สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบนำทางลานจอดรถอัจฉริยะ

INTELLIGENT PARKING GUIDANCE INFORMATION SYSTEM

เมธัส กิตติธรกุล
เมธา ชินศรี
วราภรณ์ ธรรมพัฒน์พงศ์

เลขหมู่.....
เลขทะเบียน.....119444
วัน,เดือน,ปี.....- 7 S.A. 2554

b.....
i.....

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบนำทางลานจอดรถอัจฉริยะ

INTELLIGENT PARKING GUIDANCE INFORMATION SYSTEM

นายเมธัส	กิตติธรกุล	รหัส 50011234
นายเมธา	ชินศรี	รหัส 50011235
นางสาววราภรณ์	ธรรมพัฒน์พงศ์	รหัส 50011392

อาจารย์ที่ปรึกษา

รศ.ดร.สุรพันธ์ เอื้อไพบูลย์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTELLIGENT PARKING GUIDANCE INFORMATION SYSTEM

Maytus Kittithornkun

Maetha Chinsre

Waraporn Thampatpong

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN ELECTRONICS ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2553

ภาควิชาอิเลกทรอนิกส์

คณะวิศวกรรมศาสตร์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบนำทางลานจอดรถอัจฉริยะ

ผู้จัดทำ

1.นายเมธัส กิตติธรกุล

2.นายเมธา ชินศรี

3.นางสาววราภรณ์ ธรรมพัฒศน์พงศ์

..... อาจารย์ที่ปรึกษา

(รศ.ดร.สุรพันธ์ เอื้อไพบูลย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบนำทางลานจอดรถอัจฉริยะ

เมธัส	กิตติธรกุล	รหัสประจำตัว 50011234
เมธา	ชินศรี	รหัสประจำตัว 50011235
วราภรณ์	ธรรมพัฒน์พงศ์	รหัสประจำตัว 50011392
รศ.ดร.สุรพันธ์	เอื้อไพบูลย์	อาจารย์ที่ปรึกษา
		ปีการศึกษา 2553

บทคัดย่อ

โครงงาน ระบบนำทางลานจอดรถยนต์อัจฉริยะ มีจุดมุ่งหมายเพื่อ ให้ความสะดวกสบายในการให้บริการรถยนต์ที่เข้ามาจอดรถยนต์ในแต่ละวัน และเพื่อเป็นการศึกษาการใช้งาน ไมโครคอลโทรเลอร์เบอร์ Parallax P8X32A-D40 และโปรแกรม Propellor ซึ่งใช้ภาษา Spin ในการเขียน โดยใช้วงจรตรวจจับรถเข้าออก และตรวจจับจุดจอดรถยนต์แต่ละจุด มีการทำงานอัตโนมัติ เมื่อมีรถยนต์วิ่งเข้า-ออกในลานจอดรถยนต์ตัวเซนเซอร์จะตรวจจับรถยนต์และทำการนับจำนวน ถ้ารถยนต์ไปจอดที่จุดจอดรถยนต์จะแสดงผลออกหน้าจอดิสเพลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Intelligent Parking Guidance Information System

Maytus Kittihonkun 50011234

Maetha Chinsre 50011235

Waraporn Thampatpong 50011392

Assoc.Prof .Dr.Surapan Airphaiboon Advisor

Education Year 2010

Abstract

This project on the automatic car park.The propose is comfortable in the car park at one car each day.In order to study to use Microcontroller (Parallax P8X32A-D40) and programs Propellor .Spin the language in writing.

The car detection circuit use for detector and counter car in the car park And detecting the car park, each point is automatic. When a car runs into or out of the car park sensors to detect and to count the number of cars. If there is car in the car park and show on display.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากคำแนะนำและการให้คำปรึกษาที่เป็นประโยชน์อย่างสูงจาก รศ.ดร.สุรพันธ์ เอื้อไพบูลย์ ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้ความรู้ทางด้านวิชาการและแนวคิดในการดำเนินชีวิต ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์และขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณท่านอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้มาโดยตลอด ทางผู้จัดทำจะนำความรู้ที่ได้รับไปใช้ให้เกิดประโยชน์สูงสุด

ขอกราบขอบพระคุณบิดามารดาที่คอยให้กำลังใจ และให้การสนับสนุนทางด้านการศึกษามาโดยตลอด

และสุดท้ายขอขอบคุณเพื่อนๆภาควิชาอิเล็กทรอนิกส์ ทุกคนที่เป็นกำลังใจ และร่วมทุกข์สุขกันมาโดยตลอด อีกทั้งยังให้ความช่วยเหลือในเวลาที่เดือดร้อน

เมธัส กิตติธรกุล
เมธา ชินศรี
วราภรณ์ ธรรมพัฒน์พงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ I

Abstract II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูปภาพ VI

สารบัญตาราง VII

บทที่ 1 บทนำ 1

1.1 วัตถุประสงค์ 1

1.2 บทนำ 1

1.3 ขั้นตอนการดำเนินงาน 2

บทที่ 2 หลักการและทฤษฎี 3

2.1 โครงสร้างของระบบ 3

2.2เซนเซอร์ 3

2.2.1โฟโตไอโอด 4

2.3การสะท้อน 7

2.4วงจรIC ENCODER 7

2.5 ไมโครคอนโทรลเลอร์ 8

2.5.1 คุณสมบัติเด่นของPropeller 8

2.5.2 คุณสมบัติทางเทคนิคของPropeller 10

2.5.3 หลักการทำงานของPropeller 11

2.5.3.1 ค๊อก 11

2.5.3.2 ฮับ 11

2.5.4 คอนโทรลเลอร์บอร์ดET-BASE PX32 V1.0 12

2.5.4.1 คุณสมบัติของMCUP8X32A 13

2.5.4.2 คุณสมบัติของบอร์ดET-BASE PX32 V1.0 14

2.5.4.3 โครงสร้างและบล๊อกไดอะแกรมของMCU P8X32A 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

2.5.4.4 โครงสร้างของบอร์ดET-BASE PX32 V1.0 17

2.5.5 การใช้งานโปรแกรมPropellerเบื้องต้น 20

2.5.6 การต่อสายดาวน์โหลดและการต่อสายสื่อสารRS232 23

2.5.7 ข้อกำหนดที่ควรรู้ในการพัฒนาโปรแกรมด้วยภาษาspinบนTool Propeller 25

2.6 การสื่อสารข้อมูลแบบอนุกรม 31

2.6.1 การแปลงสัญญาณข้อมูลระหว่างแบบอนุกรมและแบบขนาน 31

2.6.2 การส่งข้อมูลแบบอะซิงโครนัส 33

2.6.3 การส่งข้อมูลแบบซิงโครนัส 34

บทที่ 3หลักการออกแบบ 38

3.1 ส่วนประมวลผลหลัก 38

3.1.1 วงจรตรวจจับรถยนต์เข้าออก 38

3.1.1.1 ส่วนประกอบของวงจรเซนเซอร์แสง 39

3.1.1.2 หลักการทำงานวงจรตรวจจับรถยนต์เข้าออก 39

3.1.1.3 การติดตั้งเซนเซอร์แสงนับจำนวนรถที่วิ่งเข้าออก 41

3.1.2 วงจรประมวลผลลานจอดรถ 41

3.1.2.1 ระบบการทำงาน 42

บทที่ 4 การทดลองและผลการทดลอง 43

บทที่ 5สรุปผลการทดลอง 46

ภาคผนวก 47

บรรณานุกรม VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ 2.1 บล๊อกไดอะแกรมของระบบลานจอดรถอัจฉริยะ 3
รูปที่ 2.2 แสดงสัญลักษณ์ และการไบอัสใช้งานของโฟโต้ไดโอด 4
รูปที่ 2.3 โครงสร้างและสัญลักษณ์โฟโตไดโอด 5
รูปที่ 2.4 ส่วนประกอบหลักภายในโฟโตไดโอด 5
รูปที่ 2.5การต่อวงจรทำงานโฟโตไดโอด 6
รูปที่ 2.6 มุมตกกระทบและมุมสะท้อน 7
รูปที่ 2.7 วงจรภายใน IC74165 7
รูปที่ 2.8 บอร์ด ET-BASE PX32 v1.0 13
รูปที่ 2.9 บล๊อกไดอะแกรมของชิพP8X32A 15
รูปที่ 2.10 โครงสร้างของชิพP8X32A(DIP 40 pin) 16
รูปที่ 2.11 โครงสร้างของบอร์ดET-BASE PX32 v1.0 18
รูปที่ 2.12 หน้าต่างโปรแกรมPropeller 20
รูปที่ 2.13 หน้าต่างเมนRunและเลือกcompile current 21
รูปที่ 2.14 หน้าต่างเมนูClock mode 22
รูปที่ 2.15 การเข้าสายดาวน์โหลดและการสื่อสารRS232ระหว่าง บอร์ด PX32 และPC 23
รูปที่ 2.16 การเข้าสายRS232สำหรับรับส่งข้อมูล 24
รูปที่ 2.17 การต่อสายดาวน์โหลดเข้ากับคอมพอร์ตของPCโดยตรง 24
รูปที่ 2.18 การต่อสายดาวน์โหลดโดยผ่านตัวแปลง 25
รูปที่ 2.19 กำหนดค่าคงที่โดยใช้ Con - Block 25
รูปที่ 2.20 การประกาศตัวแปร Var – Block 26
รูปที่ 2.21 การกำหนดชื่อไฟล์จากภายนอก OBJ – Block 26
รูปที่ 2.22 PUB – Block 26
รูปที่ 2.23 การประกาศตัวแปรใช้ภายใน PUB 27
รูปที่ 2.24 Block PRI 27
รูปที่ 2.25 Block DAT 28
รูปที่ 2.26 การใช้งานคำสั่งเกี่ยวกับลูป 29
รูปที่ 2.27 รูปวงจร ET-BASE PX32 V1.0 30
รูปที่ 2.28 การส่งข้อมูลแบบอนุกรม 31
รูปที่ 2.29 การแปลงข้อมูลจากอนุกรมเป็นขนาน 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูปที่ 2.30 แปรงข้อมูลจากขนานเป็นอนุกรม 33
รูปที่ 2.31 รายละเอียดของรูปแบบการส่งข้อมูลแบบซิงโครนัส 34
รูปที่ 2.32 ภาพแสดงการส่งข้อมูลแบบซิงโครนัสในรูปแบบง่าย 35
รูปที่ 2.33 การเข้าจังหวะระหว่างสัญญาณนาฬิกาและข้อมูล 35
รูปที่ 2.34 โครงสร้างคอนเนคเตอร์DB-9 36
รูปที่ 2.35 ระดับแรงดันที่ใช้ในการสือสารข้อมูลRS232 37
รูปที่ 2.36 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ผ่านRS232 37
รูปที่ 3.1 วงจรตรวจจับรถ 38
รูปที่ 3.1.1 วงจรเซนเซอร์แสง 39
รูปที่ 3.2Flow Chart การทำงานวงจรตรวจจับรถยนต์เข้าออก 40
รูปที่ .3.3 เซนเซอร์แสงนับจำนวน 41
รูปที่ 3.4 บล๊อกไดอะแกรมวงจรประมวลผลลานจอดรถ 41
รูปที่ 3.5 Flow Chart การทำงานวงจรประมวลผลลานจอดรถ 42
รูปที่ 4.1 หน้าจอแสดงจำลองผลการทดลอง 45
รูปที่ 4.2 แสดงเหตุการณ์จำลองการทดลอง 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 1.1 แผนการดำเนินงาน 2

ตารางที่ 2.5.4 รายละเอียดPIN 17

ตารางที่ 4.1 การทดลอง 43

ตารางที่ 4.2 ผลการทดลอง 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขั้นตอนการดำเนินโครงงาน

ตารางที่1.1 แผนการดำเนินโครงงาน

หัวข้อ	เทอม 1				เทอม 2				
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ศึกษาการทำงานของคอนโทรลเลอร์ที่ต้องใช้	←	—	→						
2.ทดลองเขียนคำสั่งออกหน้าจอ		←	—	→					
3.ทดลองต่อการทำงาน IC 74HC165N			←	—	→				
4.เขียนโปรแกรมส่วนประมวลผลลานจอดรถ					←	—	→		
5.ทำรายงาน							←	—	→
6.เขียนโปรแกรมส่วนนับรถเข้าออก							←	—	→
7.ทำโมเดลลานจอดรถ								←	— →
8.ทำการทดลองและสรุปผลการทดลอง									← →
9.ตรวจเช็คและแก้ไข									← →

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขั้นตอนการดำเนินโครงงาน

ตารางที่1.1 แผนการดำเนินโครงงาน

หัวข้อ	เทอม 1				เทอม 2				
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ศึกษาการทำงานของคอนโทรลเลอร์ที่ต้องใช้	←	—	→						
2.ทดลองเขียนคำสั่งออกหน้าจอ		←	—	→					
3.ทดลองต่อการทำงาน IC 74HC165N			←	—	→				
4.เขียนโปรแกรมส่วนประมวลผลลานจอดรถ					←	—	→		
5.ทำรายงาน							←	—	→
6.เขียนโปรแกรมส่วนนับรถเข้าออก							←	—	→
7.ทำโมเดลลานจอดรถ								←	— →
8.ทำการทดลองและสรุปผลการทดลอง									↔
9.ตรวจเช็คและแก้ไข									↔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบนำทางลานจอดรถอัจฉริยะ

INTELLIGENT PARKING GUIDANCE INFORMATION SYSTEM

โดย

นาย เมธัส กิตติธรกุล รหัสประจำตัว 5001234

นาย เมธา ชินศรี รหัสประจำตัว 50011235

นางสาววราภรณ์ ธรรมพัฒน์พงศ์ รหัสประจำตัว50011392

อาจารย์ที่ปรึกษา

รศ. ดร. สุรพันธ์ เอื้อไพบูลย์

รายงานนี้เป็นส่วนหนึ่งของวิชา 01044058 PROJECT 2

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

1.1 บทนำ

ในปัจจุบันนี้ รถยนต์ถือได้ว่าเป็นปัจจัยหนึ่งในการดำรงชีวิตของมนุษย์ ที่มนุษย์ใช้เพื่ออำนวยความสะดวกสบายให้กับตนเองและผู้อื่น รถยนต์จึงถูกนำมาใช้งานมากขึ้น ทำให้สถานที่ต่างๆ ทั้งสถานที่ราชการ สถานที่ทำงาน สถานที่ท่องเที่ยว พักผ่อนหย่อนใจ รวมถึงห้างสรรพสินค้านั้น จำเป็นต้องมีที่จอดรถไว้สำหรับบริการลูกค้าหรือพนักงานแม้จะมีการเตรียมที่จอดรถไว้สำหรับบริการลูกค้า หรือพนักงาน แต่ก็ยังเกิดปัญหาคือมีที่จอดรถจำนวนจำกัดไม่เพียงพอสำหรับรถยนต์ บางแห่งจึงมีคนเฝ้าที่จอดรถตลอดเวลา เพื่อที่จะได้ทราบว่ามีช่องว่างเหลือให้จอดรถยนต์หรือไม่ จึงได้มีการพัฒนาโครงงานนี้ขึ้นเพื่อให้ได้ทราบว่าภายในลานจอดรถยนต์นั้นมีที่ว่างหรือไม่โดยไม่ต้องใช้คนเฝ้าใช้แค่ผู้ดูแลเท่านั้น

1.2 วัตถุประสงค์

1. เพื่อศึกษาหลักการทำงานของบอร์ด Propellor
2. เพื่อใช้เป็นเครื่องมือติดตั้งช่วยอำนวยความสะดวกในลานจอดรถที่ต่างๆ
3. เพื่อศึกษาการทำงานของโปรแกรม Propellor และ การใช้ภาษา .spin ในการเขียนคอลโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขั้นตอนการดำเนินโครงงาน

ตารางที่1.1 แผนการดำเนินโครงงาน

หัวข้อ	เทอม 1				เทอม 2				
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ศึกษาการทำงานของคอนโทรลเลอร์ที่ต้องใช้	←	—	→						
2.ทดลองเขียนคำสั่งออกหน้าจอ		←	—	→					
3.ทดลองต่อการทำงาน IC 74HC165N			←	—	→				
4.เขียนโปรแกรมส่วนประมวลผลลานจอดรถ					←	—	→		
5.ทำรายงาน							←	—	→
6.เขียนโปรแกรมส่วนนับรถเข้าออก							←	—	→
7.ทำโมเดลลานจอดรถ								←	— →
8.ทำการทดลองและสรุปผลการทดลอง									↔
9.ตรวจเช็คและแก้ไข									↔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

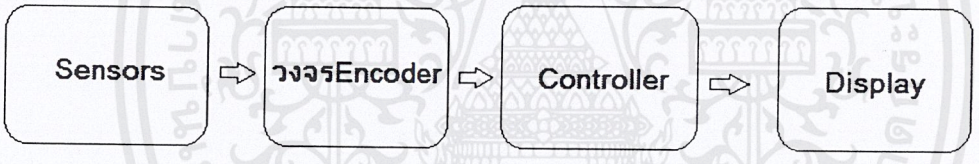
บทที่ 2

หลักการและทฤษฎี

2.1 โครงสร้างของระบบ

เซนเซอร์ที่ทางเข้าและทางออกทำหน้าที่คอยตรวจจับรถยนต์เข้า-ออกแล้วส่งค่าไปที่ส่วนควบคุม และแสดงผลการนับรถเข้า-ออกไปยังหน้าจอ โดยจะมีเซนเซอร์ตรวจจุดจอดรถยนต์แต่ละจุดในแต่ละชั้น คอยส่งค่าไปที่วงจรควบคุม แล้วส่งค่าไปแสดงยังหน้าจอเช่นกัน

ในส่วนของโปรแกรมควบคุมลานจอดรถยนต์จะจัดการในการบันทึกรายละเอียดรถยนต์เข้าและออกพร้อมทั้งรับค่าจากเซนเซอร์จากจุดจอดรถยนต์ในแต่ละจุดไปเก็บค่าต่างๆลงไว้ในฐานข้อมูลเพื่อบันทึก และสามารถแสดงแผนที่จอดรถยนต์ของแต่ละชั้นเพื่อให้สามารถดูได้ว่าจุดจอดรถยนต์ตรงไหนว่างหรือไม่ว่าง



รูปที่ 2.1 บล๊อกไดอะแกรมของระบบลานจอดรถอัจฉริยะ

2.2 เซนเซอร์ (Sensor)

เซนเซอร์ เป็นชุดอุปกรณ์หรือวงจรที่ทำหน้าที่ตรวจจับการเปลี่ยนแปลงของสิ่งแวดล้อม เช่น แสง เสียง ที่อยู่รอบตัวมาเป็นสัญญาณทางไฟฟ้าแล้วส่งไปยังส่วนควบคุมของเครื่องไฟฟ้าหรืออุปกรณ์อื่นๆ เช่น หุ่นยนต์ หากเปรียบเทียบการทำงานกับมนุษย์แล้ว เซนเซอร์ก็เหมือนตา หู หรือผิวหนังของมนุษย์ที่จะคอยรับรู้สภาพแวดล้อมภายนอกนั่นเอง ซึ่งในเบื้องต้นเราสามารถแบ่งชนิดของเซนเซอร์ได้เป็น เซนเซอร์แสง เซนเซอร์เสียง และเซนเซอร์อินฟราเรด

เซนเซอร์ใดๆนั้นจะมีหลักการเหมือนกัน คือต้องมีตัวส่งสัญญาณ และตัวรับสัญญาณ ทำงานโดยนำสัญญาณที่ได้จากตัวรับมาประมวลผล ซึ่งเซนเซอร์ที่ใช้จะเป็นเซนเซอร์แสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

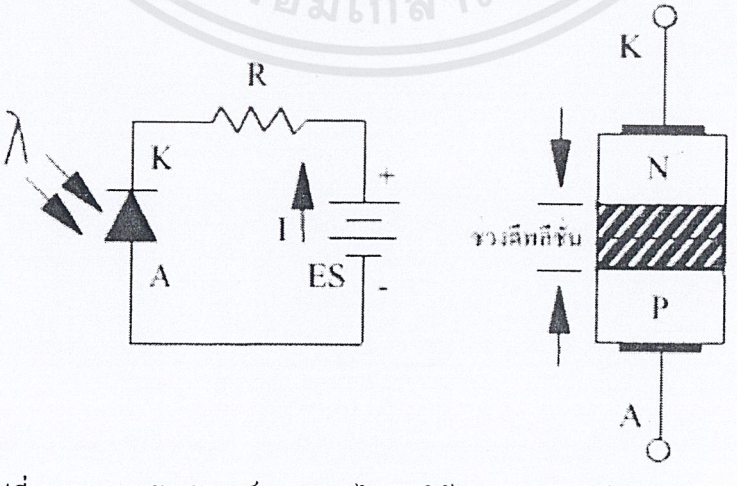
เซ็นเซอร์แสง มีหลายแบบ แต่ละแบบทำงานในลักษณะเดียวกันคือมีตัวส่งแสงอินฟาเรดไปกระทบวัตถุ ถ้าวัตถุนั้นสามารถสะท้อนแสงได้ดีแสงดังกล่าวก็จะมาเป็น อินพุตให้อุปกรณ์รับแสงซึ่งอาจจะเป็น โฟโต้ไดโอด โฟโต้ทรานซิสเตอร์ แอลดีอาร์ หรืออาจจะใช้อุปกรณ์ เซ็นเซอร์แสงแบบสำเร็จก็มีขาย คือมีทั้งตัวส่งและตัวรับในตัวถังเดียวกัน แต่ถ้าวัตถุไม่สะท้อนแสงหรือสะท้อนแสงได้น้อย เช่น วัตถุสีดำ ตัวรับก็จะไม่นำกระแสหรือนำกระแสเพียงเล็กน้อย ผลจากการเปลี่ยนแปลงของเซ็นเซอร์สามารถนำไปเข้าวงจรเปรียบเทียบแรงดันให้เป็นค่าลอจิก ซึ่งจะง่ายกับการนำไปควบคุมงานที่ได้ออกแบบ

เซนเซอร์แสงเป็นอุปกรณ์ที่เปลี่ยนความเข้มแสงให้เป็นสัญญาณทางไฟฟ้า เพื่อส่งสัญญาณให้กับส่วนควบคุมของเครื่องไฟฟ้าหรืออุปกรณ์ทำงานต่างๆ เซนเซอร์แสงนั้นภายในจะประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์

โฟโต้ไดโอด (Photo Diode) เป็นอุปกรณ์เชิงแสงชนิดหนึ่ง ที่ประกอบด้วยสารกึ่งตัวนำชนิด P และสารกึ่งตัวนำชนิด N รอยต่อจะถูกห่อหุ้มด้วยวัสดุที่แสงผ่านได้ เช่น กระจกใส โฟโต้ไดโอดจะมีอยู่ 2 แบบ คือแบบที่ตอบสนองต่อแสงที่เรามองเห็น และแบบที่ตอบสนองต่อแสงในย่านอินฟาเรด ในการรับใช้งานจะต้องต่อโฟโต้ไดโอดในลักษณะไบอัสกลับ

2.2.1 โฟโตไดโอด

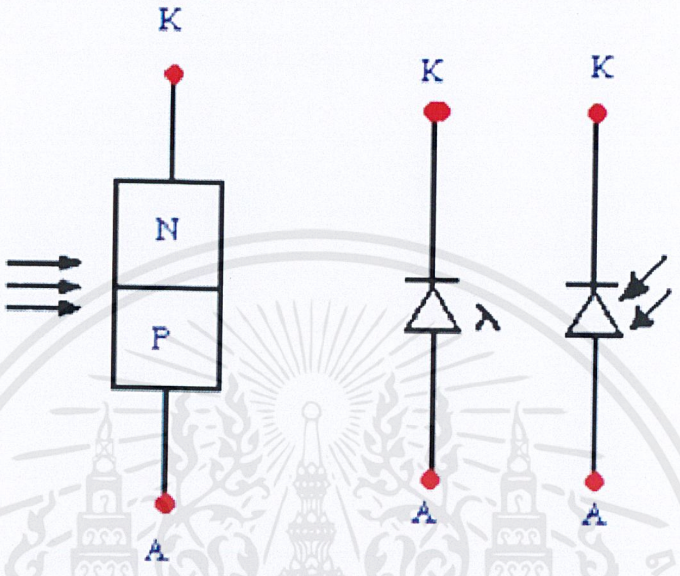
โฟโต้ไดโอด (Photo Diode) จะยอมให้กระแสไหลผ่านได้มากหรือน้อยนั้นขึ้นอยู่กับปริมาณความเข้มของแสง เมื่อโฟโต้ไดโอดได้รับไบอัสกลับ (Reverse Bias) ด้วยแรงดันค่าหนึ่งและมีแสงมาตกกระทบที่บริเวณรอยต่อ ถ้าแสงที่มาตกกระทบมีความยาวคลื่นหรือแลมด้าที่เหมาะสมจะมีกระแสไหลในวงจร โดยกระแสที่ไหลในวงจร จะแปรผกผันกับความเข้มของแสงที่มาตกกระทบ ลักษณะทั่วไปขณะไบอัสตรง (Forward Bias) จะยังคงเหมือนกับไดโอดธรรมดาคือยอมให้กระแสไหลผ่านได้



รูปที่ 2.2 แสดงสัญลักษณ์ และการไบอัสใช้งานของโฟโต้ไดโอด

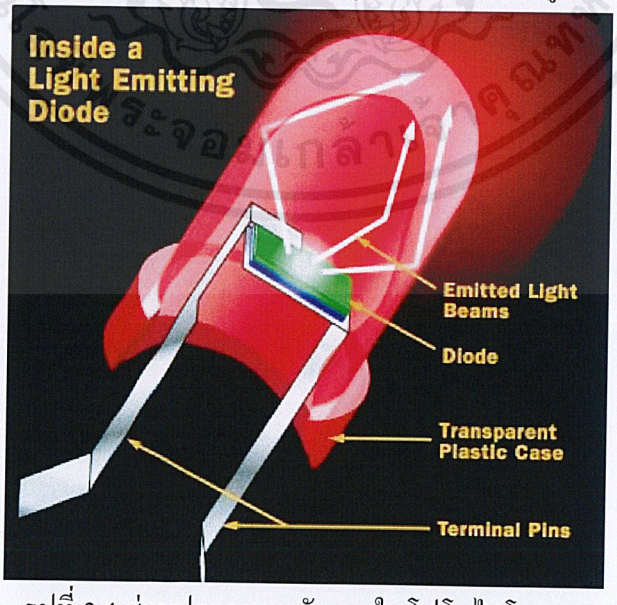
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟโตไดโอด (PHOTODIODE) คือ ไดโอดทำงานด้วยแสง เป็นอุปกรณ์สารกึ่งตัวนำอีกชนิดหนึ่ง ที่ค่าความต้านทานภายในตัวไดโอดเปลี่ยนแปลงมากขึ้นหรือน้อยลง ขึ้นอยู่กับแสงที่ส่องมากระทบกับสารกึ่งตัวนำในตัวไดโอด โฟโตไดโอด มีขาต่อออกมา ใช้งาน 2 ขา คือขาแอโนด (A) และ ขาแคโถด (K)



รูปที่ 2.3 โครงสร้างและสัญลักษณ์โฟโต้ไดโอด

จากรูปที่ 2.3 เป็นโครงสร้างของโฟโตไดโอด ประกอบด้วยสารกึ่งตัวนำ 2 ตอน เหมือนไดโอดธรรมดา เพียงแต่เพิ่มช่องรับแสงให้ผ่านเข้ารอยต่อของโฟโตไดโอด ทำให้สัญลักษณ์ของโฟโตไดโอด เหมือนสัญลักษณ์ของไดโอดธรรมดาและมีตัวแรมดา หรือลูกศรชี้เข้ากำกับอยู่ด้วย แสดงดังรูปที่ 2.4

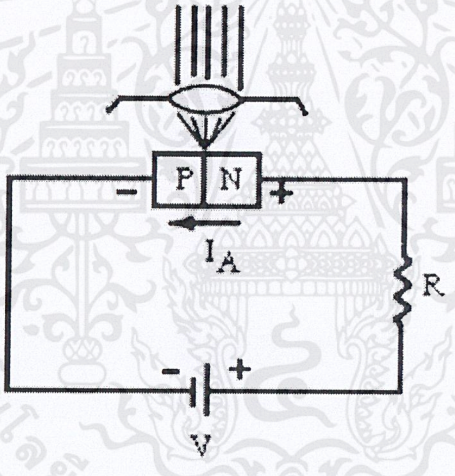


รูปที่ 2.4 ส่วนประกอบหลักภายในโฟโตไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟโต้ไดโอด (Photo diode) อุปกรณ์สารกึ่งตัวนำกระแสได้ก็เนื่องจากการให้พลังงานเพื่อดึงอิเลคตรอนให้หลุดจากบอนด์ เป็นผลทำให้เกิดอิเลคตรอนอิสระและโฮล และเมื่อให้แรงดันไฟฟ้าจะเกิดสนามไฟฟ้าในแท่งสารนั้นเป็นผลทำให้ประจุอิเลคตรองและโฮล เคลื่อนที่ โฟโต้ไดโอดจึงมีหลักการทำงานโดยอาศัยแสงในการเพิ่มพลังงานให้กับอิเลคตรอนในเนื้อสารกึ่งตัวนำ

โดยปกติไดโอดจะถูกไบแอสตรงแต่ในขณะที่ไบแอสตรงนี้ จำนวนอิเลคตรอนและโฮลที่ในเนื้อสารมีจำนวนไม่มากนัก ดังนั้นกระแสที่ไหลในวงจรจึงเป็นส่วนน้อย ครั้นเมื่อส่วนของสารกึ่งตัวนำมีแสงส่องถูก จะทำให้เนื้อสารเกิดอิเลคตรอนอิสระและโฮลเกิดขึ้นเป็นจำนวนมาก จำนวนอิเลคตรอนอิสระที่เกิดขึ้นจะแปรตรงกับความเข้มของแสงแต่เมื่อเพิ่มความเข้มของแสงจนถึงค่าหนึ่งจะไม่มีการเพิ่มของอิเลคตรอนอิสระอีกแล้วในช่วงนี้เราจะเรียกว่า ช่วงอิ่มตัว (saturation region) ในขณะที่ไม่มีแสงตกกระทบจำนวนกระแสที่ไหลผ่านตัวไดโอดนี้เรียกว่า กระแสมืด (dark current)



รูปที่ 2.5 การต่อวงจรทำงานให้โฟโตไดโอด

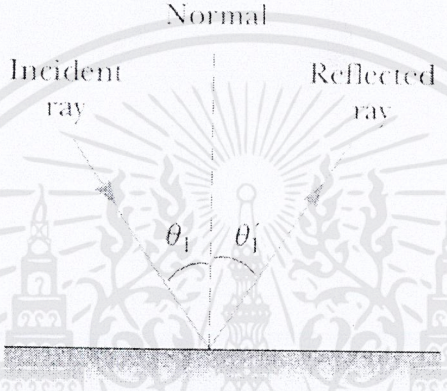
โฟโต้ไดโอดเมื่อเทียบกับ LDR (ตัวต้านทานที่แปรค่าตามแสง) แล้วโฟโต้ไดโอดมีการเปลี่ยนแปลงค่าความต้านทานเร็วกว่า LDR มาก จึงนิยมนำไปประยุกต์งานในวงจรที่ต้องการความเร็วสูง เช่น เครื่องนับสิ่งของ, ตัวรับรีโมทคอนโทรล, วงจรกันขโมยอินฟาเรดเป็นต้น

เนื่องจากโฟโต้ไดโอดให้ค่าการเปลี่ยนแปลงของกระแสต่อแสงต่ำ คืออยู่ในช่วง 1-10 A เท่านั้น ดังนั้นการใช้งานโฟโต้ไดโอดจึงต้องมีตัวขยายกระแสเพิ่มเติม ผู้ผลิตจึงหันมาใช้ทรานซิสเตอร์เป็นตัวขยายกระแสเพิ่มเติมอยู่ในตัวถังเดียวกัน ซึ่งเรียกว่า โฟโต้ทรานซิสเตอร์(Photo Transistor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

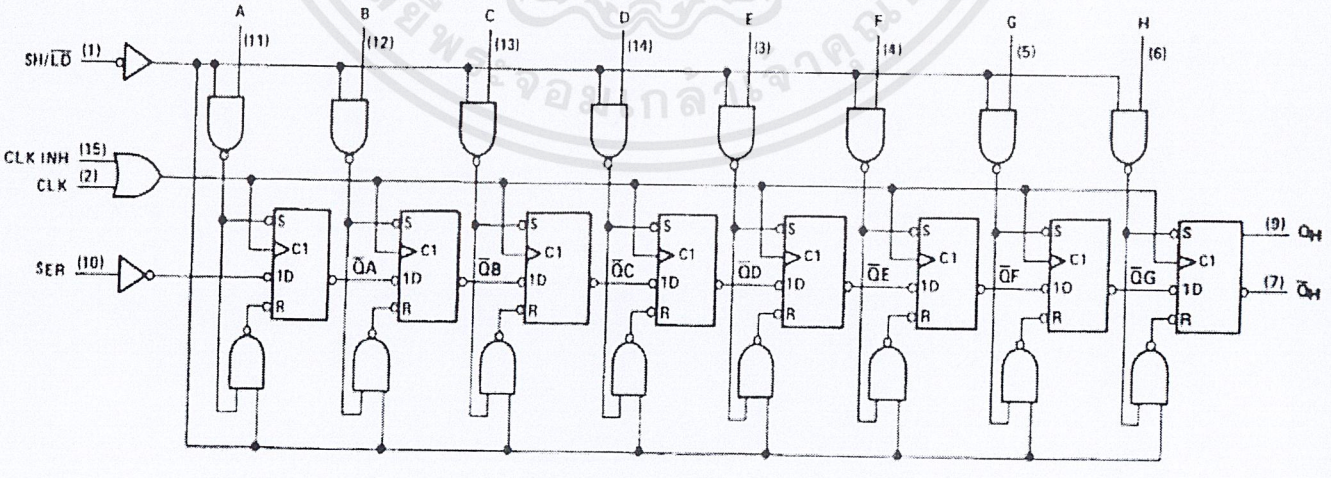
2.3**การสะท้อน (Reflection)**

การสะท้อนเกิดขึ้นเมื่อแสงเดินทางจากตัวกลางหนึ่งไปตกกระทบตัวกลางอื่น และทำให้แสงเคลื่อนที่กลับมาในตัวกลางเดิม ทิศของแสงที่สะท้อนทราบได้จากความสัมพันธ์ **มุมตกกระทบ เท่ากับมุมสะท้อน** โดยที่มุมตกกระทบและมุมสะท้อนวัดโดยเทียบกับเส้นปกติ (Normal) เส้นปกติเป็นเส้นที่ตั้งฉากกับระนาบรอยต่อของตัวกลางทั้งสอง



รูปที่ 2.6 มุมตกกระทบและมุมสะท้อนมีขนาดเท่ากัน

2.4 วงจร IC encoder



รูปที่ 2.7 วงจรภายใน IC 74165

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IC 74165 ทำหน้าที่เป็นตัว Encoder โดยจะรับข้อมูลLogic 0 หรือ Logic 1 จากตัวSensors เข้ามาเป็นแบบขนาน และจะส่งข้อมูลเป็นLogic 0 หรือ Logic 1 เป็นแบบอนุกรมไปสู่ตัวController ทีละ 8 Bit

2.5 ไมโครคอลโทรลเลอร์

Propeller มัลติคอร์ไมโครคอนโทรลเลอร์นวัตกรรมใหม่ของไมโครคอนโทรลเลอร์ 32 บิต

Propeller หรือ **โพรเพลเลอร์** คือชื่อของไมโครคอนโทรลเลอร์ 32 บิตอนุกรมใหม่จาก **Parallax** (www.parallax.com) ที่มีสถาปัตยกรรมที่พิเศษคือ **มีซีพียูภายใน 8 ตัวหรือ 8 ค็อก (cog)** ที่สามารถทำงานแยกจากกันอย่างอิสระหรือร่วมกันทำงานก็ได้ นับเป็นแนวคิดใหม่ที่ร่วมสมัย และนับเป็นการปฏิวัติวงการไมโครคอนโทรลเลอร์ 32 บิตครั้งสำคัญ

การพัฒนาโปรแกรมสำหรับโพรเพลเลอร์ทำได้ด้วยโปรแกรมภาษาใหม่ที่เรียกว่า สปิน(spin) และภาษาแอสเซมบลี โดยภาษาสปินนั้นเป็นภาษาสูงที่มีการทำงานแบบออบเจ็กต์(high-level object-based language)

2.5.1 คุณสมบัติเด่นของโพรเพลเลอร์

- ประกอบด้วย 8ซีพียู หรือเรียกว่า 8 ค็อก ที่สามารถทำงานได้พร้อมกันอย่างเป็นอิสระ โดยมีการควบคุมการใช้ทรัพยากรร่วมกันผ่านทางตัวเชื่อมโยงกลางหรือ central hub
- มีความเร็วในการทำงานสูงและด้วยการทำงานที่เป็นอิสระของแต่ละค็อกทำให้สามารถรองรับการตอบสนองต่อเหตุการณ์ต่างๆ ที่เกิดขึ้นในระบบได้อย่างเร็วเพียงพอ จึงไม่ต้องใช้กระบวนการอินเตอร์รัปต์ช่วยทำให้การเขียนโปรแกรมเพื่อรองรับการทำงานในแต่ละเหตุการณ์ลดความซับซ้อนลงได้อย่างมาก
- มีการใช้สัญญาณนาฬิกาของระบบร่วมกัน ทำให้สามารถอ้างอิงค่าเวลาหลักเดียวกันได้ทำให้การทำงานของแต่ละค็อกมีจังหวะที่สอดคล้องกัน
- ภาษาสปินซึ่งมีลักษณะเป็นโปรแกรมภาษาสูงแบบออบเจ็กต์ได้รับการออกแบบให้ง่ายต่อการเรียนรู้ และสามารถรองรับการทำงานของโพรเพลเลอร์ได้อย่างมีประสิทธิภาพสูงสุด
- ภาษาแอสเซมบลีของโพรเพลเลอร์ได้จัดเตรียมคำสั่งที่ใช้ในการตรวจสอบเงื่อนไข และมีตัวแปรที่ใช้ในการตรวจสอบการทำงานอย่างสมบูรณ์ ทั้งยังสามารถรองรับการทำงานในลักษณะที่ต้องมีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัดสินใจพร้อมๆ กันหลายเงื่อนไขด้วย พร้อมกันนั้นยังมีการคำนึงถึงการลดสัญญาณรบกวนและความเพี้ยนของสัญญาณที่เกิดขึ้นจากการประมวลผลคำสั่งและตัวคำสั่งเองมีรูปแบบการทำงานที่ตรงไปตรงมาชัดเจน ส่งผลให้ผู้พัฒนาโปรแกรมสามารถลดเวลาในการเขียนโปรแกรมลงได้อย่างมาก

- แต่ละค็อกจะประกอบด้วยตัวประมวลผลหรือโปรเซสเซอร์ที่มีการทำงานอย่างเป็นอิสระมีแรม 2 กิโลไบต์ ที่เมื่อกำหนดให้ทำงานเป็นรีจิสเตอร์ 32 บิต จะได้ทั้งสิ้นถึง 512 ตัว มีโมดูลตัวนับความสามารถสูงที่ทำงานร่วมกับเฟสล็อกลูป ทำให้แต่ละค็อกทำงานได้เร็วถึง 80 MHz มีวงจรกำเนิดสัญญาณและส่วนควบคุมพอร์ตอินพุตเอาต์พุตที่เป็นอิสระ
- สัญญาณนาฬิกาของระบบมาได้จาก 3 แหล่งคือ วงจรออสซิลเลเตอร์ RC ภายในเลือกได้ระหว่าง 12 หรือ 20 MHz จากแหล่งกำเนิดสัญญาณนาฬิกาภายนอก และจากการทวีคูณความถี่ของคริสตอลด้วยวงจรเฟสล็อกลูป โดยปกติแล้วจะเลือกใช้คริสตอล 5 MHz แล้วเลือก PLL×16 ทำให้ได้สัญญาณนาฬิกาของระบบที่มีความถี่ 80 MHz ในขณะที่ส่วนเชื่อมโยงกลางจะทำงานด้วยความถี่ที่ลดลงครึ่งหนึ่งของสัญญาณนาฬิกาหลัก
- มีขาพอร์ตอินพุตเอาต์รวม 32 ขา โดยกำหนดให้ใช้ 2 ขาสำหรับต่อหน่วยความจำอีอีพรอมสำหรับเก็บโปรแกรมของผู้ใช้งาน และอีก 2 ขาสำหรับการดาวน์โหลดโปรแกรม สามารถขับกระแสซิงก์และซอร์สสูงสุด 40mA ต่อขา
- โพรเพลเลอร์ใช้หน่วยความจำอีอีพรอมภายนอกในการเก็บโปรแกรมของผู้ใช้งาน ทำให้อายุการใช้งานของตัวชิปจึงไม่ขึ้นกับจำนวนรอบของการลบและโปรแกรมใหม่ของหน่วยความจำโปรแกรม
- การดาวน์โหลดโปรแกรมทำได้ง่ายมากเพียงต่อเข้ากับวงจรเชื่อมต่อพอร์ตอนุกรม อาทิวงจรของไอซี MAX3232 หรือต่อผ่านชิปแปลงสัญญาณพอร์ต USB เป็นพอร์ตอนุกรมอย่าง FT232RL ไม่ต้องการเครื่องโปรแกรมใดๆ เพิ่มเติม
- ด้วยความเร็วในการทำงานที่สูง และมีส่วนกำเนิดสัญญาณภาพที่มากถึง 8 ชุด ทำให้เหมาะสมอย่างมากในการนำโพรเพลเลอร์ไปใช้ในการกำเนิดสัญญาณภาพไม่ว่าจะแสดงผลด้วยจอโทรทัศน์ด้วยสัญญาณวิดีโอ หรือแสดงผลด้วยจอ VGA ด้วยสัญญาณแม่สีแสง นั่นคือพื้นฐานหลักในการสร้างเครื่องเล่นวิดีโอเกม และการสร้างระบบนำเสนอ (presentation) ภาพกราฟิกด้วยไมโครคอนโทรลเลอร์เพียงตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถเชื่อมต่อกับคีย์บอร์ดและเมาส์ได้ และเมื่อรวมกับความสามารถการสร้างสัญญาณภาพได้ จึงสามารถนำ โพรเพลเลอร์ไปสร้างเครื่องคอมพิวเตอร์ขนาดเล็กแบบใช้จอโทรทัศน์เป็นตัวแสดงผลได้
- ใช้ไฟเลี้ยงในย่าน 2.7 ถึง 3.6 V กระแสไฟฟ้าสูงสุดเมื่อขับโหลดเต็มที่คือ 300 mA
- มีตัวถังให้เลือกใช้ 3 แบบคือ DIP 40 ขา, LQFP 44 ขา และ QFN 44 ขา
- มีซอฟต์แวร์ Propeller IDE ที่มีประสิทธิภาพสูง สามารถสร้างระบบไฟล์ที่มีรูปการต่อวงจรได้ ทำให้สามารถนำโค้ดไปถ่ายทอดต่อได้สะดวก และช่วยให้การเรียนรู้ทำได้อย่างสมบูรณ์ครบวงจรและที่สำคัญซอฟต์แวร์แจกฟรี สามารถดาวโหลดเวอร์ชั่นล่าสุดได้ที่ www.parallax.com

2.5.2 คุณสมบัติทางเทคนิคของโพรเพลเลอร์

- เป็นไมโครคอนโทรลเลอร์ที่ภายในประกอบไปด้วยโปรเซสเซอร์ขนาด 32 บิตถึง 8 ชุด
- ทำงานที่แรงดัน 3.3 โวลต์ (2.7V ถึง 3.6V)
- ขาพอร์ตสามารถจ่ายกระแสซิงก์และซอร์สได้ 40 mA ต่อขา และ 100mA ต่อพอรีต (8 ขา)
- มีวงจรกำเนิดสัญญาณนาฬิกาภายใน 12 MHz หรือ 20 kHz เลือกกำหนดค่าได้
- ทำงานด้วยสัญญาณนาฬิกาจากภายนอกความถี่ตั้งแต่ดีซีถึง 80 MHz
- สามารถใช้คริสตอล 4 MHz ถึง 8 MHz ร่วมกับตัวคูณสัญญาณนาฬิกาความถี่สูงสุด 80 MHz
- สามารถเชื่อมต่อคริสตอลภายนอกได้ โดยไม่จำเป็นต้องเชื่อมต่อตัวเก็บประจุ
- หน่วยความจำของทั้งระบบแบ่งเป็นหน่วยความจำอีอีพรอม 32 กิโลไบต์ (KB) และหน่วยความจำ แรม 32 KB
- ในแต่ละโปรเซสเซอร์มีหน่วยจำแรม ตัวละ 2 KB
- การจัดการหน่วยความจำเป็นแบบ 32 บิต
- จำนวนพอร์ตอินพุตเอาต์ 32 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 หลักการทำงานของโพรเพลเลอร์

รูปที่ 1-2 แสดงให้เห็นบล็อกไดอะแกรมการทำงานภายในของโพรเพลเลอร์ ซึ่งประกอบด้วยโปรเซสเซอร์ที่ทำงานแยกกันอิสระถึง 8 ชุด โดยจะเรียกโปรเซสเซอร์เหล่านี้ว่า Cog หมายเลข 0 ถึง 7

2.5.3.1 ค็อก (Cogs)

ในแต่ละค็อกจะประกอบไปด้วยหน่วยความจำแรม 2KB โดยกำหนดเป็นหน่วยความจำแบบ 32 บิต จำนวน 512 ตัว นอกจากนี้ภายในโปรเซสเซอร์แต่ละตัวยังมีโมดูลเคาน์เตอร์แบบพิเศษพร้อมเฟสล็อกลูป 2 ตัว โมดูลสร้างสัญญาณวิดีโอ รีจิสเตอร์พอร์ตเอาต์พุต รีจิสเตอร์กำหนดทิศทางของพอร์ตอินพุตและรีจิสเตอร์ตัวอื่นๆ ซึ่งไม่ได้แสดงให้เห็นในบล็อกไดอะแกรม

ค็อกทั้ง 8 ตัวทำงานด้วยวงจรกำเนิดสัญญาณนาฬิกาหลัก ซึ่งค็อกแต่ละตัวจะอ้างอิงการทำงานกันได้ด้วยสัญญาณนาฬิกา และจะเริ่มต้นทำงานพร้อมกันและใช้ทรัพยากรด้วยกัน ค็อกแต่ละตัวสามารถสั่งให้ทำงานหรือหยุดทำงาน ได้ในขั้นตอนการทำงานของโปรแกรม และสามารถควบคุมให้แต่ละค็อกทำงานไปพร้อมๆ กันได้ โดยจะทำงานเป็นอิสระหรือ เชื่อมโยงถึงกันได้ผ่านหน่วยความจำแรมหลัก (Main RAM) ซึ่งแยกไปต่างหาก

หน่วยความจำภายในค็อกแต่ละตัว เรียกว่า ค็อกแรม (Cog RAM) โดยค็อกแรมจะแบ่งหน่วยความจำเป็นรีจิสเตอร์ขนาด 32 บิตจำนวน 512 ตัวสามารถใช้งานได้อย่างอิสระ ยกเว้น รีจิสเตอร์ 16 ตำแหน่งสุดท้ายซึ่งสงวนไว้สำหรับรีจิสเตอร์ฟังก์ชั่น พิเศษ เช่น รีจิสเตอร์เคาน์เตอร์ , รีจิสเตอร์พอร์ตอินพุตเอาต์พุต เป็นต้น

2.5.3.2 ฮับ(Hub) : ส่วนเชื่อมโยงหลัก

ฮับทำหน้าที่จัดระเบียบการทำงานของระบบทั้งหมด โดยจะยอมให้ค็อกทีละตัวเท่านั้นที่จะติดต่อกับทรัพยากรหลักของระบบ โดยจะหมุนเวียนติดต่อกับค็อกตั้งแต่หมาย 0 ถึง 7 แล้วกลับไปที่หมายเลข 0 ใหม่เป็นลักษณะวนรอบ ส่วนของฮับและระบบบัสของมันทำงานด้วยความเร็วครึ่งหนึ่งของสัญญาณนาฬิกาของทั้งระบบ ทำให้ค็อก 1 ตัวจะติดต่อทุกๆ 16 ไซเกิลของสัญญาณนาฬิกา และใช้เวลา 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไซเกิลเพื่อเอ็กซิคิวซ์คำสั่ง ดังนั้น ฮับจะติดต่อกับค็อกตัวใดตัวหนึ่งได้ อาจใช้คาบเวลาเพียง 7 ไซเกิล หรือนานถึง 22 ไซเกิล เนื่องจากจะต้องรอให้ฮับวนจนครบรอบ

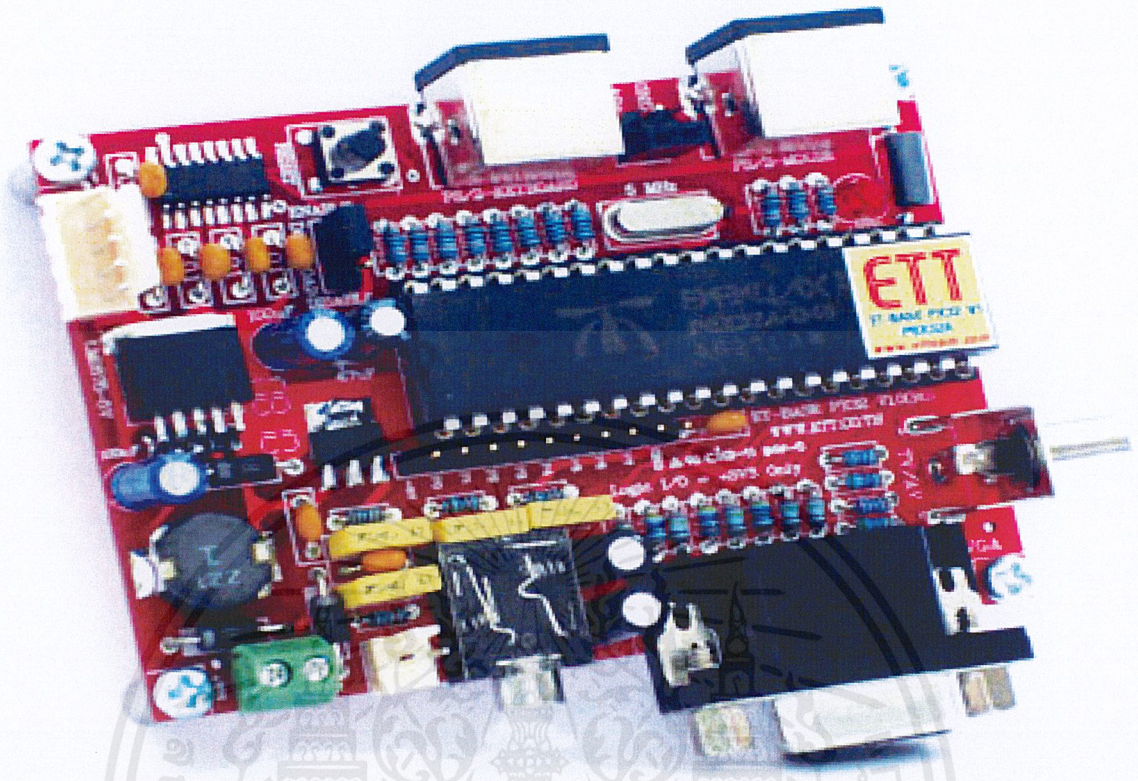
2.5.4คอลโทรลเลอร์บอร์ด ET-BASE PX32 V1.0

Propeller microcontroller Education Board

บอร์ดเรียนรู้และใช้งานไมโครคอนโทรลเลอร์ Propeller

สำหรับบอร์ด ET-BASE PX32 นี้จะเป็นบอร์ด Training โดยใช้ MCU เบอร์ P8X32A –D40 ของ PARALLAX โดยจะเป็น MCU ที่มีความไวสูง ขนาด 32 บิต 8 Cog Multiprocessor โครงสร้างของ MCU จะเป็นตัวถังแบบ DIP 40 PIN สามารถทำงานได้ที่ความถี่สูงสุด 80 MHz ทำงานที่แรงดัน 2.7-3.6 VDC การพัฒนาโปรแกรมจะใช้ Software tool "Propeller V1.06" ซึ่ง Software ตัวนี้สามารถใช้เขียนโปรแกรม, Compile Code และ Download Code ผ่านทาง RS232 ได้เลย(ไม่สามารถ Debug ดูการทำงานเป็น STEP ได้) โดยภาษาที่ใช้ในการเขียนโปรแกรมจะเป็นภาษา SPIN ซึ่งจะช่วยให้พัฒนาโปรแกรมได้ง่ายและรวดเร็วขึ้น เนื่องจากใน โปรแกรม Propeller นี้ จะมี Library ต่างๆสำหรับใช้ติดต่อระหว่างอุปกรณ์รอบข้างกับตัว MCU P8X32A ไว้ให้เรียบร้อยแล้ว ซึ่งเวลาใช้งานผู้ใช้ก็สามารถเรียก Library มาใช้ได้เลย ตัวอย่างเช่น Library RS232 , Library VGA , Library TV , Library Keyboard , Mouse เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 บอร์ด ET-BASE PX32 V1.0

2.5.4.1 คุณสมบัติของ MCU P8X32A

1) เป็น MCU 32 bit 8 Cog Multiprocessor (8 CPU ใน Chip เดียว)

2) ตัวถังแบบ DIP 40 PIN มี Port I/O 32 Pin

3) ตัว MCU ทำงานที่แรงดัน 2.7-3.6 VDC และ I/O Port สามารถขับกระแส Source/Sink ได้ 40 mA ที่ 3.3 VDC

4) ทำงานได้ที่ความถี่สูงสุด 80 MHz สามารถเลือกใช้งาน External Clock หรือ Internal Clock ได้ มี PLL อยู่ภายใน

5) มี RAM ภายในสำหรับเก็บ Code 32 Kbyte ซึ่งเวลาตัดไฟเลี้ยง MCU ออก Code ก็จะถูกลบ ดังนั้นเวลาใช้งานจริง

จะต้องต่อ I2C EEPROM ไว้ภายนอกเพื่อเก็บ Code

6) Pin ที่ถูกกำหนดให้ทำงานเป็น Input จะสามารถรับระดับแรงดัน Input ได้ไม่เกิน VDD(2.7-3.6V) เท่านั้น

7) ความเร็วในการทำงานภายในของ Chip จะอยู่ที่ 20 MIPS/cog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.2 คุณสมบัติของบอร์ด ET- BASE PX32 V1.0

1) ใช้ MCU P8X32A ตัวถังแบบ DIP 40 PIN

2) มี I2C EEPROM #24LC256 (32Kb) สำหรับใช้เก็บ Code Program

3) ตัวบอร์ดรับแรงดัน Input 6-12 VDC โดยมีชุด Regulator ปรับแรงดันลงให้เหลือ 5V และ3.3V อยู่ในบอร์ด

4) ใช้ Crystal 5.00 MHz (External) สามารถใช้ PLL ภายในตัว Chip คูณเพิ่มความถี่ให้สูงขึ้นได้ถึง 80 MHz

5) พัฒนาโปรแกรมด้วยภาษา SPIN โดยใช้ Tool "Propeller" (Freeware) ซึ่ง Tool นี้จะใช้สำหรับเขียนโปรแกรม ,Compile และ Download ภายในตัว และจะมี Library ให้ใช้ในการ Interface ระหว่างตัว MCU กับอุปกรณ์รอบข้างที่จัดไว้ให้บนบอร์ดเรียบร้อยแล้ว และสามารถเข้าไป Download Library เพิ่มเติมได้ที่ http://www.parallax.com/ ในส่วนของการใช้งานภาษา SPIN ก็สามารถดูได้จาก Help ของโปรแกรม Propeller

6) การ Download Code จะ Download ผ่านทาง RS232 โดยสามารถเลือกจากตัว Propeller ได้ว่า จะ Download Code ไปเก็บไว้ยัง EEPROM (External) หรือ RAM ภายใน MCU

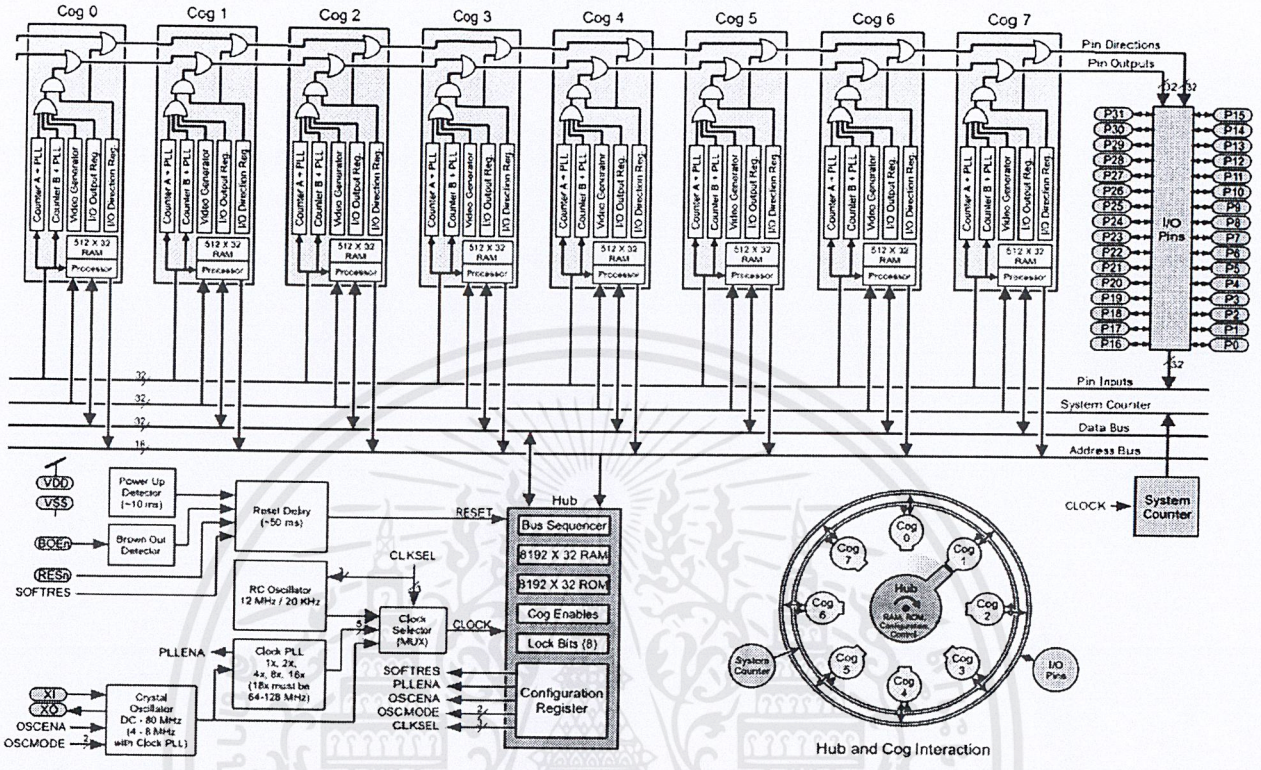
7) บอร์ดนี้ได้จัดสรร I/O ไว้ตายตัวสำหรับ Interface กับอุปกรณ์รอบข้างดังนี้

1 . Port Key Board(PS2) , 2. Port Mouse (PS2) , 3. Port RS232 , 4. Port VGA , 5. Port TV(AV)

6. Port MIC , 7. Port Headphone , 8. Port I/O 8 PIN สำหรับใช้งานด้านอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

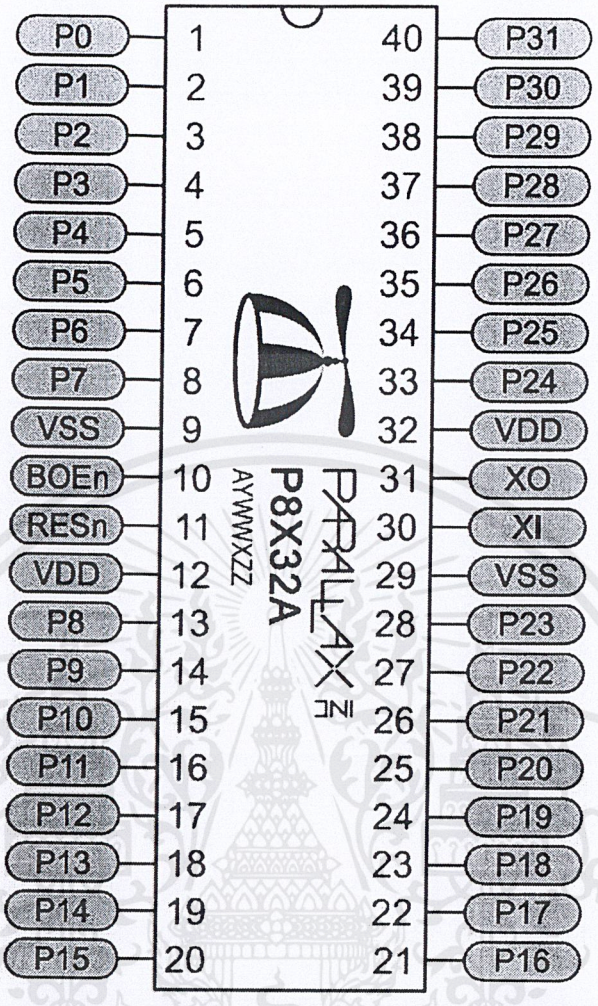
2.5.4.3โครงร้างและบล็อกไดอะแกรมของ MCU P8X32A



รูปที่ 2.9 แสดงบล็อกไดอะแกรมของ Chip P8X32A

จากรูปที่ 2.5.4.2 จะเห็นว่าโครงสร้างภายในของ Chip จะประกอบไปด้วย Processor 8 ตัว โดยจะเรียกว่า “Cog” ซึ่งถูกออกแบบให้มีการทำงานที่ความไวสูง สิ้นเปลืองพลังงานต่ำ ตัวถังมีขนาดเล็ก มีความคล่องตัวในการทำงานผ่านทาง Processorทั้ง 8 ตัวสูง โดยสามารถทำงานได้ในเวลาเดียวกันพร้อมๆกัน และเป็นอิสระต่อกัน ซึ่ง Chip นี้จะใช้การ Share ทรัพยากรผ่านทางศูนย์กลาง HUB เพื่อให้ Cog แต่ละ Cog สามารถใช้งานทรัพยากรร่วมกันได้ ในส่วนของระบบ Clock จะถูก Share ไปยังCog แต่ละ Cog โดยจะอ้างอิงที่ฐานเวลาเดียวกัน ส่วนการ Interrupt จะไม่ถูกใช้กับ Chip นี้ แต่จะใช้การกำหนดตำแหน่งที่จะกระโดดไปทำงานให้กับ Cog ต่างๆโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงโครงสร้างของ Chip P8X32A (DIP 40 PIN)

ในรูปที่ 4 จะเป็นโครงสร้างตัวถังแบบ DIP 40 PIN ของ Chip โดยจำแนก PIN ต่างๆได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่2.5.4 รายละเอียด PIN

PIN Name	Direction	Description
P0-P31	I/O	เป็น Port I/O โดยมีวงจร Logic อยู่ที่ประมาณครึ่งหนึ่งของ VDD หรือ 1.6 VDC ที่แรงดัน 3.3 VDC ใน 32 Pin นี้จะมีอยู่ด้วยกัน 4 Pin ซึ่งจะถูกกำหนดให้ทำงานในหน้าที่พิเศษหลังจาก Power-up หรือ Reset คือ P28 – I2C SCL ซึ่งจะใช้ต่อไปยัง EEPROM ภายนอก P29 – I2C SDA ซึ่งจะใช้ต่อไปยัง EEPROM ภายนอก P30 – Serial Tx ใช้ต่อสำหรับ Download Code และ ส่งข้อมูล ผ่านทาง RS232 P31 – Serial Rx ใช้ต่อสำหรับ Download Code และ รับข้อมูล ผ่านทาง RS232
VDD	---	3.3 V Power (2.7-3.6 VDC)
VSS	---	Ground
BOEn	I	Brown Out Enable(Active Low) จะใช้ต่อไปยัง VDD หรือ VSS ถ้าขานี้เป็น Low ขา RESn จะกลายเป็น Output แต่จะยังสามารถ Drive Low เพื่อ Reset Chip ได้ ถ้าเป็น high ขา RESn จะทำหน้าที่เป็น Input
RESn	I/O	Reset(Active Low) เมื่อเป็น Low ตัว Chip และ Cog ทั้งหมด จะถูก Disable PIN I/O จะถูกปล่อยลอย และ Chip จะ Restart ภายในเวลา 50ms หลังจาก Logic ที่ RESn เปลี่ยนจาก Low เป็น High
XI	I	Crystal Input
XO	O	Crystal Output

2.5.4.4โครงสร้างของบอร์ด ET-BASE PX32 V1.0

สำหรับบอร์ด ET-BASE PX32 V1.0 นี้ ได้กำหนด Port ในการ Interface กับอุปกรณ์รอบข้างเฉพาะอย่างไว้ให้เรียบร้อยแล้ว จะเหลือในส่วนของ Port I/O (P0-P7)อีก 8 Pin เท่านั้น ที่จัดไว้สำหรับให้ผู้ใช้นำไปใช้ต่อกับอุปกรณ์อื่นๆ

Port Number หน้าที่ที่ถูกต่อใช้งานบนบอร์ด ET-BASE PX32

P0-P7 ถูกกำหนดให้เป็น I/O ที่ผู้ใช้สามารถนำไปต่อใช้งานอื่นๆเพิ่มเติมได้

P8-P9 ถูกกำหนดให้ใช้สำหรับต่อ Microphone

P10-P11 ถูกกำหนดให้เป็น AUDIO Out Stereo สามารถต่อไปยังเครื่องขยาย หรือ Headphone ได้

P12-P15 ถูกกำหนดให้เป็น Port Video Out (NTSC/PAL)สำหรับต่อ เข้าช่อง AV ของ TV

P16-P23 ถูกกำหนดให้ใช้เป็น Port VGA สำหรับต่อเข้ากับจอ PC หรือ จอ LCD

P24-P25 ถูกกำหนดให้ใช้สำหรับต่อ PS/2 Mouse

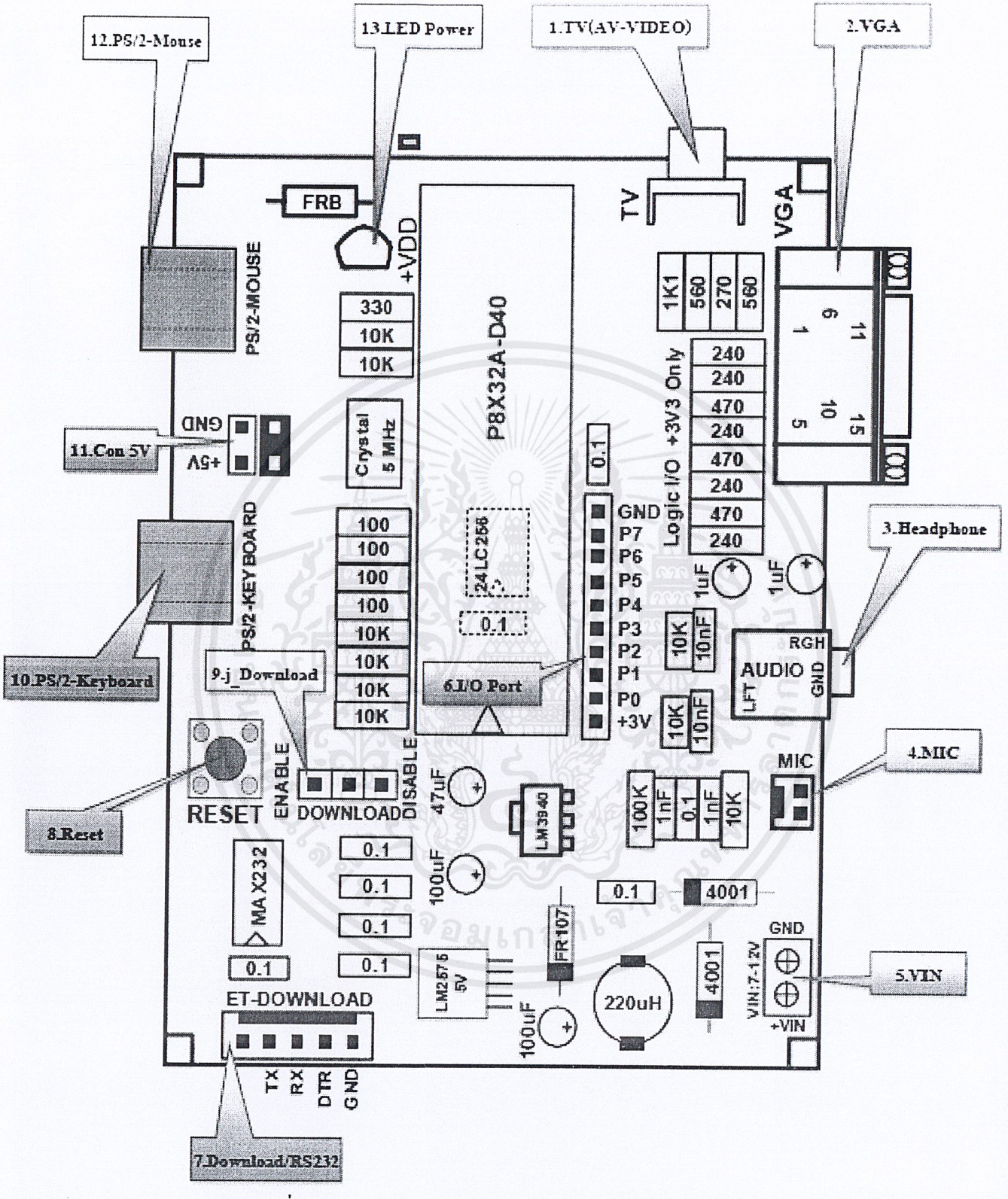
P26-P27 ถูกกำหนดให้ใช้สำหรับต่อ PS/2 Key Board

P28-P29[System] ถูกต่อเข้ากับ I2C-EPPROM เพื่อใช้เก็บ Code Program (P28:Clock ; P29:Data)

P30-P31[System] ถูกกำหนดให้ใช้เป็น Port RS232 เพื่อ Download Program และรับ-ส่งข้อมูลทาง RS232(P30:Rx;P31:Tx)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

119444



รูปที่2.11 แสดงโครงสร้างของบอร์ด ET-BASE PX32 V1.0

จากรูปที่2.5.4.4 แสดงโครงสร้างของบอร์ด ET-BASE PX32 V1.0 สามารถอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข1 : TV(AV-VIDEO) เป็นขั้วต่อ VIDEO OUT เพื่อใช้ต่อกับ TV โดยต่อเข้าที่ขั้วต่อ VIDEO IN ของ TV

หมายเลข2 : VGA เป็นขั้วต่อ VGA OUT (DB15) ใช้สำหรับต่อไปยัง Monitor ของ PC เช่น จอ LCD

หมายเลข3 : Headphone เป็นขั้วต่อ AUDIO OUT ใช้สำหรับต่อหูฟัง หรือต่อเข้ากับชุดขยายสัญญาณ

หมายเลข4 : MIC เป็นขั้วต่อ Microphone

หมายเลข5 : VIN เป็นขั้วต่อไฟเลี้ยงบอร์ด DC 7V-12 V ในการต่อจะต้องระวัง ต่อขั้วบวก-ลบ ให้ถูกต้องด้วย

หมายเลข6 : I/O Port เป็น Port I/O 8 Pin จัดไว้สำหรับให้ผู้ใช้สามารถต่อ I/O อื่นๆเพิ่มเติมได้ อาทิเช่น LCD,Key,LED หรือ SD Card เป็นต้น

ข้อควรระวัง สำหรับ I/O Port นี้ในกรณีที่ใช้งานเป็น Input จะรับแรงดันที่เข้ามายัง Port ได้ ไม่เกิน VDD หรือ 3.3 V ห้ามใช้แรงดัน 5 Vมาต่อเข้าโดยตรงเพราะจะทำให้ Port Pin เสียหายได้ ควรใช้ R มาต่อ Dividerให้เหลือ3.3V เสียก่อน

หมายเลข7 : Download/RS232 เป็นขั้วต่อใช้สำหรับ Download Program และใช้ ในการสื่อสาร รับ-ส่งข้อมูลผ่านทาง RS232 โดยมี Line Driver Max232 เป็นตัวปรับระดับสัญญาณ Rx,Tx จาก 3.3V ไปเป็น ±12V เพื่อให้สามารถต่อใช้งานร่วมกับ PC ได้

หมายเลข8 : Reset เป็น SW. สำหรับใช้ Reset การทำงานของ MCU โดยจะทำงานที่ Logic 0

*หมายเลข9 :*Jumper Download เมื่อจะ Download Program ให้ Set jumper มาทางด้าน Enable ซึ่ง Jumper นี้จะทำหน้าที่ต่อขา RES ของ MCU เข้ากับขา DTR ของขั้ว Download ถ้า Set Jumper มาทางด้าน Disable จะ

เป็นการตัดขา RES ของ MCU ออกจากขา DTR ของขั้ว Download เพื่อป้องกันสัญญาณ Reset จาก PC

หลังจากดาวน์โหลดเสร็จแล้ว และยังไม่ได้ถอดสาย Download ออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข10 : PS/2-Keyboard เป็นขั้วต่อ PS/2 ใช้สำหรับต่อ keyboard

หมายเลข11 : Con 5V เป็นขั้วต่อ DC 5 V Output สำหรับใช้ต่อไฟ 5 V จากบอร์ดไปใช้งานภายนอกได้

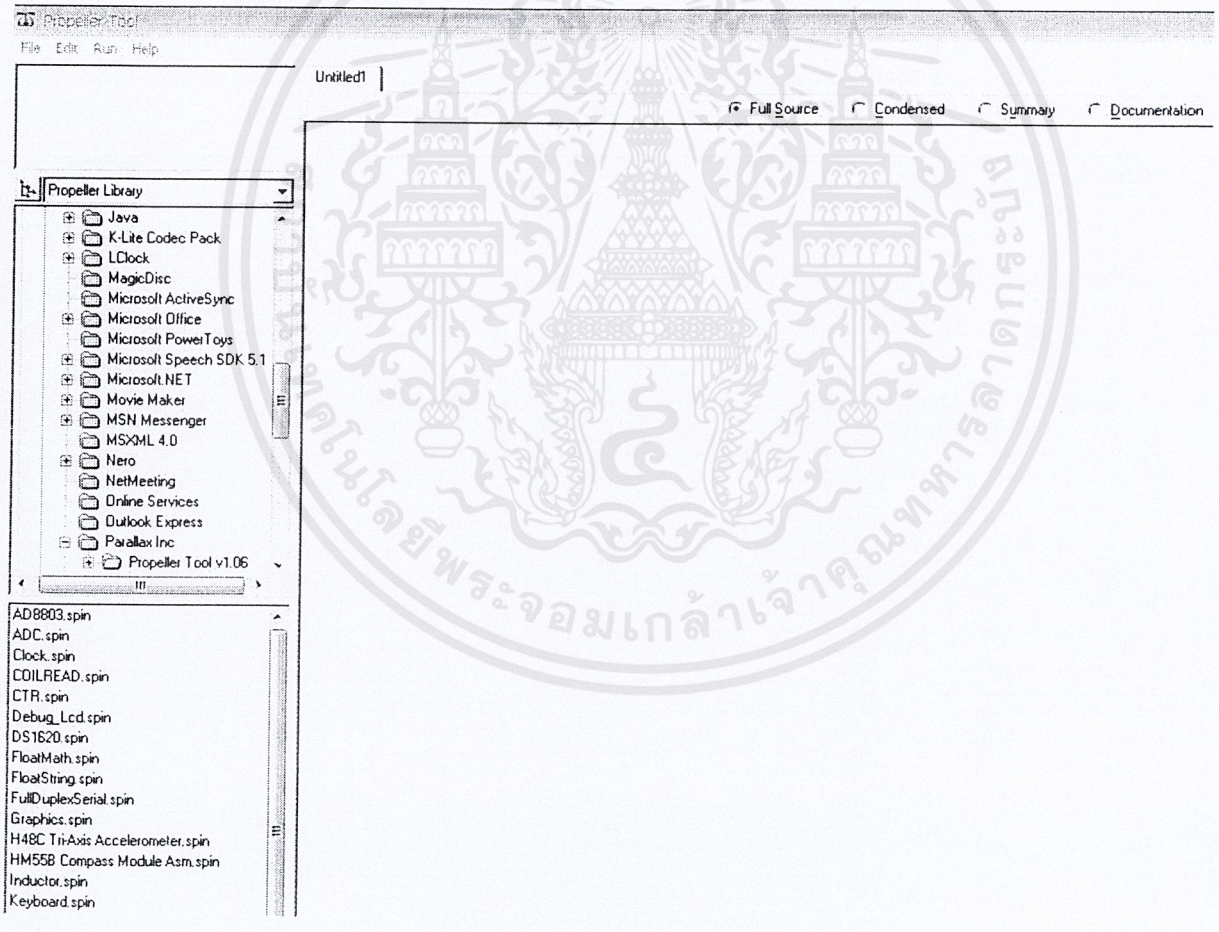
หมายเลข12 : PS/2-Mouse เป็นขั้วต่อ PS/2 ใช้สำหรับต่อ Mouse

หมายเลข13 : LED Power แสดงสะถานะการทำงานของแหล่งจ่ายไฟ

2.5.5การใช้งานโปรแกรม Propeller เบื้องต้น

1) ทำการติดตั้งโปรแกรม Propeller V1.06 ลงในเครื่อง

2) หลังจากติดตั้งแล้วให้เปิดโปรแกรม Propeller ขึ้นมาจะได้หน้าต่างดังรูปที่ 2.5.5.1



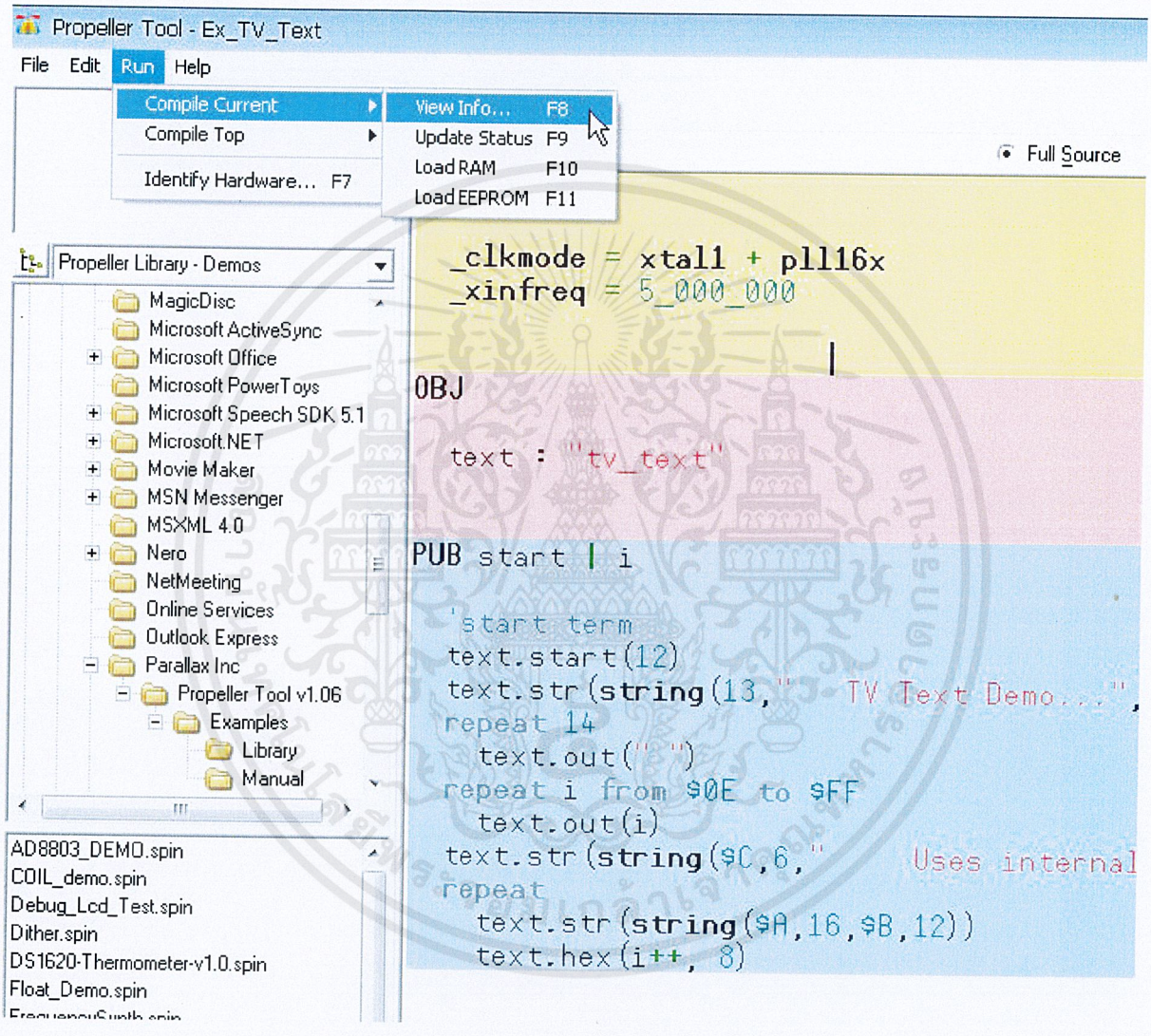
รูปที่ 2.12 หน้าต่างโปรแกรม Propeller

จากรูปในหน้าต่างด้านขวามือ(แท็บUntitled1) จะเป็นพื้นที่ว่างสำหรับใช้เขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) หลังจากเขียนโปรแกรม เสร็จก็ให้ไปที่เมนู File เลือก Save As เพื่อทำการ Save File เป็นนามสกุลจุด spin

4) เมื่อ Save File เรียบร้อยแล้ว ก็ให้ไปที่เมนู RUN และเลือกที่ Compile Current จากนั้นก็จะมีเมนูให้ผู้ใช้ เลือกต่ออีก ดังแสดงในรูปที่ 2.5.5.2 ถ้าเลือก

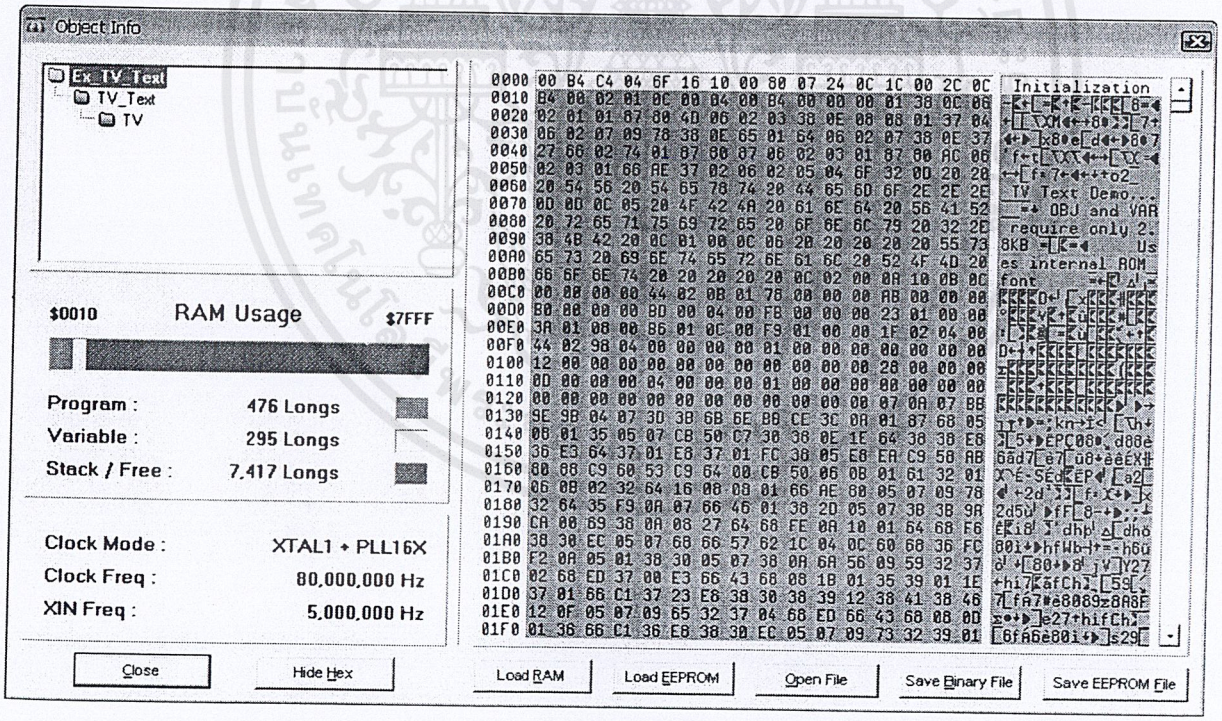


รูปที่ 2.13 หน้าต่างเมนู RUN และเลือก Compile Current

View Info... = จะเป็นการ Compile โปรแกรมอย่างเดียว และจะมีหน้าต่าง Object info รายงานข้อมูลต่างๆของโปรแกรมที่ เขียนขึ้น ดังรูปที่5.3 ซึ่งก็จะแสดงพื้นที่ RAM ที่ถูกใช้ไป แสดงความถี่ Clock ที่ผู้ใช้เลือกใช้งาน ซึ่งข้อมูลของตัวอย่างโปรแกรมในรูปที่ 5.3 นี้สามารถอธิบายได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เป็นข้อมูลของไฟล์ที่ชื่อ Ex_TV_Text โดยภายในไฟล์นี้ ได้เรียกใช้งาน Function ที่อยู่ใน File Library ที่ชื่อ TV_Text และ ใน File library TV_Text ก็ได้เรียกใช้ Function ใน File library ที่ชื่อ TV อีกต่อหนึ่ง
- ไฟล์ Ex_TV_Text นี้ ใช้เนื้อที่ RAM ในส่วนที่เป็นโปรแกรมไปทั้งหมด 476 Long และในส่วนที่เป็นตัวแปร ไปทั้งหมด 295 Long และเหลือพื้นที่ RAM ที่ยังว่างอยู่ทั้งหมด 7,417 Long โดย 1 Long จะมีค่าเท่ากับ 32 bit หรือ 4 byte
- ในโปรแกรมนี้ได้เลือกใช้ Clock Mode ในโหมด XTAL1+PLL16X โดยความถี่ Clock ที่ใช้จริงก็จะเท่ากับ 80 Mhz โดยใช้ Crystal จากภายนอก 5 MHz (5MHz x PLL16 = 80MHz) หลังจากหน้าต่าง Object Info แสดงขึ้นมาแล้ว ให้ผู้ใช้เลือกคลิกที่ปุ่ม Load Ram หรือ ปุ่ม Load EEPROM เพื่อทำการ Download Program โดยถ้าเลือกปุ่ม Load RAM โปรแกรมจะถูก Load เข้าไปเก็บไว้ยังพื้นที่ RAM ของ MCU โดยตรง ซึ่งการโหลดวิธีนี้ โปรแกรมจะถูกลบเมื่อมีการกด SW. Reset หรือเอาไฟเลี้ยงบอร์ดออก , ถ้าเลือกปุ่ม Load EEPROM โปรแกรมจะถูก Load เข้าไปเก็บไว้ยัง EEPROM ที่ต่ออยู่ภายนอกก่อน จากนั้น ตัว MCU ก็จะทำการดึง Code จาก EEPROM เข้าไป RUN ใน RAM ให้อัตโนมัติ ซึ่งการโหลดวิธี นี้ โปรแกรมจะไม่ถูกลบเมื่อมีการกด SW. Reset หรือเอาไฟเลี้ยงบอร์ดออก



รูปที่ 2.14 หน้าต่างเมนู Clock Mode ในโหมด XTAL1+PLL16X

Load RAM = จะเป็นการ Compile และ Download โปรแกรมลงใน RAM เลย จะไม่แสดงหน้าต่าง Object Info ให้เห็น โปรแกรมจะถูกลบ ถ้ามีการกด SW. Reset หรือ ตัดไฟเลี้ยงออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

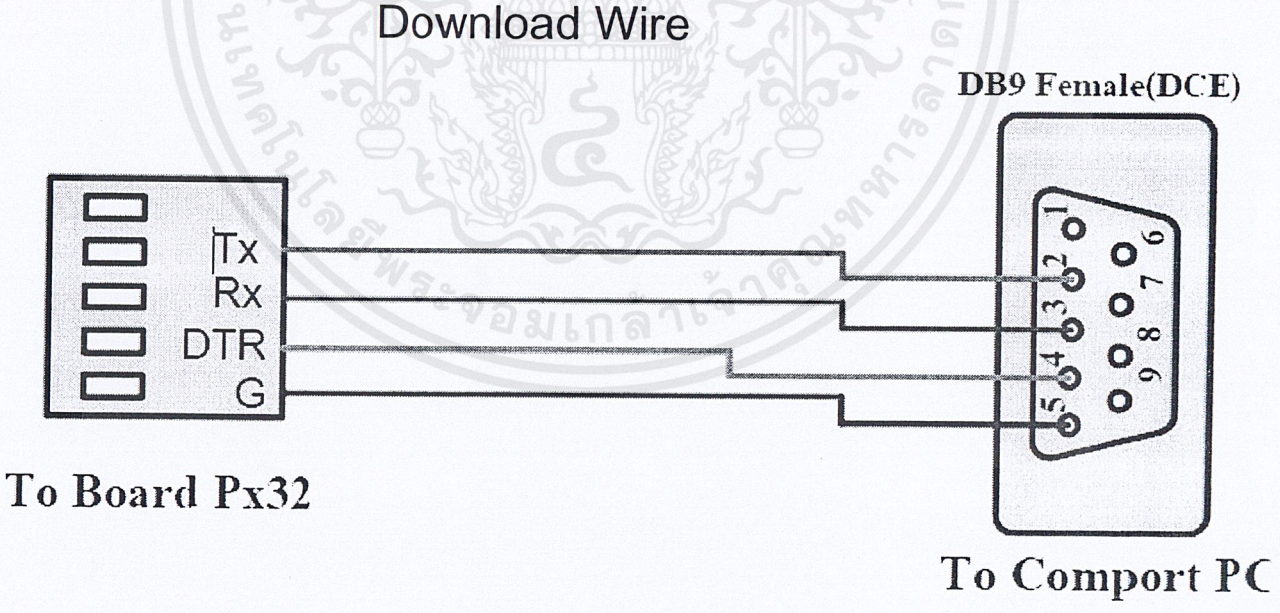
Load EEPROM = จะเป็นการ Compile และ Download โปรแกรมลงใน EEPROM ที่ต่ออยู่ภายนอกจะไม่แสดงหน้าต่าง Object Infoให้เห็นเช่นกัน โปรแกรมจะยังคงอยู่ เมื่อ มีการ Reset หรือตัดไฟเลี้ยงออก

5) สำหรับการ Set Baud Rate หรือ เลือก COM Port เพื่อใช้ในการ Download Code นั้นจะไม่จำเป็น เพราะ ตัวโปรแกรมPropeller นั้นจะทำการ Set Baud Rate และหา Com Port ให้โดยอัตโนมัติ แต่ถ้าผู้ใช้ต้องการจะกำหนด Com Port เองก็สามารถเข้าไปแก้ไขได้โดยไปที่เมนู Edit แล้วเลือก Preferences จะมีหน้าต่างขึ้นมาให้เลือก ที่แท็บ Operation ในช่องSerial Port Search : ให้เลือก Comport ที่จะใช้งาน (ปกติ = Auto)

6) สำหรับตัวโปรแกรม Propeller นี้จะใช้การพัฒนาโปรแกรมด้วยภาษา SPIN

2.5.6 การต่อสาย Download และการต่อสายสื่อสาร RS232

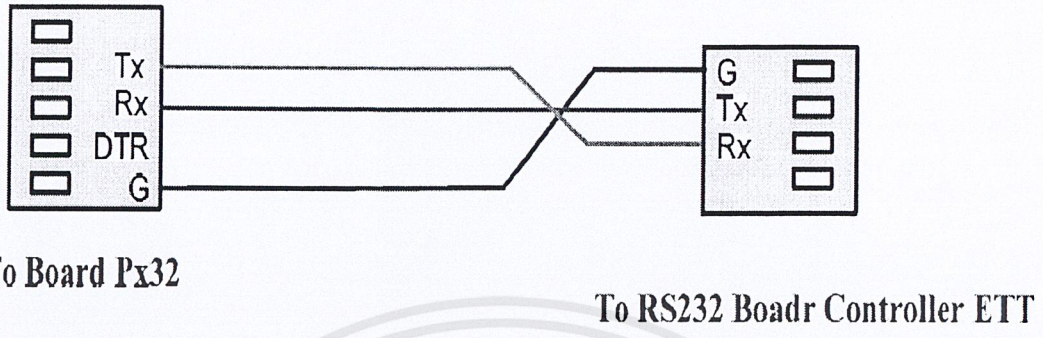
-สำหรับสาย Download Program และสายสำหรับรับ-ส่งข้อมูลทาง RS232 ระหว่างบอร์ด PX32 กับ PC จะใช้สายเส้นเดียวกัน โดยการเข้าสายแสดงดังรูปที่ 2.5.6.1



รูปที่ 2.15 แสดงการเข้าสาย Download & สายสื่อสาร RS232 ระหว่างบอร์ด Px32 และ PC

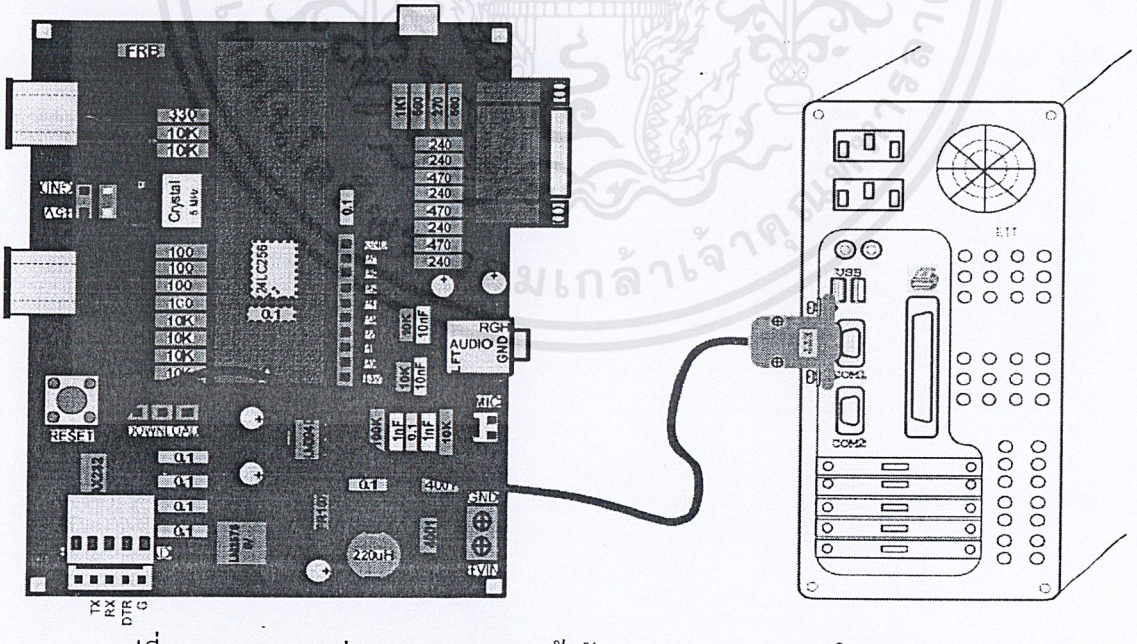
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่จะ รับ-ส่ง ข้อมูลทาง RS232 ระหว่างบอร์ด PX32 กับบอร์ด Controller อื่นๆ สามารถใช้การเข้าสายดังแสดงในรูปที่ 2.5.6.2



รูปที่ 2.16 แสดงการเข้าสาย RS232 สำหรับ รับ-ส่ง ข้อมูล ระหว่างบอร์ด PX32 กับบอร์ด Controller อื่นๆ

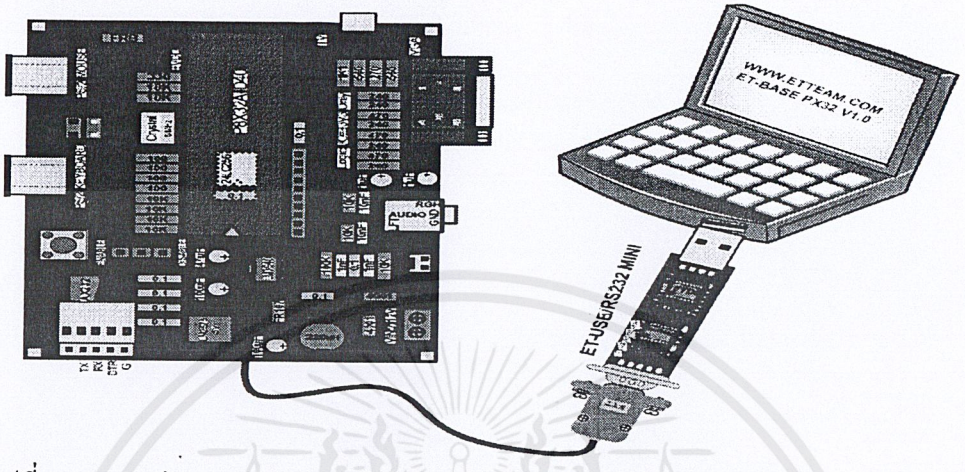
-เมื่อจะทำการ Download Program ลงบนบอร์ด Px32 ก็ให้ทำการต่อสาย download ทางด้าน Connector 5 Pin เข้ากับขั้วต่อของบอร์ด PX32 ส่วนทางด้าน Connector DB9 ก็ให้ต่อเข้ากับ Com Port ของ PC ที่ใช้งานดังรูปที่ 7.3 และทำการ Set Jumper Download(หมายเลข9) บนบอร์ด PX32 มาทางด้าน Enable , เมื่อจะทำการรับ-ส่ง ข้อมูล ผ่านทาง RS232 ของบอร์ดPX32 กับ PC หรือกับบอร์ด Controller อื่นๆ หลังจาก Download โปรแกรมเรียบร้อยแล้ว ก็ควรจะ Set JumperDownload(หมายเลข9) บนบอร์ด PX32 มาทางด้าน Disable



รูปที่ 2.17 แสดงการต่อสาย Download เข้ากับ Com-Port ของ PC โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ใช้เครื่อง PC หรือ Note Book ที่ไม่มีขั้วต่อ Com-Port ผู้ใช้จะต้องใช้ชุดแปลงจาก USB ไปเป็น RS232เสียก่อน โดยถ้าใช้ชุดแปลงของ ETT รุ่น ET-USB/RS232 Mini ก็สามารถต่อใช้งานดังแสดงในรูปที่ 2.5.6.4



รูปที่ 2.18 การต่อสาย Download โดยผ่านตัวแปลงชุด ET-USB/RS232 Mini

2.5.7 ข้อกำหนดที่ควรรู้ในการพัฒนาโปรแกรมด้วยภาษา SPIN บน Tool Propeller

1) สำหรับภาษา SPIN นี้เวลาใช้งานเราจะมองออกเป็น Block ซึ่งมีอยู่ด้วยกันทั้งหมด 6 Block โดยเวลาจะเขียนก็จะต้องพิมพ์ชื่อเฉพาะของแต่ละ Block ที่ถูกกำหนดไว้ตายตัวด้วย โดยเวลาพิมพ์ชื่อเฉพาะของแต่ละ Block นี้ มีข้อกำหนดว่าจะต้องเริ่มพิมพ์ที่Colum ซ้ายสุดของหน้าต่างที่จะใช้เขียนโปรแกรมเสมอ โดยชื่อ Block ทั้ง 6 Block มีดังนี้

1.1) **CON** - Block นี้จะใช้สำหรับกำหนดค่าคงที่ให้กับตัวแปรที่จะใช้งาน ซึ่งตัวแปรที่ประกาศใน Block นี้สามารถเรียกใช้ได้ภายในไฟล์เดียวกัน ตัวอย่างเช่น

```
CON

  _clkmode      = xtal1 + pll16x
  _xinfreq      = 5_000_000
      tt        = 20
```

รูปที่ 2.19 กำหนดค่าคงที่โดยใช้ **CON** - Block

1.2.) **VAR** - Block นี้จะใช้สำหรับประกาศตัวแปรที่จะใช้ในการเขียนโปรแกรม โดยตัวแปรที่ประกาศใน Block นี้สามารถเรียกใช้งานได้ภายใน Project File เดียวกัน ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
VAR
        long    testerror
        long    vlong
        word    vword
        byte    vbyte
```

รูปที่ 2.20 การประกาศตัวแปร**VAR** - Block

1.3.) **OBJ** - Block นี้จะใช้สำหรับกำหนดชื่อไฟล์จากภายนอกเข้ามา เพื่อให้สามารถเรียกใช้ฟังก์ชันที่มีอยู่ในไฟล์จากภายนอกได้ (ไฟล์จากภายนอกที่ไม่ใช่ไฟล์ Library ของตัว Propeller เวลาจะเรียกใช้ฟังก์ชันที่อยู่ในไฟล์นั้นๆผู้ใช้จะต้อง Copy File นั้นมาไว้ที่เดียวกับไฟล์ Project ที่ผู้ใช้เขียนอยู่ด้วย) การดูฟังก์ชันที่อยู่ภายใน ไฟล์Library ของ Propeller ทำได้โดย ดูรูปที่ 6.1 ในช่องที่ Mouse ชี้อยู่ให้เลือก "Propeller Library" จากนั้นในหน้าต่างด้านล่าง ก็จะแสดงชื่อไฟล์ .spin ออกมาให้เห็นซึ่งไฟล์เหล่านี้ก็คือ File Library ของตัว Propellerที่มี ไว้ให้เรียกใช้ เมื่อดับเบิลคลิกที่File Library ไฟล์ก็จะถูกเปิดขึ้นมา และจะมีฟังก์ชั่นต่างๆถูกเขียนไว้ ซึ่งฟังก์ชั่นเหล่านั้นผู้ใช้ สามารถเรียกมาใช้ในโปรแกรมที่ผู้ใช้เขียนอยู่ได้ ตัวอย่างเช่น

```
OBJ
        term    : "tv_terminal"
```

รูปที่ 2.21 การกำหนดชื่อไฟล์จากภายนอก **OBJ** - Block

เวลาจะเรียกใช้งานฟังก์ชันที่อยู่ในไฟล์ Library " tv_terminal" ก็ให้แทนด้วย term.*ชื่อฟังก์ชัน* เช่น term.out(12) เป็นต้น

1.4.) **PUB** - สำหรับ Block นี้จะมีไว้สำหรับให้ผู้ใช้เขียนโปรแกรมที่ต้องการลงไป และสามารถเรียกใช้งานฟังก์ชันที่เขียนอยู่ใน block นี้ ได้ทั้งภายใน Project File เดียวกัน หรือต่าง Project File กันก็ได้ ในทุกๆ Project File ที่สร้างขึ้น จะต้องมี Block PUB อยู่อย่างน้อย 1 Block โดย Block PUB Block แรก ตัว Propeller จะถือเป็น Blockmain ของโปรแกรม ส่วน Block PUB ที่อยู่ต่อจาก Block PUB แรกจะถูกมองเป็นเพียงโปรแกรมย่อยมีไว้สำหรับให้โปรแกรมหลักเรียกใช้ โดยใน Block นี้ เมื่อพิมพ์คำว่า PUB แล้วจะต้องตั้งชื่อให้กับ Block PUBด้วยซึ่งชื่อนี้ก็จะเป็นเหมือนชื่อของFunction นั่นเอง ตัวอย่างเช่น

```
PUB stop

  vga.stop
```

รูปที่ 2.22 **PUB** - Block สำหรับให้ผู้ใช้เขียนโปรแกรมที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาจะประกาศตัวแปรใช้ภายใน PUB นั้นๆ ก็ให้ใส่ | ต่อจากชื่อ แล้วจึงตามด้วยชื่อตัวแปร ซึ่งตัวแปรที่ ประกาศนี้จะใช้งานได้เฉพาะภายใน PUB เท่านั้น และตัวแปรที่ประกาศจะมีขนาด 32 bit

เวลาจะรับค่าจากภายนอกเข้ามาใช้ยังฟังก์ชัน ก็ให้ใส่ **(ชื่อตัวแปร1,ตัวแปร2)** ต่อจากชื่อ โดยภายในวงเล็บก็ให้ใส่ชื่อตัวแปรที่ใช้รับการส่งผ่านค่าเข้ามาในฟังก์ชันด้วย

เวลาจะส่งผ่านค่าจากฟังก์ชันออกไปยังภายนอก ก็ให้ใส่ : ต่อจากชื่อ แล้วตามด้วยชื่อตัวแปรที่จะส่งผ่านค่าออกไป ดังตัวอย่าง

```
PUB Ticks(Pin) : Microseconds | cnt1, cnt2

  outa[Pin]~
  dira[Pin]~~
  outa[Pin]~~
  outa[Pin]~
  dira[Pin]~

  waitpne(0, |< Pin, 0)
  cnt1 := cnt
  waitpeq(0, |< Pin, 0)
  cnt2 := cnt
  Microseconds := (||(cnt1 - cnt2) / (clkfreq / 1_000_000)) >> 1
```

รูปที่ 2.23 การประกาศตัวแปรใช้ภายใน PUB

ในตัวอย่างนี้ชื่อ PUB คือ Ticks ส่วน (Pin) เป็นการประกาศตัวแปรเพื่อใช้รับค่าจากภายนอกเข้ามาเก็บไว้ที่ตัว แปร Pin ส่วน : Microsecond เป็นการประกาศตัวแปรเพื่อใช้ส่งผ่านค่าออกจากฟังก์ชัน และสุดท้าย | cnt1,cnt2 เป็นการประกาศตัวแปรไว้ใช้งานภายใน Block PUB Ticks เวลาใช้งานจริงก็ประกาศในส่วนที่จะใช้งานก็พอ ไม่จำเป็นต้องประกาศจนครบทั้งหมดก็ได้

1.5.) **PRI** -สำหรับ Block นี้จะมีไว้สำหรับให้ผู้ใช้เขียนโปรแกรมที่ต้องการลงไปเหมือนกับ Block PUB แต่จะต่างตรงที่Function ที่เขียนอยู่ใน Block นี้จะถูกมองเป็นโปรแกรมย่อยสำหรับให้เรียกใช้ภายใน Project File เดียวกันเท่านั้น ไม่สามารถเรียกใช้งานคนละ Project File ได้ ส่วนการประกาศใช้งานตัวแปรก็จะเหมือนกับ Block PUB

```
PRI bound(i, delta) : b | d

  d := bx_div[i]
  b := bx_min[i] + (bx_acc[i] := bx_acc[i] + delta
```

รูปที่ 2.24 Block PRI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6.) **DAT** - สำหรับ Block นี้จะมีไว้สำหรับ กำหนด data table หรือ สำหรับ เขียน Assembly Code ตัวอย่างเช่น

```
DAT

'*********************************
'* Assembly language VGA driver *
'*********************************

                        org

entry                   mov     taskptr,#tasks

                        mov     x,#8
:init                   jmpret  taskret,taskptr
                        djnz    x,#:init


vgacolors               long    $C000C000       'red
                        long    $C0C00000
                        long    $08A808A8       'green
                        long    $0808A8A8
                        long    $50005000       'blue
                        long    $50500000
```

รูปที่ 2.25 Block DAT

Block ทั้ง 6 นี้เวลาจะใช้งานผู้ใช้สามารถประกาศใช้เฉพาะ Block ที่จะใช้งานเท่านั้นก็ได้ไม่จำเป็นต้องประกาศทุกBlock และในแต่ละ Block สามารถประกาศใช้ซ้ำกันได้มากกว่า 1 Block โดยใน Block PUB เวลาประกาศใช้ซ้ำกัน ชื่อของฟังก์ชันที่พิมพ์ต่อจาก PUB จะต้องไม่เหมือนกัน

2) ตัวอักษรที่พิมพ์ไม่ว่าจะเป็นการพิมพ์คำสั่ง หรือ การพิมพ์ชื่อตัวแปรก็ตาม สามารถพิมพ์โดยใช้ ตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก แทนกันได้ โดยตัว Propeller จะมองเป็นตัวแปรตัวเดียวกัน จะไม่มองแยกกันคนละตัวเหมือนภาษา C

3) เวลาแทนค่าเลข ฐาน10 เกิน 3 หลักจะต้องคั่นด้วย _ เสมอ เช่น x := 1_200

4) การใช้เครื่องหมายเท่ากับ ถ้าใช้ใน Block CON จะใช้ = ,ถ้าใช้ใน Block OBJ จะใช้ : ,ถ้าใช้ใน Block PUB จะใช้ := ,

5) การใช้เลขฐานใน Propeller : $ นำหน้าค่าคงที่ เช่น $56 จะใช้แสดงถึงตัวเลขฐาน 16 , % นำหน้าค่าคงที่(1และ0) เช่น%1001 จะแสดงถึงตัวเลขฐาน2 , %% นำหน้าค่าคงที่ (0,1,2,3) เช่น %%2130

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมองเป็น Quaternary Number ก็คือให้มองเลข 1หลัก มีขนาด 2 บิต จากตัวอย่าง %%2130 = 10 01 11 00 (ฐาน2) = 9C (ฐาน16)

6) ในการใช้งานคำสั่งเกี่ยวกับ Loop เช่น repeat หรือ คำสั่งเงื่อนไข เช่น if หลังจากพิมพ์คำสั่งแล้ว ส่วนของโปรแกรมที่จะพิมพ์ในบรรทัดต่อมาเพื่อให้ทำงานภายใต้คำสั่ง repeat หรือ if จะต้องพิมพ์ให้เยื้องกับคำสั่งเหล่านี้มาทางขวามืออย่างน้อย 1 ช่องว่าง ถ้าพิมพ์ให้อยู่ในระดับเดียวกัน หรือเยื้องไปทางด้านซ้ายมือ ตัว Propeller จะถือว่าโปรแกรมในบรรทัดนั้นอยู่ภายนอกคำสั่ง repeat หรือ if เป็นต้น ตัวอย่างเช่น

```
PRI BlinkingLED
  Pin := 07
  DirA[Pin] := Out
  Repeat
    OutA[Pin] := High              'LED ON
    WaitCnt(40_000_000 + Cnt)  'ONE-HALF SECOND WAIT
    OutA[Pin] := Low               'LED OFF
    WaitCnt(40_000_000 + Cnt)  'ONE-HALF SECOND WAIT
```

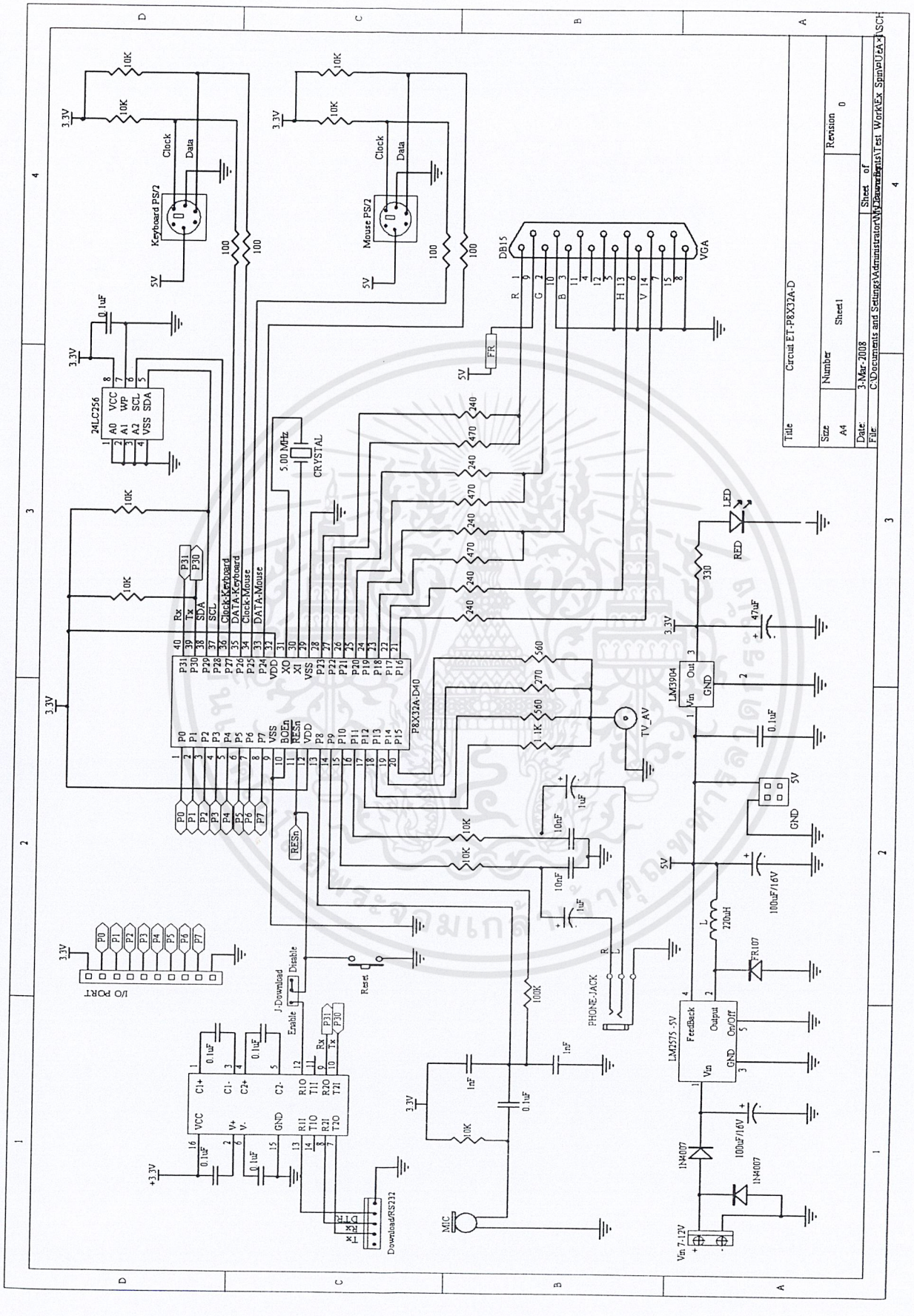
รูปที่ 2.26 การใช้งานคำสั่งเกี่ยวกับ Loop เช่น repeat

7) ในการเขียน Comment อธิบายโปรแกรมจะนำด้วยสัญลักษณ์ ' หรือ '' แล้วตามด้วย Comment หรือเขียนไว้ในเครื่องหมายปีกกา {comment} หรือ {{comment}}

8) ใน Block PUB ที่ใช้เป็น Block main ของโปรแกรม ถ้าโปรแกรมที่เขียนใน Block เป็นแบบ Loop เปิด ไม่มีการทำงาน
วนซ้ำภายใน Loop ใดๆ ก็ควรจะจบโปรแกรมด้วยคำสั่ง Repeat เพื่อไม่ให้ MCU วนกลับไป Reset เริ่มโปรแกรมใหม่

รูปต่อไป รูปที่ 2.27 รูปวงจร ET-BASE PX32 V1.0

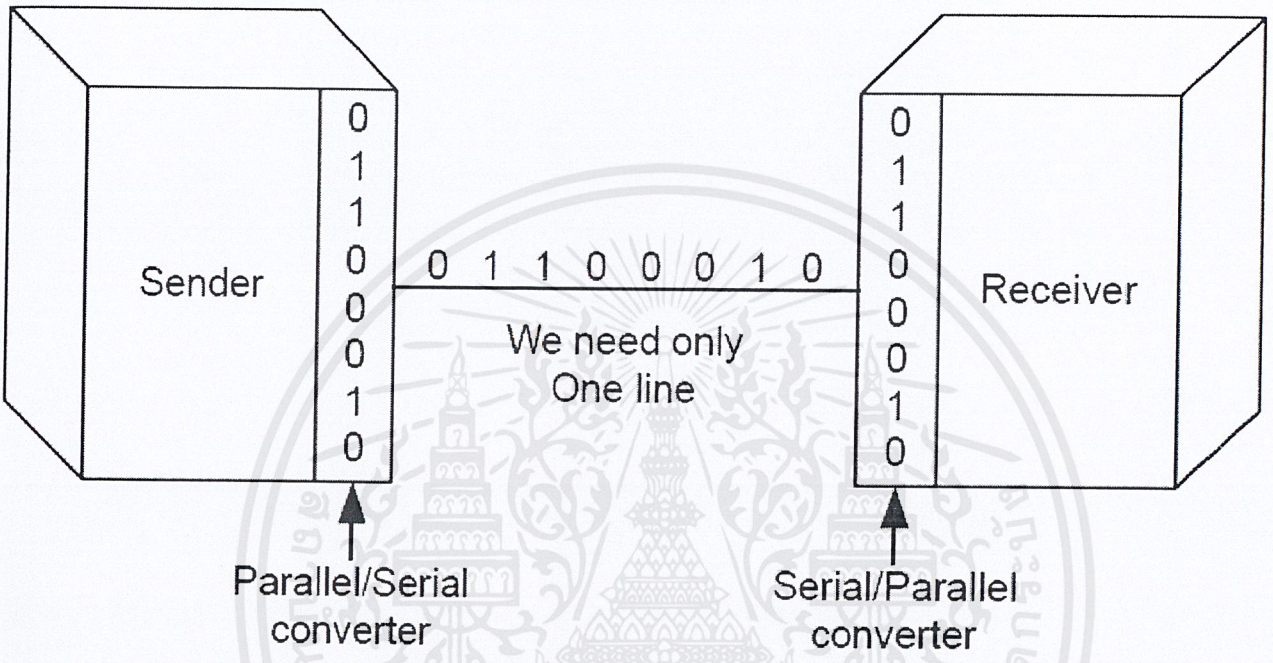
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การสื่อสารข้อมูลแบบอนุกรม(Serial Transmission)

วิธีการสื่อสารข้อมูลแบบอนุกรมนั้น สัญญาณจะทยอยส่งไปตามสายสื่อสารเพียงเส้นเดียวซึ่งแสดงได้ดังรูปที่ 2.18 จะเห็นได้ว่าบิตจะทยอยส่งออกมาทีละบิตจากต้นทางไปยังปลายทาง โดยปลายทางจะทำการรวบรวมสัญญาณหรือบิตที่ทยอยส่งมาจนครบ 8 บิต หรือ 1 ไบท์ เพื่อนำไปใช้งานต่อไป



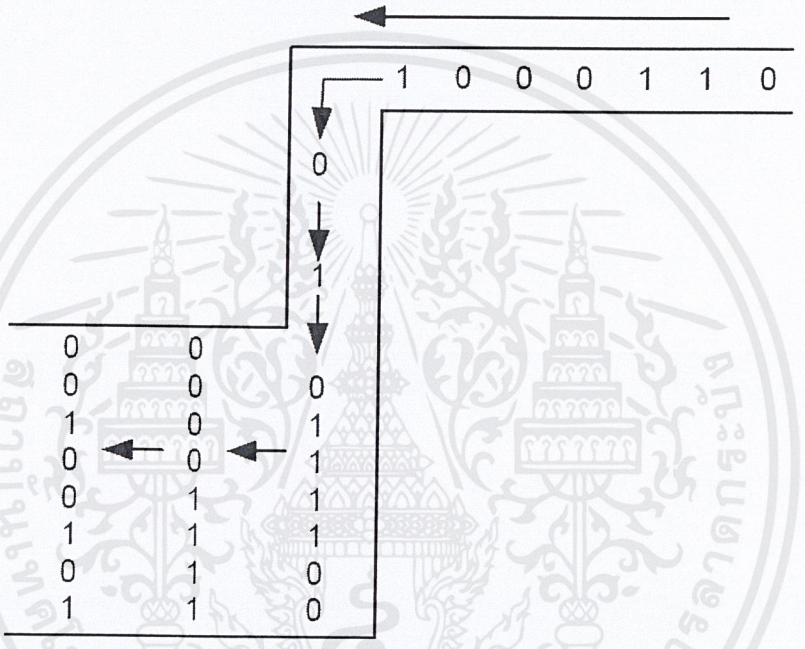
รูปที่ 2.28 การส่งข้อมูลแบบอนุกรม

2.6.1 การแปลงสัญญาณข้อมูลระหว่างแบบอนุกรมและแบบขนาน

ในการแปลงรูปแบบข้อมูลระหว่างอนุกรมและขนาน จะอาศัยรีจิสเตอร์เพื่อเป็นบัฟเฟอร์ในการเก็บข้อมูลชั่วคราว เช่น ข้อมูลที่ส่งเข้ามาเป็นรูปแบบอนุกรมซึ่งจะส่งเรียงเข้ามาทีละบิต และเมื่อเข้ามาถึงปลายทาง บิตแต่ละบิตจะถูกนำมาจัดเก็บเรียงลำดับกันอยู่ในบัฟเฟอร์ จนกระทั่งครบตามจำนวนบิตที่ต้องการ เช่น เรียงกันจนครบ 8 บิต จากนั้นรีจิสเตอร์ก็จะส่งข้อมูลทั้งชุดนี้ออกไปด้วยการส่งสัญญาณให้หน่วยประมวลผลรับทราบ เพื่อให้โปรแกรมนำไบท์เหล่านั้นไปประมวลผลต่อไป ในขณะที่หากต้องการแปลงข้อมูลในรูปแบบขนานกลับไปเป็นแบบอนุกรม ก็สามารถกระทำได้ด้วยกระบวนการดังกล่าวในทิศทางตรงข้ามโดยพิจารณาจากรูปที่ 2.19 และรูปที่ 2.20 ที่แสดงขั้นตอนการแปลงข้อมูลระหว่างแบบอนุกรมและแบบขนาน ซึ่งกระบวนการแปลงสัญญาณข้อมูลเหล่านี้จะมีวงจรพิเศษที่เรียกว่า **UART(Universal Asynchronous Receiver Transmitter)**ที่ทำการแปลงข้อมูลแบบขนานเป็นแบบอนุกรมหรือแปลงกลับจากแบบอนุกรมเป็นแบบขนานนอกจากนี้ยังมีวงจรที่เรียกว่า **USART(Universal**

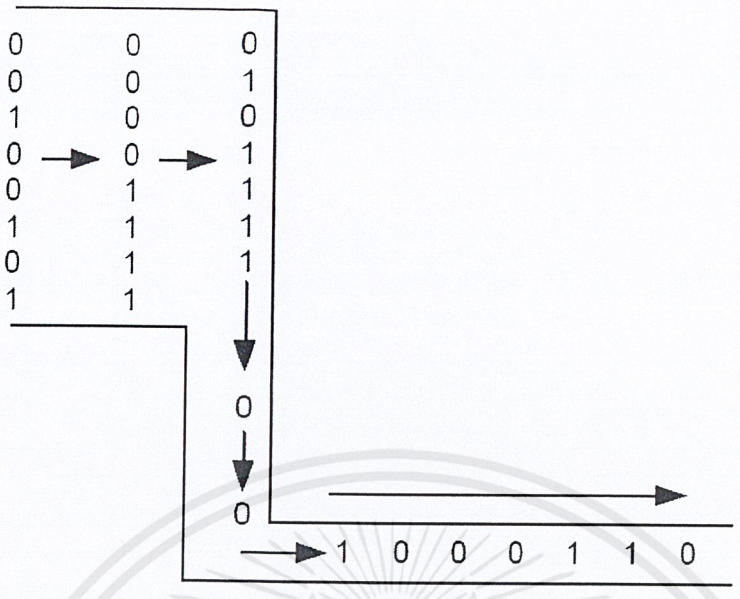
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Synchronous/Asynchronous Receiver Transmitter) ซึ่งมีคุณสมบัติเช่นเดียวกับ UART แต่จะมีส่วนของการทำงานกับการซิงโครนัสข้อมูลด้วยปัญหาของการส่งข้อมูลแบบอนุกรม คือเรื่องแบ่งตัวอักขระแต่ละตัวว่าจะแบ่ง ณ ตำแหน่งบิตใดซึ่งทั้งฝ่ายต้นทางและปลายทางจะต้องมีข้อตกลงร่วมกัน กล่าวคือ ทั้งฝ่ายต้นทางและปลายทางจะต้องรับรู้ร่วมกันว่าจะต้องแบ่งแต่ละตัวอักขระ ณ ตำแหน่งบิตใด เนื่องจากบิตแต่ละบิตจะทยอยส่งมาเป็นลำดับเรื่อยๆ ดังนั้น การส่งข้อมูลแบบอนุกรมจึงมีวิธีการอยู่ 2 วิธีด้วยกัน คือ "การส่งข้อมูลแบบอะซิงโครนัส" และ "การส่งข้อมูลแบบซิโครนัส"



รูปที่ 2.29 การแปลงข้อมูลจากแบบอนุกรมไปเป็นแบบขนาน(Serial-to-Parallel)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



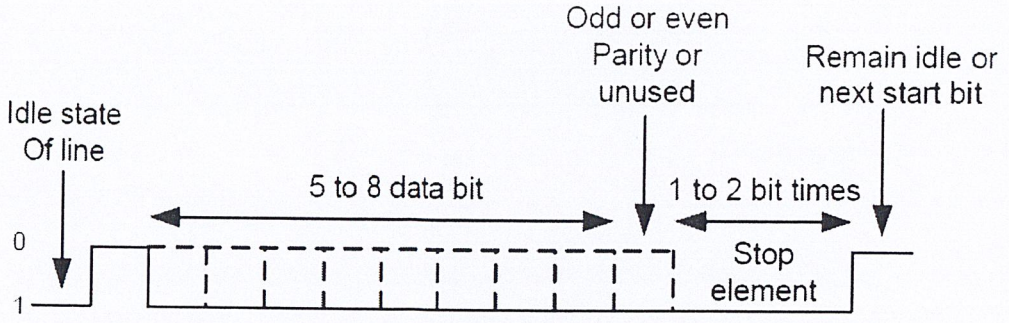
รูปที่ 2.30 การแปลงข้อมูลจากแบบขนานไปเป็นแบบอนุกรม(Parallel-to-Serial)

2.6.2 การส่งข้อมูลแบบอะซิงโครนัส(Asynchronous Transmission)

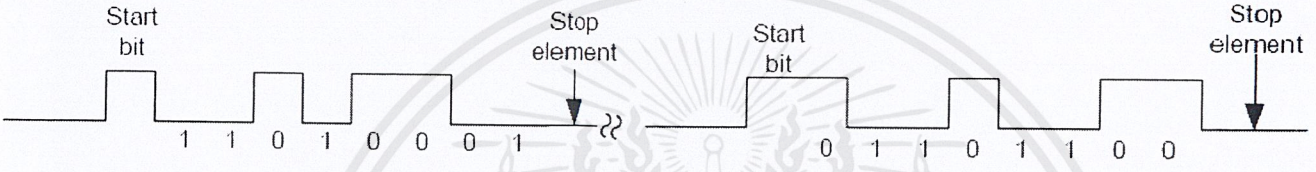
การส่งข้อมูลแบบอะซิงโครนัสเป็นการสื่อสารด้วยวิธีการส่งอักขระ(Character) แต่ละตัวณ เวลาใดก็ได้โดยฝ่ายส่งข้อมูลและฝ่ายรับข้อมูลต่างก็มีสัญญาณนาฬิกาควบคุมจังหวะการทำงานด้วยตัวเอง จึงทำให้การทำงานของทั้งสองฝ่ายไม่สอดคล้องกันตามจังหวะนาฬิกา กล่าวคือจะเป็นอิสระต่อกัน แต่อย่างไรก็ตามสัญญาณนาฬิกาทั้งสองฝ่ายนั้นจะต้องมีความถี่เท่ากัน

ในสภาวะนิ่งเฉย(Idle State)ที่ไม่มีการส่งข้อมูลใดๆ(บางครั้งเรียกสภาวะ Marking) จะถูกกำหนดให้สัญญาณมีค่าเป็น “1” แต่เมื่อมีการส่งข้อมูลระดับสัญญาณจะถูกกำหนดให้เป็น “0” อยู่ช่วงเวลาหนึ่งทำให้เกิดเป็นบิตขึ้นมาหนึ่งบิตที่เรียกว่า **บิตเริ่มต้น(Start Bit)** เพื่อบ่งบอกว่านับจากนี้ไปจะมีข้อมูลส่งมา และเมื่อฝ่ายส่งได้ส่งบิตข้อมูลจนครบตามจำนวนบิตที่ต้องการแล้ว จากนั้นก็จะส่งข้อมูลอีกบิตหนึ่ง ซึ่งระดับสัญญาณจะถูกกำหนดให้เป็น “1” เป็นตัวปิดท้ายที่เรียกว่า **บิตสิ้นสุด(Stop Bit)** เพื่อบ่งบอกให้รู้ว่าได้ส่งข้อมูลครบแล้ว โดยสัญญาณ “1” ที่เป็นบิตสิ้นสุดนี้จะส่งมานานช่วงระยะเวลาหนึ่ง ทำให้ฝ่ายรับได้รับรู้ทันทีว่ามีการส่งบิตสิ้นสุดมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) Character format



รูปที่ 2.31 รายละเอียดของรูปแบบการส่งข้อมูลแบบอะซิงโครนัส

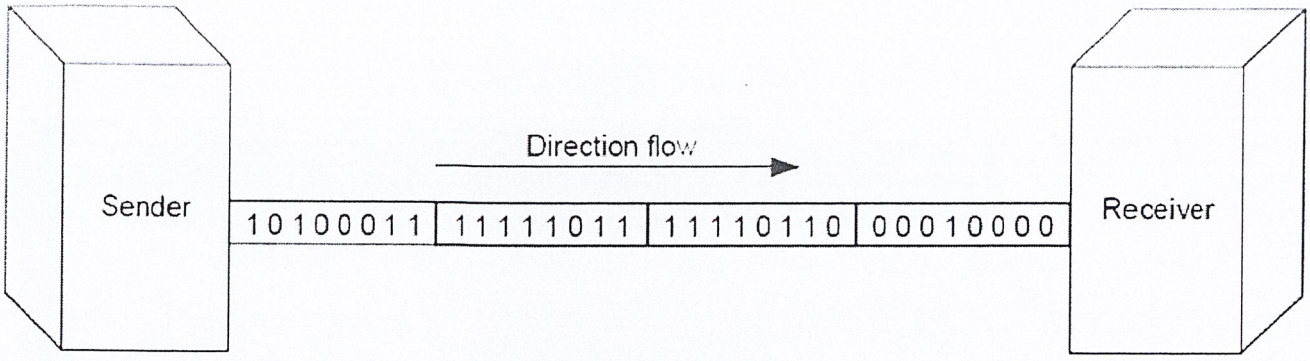
หลังจากที่ได้รับข้อมูลครบตามจำนวนบิต ตัวอักขระตัวที่สองที่จะส่งเป็นลำดับถัดไป ก็ไม่ต้องมีเวลาเท่ากันเสมอ ซึ่งหากยังไม่มีข้อมูล สัญญาณก็จะอยู่ในสถานะนิ่งเฉยเพื่อรอการส่งข้อมูลลำดับถัดไป

2.6.3 การส่งข้อมูลแบบซิงโครนัส(Synchronous Transmission)

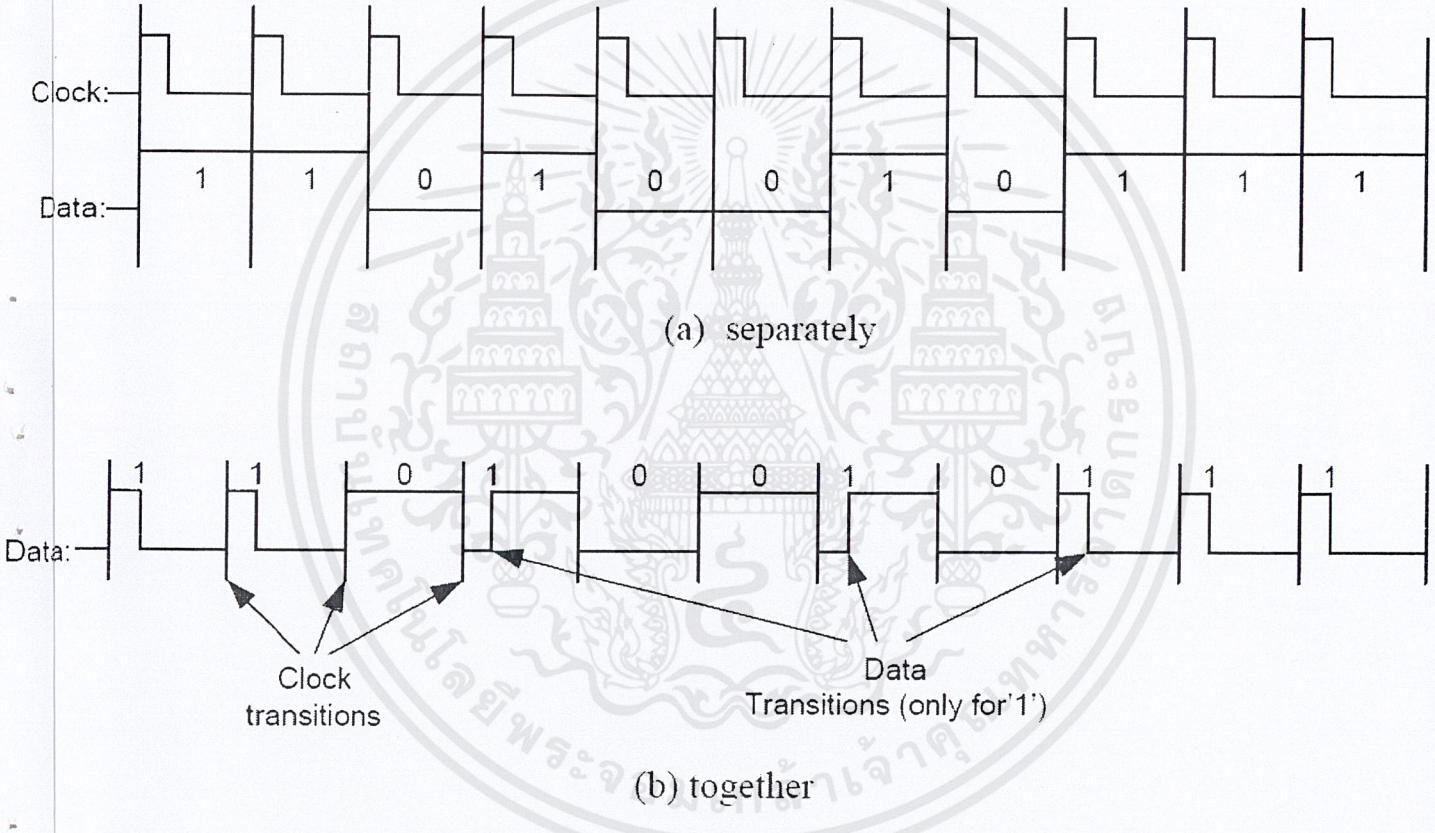
การส่งข้อมูลแบบซิงโครนัสเป็นการส่งข้อมูลแบบกลุ่ม โดยบิตที่ทยอยส่งเข้ามาจะมีการรวมกลุ่มกันให้มีขนาดใหญ่ขึ้นที่เรียกว่าเฟรมหรือบล็อกข้อมูล ซึ่งอาจจะประกอบด้วยหลายๆอักขระด้วยกัน โดยระหว่างการส่งจะปราศจากช่องว่าง รวมถึงบิตเริ่มต้นและบิตสิ้นสุด ซึ่งทำให้ไม่มีอะไรมาคั่นระหว่างข้อมูลแต่ละตัว จึงทำให้การกะจังหวะ(Timing) กลายเป็นสิ่งสำคัญมากสำหรับการส่งข้อมูลแบบซิงโครนัส กล่าวคือ ทั้งฝ่ายส่งข้อมูลและฝ่ายรับข้อมูลจะต้องทำงานสอดคล้องกันตามจังหวะของสัญญาณนาฬิกาโดยฝ่ายรับข้อมูลจะได้รับสัญญาณนาฬิกามาจากฝ่ายส่งข้อมูล

ดังนั้นวิธีการส่งข้อมูลชนิดนี้สัญญาณนาฬิกาทั้งฝ่ายรับละฝ่ายส่งจะต้องซิงโครไนซ์กันฝ่ายส่งอาจส่งสัญญาณนาฬิกาแยกออกมาพร้อมกับข้อมูลดังรูปที่2.23(a) แต่วิธีการนี้จะใช้ได้ดีกับระยะทางใกล้ๆ เพราะจะเกิดปัญหาเรื่องของสัญญาณที่อ่อนตัวลง แต่ก็มีอีกวิธีหนึ่งดังรูปที่ 2.23(b)ซึ่งจะเป็นการนำเอาสัญญาณนาฬิการวมเข้ากับสัญญาณข้อมูล เช่น การเข้ารหัสแบบแมนเชสเตอร์ในระบบแลน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



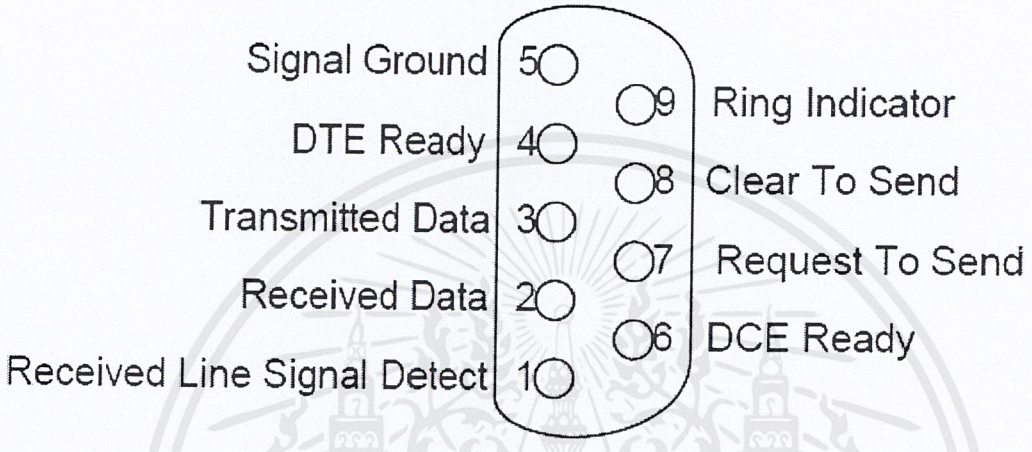
รูปที่ 2.32 ภาพแสดงการส่งข้อมูลแบบซิงโครนัสที่นำเสนอในรูปแบบง่าย



รูปที่ 2.33 ภาพแสดงการเข้าจังหวะระหว่างสัญญาณนาฬิกาและข้อมูลในการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

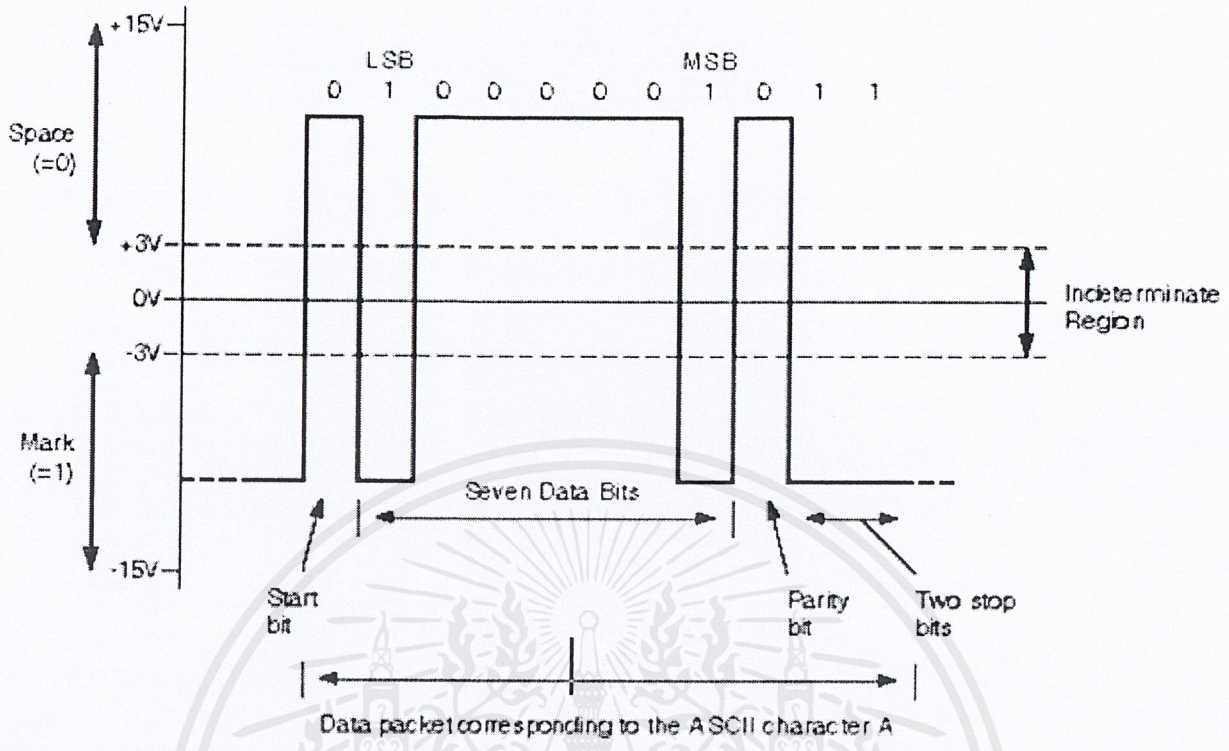
มาตรฐานRS232-C เป็นมาตรฐานในการรับและส่งข้อมูลระหว่าง DTE (Data TerminalEquipment) กับ DCE (Data Communication Equipment) โดยใช้เทคนิคการสื่อสารไบนารีแบบอนุกรม โดยการสื่อสารแบบอนุกรมที่นิยมใช้กับเครื่องคอมพิวเตอร์เป็นการสื่อสารแบบอะซิงโครนัส รูปที่ 2.24 แสดงลักษณะโครงสร้างของคอนเนคเตอร์ DB-9 ที่ใช้ในการติดต่อสื่อสาร



รูปที่ 2.34 แสดงโครงสร้างคอนเนคเตอร์ DB-9

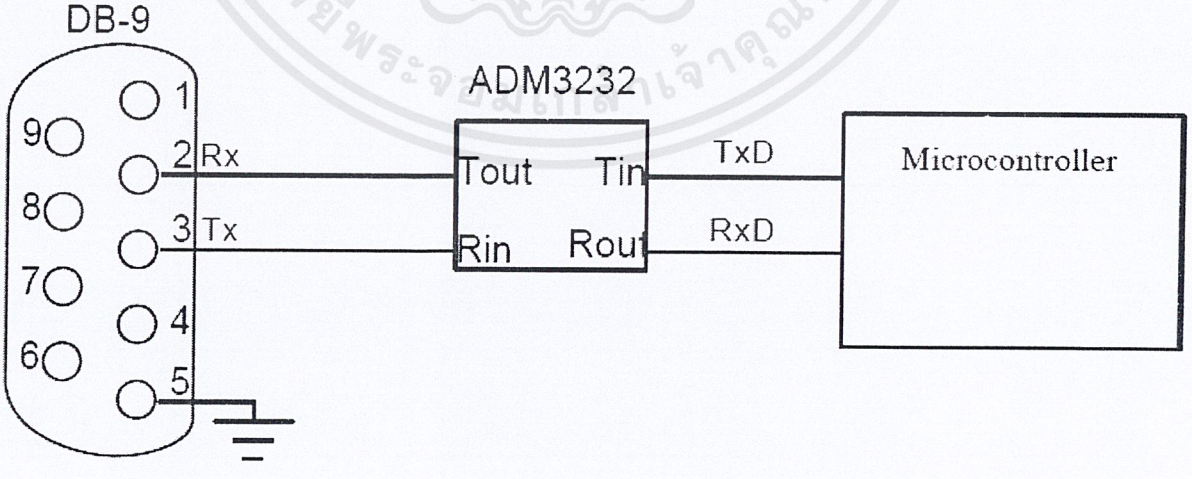
Received Line Signal Detect	: ขานี้จะ active เมื่อมีการส่งสัญญาณ Carrier จากโมเด็ม
Received Data	: ขานี้สำหรับรับข้อมูลอนุกรมจากคอมพิวเตอร์
Transmitted Data	: ขานี้สำหรับส่งข้อมูลอนุกรมจากคอมพิวเตอร์
DTE Ready	: ใช้บอกอุปกรณ์ปลายทางว่าต้องการติดต่อ
Signal Ground	: กราวด์ของวงจร
DCE Ready	: ใช้คู่กับขา DTE Ready ในการตรวจสอบการติดต่อ
Request To Send	: ใช้บอกอุปกรณ์ปลายทางให้ส่งข้อมูลกลับมาให้
Clear To Send	: ใช้ตรวจสอบว่าอุปกรณ์ปลายทางพร้อมส่งข้อมูลหรือไม่
Ring Indicator	: ขานี้จะ active เมื่อโมเด็มได้รับสัญญาณจากโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 แสดงระดับแรงดันที่ใช้ในการสื่อสารข้อมูลของ RS232-C

ในการเชื่อมต่อระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์นั้นแต่ละส่วนมีมาตรฐานต่างกันจึงต้องทำการแปลงแรงดันให้อยู่ในมาตรฐานการใช้งานของไมโครคอนโทรลเลอร์โดยใช้วงจรรวมเบอร์ ADM3232



รูปที่ 2.36 แสดงการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ผ่าน RS232-C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการออกแบบ

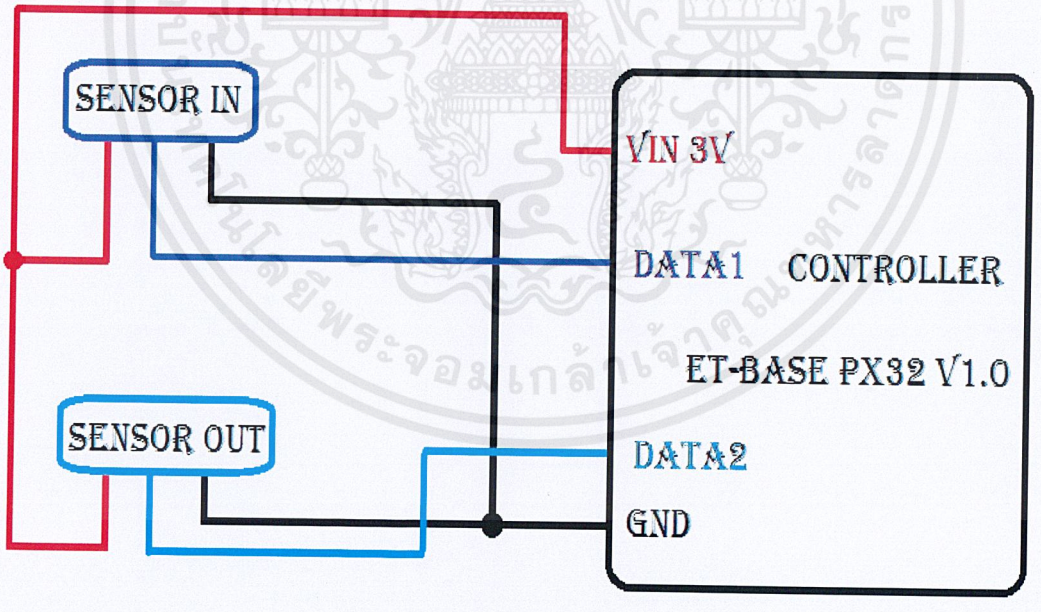
3.1 ส่วนประมวลผลหลัก

วงจรประมวลผลหลักนี้จะประกอบไปด้วย วงจรย่อยๆ 3 ส่วน คือ

- วงจรตรวจจับรถยนต์
- วงจรประมวลผลลานจอดรถ

3.1.1 วงจรตรวจจับรถยนต์เข้า-ออก

วงจรตรวจจับรถนี้ใช้เพื่อตรวจจับรถยนต์ที่วิ่งเข้า-ออกจากลานจอดรถและนับจำนวน แล้วส่งผลที่ได้เป็นข้อมูลออกทางหน้าจอ VGA หรือ TV ซึ่งวงจรดังกล่าวจะใช้วงจรเซนเซอร์แสงเป็นตัวตรวจจับ

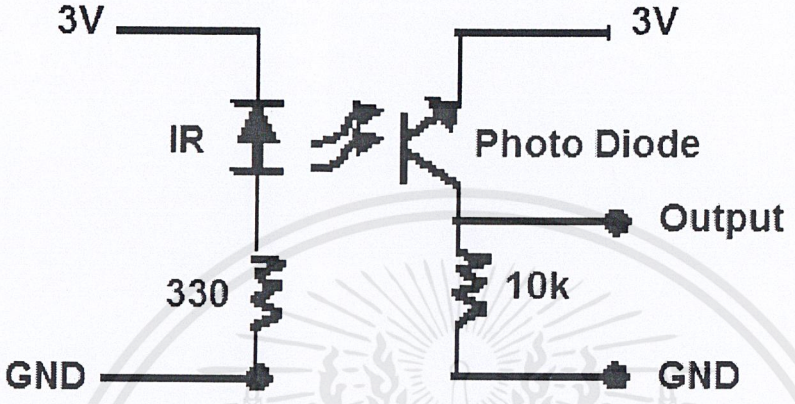


รูปที่ 3.1 วงจรตรวจจับรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.1 ส่วนประกอบของวงจรเซนเซอร์แสง

1. โฟโตไดโอดตัวรับและส่งในตัว
2. ตัวต้านทานขนาด 330Ω, 10kΩ

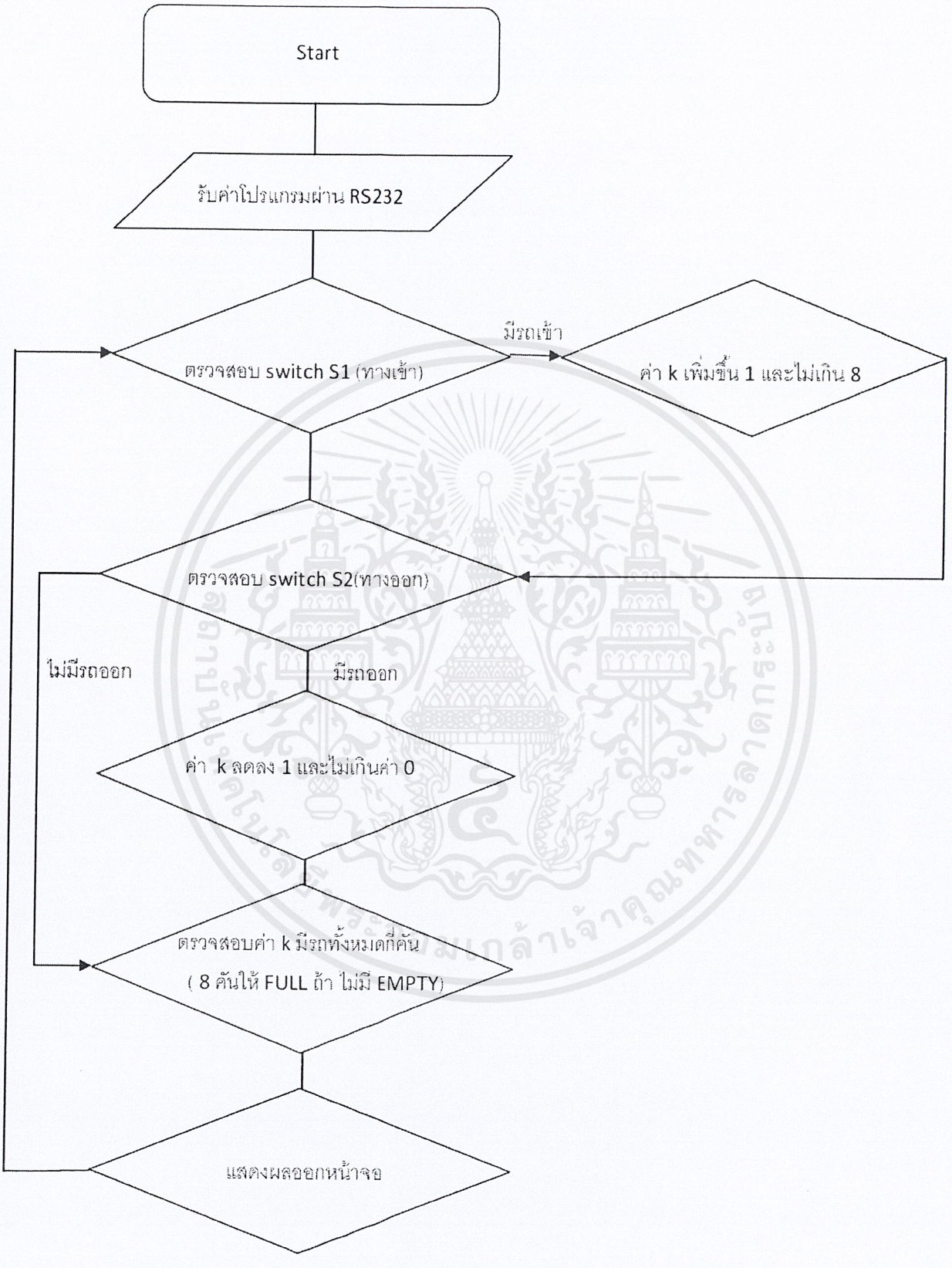


รูปที่ 3.1.1 วงจรเซนเซอ์แสง

3.1.1.2 หลักการทำงานวงจรตรวจจับรถยนต์เข้า-ออก

วงจรตรวจจับรถยนต์เข้า-ออกมีหลักการทำงาน คือ เซนเซอร์แสงที่ใช้แบบสำเร็จ คือมีทั้งตัวส่งและตัวรับในตัวถังเดียวกัน โดยจะมีตัวส่งซึ่งใช้เป็นหลอด LED อินฟราเรดส่งแสงอินฟาเรดไปกระทบวัตถุ หากวัตถุนั้นสามารถสะท้อนแสงได้ดีแสงดังกล่าวก็จะมาเป็นอินพุตให้ตัวรับแสงซึ่งในวงจรนี้ใช้เป็นโฟโต้ไดโอด จึงเกิดกระแสไหลผ่านที่ขาเบส ไปควบคุมกระแสให้ไหลจากขาคอลเลกเตอร์มายังขาอีมิตเตอร์มีปริมาณเพิ่มมากขึ้น ทำให้โฟโต้ไดโอดสามารถนำกระแสได้ กระแสนี้ทำให้มีแรงดันตกคร่อมตัวต้านทาน 10KΩเพิ่มขึ้น โดยค่าแรงดันจะมากหรือน้อยขึ้นอยู่กับปริมาณแสงอินฟราเรดที่โฟโต้ไดโอดได้รับจากLED อินฟราเรด แล้วนำผลที่ได้จากการเปลี่ยนแปลงของเซ็นเซอร์สามารถนำไปเข้าบอร์ดไมโครคอนโทรลเลอร์ บอร์ด ET- BASE PX32 V1.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

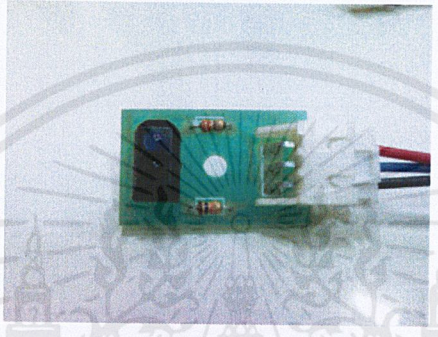


รูปที่ 3.2 โฟลว์ชาร์ทการทำงานวงจรตรวจจับรถยนต์เข้า-ออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.3 การติดตั้งเซนเซอร์แสงนับจำนวนรถที่วิ่งเข้าออก

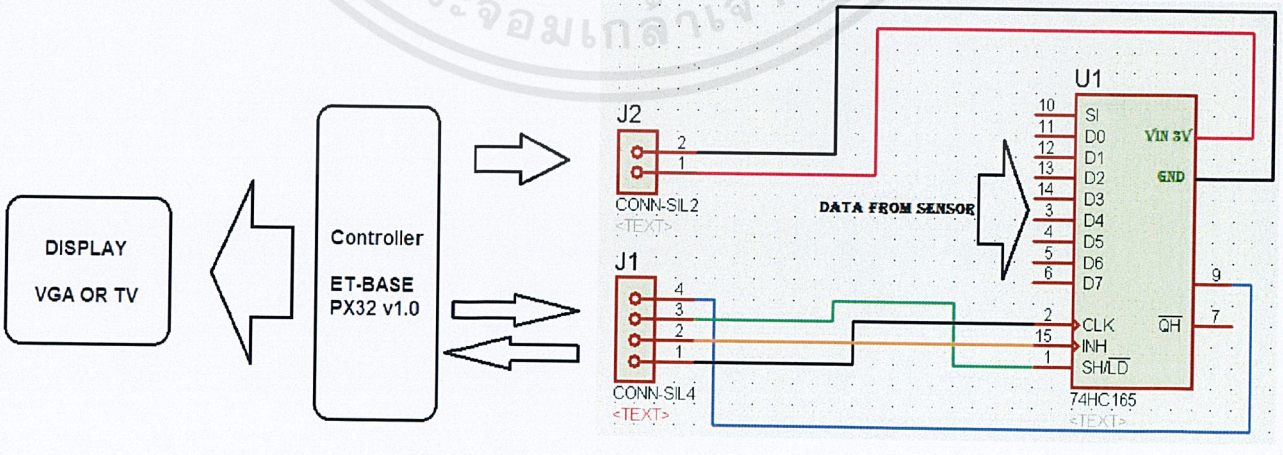
เนื่องจากการนับจำนวนรถที่วิ่งเข้าออกนั้นอาจเกิดการผิดพลาดขึ้นได้เนื่องจากสิ่งรบกวนจากสิ่งแวดล้อมต่างๆ เช่น จะสามารถแยกแยะได้อย่างไรว่าสิ่งที่นับอยู่เป็นรถจริงหรือไม่ จึงต้องออกแบบการติดตั้งวงจรเซนเซอร์ให้สามารถแยกได้ว่าสิ่งที่นับเป็นรถ โดยการติดเซนเซอร์แสงจำนวนสองชุดในระยะห่างกันพอสมควรเพื่อแยกขนาดวัตถุ หากเซนเซอร์สามารถตรวจจับวัตถุได้พร้อมกันจึงจะทำการนับ



รูปที่3.3 เซนเซอร์แสงนับจำนวน

3.1.2 วงจรประมวลผลลานจอดรถ

วงจรประมวลผลลานจอดรถจะเป็นส่วนที่รับข้อมูลจากเซนเซอร์ตามที่จอดรถแต่ละช่องจอ แล้วแสดงแผนภาพลานจอดรถออกทางหน้าจอVGAหรือTV โดยแสดงภาพที่จอดรถใดว่างหรือไม่ว่างอยู่บ้างเพื่อสะดวกต่อผู้ใช้บริการที่จอดรถในลานจอดรถนั้นๆ

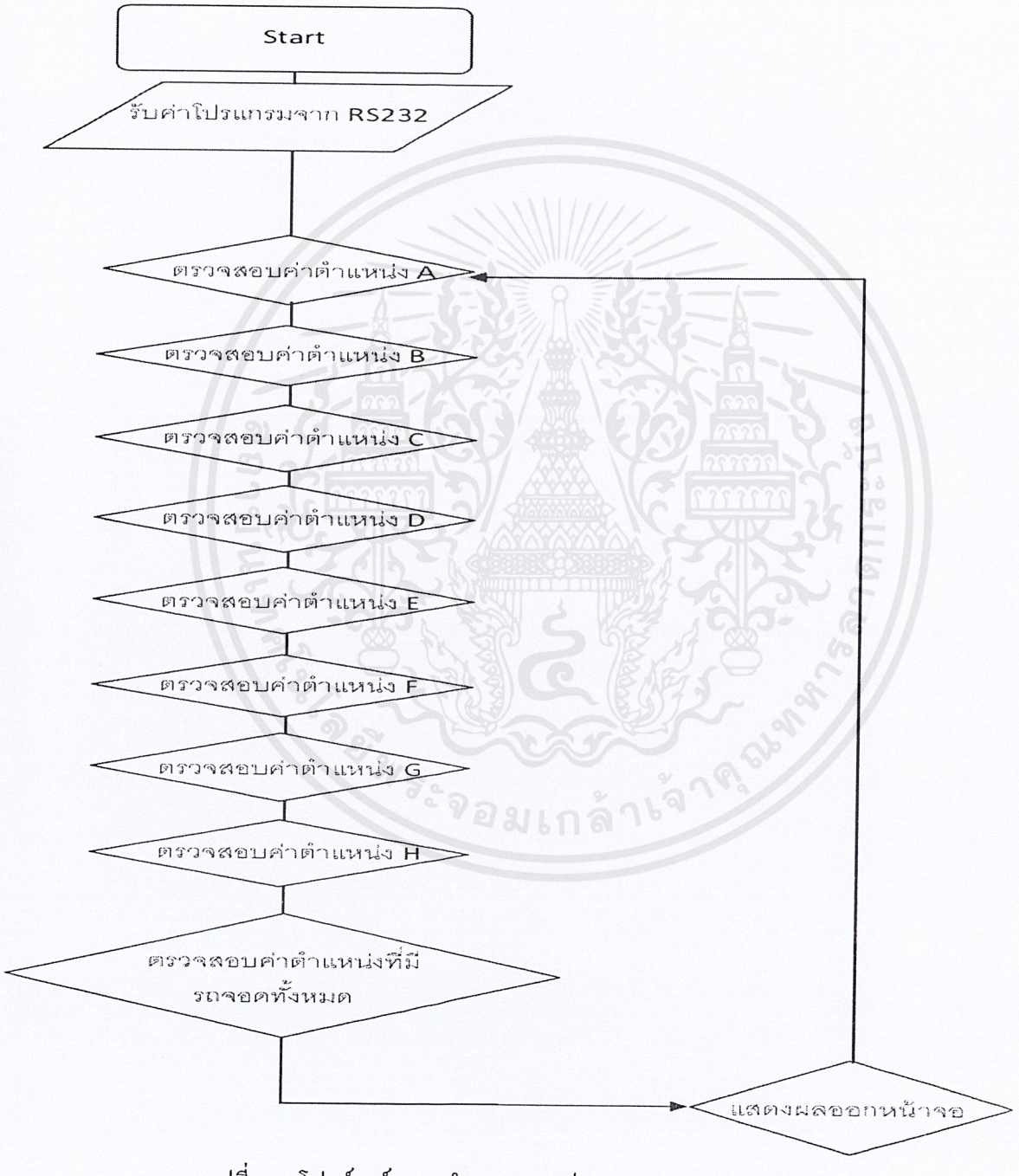


รูปที่3.4 บล๊อกไดอะแกรมวงจรประมวลผลลานจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1 ระบบการทำงาน

ในส่วนนี้ IC 74HC165N จะรับข้อมูลที่ได้จากเซนเซอร์ตามที่จอดรถในแต่ละช่องจอดรถ โดยรับข้อมูลเข้ามาเป็นแบบขนาน จากนั้น IC 74HC165N จะทำหน้าที่ส่งข้อมูลออกเป็นแบบอนุกรมส่งให้ตัวบอร์ดคอนโทรลเลอร์(ET-BASE PX32 v1.0) เพื่อทำการประมวลผลภาพออกหน้าจอ VGA หรือ TV แล้วแต่ตามต้องการ



รูปที่ 3.5 โฟลว์ชาร์ทการทำงานวงจรประมวลผลลานจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ตารางที่ 4.1 การทดลอง

การทดลองครั้งที่	เซนเซอร์ทางเข้า	เซนเซอร์ทางออก	เซนเซอร์ในลานจอดรถ
1	-	-	-
2	มีรถเข้า 8 คัน	-	มีรถจอดที่ตำแหน่ง A,B,D,E และ G และอีก 3 คันกำลังหาที่จอดอยู่
3	มีรถเข้า 8 คัน	มีรถออก 6 คัน	มีรถจอดที่ตำแหน่ง C และอีก 1คัน กำลังหาที่จอดอยู่

การทดลองครั้งที่ 1 ยังไม่นำรถจำลองเข้าลานจอดรถ สังเกตผลลานจอดรถและจำนวนรถที่หน้าจอแล้วบันทึกผล

การทดลองครั้งที่ 2 นำรถเข้าลานจอดรถ 8 คัน โดยไม่มีการนำรถออกที่ทางออก นำรถไปจอดที่ตำแหน่ง A,B,D,E และ G ตามลำดับ สังเกตผลลานจอดรถและจำนวนรถที่หน้าจอแล้วบันทึกผล

การทดลองครั้งที่ 3 นำรถเข้าลานจอดรถ 8 คันก่อน แล้วจึงนำรถออกจากลานจอดรถ 6 คัน(เหลือรถอยู่ในลานจอดรถ 2 คัน) นำรถที่เหลือจอดที่ตำแหน่ง C สังเกตผลลานจอดรถและจำนวนรถที่เปลี่ยนแปลงที่หน้าจอแล้วบันทึกผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดลอง

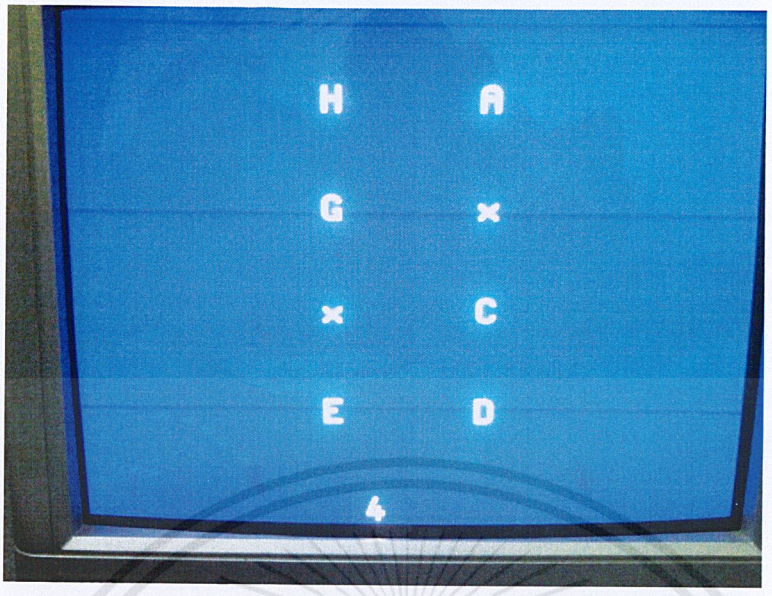
ผลการทดลองครั้งที่	ตัวเลขแสดงจำนวนรถเข้า-ออก	ภาพแสดงตำแหน่งที่มีรถจอดและตำแหน่งที่ไม่มีรถจอด
1	EMPTY	แสดงอักษร A,B,C,D,E,F,G,H บนหน้าจอ VGA (ไม่มีรถคันใดจอดอยู่)
2	FULL	แสดงอักษร x,x,C,x,x,F,x,H บนหน้าจอ VGA (มีรถจอดอยู่ที่ตำแหน่ง A,B,D,E และ G)
3	2	แสดงตัวอักษร A,B,x,D,E,F,G,H บนหน้าจอ VGA (มีรถจอดอยู่ที่ตำแหน่ง C)

ผลการทดลองครั้งที่ 1 ภาพลานจอดรถบนหน้าจอแสดงอักษร A,B,C,D,E,F,G,H ตามลำดับ และบริเวณตำแหน่งที่แสดงตัวเลขรถเข้าออกบนหน้าจอแสดงคำว่า EMPTY

ผลการทดลองครั้ง 2 ภาพลานจอดรถบนหน้าจอแสดงอักษร x ตรงบริเวณตำแหน่งที่จอด A,B,D,E และ G ตามลำดับ และบริเวณตำแหน่งที่แสดงตัวเลขรถเข้าออกบนหน้าจอแสดงคำว่า FULL

ผลการทดลองครั้งที่ 3 ภาพลานจอดรถบนหน้าจอแสดงอักษร x ตรงบริเวณตำแหน่งที่จอด C และแสดงตัวเลข 2 ตรงบริเวณตำแหน่งแสดงจำนวนรถเข้าออกบนหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 หน้าจอแสดงจำลองผลการทดลอง

รูปที่ 4.2 แสดงภาพเหตุการณ์จำลองการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1สรุปผลการทดลอง

จากการทดลองที่ 1 ขอสรุปว่า โปรแกรมนับรถเข้า-ออกสามารถทำงานได้ตามที่ต้องการ สามารถตรวจจับจำนวนช่องสูงสุด 8 ช่อง โดยใช้บอร์ดET Base PX32 v1.0 โดยมีการทำงาน คือจะทำการนับขึ้นทีละ 1 เมื่อมีรถขับเข้าผ่านเซนเซอร์ที่ทางเข้า และจะทำการนับลงทีละ 1 เมื่อมีรถขับออกผ่านเซนเซอร์ที่ทางออก เนื่องจากมีที่จอดรถจำนวน 8 ช่อง โปรแกรมที่นับขึ้น-นับลงจึงนับได้ถึง 8 คัน แต่จะโชว์เลขที่นับได้สูงสุดแค่เลข 7 เพราะเลข 8 จะถูกแทนด้วยคำว่า “FULL” และ “EMPTY” ออกทางหน้าจอแสดงผล

สามารถทำการขยายจำนวนช่องที่ตรวจจับได้โดยการเพิ่มIC 74165 โดยการต่ออนุกรมกันไปเรื่อยๆ

จากการทดลองที่ 2 ขอสรุปว่า โปรแกรมตรวจจับรถยนต์สามารถทำงานได้ตามที่ต้องการ โดยมีรถเข้าจอดในช่องจอดรถใดๆ หน้าจอจะแสดงผลเป็นตัว X ซึ่งจะหมายความว่าช่องจอดรถนั้นมีรถจอดอยู่ แต่หากหน้าจอแสดงเป็นชื่อช่องก็จะหมายถึงช่องจอดรถนั้นว่างอยู่

5.2ปัญหาในการดำเนินงาน

1. เนื่องจากเซนเซอร์ที่ใช้ เป็นเซนเซอร์แสงที่มีหลักการทำงานแบบสะท้อนแสง มีการสะท้อนแสงไม่คงที่ เมื่อมีการเคลื่อนที่ของวัตถุเพียงนิดเนียวมุมของการสะท้อนแสงจะเลื่อนไปด้วย ทำให้รับค่าได้ผิดพลาด

2. สวิตช์ที่ใช้บางตัว ตัวสวิตช์ปุ่มกดสลับกับสิ่งที่เขียนบอกบนตัวสวิตช์ ทำให้เกิดการเสียเวลาในการทำงาน ต้องมีการตรวจสอบอย่างละเอียด โดยไล่ตรวจสอบโมเดลทั้งหมด

3.การจ่ายไฟเลี้ยงเซนเซอร์ แรงดันไฟที่พาวเวอร์ซัพพลายแสดงนั้นไม่ตรงกับค่าแรงดันที่จ่ายออกมาจริงและไม่คงที่ ทำให้เซนเซอร์ทำงานผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Code วงจรตรวจจับนับรถยนต์เข้า-ออก

```
CON

_clkmode = xtal1 + pll16x

_xinfreq = 5_000_000

OBJ

 text : "vga_text"

VAR

byte n

byte o

long k


PUB SENSOR

 text.start(16)

 text.str(string($A,14,$B,14))

 text.str(string("EMPTY"))

 repeat

 n:=ina[5]

 o:=ina[6]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if n==1
 n:=ina[5]
 if n==0
  k++
  text.str(string($A,14,$B,14))
  if k=>8
   k:=8
   text.str(string("     "))
   text.str(string($A,14,$B,14))
   text.str(string("FULL"))
  else
   text.str(string("     "))
   text.str(string($A,14,$B,14))
   text.dec(k)
if o==1
 o:=ina[6]
 if o==0
  --k
  text.str(string($A,14,$B,14))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if k=<0

 k:=0

 text.str(string("     "))

 text.str(string($A,14,$B,14))

 text.str(string("EMPTY"))

else

 text.str(string("     "))

 text.str(string($A,14,$B,14))

 text.dec(k)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code วงจรตรวจจับรถยนต์ในช่องจอดรถ

```
CON

_clkmode = xtal1 + pll16x

_xinfreq =25_000_000


OBJ

 text : "vga_text"


VAR

byte x[8]

byte i

byte j

byte n

byte o

long k


PUB Main

 dira[0]~~

 dira[1]~~
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 dira[2]~~

 text.start(16)

 text.str(string($A,14,$B,14))

 text.str(string("EMPTY"))

repeat

 outa[0]:=0

 waitcnt(800_000+cnt)

 outa[0]:=1

 waitcnt(800_000+cnt)

 i++

 if i==1

 outa[1]:=1

 outa[2]:=1

if i==3

 outa[2]:=0

if i==4

 outa[2]:=1

if i=>6

 outa[1]:=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
x[j]:=ina[4]

j++

if j==8

 text.str(string($A,9,$B,2))

 if x[3]==1

  text.str(string("  x  "))

 else

  text.str(string("  H  "))

 if x[4]==1

  text.str(string("  x  "))

 else

  text.str(string("  A  "))

text.str(string($A,9,$B,5))

if x[2]==1

 text.str(string("  x  "))

else

 text.str(string("  G  "))

if x[5]==1

 text.str(string("  x  "))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 else

  text.str(string("  B  "))

 text.str(string($A,9,$B,8))

 if x[1]==1

  text.str(string("  x  "))

 else

  text.str(string("  F  "))

 if x[6]==1

  text.str(string("  x  "))

 else

  text.str(string("  C  "))

 text.str(string($A,9,$B,11))

 if x[0]==1

  text.str(string("  x  "))

 else

  text.str(string("  E  "))

 if x[7]==1

  text.str(string("  x  "))

 else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  text.str(string("   D   "))

i:=0

j:=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้