเข้าใจได้ง่ายและจะเป็นบล็อกไดอะแกรมภายในชิปควบคุมแอลซีดีเบอร์ HD44780 ซึ่งใช้ในโมดูลแอลซีดี แบบอักขระประกอบด้วย

บัฟเฟอร์อินเอาท์พุท เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอกเพื่อที่ถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งอุปกรณ์ภายนอก เพื่อนำไปควบคุมแสดงผล

รีจิสเตอร์ข้อมูล เป็นรีจิสเตอร์รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล เป็นหน่วยความจำที่ทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมนี้จะนำข้อมูลในดีดีแรมนี้ไปเปิดตาราง (LOOK UP LABLE) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษรเพื่อนำไปแสดงผลที่ตัวแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

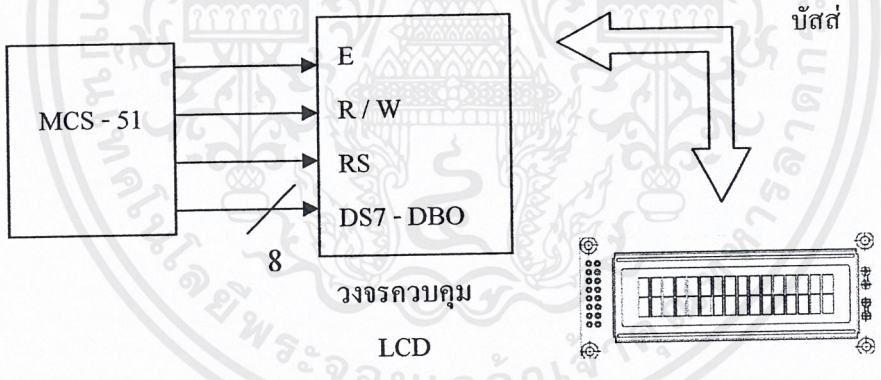
รอมเก็บตัวอักษร เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญญาลักษณ์ที่อ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลในดีดีแรม

แรมเก็บตัวอักษร เป็นความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนการอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน ดีดีแรม แล้วตัวควบคุมจะมีอ่านค่าจาก CGROM เอง

แฟลก บีซี่ (BUSY) เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลคำสั่งหรือไม่ ดังนั้น ก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลกบีซี่ (BUSY) นี้เสียก่อน

2.1.6 แอลซีดีโมดูล (LCD Module)

LCD (Liquid Crystal Display) เป็นอุปกรณ์แสดงผลได้ทั้งตัวเลข ตัวอักษร และกราฟิก เป็นอุปกรณ์แบบหนึ่งที่ได้รับความนิยมในการใช้งานสูงมาก โดยเฉพาะอย่างยิ่งเมื่อนำมาเชื่อมต่อกับไมโครคอนโทรลเลอร์เนื่องจากราคาถูก ปัจจุบันได้มีการพัฒนาแอลซีดีที่สามารถแสดงสีได้ด้วย การใช้งานแอลซีดี จะต้องมีวงจรควบคุมการทำงานแยกต่างหาก โดยปกติจะรวมอยู่กับจอแอลซีดี จึงรวมเรียกกันว่าแอลซีดีโมดูล การเชื่อมต่อแอลซีดีโมดูลกับไมโครคอนโทรลเลอร์แสดงได้ดังรูป

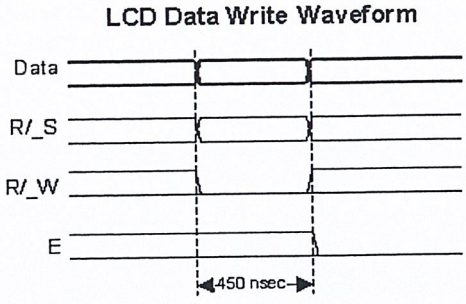


รูปที่ 2.11 วงจรควบคุมแอลซีดีเชื่อมต่อ กับจอแอลซีดีผ่านบัสส่งข้อมูล

จากรูปวงจรควบคุมแอลซีดีจะเชื่อมต่อกับจอแอลซีดีผ่านบัสส่งข้อมูล โดยวงจรควบคุมจะมีขาเชื่อมต่อไปยังไมโครคอนโทรลเลอร์ ได้แก่ ขา Vss, ขา Vdd, ขา Vo, ขา E, ขาR/W, ขาRS และขาข้อมูล DB7 ถึง DB0 โดยหน้าที่ของขาต่างๆ อธิบาย ดังนี้

- ขา Vss เป็นขาที่ต่อกราวน์ของวงจร
- ขา Vdd เป็นขาที่ต่อไฟเลี้ยง
- ขา Vo เป็นขาอินพุทรับแรงดันเพื่อปรับความเข้มของการแสดงผล
- ขา E (Enable) เป็นขา enable ให้แอลซีดีทำงานโดยต้องป้อนสัญญาณพัลส์ (Pulse) เข้าไปโดยพัลส์แต่ละลูกจะต้องมีความกว้าง 450 นาโนวินาทีเป็นอย่างน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

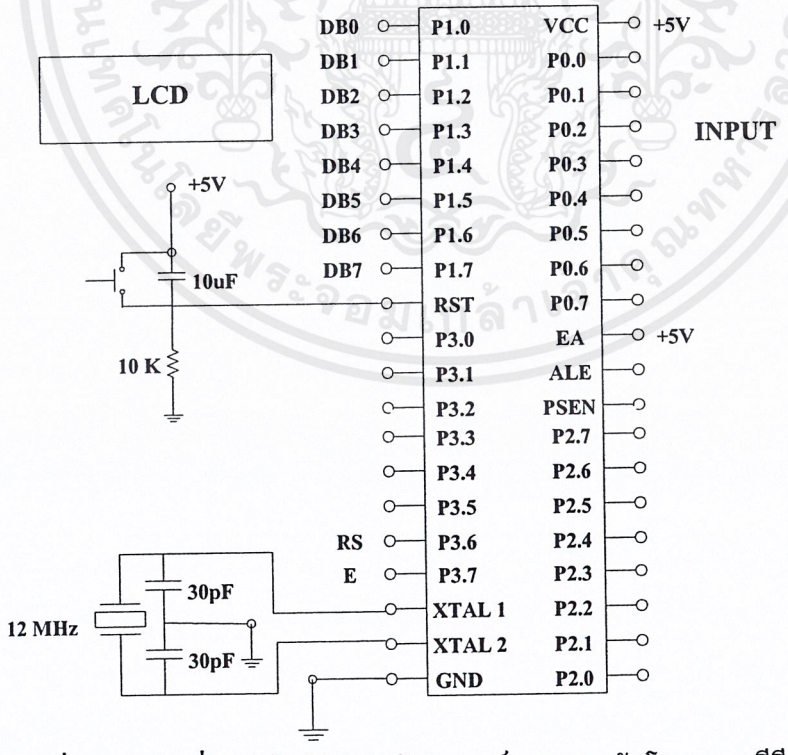


รูปที่ 2.12 การป้อนสัญญาณพัลส์ให้แอลซีดีทำงาน

- ขา R/W (Read/Write) ถ้าเป็นลอจิก “1” จะเป็นการอ่านข้อมูลจากแอลซีดี และถ้าเป็นลอจิก “0” จะเป็นการเขียนข้อมูลไปที่แอลซีดี

- ขา RS (Register Select) เนื่องจากภายในแอลซีดีโมดูล มีรีจิสเตอร์ที่สำคัญ 2 ตัวคือ รีจิสเตอร์เก็บคำสั่ง และรีจิสเตอร์เก็บข้อมูล โดยขา RS นี้จะเป็นตัวกำหนดว่าข้อมูลที่ส่งไปจะเป็นคำสั่งหรือว่าข้อมูล ถ้าเป็นลอจิก “1” แสดงว่าข้อมูลที่ส่งไปเป็นข้อมูลที่จะแสดงบนหน้าจอแอลซีดี แต่เป็นลอจิก “0” แสดงว่าข้อมูลที่ส่งไปนั้นเป็นคำสั่ง

- ขา DB7-DB0 ใช้รับ-ส่งข้อมูลกับไมโครคอนโทรลเลอร์



รูปที่ 2.13 วงจรเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับโมดูลแอลซีดี 16x2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6.1 คำสั่งควบคุมโมดูลแอลซีดีที่สำคัญ

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น 0 แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูลแอลซีดี ของชิปควบคุม HD44780 ที่สำคัญ

- คำสั่งเคลียร์หน้าจอแสดงผล

มีข้อมูลคำสั่งเป็น 0x01 เป็นคำสั่งเขียนข้อมูลช่องว่างหรือ Space เข้าไปในหน่วยความจำข้อมูลสำหรับแสดงผลหรือ DDRAM ภายในโมดูลแอลซีดี เมื่อเคลียร์จอแสดงผลแล้ว จะกำหนดให้เคอร์เซอร์ (cursor : สัญลักษณ์พิเศษที่ใช้ในการแสดงตำแหน่งสำหรับแสดงตัวอักษรโมดูลแอลซีดี บางครั้งมีลักษณะเป็นรูปสี่เหลี่ยมผืนผ้าขนาด 5x7 จุดหรือเป็นขีด สามารถสั่งให้กระพริบหรือไม่ก็ได้) กลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล

- คำสั่ง Return Home

มีค่าของข้อมูลเท่ากับ 0x02 หรือ 0x03 ก็ได้ (แนะนำให้ใช้ 0x02) เป็นการกำหนดให้เคอร์เซอร์ไปแสดงยังตำแหน่งซ้ายสุดของจอแสดงผล โดยข้อมูลที่แสดงบนหน้าจอแสดงผลจะไม่มีการเปลี่ยนแปลงนั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode set)

มีรูปแบบคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

บิต I/D ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของหน่วยความจำแสดงผล เพิ่มหรือลดลง

"0" แอดเดรสลดลง 1 แอดเดรส

"1" แอดเดรสเพิ่มขึ้น 1 แอดเดรส

บิต S ใช้กำหนดลักษณะการแสดงผล

"0" เมื่อเกิดตัวอักษรใหม่ เคอร์เซอร์เลื่อนไปทางขวามือ

"1" เมื่อเกิดตัวอักษรใหม่ เคอร์เซอร์อยู่ที่เดิมแต่ตัวอักษรเลื่อนไปทางซ้าย

ที่นิยมใช้กันมากที่สุดคือข้อมูลคำสั่ง 0x06 หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของหน่วยความจำสำหรับแสดงผลจะเพิ่มขึ้น

- คำสั่งควบคุมการแสดงผล

มีรูปแบบคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าเป็น "0" เป็นการปิดจอแสดงผล ถ้าเป็น "1" เป็นการเปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าเป็น "0" เป็นการปิดตัวเคอร์เซอร์หรือไม่แสดงเคอร์เซอร์ ถ้าเป็น "1" เป็นการแสดงตัวเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าต้องการให้เคอร์เซอร์กะพริบ ต้องกำหนดให้เป็น "1"

คำสั่งที่ใช้บ่อยคือ 0x0C เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0x0F เป็นการสั่งให้เปิดจอแสดงผลและแสดงเคอร์เซอร์ด้วย

- คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร

มีรูปแบบคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	1	S/C	R/L	*	*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลื่อนเคอร์เซอร์และตัวอักษรขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14h-17h
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1bh
1	1	เลื่อนตัวอักษรใหม่ไปทางซ้าย	1ch-1fh

- กำหนดฟังก์ชั่นการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	1	DL	N	F	*	*

บิต DL ใช้ในการกำหนดจำนวนบิตในการติดต่อกับโมดูลแอลซีดี

"0" กำหนดให้ทำงานในโหมด 4 บิต

"1" กำหนดให้ทำงานในโหมด 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดที่ต้องการให้แสดงผล

"0" แสดงผล 1 บรรทัด

"1" แสดงผล 2 บรรทัดหรือมากกว่า

แต่สำหรับโมดูลแอลซีดีขนาด 16 อักษร 1 บรรทัดที่มีจำหน่ายในประเทศไทย ต้องกำหนดให้บิตนี้เป็น "1" เนื่องจากหน่วยความจำสำหรับเก็บข้อมูลเพื่อการแสดงผลไม่ได้อยู่ต่อเนื่องกัน กล่าวคือแบ่งเป็น 2 ช่วงคือ ที่แอดเดรส 0x00 – 0x07 และ 0x40 – 0x47 โดยแอดเดรส 0x40 จะเป็นแอดเดรสเริ่มต้นของหน่วยความจำสำหรับการแสดงผลในบรรทัดที่สองของโมดูลแอลซีดี ที่มีมากกว่า 1 บรรทัด จึงทำให้เมื่อต้องการติดต่อกับโมดูลแอลซีดี 1 บรรทัด จึงจำเป็นต้องกำหนดให้เป็น "1"

บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล

"0" แสดงผลแบบ 5x7 จุด

"1" แสดงผลแบบ 5x10 จุด

- คำสั่งการเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น 0 บิต 6 เป็น 1 ส่วนอีก 6 บิต ที่เหลือจะแสดงด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งก่อนที่จะอ่านหรือเขียนข้อมูลให้ โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00-3H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น 1 และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8ch-0ffh ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานที่บิต N ด้วย หากบิต N เป็น 0 แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CH และถ้าบิต เป็น 1 แอดเดรสของ จอยู่ระหว่าง และถ้าบิต เป็น 1 แอดเดรสของ จะมี 2 ช่วงคือ 8CH-87H และ 0C0H-0C7H

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้จะบอกสถานะของตัวควบคุมแอลซีดีพร้อมที่จะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น 0 แสดงว่าตัวควบคุมแอลซีดีว่าพร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น 1 แสดงว่า ขณะนี้ ตัวควบคุมแอลซีดี ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น 1 ด้วย แต่สัญญาณที่ ยังต้องเป็น อยู่เพราะข้อมูลนั้นเป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAMและ DDRAM ด้วย โดยบิต 0- บิต 6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

- การเขียนคำสั่งและข้อมูลให้โมดูลแอลซีดี

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูลแอลซีดีแสดงผลตามที่ผู้ใช้งานต้องการนั้นจะต้องส่งคำสั่ง (Instruction) แล้วกำหนดโหมดการทงานให้แก่โมดูลของแอลซีดีก่อนหลังจากนั้นจึงค่อยส่งข้อมูล (data) ที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของโมดูลแอลซีดี มี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งแลข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก 0 หมายความว่าข้อมูลที่ป้อนให้กับโมดูลแอลซีดี ขณะนี้เป็นคำสั่งในทางตรงข้ามหากขาRS ได้รับลอจิก 1 ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการอ่านหรือเขียนข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น 1 เพื่อแจ้งให้ตัวควบคุมภายในโมดูลแอลซีดี ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านหรือเขียนข้อมูลต้องกำหนดให้ขา R/W เป็น 1 ข้อมูล ขนาด 8 บิต หรือ 4 บิต ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ **CGRAM** หรือ DDRAM ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จังหวะการทำงานของแอลซีดีโมดูล

ในการติดต่อกับโมดูลแอลซีดี จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่ง หรือข้อมูล เนื่องจากต้องรอให้คอนโทรลเลอร์ภายในแอลซีดีโมดูลแปลความหมายรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยเสียก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

ดังนั้นในการใช้งานโมดูลแอลซีดีผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้โมดูลแอลซีดีพร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่โมดูลแอลซีดี ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้โมดูลแอลซีดี ทำการเตรียมความพร้อมหรืออินนิเชียล หลังจากนั้นก็กำหนดลอจิกให้แก่ขา RS ของโมดูลแอลซีดี แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาที เพื่อให้คอนโทรลเลอร์แอลซีดีโมดูล แปลความหมายของขาลอจิกที่ขา RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการจะไปแสดงผล จากนั้นจะเป็นการส่ง มูลมารอที่บัสข้อมูล D0-D7 ขั้นตอนต่อไปนี้จะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิลโมดูลแอลซีดี ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของโมดูลแอลซีดี ต้องเป็นพัลส์ของขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะการทำงาน 1 รอบของโมดูลแอลซีดี จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรมอินนิเชียลแอลซีดี โปรแกรมหน่วงเวลา และโปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิลโมดูลแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 แสดงคำสั่งควบคุม LCD

คำสั่ง (ฐานสิบหก)	คำอธิบาย
01	เคลียร์หน้าจอแอลซีดี พร้อมเลื่อนเคอร์เซอร์ไปยังตำแหน่งเริ่มต้นซ้ายสุด
02	เลื่อนเคอร์เซอร์ไปยังตำแหน่งเริ่มต้นซ้ายสุด
04	ทุกครั้งที่ข้อมูลแสดง เลื่อนเคอร์เซอร์ไปทางขวาอัตโนมัติ
05	เลื่อนเคอร์เซอร์ไปทางขวา
06	ทุกครั้งที่ข้อมูลแสดง เลื่อนเคอร์เซอร์ไปทางซ้ายอัตโนมัติ
07	เลื่อนเคอร์เซอร์ไปทางซ้ายอัตโนมัติ
08	ไม่แสดงข้อความและไม่แสดงเคอร์เซอร์
0A	ไม่แสดงข้อความ แต่แสดงเคอร์เซอร์
0C	แสดงข้อความ ไม่แสดงเคอร์เซอร์
0E	ไม่แสดงข้อความและไม่แสดงเคอร์เซอร์
0F	ไม่แสดงข้อความและไม่แสดงเคอร์เซอร์กระพริบ
10	เลื่อนตำแหน่งเคอร์เซอร์ไปทางซ้าย
14	เลื่อนตำแหน่งเคอร์เซอร์ไปทางขวา
18	เลื่อนตัวอักษรไปทางซ้าย
1C	เลื่อนตัวอักษรไปทางซ้าย
38	กำหนดแสดงผลแบบ 2 บรรทัด ขนาดตัวอักษร 5x8 จุด และส่งข้อมูลครั้งละ 8 บิต

2.2 ลักษณะลายนิ้วมือ

โดยทั่วไปผิวหนังบริเวณปลายนิ้วมือ ประกอบด้วยลายเส้นสองชนิด ชนิดที่หนึ่งเราเรียกว่าเส้นนูน (Ridges) อีกชนิดเรียกว่า รอยร่อง หรือ เส้นร่อง (Furrow) จะอยู่สลับกันตลอดไป โดยถ้าเราใช้หมึกสีดำทาบบนนิ้วมือ และกดนิ้วมือลงบนกระดาษขาวจะได้ลายเส้นสีขาวและดำสลับกัน เราเรียกว่าเส้นสีดำว่าเส้นนูน เราเรียกเส้นสีขาวว่าเส้นร่อง

รูปที่ 2.14 แสดงลักษณะลายมือทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดลักษณะสำคัญบนลายนิ้วมือ (Characteristics) คือ ตำหนิต่างๆ บนลายนิ้วมือ สามารถแบ่งได้เป็น 2 ลักษณะดังนี้

1. ตำหนิและลักษณะต่างๆ ของลายเส้นทั่วๆ ไปเช่น เส้นตรง, เส้นโค้ง, จุด, เส้นแตก, เส้นวกกลับ, เส้นเวียน, เส้นขาด, เส้นทะเลสาบและเส้น 2 เส้นมาพบกัน

2. ลักษณะพิเศษบางอย่าง เช่น

ไบฟูร์เคชัน คือ เส้นขอบหนึ่งที่ได้ถูกแยกออกเป็น 2 เส้นหรือมากกว่า 2 เส้น

ไดเวอร์เจนซ์ คือ เส้นขอบที่ได้วิ่งขนานกันมาหรือเกือบจะขนานและได้แยก ถ่างออกไป

จุดมินูเทีย (Minutiae) คือ จุดบนปลายเส้นหยุดหรือเส้นแยก

เราสามารถแยกลักษณะของลายนิ้วมือที่เป็นจุดสำคัญต่างๆ ดังนี้

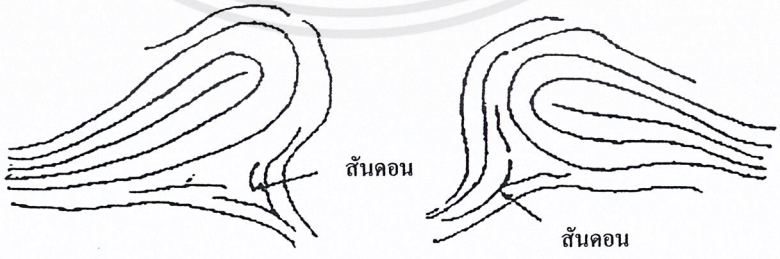
2.2.1 เส้นขอบ (Type Line)

หมายถึง เส้นขนานคู่ในสุดซึ่งได้เดินขบวนกันมาพอสมควรแล้วควรแยกออกจากกัน เพื่อโอบล้อมหรือพยายามโอบล้อมบริเวณลายนิ้วมือที่อยู่ภายในและ เส้นขอบไม่จำเป็นจะต้องเป็นเส้นยาว และราบเรียบติดต่อกันตลอดอาจจะเป็นเส้นที่ขาดกลางเส้น แต่เราก็ยังจัดเป็นเส้นขอบด้วย

รูปที่ 2.15 เส้นขอบ

2.2.2 เส้นดอน (Delta)

หมายถึง ลายเส้นในนิ้วมือ ซึ่งอยู่ตรงหน้า และใกล้ที่สุดกับกึ่งกลางปากทางแยกเส้นขอบ

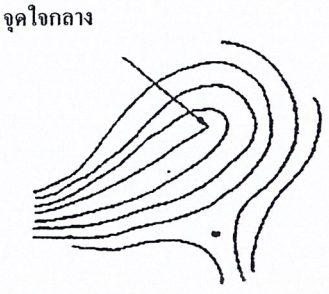


รูปที่ 2.16 เส้นดอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 จุดกลางใจ (Core)

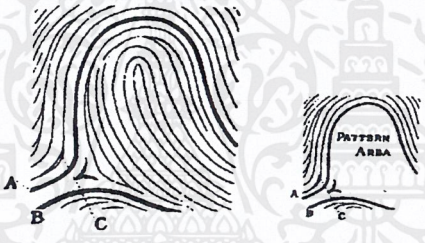
หมายถึง จุดใดจุดหนึ่งบนปลายเส้น หรือไหล่ของเส้นวกกลับที่อยู่บนสุด



รูปที่ 2.17 จุดใจกลาง

2.2.4 บริเวณลายนิ้วมือที่อยู่ภายใน (Pattern Area)

หมายถึง พื้นที่บริเวณภายในของลายนิ้วมือที่ถูกเส้นขอบล้อมไว้



รูปที่ 2.18 ลักษณะของลายนิ้วมือที่เป็นจุดสำคัญ

2.3 ชนิดและรูปแบบของลายนิ้วมือ

เราสามารถจัดรูปแบบลายนิ้วมือออกเป็น 4 กลุ่มคือ

2.3.1 กลุ่มเส้นโค้ง (Arch)

เป็นแบบแผนลายนิ้วมือที่มีลักษณะลายเส้นตั้งต้นจากขอบเล็บข้างหนึ่งไหลออกไปอีกข้างหนึ่ง โดยไม่มีจุดศูนย์กลางไม่มีจุดสันดอน แบบโค้งยังแบ่งออกได้เป็น 2 แบบแผน ได้แก่ โค้งราบ (plain arch) มีลักษณะของเส้นโค้งไม่สูงชัน ซึ่งต่างจากแบบโค้งกระโจมหรือโค้งเจดีย์ (tented arch) ที่มีเส้นโค้งตรงกลางหนึ่งเส้นหรือมากกว่าหนึ่งเส้นพุ่งขึ้นพบกันตรงกลางเป็นมุมแหลมหรือมุมฉาก ลายนิ้วมือแบบโค้งจึงไม่มีจุดสันดอน และไม่มีจุดศูนย์กลาง ดังนั้นจำนวนเส้นลายนิ้วมือจึงเป็นศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.19 กลุ่มเส้นโค้ง

2.3.1.1 โค้งราบ (Plain Arch =PA)

ลายเส้นวิ่งหรือไหลออกไปข้างหน้า ไม่เกิดมุมแหลมหรือเส้นพุ่งขึ้นตรงกลาง

รูปที่ 2.20 โค้งราบ และโค้งราบสูง

2.3.1.2 โค้งกระโจม (Tented Arch=TA)

ลายเส้นตรงกลางเกิดเป็นเส้นพุ่งขึ้นจากแนวนอนเป็นมุมแหลมหรือมุมฉาก

รูปที่ 2.21 โค้งกระโจม

2.3.2 กลุ่มมัดหวาย (Loop)

เป็นแบบแผนลายนิ้วมือที่พบมาก ประมาณ 65% ของแบบแผนทั้งหมด มีลักษณะเป็นรูปเกือกม้าที่มีปลายเส้นเกือกม้าปัดออกไปทางใดทางหนึ่ง (นิ้วก้อย หรือนิ้วหัวแม่มือ ของมือนั้น) ถ้าปลายเส้นเกือกม้าปัดไปทางนิ้วก้อย เรียกว่า มัดหวายปัดก้อย (ulnar loop) ถ้าปัดไปทางนิ้วหัวแม่มือ เรียกว่า มัดหวายปัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวแม่มือ (radial loop) ลายนิ้วมือแบบมัดหวายทั้งสองแบบจะมีจุดสันดอนหนึ่งแห่งและจุดศูนย์กลางหนึ่งจุด จำนวนเส้นลายนิ้วมือ (ridge count) จึงมีหนึ่งจำนวน คือจำนวนเส้นจากจุดศูนย์กลางถึงจุดสันดอน

รูปที่ 2.22 กลุ่มมัดหวาย

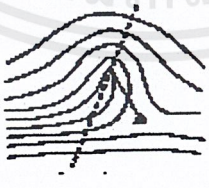
2.3.2.1 มัดหวายปัดขวา (Right Slant Loop =RSL)

มีเส้นดอยเพียงจุดเดียว มีเส้นวกหลักที่สมบูรณ์อย่างน้อยหนึ่งเส้นมีทิศทางไปด้านขวา

รูปที่ 2.23 มัดหวายปัดขวา

2.3.2.2 มัดหวายปัดซ้าย (Left Slant Loop=LSL)

มีสันดอนเพียงจุดเดียว มีเส้นวกหลักที่สมบรูณ์อย่างน้อย 1 เส้นมีทิศทางไปด้านซ้าย



รูปที่ 2.24 มัดหวายปัดซ้าย

2.3.2.3 มัดหวายคู่ (Double =D)

มีลักษณะคล้ายกับลายนิ้วมือแบบมัดหวายข้างบนแต่มากอดหรือกล้ำกันจนเกิดสันดอน 2 จุด โดยไม่จำเป็นต้องมีขนาดเท่ากัน ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.25 มัดหวายคู่

2.3.3 กลุ่มก้นหอย (Whorl =W)

ลายนิ้วมือที่มีเส้นเวียนรอบเป็นวงจรลักษณะคล้ายรูปไข่วงกลมหรือมีลักษณะอื่นๆ ประกอบด้วย

เป็นแบบแผนลายนิ้วมือที่พบประมาณ 30% ของแบบแผนลายนิ้วมือทุกแบบ มีลักษณะเป็นลายเส้นวนเวียนเป็นรูปก้นหอยหรือเป็นวง มีจุดสันดอนสองแห่งขึ้นไป และจุดศูนย์กลางหนึ่งจุด ดังนั้นจึงมีค่าจำนวนเส้นลายนิ้วมือสองค่า เพื่อความสะดวกในการจำแนกประเภทลายนิ้วมือ ดังนั้นลายนิ้วมือแบบก้นหอย จึงหมายรวมถึง ลายนิ้วมือที่ไม่จัดอยู่ในแบบโค้งหรือมัดหวาย

รูปที่ 2.26 กลุ่มก้นหอย

2.3.3.1 ก้นหอยธรรมดา (Plain Whorl = W)

รูปที่ 2.27 ก้นหอยธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.2 ก้นหอยกระเป๋ากลางปัดขวา (Right Central Pocket = RCP)

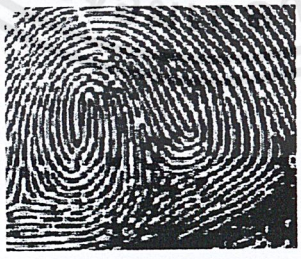
รูปที่ 2.28 ก้นหอยกระเป๋ากลางปัดขวา

2.3.3.3 ก้นหอยกระเป๋ากลางปัดซ้าย (Left Central Pocket = LCP)

รูปที่ 2.29 ก้นหอยกระเป๋ากลางปัดซ้าย

2.3.4 กลุ่มซับซ้อน (Accidental Whorl = AW)

ลายนิ้วมือที่มีลักษณะพิเษษที่ไม่สามารถจัดเข้ากลุ่มใดกลุ่มหนึ่งได้มักประกอบจากลายนิ้วมือ 2 กลุ่มมาผสมกัน



รูปที่ 2.30 กลุ่มซับซ้อน

ลักษณะเฉพาะที่ หรือเรียกจุดสำคัญ (minutia point) ซึ่งมีลักษณะพิเศษบนลายนิ้วมือดังต่อไปนี้

ก. Ridge ending หมายถึง จุดที่เริ่มต้นหรือสิ้นสุดของลายเส้นนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. Ridge bifurcation or fork เป็นลายเส้นจากเส้นเดี่ยวที่แยกออกจากกันเป็นสองเส้นหรือมากกว่า หรือในทางกลับกันอาจเรียกว่าลายเส้นสองเส้นมารวมกันเป็นเส้นเดือด

ค. Ridge divergence เป็นลายเส้นสองเส้นคู่ขนานกันหรือเกือบขนาน

ง. Dot หรือ island เป็นลายเส้นที่สั้นมากจนดูเหมือนเป็นจุดเล็กๆ

จ. Short ridge เป็นลายเส้นที่สั้นแต่ไม่สั้นมากถึงกับเป็นจุดเล็กๆ

ฉ. Enclosure or lake เป็นลายเส้นที่แยกออกเป็นสองเส้น แล้วกลับมารวมกันใหม่ จึงมีพื้นที่ปิดเกิดขึ้น

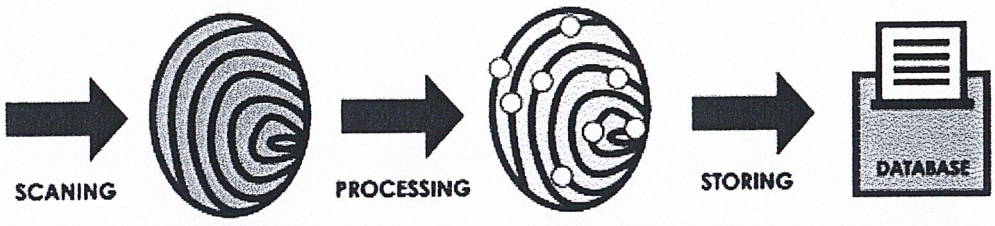
ช. Hook เป็นลายเส้นของเส้นเดียวแต่แยกออกเป็น 2 เส้นโดยที่เส้นหนึ่งสั้นอีกเส้นหนึ่งยาว ดูคล้ายตะขอ

ซ. Miscellaneous เป็นลายเส้นที่มีลักษณะไม่ตรงกับทั้ง 7 แบบที่กล่าวมาแล้ว เช่น เป็นลายเส้นที่แยกจากหนึ่งเส้นเป็นสามเส้นเรียก Trifurcation

2.4 หลักการวิเคราะห์ และการทำงานของการสแกนลายนิ้วมือ

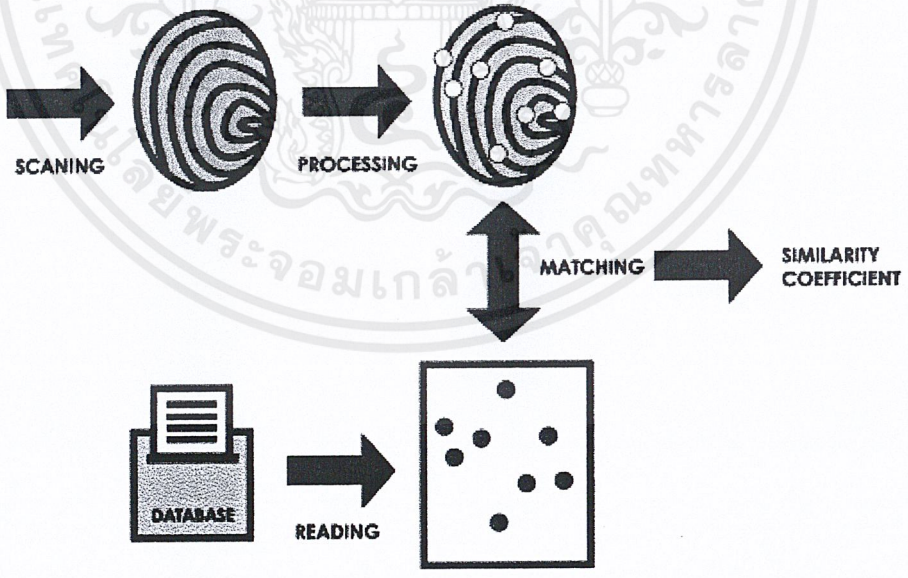
การวิเคราะห์ลายนิ้วมือของบุคคลโดยทั่วไปนั้น จะเริ่มด้วยการนำลายนิ้วมือของแต่ละบุคคลแต่ละนิ้ว มาหาจุดลักษณะเฉพาะที่สำคัญกระบวนการแรกเริ่มของการตรวจพิสูจน์ลายนิ้วมือคือ การอ่านภาพลายนิ้วมือเข้ามาเก็บไว้ในหน่วยความจำถาวรซึ่งในส่วนนี้จะใช้ EEPROM เป็นส่วนที่เก็บข้อมูลไว้ โดยข้อมูลที่อ่านหรือสแกนเข้ามานั้นจะนำมาผ่านการประมวลผล (Processing) ก่อนแล้วจึงเก็บข้อมูลนั้นไว้ ซึ่งข้อมูลนี้จะถูกเก็บไว้เป็นต้นแบบ หรือ รหัสของผู้ใช้แต่ละคนในขั้นตอนก่อนที่จะนำลายนิ้วมือเข้าไปเก็บนั้นจะต้องผ่านขั้นตอนของการประมวลผล (Pre-Processing) ก่อนในกระบวนการนี้จะทำให้ภาพที่ได้รับการสแกนเข้ามาเกิดความสมบูรณ์มากขึ้นเพราะเมื่อเครื่องได้รับการสแกนภาพเข้ามาแล้ว ภาพที่อ่านได้อาจไม่ชัดเจน พร่าเลือน ก็จะทำให้การประมวลผลในขั้นตอนถัดไปทำได้ด้วยความยากลำบากหรือทำไม่ได้ ซึ่งจะทำให้ผลที่ได้ก็อาจไม่ถูกต้องตามที่ควรจะเป็น เมื่อเกิดปัญหาเช่นนี้ในกระบวนการนี้จึงได้มีการกระทำหลายกระบวนการด้วยกันคือการกำจัดสัญญาณรบกวน การปรับความมืดสว่าง ความแตกต่างของตัวภาพและฉากของภาพ การแปลงภาพเป็นภาพสองระดับ (Binary) การทำให้เส้นลายนิ้วมือบาง (Thinning) การปรับภาพหลังจากแปลงภาพเป็นสองระดับ การหาค่า Threshold ของการปรับภาพเป็นภาพสองระดับ และอื่นๆ อีกมาก ซึ่งกระบวนการจะมากหรือน้อยขึ้นอยู่กับว่าตัวอุปกรณ์นั้น มีการอ่านค่าลายนิ้วมือที่ได้ภาพออกมาละเอียดและสมบูรณ์แค่ไหน เมื่อได้ลายนิ้วมือที่ผ่านการประมวลผลแล้ว ก็จะนำข้อมูลหรือภาพนี้ไปจัดเก็บในหน่วยความจำถาวร (EEPROM) ซึ่งสามารถลบข้อมูลใหม่ด้วยไฟฟ้า โดยภาพที่ถูกจัดเก็บไว้นี้จะถูกเก็บไว้เพื่อใช้ในการเปรียบเทียบกับลายนิ้วมือที่ได้รับการสแกนเข้ามาเมื่อนำตัวอุปกรณ์นี้ไปใช้งานรูปที่ 2.31 แสดงกระบวนการทำงานของการเริ่มใช้งานจากรูปที่ 2.31 เริ่มด้วยการสแกนลายนิ้วมือเข้ามาแล้วนำภาพที่ได้ผ่านการประมวลผลซึ่งจะได้ภาพที่มีประสิทธิภาพมากขึ้นแล้วจึงเก็บภาพนั้นไว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 แสดงภาพการทำงานของระบบ

จากรูปที่ 2 .31 เริ่มด้วยการสแกนลายนิ้วมือเข้ามาแล้วนำภาพที่ได้ผ่านการประมวลผลซึ่งจะได้ภาพที่มีประสิทธิภาพมากขึ้นแล้วจึงเก็บภาพนั้นไว้หลังจากเก็บภาพไว้แล้วนั้นก็มาถึงขั้นตอนการนำไปใช้งาน เมื่อตัวอุปกรณ์ได้ถูกบันทึกหรือเก็บลายนิ้วมือของผู้ที่จะนำไปใช้แล้ว ขั้นตอนในการใช้ก็จะคล้ายกับตอนอ่านลายนิ้วมือเข้ามาเก็บไว้ เพียงแต่การอ่านเข้ามาครั้งนี้ข้อมูลที่ได้จะถูกนำไปเก็บไว้ที่หน่วยความจำชั่วคราว (RAM) ซึ่งหลังจากสแกนเข้ามาแล้วประมวลผลแล้วก็จะทำการเก็บข้อมูลไว้ที่ส่วนของหน่วยความจำชั่วคราว ส่วนในขั้นตอนถัดไปก็จะนำข้อมูลที่เก็บอยู่ในส่วนของหน่วยความจำถาวร กับส่วนที่เก็บอยู่ในหน่วยความจำชั่วคราวนั้นมาทำการเปรียบเทียบกัน (Matching)



รูปที่ 2.32 แสดงการทำงานของระบบ

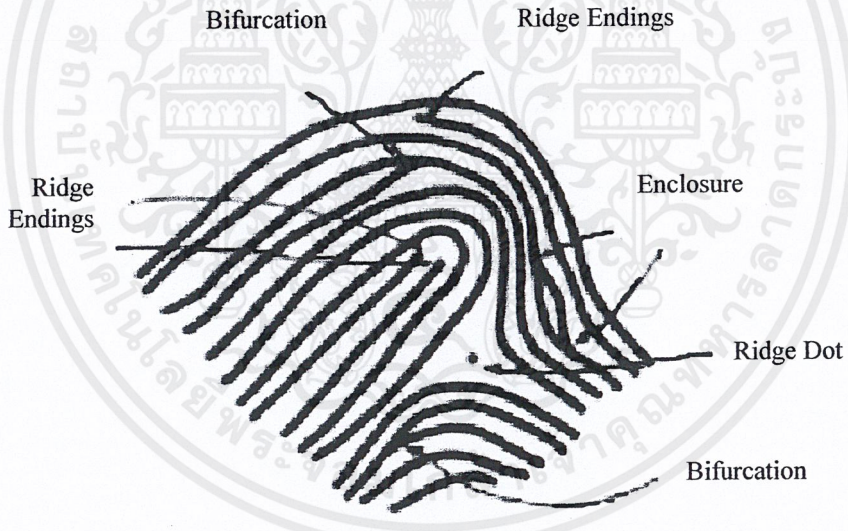
จากรูปที่ 2.32 จะแสดงให้เห็นถึงกระบวนการเปรียบเทียบลายนิ้วมือที่ได้รับการสแกนเข้ามาโดยเริ่มที่ การสแกนภาพเข้ามา แล้วทำการประมวลผลขั้นตอนเดียวกันกับการจัดเก็บตอนแรกแล้วนำภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บไว้ใน ตอนแรกมาเปรียบเทียบกับภาพที่สแกนเข้ามา ณ ตอนนั้นเพื่อเปรียบเทียบว่ามีความเหมือนหรือแตกต่างมา เพียงใดทั้ง 2 ขั้นตอนนั้นต้องผ่านการประมวลผลซึ่งจะทำให้ได้ภาพที่มีประสิทธิภาพในการเปรียบเทียบมาขึ้นซึ่งขั้นตอนนี้มีหลักการและทฤษฎี มากมายที่ต้องทำความเข้าใจและศึกษาเป็นตัวอย่างในการทำการประมวลผลภาพก่อนทำขั้นตอนอื่น ซึ่งผลที่ได้จากการทำจะทำให้ได้จุดลักษณะเฉพาะซึ่งจุดเหล่านี้เองจะเป็นสิ่งในการเปรียบเทียบลายนิ้วมือของแต่ละคนหรือกล่าวได้ว่าเป็นตัวบ่งชี้ความแตกต่างของนิ้วแต่ลการจดจำลายนิ้วมือ (Fingerprint Recognition)

2.4.1 การตรวจสอบลายนิ้วมือ

สำหรับการตรวจดูลายนิ้วมือ 2 รายการ ว่าเป็นลายเดียวกันหรือไม่นั้น มีหลากหลายวิธีด้วยกัน แต่วิธีที่เป็นที่นิยมกันมากที่สุด ได้แก่การเปรียบเทียบจุดสังเกตเล็กๆจุดที่ว่านี้คือจุดที่เส้นลายนิ้วมือ มาบรรจบกัน หรือ แยกออกจากกัน หรือ เป็นจุดจบของเส้น และจุดที่ว่านี้ของแต่ละคนจะไม่เหมือนกัน (FBI ได้สรุปแล้วว่า คนแต่ละคนจะมีไม่มีทางมีจุดสังเกตที่ว่านี้เหมือนกัน เกินกว่า 8 จุด) รูปข้างล่าง แสดงตัวอย่างของจุดที่ว่านี้



รูปที่ 2.33 แสดงจุดที่เส้นลายนิ้วมือ

รายการของจุดสังเกต ได้แก่

- Bifurcation เป็นจุดที่แยกเส้นออกเป็นหลายเส้น
- Divergence เป็นจุดที่เส้น 2 เส้นที่อยู่ขนานกัน แยกออกจากกัน หรือ เข้ามารวมกัน
- Enclosure เป็นลักษณะที่เส้นแยกออกจากกัน และ กลับมารวมกันเป็นเส้นเดียวกันใหม่
- Endings คือตำแหน่งที่เส้นนั้นสิ้นสุดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการบันทึกลายนิ้วมือ ก็จะมีการบันทึกตำแหน่งของจุดสังเกตเหล่านี้ไว้ ซึ่งปกติแล้วจะบันทึกไว้ประมาณ 30-40 จุดต่อ 1 ลายนิ้วมือ และนำข้อมูลมาเปรียบเทียบกับลายนิ้วมือที่สแกนเข้ามาว่าตรงกันหรือไม่

การเก็บข้อมูลลายนิ้วมือลงฐานข้อมูลขั้นแรกจะเก็บทำการเก็บตัวอย่างลายนิ้วมือเพื่อบันทึกลงฐานข้อมูล โดยขั้นตอนการเก็บลงฐานข้อมูล ดังนี้

1.เก็บภาพลายนิ้วมือ โดยการเก็บภาพลายนิ้วมือจะทำการสแกนนิ้วแต่ละคนมากกว่า 1 ครั้ง เนื่องจาก การเก็บภาพแต่ละครั้งจะมี การคลาดเคลื่อนจากตำแหน่งเดิมและการลงน้ำหนักไม่เท่ากัน

2.ทำการค้นหาจุดลายละเอียดบนเส้นลายนิ้วมือ โดยการค้นจุดสังเกตเล็กๆของแต่ละภาพที่เป็นนิ้วเดียวกัน ทำการแปลงข้อมูลลายนิ้วเป็นการสร้างรหัสแทน

3.บันทึกรหัสข้อมูลลายนิ้วมือลงในฐานข้อมูล โดยมีการเก็บแบบข้อมูลการบันทึกแบบ 1: N คือ หนึ่งคนสามารถบันทึกลายนิ้วมือได้มากกว่า 1 ลายนิ้วมือ การตรวจสอบแบบ 1:1 คือผลการตรวจสอบลายนิ้วมือนั้นไม่ว่าจะใช้นิ้วใดก็ตาม (ที่บันทึกไว้แล้ว) ผลออกมาจะเป็นหนึ่งคนนั้นเท่านั้นวิธีค้นหาลายนิ้วมือโดยการทำการแปลงข้อมูลลายนิ้วเป็นรหัส และนำค่ารหัสที่ได้ไปเปรียบเทียบกับค่ารหัสลายนิ้วมือที่เก็บไว้ในฐานข้อมูล เมื่อค่ารหัสตรงหรือคลาดเลื่อนต่างกันน้อยกว่าที่กำหนด ก็จะแสดงบุคคลที่เป็นเจ้าของลายนิ้วมือนั้น

2.4.2 ระบบข้อมูลและการจัดการข้อมูล

ข้อมูลนำเข้า และการจัดการข้อมูลลายนิ้วมือที่ต้องการตรวจสอบโดยการใช้ fingerprint ในการรับภาพลายนิ้วมือ โดยลายนิ้วมือที่ได้นำมานั้นอาจจะมีปัญหาในการประมวลผลเพื่อค้นหาข้อมูลบุคคล เช่น ความสกปรกของลายนิ้วมือ ความแห้งหรือความเปียกบนผิวหนัง และความแปรปรวนของความสว่างบนภาพ เป็นต้น ดังนั้นจึงต้องมีการใช้เทคนิค Gabor Filter ร่วมกับ Voting Algorithms เพื่อใช้ในการขจัดปัญหาดังกล่าว และหลังจากที่ได้ภาพลายนิ้วมือที่มีความชัดเจน ก็นำไปเข้ากระบวนการ Support Vector Machine ซึ่งเป็นกระบวนการรู้จำ แล้วทำการเปรียบเทียบกับแม่แบบที่เก็บไว้ในฐานข้อมูลเพื่อตรวจสอบความเหมือนกันระหว่างภาพลายนิ้วมือตัวอย่างกับแม่แบบในฐานข้อมูล

2.5 เครื่องสแกนลายนิ้วมือ รุ่น U.are.U 4000

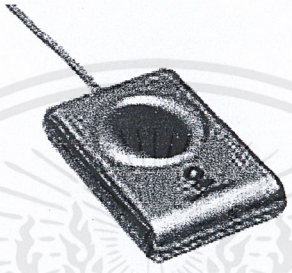
เครื่องสแกนลายนิ้วมือรุ่น U.are.U 4000เป็นเครื่องสแกนลายนิ้วมือแบบสารกึ่งตัวนำซึ่งใช้หลักการทางชีววิทยา (Biometrics) เพื่อทำการวิเคราะห์ลายนิ้วมือ จึงทำให้ยากต่อการลอกเลียนแบบ รุ่น U.are.U 4000 เป็นอุปกรณ์จับภาพลายนิ้วมือที่มีการติดต่อผ่านพอร์ต USB เครื่องสแกนลายนิ้วมือจะมีการจับภาพลายนิ้วมือได้ถูกต้องแม่นยำโดยอัตโนมัติ และควบคุมการตรวจสอบลักษณะของลายนิ้วมือซึ่งภาพลายนิ้วมือที่ได้จะเป็นการเข้ารหัสภายในตัว U.are.U 4000 ก่อนที่จะส่งไปยังฐานข้อมูล History กระบวนการเพิ่มความปลอดภัยในตัวระบบของตัว U.are.U 4000 ในขณะสแกนลายนิ้วมือจะไม่มีการปล่อยกระแสไฟฟ้าออกจากหน้าจอสแกนทำให้ไม่เป็นอันตรายต่อร่างกาย ในการสแกนจะใช้งานได้ดีกับรายนิ้วมือที่ ชื้น แห้ง หรือ หยาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- U.are.U 4000 ทำงานด้วยการเชื่อมต่อกับ USB พอร์ต ผู้ใช้ใช้งานได้ง่ายเพียงวางนิ้วที่จอแสดงเซ็นเซอร์ไฟสีแดง และเครื่องจะจับภาพของลายนิ้วมือด้วยความเร็วสูง

- U.are.U 4000 ติดตั้งใช้งานสะดวก และให้ความมั่นใจในการบันทึกลายนิ้วมือจากหัวอ่าน U.are.U 4000 สู่โปรแกรม

- Time Attendance software โดยสามารถที่จะจับคู่ลายนิ้วมือที่ทำการบันทึกแล้วกับการสแกนลายนิ้วมือได้เร็ว และแม่นยำ เหมาะสำหรับการใช้งานที่ต้องการความแม่นยำในการบันทึก



รูปที่ 2.34 เครื่องสแกนลายนิ้วมือรุ่น U.are.U 4000

2.5.1 คุณสมบัติ U.are.U 4000

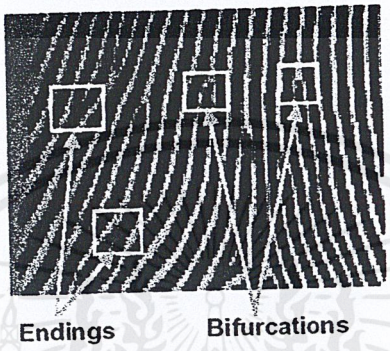
- มีขนาดเล็ก
- คุณภาพในการบันทึกภาพสูง
- ปฏิเสธลายนิ้วมือแฝง
- สามารถอ่านลายนิ้วมือได้ที่ค่าผิดพลาดถึง 30 องศา
- ใช้งานได้ดีกับรายนิ้วมือที่ ชื้น แห้ง หรือ หยาบ
- ใช้งานได้กับแอปปลิเคชั่น U.are.Uทั้งหมด
- สนับสนุนระบบปฏิบัติการ Windows 98, Me, NT 4.0,2000, XP
- สามารถนำไปพัฒนาร่วมกับชุดพัฒนาโปรแกรมSDK (Software Develop Kit) เพื่อทำเป็นโปรแกรมต่างๆ ได้ตามความต้องการ

2.5.2 คุณสมบัติพิเศษของเครื่องสแกนลายนิ้วมือ รุ่น U.are.U 4000

- สามารถเก็บลายนิ้วมือได้ทั้ง 10 นิ้ว
- บันทึกการแล้วสามารถส่งข้อมูลออกเป็นไฟล์ข้อมูล (Text Files) ไปใช้งานกับการคำนวณได้ทุกชนิด
- ใช้งานง่าย เพียงติดตั้งโปรแกรมแล้วทำการเก็บลายนิ้วมือต้นฉบับ ก็สามารถใช้งานได้ทันที
- โปรแกรม Time Attendance สามารถคำนวณเวลาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

U.are.U 4000 ใช้เทคนิคไฮบริดจ์ (hybrid) ในการเปรียบเทียบลายนิ้วมือ โดยจะทำการตรวจสอบลักษณะเด่นของลายนิ้วมือรอบๆ จุดใจกลางออกไป ลักษณะเด่นดังกล่าวนี้มีอยู่หลายรูปแบบที่พบเห็นส่วนมากได้แก่ Ending Bifurcation Island Lake และDot สำหรับลายนิ้วมือที่เราให้ความสำคัญนั้นมีอยู่ 2 ลายเท่านั้น คือ Bifurcation กับ Ending เนื่องจากการเกิดตำแหน่งลายนิ้วมือที่สำคัญของแต่ละบุคคลนั้นจะไม่เกิดที่ตำแหน่งเดียวกันจึงได้ทำการหาตำแหน่งเดียวกับจึงได้ทำการหาตำแหน่งที่สำคัญของลายนิ้วมือนี้แล้วนำไปเก็บไว้เป็นตำแหน่งอ้างอิง



รูปที่ 2.35 ตำแหน่งที่สำคัญของลายนิ้วมือที่สำคัญ

รายละเอียด

- ความสามารถของหัวอ่านในการแยกจุดถึง: 512 dpi
- ขนาดภาพ: 14.6 mm
- 18.1 mm (ความยาว)
- 8-bit สีขาว/ดำ (256ระดับสี)
- Sensor ขนาด: approx. 79 x มม. 49xมม. 19 มม

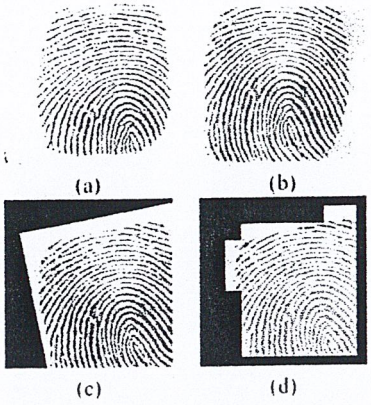
2.5.3 เทคนิคการเปรียบเทียบแบไฮบริดจ์ (The hybrid matching approach)

เทคนิคการเปรียบเทียบไฮบริดจ์เป็นการรวมเอารายละเอียดปลีกย่อยของลายนิ้วมือมาทำการเปรียบเทียบโดยมีหลักการดังนี้

2.5.4 การจัดภาพ (Image Alignment)

จากภาพที่จะทำการอ้างอิงและภาพที่รับมาเพื่อทำการเปรียบเทียบ โดยจะหารายละเอียดปลีกย่อยของลายนิ้วมือทั้งสองเพื่อใช้ในการอธิบายอัลกอลิทึม โดยในรูป 2.36 (a) เป็นรูปลายนิ้วมือที่ใช้อ้างอิง (b) เป็นลายนิ้วมือที่รับมาได้ ซึ่งมีรายละเอียดใกล้เคียงกันของแต่ละจุดปลีกย่อย การจัดภาพขั้นตอนแรกต้องหารายละเอียดปลีกย่อย การจัดภาพขั้นตอนแรกต้องหารายละเอียดปลีกย่อยของจุดที่ตรงกัน ซึ่งในรูป (c) เป็นภาพที่ทำการหมุนแล้วรายละเอียดที่ได้ใกล้เคียงกับภาพ (a) และรูป (d) เป็นการตัดบางส่วนของภาพออกเพื่อให้องค์ประกอบของภาพใกล้เคียงกับภาพอ้างอิงมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 การจัดตำแหน่งของภาพอินพุท

(a) เป็นรูปลายนิ้วมือที่ใช้อ้างอิง

(b) เป็นลายนิ้วมือที่รับมา

(c) เป็นภาพที่ทำการหมุน

(d) เป็นการตัดบางส่วนของภาพ

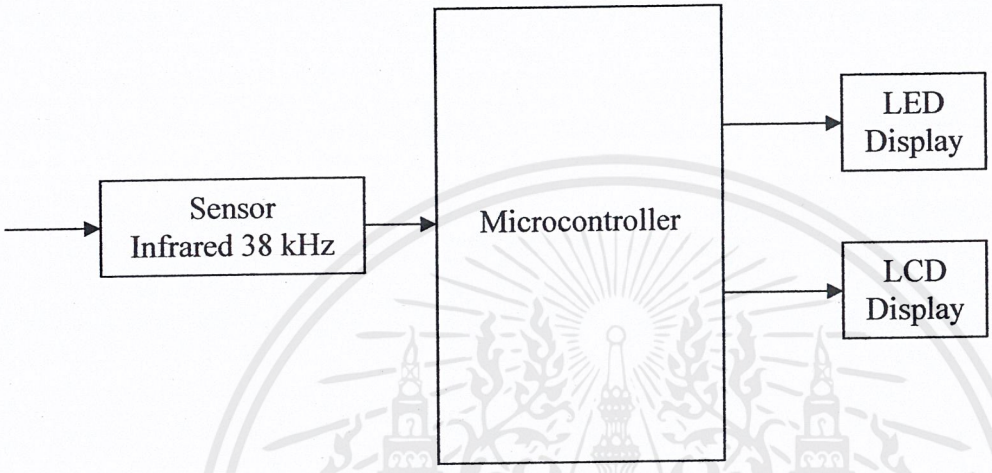
รูปที่ 2.37 (a) การจัดตำแหน่งลายนิ้วมือทั้งสองที่เหมือนกัน

(b) การจัดตำแหน่งลายนิ้วมือทั้งสองที่ไม่เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

โครงงานนี้เป็นการนำไมโครคอนโทรลเลอร์มาประยุกต์ใช้การทำงานของระบบที่จอดรถอัตโนมัติ โดยที่มีขบวนการดังบล็อกไดอะแกรมข้างล่างนี้



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของโครงงาน

3.1 โครงสร้างทางฮาร์ดแวร์ (Hardware)

3.1.1 ส่วนของวงจรเซนเซอร์

สร้างวงจรอินฟาเรดด้านส่ง จะใช้พัลส์ที่สร้างโดย ไอซี 555 ซึ่งความถี่ที่ต้องการสร้างกำหนดได้จากค่า R_1 , R_2 และC_1 ซึ่งคำนวณได้จากสมการ

$$f = \frac{1}{T} = \frac{1.44}{(R_1 + 2R_2)C_1} \qquad (1)$$

ช่วงการเก็บประจุของ C_1 : t1 = $0.693*(R_1+R_2)*C_1$

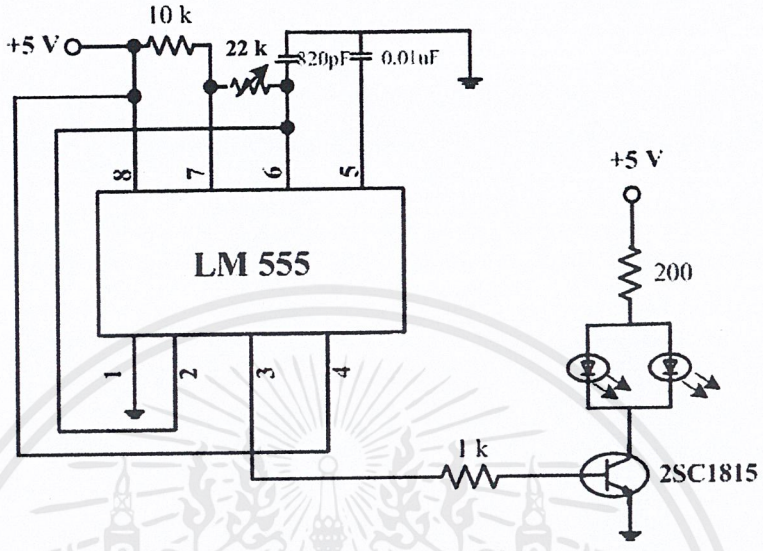
ช่วงการคายประจุของ C_1 : t2 = $0.693*R_1*C_1$

และคำนวณหาคาบเวลาจากสมการ

$$T = t_1 + t_2 = 0.693(R_1 + R_2)C_1 \qquad (2)$$

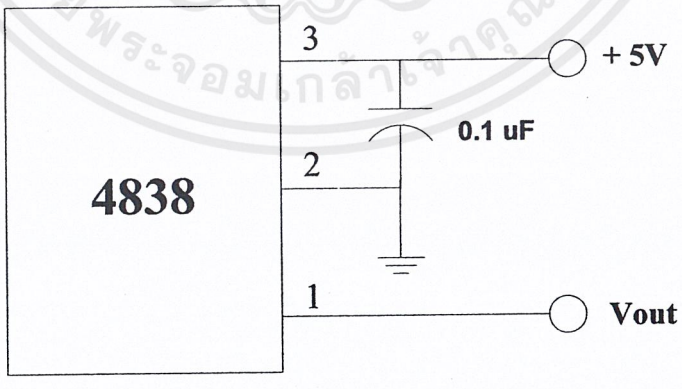
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในวงจรที่ใช้ต้องการสร้างพัลส์ความถี่ที่ 38 kHz เพื่อให้ตรงกับความถี่ของวงจรทางด้านรับสัญญาณอินฟราเรด โดยสัญญาณพัลส์ที่สร้างได้จะถูกส่งไปยังวงจรไดร์ฟซึ่งจะใช้เพาเวอร์ทรานซิสเตอร์เบอร์ 2SC1815N



รูปที่ 3.2 แสดงวงจรอินฟราเรดทางด้านส่ง

ส่วนวงจรทางด้านรับสัญญาณอินฟราเรดจะใช้โฟโต้ไดโอด (Photo Diode) เบอร์ 4838 ซึ่งสามารถรับสัญญาณอินฟราเรดที่ความถี่ 38 kHz ซึ่งภายในโฟโต้ไดโอดเบอร์ 4838 จะประกอบด้วยส่วนรับสัญญาณอินฟราเรด และภาคขยายสัญญาณเอาท์พุท



รูปที่ 3.3 แสดงวงจรอินฟราเรดทางด้านรับ

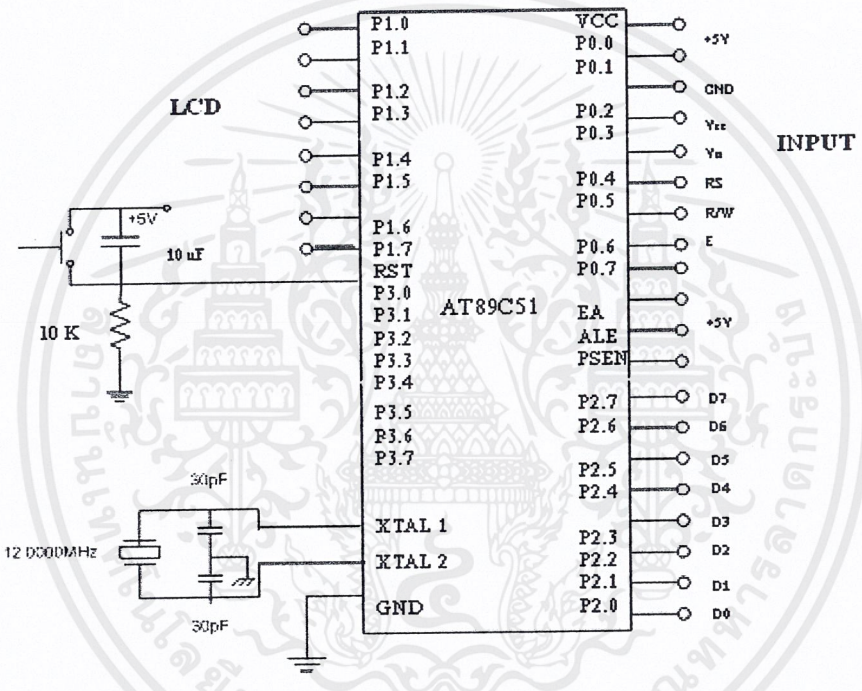
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ส่วนของไมโครคอนโทรลเลอร์

AT89C51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต และเป็นไอซีประเภท CMOS ทำให้มีประสิทธิภาพสูง กินไฟต่ำ มีหน่วยความจำแฟลช บรรจุภายในตัวขนาด 8 กิโลไบต์ ทั้งยังทำงานร่วมและทดแทนไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ได้ทั้งชุดคำสั่งและการจัดเรียงขา

การที่มีหน่วยความจำแบบแฟลช ทำให้เครื่องคอมพิวเตอร์สามารถทำการลบและเขียนโปรแกรมได้โดยตรง โดยไม่ต้องถอด ไมโครคอนโทรลเลอร์ออกจากแผงวงจร ในลักษณะที่เรียกว่า อิน ซิสเต็ม โปรแกรมมิ่ง (In System Programming) ซึ่งเหมาะต่อการพัฒนาโปรแกรม

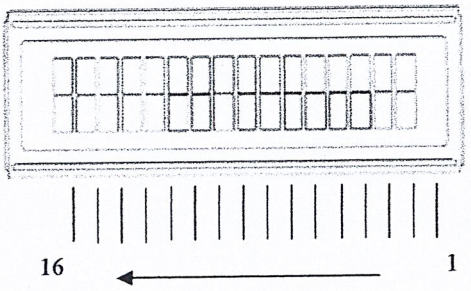
ในโครงงานนี้ทำการต่อวงจรไมโครคอนโทรลเลอร์ เพื่อควบคุมการทำงานของวงจรต่างๆดังรูป



รูปที่ 3.4 แสดงการต่อวงจรไมโครคอนโทรลเลอร์เพื่อควบคุมวงจรต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ส่วนของจอแสดงผลแอลซีดี



รูปที่ 3.5 แสดงขาแอลซีดี

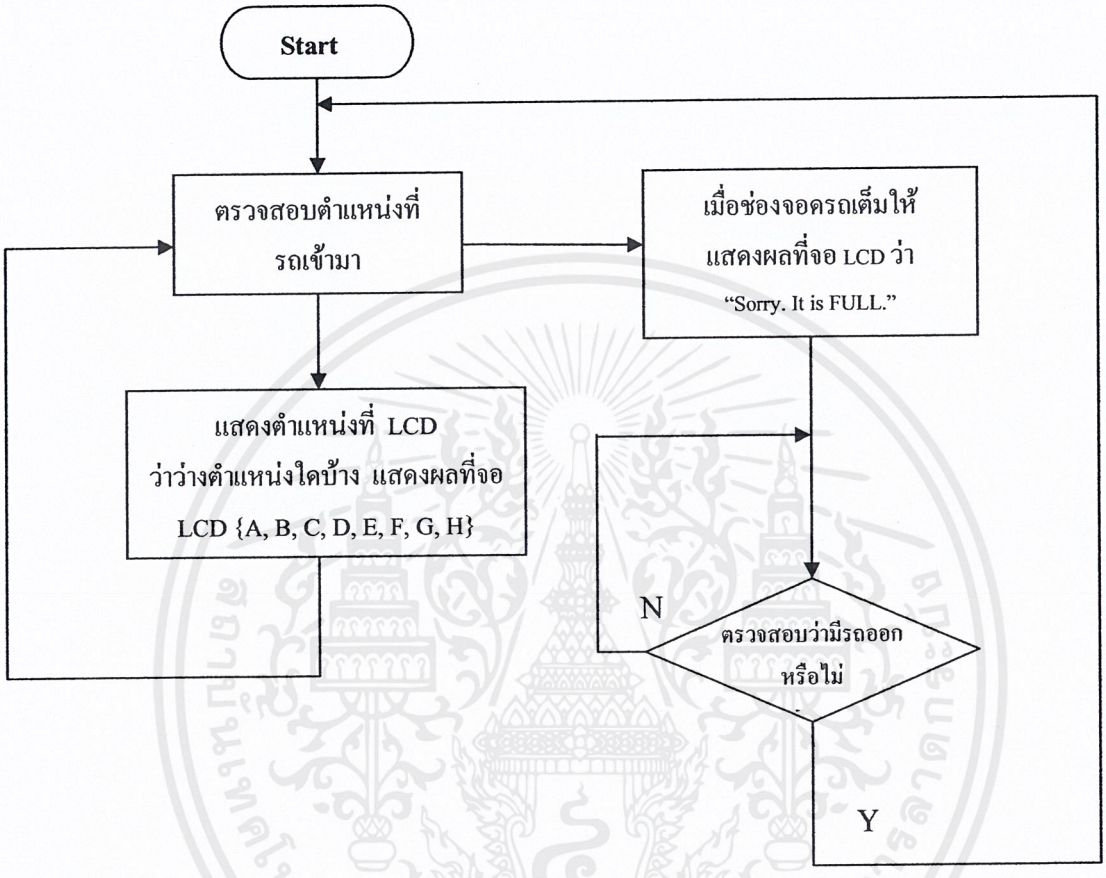
ตารางที่ 3.1 แสดงขาการทำงานของแอลซีดี

ขา	สัญญาลักษณ์	การทำงาน
1	Vss	0 V
2	V_{DD}	5 V
3	V_0	ต่อตัวต้านทานที่ปรับค่าได้เพื่อปรับความเข้มหน้าจอแสดงผล
4	RS	ตัวควบคุมการทำงาน
5	R/W	ตัวควบคุมการทำงาน
6	E	ตัวควบคุมการทำงาน
7	DB0	อินพุท
8	DB1	อินพุท
9	DB2	อินพุท
10	DB3	อินพุท
11	DB4	อินพุท
12	DB5	อินพุท
13	DB6	อินพุท
14	DB7	อินพุท
15	A	Black light
16	K	Black light

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โครงสร้างทางซอฟท์แวร์

3.2.1 โฟลว์ชาร์ตแสดงการทำงานของโปรแกรมทางด้านรับ-ด้านส่ง

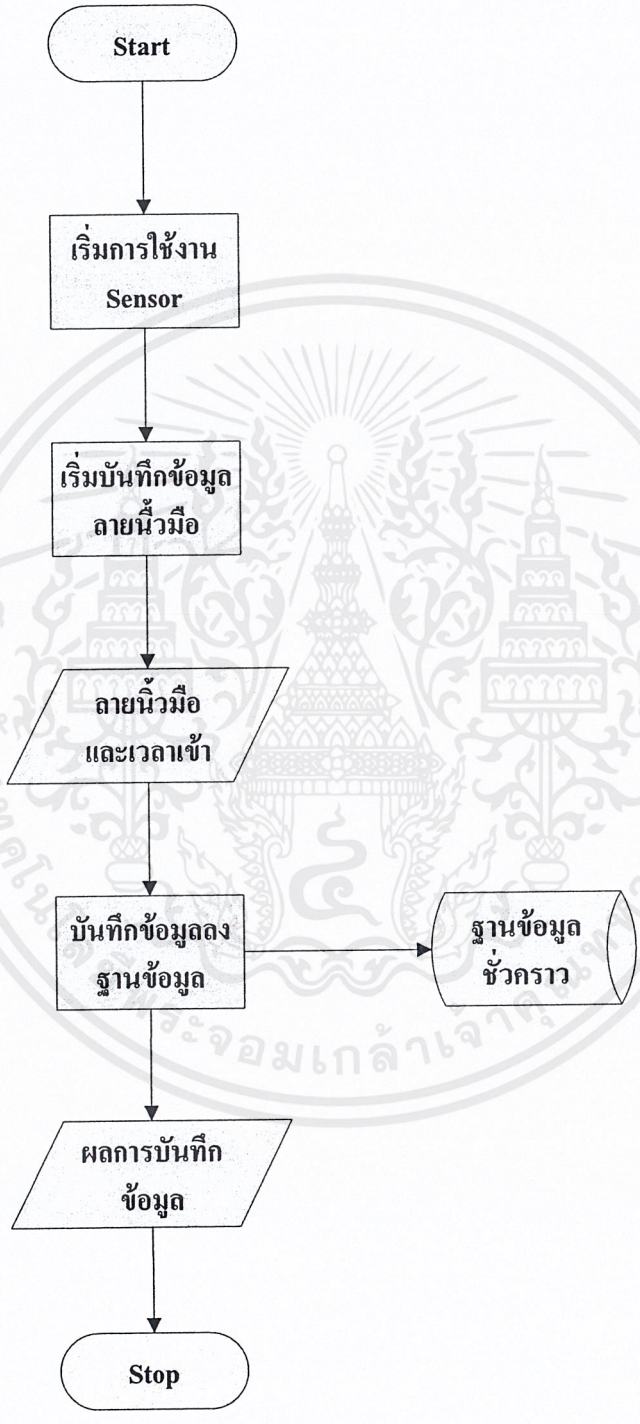


รูปที่ 3.6 แสดงโฟลว์ชาร์ตโปรแกรมหลักการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โฟลว์ชาร์ตควบคุมการทำงานของเครื่องสแกนลายนิ้วมือ

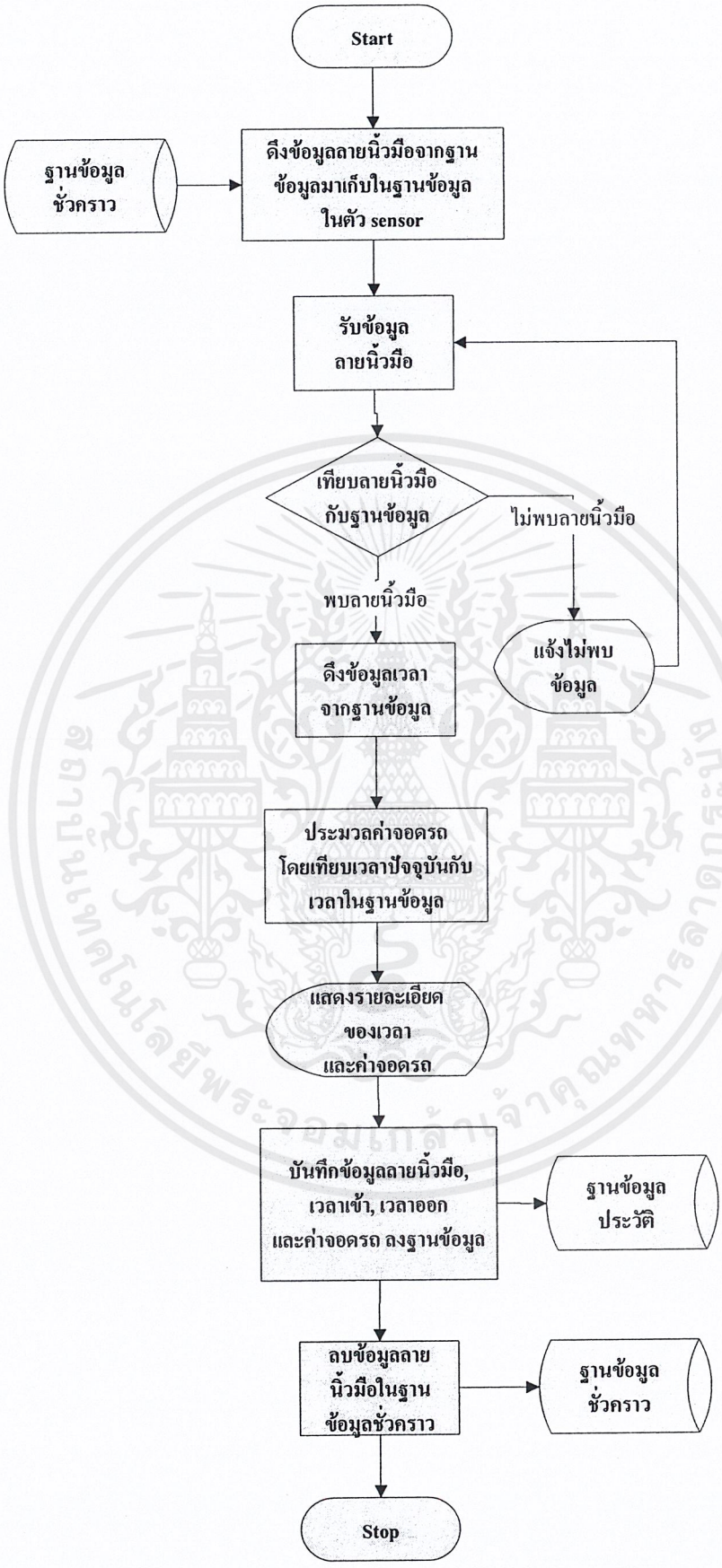
- ทางเข้า



รูปที่ 3.7 แสดงโฟลว์ชาร์ตโปรแกรมหลักการทำงานของเครื่องสแกนลายนิ้วมือ ส่วนทางเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทางออก



รูปที่ 3.8 แสดงโฟลว์ชาร์ตโปรแกรมหลักการทำงานของเครื่องสแกนลายนิ้วมือ ส่วนทางออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

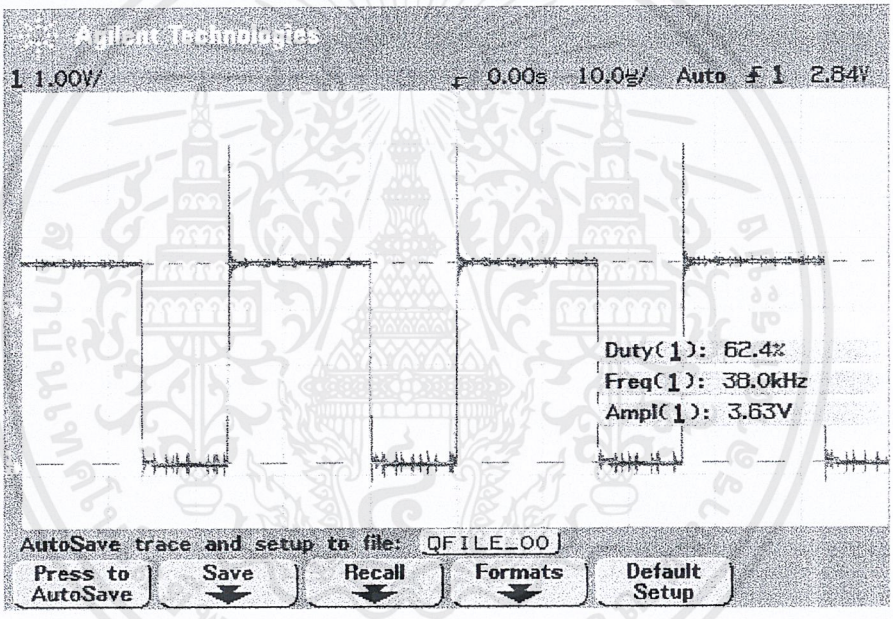
4.1 ขั้นตอนการทดลองส่วนที่ 1

ในโครงงานนี้แบ่งเป็น 2 ส่วน คือส่วนของฮาร์ดแวร์และซอฟท์แวร์ โดยได้ทำการทดลองในส่วนของฮาร์ดแวร์ ตามวงจรในบทที่ 3 ในส่วนของซอฟท์แวร์นั้น จะได้ทำการทดลองเขียนโปรแกรมตามโฟว์ชาร์ทที่แสดงแล้วในบทที่ 3

4.1.1 ผลการทดลอง

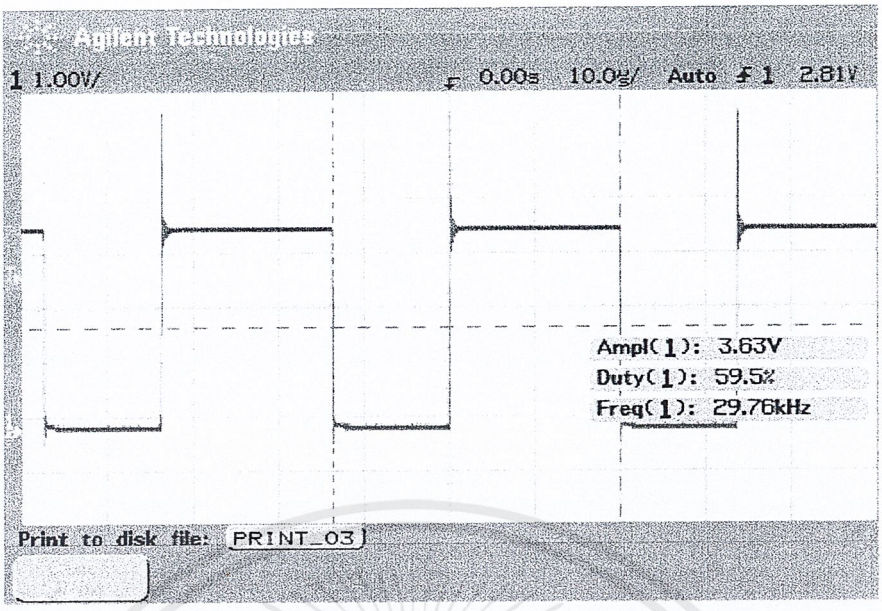
4.1.1.1 ผลการทดลองของวงจรเซนเซอร์ทางด้านส่งและด้านรับ

<u>ผลจากด้านส่ง</u> โดยวัดสัญญาณทางด้านส่งที่ขา 3 ของไอซี 555

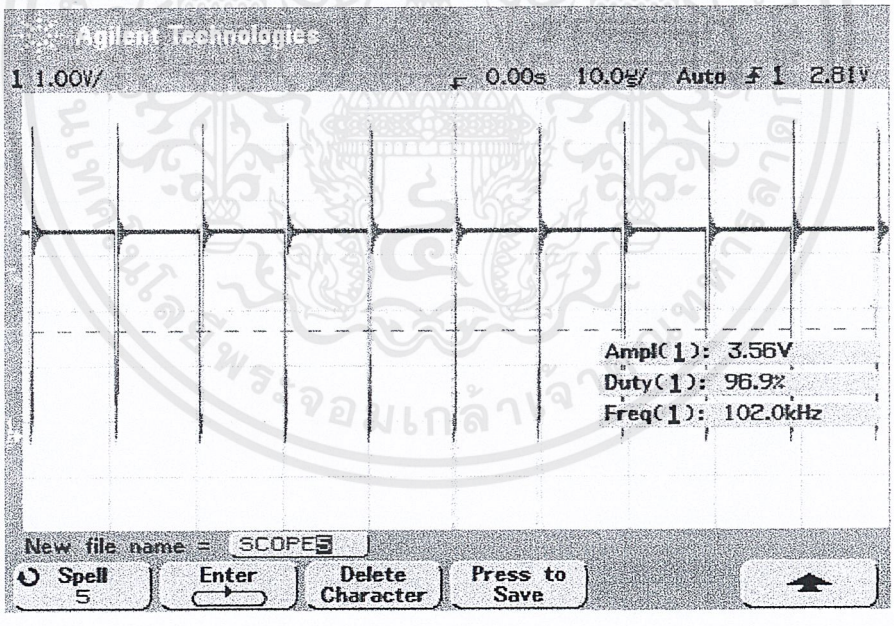


รูปที่ 4.1 แสดงสัญญาณด้านส่งเมื่อปรับค่าความต้านทานจนได้ความถี่ที่ค่า 38 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



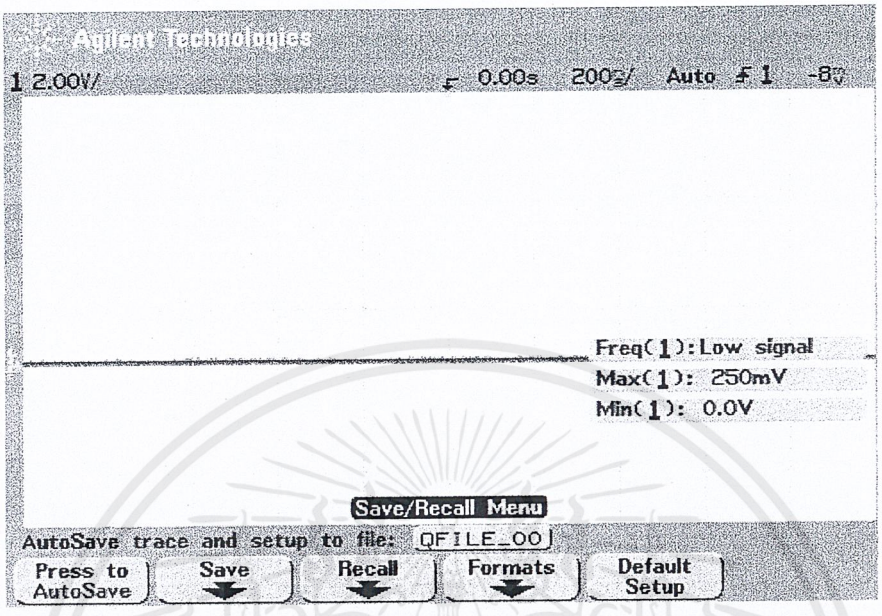
รูปที่ 4.2 แสดงสัญญาณด้านส่งเมื่อปรับค่าตัวความต้านทานปรับค่าได้ 22 k โอมห์ ไปที่ค่ามากสุด



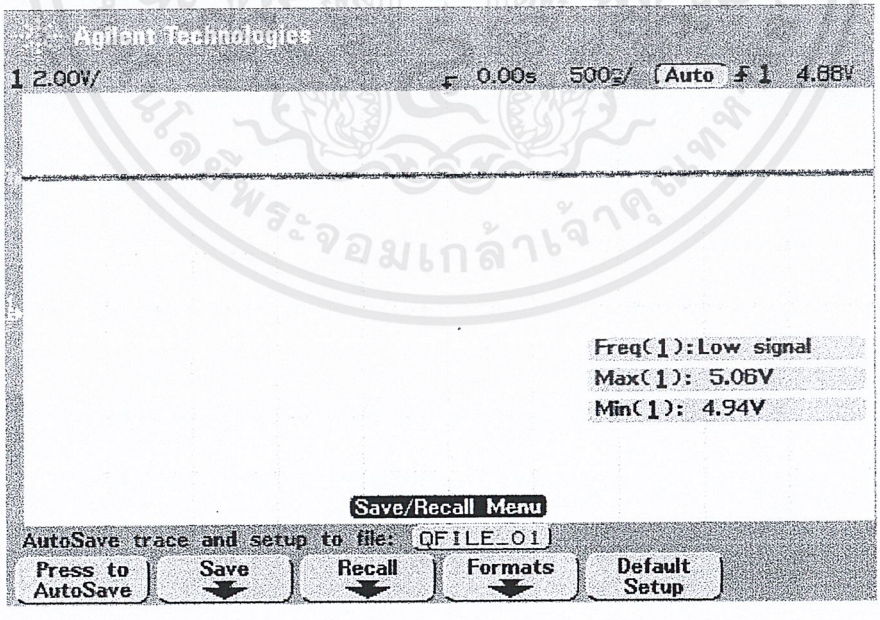
รูปที่ 4.3 แสดงสัญญาณด้านส่งเมื่อปรับค่าตัวความต้านทานปรับค่าได้ 22 k โอมห์ ไปที่ค่าน้อยสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากด้านรับ โดยจับที่ขาที่ 1 ของตัว 4838



รูปที่ 4.4 รูปจากสัญญาณรับที่รับได้จากอินฟราเรด ช่วงที่ตรวจจับไม่พบ



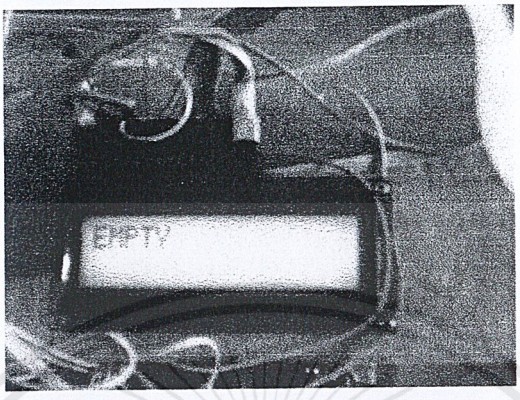
รูปที่ 4.5 รูปจากสัญญาณรับที่รับได้จากอินฟราเรด ช่วงที่ตรวจจับพบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

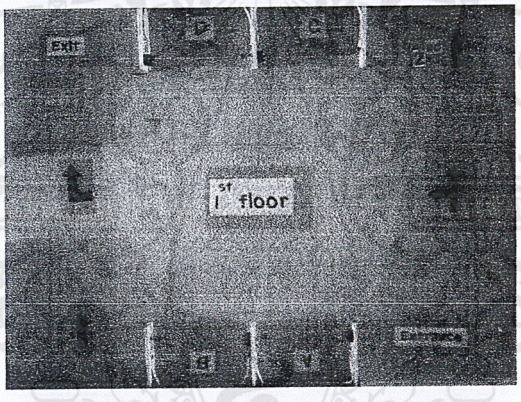
4.1.1.2 ผลการทดลองในส่วนของซอฟท์แวร์

ซึ่งจะเป็นผลการทดลองที่แสดงผลทางแอลซีดี

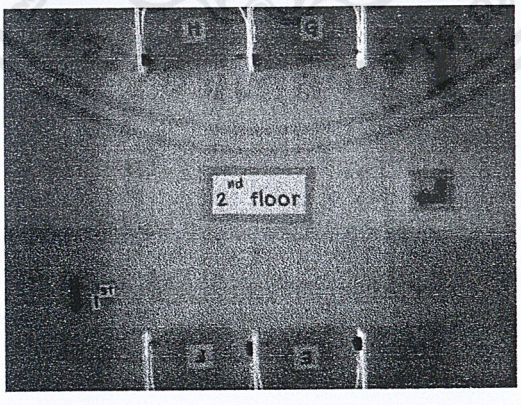
- ในกรณีที่ที่จอดรถว่างแอลซีดีแสดงดังรูป



(ก)



(ข)



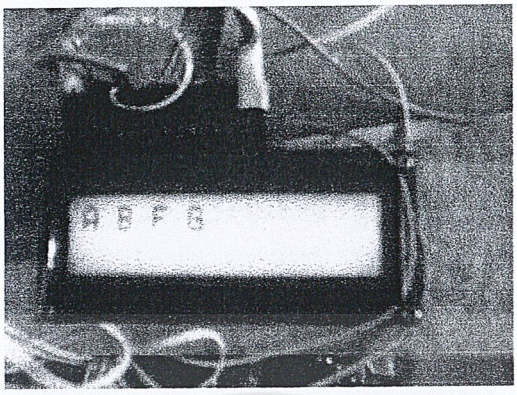
(ค)

รูปที่ 4.6 (ก) การแสดงผลของหน้าจอแอลซีดีขณะยังไม่มีรถจอด

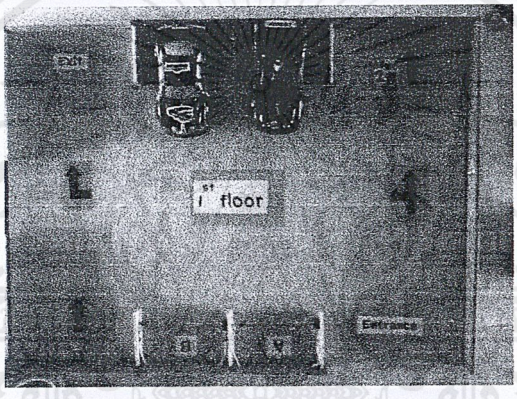
(ข) และ (ค) ลานจอดขณะยังไม่มีรถจอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

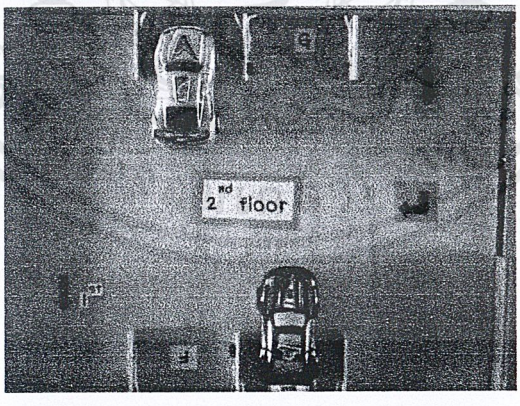
- ในกรณีที่มีรถจอดที่ช่องแอลซีดีจะแสดงผลที่จอดรถที่ว่างอยู่ออกทางหน้าจอ ดังรูป



(ก)



(ข)

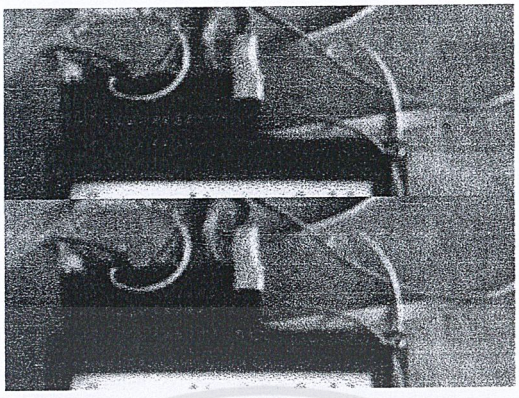


(ค)

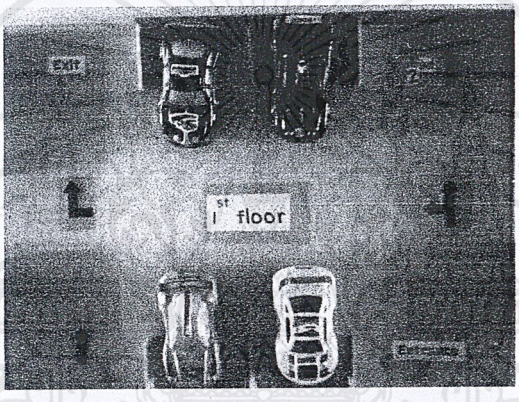
รูปที่ 4.7 (ก) การแสดงผลของหน้าจอแอลซีดีขณะมีรถจอดบางส่วน
(ข) และ (ค) ลานจอดขณะยังไม่มีรถจอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

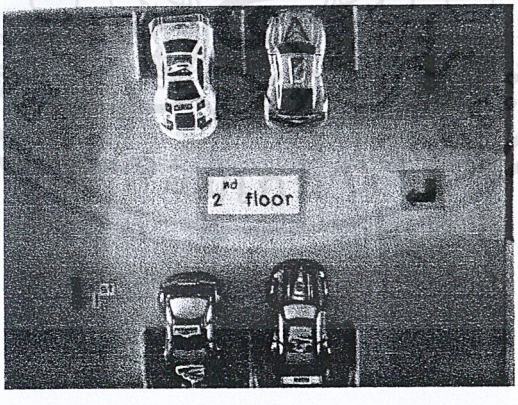
- ในกรณีที่มีรถจอดครบทุกช่องแอลซีดีจะแสดงดังรูป



(ก)



(ข)



(ค)

รูปที่ 4.8 (ก) การแสดงผลของหน้าจอแอลซีดีขณะมีรถจอดครบทุกที่จอด
(ข) และ (ค) ลานจอดขณะมีรถจอดครบทุกที่จอด

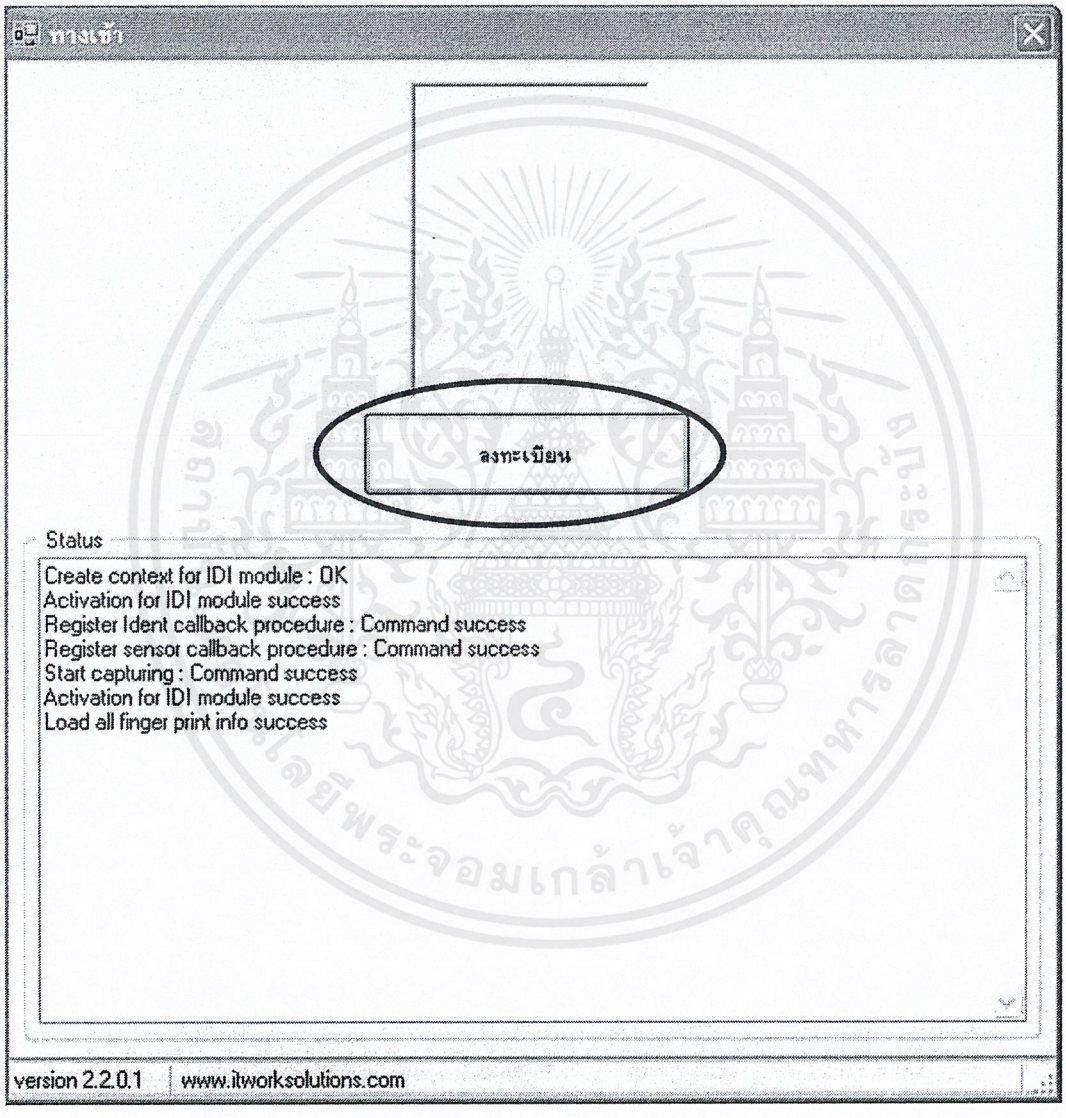
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ขั้นตอนการทดลองส่วนที่ 2

ในโครงงานของการทดลองส่วนที่ 2 นี้เป็นการนำเครื่องสแกนลายนิ้วมือมาประยุกต์กับส่วนที่ 1 เพื่อทำการสร้างระบบการจอดรถ โดยส่วนนี้จะทำการเขียนโปรแกรมไปใช้ควบคุมการทำงานของเครื่องสแกนลายนิ้วมือ

4.2.1 ผลการทดลอง

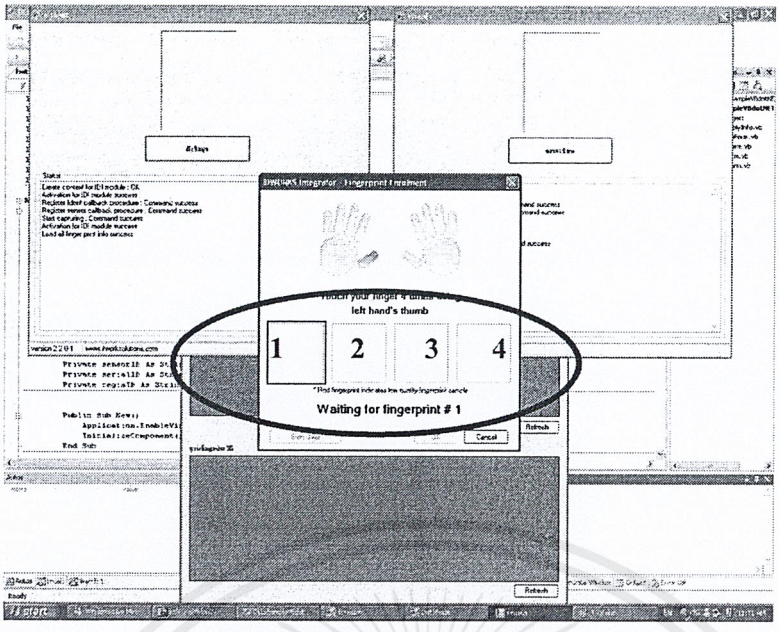
- การเริ่มต้นเปิดโปรแกรมเพื่อทำการลงทะเบียนการเข้าจอดรถ



รูปที่ 4.9 แสดงฟอร์มการลงทะเบียนเข้าจอดรถ

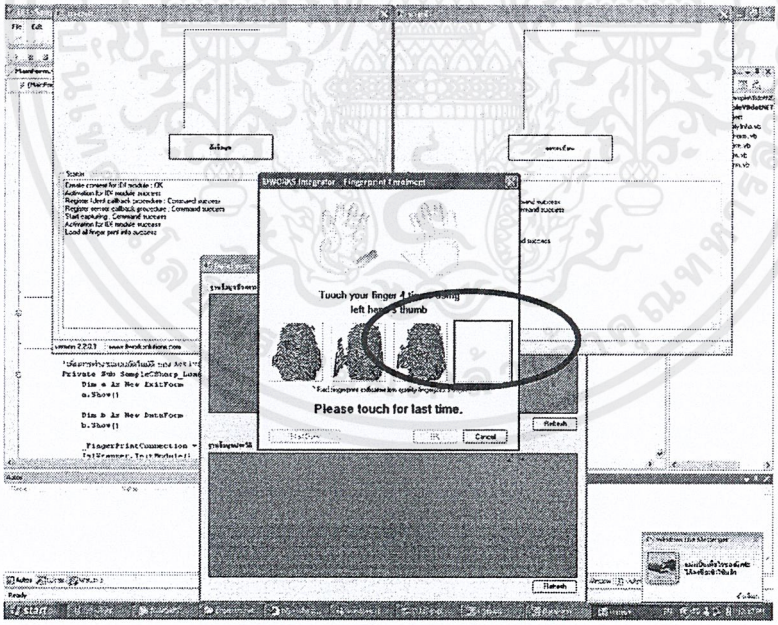
- เมื่อแสดงดังรูปข้างต้น เราจะทำการเริ่มต้นลงทะเบียนโดยคลิกที่ “ลงทะเบียน” และจะแสดงได้ดังรูป4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงช่องลงทะเบียนด้วยการสแกนลายนิ้วมือ

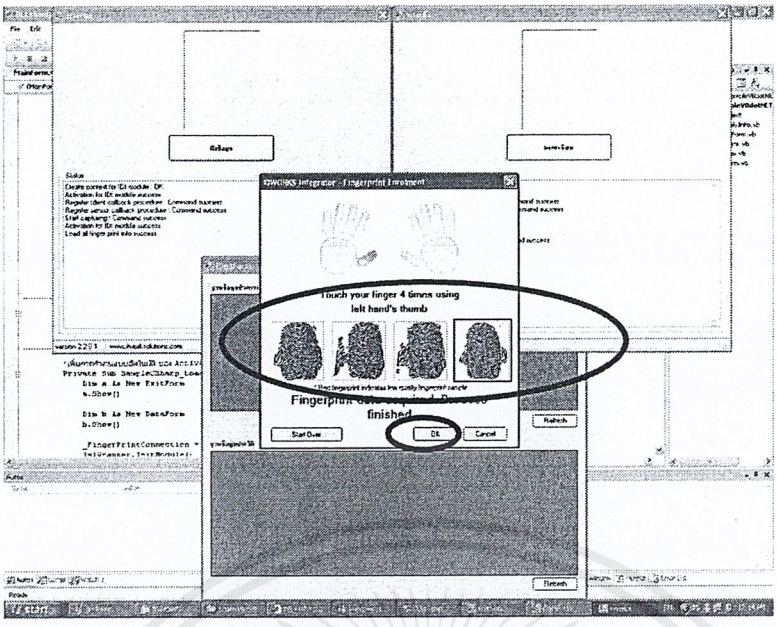
- จากรูป 4.10 เราจะทำการลงทะเบียน ด้วยการสแกนลายนิ้วมือ 4 ครั้ง และเมื่อเราสแกน มาแล้วนำภาพที่ได้ผ่านการประมวลผลและแสดงได้ดังรูปที่ 4.11



รูปที่ 4.11 ลงทะเบียนด้วยการสแกนลายนิ้วมือ 3 ครั้ง

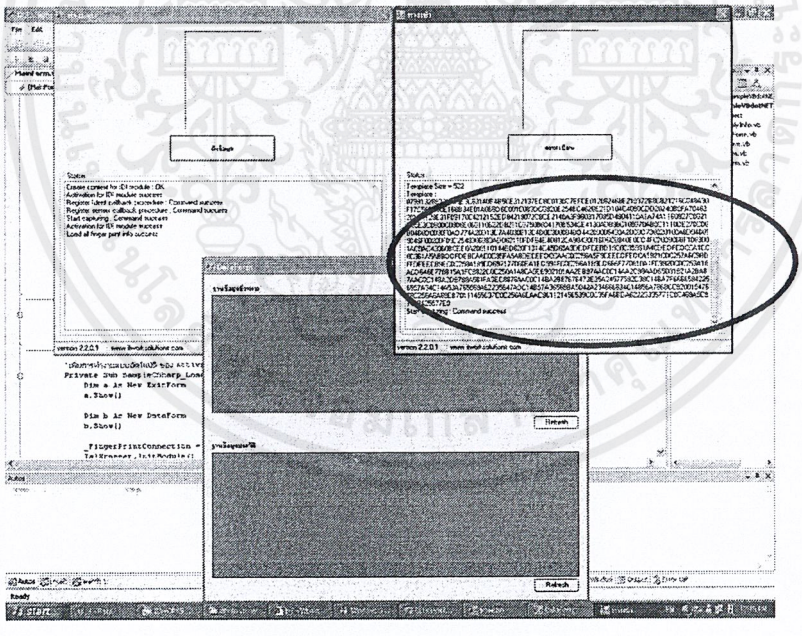
- จากรูปที่ 4.11 เป็นการเริ่มลงทะเบียนแล้ว 3 ครั้ง และเหลืออีกหนึ่งครั้ง เมื่อเราทำการสแกนครั้งสุดท้าย ก็จะแสดงดังรูป 4.12 เราก็ทำการคลิกที่ "OK"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



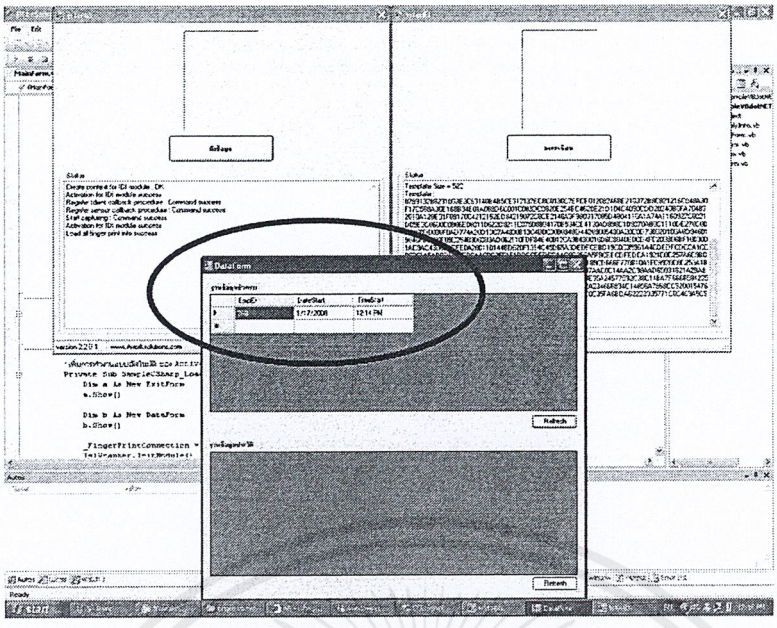
รูปที่ 4.12 ลงทะเบียนด้วยการสแกนลายนิ้วมือครบทุกช่อง

- จากรูปที่ 4.13 เป็นฐานข้อมูลที่ถูกจัดเก็บ หลังจากการสแกนลายนิ้วมือละทะเบียนเรียบร้อยแล้ว



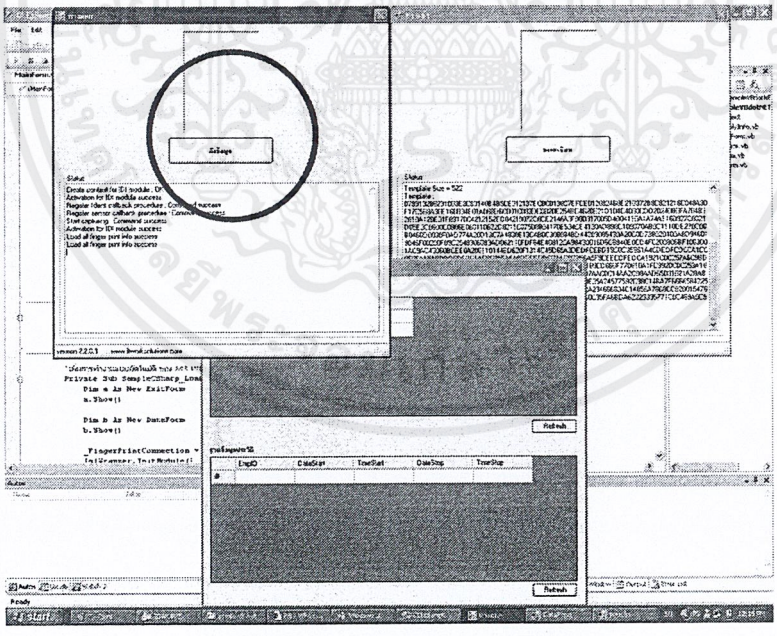
รูปที่ 4.13 แสดงรหัสฐานข้อมูลที่แปลงจากลายนิ้วมือที่สแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงข้อมูลที่อยู่ในฐานข้อมูลชั่วคราว

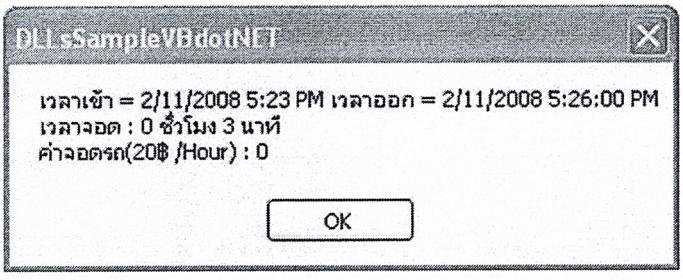
- จากรูป 4.14 เป็นการเก็บข้อมูลของผู้ใช้งาน โดยระบบจะทำการ Run ID อัตโนมัติ และจัดเก็บข้อมูลวันและเวลาของผู้ใช้งานโดยอ้างอิง ID ที่ระบบเซตให้ ในช่วง 0-999 โดยจะจัดเก็บอยู่ในฐานข้อมูลชั่วคราว



รูปที่ 4.15 การดึงข้อมูลการใช้บริการเพื่อคำนวณค่าบริการ

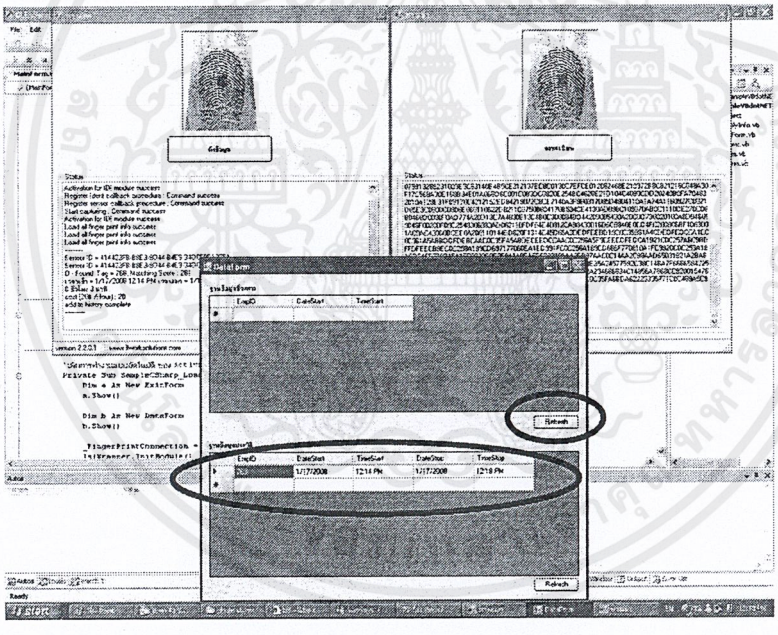
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากรูปที่ 4.15 เมื่อเราต้องการออกจากการใช้บริการเราจะทำการดึงข้อมูลจากฐานข้อมูลด้วยการสแกนลายนิ้วมือ และระบบภายในจะทำการเปรียบเทียบลายนิ้วมือเพื่อหาฐานข้อมูลที่ลงทะเบียนไว้ในข้างต้น



รูปที่ 4.16 เปรียบเทียบลายนิ้วมือและแสดงลายละเอียดการจอดรถ

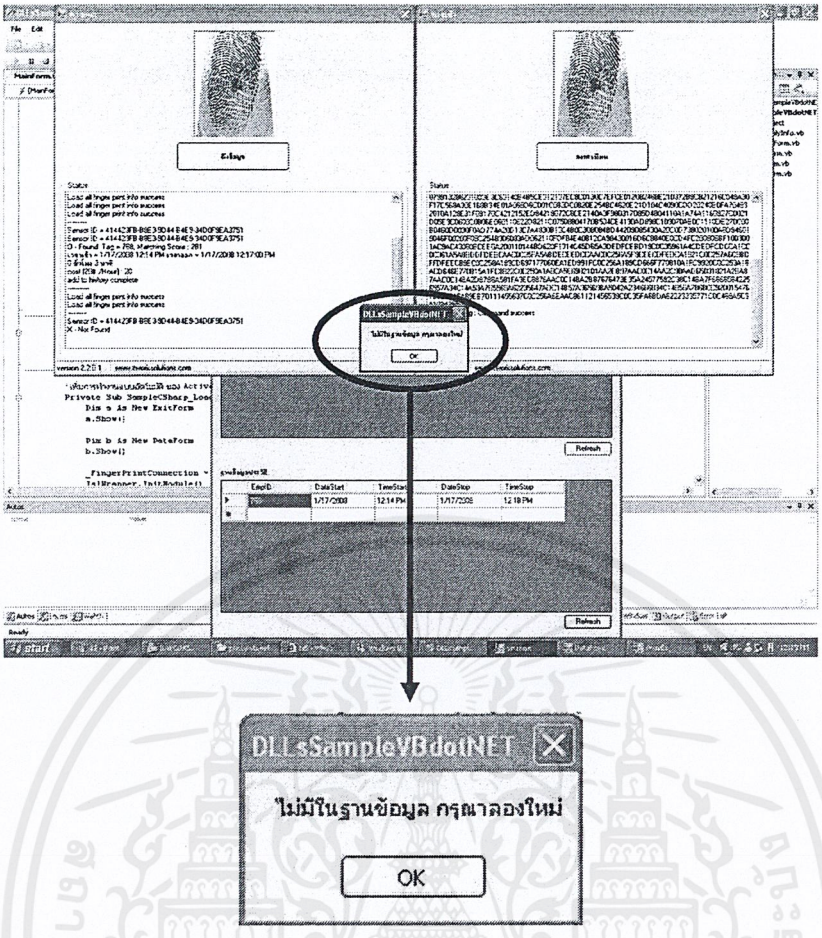
- จากรูปที่ 4.16 เป็นการเปรียบเทียบลายนิ้วมือ เพื่อดึงข้อมูลที่ลงทะเบียนไว้ข้างต้น และทำการประมวลผลแสดงหาจำนวนชั่วโมงการจอด และราคาค่าจอดรถ



รูปที่ 4.17 ข้อมูลที่อยู่ภายในฐานข้อมูลประวัติ

- จากรูปที่ 4.17 เมื่อทำการคลิกที่ “refresh” ข้อมูลที่อยู่ในฐานข้อมูลอ้างอิงจะทำการลบข้อมูลออก แต่จะถูกเก็บบันทึกในฐานข้อมูลประวัติ

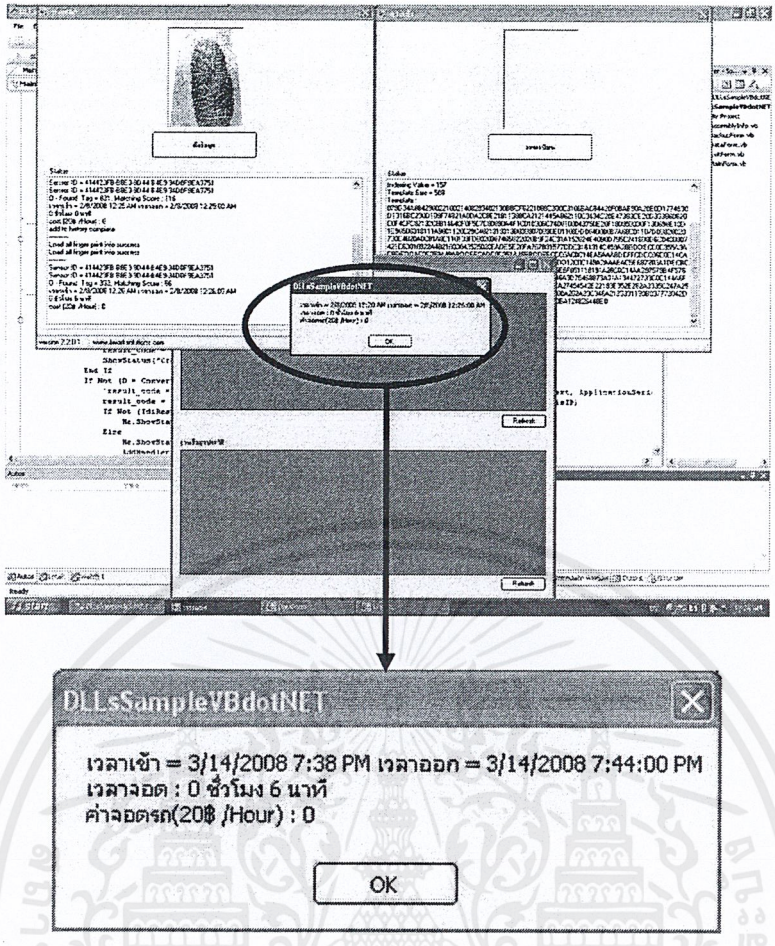
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 การไม่ตรวจพบข้อมูลในฐานข้อมูล

- จากรูปที่ 4.18 เป็นการแสดงว่าไม่พบฐานข้อมูลที่ทำการตรวจสอบ เนื่องจากลายนิ้วมือที่สแกนนั้น ไม่ตรงการฐานข้อมูลที่มีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

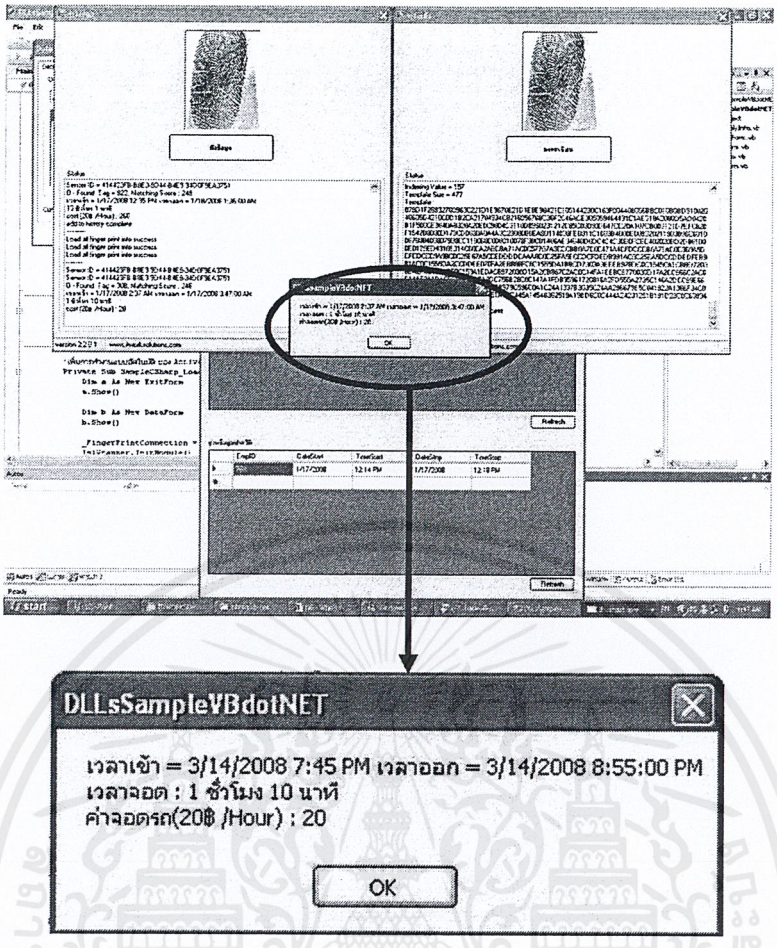


รูปที่ 4.19 การแสดงเวลาค่าจอดรถที่เวลารวม 6 นาที

- จากรูปที่ 4.19 เป็นการประมวลผลแสดงหาจำนวนชั่วโมงการจอด และราคาค่าจอดรถ โดยจะทำการแสดงลายละเอียดดังนี้

1. วันและเวลาเริ่มจอด [3/14/2008 7:38 PM]
2. วันและเวลาออก [3/14/2008 7:44 PM]
3. เวลาที่จอด [6 นาที]
4. ค่าจอดรถ [20 บาท/ชั่วโมง = 0 บาท]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

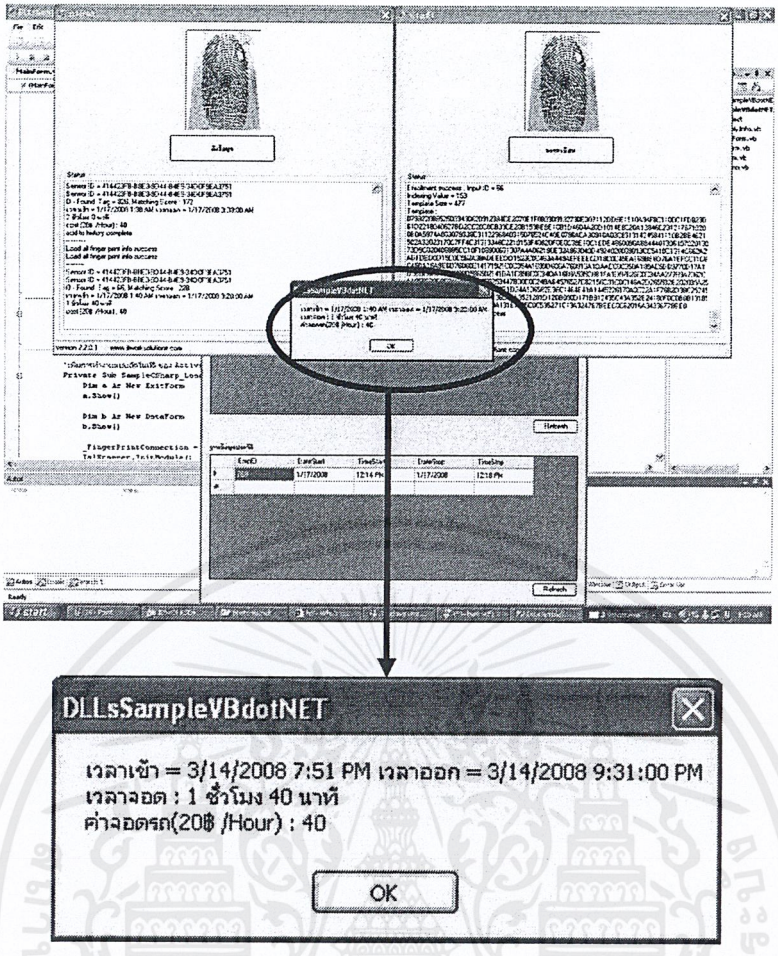


รูปที่ 4.20 การแสดงเวลาค่าจอดรถที่เวลารวม 1 ชั่วโมง 10 นาที

- จากรูปที่ 4.20 เป็นการประมวลผลแสดงหาจำนวนชั่วโมงการจอด และราคาค่าจอดรถ โดยจะทำการแสดงลายละเอียดดังนี้

1. วันและเวลาเริ่มจอด [3/14/2008 7:45 PM]
2. วันและเวลาออก [3/14/2008 8:55 PM]
3. เวลาที่จอด [1 ชั่วโมง 10 นาที]
4. ค่าจอดรถ [20 บาท/ชั่วโมง = 20 บาท]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 การแสดงเวลาค่าจอดรถที่เวลารวม 1 ชั่วโมง 40 นาที

- จากรูปที่ 4.21 เป็นการประมวลผลแสดงหาจำนวนชั่วโมงการจอด และราคาค่าจอดรถ โดยจะทำการแสดงลายละเอียดดังนี้

1. วันและเวลาเริ่มจอด [3/14/2008 7:51 PM]
2. วันและเวลาออก [3/14/2008 9:31 PM]
3. เวลาที่จอด [1 ชั่วโมง 40 นาที]
4. ค่าจอดรถ [20 บาท/ชั่วโมง = 40 บาท]

ค่าจอดรถ 20 บาทต่อชั่วโมง แต่เนื่องจากระบบจะคำนวณคิดค่าจอดรถโดยเริ่มตั้งแต่นาทีแรกจนถึง 30 นาทีจะสามารถเข้าจอดได้ฟรี นาทีที่ 31 จนถึง 1 ชั่วโมง 30 นาที จะถูกปัดคิดเป็น 1 ชั่วโมง คิดที่ราคา 20 บาท และตั้งแต่นาทีที่ 31 จะถูกปัดเพิ่มเป็น 2 ชั่วโมง คิดที่ราคา 40 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและบทวิจารณ์

5.1 สรุปผลการทดลอง

จากการทดลองการทำงานของระบบบริหารการจอดรถ ซึ่งสามารถใช้ประโยชน์ได้ดังนี้คือเมื่อนำรถเข้าไปจอดที่ลานจอดรถจะสามารถรู้ได้ว่าที่จอดรถตำแหน่งไหนว่างบ้าง โดยใช้อินฟราเรดในการตรวจจับรถที่เข้ามาจอดแล้วประมวลผลโดยไมโครคอนโทรลเลอร์ด้วยภาษาซี และจะแสดงผลทางจอแอลซีดี

มีการตรวจสอบลายนิ้วมือด้วยโปรแกรมเพื่อใช้ในการคิดเงิน โดยจะต้องสแกนลายนิ้วมือตรงทางเข้าเพื่อบันทึกเวลาเข้า และสแกนลายนิ้วมือตรงทางออก จากนั้นเครื่องจะทำการแสดงผลเวลาเข้า เวลาออก เวลาจอดทั้งหมด และจำนวนเงิน เพื่อเป็นการประหยัดเวลา ประหยัดทรัพยากร และยังเป็นการรักษาความปลอดภัยด้วยอีกทางหนึ่ง

5.2 วิจารณ์การทดลอง

เนื่องจากเรานำอินฟราเรดมาใช้เป็นวงจรเซนเซอร์ ที่มีทั้งภาครับและภาคส่ง โดยภาครับเรากำหนดใช้ตัวรับที่รับความถี่ได้ 38 กิโลเฮิรตซ์ ทำให้ต้องปรับภาคส่งให้มีความถี่เดียวกัน จึงจะสามารถรับส่งกันได้ ส่วนของจอแสดงผลแอลซีดีที่ตรงทางเข้าจะบอกตำแหน่งที่ว่าง จะไม่สามารถแสดงผลได้ทันทีในจังหวะที่รถเข้าไปแล้วแต่ยังไม่ถึงที่จอดรถ ทำให้ตัวเซนเซอร์ยังตรวจจับไม่ได้ อาจจะทำให้รถที่เข้าไปแต่ยังไม่มีที่จอดติดค้างอยู่ในลานจอดรถ

ความผิดพลาดในการตรวจสอบลายนิ้วมือมีค่าน้อยมาก ซึ่งน้อยคนนักที่จะไม่สามารถใช้เครื่องสแกนลายนิ้วมือนี้ได้ เพราะมีเพียงไม่กี่คนเท่านั้นที่จะมีลายนิ้วมือไม่สมบูรณ์ทั้ง 10 นิ้ว และในการสแกนลายนิ้วมือตรงทางเข้าและออกต้องใช้ลายนิ้วมือของคนคนเดียวกันและนิ้วเดิมเครื่องจึงสามารถตรวจจับได้ ซึ่งเป็นทั้งผลดีและผลเสีย และเครื่องสแกนนี้ยังมีข้อจำกัดตรงที่ถ้ามือที่สแกนเปียกก็ไม่สามารถสแกนได้

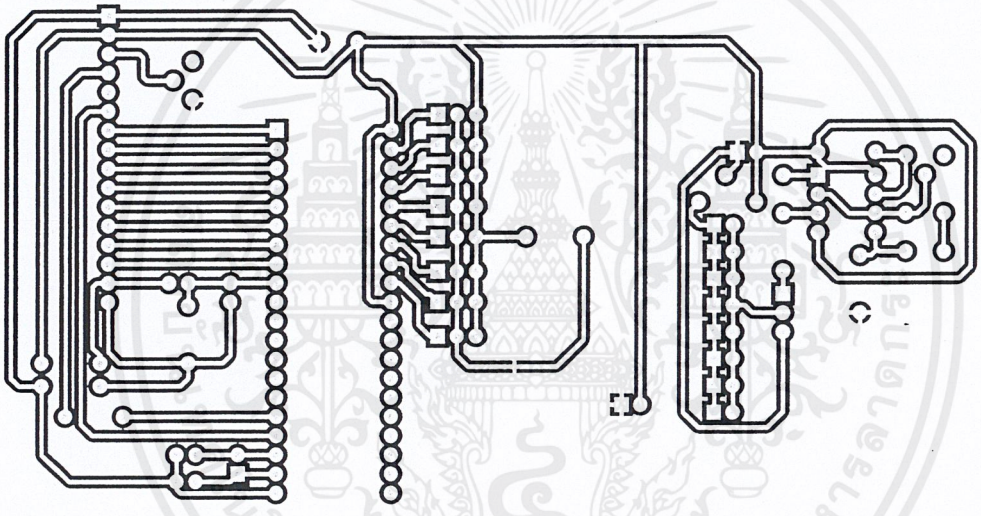
5.3 แนวทางในการพัฒนา

เราจะทำการพัฒนาให้ตัวไมโครคอนโทรลเลอร์ที่ใช้ในการระบุที่จอดรถ ให้สามารถลิงค์กับตัวโปรแกรมที่ใช้กับเครื่องสแกนลายนิ้วมือได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

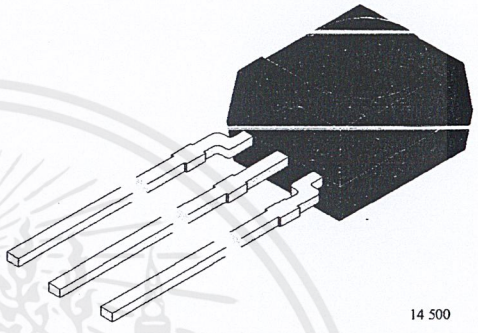
Photo Modules for PCM Remote Control Systems

Available types for different carrier frequencies

Type	fo	Type	fo
TSOP4830	30 kHz	TSOP4833	33 kHz
TSOP4836	36 kHz	TSOP4837	36.7 kHz
TSOP4838	38 kHz	TSOP4840	40 kHz
TSOP4856	56 kHz		

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.

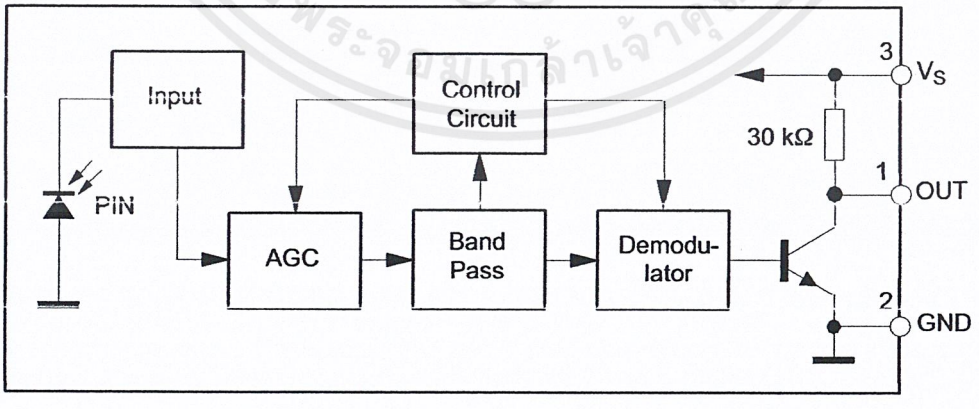


14 500

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
- High immunity against ambient light
- Continuous data transmission possible (800 bit/s)
- Suitable burst length ≥10 cycles/burst

Block Diagram



9612226

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

T_{amb} = 25°C

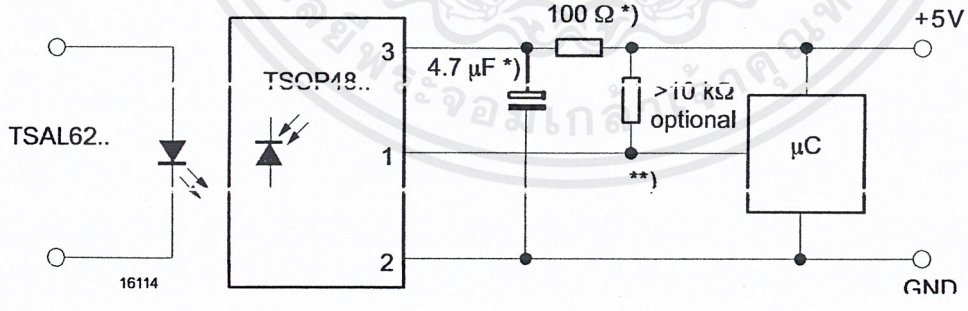
Parameter	Test Conditions	Symbol	Value	Unit
Supply Voltage	(Pin 3)	V_S	–0.3...6.0	V
Supply Current	(Pin 3)	I_S	5	mA
Output Voltage	(Pin 1)	V_O	0.3...6.0	V
Output Current	(Pin 1)	I_O	5	mA
Junction Temperature		T_j	100	°C
Storage Temperature Range		T_{stg}	–25...+85	°C
Operating Temperature Range		T_{amb}	–25...+85	°C
Power Consumption	(T_{amb} ≦ 85 °C)	P_{tot}	50	mW
Soldering Temperature	t ≦ 10 s, 1 mm from case	T_{sd}	260	°C

Basic Characteristics

T_{amb} = 25°C

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Supply Current (Pin 3)	V_S = 5 V, E_v = 0	I_{SD}	0.8	1.1	1.5	mA
	V_S = 5 V, E_v = 40 klx, sunlight	I_{SH}		1.4		mA
Supply Voltage (Pin 3)		V_S	4.5		5.5	V
Transmission Distance	E_v = 0, test signal see fig.7, IR diode TSAL6200, I_F = 250 mA	d		35		m
Output Voltage Low (Pin 1)	I_{OSL} = 0.5 mA, E_e = 0.7 mW/m²	V_{OSL}			250	mV
Irradiance (30 – 40 kHz)	Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$, test signal see fig.7	$E_{e\,min}$		0.2	0.4	mW/m²
Irradiance (56 kHz)	Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$, test signal see fig.7	$E_{e\,min}$		0.3	0.6	mW/m²
Irradiance	$t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$	$E_{e\,max}$	30			W/m²
Directivity	Angle of half transmission distance	$\varphi_{1/2}$		±45		deg

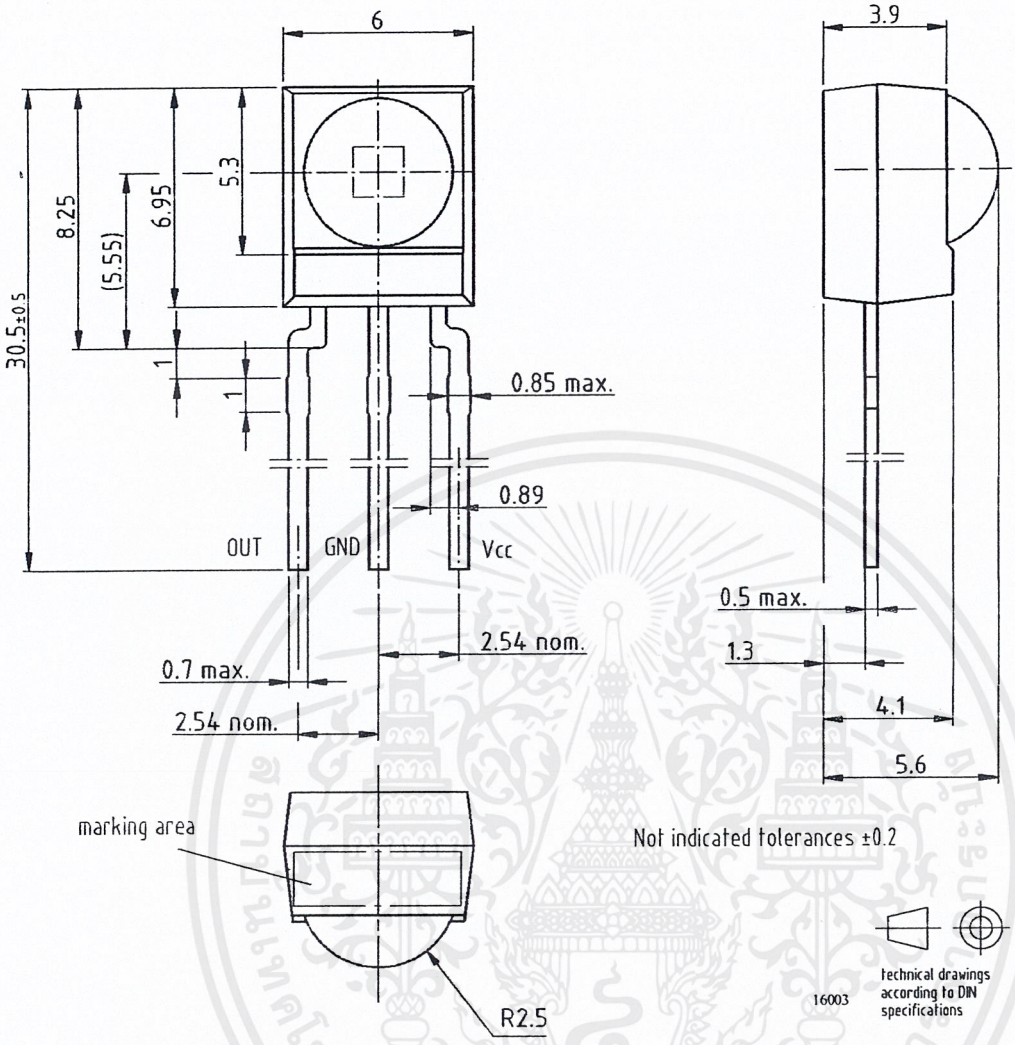
Application Circuit



*) recommended to suppress power supply disturbances
**) The output voltage should not be hold continuously at a voltage below 3.3V by the external circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions in mm



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Analog-to-Digital Conversion Utilizing the AT89CX051 Microcontrollers

The Atmel AT89C1051 and AT89C2051 microcontrollers feature on-chip Flash, low pin count, wide operating voltage range and an integral analog comparator. This application note describes two low-cost analog-to-digital conversion techniques which utilize the analog comparator in the AT89C1051 and AT89C2051 microcontrollers.

RC Analog-to-Digital Converter

This conversion method offers an extremely low component count at the expense of accuracy and conversion time. In the example presented below, resolution is better than 50 millivolts, accuracy is somewhat less than a tenth of a Volt and conversion time is seven milliseconds or less.

As shown in Figure 1, the RC analog-to-digital conversion method requires only two resistors and a capacitor in addition to the AT89CX051 microcontroller. A microcontroller output (pin 11), which swings from approximately ground to V_{CC}, alternately charges and discharges the capacitor connected to the non-inverting input of the internal comparator (pin 12). The microcontroller measures the time required for the voltage on the capacitor to match the unknown voltage applied to the inverting input of the internal comparator (pin 13). The unknown voltage is a function of the measured time.

The HP5082-7300 LED displays shown in Figure 1 are not required for the conversion, but are utilized by the software to implement a simple two-digit voltmeter. The result of the analog-to-digital conversion is displayed in volts and tenths of a volt on the two displays. The voltmeter application does not utilize the full resolution of the RC conversion software, but serves to demonstrate the method as well as providing a tool for debug.

The waveform for a typical capacitor charge/discharge cycle is shown in Figure 2. The discharge portion of the curve is identical to the charge portion rotated about the line $V_C = V_{CC}/2$. The equations and discussion below apply to the charge portion of the cycle, except where indicated.

The voltage on the capacitor as a function of time is given by the exponential equation:

$$V_C = V_{CC} (1\text{-}e^{-t/RC}) \qquad (1)$$

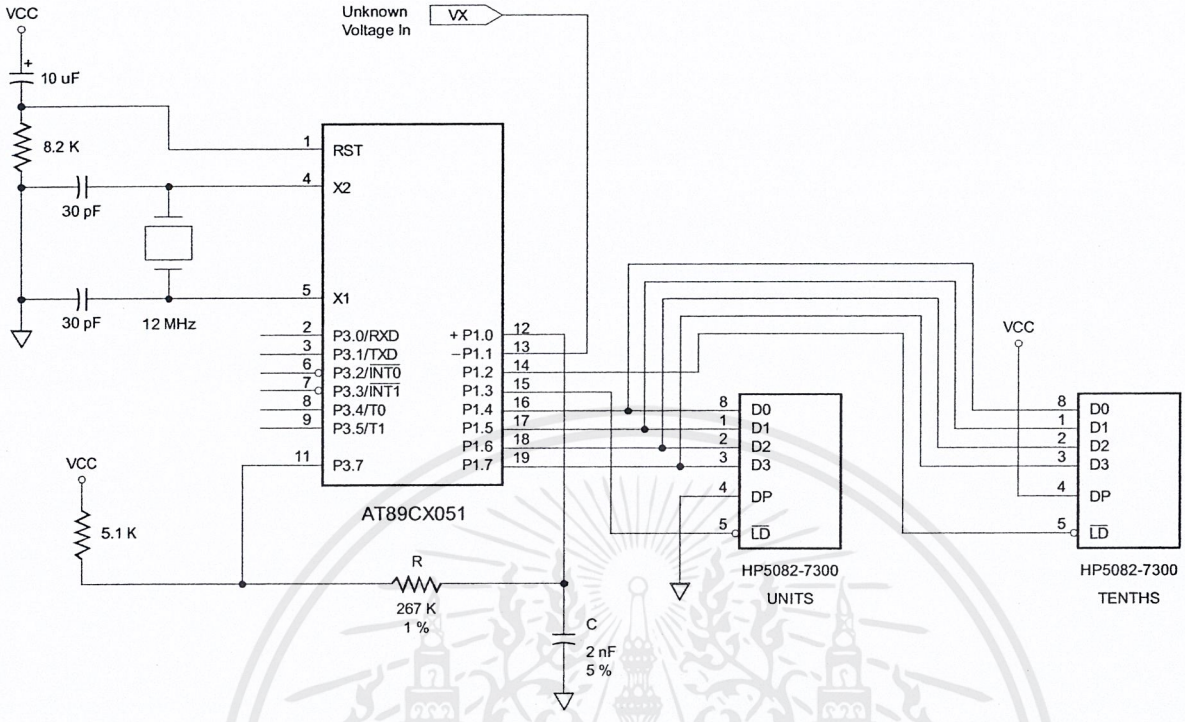
where V_C is the voltage on the capacitor at time t, V_{CC} is the supply voltage and RC is the product of the values of the resistor and capacitor. Note that voltage is expressed in Volts, time in seconds, resistance in Ohms and capacitance in Farads. The product RC is also known as the "time constant" of the network and affects the shape of the waveform. The waveform is steepest when capacitor charging or discharging begins and flattens with time.

The first problem with the RC conversion method is the difficulty of solving the exponential equation without utilizing floating point calculations and transcendental functions. On a compressed time scale, the exponential curve appears straight over much of its length, suggesting that it might be approximated by a line. This scheme fails due to the continuous variation in slope over the length of the curve, which produces significant error. It also does not address the problem where the curve rolls off severely near the asymptote at V_{CC}.

0524A-B–12/97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 1. Two-Digit Voltmeter



The microcontroller need not solve the exponential equation in real time if a lookup table is used to map pre-calculated values to each sampled time interval. This scheme allows the data to be encoded and formatted as required by the application while simplifying the conversion software. Symmetries in the data may be exploited to reduce the size of the table.

The second problem with the RC conversion method is the substantial error which results from variations in component values. Figure 3 shows an exaggerated view of the variation in the voltage on the capacitor due to variations in the values of the resistor and capacitor. As shown in the figure, the variation in the voltage on the capacitor decreases as the voltage on the capacitor decreases.

The symmetry of the capacitor charge/discharge cycle can be exploited to reduce the effect of variations in component values on conversion accuracy. This is done by utilizing the charge portion of the cycle to measure voltages less than $V_{CC}/2$ and the discharge portion to measure voltages greater than $V_{CC}/2$. The worst case error is reduced to the error at $V_{CC}/2$.

Before component values can be assigned, the time interval at which the comparator output is to be sampled must be determined. The sample interval should be as short as possible to maximize converter resolution and minimize conversion time. The sample interval is limited by the time required to execute the requisite code, which is determined by the clock rate of the microcontroller. In the voltmeter application, the microcontroller operates with a 12-MHz clock, resulting in a sample interval of five microseconds.

The time constant (RC) affects the shape of the capacitor charge/discharge waveform. The value of the time constant must be chosen so that the steepest parts of the waveform are resolvable to the desired resolution. The steepest part of the charge portion of the waveform occurs near the origin, while the steepest part of the discharge portion occurs near V_{CC}. Due to the symmetry of the waveform, the same time constant may be used for measurements made on either portion of the waveform.

Figure 4 shows an expanded view of the relationship between voltage and sample time near the origin. In the figure, ΔV is the desired voltage resolution of the converter and Δt is the sample interval determined previously. The curve labeled 'V_C' represents the voltage on the capacitor, which appears linear at this scale. In the figure, the slope of the curve is ideal, causing sampling to occur near the center of the voltage intervals. The slope of the curve may be less than shown, but may not be greater, or resolution will be lost. Note that the first sample is offset from the origin by 1/2 Δt to center the sample in the first voltage interval.

To obtain the minimum value of the time constant which will produce the required slope at the first sample, solve Equation 1 for RC:

$$RC = -t/1n(1-V_C/V_{CC}) \quad (2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

precision components, variations in component values may contribute an error of ±0.104-volt, as shown below.

To calculate the worst case error at V_C = 2.5-volts, first determine the corresponding t at the nominal values of R and C using Equation 3:

$$t = -RnomCnom \cdot \ln(1-V_C/V_{CC})$$
$$= -RnomCnom \cdot \ln(1-2.5/5.0)$$
$$= -RnomCnom \cdot \ln(0.5).$$

Substitute for t in Equation 1 to get minimum V_C:

$$V_C min = V_{CC}\,(1-e^{-t/(Rmax\ Cmax)})$$
$$= V_{CC}\,(1-e^{(Rnom\ Cnom/Rmax\ Cmax)\ln(0.5)})$$
$$= 5\,(1-e^{\ln(0.5)/(1.01 \cdot 1.05)})$$
$$\cong 2.399\ V$$

Again, for maximum VC:

$$V_C max = V_{CC}\,(1-e^{-t/(Rmin\ Cmin)})$$
$$= V_{CC}\,(1-e^{(Rnom\ Cnom/Rmin\ Cmin)\ln(0.5)})$$
$$= 5\,(1-e^{\ln(0.5)/(0.99 \cdot 0.95)})$$
$$\cong 2.607\ V$$

The results show a variation of 0.208-volts at 2.5-volts, or a worst case error of ±0.104-volts. The worst case conversion error may be further reduced by utilizing components with tighter tolerances. Conversion accuracy and linearity are also affected by the characteristics of the capacitor. The capacitor used in the voltmeter prototype is a polystyrene film type, which not only provides good accuracy, but minimizes error due to dielectric absorption and other effects.

Error sources which have not been examined include: comparator limitations; asymmetries between the charge and discharge portions of the cycle; failure of the voltage on the capacitor to reach ground or V_{CC}; variations in V_{CC}. The contributions to conversion error made by these sources can be expected to increase error to somewhat more than the value due to component tolerances alone.

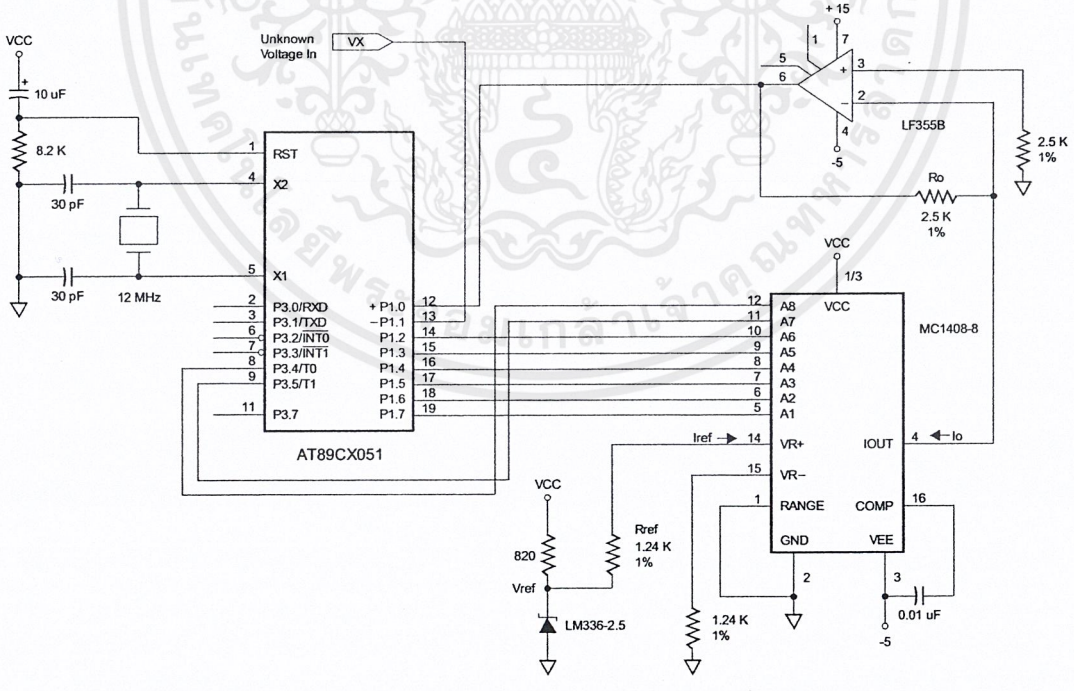
Successive Approximation Analog-to-Digital Converter

This conversion method offers good resolution and accuracy and a short conversion time at the expense of increased component count.

Successive approximation (SA) ADCs incorporate a digital-to-analog converter (DAC), a comparator and a successive approximation register (SAR). The SAR controls the conversion by performing a search for the binary code which, when fed to the DAC, will produce an output matching the voltage to be converted. The comparator compares the DAC output to the unknown voltage and returns the result to the SAR.

The SAR begins the search with the most significant DAC bit, which controls the widest output variation, and moves toward the least significant bit, causing the DAC output to "zero in" on the unknown value. The result of the trial is the

Figure 5. Successive Approximation Analog-to-Digital Converter



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

February 2000

LM555
Timer

General Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200mA or drive TTL circuits.

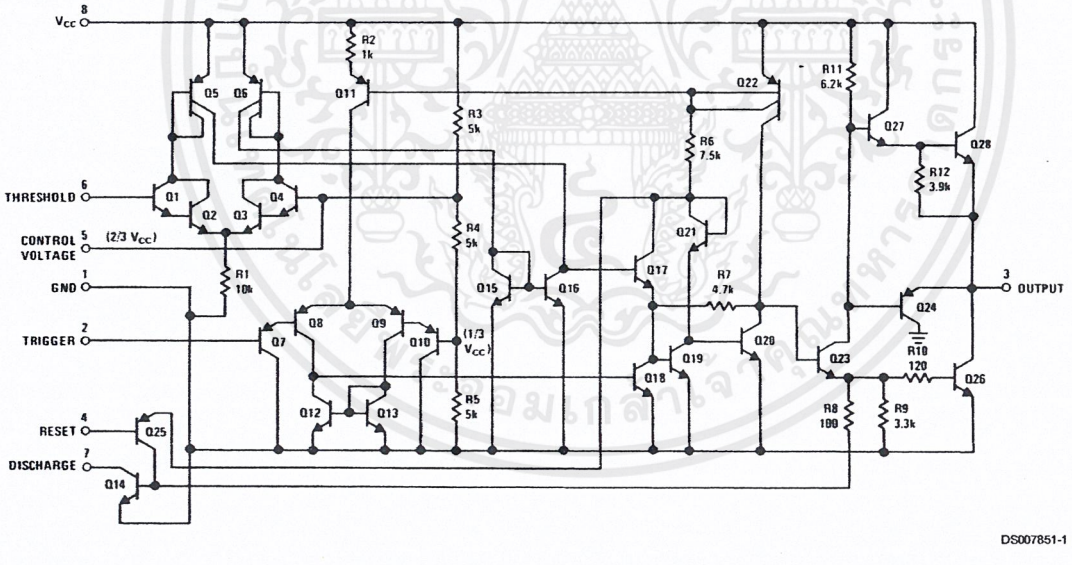
Features

- Direct replacement for SE555/NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes
- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per °C
- Normally on and normally off output
- Available in 8-pin MSOP package

Applications

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

Schematic Diagram



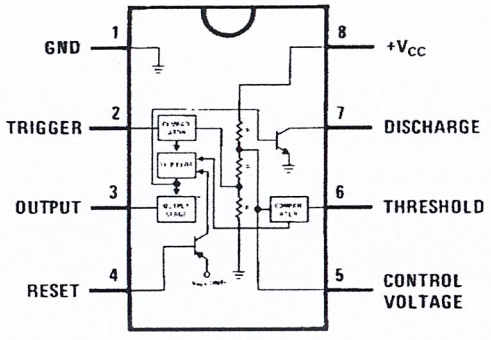
DS007851-1

© 2000 National Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagram

Dual-In-Line, Small Outline
and Molded Mini Small Outline Packages



DS007851-3

Top View

Ordering Information

Package	Part Number	Package Marking	Media Transport	NSC Drawing
8-Pin SOIC	LM555CM	LM555CM	Rails	M08A
	LM555CMX	LM555CM	2.5k Units Tape and Reel	
8-Pin MSOP	LM555CMM	Z55	1k Units Tape and Reel	MUA08A
	LM555CMMX	Z55	3.5k Units Tape and Reel	
8-Pin MDIP	LM555CN	LM555CN	Rails	N08E

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+18V
Power Dissipation (Note 3)	
LM555CM, LM555CN	1180 mW
LM555CMM	613 mW
Operating Temperature Ranges	
LM555C	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Soldering Information

Dual-In-Line Package	
Soldering (10 Seconds)	260°C
Small Outline Packages	
(SOIC and MSOP)	
Vapor Phase (60 Seconds)	215°C
Infrared (15 Seconds)	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics (Notes 1, 2)

(T_A = 25°C, V_{CC} = +5V to +15V, unless othewise specified)

Parameter	Conditions	Limits			Units
		LM555C			
		Min	Typ	Max	
Supply Voltage		4.5		16	V
Supply Current	V_{CC} = 5V, R_L = ∞		3	6	mA
	V_{CC} = 15V, R_L = ∞ (Low State) (Note 4)		10	15	mA
Timing Error, Monostable					
Initial Accuracy			1		%
Drift with Temperature	R_A = 1k to 100kΩ, C = 0.1µF, (Note 5)		50		ppm/°C
Accuracy over Temperature			1.5		%
Drift with Supply			0.1		%/V
Timing Error, Astable					
Initial Accuracy			2.25		%
Drift with Temperature	R_A, R_B = 1k to 100kΩ, C = 0.1µF, (Note 5)		150		ppm/°C
Accuracy over Temperature			3.0		%
Drift with Supply			0.30		%/V
Threshold Voltage			0.667		x V_{CC}
Trigger Voltage	V_{CC} = 15V		5		V
	V_{CC} = 5V		1.67		V
Trigger Current			0.5	0.9	µA
Reset Voltage		0.4	0.5	1	V
Reset Current			0.1	0.4	mA
Threshold Current	(Note 6)		0.1	0.25	µA
Control Voltage Level	V_{CC} = 15V	9	10	11	V
	V_{CC} = 5V	2.6	3.33	4	V
Pin 7 Leakage Output High			1	100	nA
Pin 7 Sat (Note 7)					
Output Low	V_{CC} = 15V, I_7 = 15mA		180		mV
Output Low	V_{CC} = 4.5V, I_7 = 4.5mA		80	200	mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applications Information

MONOSTABLE OPERATION

In this mode of operation, the timer functions as a one-shot (*Figure 1*). The external capacitor is initially held discharged by a transistor inside the timer. Upon application of a negative trigger pulse of less than 1/3 V_{CC} to pin 2, the flip-flop is set which both releases the short circuit across the capacitor and drives the output high.

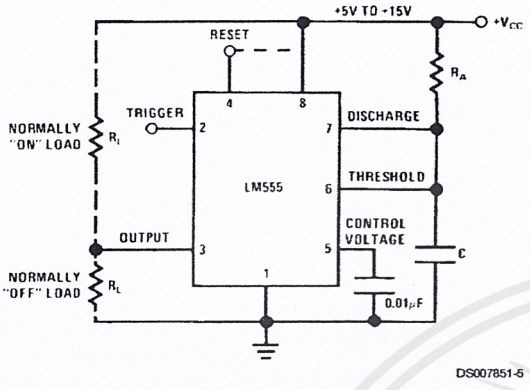
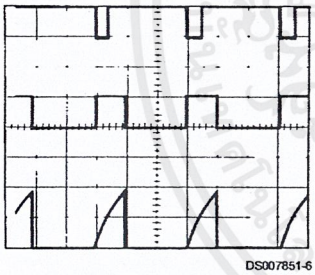


FIGURE 1. Monostable

The voltage across the capacitor then increases exponentially for a period of t = 1.1 R_A C, at the end of which time the voltage equals 2/3 V_{CC}. The comparator then resets the flip-flop which in turn discharges the capacitor and drives the output to its low state. *Figure 2* shows the waveforms generated in this mode of operation. Since the charge and the threshold level of the comparator are both directly proportional to supply voltage, the timing internal is independent of supply.



V_{CC} = 5V
TIME = 0.1 ms/DIV.
R_A = 9.1kΩ
C = 0.01µF

Top Trace: Input 5V/Div.
Middle Trace: Output 5V/Div.
Bottom Trace: Capacitor Voltage 2V/Div.

FIGURE 2. Monostable Waveforms

During the timing cycle when the output is high, the further application of a trigger pulse will not effect the circuit so long as the trigger input is returned high at least 10µs before the end of the timing interval. However the circuit can be reset during this time by the application of a negative pulse to the reset terminal (pin 4). The output will then remain in the low state until a trigger pulse is again applied.

When the reset function is not in use, it is recommended that it be connected to V_{CC} to avoid any possibility of false triggering.

Figure 3 is a nomograph for easy determination of R, C values for various time delays.

NOTE: In monostable operation, the trigger should be driven high before the end of timing cycle.

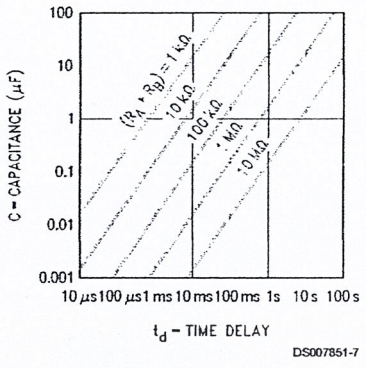


FIGURE 3. Time Delay

ASTABLE OPERATION

If the circuit is connected as shown in *Figure 4* (pins 2 and 6 connected) it will trigger itself and free run as a multivibrator. The external capacitor charges through R_A + R_B and discharges through R_B. Thus the duty cycle may be precisely set by the ratio of these two resistors.

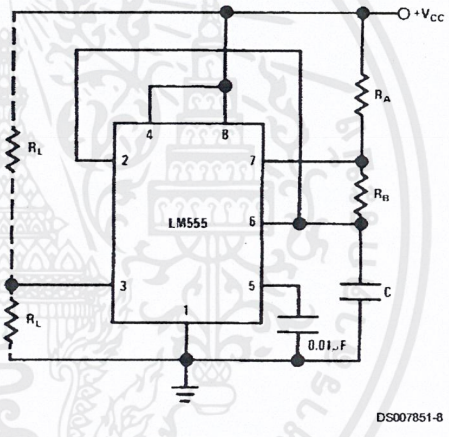


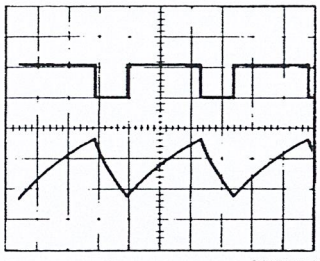
FIGURE 4. Astable

In this mode of operation, the capacitor charges and discharges between 1/3 V_{CC} and 2/3 V_{CC}. As in the triggered mode, the charge and discharge times, and therefore the frequency are independent of the supply voltage.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applications Information (Continued)

Figure 5 shows the waveforms generated in this mode of operation.



DS007851-9

V_{CC} = 5V Top Trace: Output 5V/Div.
TIME = 20μs/DIV. Bottom Trace: Capacitor Voltage 1V/Div.
R_A = 3.9kΩ
R_B = 3kΩ
C = 0.01μF

FIGURE 5. Astable Waveforms

The charge time (output high) is given by:

$$t_1 = 0.693\ (R_A + R_B)\ C$$

And the discharge time (output low) by:

$$t_2 = 0.693\ (R_B)\ C$$

Thus the total period is:

$$T = t_1 + t_2 = 0.693\ (R_A + 2R_B)\ C$$

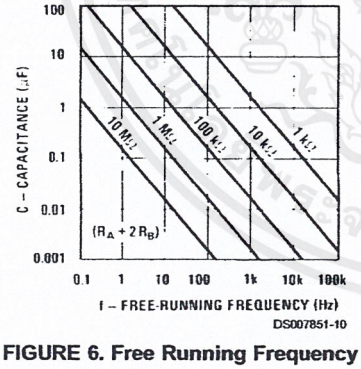
The frequency of oscillation is:

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)\ C}$$

Figure 6 may be used for quick determination of these RC values.

The duty cycle is:

$$D = \frac{R_B}{R_A + 2R_B}$$

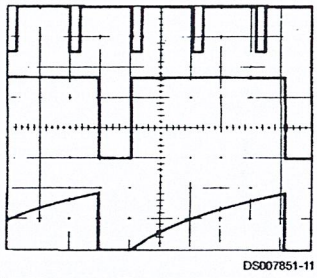


DS007851-10

FIGURE 6. Free Running Frequency

FREQUENCY DIVIDER

The monostable circuit of *Figure 1* can be used as a frequency divider by adjusting the length of the timing cycle. *Figure 7* shows the waveforms generated in a divide by three circuit.



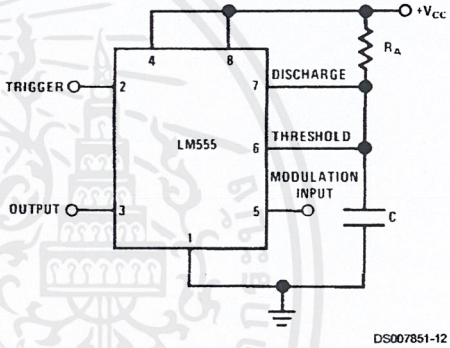
DS007851-11

V_{CC} = 5V Top Trace: Input 4V/Div.
TIME = 20μs/DIV. Middle Trace: Output 2V/Div.
R_A = 9.1kΩ Bottom Trace: Capacitor 2V/Div.
C = 0.01μF

FIGURE 7. Frequency Divider

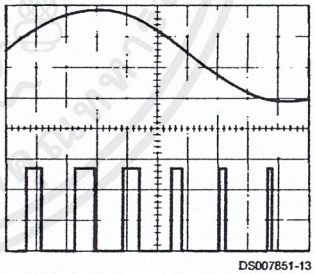
PULSE WIDTH MODULATOR

When the timer is connected in the monostable mode and triggered with a continuous pulse train, the output pulse width can be modulated by a signal applied to pin 5. *Figure 8* shows the circuit, and in *Figure 9* are some waveform examples.



DS007851-12

FIGURE 8. Pulse Width Modulator



DS007851-13

V_{CC} = 5V Top Trace: Modulation 1V/Div.
TIME = 0.2 ms/DIV. Bottom Trace: Output Voltage 2V/Div.
R_A = 9.1kΩ
C = 0.01μF

FIGURE 9. Pulse Width Modulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applications Information (Continued)

PULSE POSITION MODULATOR

This application uses the timer connected for astable operation, as in *Figure 10*, with a modulating signal again applied to the control voltage terminal. The pulse position varies with the modulating signal, since the threshold voltage and hence the time delay is varied. *Figure 11* shows the waveforms generated for a triangle wave modulation signal.

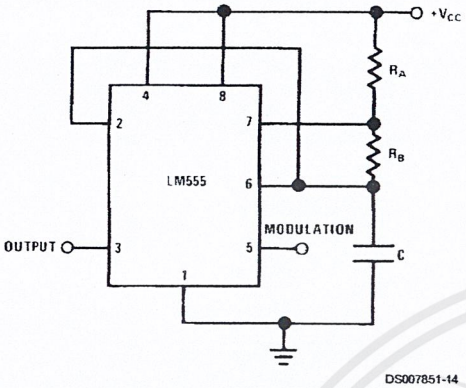
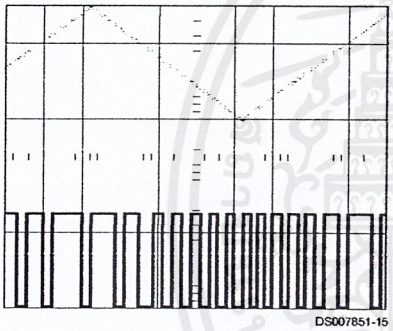


FIGURE 10. Pulse Position Modulator



V_{CC} = 5V
TIME = 0.1 ms/DIV.
R_A = 3.9kΩ
R_B = 3kΩ
C = 0.01μF

Top Trace: Modulation Input 1V/Div.
Bottom Trace: Output 2V/Div.

FIGURE 11. Pulse Position Modulator

LINEAR RAMP

When the pullup resistor, R_A, in the monostable circuit is replaced by a constant current source, a linear ramp is generated. *Figure 12* shows a circuit configuration that will perform this function.

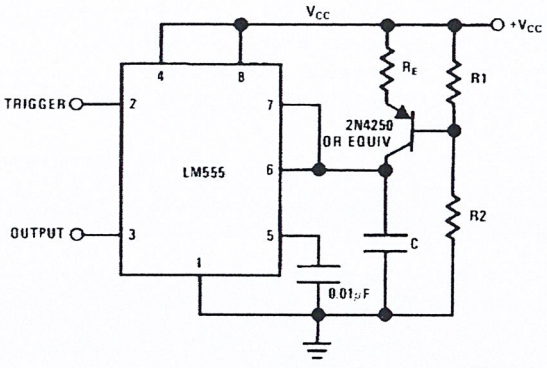


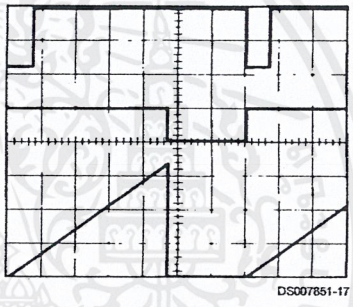
FIGURE 12.

Figure 13 shows waveforms generated by the linear ramp. The time interval is given by:

$$T = \frac{2/3\ V_{CC}\ R_E\ (R_1 - R_2)\ C}{R_1\ V_{CC} - V_{BE}\ (R_1 + R_2)}$$

$V_{BE} \cong 0.6V$

$V_{BE} \cong 0.6V$



V_{CC} = 5V
TIME = 20μs/DIV.
R_1 = 47kΩ
R_2 = 100kΩ
R_E = 2.7 kΩ
C = 0.01 μF

Top Trace: Input 3V/Div.
Middle Trace: Output 5V/Div.
Bottom Trace: Capacitor Voltage 1V/Div.

FIGURE 13. Linear Ramp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applications Information (Continued)

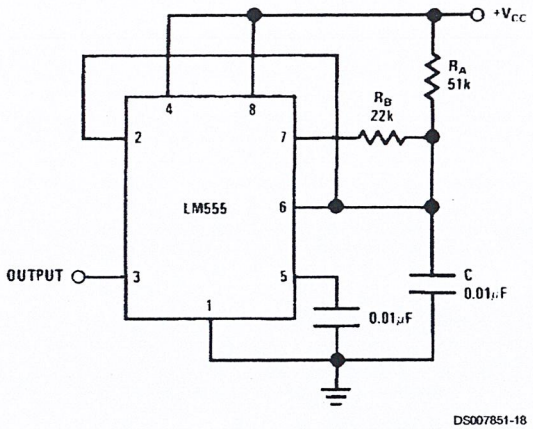
50% DUTY CYCLE OSCILLATOR

For a 50% duty cycle, the resistors R_A and R_B may be connected as in *Figure 14*. The time period for the output high is the same as previous, $t_1 = 0.693\ R_A\ C$. For the output low it is t_2 =

$$\left[(R_A R_B)/(R_A + R_B)\right] C\ ln\left[\frac{R_B - 2R_A}{2R_B - R_A}\right]$$

Thus the frequency of oscillation is

$$f = \frac{1}{t_1 + t_2}$$



DS007851-18

FIGURE 14. 50% Duty Cycle Oscillator

Note that this circuit will not oscillate if R_B is greater than 1/2 R_A because the junction of R_A and R_B cannot bring pin 2 down to 1/3 V_{CC} and trigger the lower comparator.

ADDITIONAL INFORMATION

Adequate power supply bypassing is necessary to protect associated circuitry. Minimum recommended is 0.1µF in parallel with 1µF electrolytic.

Lower comparator storage time can be as long as 10µs when pin 2 is driven fully to ground for triggering. This limits the monostable pulse width to 10µs minimum.

Delay time reset to output is 0.47µs typical. Minimum reset pulse width must be 0.3µs, typical.

Pin 7 current switches within 30ns of the output (pin 3) voltage.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#pragma SMALL
#include<reg51.h>
sfr DATA_BUS = 0x90;
sbit RS = P3^6;
sbit E = P3^7;
sbit in1 = P0^0;
sbit in2 = P0^1;
sbit in3 = P0^2;
sbit in4 = P0^3;
sbit in5 = P0^4;
sbit in6 = P0^5;
sbit in7 = P0^6;
sbit in8 = P0^7;
void lnitlO(void);
void lnitl1(void);
void delay(void);
static void EnableLCD(int t);
void ClearScreen(void);
void PutChar(char c);
void type2(void);
void type(void);
void type3(void);
void type4(void);
void type5(void);
void type6(void);
void type7(void);
void type8(void);
void type9(void);
void type10(void);
void main(void)
{
        lnitlO();
while(1)
{       ClearScreen();

        if(in1==0)
        {
                type();
        }
        if(in2==0)
        {
                type2();
        }
        if(in3==0)
        {
                type3();
        }
        if(in4==0)
        {
                type5();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        if(in5==0)
        {
                type6();
        }
        if(in6==0)
        {
                type7();
        }
        if(in7==0)
        {
                type8();
        }
        if(in8==0)
        {
                type9();
        }
        if(in1==1&&in2==1&in3==1&&in4==1&&in5==1&in6==1&&in7==1&&in
8==1)
        {       ClearScreen();
                type4();
        }
        if(in1==0&&in2==0&in3==0&&in4==0&&in5==0&in6==0&&in7==0&&in
8==0)
        {       ClearScreen();
                lnitl1();
                type10();
        }
}
}
void lnitl1(void)
{       RS = 0;
        DATA_BUS = 0x30;EnableLCD(255);
        DATA_BUS = 0x01;EnableLCD(255);
        DATA_BUS = 0x0C;EnableLCD(255);
        DATA_BUS = 0x14;EnableLCD(255);
        delay();
        RS = 1;
}
void type(void)
{
        PutChar('A');
        PutChar(' ');
        delay();
}
void delay()
{       int i,j;
        for(i=0;i<1000;i++)
        for(j=0;j<100;j++)
                i = i+0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void type4(void)
{       PutChar('S');
        PutChar('o');
        PutChar('r');
        PutChar('r');
        PutChar('y');
        PutChar('.');
        PutChar('I');
        PutChar('t');
        PutChar(' ');
        PutChar('i');
        PutChar('s');
        PutChar(' ');
        PutChar('F');
        PutChar('U');
        PutChar('L');
        PutChar('L');
        delay();
}
void type2(void)
{
        PutChar('B');
        PutChar(' ');
        delay();
}
void type3(void)
{
        PutChar('C');
        PutChar(' ');
        delay();
}
void type5(void)
{
        PutChar('D');
        PutChar(' ');
        delay();
}
void type6(void)
{
        PutChar('E');
        PutChar(' ');
        delay();
}
void type7(void)
{
        PutChar('F');
        PutChar(' ');
        delay();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void type8(void)
{
        PutChar('G');
        PutChar(' ');
        delay();
}
void type9(void)
{
        PutChar('H');
        PutChar(' ');
        delay();
}
void type10(void)
{
        PutChar('E');
        PutChar('M');
        PutChar('P');
        PutChar('T');
        PutChar('Y');
        PutChar(' ');
        delay();
}
static void EnableLCD(int t)
{       unsigned char i;
        E = 1;
        for(i=0;i<t;i++)i=i;
        E = 0;
        for(i=0;i<t;i++)i=i;
}
void lnitlO(void)
{       RS = 0;
        DATA_BUS = 0x38;EnableLCD(255);
        DATA_BUS = 0x01;EnableLCD(255);
        DATA_BUS = 0x0C;EnableLCD(255);
        DATA_BUS = 0x18;EnableLCD(255);
        delay();
        RS = 1;
}
void ClearScreen(void)
{       RS = 0;
        DATA_BUS = 0x01;EnableLCD(255);
        RS = 1;
}
void PutChar(char c)
{
        DATA_BUS=c;EnableLCD(45);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Imports System
Imports System.Drawing
Imports System.Collections
Imports System.ComponentModel
Imports System.Windows.Forms
Imports System.Data
Imports System.Data.OleDb
Imports Microsoft.VisualBasic
Imports Itworks.Idworks

Namespace DLLsSampleVBdotNET
    Public Class MainForm
        Inherits System.Windows.Forms.Form
        Private components As System.ComponentModel.Container = Nothing
        Private MainStatusBar As System.Windows.Forms.StatusBar
        Private StatusGroupBox As System.Windows.Forms.GroupBox
        Friend StatusTextBox As System.Windows.Forms.TextBox
        Private statusBarPanel2 As System.Windows.Forms.StatusBarPanel
        Private statusBarPanel1 As System.Windows.Forms.StatusBarPanel
        Private FingerPrintPictureBox As
System.Windows.Forms.PictureBox
        Private WithEvents EnrollmentButton As
System.Windows.Forms.Button

        Private Const _FINGERPRINT_CONNECTION_STRING As String =
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Program Files\IT
WORKS\IDWORKS Integrator\Samples\db\IDWORKSDemo.mdb;Persist Security
Info=False"
        Private _FingerPrintConnection As OleDbConnection
        Private _IdiContext As UInt32 = Convert.ToUInt32(0)
        Private _FpeContext As UInt32 = Convert.ToUInt32(0)

        Private sensorID As String = "9DC6E3BB-B5E2-E341-9AFD-
5FB220F46A45"
        Private serialID As String = "6A1B-9B36-24BB-D4C3"
        Private regisID As String = "4AF3-EABC-6F9F-F2E5-04B4-14B6-
F79A-EF74"


        Public Sub New()
            Application.EnableVisualStyles()
            InitializeComponent()
        End Sub

        Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
            If disposing Then
                If Not (components Is Nothing) Then
                    components.Dispose()
                End If
            End If
            MyBase.Dispose(disposing)
        End Sub

        Private Sub InitializeComponent()
            Me.EnrollmentButton = New System.Windows.Forms.Button
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Me.FingerPrintPictureBox = New
System.Windows.Forms.PictureBox
            Me.statusBarPanel1 = New
System.Windows.Forms.StatusBarPanel
            Me.statusBarPanel2 = New
System.Windows.Forms.StatusBarPanel
            Me.StatusTextBox = New System.Windows.Forms.TextBox
            Me.StatusGroupBox = New System.Windows.Forms.GroupBox
            Me.MainStatusBar = New System.Windows.Forms.StatusBar
            CType(Me.FingerPrintPictureBox,
System.ComponentModel.ISupportInitialize).BeginInit()
            CType(Me.statusBarPanel1,
System.ComponentModel.ISupportInitialize).BeginInit()
            CType(Me.statusBarPanel2,
System.ComponentModel.ISupportInitialize).BeginInit()
            Me.StatusGroupBox.SuspendLayout()
            Me.SuspendLayout()
            '
            'EnrollmentButton
            '
            Me.EnrollmentButton.FlatStyle =
System.Windows.Forms.FlatStyle.System
            Me.EnrollmentButton.Location = New
System.Drawing.Point(189, 187)
            Me.EnrollmentButton.Name = "EnrollmentButton"
            Me.EnrollmentButton.Size = New System.Drawing.Size(176, 45)
            Me.EnrollmentButton.TabIndex = 2
            Me.EnrollmentButton.Text = "ลงทะเบียน"
            '
            'FingerPrintPictureBox
            '
            Me.FingerPrintPictureBox.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D
            Me.FingerPrintPictureBox.Location = New
System.Drawing.Point(215, 12)
            Me.FingerPrintPictureBox.Name = "FingerPrintPictureBox"
            Me.FingerPrintPictureBox.Size = New
System.Drawing.Size(128, 169)
            Me.FingerPrintPictureBox.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
            Me.FingerPrintPictureBox.TabIndex = 5
            Me.FingerPrintPictureBox.TabStop = False
            '
            'statusBarPanel1
            '
            Me.statusBarPanel1.AutoSize =
System.Windows.Forms.StatusBarPanelAutoSize.Contents
            Me.statusBarPanel1.Name = "statusBarPanel1"
            Me.statusBarPanel1.Text = "version 2.2.0.1"
            Me.statusBarPanel1.Width = 88
            '
            'statusBarPanel2
            '
            Me.statusBarPanel2.AutoSize =
System.Windows.Forms.StatusBarPanelAutoSize.Spring
            Me.statusBarPanel2.Name = "statusBarPanel2"
            Me.statusBarPanel2.Text = "www.itworksolutions.com"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Me.statusBarPanel2.Width = 455
            '
            'StatusTextBox
            '
            Me.StatusTextBox.Location = New System.Drawing.Point(8, 16)
            Me.StatusTextBox.Multiline = True
            Me.StatusTextBox.Name = "StatusTextBox"
            Me.StatusTextBox.RightToLeft =
System.Windows.Forms.RightToLeft.No
            Me.StatusTextBox.ScrollBars =
System.Windows.Forms.ScrollBars.Both
            Me.StatusTextBox.Size = New System.Drawing.Size(528, 248)
            Me.StatusTextBox.TabIndex = 16
            '
            'StatusGroupBox
            '
            Me.StatusGroupBox.Controls.Add(Me.StatusTextBox)
            Me.StatusGroupBox.FlatStyle =
System.Windows.Forms.FlatStyle.System
            Me.StatusGroupBox.Location = New System.Drawing.Point(8,
248)
            Me.StatusGroupBox.Name = "StatusGroupBox"
            Me.StatusGroupBox.Size = New System.Drawing.Size(544, 273)
            Me.StatusGroupBox.TabIndex = 4
            Me.StatusGroupBox.TabStop = False
            Me.StatusGroupBox.Text = " Status "
            '
            'MainStatusBar
            '
            Me.MainStatusBar.Location = New System.Drawing.Point(0,
530)
            Me.MainStatusBar.Name = "MainStatusBar"
            Me.MainStatusBar.Panels.AddRange(New
System.Windows.Forms.StatusBarPanel() {Me.statusBarPanel1,
Me.statusBarPanel2})
            Me.MainStatusBar.ShowPanels = True
            Me.MainStatusBar.Size = New System.Drawing.Size(560, 22)
            Me.MainStatusBar.TabIndex = 6
            '
            'MainForm
            '
            Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
            Me.ClientSize = New System.Drawing.Size(560, 552)
            Me.Controls.Add(Me.MainStatusBar)
            Me.Controls.Add(Me.FingerPrintPictureBox)
            Me.Controls.Add(Me.StatusGroupBox)
            Me.Controls.Add(Me.EnrollmentButton)
            Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedSingle
            Me.MaximizeBox = False
            Me.MinimizeBox = False
            Me.Name = "MainForm"
            Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
            Me.Text = "ทางเข้า"
            CType(Me.FingerPrintPictureBox,
System.ComponentModel.ISupportInitialize).EndInit()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CType(Me.statusBarPanel1,
System.ComponentModel.ISupportInitialize).EndInit()
            CType(Me.statusBarPanel2,
System.ComponentModel.ISupportInitialize).EndInit()
            Me.StatusGroupBox.ResumeLayout(False)
            Me.StatusGroupBox.PerformLayout()
            Me.ResumeLayout(False)

        End Sub

        <STAThread()> _
        Shared Sub Main()
            Application.Run(New MainForm)
        End Sub

        'เพิ่มการทำงานแบบอัตโนมัติ ของ Activate และ LoadData
        Private Sub SampleCSharp_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
            Dim a As New ExitForm
            a.Show()

            Dim b As New DataForm
            b.Show()

            _FingerPrintConnection = New
OleDbConnection(_FINGERPRINT_CONNECTION_STRING)
            IslWrapper.InitModule()
            Try
                If IslResultTypes.OK = IslWrapper.IsModuleInited Then
                    Dim sensor_count As Integer = 0
                    Dim result_code As IslResultTypes =
IslWrapper.GetDeviceCount(sensor_count)
                    If sensor_count > 0 Then
                        Dim sensor_list As IslWrapper.DeviceInfo()
                        If IslResultTypes.OK =
IslWrapper.GetDeviceList(sensor_list, sensor_count) Then
                            'SensorSerialComboBox.Items.Clear()
                            Dim i As Integer = 0
                            While i < sensor_count

'SensorSerialComboBox.Items.Add(sensor_list(i).SensorID.ToString)

System.Math.Min(System.Threading.Interlocked.Increment(i), i - 1)
                            End While
                            'SensorSerialComboBox.SelectedIndex = 0
                        End If
                    End If
                End If
            Finally
                IslWrapper.FinalizeModule()
            End Try
            IdiWrapper.InitModule()
            FpeWrapper.InitModule()

            ' เพิ่มการทำงานแบบอัตโนมัติ
            ActivateSensor()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LoadAllFingerPrintInfo()


        End Sub

        ' เพิ่มมา เพิ่มโหลดข้อมูลอัตโนมัติ
        Private Sub LoadAllFingerPrintInfo()
            IdiWrapper.fp_ClearAll(_IdiContext)
            Dim fingerprint_dataset As DataSet = New DataSet
            Dim fingerprint_adapter As OleDbDataAdapter = New
OleDbDataAdapter
            Dim fingerprint_command_builder As OleDbCommandBuilder =
New OleDbCommandBuilder(fingerprint_adapter)
            fingerprint_adapter.SelectCommand = New
OleDbCommand("SELECT * FROM IDIFINGERPRINT ORDER BY IndexValue",
_FingerPrintConnection)
            fingerprint_adapter.Fill(fingerprint_dataset)
            For Each each_row As DataRow In
fingerprint_dataset.Tables(0).Rows
                Dim fp_info As FingerprintInfo = New FingerprintInfo
                LoadFingerPrintInfo(each_row, fp_info)
                IdiWrapper.fp_Add(_IdiContext, fp_info)
            Next
            Me.ShowStatus("Load all finger print info success")
        End Sub

        Private Sub MainForm_Closed(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Closed
            FpeWrapper.CloseContext(Convert.ToUInt32(_FpeContext))
            FpeWrapper.FinalizeModule()
            IdiWrapper.CloseContext(Convert.ToUInt32(_IdiContext))
            IdiWrapper.FinalizeModule()
        End Sub

        ' เพิ่มมา เพื่อ Activate Sensor อัตโนมัติ
        Private Sub ActivateSensor()
            Dim result_code As IdiResultTypes
            If 0 = Convert.ToInt64(_IdiContext) Then
                result_code = IdiWrapper.CreateContext(_IdiContext)
                ShowStatus("Create context for IDI module : " +
result_code.ToString)
            End If
            If Not (0 = Convert.ToInt64(_IdiContext)) Then
                'result_code = IdiWrapper.ActivateSensor(_IdiContext,
SensorTypes.UareU, SensorSerialComboBox.Text,
ApplicationSerialTextBox.Text, ApplicationKeyTextBox.Text)
                result_code = IdiWrapper.ActivateSensor(_IdiContext,
SensorTypes.UareU, sensorID, serialID, regisID)
                If Not (IdiResultTypes.OK = result_code) Then
                    Me.ShowStatus("Activation for IDI module fail : " +
Me.IdiRCodeToString(result_code))
                Else
                    Me.ShowStatus("Activation for IDI module success")
                    AddHandler IdiWrapper.OnIdentEvent, AddressOf
IdiWrapper_OnIdentEvent
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Me.ShowStatus("Register Ident callback procedure :
" + Me.IdiRCodeToString(result_code))
                    AddHandler IdiWrapper.OnImageCapture, AddressOf
IdiWrapper_OnImageCapture
                    Me.ShowStatus("Register sensor callback procedure :
" + Me.IdiRCodeToString(result_code))
                    result_code =
IdiWrapper.StartCapturing(_IdiContext, sensorID)
                    Me.ShowStatus("Start capturing : " +
Me.IdiRCodeToString(result_code))
                    'ActivateSensorButton.Enabled = False
                    'LoadAllFingerPrintInfoButton.Enabled = True
                    Me.ShowStatus("Activation for IDI module success")
                End If
            End If
        End Sub


        Private Sub ActivateSensorButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
            Dim result_code As IdiResultTypes
            If 0 = Convert.ToInt64(_IdiContext) Then
                result_code = IdiWrapper.CreateContext(_IdiContext)
                ShowStatus("Create context for IDI module : " +
result_code.ToString)
            End If
            If Not (0 = Convert.ToInt64(_IdiContext)) Then
                'result_code = IdiWrapper.ActivateSensor(_IdiContext,
SensorTypes.UareU, SensorSerialComboBox.Text,
ApplicationSerialTextBox.Text, ApplicationKeyTextBox.Text)
                result_code = IdiWrapper.ActivateSensor(_IdiContext,
SensorTypes.UareU, sensorID, serialID, regisID)
                If Not (IdiResultTypes.OK = result_code) Then
                    Me.ShowStatus("Activation for IDI module fail : " +
Me.IdiRCodeToString(result_code))
                Else
                    Me.ShowStatus("Activation for IDI module success")
                    AddHandler IdiWrapper.OnIdentEvent, AddressOf
IdiWrapper_OnIdentEvent
                    Me.ShowStatus("Register Ident callback procedure :
" + Me.IdiRCodeToString(result_code))
                    AddHandler IdiWrapper.OnImageCapture, AddressOf
IdiWrapper_OnImageCapture
                    Me.ShowStatus("Register sensor callback procedure :
" + Me.IdiRCodeToString(result_code))
                    result_code =
IdiWrapper.StartCapturing(_IdiContext, sensorID)
                    Me.ShowStatus("Start capturing : " +
Me.IdiRCodeToString(result_code))
                    'ActivateSensorButton.Enabled = False
                    'LoadAllFingerPrintInfoButton.Enabled = True
                    Me.ShowStatus("Activation for IDI module success")
                End If
            End If
        End Sub

        'แก้ การบันทึก EmpID โดยการ Random ตัวเลข
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Private Sub EnrollmentButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
EnrollmentButton.Click
            Dim idi_result As IdiResultTypes
            Dim fpe_result As FpeResultTypes
            If Not (0 = Convert.ToInt64(_IdiContext)) Then
                IdiWrapper.StopCapturing(_IdiContext, sensorID)
                IdiWrapper.DeactivateSensor(_IdiContext,
SensorTypes.UareU, sensorID)
            End If
            If 0 = Convert.ToInt64(_FpeContext) Then
                fpe_result = FpeWrapper.CreateContext(_FpeContext)
                Me.ShowStatus("Create context for FPE module : " +
Me.FpeRCodeToString(fpe_result))
            End If
            If Not (0 = Convert.ToInt64(_FpeContext)) Then
                Try
                    fpe_result = FpeWrapper.ActivateSensor(_FpeContext,
SensorTypes.UareU, sensorID, serialID, regisID)
                    If Not (FpeResultTypes.OK = fpe_result) Then
                        Me.ShowStatus("Activation for FPE module fail :
" + Me.FpeRCodeToString(fpe_result))
                    Else
                        Me.ShowStatus("Activation for FPE module
success")
                        Dim
raw_data(Convert.ToInt32(FpeWrapper.MaximumRawDataSize)) As Byte
                        Dim index As Integer = 0
                        While index <
Convert.ToInt32(FpeWrapper.MaximumRawDataSize)
                            raw_data(index) = 0

System.Math.Min(System.Threading.Interlocked.Increment(index), index -
1)
                        End While
                        Dim raw_data_length As UInt32
                        Dim indexing_value As Integer = 0
                        Dim finger_index As Integer = 4
                        fpe_result =
FpeWrapper.StartEnrolment(_FpeContext, finger_index, raw_data,
FpeWrapper.MaximumRawDataSize, raw_data_length, indexing_value)
                        If Not (FpeResultTypes.OK = fpe_result) Then
                            Me.ShowStatus("Enrollment fail : " +
Me.FpeRCodeToString(fpe_result))
                        Else
                            If
raw_data_length.CompareTo(Convert.ToUInt32(0)) > 0 Then
                                'Dim input_id As Integer =
Convert.ToInt32(Interaction.InputBox("Please enter ID", "IDWORKS Sample
VB.NET", "999", Me.Location.X + 105, Me.Location.Y + 215), 10)
                                ' Random ตัวเลข เพิ่มเก็บใน EmpID ของข้อมูล
                                Dim r As Random = New Random
                                Dim input_id As Integer = r.Next(1,
999)
                                Dim fingerprint_dataset As DataSet =
New DataSet
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                Dim fingerprint_adapter As
OleDbDataAdapter = New OleDbDataAdapter
                                Dim fingerprint_command_builder As
OleDbCommandBuilder = New OleDbCommandBuilder(fingerprint_adapter)
                                fingerprint_adapter.SelectCommand = New
OleDbCommand("SELECT * FROM IDIFINGERPRINT WHERE (1=0)",
_FingerPrintConnection)

fingerprint_adapter.Fill(fingerprint_dataset)
                                Dim new_row As DataRow =
fingerprint_dataset.Tables(0).NewRow
                                Dim output_template As String = ""
                                Dim
buffer(Convert.ToInt32(raw_data_length)) As Byte
                                Dim index2 As Integer
                                While index2 <
Convert.ToInt32(raw_data_length)
                                    buffer(index2) = raw_data(index2)
                                    output_template +=
buffer(index2).ToString("X2")

System.Math.Min(System.Threading.Interlocked.Increment(index2), index2
- 1)
                                End While
                                SaveFingerprintInfo(input_id,
finger_index, indexing_value, buffer, new_row)

fingerprint_dataset.Tables(0).Rows.Add(new_row)

fingerprint_adapter.Update(fingerprint_dataset)
                                Dim fp_info As FingerprintInfo = New
FingerprintInfo
                                fp_info.Tag = input_id
                                fp_info.FingerIndex = finger_index
                                fp_info.IndexingValue = indexing_value
                                fp_info.FingerData = CType(buffer,
Object)
                                idi_result =
IdiWrapper.fp_Add(_IdiContext, fp_info)
                                ShowStatus("Enrollment success : Input
ID = " + fp_info.Tag.ToString)
                                ShowStatus("Indexing Value = " +
fp_info.IndexingValue.ToString)
                                ShowStatus("Template Size = " +
raw_data_length.ToString)
                                ShowStatus("Template : " +
output_template)
                            End If
                        End If
                    End If
                Finally
                    fpe_result = FpeWrapper.CloseContext(_FpeContext)
                    _FpeContext = Convert.ToUInt32(0)
                End Try
            End If
            If Not (0 = Convert.ToInt64(_IdiContext)) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                idi_result = IdiWrapper.ActivateSensor(_IdiContext,
SensorTypes.UareU, sensorID, serialID, regisID)
                idi_result = IdiWrapper.StartCapturing(_IdiContext,
sensorID)
                Me.ShowStatus("Start capturing : " +
Me.IdiRCodeToString(idi_result))
            End If
        End Sub

        Private Sub LoadAllFingerPrintInfoButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
            IdiWrapper.fp_ClearAll(_IdiContext)
            Dim fingerprint_dataset As DataSet = New DataSet
            Dim fingerprint_adapter As OleDbDataAdapter = New
OleDbDataAdapter
            Dim fingerprint_command_builder As OleDbCommandBuilder =
New OleDbCommandBuilder(fingerprint_adapter)
            fingerprint_adapter.SelectCommand = New
OleDbCommand("SELECT * FROM IDIFINGERPRINT ORDER BY IndexValue",
_FingerPrintConnection)
            fingerprint_adapter.Fill(fingerprint_dataset)
            For Each each_row As DataRow In
fingerprint_dataset.Tables(0).Rows
                Dim fp_info As FingerprintInfo = New FingerprintInfo
                LoadFingerPrintInfo(each_row, fp_info)
                IdiWrapper.fp_Add(_IdiContext, fp_info)
            Next
            Me.ShowStatus("Load all finger print info success")
        End Sub

        'check รายนิ้วมือ เพื่อส่งข้อมูลไปคำนวณ
        Private Sub IdiWrapper_OnIdentEvent(ByVal e As
Itworks.Idworks.IdentEventArgs)
            Me.ShowStatus("----------")



            Select Case e.IdentMessage.MsgID
                '0 indicate that IDWORKS cat match this finger with
fingerprint data in memory
                '(add fingerprint data to memory via idi_fp_Add
function)
                'and pass fingerprint info (include tag) via
IdentEventMsg.
                Case 0
                    'Dim str As String = CalTime(e.IdentMessage.Tag)

                    'Try
                    If e.IdentMessage.SensorInfo.SensorID = sensorID
Then
                        Me.ShowStatus("Sensor ID = " +
e.IdentMessage.SensorInfo.SensorID)
                        Me.ShowStatus("O - Found  Tag = " &
e.IdentMessage.Tag & ", Matching Score : " &
e.IdentMessage.QualityScore.ToString) ' & " Time = ") '& str)
                        CalTime(e.IdentMessage.Tag)
                    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    'Catch ex As Exception
                    'MsgBox(ex.Message)
                    'End Try

                    '1 indicate that IDWORKS can not match this finger
with any fingerprint data in memory.
                Case 1
                    If e.IdentMessage.SensorInfo.SensorID = sensorID
Then
                        Me.ShowStatus("X - Not Found")
                        MsgBox("ไม่มีในฐานข้อมูล กรุณาลองใหม่")
                    End If

            End Select

            Me.ShowStatus("-----------")
        End Sub

        ' คำนวณเวลา และค่าจอด
        Private Sub CalTime(ByVal tag As String)
            Dim textTime As String
            Dim textCost As String
            Dim textRemain As String

            Dim timeStop As New Date
            timeStop = Date.Now.ToLongTimeString
            Dim fingerprint_dataset As DataSet = New DataSet
            Dim fingerprint_adapter As OleDbDataAdapter = New
OleDbDataAdapter
            Dim fingerprint_command_builder As OleDbCommandBuilder =
New OleDbCommandBuilder(fingerprint_adapter)
            fingerprint_adapter.SelectCommand = New
OleDbCommand("SELECT * FROM IDIFINGERPRINT WHERE EmpID = '" & tag &
"'", _FingerPrintConnection)
            fingerprint_adapter.Fill(fingerprint_dataset)

            Dim timeStartStr As String
            timeStartStr =
fingerprint_dataset.Tables(0).Rows(0).Item("TimeStart")

            Dim timeStart As Date = CDate(timeStartStr)


            Me.ShowStatus("เวลาเข้า = " & timeStartStr & " เวลาออก = " &
timeStop)
            textTime = "เวลาเข้า = " & timeStartStr & " เวลาออก = " & timeStop

            'Dim timeStart As New Date(timeStartDate)
            Dim remainHour As Integer = DateDiff(DateInterval.Hour,
timeStart, timeStop)
            Dim remainMinute As Integer = DateDiff(DateInterval.Minute,
timeStart, timeStop)

            Dim cost As Integer
            If remainMinute > 59 Then
                remainMinute = remainMinute - 60 * (remainHour)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Me.ShowStatus(remainHour & " ชั่วโมง " & remainMinute & "
นาที")
                textRemain = remainHour & " ชั่วโมง " & remainMinute & "
นาที"

                If remainMinute > 30 Then
                    remainHour += 1
                End If
                cost = 20 * remainHour
            Else
                Me.ShowStatus(remainHour & " ชั่วโมง " & remainMinute & "
นาที")
                textRemain = remainHour & " ชั่วโมง " & remainMinute & "
นาที"

                cost = 20
            End If

            Me.ShowStatus("cost (20฿ /Hour) : " & cost)
            textCost = "(20฿ /Hour) : " & cost

            ShowMsgBox(textTime, textRemain, textCost)

            _FingerPrintConnection.Open()
            fingerprint_adapter.DeleteCommand = New
OleDbCommand("DELETE * FROM IDIFINGERPRINT WHERE EmpID = '" & tag &
"'", _FingerPrintConnection)
            fingerprint_adapter.DeleteCommand.ExecuteNonQuery()
            _FingerPrintConnection.Close()

        End Sub

        Private Sub IdiWrapper_OnImageCapture(ByVal e As
ImageCaptureEventArgs)
            FingerPrintPictureBox.Image = e.FingerImage
            FingerPrintPictureBox.Show()
        End Sub

        ' เพิ่มการบันทึกเวลาลงฐานข้อมูล
        Private Sub SaveFingerprintInfo(ByVal tag As Integer, ByVal
fingerIndex As Integer, ByVal indexValue As Integer, ByVal buffer As
Byte(), ByVal fingerprintRow As DataRow)
            fingerprintRow("EmpID") = tag
            fingerprintRow("FingerIndex") = fingerIndex
            fingerprintRow("IndexValue") = indexValue
            fingerprintRow("Features") = buffer
            'เพิ่มเวลา
            Dim time As New Date
            fingerprintRow("DateStart") = Date.Now.ToShortDateString
            fingerprintRow("TimeStart") = Date.Now.ToShortTimeString


        End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Private Sub LoadFingerPrintInfo(ByVal fingerprintRow As
DataRow, ByVal fingerprintInfo As FingerprintInfo)
            fingerprintInfo.Tag =
Convert.ToInt32(fingerprintRow("EmpID"))
            Dim objBuffer As Object = fingerprintRow("Features")
            Try
                fingerprintInfo.FingerIndex =
Convert.ToInt32(fingerprintRow("FINGERINDEX"))
            Catch
                fingerprintInfo.FingerIndex = 0
            End Try
            Try
                fingerprintInfo.IndexingValue =
Convert.ToInt32(fingerprintRow("INDEXVALUE"))
            Catch
                fingerprintInfo.IndexingValue = 0
            End Try
            fingerprintInfo.FingerData = objBuffer
        End Sub

        Private Function IdiRCodeToString(ByVal idiResult As
IdiResultTypes) As String
            Dim result As String = ""
            Select Case idiResult
                Case IdiResultTypes.OK
                    result = "Command success"
                    ' break
                Case IdiResultTypes.GeneralFail
                    result = "General failure"
                    ' break
                Case IdiResultTypes.InitializeFail
                    result = "Initialize failed"
                    ' break
                Case IdiResultTypes.ModuleNotInitialize
                    result = "Module not initialize"
                    ' break
                Case IdiResultTypes.SensorNotFound
                    result = "Sensor not found"
                    ' break
                Case IdiResultTypes.SensorAlreadyActivated
                    result = "Sensor already activate"
                    ' break
                Case IdiResultTypes.InvalidSensorType
                    result = "Invalid sensor type"
                    ' break
                Case IdiResultTypes.InvalidRegistration
                    result = "Invalid registration"
                    ' break
                Case IdiResultTypes.InvalidContext
                    result = "Invalid context"
                    ' break
                Case Else
                    result = "Unknow failure"
                    ' break
            End Select

            Return result
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End Function

        Function FpeRCodeToString(ByVal fpeResult As FpeResultTypes) As
String
            Dim result As String = ""
            Select Case fpeResult
                Case FpeResultTypes.OK
                    result = "Command success"
                    ' break
                Case FpeResultTypes.InvalidContext
                    result = "Invalid context"
                    ' break
                Case FpeResultTypes.GeneralFail
                    result = "General failure"
                    ' break
                Case FpeResultTypes.ModuleNotInitialize
                    result = "Module not initialize"
                    ' break
                Case FpeResultTypes.SensorNotFound
                    result = "Sensor not found"
                    ' break
                Case FpeResultTypes.InternalError
                    result = "Internal error"
                    ' break
                Case Else
                    result = "Unknow error"
                    ' break
            End Select

            Return result
        End Function

        Function IslRCodeToString(ByVal islResult As IslResultTypes) As
String
            Dim result As String = ""
            Select Case islResult
                Case IslResultTypes.OK
                    result = "Command success"
                    ' break
                Case IslResultTypes.ModuleNotInitialize
                    result = "Module not initialize"
                    ' break
                Case IslResultTypes.AlreadyInited
                    result = "Module already initialize"
                    ' break
                Case Else
                    result = "Unknow error"
                    ' break
            End Select

            Return result
        End Function

        Private Sub ShowStatus(ByVal strStatus As String)
            StatusTextBox.Text += strStatus + "" &
Microsoft.VisualBasic.Chr(13) & "" & Microsoft.VisualBasic.Chr(10) & ""
            StatusTextBox.Focus()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            StatusTextBox.SelectionStart = StatusTextBox.Text.Length
            StatusTextBox.SelectionLength = 0
            StatusTextBox.ScrollToCaret()
        End Sub

        ' แสดงผลแบบ MSG Box
        Private Sub ShowMsgBox(ByVal textTime As String, ByVal
textRemain As String, ByVal textCost As String)
            MsgBox("" & textTime & "" & Chr(13) + Chr(10) & "เวลาจอด : " &
textRemain & Chr(13) + Chr(10) & "ค่าจอดรถ" & textCost)
        End Sub

    End Class
End Namespace
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลือจากบุคคลหลายๆท่าน ทางผู้จัดทำขอกราบขอบพระคุณ คุณพ่อ คุณแม่และญาติพี่น้องทุกท่านที่ให้การสนับสนุน และเป็นกำลังใจให้แก่ผู้จัดทำตลอดมา

ขอกราบขอบพระคุณอาจารย์ที่ปรึกษา ผศ.ดร.สุทธิชัย นพนาคีพงษ์ ที่ได้กรุณาให้คำปรึกษาคำแนะนำ พร้อมทั้งคณาจารย์ทุกท่านที่ได้ประสิทธิประสาทความรู้แก่ผู้จัดทำ

ขอขอบคุณ คุณกฤษณ์ ก้อนแก้ว ตลอดจนเพื่อนๆภาคโทรคมนาคมและภาคอื่นๆที่ให้คำแนะนำและช่วยแก้ไขทฤษฎีในส่วนที่ผิดพลาดตลอดจนให้กำลังใจจนปริญญานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายผู้จัดทำหวังว่าปริญญานิพนธ์นี้จะเป็นประโยชน์ต่อผู้อ่านบ้างไม่มากก็น้อย

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. จำลอง ครูอุตสาหะ “VB.NET ฉบับโปรแกรมเมอร์”, เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2545.
2. ชาญชัย อุทาหรณ์, สามารถ วาฑิตศุภผล และ อรชุน ข่ายแก้ว “โทรศัพท์แบบใช้ลายนิ้วมือตรวจสอบก่อนโทร”, ปริญญานิพนธ์ (วศ.บ. (วิศวกรรมโทรคมนาคม)) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2545.
3. ธนิต ฉิมคง และ ศานุพงษ์ จันทร์หอม “ที่จอดรถระบบอัตโนมัติ”, ปริญญานิพนธ์ (วศ.บ. (วิศวกรรมอิเล็กทรอนิกส์)) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2544.
4. ธีรบูลย์ หล่อวิเชียรรุ่ง “ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C = MCS-51”, อินโนเวตีฟ เอ็กเพอริเมนต์, 2548.
5. นคร ภักดีชาติม “การทดลองและใช้งานไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C ฉบับ AT89Cx051”, อินโนเวตีฟ เอ็กเพอริเมนต์, 2521.
6. ศุภชัย สมพานิช “Advanced Database Programming ด้วย VB 6+ VB.Net”, ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์, 2548.
7. อุดม รานอก “ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51”, ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์, 2548.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้