

ระบบตรวจสอบและจัดเก็บข้อมูลเครื่องจักร
(CHECKING AND MACHINE DATA STORAGE SYSTEM)



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

นพ.
ก. 334 ข
SEA

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหมึก.....
เลขทะเบียน... 46418
วัน, เดือน, ปี... 1 มิ.ย. 2546

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

ผู้ปฏิบัติงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHECKING AND MACHINE DATA STORAGE SYSTEM

By

Mr. KASAMA RUKSA
Mr. SURATAP SANGAROON

Project Report Submitted in Partial Fulfillment of the Requirements

For the Bachelor's Degree

Department of Information Engineering

Faculty of Engineering

King Mongkut's Institute of Technology Ladkrablang

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

ระบบตรวจสอบและจัดเก็บข้อมูลเครื่องจักร

โดย

นาย กษม รักษา

นาย สุรเทพ แสงอรุณ

อาจารย์ผู้ควบคุมปริญญานิพนธ์

อาจารย์ มนต์ชัย แซ่มซ้อย

ระดับการศึกษา

ปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีโทรคมนาคม

วิศวกรรมสารสนเทศ

ปีการศึกษา

2544

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
อุตสาหกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง

.....
อาจารย์มนต์ชัย แซ่มซ้อย

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

ระบบตรวจสอบและจัดเก็บข้อมูลเครื่องจักร

โดย

นาย กษม รักษา รหัสประจำตัว 43015799

นาย สุรเทพ แสงอรุณ รหัสประจำตัว 43015848

อาจารย์ผู้ควบคุมปริญญานิพนธ์

อาจารย์ มนต์ชัย แซ่มซ้อย

ระดับการศึกษา

ปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีโทรคมนาคม

วิศวกรรมสารสนเทศ

ปีการศึกษา

2544

บทคัดย่อ

ในปริญญานิพนธ์ฉบับนี้ได้นำเสนอระบบจัดเก็บและวิเคราะห์ข้อมูล สำหรับเครื่องจักร ในโรงงานอุตสาหกรรม เพื่อทำแผนสำหรับการซ่อมบำรุงได้อย่างมีประสิทธิภาพ โครงสร้างของระบบประกอบด้วยสองส่วน คือ ส่วนของการรับข้อมูลที่เป็นอาการเสียต่างๆ เข้าสู่ระบบการจัดเก็บข้อมูล โดยใช้ไมโครคอนโทรลเลอร์ (MCS-51) เพื่อสร้างรหัสพิเศษ โดยข้อมูลรหัสพิเศษเหล่านี้จะถูกส่งสู่ส่วนประมวลผล โดยการเชื่อมต่อแบบมาตรฐานการส่งอนุกรม RS-232 และส่วนที่สอง ซึ่งเป็นส่วนของไมโครคอมพิวเตอร์ที่จะรับข้อมูลรหัสพิเศษที่ได้มาจากส่วนแรกมาทำการเปลี่ยนข้อมูลให้อยู่ในรูปแบบของฐานข้อมูล เพื่อที่จะสามารถเรียกข้อมูลเหล่านั้นมาวิเคราะห์ได้อย่างมีประสิทธิภาพ จากผลการทดลอง ระบบสามารถจัดเก็บและวิเคราะห์ข้อมูลการเสียของเครื่องจักรได้ตามวัตถุประสงค์ ระบบนี้จะช่วยเพิ่มความสะดวกต่อผู้ใช้งาน และนำไปสู่การวางแผนเพื่อบำรุงรักษาเครื่องจักรเหล่านั้นได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THESIS TITLE	CHECKING AND MACHINE DATA STORAGE SYSTEM
STUDENT	Mr. KASAMA RUGSA NO.43015799
	Mr. SURATAP SANGAROON NO.43015848
ADVISOR	Mr. MONCHAI CHAMCHOY
COURSE	Bachelor of Industrial Technology in Telecommunication
DEPARTMENT	Information Engineering
YEAR	2001

Abstract

In this project, the machine data acquisition and system analysis is presented. The system is designed to provide the effective planning for maintenance. It consists of two major parts , the data acquisition and the computer analysis. The former one is used to enter the error codes into the storage system via microcontroller (MCS-51). These error codes are encoded and transmitted to the processing unit with RS-232 serial communication. The later is microcomputer which is for processing the encoded error codes and converting it to database format. From the experimental result , it is shown that the system can solve the incurred problem effectively.

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	VII
บทที่ 1.	
1.1 บทนำ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินโครงการ	2
บทที่ 2.	
ทฤษฎี	3
2.1 ระบบคอมพิวเตอร์พื้นฐาน	3
2.2 ไมโครคอนโทรลเลอร์ MCS-51	4
2.2.1 โครงสร้างของไมโครคอนโทรลเลอร์	4
2.2.2 โครงสร้างและสถาปัตยกรรมของ MCS-51 แบบแฟลช	7
2.2.3 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	9
2.2.4 โครงสร้างและการทำงานของพอร์ต	11
2.2.5 หน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช	15
2.2.6 แอ็กคิวเมเตเตอร์	18
2.3. CP-AT32 Plus	20
2.3.1 Specification	20
2.3.2 รายละเอียดเบื้องต้น ของ CP-AT32Plus	21
2.3.3 Features of board.	22
2.3.4 Features of CPU	23
2.3.5 Memory Map	24
2.3.6 ส่วนต่างๆของ CP-AT32Plus	25
2.4. มาตรฐานRS-232	29
2.4.1 การสื่อสารทางเดียว	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	2.4.2 การสื่อสารสองทาง	33
	2.4.3 โมเด็ม (Null Modem)	35
บทที่ 3.	หลักการออกแบบ	36
	3.1 ส่วนต่างๆของ โครงงาน	36
	3.2 หลักการออกแบบ SOFTWARE	38
บทที่ 4.	ผลการทดลอง	41
บทที่ 5	สรุปผลและวิจารณ์	43
บรรณานุกรม		44
ภาคผนวก	ก. ไฟลชวาร์ต ข. โปรแกรมการทำงาน ค. วงจรและ PORT ต่างๆ ง. DATA SHEET	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 อุปกรณ์พื้นฐานของไมโครคอนโทรลเลอร์	5
รูปที่ 2.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์	6
รูปที่ 2.3 แสดงตำแหน่งขั้วต่อของ 8255	26
รูปที่ 2.4 ตำแหน่งคอนเน็คเตอร์ 34 PIN	26
รูปที่ 2.5 แสดงขั้วต่อแบบ IDC ขนาด 10 PIN	27
รูปที่ 2.6 แสดงการใช้ RS-232-C เชื่อมต่ออุปกรณ์	29
รูปที่ 2.7 แสดงการเชื่อมต่อทางเดียวของ RS-232	31
รูปที่ 2.8 แสดงการสื่อสารทางเดียวจาก DTE ไปยัง DCE	33
รูปที่ 2.9 แสดงการสื่อสารทางเดียวจาก DCE ไปยัง DTE	33
รูปที่ 2.10 แสดงการสื่อสารสองทางด้วยวงจรแฮนด์เช็กกึ่งหลัก	34
รูปที่ 3.1 BLOCK DIAGRAM ของ HARDWARE	36
รูปที่ 3.2 BLOCK DIAGRAM ของ SOFTWARE	37
รูปที่ 4.1 ตัวอย่าง PROGRAM ที่ทดลองติดต่อกับฐานข้อมูล	41
รูปที่ 4.1 ตัวอย่าง PROGRAM ที่ทดลองติดต่อกับ ไมโครคอนโทรลเลอร์	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์	6
ตารางที่ 2.2 รายละเอียดของ MCS-51 แบบแฟลช	9
ตารางที่ 2.3 หน้าที่พิเศษของ PORT 1 ของ MCS-51 แบบแฟลช	13
ตารางที่ 2.4 สแต็กพอยน์เตอร์	19
ตารางที่ 2.5 SPECIFICATION ของ CP-AT32 PLUS	20
ตารางที่ 2.6 กำหนดค่า REGISTER AUXR	24
ตารางที่ 2.7 ADDRESS ของพอร์ต 8255	25
ตารางที่ 2.8 ขอบคลุมของ NC6B595,74HC595,8255ตัวที่ 3	27
ตารางที่ 2.9 แสดงการเชื่อมต่อ RS-232 มาตรฐานเก้าเส้น	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ สำเร็จลุล่วงลงไปได้ด้วยความร่วมมือและความตั้งใจในการทำงานของเพื่อนร่วมกลุ่ม ทั้งได้รับคำแนะนำจากอาจารย์ที่ปรึกษา และแนวความคิดจากรุ่นพี่ และยังคงตอบคำถามเวลาที่ทำงานติดขัดหรือประสบปัญหาต่างๆ อีกเพื่อนๆที่คอยถามความคืบหน้าและขอบพระคุณ พี่ทิวี่ที่ช่วยตอบปัญหาและให้คำแนะนำ คณะผู้จัดทำขอกล่าวขอบคุณไว้ ณ โอกาสนี้

นาย กษม รักษา
นาย สุรเทพ แสงอรุณ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในโรงงานอุตสาหกรรมส่วนใหญ่ นั้นจะมีเครื่องจักรใหญ่หลายตัว และมักจะมีข้อผิดพลาด... บางประการของเครื่องจักรในเวลาทำงาน และจะมีช่างคอยตรวจเช็คเครื่องเป็นกะอยู่แล้วถ้ามีปัญหาเกิดขึ้นกับเครื่องจักร ช่างประจำก็จะลงเวลาและปัญหาของเครื่องเอาไว้ในใบบันทึกประจำวัน แต่ระบบนี้จะทำการลงเวลาและปัญหาในการหยุดเครื่อง โดยที่เมื่อช่างพบปัญหาเกิดขึ้น ก็จะไปป้อนรหัสที่เป็น code ลงในคีย์บอร์ดของบอร์ดไมโครคอนโทรลเลอร์ เพื่อที่จะส่งต่อไปยัง ส่วนของคอมพิวเตอร์เพื่อจัดเก็บข้อมูลปัญหาของเครื่องจักรเอาไว้ และจะมีประโยชน์ในการประมวลผลในแต่ละเดือนถึงปัญหาของเครื่องจักรแต่ละตัว เพื่อเป็นประโยชน์ในการซ่อมแซม

1.2 วัตถุประสงค์

1. เพื่อการศึกษการเชื่อมต่อไมโครคอนโทรลเลอร์ให้สามารถสื่อสารกับพอร์ตคอมพิวเตอร์ได้
2. เพื่อที่จะใช้โครงการนี้มาจัดเก็บข้อมูลของเครื่องจักรแต่ละตัว ถึงแต่ละสาเหตุการเสียที่แตกต่างกันเพื่อจัดเก็บไว้
3. เพื่อที่สามารถจะนำข้อมูลที่จัดเก็บนั้น นำมาทำการประมวลผลในแต่ละช่วงเวลาได้เพื่อที่จะเป็นประโยชน์ในการซ่อมแซมบำรุงรักษาได้ถูกจุดและถูกต้อง

1.3 ขอบเขตของโครงการ

โครงการนี้สามารถจะนำไปใช้ได้จริง แต่อาจจะต้องประยุกต์ให้เหมาะสมกับระบบของเครื่องจักรที่ใช้ สามารถที่จะเพิ่มสาเหตุการเสียต่างๆได้ตามลักษณะเครื่องจักรที่ใช้ทำงาน แต่โครงการนี้นั้นจะเป็นการจำลองการใช้งานเครื่องจักรหนึ่งตัว ซึ่งสมมุติให้เป็นเครื่องผลิตพลาสติกซึ่งจะมีระบบต่างๆ เช่น

1. ระบบแหล่งจ่ายไฟฟ้า
2. ระบบควบคุม
3. ระบบไฮดรอลิก
4. ระบบหล่อเย็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเราจะเอาระบบเหล่านี้มาหาสาเหตุที่จะทำให้เกิดปัญหาเช่น ปัญหาเกี่ยวกับระบบควบคุม
 จำกัด ซึ่งจะทำให้เครื่องจักรนั้นจะต้องหยุดชะงักลง และจากตรงนี้เองที่โครงการนี้จะเริ่มมีบทบาท
 ขึ้นมาโดยที่เมื่อเครื่องจักรเสีย ช่างประจำเครื่องก็จะสามารถบันทึกสาเหตุของการเสียได้โดย
 ผ่านทาง KEYPAD ซึ่งจะส่งค่าไปโดยใช้ไมโครคอนโทรลเลอร์เพื่อนำค่าที่เป็นตัวแปรนี้ไปเก็บ
 ไว้ในฐานข้อมูลของ COMPUTER ซึ่งจะสามารถบันทึก วัน/เดือน/ปี และเวลาของสาเหตุการเสีย
 ของเครื่องจักรเอาไว้เพื่อรอเวลาที่ให้นำข้อมูลเหล่านั้นมาประเมินผลได้ ซึ่งจะเป็นประโยชน์ในการ
 ซ่อมและบำรุงรักษาต่อไปในอนาคตได้

1.4 ขั้นตอนการดำเนินงาน

การทำงานในโครงการนี้มีขั้นตอนการทำงานที่จะสามารถแบ่งออกเป็นสองส่วนคือ

1.4.1 HARDWARE

- โดยเริ่มจากการเตรียม BOARD MICROCONTROLLER MCS-51 ซึ่งจะเป็น BOARD
 MICROCONTROLLER ของ ETT ซึ่งจะมี MCS-51 นั้นเป็น CPU ซึ่งจะมี PORT ของ RS-232 ที่จะ
 เชื่อมต่อกับ SERIAL PORT มาให้ด้วย และมี PORT ที่สามารถที่จะต่อกับ KEYPAD ได้

- เตรียม COMPUTER เอาไว้สำหรับการเขียนโปรแกรม

1.4.2 SOFTWARE

- เริ่มด้วยการศึกษาโปรแกรมที่จะใช้เขียนโปรแกรม

- เริ่มเขียนโปรแกรมที่จะควบคุมไมโครคอนโทรลเลอร์ MCS-51 ซึ่งจะใช้ภาษา C โดยเริ่ม
 แรกจะเริ่มเขียนการสแกน KEY ทุกคีย์ที่ต่อกับ PORT ของไมโครคอนโทรลเลอร์และส่งต่อออกมา
 ที่ PORT RS-232 ต่อไป

- เขียนโปรแกรมที่จะทำการติดต่อระหว่างไมโครคอนโทรลเลอร์ กับ SERIAL PORT ของ
 COMPUTER โดยจะใช้โปรแกรม Microsoft Visual Basic 6.0 เขียนเพื่อที่จะทดลองการรับค่าจาก
 ไมโครคอนโทรลเลอร์มาแสดงผลในโปรแกรม

- เขียนโปรแกรมที่จะมาใช้งานจริง โดยใช้โปรแกรม Delphi 5 และนำ COMPONENT
 ของ MS COM ของ Microsoft Visual basic 6.0 มาร่วมใช้ด้วยกัน และเขียนโปรแกรมให้รับค่าที่ส่ง
 จากไมโครคอนโทรลเลอร์ MCS-51 ได้

- เขียนโปรแกรมฐานข้อมูลขึ้นมาเพื่อจะรองรับค่าเพื่อจะแปลงค่าที่ส่งมาเป็นตัวแปร โดย
 ระหว่างโปรแกรมหลักและตัวฐานข้อมูล DATA DESKTOP จะมีตัวเชื่อมคือ BDE ADMINISTER

- ทดลองส่งค่าและออกแบบลักษณะหน้าต่างของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

2.1 ระบบคอมพิวเตอร์พื้นฐาน

คอมพิวเตอร์นั้นมีส่วนประกอบที่สำคัญเช่น หัวใจหลักคือ หน่วยประมวลผล กลางหรือ CPU (Central processing unit) จะทำหน้าที่ประมวลผลข้อมูลต่าง ๆ เพื่อนำผลลัพธ์ที่ได้ไปทำการควบคุมระบบต่อไป ภายใน CPU ประกอบด้วย

1. หน่วยจัดการทางคณิตศาสตร์ (Arithmetic Logic Unit : ALU) ทำหน้าที่จัดการด้านการคำนวณทางคณิตศาสตร์และลอจิก แล้วนำผลลัพธ์หรือข้อมูลที่ได้ส่งให้แก่ CPU เพื่อประมวลผล และควบคุมการทำงานต่อไป

2. หน่วยเก็บข้อมูลชั่วคราว หรือ รีจิสเตอร์ (Register) ทำหน้าที่เก็บข้อมูล เพื่อเตรียมเข้าสู่กระบวนการประมวลผล เก็บข้อมูลที่ได้จากการประมวลโดย ALU นำข้อมูลเพื่อส่งออกไปยังอุปกรณ์ภายนอก

3. แฟล็ก (Flag) ทำหน้าที่แสดงภาวะการทำงานของ CPU ว่าขณะนี้เกิดอะไรขึ้นมีสถานะลอจิกเป็นอย่างไร ซึ่งสามารถใช้ประโยชน์จากแฟล็กนี้ในการตั้งเงื่อนไขในการทำงานของระบบคอมพิวเตอร์

4. สแต็กและโปรแกรมเคาท์เตอร์ (Stack & Program counter) สแต็กทำหน้าที่เป็นที่พักข้อมูล เมื่อ CPU ว่าขณะนี้ CPU ทำงานไปถึงแอดเดรสใด หรือเป็นเสมือนมอนิเตอร์แสดงตำแหน่งการทำงานของ CPU นั้นเอง

ส่วนที่สอง คือ หน่วยความจำซึ่งมีทั้งแบบรอม (Read Only Memory : ROM) และแรม (Read Access Memory : Ram) โดยรอมจะทำหน้าที่เก็บโปรแกรมควบคุมการทำงานและควบคุมฮาร์ดแวร์ทั้งหมดหรือที่เรียกว่า โปรแกรมมอนิเตอร์ (monitor program) และหน่วยความจำแรมใช้เป็นที่พักข้อมูลระหว่างรอการประมวลผล หรือเก็บข้อมูลเพื่อเตรียมการส่งออกไปยังอุปกรณ์ภายนอก

ส่วนพอร์ตอินพุตจะใช้เชื่อมต่ออุปกรณ์อินพุตที่จะส่งข้อมูลมายังคอมพิวเตอร์ได้แก่ คีย์บอร์ด เป็นต้น

ส่วนพอร์ตเอาต์พุตจะใช้เชื่อมต่ออุปกรณ์เอาต์พุตเพื่อแสดงผลการทำงาน ได้แก่ จอคอมพิวเตอร์ เครื่องพิมพ์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่ง ที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันคือ "ไมโคร" (micro) ซึ่งหมายถึงไมโครโปรเซสเซอร์ (microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วย หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า "คอนโทรลเลอร์" (controller) หมายถึง อุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียน โปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.2.1 โครงสร้างของไมโครคอนโทรลเลอร์

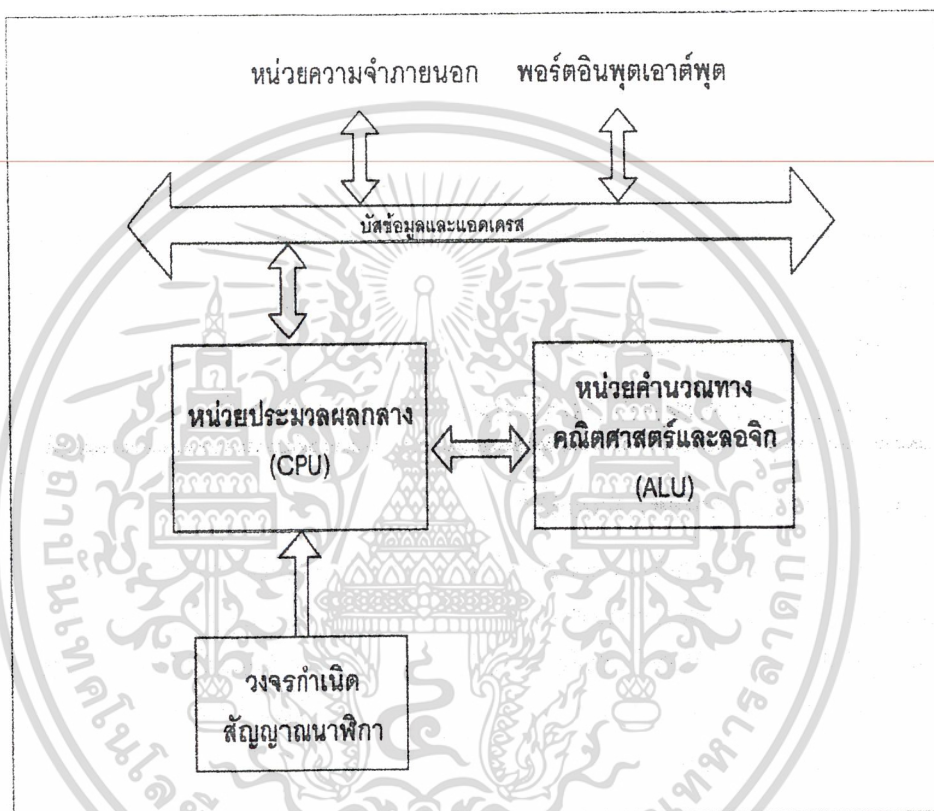
ในรูปที่ 2.1 โครงสร้างของไมโครโปรเซสเซอร์ ซึ่งประกอบไปด้วย หน่วยประมวลผลกลางหน่วยคำนวณทางคณิตศาสตร์และลอจิก บัสข้อมูลและแอดเดรสสำหรับติดต่อกับหน่วยความจำภายนอกและวงจรมกำเนิดสัญญาณนาฬิกา ดังนั้นหมายความว่า การใช้งานไมโครโปรเซสเซอร์จะต้องเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอก และถ้าหากต้องการเชื่อมต่ออุปกรณ์อินพุตและเอาต์พุต ต้องอาศัยอุปกรณ์ที่เรียกว่า ไอซีขยายพอร์ต (Port expander) ทำให้การสร้างระบบควบคุมจึงต้องการอุปกรณ์จำนวนมาก ส่งผลให้ขนาดของระบบใหญ่พอสมควร

ในรูปที่ 2.2 แสดงโครงสร้างของไมโครคอนโทรลเลอร์ จะเห็นได้อย่างชัดเจนว่า ภายในไมโครคอนโทรลเลอร์มีอุปกรณ์พื้นฐานเหมือนกับไมโครโปรเซสเซอร์ หากแต่จะบรรจุหน่วยความจำโปรแกรม หน่วยความจำข้อมูล และพอร์ตอินพุตเอาต์พุตไว้ในพร้อมสรรพ ผู้ใช้งานจึงเพียงแค่เขียน โปรแกรมควบคุมลงบนตัวไมโครคอนโทรลเลอร์ แล้วต่ออุปกรณ์ที่ใช้ในการสร้างวงจรมกำเนิดสัญญาณนาฬิกา อาทิ คริสตัล ตัวเก็บประจุเป็นต้น

สุดท้ายเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตเข้ากับพอร์ต ก็สามารถที่จะนำไปใช้งานได้ และไมโครคอนโทรลเลอร์ก็ยังสามารถเชื่อมต่อกับหน่วยความจำภายนอกได้เช่นกัน โดยพิจารณา

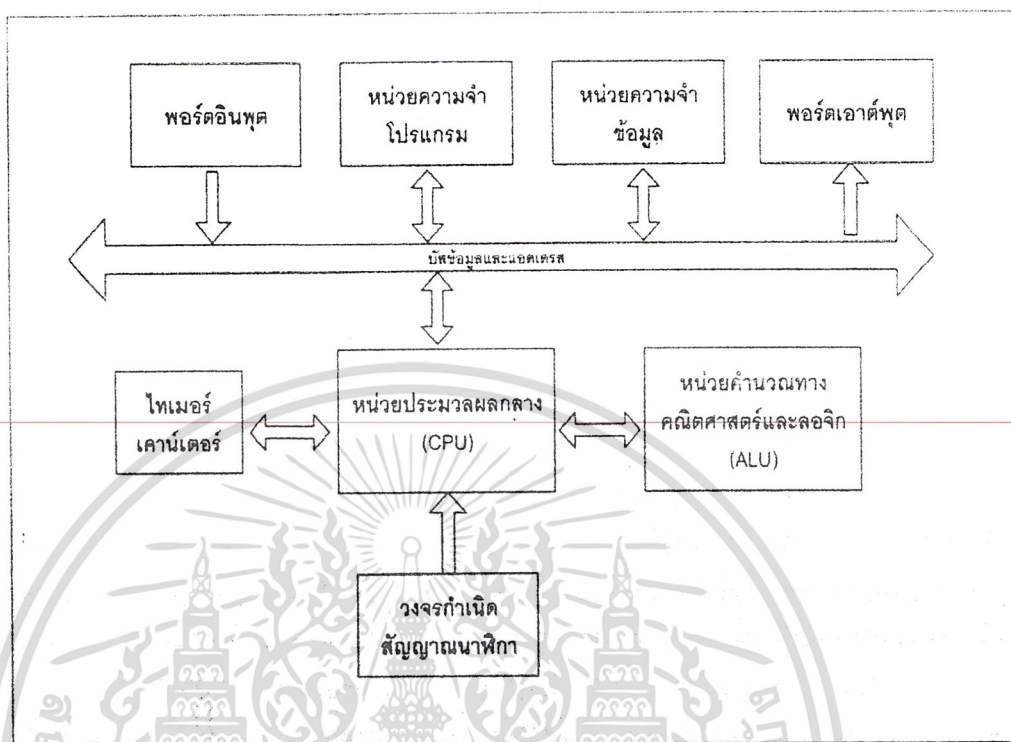
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้หน่วยความจำภายนอกนั้นเป็นอุปกรณ์อินพุตเอาต์พุตตัวหนึ่ง แล้วใช้พอร์ตที่มีอยู่ทำการติดต่อกับหน่วยความจำภายนอกได้ต่อไป



รูปที่ 2.1 อุปกรณ์พื้นฐานภายในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

ตารางที่ 2.1 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์

คุณสมบัติ	ไมโครโปรเซสเซอร์	ไมโครคอนโทรลเลอร์
ขนาดของหน่วยประมวลผลกลาง	ไม่น้อยกว่า 8 บิต	ส่วนใหญ่จะมีขนาด 8 บิต
หน่วยคำนวณทางคณิตศาสตร์และลอจิก	มีอยู่ภายใน	มีอยู่ภายใน
วงจรถักเก็บข้อมูล	มีอยู่ภายใน	มีอยู่ภายใน
การเชื่อมต่อกับหน่วยความจำโปรแกรม	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายในและภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์ (ต่อ)

คุณสมบัติ	ไมโครคอมพิวเตอร์	ไมโครคอนโทรลเลอร์
การเชื่อมต่อกับหน่วยความจำข้อมูล	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายในและภายนอก
การเชื่อมต่อกับพอร์ตอินพุตเอาต์พุต	เชื่อมต่อภายนอกเท่านั้น	มีอยู่ภายในและสามารถขยายได้
ไทเมอร์เคาน์เตอร์	ไม่มีในชิพียุขขนาดเล็ก	มีอย่างน้อย 1 ตัวขนาด 8-บิต
วอตช์ด็อกไทเมอร์	ไม่มีในชิพียุขขนาดเล็ก	มีอย่างน้อย 1 ตัว
จำนวนขาต่อใช้งาน	ไม่น้อยกว่า 40 ขา	มีตั้งแต่ 8 ขาขึ้นไป

2.2.2 โครงสร้างและสถาปัตยกรรม ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่ใช้เรียนรู้จะอ้างอิงถึงไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลช (flash memory) ของ Atmel Corporation มีเบอร์ชิ้นต้นด้วย AT89 เหตุผลที่ใช้ไมโครคอนโทรลเลอร์แบบนี้ในการเรียนรู้เพื่อใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มีด้วยกันหลายประการดังนี้

1. หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ทำให้สามารถลบและเขียนใหม่ได้นับพันครั้งจึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิปเดี่ยว ส่งผลให้สามารถใช้งานพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ

2. ต้นทุนและเวลาในการพัฒนาระบบไมโครคอนโทรลเลอร์ลดลงอย่างมาก เนื่องจากไม่ต้องใช้เครื่องมือพัฒนาจำพวกอิมูเลเตอร์และเครื่องโปรแกรมอีพรอม

3. บริษัทผู้ผลิตได้ทำการผลิตไมโครคอนโทรลเลอร์ตระกูลนี้ออกมาหลายเบอร์ และมีความสามารถแตกต่างกันไป ทำให้มีทางเลือกในการใช้งานสูง

4. ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ ทำให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้เป็นอย่างดี

5. ในบางเบอร์ของไมโครคอนโทรลเลอร์ที่ผลิตโดย Atmel สามารถทำการโปรแกรมข้อมูลโดยที่ไม่ต้องถอดตัวไมโครคอนโทรลเลอร์ออกมาทำการโปรแกรมใหม่ หรือเรียกว่า การโปรแกรมในวงจรถ หรือ ในระบบ (In-system programming) ทำให้การพัฒนาหรือการซ่อมบำรุง ตลอดจนการปรับปรุงหรืออัปเดตข้อมูลในหน่วยความจำโปรแกรมทำได้สะดวก ภายใต้งบประมาณที่ไม่สูงมากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ชุดคำสั่งและสถาปัตยกรรมพื้นฐานเหมือนกับไมโครคอนโทรลเลอร์ MCS-51 ของผู้ผลิตอื่น ไม่ว่าจะอินเทล, ซีเมนส์ หรือดัลลัส

คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบพูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx
- มีวอตช์ด็อกไทมเมอร์ในตัว สำหรับในอนุกรม AT89Sxx

โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx เห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา หากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว

สำหรับโครงสร้างพื้นฐานของอนุกรม AT89Sxx จะเห็นได้ว่า มีส่วนประกอบที่เพิ่มเติมแตกต่างจาก AT89Cxx อยู่หลายส่วน อาทิ วงจรเชื่อมต่ออนุกรมแบบ SPI ซึ่งในไมโครคอนโทรลเลอร์อนุกรม ใช้ในการเขียนข้อมูลในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิปออกไปจากระบบหรือเรียกว่า การโปรแกรมในวงจร ไทเมอร์เคาน์เตอร์ขนาด 16 บิตที่เพิ่มเติมเข้ามาอีกหนึ่งตัวเป็น ไทเมอร์ 2 และวงจรวอตช์ด็อกที่ใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน โดยมีรายละเอียดขั้นต้น ดังนี้

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้ โดยการเขียนข้อมูล " 1 " ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อกับด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้นอกจากนี้ขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0- A7) และขาข้อมูล (D0- D7) โดยใช้ขบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ตารางที่ 2.2 รายละเอียดบางส่วน of ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เบอร์ของไมโครคอนโทรลเลอร์	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล	จำนวนไทมเมอร์เคาน์เตอร์ 16 บิต
AT89C1051	แบบแฟลชขนาด 1 กิโลไบต์	แรม 64 ไบต์	1
AT89C2051	แบบแฟลชขนาด 2 กิโลไบต์	แรม 128 ไบต์	2
AT89C51	แบบแฟลชขนาด 4 กิโลไบต์	แรม 128 ไบต์	2
AT89C52	แบบแฟลชขนาด 8 กิโลไบต์	แรม 256 ไบต์	3
AT89C55	แบบแฟลชขนาด 20 กิโลไบต์	แรม 256 ไบต์	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 รายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช (ต่อ)

เบอร์	หน่วยความจำ	หน่วยความจำ ข้อมูล	จำนวนไทเมอร์ เคาน์เตอร์ 16 บิต
AT89S8252	แบบแฟลชขนาด 8 กิโลไบต์	แรม 256 ไบต์ อีอีพรอม 2 กิโลไบต์	3
AT89S53	แบบแฟลชขนาด 12 กิโลไบต์	แรม 256 ไบต์	3

ขาพอร์ต 1 (P1.0- P1.7) มี 8 ขา แต่ละขาสามารถกำหนดได้ให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล " 1 " ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อดัวยและ P1.1เป็นขาอินพุตทริกเกอร์ไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อโปรแกรมข้อมูลในระบบ

ขาพอร์ต 2 (P2.0 -P2.7) มี 8 ขาแต่ละขาสามารถกำหนดได้ให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล " 1 " ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อดัวยส่งผลให้ขาพอร์ตนั้นมีสถานะปลั๊กลอย(float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนี้ขาพอร์ตนี้อยู่ถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8- A15)

ขาพอร์ต 3 (P3.0 -P3.7) มี 8 ขาแต่ละขาสามารถกำหนดได้ให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล " 1 " ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อดัวยส่งผลให้ขาพอร์ตนั้นมีสถานะปลั๊กลอย(float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนี้ขาพอร์ตนี้อยู่เป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{INT0}$

P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 1 หรือขา $\overline{INT1}$

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณ ไทเมอร์จากภายนอกช่อง 0 หรือขา TO

P3.5 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 1 หรือขา T1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.6 ใช้เป็นขาสัญญาณ \overline{WR} ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ \overline{RD} ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซ็ต (Reset) ใช้ในการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซ็ตสถานะที่ขานี้ต้องอยู่ในระดับรีเซ็ตอย่างน้อย 2 แมกซีนไซเคิล โดยที่วงจรถูกกำหนดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา $\overline{ALE}/PROG$ (Address Latch Enable/ Program pulse input) เป็นขาที่ใช้ควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา \overline{PSEN} (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์ต้องอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมกซีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูล ขานี้จะไม่มีสัญญาณใดๆออกมา

ขา \overline{EA}/V_{pp} (External Access enable / Programming voltage input) ใช้สำหรับเลือกการติดต่อกับหน่วยความจำโปรแกรม จากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.2.4 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาพุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงจรถ่ายและวงจรถับตลอดจนบัฟเฟอร์อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปเผยแพร่โดยไม่ผ่านการคัดค้าน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางงานนอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์โคม และวงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเป็นอย่างไร

วงจรแลตซ์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิปฟล็อปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตซ์ สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ตและสัญญาณอ่านข้อมูลจากวงจรแลตซ์ ส่วนการเขียนข้อมูลไปยังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิปฟล็อป ในขณะที่ข้อมูลจะผ่านมาทางขาบั๊ตข้อมูลภายในเข้าสู่ขา D ของดีฟลิปฟล็อป

ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่า ต้องการใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตหรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์ เนื่องจากขาพอร์ต 0 ไม่มีวงจรพูลอัพภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัพภายนอกที่ขาพอร์ต 0 ทุกขาด้วย

การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล " 1 " มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการจับสัญญาณเอาต์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัพภายในโดยตรงส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกดึงเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดให้ทำงานในสถานะลอจิก "0" จะดีและสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลท

ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.0	AT89C52/AT89Sxx	ขา T2 เป็นขาอินพุตนับค่าของ ไทเมอร์/เคาน์เตอร์และเป็นขา
P1.1	AT89C52/AT89Sxx	ควบคุมทิศทางของสัญญาณ
P1.4	AT89Sxx	ขา SS (Slave Select) เป็นขา เลือกการติดต่อในกรณีที่ไมโคร คอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ใน ระบบการติดต่อแบบ SPI
P1.5	AT89Sxx	ขาMOSI (Master data output, Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT89Sxx	ขา MISO (Master data input, Slave data output) ใช้ในการติดต่อกับ พอร์ต SPI
P1.7	AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับ พอร์ต SPI

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล "0" ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล "0" ไปยังวงจรถ่าย ซึ่งก็จะส่งต่อไปขับเฟด ทำให้เฟดทำงาน ที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก "0" ขึ้น ในทางตรงข้ามหากต้องการส่งข้อมูล "1" ออกไป ก็ให้เขียนข้อมูล "1" ไปยังวงจรถ่าย วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรถ่าย อับภายในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่บัพเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีกรอ่านข้อมูลที่บัพเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานเป็นพอร์ตเอาต์พุต แต่ละขา (หรือแต่ละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (source current) ได้สูงสุด 10 mA และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 mA สำหรับพอร์ต 0 และ 15 mA สำหรับพอร์ต 1-3 ในกรณีที่ใช้งานทุกพอร์ตเอาต์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแส จึงควรต่อวงจรบัพเฟอร์ทางเอาต์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

การอ่านค่าลอจิกจากพอร์ต

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลทที่จะสามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะคือ อ่านจากขาพอร์ตโดยตรง และอ่านจากวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ตต่อจากขามอสทรานซิสเตอร์ชนิด NPN และขาอิมิตเตอร์จากทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล "1" ไปยังทรานซิสเตอร์ จะทำให้ทรานซิสเตอร์ทำงานสถานะลอจิกที่ขาพอร์ตจะเป็น "0" เนื่องจากเมื่อทรานซิสเตอร์ทำงาน จะเสมือนว่าขาพอร์ตนั้นถูกต่อลงกราวด์ ทำให้หากอ่านค่าลอจิกที่ขาพอร์ตจะได้ผลตรงข้ามกับที่ส่งออกมา แต่ถ้าหากทำงานอ่านค่าลอจิกที่วงจรแลตช์ จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริง ดังนั้น ในการอ่านค่าลอจิกจากพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียูและลำดับขั้นตอนการประมวลผลคำสั่ง ในการประมวลผลคำสั่งของซีพียูจะมีขั้นตอนหลักๆ 2 ขั้นตอนคือ กระบวนการเฟตช์ (fetch) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผล ขั้นตอนต่อมาคือ กระบวนการเอ็กซีคิวต์ (execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมา โดยกระบวนการก่อนหน้า เมื่อทำการเอ็กซีคิวต์คำสั่งเรียบร้อยแล้วก็จะไปเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป

เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์จะเกิดการรีเซ็ตในลักษณะที่เรียกว่า เพนเวอร์ ออนรีเซ็ต (power on reset) ซีพียูเริ่มต้นทำงานที่แอดเดรส 0000H ของหน่วยความจำ

โปรแกรม จังหวะการทำงานของซีพียูจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจากกรอบเอกสารนี้เขียนเอกสารที่จะวิเคราะห์การทำงานของไมโครคอนโทรลเลอร์ เมื่อผู้ดูแลเห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานหรือแมนชีนไซเคิล (machine cycle) เป็นไดอะแกรมเวลาแสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 โดยใน 1 รอบการทำงานหรือแมนชีนไซเคิลจะแบ่งย่อยออกเป็น 6 สเตต (state) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตตมีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกา ถ้าสัญญาณนาฬิกามีความถี่ 12 MHz จะมีคาบเวลาเท่ากับ 1 ms คาบเวลาทั้งสองภายในหนึ่งสเตตจะเรียกว่า เฟส 1 (phase1) และเฟส 2 (phase2)

2.2.5 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจิสเตอร์ฟังก์ชันพิเศษที่มีใช้ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำภายในหลักๆ อยู่ 2 ส่วนคือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล ซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไปในแต่ละเบอร์ บทนี้จะกล่าวถึงรายละเอียดของการจัดสรรหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช การเชื่อมต่อกับหน่วยความจำภายนอก และข้อมูลเบื้องต้นของรีจิสเตอร์ฟังก์ชันพิเศษใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

หน่วยความจำโปรแกรม

การจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในเบอร์ต่างๆ ที่นิยมใช้งาน อันประกอบด้วยเบอร์ AT89C51 และ AT89C52 จะเห็นได้ว่า ทั้งสองเบอร์สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียวหรือรวมกับภายนอกหรือเลือกใช้ที่หน่วยความจำภายนอกอย่างเดียวก็ได้ โดยภายใน AT89C51 จะมีหน่วยความจำโปรแกรมภายในที่ 4 กิโลไบต์ ในขณะที่ AT89C52 จะมีขนาด 8 กิโลไบต์

ในกรณีที่ใช้หน่วยความจำภายในและภายนอกรวมกัน หากใช้ AT89C51 ก็จะสามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์ และถ้าใช้เบอร์ AT89C52 จะสามารถติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ 56 กิโลไบต์

หน่วยความจำโปรแกรมเก็บข้อมูลโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ หรือที่เรียกว่า โปรแกรมมอนิเตอร์ (monitor program) หากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิดอีพรอม (EPROM :Erasable Programmable Read-Only Memory) ซึ่งสามารถทำการอ่านได้เพียงอย่างเดียว

หน่วยความจำโปรแกรมที่มีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซ็ตให้เริ่มต้นการทำงาน จะต้องมาเริ่มต้นที่แอดเดรส 0000H นี้เสมอ อย่างไรก็ตาม ในพื้นที่ของหน่วยความจำเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ดูแลเห็นว่าไม่เหมาะสมในการนำมาใช้ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมไม่ว่าจะใช้งานจากภายในหรือภายนอกก็ตาม ต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้ สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H

พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH

พื้นที่สำหรับบริการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H

พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH

พื้นที่สำหรับบริการอินเตอร์รัปต์ของการสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 0023H

พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH

กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชที่มีหน่วยความจำโปรแกรมภายใน แต่ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกด้วย สามารถทำได้โดยต้องกำหนดแอดเดรสของหน่วยความจำโปรแกรมให้ต่อจากแอดเดรสสุดท้ายของหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์ ยกตัวอย่าง ไมโครคอนโทรลเลอร์ AT89C51 มีหน่วยความจำโปรแกรมขนาด 4 กิโลไบต์ มีแอดเดรสอยู่ระหว่าง 0000H-0FFFH เมื่อต่อหน่วยความจำโปรแกรมภายนอกต้องกำหนดให้แอดเดรสอยู่ในช่วง 1000H-FFFFH

การต่อหน่วยความจำภายนอก ขาพอร์ต P0.0-P0.7 ใช้เป็นขาข้อมูล DO-07 และขาแอดเดรสไบต์ต่ำ โดยผ่านวงจรแลตซ์ ซึ่งปกติใช้ไอซีเบอร์ 74HC573 และใช้สัญญาณ ALE และ PS EN ในการเลือกใช้งานขา P0.0-P0.7 ใช้ในการเชื่อมติดกับขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์ จะเหลือขาพอร์ตเพียง 16 บิต คือที่ขาพอร์ต P1.0-P1.7 และ P3.7

หน่วยความจำข้อมูล

มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายใน โดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 สามารถติดต่อกับหน่วยความจำข้อมูลภายนอก ได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOV X ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อกับหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชแสดงให้เห็นได้ว่ามีลักษณะคล้ายกับการติดต่อกับหน่วยความจำโปรแกรมภายนอก แตกต่างกันที่มีสัญญาณที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอกนั้นคือ ขา \overline{RD} และ \overline{WR}

สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 ทุกเบอร์จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM:Random Access Memory) โดยแต่ละเบอร์จะมีขนาดแตกต่างกันไป ในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ ในขณะที่เบอร์เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วนคือ หน่วยความจำข้อมูลส่วนล่าง (lower) , ส่วนบน (upper) และ รีจิสเตอร์ฟังก์ชันพิเศษ (SFR: Special Function Register)

จะเห็นได้ว่า หน่วยความจำข้อมูลส่วนบน และรีจิสเตอร์ฟังก์ชันพิเศษมีตำแหน่งทับซ้อนกันแต่จะใช้การติดต่อที่แตกต่างกันและในไมโครคอนโทรลเลอร์ MCS-51 บางเบอร์จะไม่มีหน่วยความจำข้อมูลส่วนบน

ขนาดของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยแท้จริงแล้วมีเพียง 256 ไบต์ แต่ด้วยการจัดการเข้าถึงที่แตกต่างกัน จึงดูเหมือนว่าไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช มีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อมสำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์เช่นกันมีแอดเดรสอยู่ที่ 80H-FFH สามารถเข้าถึงได้โดยอ้อมเท่านั้น ในขณะที่รีจิสเตอร์ SFR มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกับหน่วยความจำข้อมูลส่วนบน แต่สำหรับรีจิสเตอร์ SFR ในการเข้าถึงแบบโดยตรง ดังนั้นเพื่อความสะดวกและง่ายตลอดจนป้องกันความสับสนในการเขียนโปรแกรมสำหรับผู้เริ่มต้น จึงควรใช้หน่วยความจำข้อมูลภายใน เพียง 128 ไบต์ จากหน่วยความจำข้อมูลส่วนล่างร่วมกับรีจิสเตอร์ SFR

หน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H- 2FH เป็นพื้นที่สำหรับใช้งานทั่วไปสามารถเข้าถึงได้ในระดับบิต (Bit addressable) และหน่วยความจำข้อมูลที่เหลือ 80 ไบต์ จะต้องแบ่งส่วนหนึ่งสำรองไว้เป็นพื้นที่ของสแต็ก (stack : ที่พักข้อมูลชั่วคราวในกรณีที่ใช้ฟังก์ชันการกระโดดไปทำงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำในส่วนนี้ต้องใช้การเข้าถึงในระดับไบต์

รีจิสเตอร์แสดงสถานะของโปรแกรม

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต จึงสามารถกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตัวนี้ได้ อย่างไรก็ตามมีแอดเดรสอยู่ที่ DOH ทำหน้าที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้นจะเรียกสถานะต่างๆของ โปรแกรมว่า แฟล็ก (flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนแปลงสถานะขึ้นผลของการเปลี่ยนแปลงนั้น จะมาปรากฏที่บิตต่างๆ ของรีจิสเตอร์ PSW รายละเอียดของแต่ละบิตในรีจิสเตอร์ PSW

จะเห็นได้ว่า นอกจากรีจิสเตอร์ PSW ถูกใช้ในการเก็บสถานะของโปรแกรมแล้วที่บิต RS0 และ RS1 ยังใช้ในการเลือกแบงก์ของหน่วยความจำส่วนล่าง ซึ่งเป็นพื้นที่ของรีจิสเตอร์ R0-R7 มักนิยมใช้แบงก์ 0 เป็นลำดับแรก หากไม่เพียงพอจึงเลือกในแบงก์อื่นๆมาใช้ แต่ต้องระมัดระวังในการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดค่าและระดับการติดต่อให้สัมพันธ์นั้นอาจทำให้การเขียน โปรแกรมเกิดความสับสน ดังนั้น สำหรับผู้เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ MCS-51 จึงควรเลือกใช้รีจิสเตอร์ R0-R7 ในแบงก์ 0 เพียงแบงก์เดียวให้ชำนาญเสียก่อน

การกำหนดค่ารีจิสเตอร์ PSW เพื่อเลือกใช้งานรีจิสเตอร์ R0-R7 ควรกำหนดไว้ที่ตอนต้นของโปรแกรมเสมอ เพื่อจะได้เขียนโปรแกรมติดต่อกับรีจิสเตอร์ R0-R7 ได้อย่างสะดวกและไม่เกิดความผิดพลาด

2.2.6 แอ็กคิวมูลเตเตอร์

มีขนาด 8บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งในการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะส่งข้อมูลหรือผลลัพธ์ที่ได้ให้แก่ซีพียู เพื่อทำการประมวลผลต่อไป อาจเรียกสั้นๆว่า รีจิสเตอร์ A หรือ ACC รีจิสเตอร์นี้สามารถเข้าถึงระดับบิตได้

รีจิสเตอร์B
มีขนาด 8บิต มีแอดเดรสอยู่ที่ F0H มีหน้าที่พิเศษคือ หากต้องการคูณหรือหารทางคณิตศาสตร์ต้องนำข้อมูลที่ต้องการหาร หรือ คูณมาเก็บไว้ในรีจิสเตอร์ B แล้วจึงกระทำคำสั่งการคูณหรือหาร กับค่าในรีจิสเตอร์ A ต่อไป

ในกรณีที่ไม่ได้มีความต้องการคูณ หรือ หารข้อมูลสามารถใช้รีจิสเตอร์ B นี้ ในการเก็บข้อมูลทั่วไปได้เหมือนกับรีจิสเตอร์ปกติ และสามารถเข้าถึงในระดับบิตได้เช่นเดียวกับรีจิสเตอร์ A

โปรแกรมเคาน์เตอร์

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียูจะต้องไปทำงานรีจิสเตอร์ PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่ได้จัดสรรไว้ร่วมกับรีจิสเตอร์ SFR ตัวอื่นๆ การเปลี่ยนแปลงค่าของรีจิสเตอร์ PC จะขึ้นอยู่กับผลของการกระทำคำสั่งภายในหน่วยความจำ โปรแกรมที่ผู้เขียน โปรแกรมกำหนด

รีจิสเตอร์ PC มีความสำคัญมากโดยเฉพาะอย่างยิ่งในการตรวจสอบการทำงานของโปรแกรมว่า ดำเนินไปตามลำดับขั้นตอนที่กำหนดไว้หรือไม่

CY : แพลททค (Carry flag) เป็น"1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์ และลอจิก แล้วค่าของ แอ็กคิวมูลเตเตอร์ เกิน 255 (ฐานสิบ) หรือ FFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC : แพลกทดเสริม(Auxiliary Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์ แล้วทำให้เกิดการทดข้ามจากบิต 3 มายังบิต 4

FO : แพลกใช้งานทั่วไป เมื่อผู้เขียน โปรแกรมกำหนดค่าที่บิตนี้แล้วไม่ว่าจะ กระทำคำสั่งใดๆ ที่บิตนี้จะ ไม่มีการเปลี่ยนแปลง

RS1 : บิตเลือกรีจิสเตอร์แเบงก์ (Rigister Select 1) ใช้ร่วมงานกับบิต RS0 เพื่อเลือกแเบงก์ของรีจิสเตอร์ R0-R7

RS0 : บิตเลือกรีจิสเตอร์แเบงก์ (Rigister Select 0) ใช้งานร่วมกับบิต RS1 เพื่อเลือกแเบงก์ของรีจิสเตอร์ R0-R7

OV : บิตเกิน (Overflow) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลจิกแล้ว ทำให้เกิดการทดข้ามจากบิต 6 มายังบิต 7 ของแเอกคิวมูลเตอร์

- : บิตนี้ผู้ใช้งานสามารถกำหนดใช้งาน ได้อย่างอิสระ

P : บิตพาริตี (Parity) ใช้ในการตรวจสอบจำนวนค่า "1" ภายในแเอกคิวมูลเตอร์ ถ้าหากในแเอกคิวมูลเตอร์มีจำนวนบิตที่เป็น "1" รวมกันเป็นเลขคู่ บิตนี้จะเป็น "0" ถ้ารวมกันเป็นเลขคี่บิตนี้จะเป็น "1" หรือรีจิสเตอร์ตัวชี้สแตก มีขนาด 8 บิต มีแเอคเครสอยู่ที่ 81 H ใช้ในการเก็บค่าตำแหน่งของตัวชี้สแตก ซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระ โดคไปทำงานที่โปรแกรมย่อย หรือกระ โดคโปรแกรมย่อยไปยัง โปรแกรมหลัก เมื่อมีการรีเซตเกิดขึ้น จะแสดงในตารางที่ 2.4

ตารางที่ 2.4 สแตกพอยน์เตอร์ (Stack Pointer : SP)

SI	RS0	แเบงก์ของรีจิสเตอร์	ช่วงแเอคเครส
0	0	แเบงก์ 0	00H-07H
1	1	แเบงก์ 1	08H-0FH
0	0	แเบงก์ 2	10H-17H
1	1	แเบงก์ 3	18H-1FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 CP-AT32 Plus

ด้วยความโดดเด่นในหลายๆ ด้านของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทำให้มีผู้นำไปใช้งานอย่างแพร่หลาย ซึ่งมีผลทำให้ไมโครคอนโทรลเลอร์ในตระกูลนี้ได้รับความนิยมอย่างต่อเนื่องตลอดมา จึงทำให้มีผู้ผลิตหลายบริษัทที่ผลิตไมโครคอนโทรลเลอร์ในตระกูลนี้ออกมาสู่ตลาดเพื่อตอบสนองความต้องการของผู้ใช้ โดยผู้ผลิตในแต่ละรายจะเพิ่มความสามารถด้านต่างๆ ลงไปในไมโครคอนโทรลเลอร์ตระกูลนี้ อาทิเช่น การบรรจุหน่วยความจำแบบ FLASH ลงในตำแหน่ง ROM, การเพิ่มหน่วยความจำแบบ EEPROM เพื่อใช้ในการเก็บข้อมูล, เพิ่มความเร็วของการทำงานใน 1 รอบคำสั่ง และอื่นๆ อีกมากมาย

ด้วยเหตุนี้เองที่เป็นแรงผลักดันให้ไมโครคอนโทรลเลอร์ตระกูลนี้ถูกนำไปใช้งานกว้างขวางมากขึ้น ดังนั้นทาง ETT จึงได้พัฒนาบอร์ด CP-AT32 PLUS ขึ้นมา โดยในส่วนของบอร์ด CP-AT32 PLUS ถูกออกแบบมาเพื่อให้ใช้งานกับ CPU P89C51RD2 เพื่อให้สามารถเขียนโปรแกรมได้มากถึง 64 Kbyte และสามารถทำงานได้เร็วกว่า CPU ในตระกูลเดียวกันถึง 2 เท่าเมื่อใช้ Xtal ค่าเดียวกัน ซึ่งบอร์ดรุ่นนี้ออกแบบมาให้ใช้ค่า Xtal 18.432 MHz โดยบอร์ดรุ่นนี้ยังมีคุณสมบัติเด่นๆ อีกมากมาย ซึ่งรายละเอียดจะอยู่ในเนื้อหา

การพัฒนาโปรแกรมก็มีความสะดวกมากกว่าบอร์ดรุ่นเก่าๆ โดยสามารถ Download โปรแกรมจาก PC ลงในบอร์ด โดยผ่านทาง RS-232 โดยไม่ต้องมีอุปกรณ์อื่นเสริม ซึ่งทำให้ประหยัดค่าใช้จ่าย ดังนั้นทาง ETT จึงได้พัฒนาบอร์ด CP-AT 32 PLUS เพื่อตอบสนองกับผู้ใช้และผู้ผลิต CPU

2.3.1 SPECIFICATION

ตารางที่ 2.5 SPECIFICATION ของ CP-AT32 PLUS

CPU	P89C51RD2 @ 18.432 MHz x 2 Clock
XTAL	18.432 MHz x 2 Clock
ROM	64 Kbyte Internal
RAM	32 Kbyte External
8255 I/O PORT	9 PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 SPECIFICATION ของ CP-AT32 PLUS (ต่อ)

LCD	CHARACTER / GRAPHIC
RS232	YES (MAX232)
RS422	YES (MAX3086)
RS485(FULL)	YES (MAX3086)
RS485(HALF)	YES (MAX3086)
ADC	0-5 V 12 BIT 2 CHANNEL (LTC 1298)
DAC	10 V 8 BIT 4 CHANNEL (MAX500)
EEPROM	24xx
OPEN COLLECTOR	400 Ma 8 BIT (NC6B595)
RAM BACK UP	YES (DS1210)
I2 C BUS	YES
SPI BUS	YES
1 - WIRS	YES
POWER SUPPLY	YES
ISP DOWNLOAD	YES

2.3.2 รายละเอียดเบื้องต้น

บอร์ด CP-AT32 PLUS เป็นบอร์ดที่มีพอร์ตและ อุปกรณ์ใช้งานอื่น ๆ จำนวนมาก ซึ่งสามารถตอบสนองต่อความต้องการใช้งานที่งานมีขนาดใหญ่ได้ โดยบอร์ด CP-AT32 PLUS ใช้ CPU P89C51RD2 ซึ่งสามารถทำงานได้เร็วกว่า CPU ในตระกูลเดียวกันได้เป็น 2 เท่า เมื่อใช้ความถี่ของ XTAL ค่าเดียวกัน โดยสามารถรองรับการเขียนโปรแกรมที่มีขนาดใหญ่ได้ เนื่องจาก CP-AT32 PLUS มี ROM ขนาด 64 Kbyte และ RAM ขนาด 32 Kbyte โดยสามารถเก็บข้อมูลใน RAM ภายนอกได้เมื่อไม่มีไฟจ่ายให้กับบอร์ด โดยเพียงใส่ IC DS1210 และ BAT 3 V ลงบนบอร์ดเท่านั้น

จากที่กล่าวมาแล้วบอร์ด CP-AT32 PLUS เหมาะกับการนำไปใช้ควบคุมงานขนาดใหญ่ เนื่องจากตัวบอร์ดมีพอร์ตอินพุตและเอาต์พุตมากถึง 9 พอร์ต (8255 3 ตัว) อีกทั้งยังมี PORT ที่เป็น OPEN COLLECTOR คือ 8 BIT เพื่อใช้ควบคุมโหลดได้โดยตรง เช่น RELAY ในงานที่มีการเชื่อมต่อสัญญาณอนาลอกนั้น บอร์ด CP-AT32PLUS มี ADC ขนาด 12 บิต จำนวน 2 Channel ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถกำหนดโหมดการทำงานเป็นแบบ Single-End หรือ Different ได้ ในส่วนของการแปลง จากสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก หรือ DAC นั้น บอร์ด CP-AT32PLUS มี DAC ขนาด 8 บิต จำนวน 4 ช่องที่สามารถปรับแรงดันเอาต์พุตที่ Full-Scale ได้ถึง 2 – 10V

การสื่อสารข้อมูลแบบอนุกรมสามารถสื่อสารได้ 2 ระบบคือ RS232 และ RS422/RS485 โดยที่ RS485 สามารถใช้งานได้ทั้ง FULL DUPLEX และ HALF DUPLEX

ในส่วนของ การแสดงผลของบอร์ด CP-AT32PLUS นั้น สามารถแสดงผลได้ 2 แบบคือ การแสดงผลด้วย LCD ซึ่งสามารถแสดงผลได้ทั้ง GRAPHIC & CHARACTER และการแสดงผลด้วย 7-SEGMENT ซึ่งสามารถแสดงผลได้ถึง 8 DIGIT โดยใช้บอร์ด ET-SDP8 มาประกอบรวม

บอร์ด CP-AT32 PLUS ยังสนับสนุนการเชื่อมต่ออุปกรณ์ภายนอกอีกหลายชนิด อาทิเช่น KEY 4x4 , 12 C BUS , SPI BUS, 1-WIRE และอื่นๆ

2.3.3 FEATURES OF BORAD

- P89C51RD2 @ 18.432MHz With x2 Clock Speed
- 64 Kbyte Rom (Flash) In CPU
- 32 Kbyte Ram
- 9 I/O Port With 8255
- Compatible MCS-51 20 PIN Port
- 12C Expansion Port
- SPI Expansion Port
- 1-Wire Expansion Port
- ET-SDP8 (MAX7219) Port
- Graphics & Character LCD Port
- Key 4 x 4 Port
- 1 Mini Speaker
- 8 Output Open Collector
- ADC 12 BIT 2 Channel With Sing-End or Different Mode
- DAC 8 บิต 4 Channel Full Scale 1 –10V
- Serial Eeprom With 24xx Family
- RTC DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RAM BACK UP With DS1210
- Port RS232
- Port RS422/RS485
- In System Programming Download
- On Board Power Supply

2.3.4 FEARTURE OF CPU

- 8051 Central Processing Unit
- On-Chip Flash Program Memory with In-System Programming (ISP) and In-Application Programming (IAP) capability
- Boot ROM contains low level Flash programming routines for downloading via UART
- Can be programmed by the end-user application(IAP)
- 6 clock per machine cycle operation(standard)
- 12 clock per machine cycle operation(standard)
- Speed up to 20 MHz with 6 clock cycles per machine cycle (40 MHz equivalent performance); up to 33 MHz with 12 clock per machine cycle
- Fully static operation
- Ram expandable externally to 64 kB
- 4 level priority interrupt
- 7 interrupt source
- Four 8 bit i/o Port
- Full-duplex enhanced UART
 - Framing error detection
 - Automatic address recognition
- Power Control Mode
 - Clock and be stopped and resumed
 - Idle mode
 - Power down mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Programable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Programable Counter Array
- Programmable Counter Array(PCA)
- PWM
- Capture / compare

2.3.5 MEMORY MAP

บอร์ด CP-AT32 PLUS ออกแบบให้สามารถใช้หน่วยความจำแบบ ROM สำหรับเขียนโปรแกรมขนาด 64 Kbyte และหน่วยความจำสำหรับเก็บแบบ RAM(62256) สำหรับเก็บข้อมูลอีก 32 Kbyte

โดยหน่วยความจำแบบ ROM ขนาด 64 Kbyte จะเป็นหน่วยความจำแบบ Flash ที่อยู่ในตัว CPU

ภายใน CPU จะมีหน่วยความจำแบบ RAM เหมือนกับ CPU ตัวอื่น ๆ ในตระกูล MCS-51 คือ ตำแหน่งที่ 00H – 7FH เป็นหน่วยความจำขนาด 128 ไบต์ ที่สามารถเข้าถึงข้อมูลได้โดยตรงและตำแหน่งที่ 80H – FFH เป็นหน่วยความจำแบบ RAM อีก 128 ไบต์ที่สามารถเข้าถึงข้อมูลโดยทางอ้อม

แต่ที่แตกต่างจาก CPU ตัวอื่นก็คือ ตั้งแต่ตำแหน่ง 0100H – 03FFH จะมีหน่วยความจำอยู่ภายในแต่สามารถเข้าถึงข้อมูลได้โดยคำสั่ง MOVX เหมือนกับการเข้าถึงหน่วยความจำภายนอกแต่จะต้องกำหนดค่าในรีจิสเตอร์ AUXR เพื่อเลือกที่จะเข้าถึงข้อมูลภายในหรือจะเข้าถึงข้อมูลภายนอก (62256) ซึ่งสามารถสรุปเป็นตารางดังนี้

ตารางที่ 2.6 กำหนดค่า REGISTER AUXR

MEMORY	LOCATION	ADDRESS	CAPABILITY
ROM(Flash)	Internal	0000H-FFFFH	64 Kbyte
RAM	Lower Internal Ram	00H-7FH	128 Byte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 กำหนดค่า REGISTER AUXR (ต่อ)

RAM	Upper Internal Ram	80H-FFH	128 Byte
RAM	Internal Eram	0100H-03FFH	768 Byte

I/O MAP

I/O	ADDRESS
8255 #1	8000H – 8003H
8255 #2	9000H – 9003H
8255 #3	A000H – A003H

2.3.6 ส่วนต่างๆของ CP-AT32Plus

บอร์ด CP-AT32 PLUS ออกแบบให้มีพอร์ต 8255 จำนวน 3 ตัว โดย 8255 แต่ละตัวจะมี ADDRESS 4 ตำแหน่ง ซึ่งแสดงดังตารางที่ 2.7

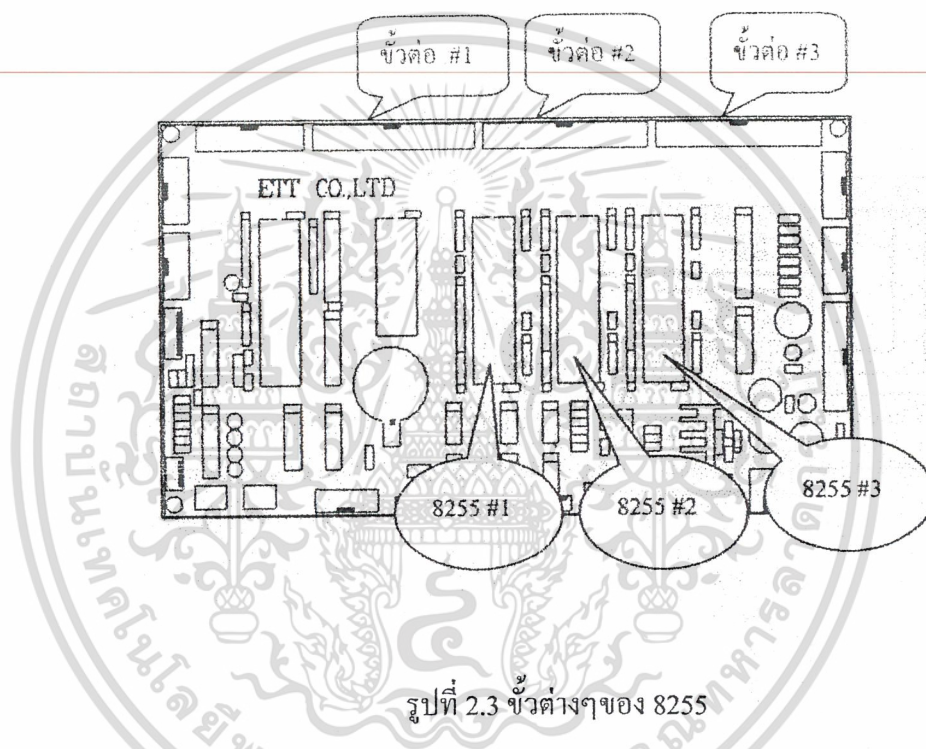
ตารางที่ 2.7 ADDRESS ของพอร์ต 8255

CHIP	PORT	ADDRESS
8255 #1	PORTA	8000H
	PORTB	8001H
	PORTC	8002H
	CONTROL PORT	8003H
8255 #2	PORTA	9000H
	PORTB	9001H
	PORTC	9002H
	CONTROL PORT	9003H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

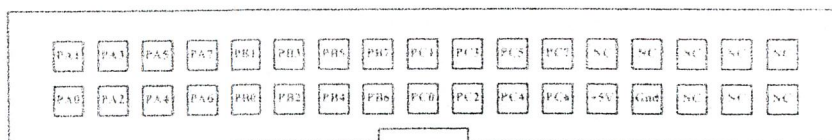
ตารางที่ 2.7 ADDRESS ของพอร์ต 8255 (ต่อ)

8255 #3	PORTA	A000H
	PORTB	A001H
	PORTC	A002H
	CONTROL PORT	A003H



รูปที่ 2.3 ขั้วต่างๆของ 8255

โดย PORT A , PORT B , PORT C ของ 8255 แต่ละตัวจะเชื่อมต่อกับคอนเน็กเตอร์ขนาด 34 PIN (72IOZ80) ซึ่งมีตำแหน่งขาตั้งรูปด้านล่าง



รูปที่ 2.4 ตำแหน่งคอนเน็กเตอร์ 34 PIN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

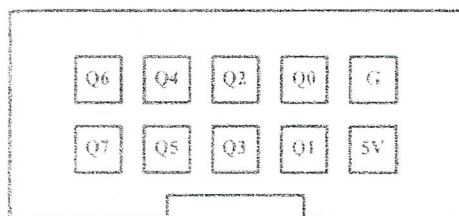
โดย 8255 จะต้องถูกควบคุมโดย CONTROL WORD ดังนี้

NC6B595 จะถูกควบคุมโดย CONTROL WORD ผ่านทาง 8255 ตัวที่ 3 ซึ่งอยู่ในตำแหน่ง A000H – A003H โดยสามารถเปรียบเทียบขาของ NC6B595 และ 74HC595 ได้ในตารางที่ 2.8 ซึ่งในตารางจะมีขาของ 8255 ที่ใช้ควบคุมอยู่ด้วย

ตารางที่ 2.8 ขาควบคุมของ NC6B595,74HC595,8255ตัวที่3

ขาควบคุมของ NC6B595	ขาควบคุมของ 74HC595	ขาของ 8255 ตัวที่ 3 ที่ ใช้ควบคุม
Sin	Si	PA4
STR	RCK	PA6
CLK	SCK	PA5
R	SCLK	5V
OE	G	GND

จากที่ได้กล่าวมาว่าเอาต์พุตของ NC6B595 เป็นแบบ OPEN COLLECTOR จึงสามารถนำไปขับโหลดได้โดยตรง แต่ถ้านำไปขับโหลดประเภท INDUCTOR เช่น RELAY จะต้องมิ DIODE ต่อกับโหลดอยู่ด้วย โดยในบอร์ด CP-AT32 PLUS ได้ออกแบบให้สามารถใส่ DIODE ได้เลย โดยถ้าใช้ DIODE ที่ออกแบบมากับ CP-AT32 PLUS จะต้องเป็นโหลดที่ใช้ไฟ 5V เท่านั้น หากเป็นโหลดที่ใช้แรงดันไปที่สูงกว่านี้จะต้องถอด DIODE บนบอร์ด CP-AT32 PLUS ออกแล้วจึงใช้ DIODE กัน โหลดภายนอกแทน โดยสัญญาณ Q0 – Q7 จะต่อเข้ากับขั้วต่อแบบ IDC ขนาด 10 PIN ดังนี้



รูปที่ 2.5 ขั้วต่อแบบ IDC ขนาด 10 PIN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POWER SUPPLY

บอร์ด CP-AT32 PLUS มีชุด POWER SUPPLY อยู่บนบอร์ดอยู่แล้ว โดยเพียงแต่ป้อนไฟเข้าที่ขั้วอินพุตก็สามารถใช้งานได้ โดยบนบอร์ดออกแบบให้มีขั้วป้อนไฟ 2 ขั้ว โดยขั้วแรกเป็น JACK DC ที่เหมาะกับ ADAPTER ส่วนอีกขั้วหนึ่งเป็นลักษณะของ TERMINAL ขั้วนี้คือ 2 PIN ที่เหมาะกับการต่อกับสายไฟ ซึ่งทั้ง 2 ขั้วนี้จะขนานกันอยู่ สามารถที่จะเลือกใช้ขั้วใดขั้วหนึ่งก็ได้ตามความเหมาะสม

โดยชุด POWER SUPPLY จะมีชุด Regulator เพื่อปรับระดับแรงดันไฟให้อยู่ที่ 5 VDC คงที่ เพื่อใช้งานบนบอร์ด โดยที่ส่วนของ Rectifier จะใช้วงจรบริดจ์ ซึ่งทำให้บอร์ด CP-AT32 PLUS สามารถรับไฟได้ทั้งไฟกระแสตรง และไฟกระแสสลับ ซึ่งถ้าเป็นการใช้ไฟกระแสตรงก็ไม่ต้องระวังเรื่องขั้วต่อ เนื่องจากส่วนของ Rectifier เป็นวงจรบริดจ์

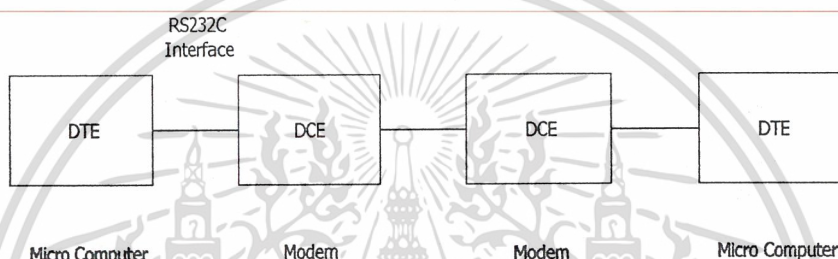
DOWNLOAD PROGRAM

การ DOWNLOAD โปรแกรมสามารถทำได้โดยการต่อสาย RS232 ของบอร์ด CP-AT32 PLUS เข้ากับ RS232 ของ PC โดยบนบอร์ด CP-AT32 PLUS จะต้องมี LINE DRIVER ของ RS232 (MAX232) และจะต้องถอด LINE DRIVER ของ RS422/485 (MAX3086) ออก และจะต้องจ่ายไฟเข้าบอร์ดด้วย

หลังจากนั้นให้กดปุ่ม RESET และปุ่ม PSEN บนบอร์ด CP-AT32 PLUS พร้อมกัน โดยให้ปล่อยปุ่ม RESET ก่อน แล้วปล่อยปุ่ม PSEN ตามมา ซึ่งจะทำให้ CPU เข้าสู่โหมดโปรแกรมหลังจากนั้นให้ RUN โปรแกรม WINISP ให้กำหนดค่า PARAMETER ต่าง ๆ เช่น เลือก CHIP เป็น P89C51RD2 , Xtal เท่ากับ 18.432 MHz (รายละเอียดดูในหัวข้อ การใช้โปรแกรม DOWNLOAD) หลังจากทีโปรแกรม DOWNLOAD เรียบร้อยแล้วให้ RUN โปรแกรมที่ถูก DOWNLOAD ลงไปโดยกดปุ่ม RESET เพียงปุ่มเดียว โดยก่อนที่จะกดปุ่ม RESET เพื่อให้ CPU ทำงาน จะต้องเขียนให้ Status Byte เป็น 00H ก่อน

2.4 มาตรฐาน RS-232

เป็นมาตรฐานที่ใช้กันกว้างขวางมากที่สุดถูกประกาศใช้ในปี 1960 โดย Electronic Industries Association (EIA) เพื่อกำหนดการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Communication Equipment : DCE) โดยที่ RS ย่อมาจาก Recommended Standard และ 232 เป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ C เป็นหมายเลขบับท้ายสุดของมาตรฐานตัวนี้สำหรับผู้ที่ไม่โครคอมพิวเตอร์ DTE ก็หมายถึงตัวไมโครคอมพิวเตอร์และ DCE ก็หมายถึง โมเด็ม



รูปที่ 2.6 การใช้ RS-232-C เชื่อมต่อกับอุปกรณ์

สัญญาณที่ใช้ทั้งหมดใน RS-232-C

Protective Ground (PG ขาที่ 1) หมายถึงตัวถังของเครื่องหรือสายดิน Transmit Data (TD ขาที่ 2) เป็นสัญญาณส่งออกจาก DTE (ตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือไมโครคอมพิวเตอร์เครื่องอื่นหรือเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออกสถานะของลอจิกจะมีค่าเท่ากับ “1” สถานะ “OFF” หรือเทียบเท่ากับ stop bit ไม่ว่าจะ เป็นระบบอะไร DTE ต้องไม่ส่งข้อมูลออกไปจนกว่าสัญญาณเหล่านี้

1. Request To Send : RTS

2. Clear To Send : CTS

3. Data Set Ready : DSR

4. Data Terminal Ready : DTR อยู่ในสถานะ “ON” Receive Data (RD ขาที่ 3) เป็นทางของสัญญาณเข้าไปยัง DTE เมื่อไม่มีสัญญาณรับเข้ามา ขานี้จะมีสถานะทางลอจิกเป็น “1” หรือสถานะ “OFF” Request To Send (RTS ขาที่ 4) จาก DTE ไปยัง DCE สถานะ “ON” คือ บังคับให้ DCE อยู่ใน Transmitting Mode ต่อไปสถานะ “OFF” คือ บังคับให้ DCE อยู่ใน Receiving Mode ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนจาก “OFF” ไป “ON” เป็นการบอกให้ DCE จัดการกับระบบสื่อสารเพื่อให้ช่องสัญญาณต่อเชื่อมและให้สัญญาณ Clear To Send : CTS กลับมาเป็นการบอกว่าส่งได้

การเปลี่ยนจาก “ON” ไป “OFF” เป็นการบอกให้ DCE ส่งข้อมูลผ่านช่องสื่อสารให้หมดแล้วกลับไปอยู่ใน Receiving Mode พร้อมกับให้ CTS เป็น “0” Clear To Send (CTS ขาที่ 5) จาก DTE ไป DTE สถานะ “ON” หมายความว่าข้อมูลจาก DTE ขาที่ 2 จะถูกส่งต่อไปใน ช่องทางสื่อสาร (โมเด็มส่งข้อมูลออกสายโทรศัพท์) ทันที CTS จะ “ON” หลังจาก DRS และ RTS อยู่ในสถานะ “ON” และการเชื่อมวงจรสื่อสาร (ชุมสายโทรศัพท์) เสร็จเรียบร้อยแล้ว Data Set Ready (DTS ขาที่ 6) จาก DCE ไป DTE คือ ความพร้อมของ โมเด็มนั่นเอง จะเป็น “ON” ต่อเมื่อ

1. DCE (โมเด็ม) เปิดเครื่องอยู่ และอยู่ในสถานะ off-hook (เหมือนยกหูโทรศัพท์)
2. DCE ไม่อยู่ใน test mode
3. DCE ทำการส่งสัญญาณไปยังชุมสายเสร็จแล้ว DRS อยู่ในสถานะ “ON” เป็นการบอก DTE ว่าโมเด็มที่ต่อเข้าสายโทรศัพท์เรียบร้อยแล้ว และพร้อมที่จะส่ง DSR อยู่ในสถานะ “OFF” หมายถึงให้ DTE ตรวจ Ring Indicator คือสายร่วม ของสัญญาณทุกตัว Carrier Detect (CD ขาที่ 8)

จาก DCE ไป DTE

สถานะ “ON” หมายถึงว่า DCE จับ สัญญาณพาหะในช่องทางสื่อสารที่จะทำการดีโมดูเลตได้ สถานะ “OFF” คือไม่ได้รับสัญญาณอะไรเลย หรือได้ รับสัญญาณแต่ไม่สามารถโมดูเลตเอาข้อมูลออกมาได้ Data Terminal Ready (DTR ขาที่ 20)

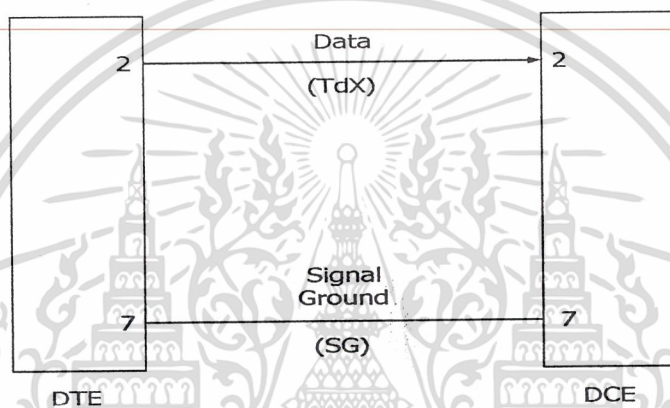
จาก DTE ไป DCE

สถานะ “ON” หมายถึงว่า DCE เตรียมเพื่อเชื่อมต่อกับตัวอื่น และรักษาช่องทางติดต่อไว้ต่อไป (การเชื่อม Channel ทำได้หลายทางคือ หมุนเรียกด้วยมือหรืออัตโนมัติ) ถ้า DCE สามารถตอบรับสัญญาณเรียก (Call) ได้ ก็ให้ตอบรับ (answering) เมื่อมีสัญญาณเรียก Ring Indicator และ DTR On อยู่

สถานะ “OFF” คือวางหู และเมื่อ “OFF” แล้วไม่ต้อง “ON” อีกจนกว่า DSR จะ “OFF” Ring Indicator (RI ขาที่ 22) จาก DCE ไป DTE เหมือนสัญญาณเรียกของโทรศัพท์ แต่เป็นดิจิทัล ใช้ในระบบตอบรับอัตโนมัติ (Auto-answer) สัญญาณนี้จะ “ON” เมื่อมีสัญญาณกระดิ่งเข้ามา และจะ “OFF” ระหว่างเสียงดัง

2.4.1 การสื่อสารทางเดียว

วงจรหลักที่ถูกใช้สำหรับการสื่อสารมีอยู่สามวงจร คือสายเส้นที่ 2 สำหรับข้อมูลจาก DTE ไปยัง DCE สายที่ 3 สำหรับข้อมูลจาก DCE ไปยัง DTE และสายเส้นที่ 7 สำหรับซิกแนลกราวด์ (Signal Ground) ซึ่งเป็นจุดอ้างอิงสำหรับขั้วและแรงดันไฟฟ้าของสายอื่น ในกรณีที่ยากที่สุดซึ่งมีเพียงอุปกรณ์ตัวหนึ่งส่งและอีกตัวรับ ใช้สายเพียงสองเส้นก็เพียงพอคือ สายที่ 2 และ 3 และสายเส้นที่ 7 ดังรูปที่ 2.7



รูปที่ 2.7 การเชื่อมต่อทางเดียวอย่างง่าย

ฮาร์ดแวร์แฮนด์เช็กกิ้ง (Hardware hand shaking)

ในหลาย ๆ กรณี อุปกรณ์ฝ่ายส่งจำเป็นต้องรู้ว่าฝ่ายรับพร้อมที่จะรับข้อมูลหรือไม่ ตัวอย่างเช่น การส่งข้อมูลไปเครื่องพิมพ์ ความเร็วของการสื่อสารอาจเร็วกว่าความเร็วของเครื่องพิมพ์ เครื่องพิมพ์ต้องระงับการส่งข้อมูลของคอมพิวเตอร์จนกว่ามันพร้อมที่จะรับข้อมูล ในทำนองเดียวกันกับการส่งข้อมูลจากคอมพิวเตอร์เครื่องหนึ่งไปอีกเครื่องหนึ่ง และคอมพิวเตอร์ตัวที่ 2 ไม่สามารถประมวลผลข้อมูลได้เร็วเท่ากับอัตราที่ข้อมูลเข้า ทั้งสองกรณี ต้องมีข่าวสารส่งจากอุปกรณ์ฝ่ายรับไปยังอุปกรณ์ฝ่ายส่งเพื่อแจ้งว่าพร้อมหรือไม่ ข่าวสารนี้เรียกว่า โฟลว์คอนโทรล (flow control) หรือ แฮนด์เช็กกิ้ง (Hand shaking) แฮนด์เช็กกิ้งมีสองแบบ คือ ฮาร์ดแวร์และซอฟต์แวร์ ทั้งคู่เกี่ยวข้องกับสัญญาณที่ส่งมาจากอุปกรณ์ฝ่ายรับไปยังอุปกรณ์ฝ่ายส่ง ในฮาร์ดแวร์แฮนด์เช็กกิ้ง อุปกรณ์ฝ่ายรับจะส่งแรงดันไฟฟ้าบวกไปตามวงจรแฮนด์เช็กกิ้ง เมื่อมันพร้อมที่จะรับข้อมูล เมื่อคอมพิวเตอร์ฝ่ายส่งได้รับแรงดันไฟฟ้าลบ มันจะรู้ว่าต้องหยุดส่งข้อมูลในซอฟต์แวร์แฮนด์เช็กกิ้ง สัญญาณแฮนด์เช็กกิ้งเป็นอักขระพิเศษที่ถูกส่งไปตามวงจรข้อมูลแทนวงจรแฮนด์เช็กกิ้ง การใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮาร์ดแวร์เซ็คกิ้งอย่างน้อยต้องมีการเชื่อมต่อเพิ่มเติมหนึ่งเส้นเพื่อนำพาสัญญาณ ทำให้จำนวนของสายทั้งหมดเป็นสามเส้นคือ สายข้อมูล ซิกแนลกราวด์ และแฮนเซ็คกิ้ง

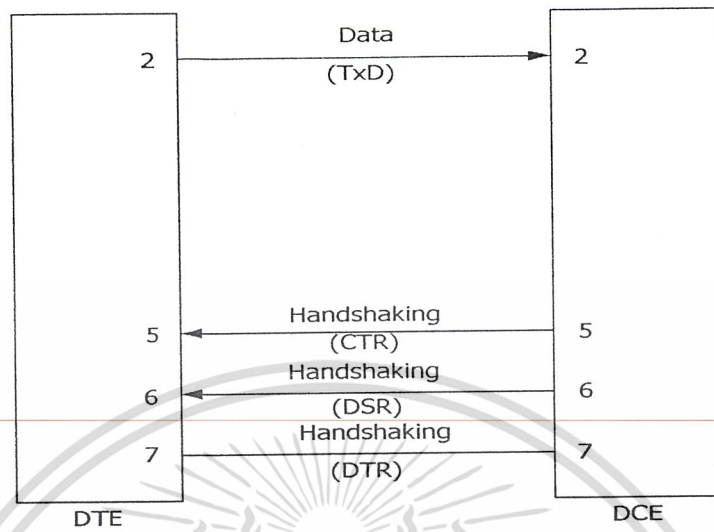
จาก DTE ไปสู่ DCE

เมื่ออุปกรณ์ DTE ส่งข้อมูลไปที่อุปกรณ์ DCE ข้อมูลจะถูกส่งไปตามสายที่ 2 และใช้สายที่ 7 ซิกแนลกราวด์ ตามที่ปกติอุปกรณ์ DCE ควบคุมการส่งอุปกรณ์แฮนเซ็คกิ้งจากอุปกรณ์ DTE บนสายเส้นที่ 6 เชื่อว่า DSR (Data Set Ready) ถ้าเครื่องพิมพ์เป็น DCE และคอมพิวเตอร์เป็น DTE สายเส้นที่ 6 บนคอมพิวเตอร์ควรถูกเชื่อมต่อเข้ากับสายเส้นที่ 6 ของเครื่องพิมพ์ และเครื่องพิมพ์จะรักษาแรงดันไฟฟ้าบวกบนสายเส้นที่ 6 ตราบเท่าที่มันสามารถรับข้อมูล เมื่อมันต้องการที่จะบอกคอมพิวเตอร์ให้หยุดการส่งข้อมูล มันจะลดแรงดันไฟฟ้าบนสายที่ 6 ให้เป็นสถานะลบ

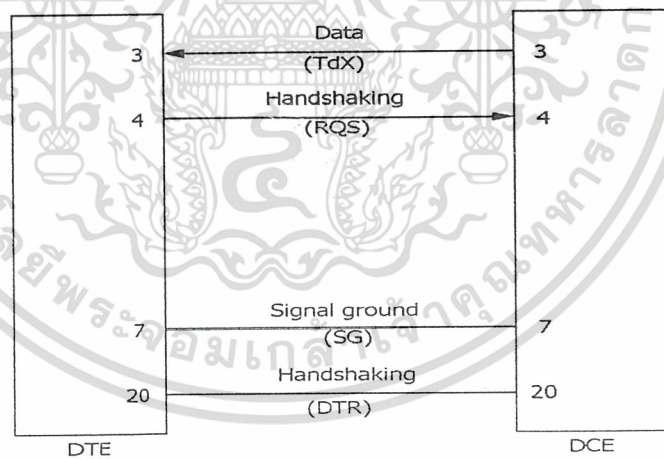
บ่อยครั้งที่วงจรแฮนเซ็คกิ้งชุด 2 ที่คือเส้นที่ 5 ถูกใช้โดยอุปกรณ์ DCE เพื่อควบคุมการส่งจากอุปกรณ์ DTE วงจรนี้มีชื่อว่า CTS (Clear To Send) เมื่อสายแฮนเซ็คกิ้งทั้งสองถูกใช้ อุปกรณ์ DTE ต้องได้รับการออกแบบให้ส่งข้อมูลต่อเมื่อสายทั้งสองเป็น ไฮ (high) หรือแรงดันไฟฟ้าบวก บางครั้งสายนั้นอาจมีความหมายต่างไป เช่น เส้นหนึ่งอาจบอกอุปกรณ์ผ่าน ส่งให้หยุดการพิมพ์ จนกระทั่งข้อมูลถูกพิมพ์ไปได้จำนวนหนึ่งและเส้นที่เหลืออาจแจ้งว่ากระดาษของเครื่องพิมพ์หมด อย่างไรก็ตามความหมายเหล่านี้ไม่ได้มีมาตรฐานเนื่องจากคอมพิวเตอร์หลายชนิดถูกโปรแกรมไม่ให้ส่งข้อมูล ถ้าแฮนเซ็คกิ้งทั้งสองไม่เป็น high แม้แต่กับเครื่องพิมพ์ที่ไม่ได้กำหนดความหมายชุดพิเศษกับสายชุดที่สองอย่างน้อยก็ควรรักษาแรงดันไฟฟ้าบวกไว้ อย่างไรก็ตามบางครั้งสัญญาณชุดที่สองต้องถูกสร้างหลอกขึ้นมาโดยการต่อมันเข้ากับชุดแรก

DCE ไปยัง DTE

เมื่ออุปกรณ์ DCE สื่อสารกับอุปกรณ์ DTE สายเส้นที่ 3 ต้องถูกใช้สำหรับส่งผ่านข้อมูล และถ้าต้องการแฮนเซ็คกิ้ง จะใช้เส้นที่ 20 เพื่อส่งแฮนเซ็คกิ้งจากอุปกรณ์ DTE ไปยังอุปกรณ์ DCE สายเส้นที่ 20 มีชื่อว่า DTR (Data Terminal Ready) สายแฮนเซ็คกิ้งชุดที่สองคือสายเส้นที่ 4 เรียกว่า Request to Send (RQS RT 1S4) ไม่ได้ถูกนำมาใช้เสมอไป



รูปที่ 2.8 การสื่อสารทางเดียวพร้อมด้วยแฮนด์เช็กกิ้งจาก DTE ไปยัง DCE

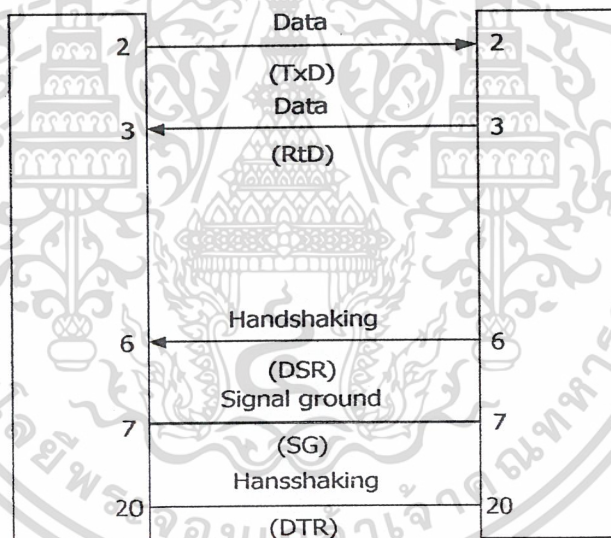


รูปที่ 2.9 การสื่อสารทางเดียวพร้อมด้วยแฮนด์เช็กกิ้งจาก DCE ไปยัง DTE

2.4.2 การสื่อสารสองทาง

หลายกรณีที่ข้อมูลถูกส่งผ่านในหลายทิศทาง โดยเฉพาะเมื่อคอมพิวเตอร์สองตัวสื่อสารกัน รวมทั้งกรณีที่ใช้ฮาร์ดแวร์แฮนด์เช็กกิ้งด้วยเช่นกัน จำนวนสายที่น้อยที่สุดที่จำเป็นในการสื่อสารเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

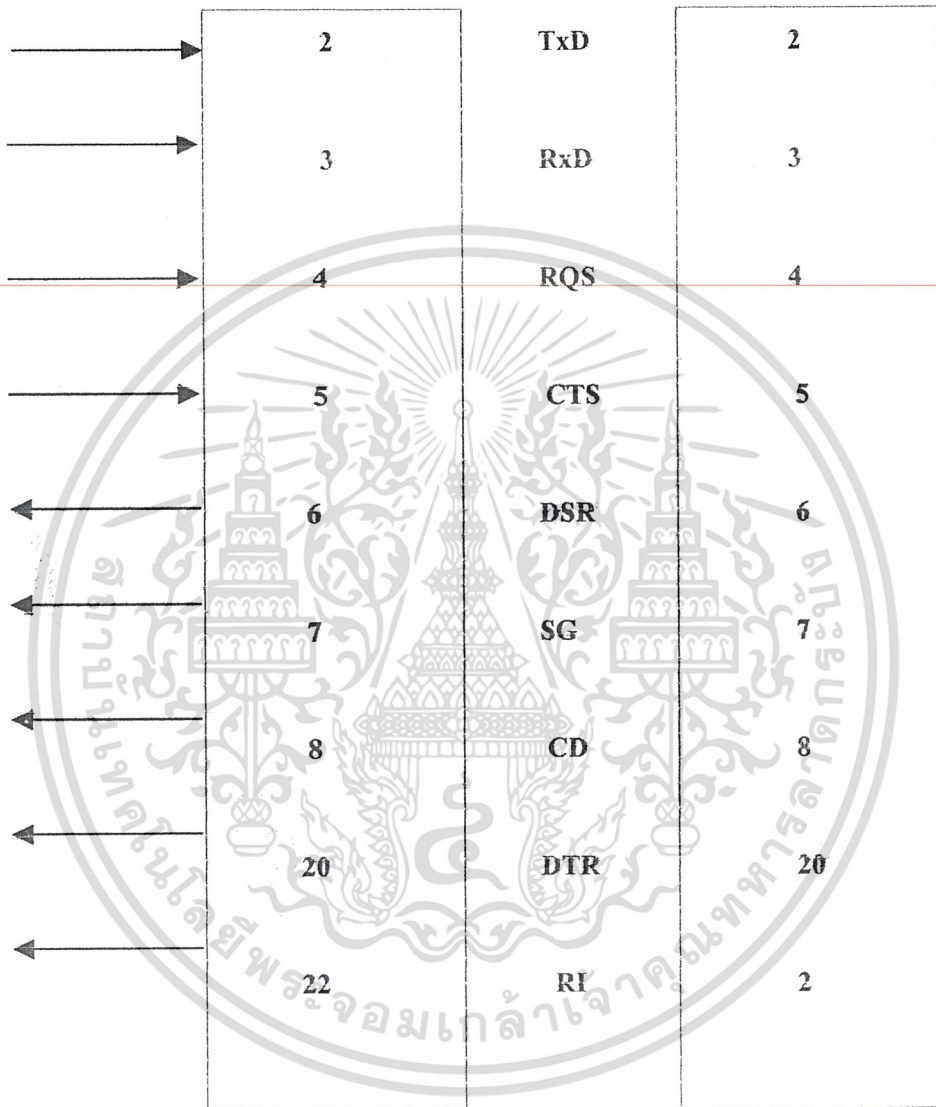
สารสองทางคือ สามเส้น ได้แก่ สายข้อมูลในแต่ละทิศทาง และซิกแนลกราวด์ การเพิ่มเติมสายแฮนด์เช็คกิ้งในแต่ละทิศทางทำให้สายรวมเป็นห้าเส้น ดังแสดงในรูปที่ 2.9 เมื่อสายแฮนด์เช็คกิ้งชุดที่สองถูกนำมาใช้เพิ่มเติมในแต่ละทิศทาง สายทั้งหมดที่ใช้มีเจ็ดเส้น เพื่อให้โมเด็มสามารถให้ข้อมูลมากขึ้น แก่คอมพิวเตอร์หรือเทอร์มินอล ได้แก่ CD (Carrier Detect) ถูกเชื่อมต่อเข้ากับขา 8 เพื่อแจ้งการคงอยู่ของสัญญาณพาหะ และ RI (Ring Indicator) ถูกเชื่อมต่อเข้ากับขา 22 เพื่อแสดงว่าโมเด็มกำลังถูกเรียกโดยอุปกรณ์ระยะไกล ซึ่งก็คือการตรวจสอบสัญญาณครั้งนั่นเอง จำนวนวงจรทั้งหมดจะกลายเป็น 9 วงจร ตามรูปที่ 2.10 แม้ว่ายังมีวงจรอื่นอีกหลายวงจรถูกกำหนดโดย RS-232-C แต่ทั้งเก้าวงจรมีถูกใช้กันมากที่สุด และเป็นเหตุผลที่ไมโครคอมพิวเตอร์ใช้หัวต่อ 5 ขาแทน 25 ขา สำหรับนำพาสัญญาณที่จำเป็นทั้งหมดของ RS-232-C



รูปที่ 2.10 การสื่อสารสองทางพร้อมด้วยวงจรแฮนด์เช็คกิ้งหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 แสดงการเชื่อมต่อ RS-232 มาตรฐานเก้าเส้น



2.4.3 นัลโมเด็ม (Null Modem)

ดังที่ได้กล่าวมาก่อนหน้านี้แล้วว่าในตอนแรก RS-232-C นำไปใช้สำหรับกำหนดการเชื่อมต่อระหว่างเทอร์มินอลซึ่งเป็น DTE กับโมเด็มซึ่งเป็น DCE ต่อมาได้มีการนำไปประยุกต์ที่ใช้กับการเชื่อมต่อระหว่างอุปกรณ์อื่นๆ อีกหลายชนิดที่ไม่ได้กำหนดให้เป็น DTE หรือ DCE อย่างเป็นกิจลักษณะ เช่น ไมโครคอมพิวเตอร์หรือเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการออกแบบ

3.1 ส่วนต่างของโครงการ

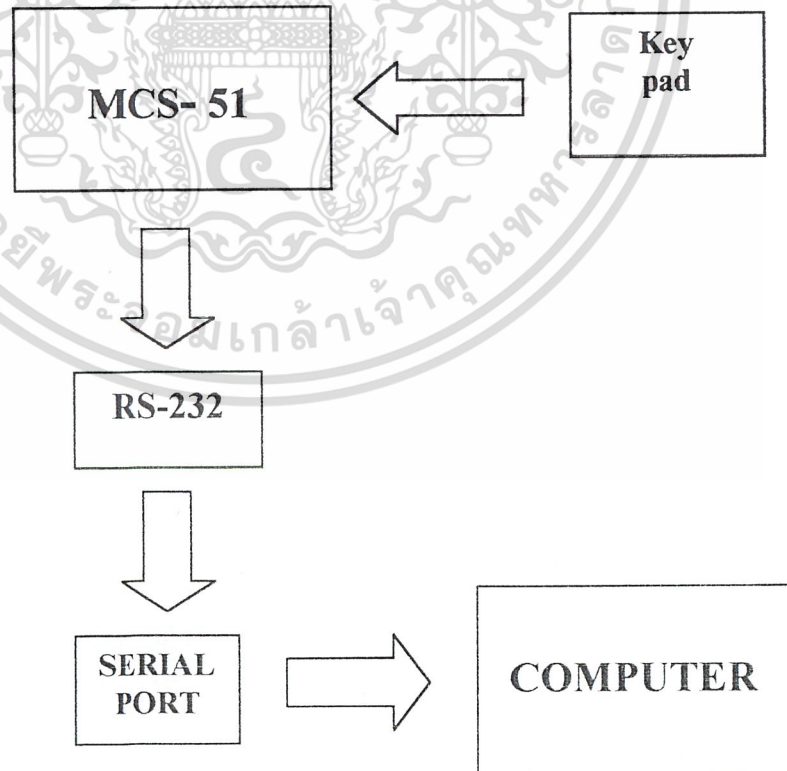
3.1.1 HARD WARE

ระบบตรวจสอบและจัดเก็บข้อมูลเครื่องจักรมีส่วนประกอบต่างๆทางด้าน hardware ดังต่อไปนี้

1. ส่วนแรกนั้นเป็นส่วนของ บอร์ด CP-AT32 Plus ซึ่งจะมีไมโครคอนโทรลเลอร์ MCS-51 และ port RS-232 เพื่อที่จะติดต่อทางด้าน SERIAL PORT

2. ส่วนที่สองนั้นคือ COMPUTER เพื่อที่จะรับค่าจาก CP-AT32 Plus ผ่านทาง SERIAL PORT และจะนำไปเก็บไว้ในฐานข้อมูลต่อไป

BLOCK DIAGRAM



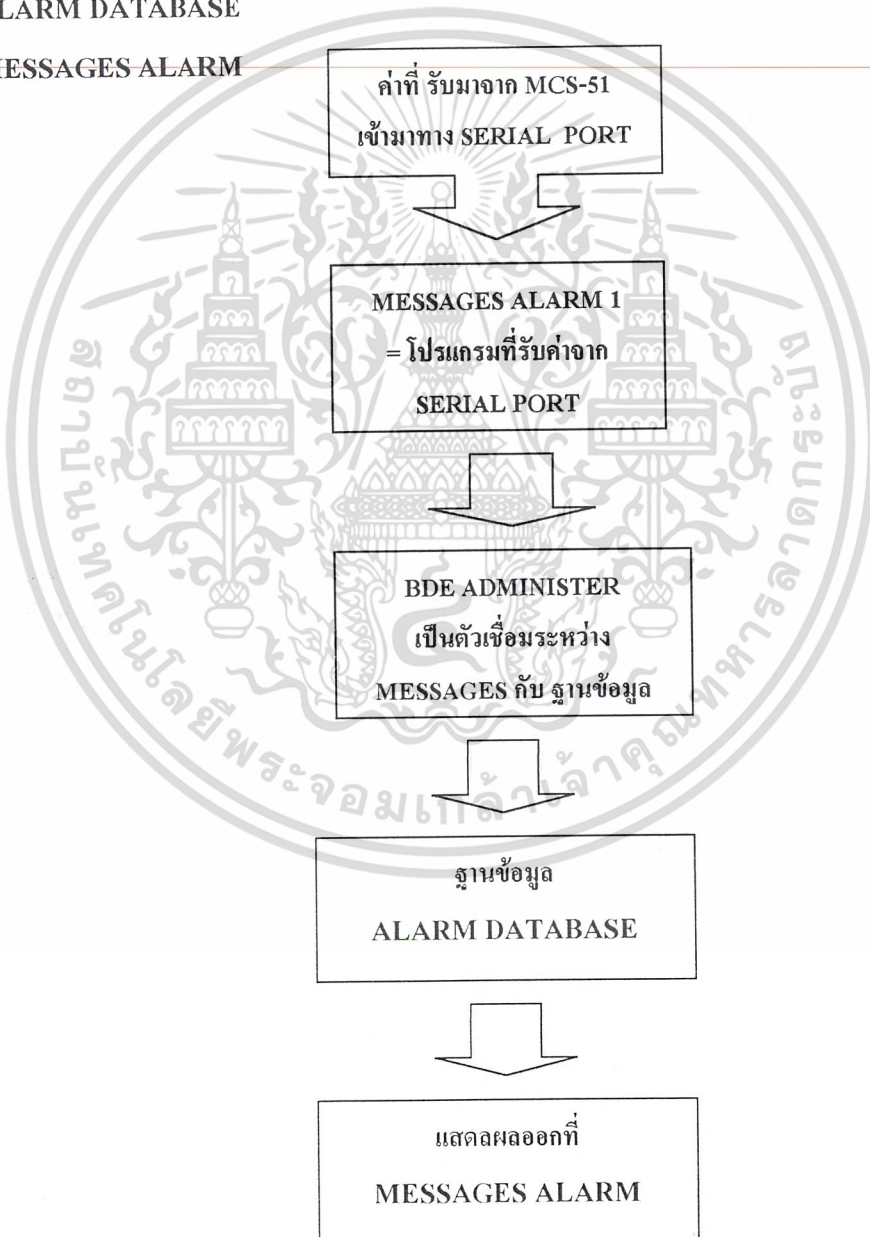
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.1 BLOCK DIAGRAM ของ HARDWARE
ใช้เฉพาะในวงการศึกษาเท่านั้น เมื่อผู้เผยแพร่เห็นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 SOFTWARE

การออกแบบในส่วนของ SOFTWARE นั้นจะต้องใช้ความเข้าใจและหลักการ และต้อง
 เข้าใจการเชื่อมต่อฐานข้อมูลด้วย

ในส่วนของ SOFTWARE นั้นจะมีส่วนประกอบดังต่อไปนี้

1. MICROCONTROLLER PROGRAM
2. BDE ADMINISTER
3. ALARM DATABASE
3. MESSAGES ALARM



รูปที่ 3.2 BLOCK DIAGRAM ของ SOFTWARE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 หลักการออกแบบ SOFTWARE

การออกแบบโปรแกรมเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ MCS-51 กับ SERIAL PORT

ในส่วนการทำงานของ SOFTWARE นั้นจะต้องมีการทำงานของโปรแกรมที่จะจัดทำขึ้นมารองรับการเชื่อมต่อระหว่าง ไมโครคอนโทรลเลอร์กับ COMPUTER ซึ่งในโปรแกรมนั้นจะต้องทำการเซต POTAL ซึ่งเป็นข้อกำหนดวิธีการส่งข้อมูล

สามารถทำได้โดยใช้ VB Control ที่ชื่อว่า MSComm โดยที่คุณต้อง กำหนด Custom Control เข้าไปที่ เมนู Project--->Components แล้วเลือกที่ช่อง MSComm ก็จะปรากฏ เป็นรูปไอคอนโทรศัพท์มือถือ ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของโปรแกรม

การเขียนโปรแกรมติดต่อกับ Serial Port สามารถทำได้ 2 วิธี คือ

1.การติดต่อแบบอินเตอร์รัพต์

ขบวนการอินเตอร์รัพต์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงการอุปกรณ์ จะส่งเป็นรหัสแอสกี เราจะเขียนโปรแกรมอินเตอร์รัพต์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

2.การติดต่อแบบโพลลิง

ในระบบพีซี การโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบต์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่ได้รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่า อัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่าข้อมูลที่ต้องส่งเข้ามา เรียกว่า "Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การ โพลแบบ "Round Robin" แทน แต่ใน VB เราจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียน โปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop

-**ComPort** คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อRS-232 (Com1,Com2)

Setting คือ เราต้องกำหนดอัตรา Baud,Parity,Data(จำนวนบิต),Stop ตัวอย่าง 1200,n,8,1 เป็นต้น

1. Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน

ตัวอย่าง `MSComm1.CommPort=1`

ในที่นี้เลือกจะใช้ Com1อยู่ที่ด้านหลังเครื่องคอมพิวเตอร์

2. Property ชื่อ Settings คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตรา ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

`MSComm1.Settings="Baud(อัตราการรับส่งข้อมูล),(ถ้าไม่ใช่ใส่ N,จำนวนบิตข้อมูล,บิตสตอป"`

ตัวอย่าง `MSComm1.Settings="1200,N,8,1"`

3.Property ชื่อ InputLen คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์

ตัวอย่าง `MSComm1.InputLen=1`

4. Property ชื่อ PortOpen คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด =True ถ้าปิด =False

ตัวอย่าง `MSComm1.PortOpen=True`

5. Property ชื่อ Rthreshold คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล (Comport) มันทำให้เกิดCommEvent ใน OnComm Event

ตัวอย่าง `MSComm1.Rthreshold=1`

วิธีการรับส่งข้อมูลจาก SerialPort

จากวิธีเขียน โปรแกรมด้านบนเป็นการกำหนดค่าเริ่มต้นให้กับคอมพอร์ตและเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้นก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้
Output = ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต **Input =** เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่งจะอ่านข้อมูลเข้ามาจากทางพอร์ต RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ VB เขียนโปรแกรมติดต่อ I/O ผ่านทาง Port ของเครื่องคอมพิวเตอร์ ไม่ว่าจะผ่านทาง Serial Port (RS-232) หรือที่รู้จักในชื่อ Com1, Com2 และ Parallel Port หรือ Printer Port นั้นเอง หรืออาจใช้ Card I/O 8255 ซึ่งเป็นการขยาย Port I/O ของ Parallel ก็สามารถทำการติดต่อกับ Hardware ภายนอกผ่าน Port ดังที่กล่าวมาได้ อีกทั้งยังสามารถติดต่อผ่านระบบ Network โดยผ่านช่องทางการติดต่ออย่าง TCP/IP จะเห็นได้ว่า VB ก็สามารถทำงานด้าน Interface Hardware ได้ไม่แพ้ภาษาตัวอื่นๆและง่ายกว่า จึงทำให้สามารถทำความเข้าใจเพื่อจะนำไปใช้งานได้สะดวกและรวดเร็ว

Serial Port (RS-232)

สามารถติดต่อกับอุปกรณ์ต่างๆที่มีการติดต่อกับอุปกรณ์ภายนอกผ่านทาง RS-232 อย่างเช่น เครื่องขั้วนำหนัก รวมถึง โทลคเซล(เป็นเซ็นเซอร์ชนิดหนึ่งใช้สำหรับวัดน้ำหนักซึ่งที่ชุดแสดงผลภายในเป็นชุดไมโครคอนโทรลเลอร์ จะมีสัญญาณรับส่งทาง RS-232) เครื่องวัดงานทางด้านไฟฟ้า, ไมโครคอนโทรลเลอร์, ความคุมอุปกรณ์ไฟฟ้า, โอนถ่ายข้อมูลในฮาร์ดดิสก์ระหว่างเครื่องคอมพิวเตอร์ด้วยกัน, ความคุมเสตปปีงมอเตอร์ เป็นต้น ข้อดีของการติดต่อข้อมูลกันผ่านทาง RS-232 ก็คือสามารถใช้ได้ในระยะทางไกลๆระหว่างอุปกรณ์ ที่ติดต่อกัน

เนื่องจากที่ Microsoft Visual Basic 5,6 จะมีตัวคอนโทรลชื่อ MS Comm ที่ใช้ติดต่อกับ Serial Port (RS-232) ให้ไว้อยู่แล้วไม่จำเป็นต้องเขียนโปรแกรมให้ยุ่งยากทำให้การพัฒนาโปรแกรมในด้านนี้ได้เร็วและเป็นมาตรฐานในการเขียนโปรแกรมเดียวกันของทุกโปรแกรม

บทที่ 4

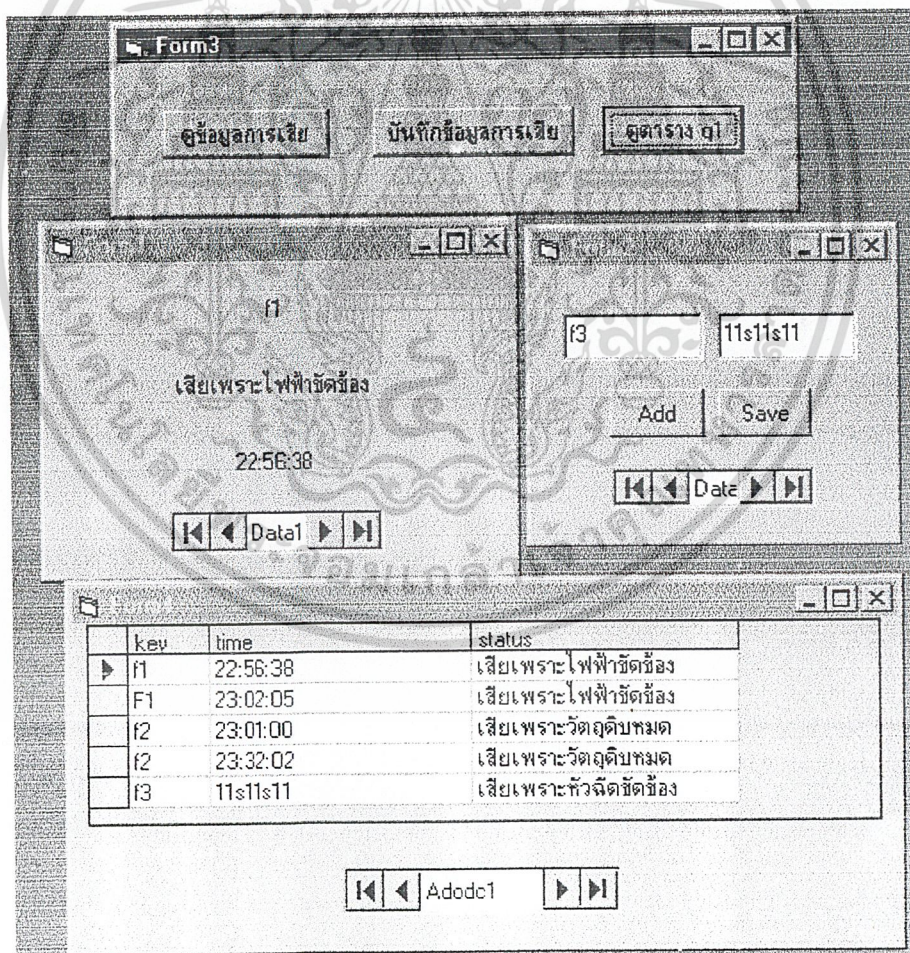
ผลการทดลอง

4.1 การติดต่อฐานข้อมูล

- เพื่อที่ทดลองการใช้งานของฐานข้อมูล
- เพื่อที่จะทดลองการดึงข้อมูล มาสู่ ฐานข้อมูล

ผลที่ได้

สามารถที่จะติดต่อระหว่างฐานข้อมูลและส่งค่าที่เป็นรหัสตัวแปรเพื่อที่จะส่งและเก็บไว้ในฐานข้อมูล และแสดงผลออกมาได้อย่างถูกต้อง



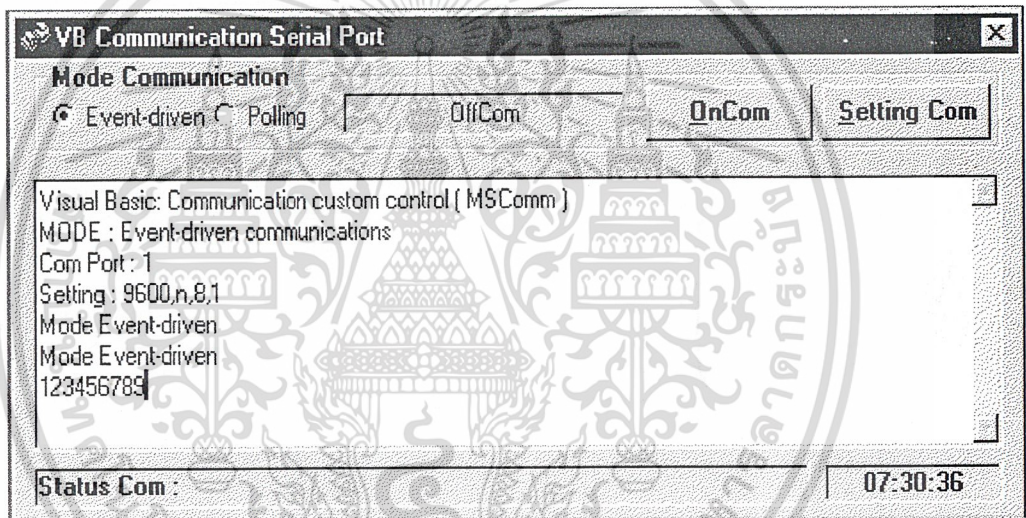
รูปที่ 4.1 ตัวอย่าง PROGRAM ที่ทดลองติดต่อกับฐานข้อมูล
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การเชื่อมต่อ RS-232 กับ SERIAL PORT

-เพื่อที่จะทดลองส่งค่าจากบอร์ด CP-AT32Plus ผ่าน ซีเรียลพอร์ตเพื่อติดต่อกับcomputer โดยใช้มาตรฐาน RS-232

ผลที่ได้

-สามารถที่จะรับค่าจาก MCS-51 ที่ส่งมาและแสดงค่าของมาเป็นตัวเลข 1-9 ผ่านโปรแกรม VB Communication serial Port



รูปที่ 4.2 ตัวอย่าง PROGRAM ที่ทดลองติดต่อกับ ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

ในการออกแบบในส่วนต่างๆของโครงการ ระบบตรวจสอบและจัดเก็บข้อมูลเครื่องจักร จะต้องคำนึงถึงประโยชน์และผลลัพท์ที่จะออกมาเมื่อเสร็จแล้ว ที่จะต้องมีความเที่ยงตรงเกิด ความผิดพลาดน้อยที่สุด และสามารถนำประยุกต์ให้ใช้งานได้ โดยโครงการนี้จะสามารถนำไป จัดเก็บข้อมูลเรื่องปัญหาต่างๆของเครื่องจักร ซึ่งจะทำงานโดยผ่านทาง KEY PAD เพื่อทำการรับ ค่ามาซึ่งจะมีไมโครคอนโทรลเลอร์คอยรับค่ามาแล้วส่งผ่านมายัง RS-232 เพื่อที่สามารถที่จะเชื่อม ติดต่อกับ COMPUTER โดยผ่านทาง SERIAL PORT เพื่อที่ COMPUTER จะรับค่าที่ส่งมาและ แทนค่าที่เป็นตัวแปรนั้นให้เป็นสาเหตุปัญหาที่เกิดขึ้น และจัดเก็บเอาไว้เป็นฐานข้อมูล เพื่อที่จะ สามารถที่จะนำข้อมูลเหล่านั้นมาประมวลผลต่อไป

ปัญหาที่พบในการทำโครงการนี้คือ การสื่อสารระหว่าง ไมโครคอนโทรลเลอร์กับ COMPUTER จะต้องเขียนโปรแกรมขึ้นมารับข้อมูลที่จะส่งมา ซึ่งจะมีโปรแกรมหลายชนิดที่ สามารถจะใช้ได้ ตอนแรกนั้นยังไม่สามารถติดต่อได้ และใช้เวลาศึกษาอยู่นานจึงจะเขียน โปรแกรมขึ้นมาติดต่อได้ โดยตอนแรกนั้นยังไม่มีการเชื่อมต่อกับฐานข้อมูล เพียงแค่แสดงผล ออกมาให้เห็นก่อน แล้วจึงทำฐานข้อมูลขึ้นมาเชื่อมต่อกัน

ในการพัฒนาต่อไปนั้นสามารถที่จะนำไปประยุกต์ต่อไปได้ โดยการใช้วงจรเซ็นเซอร์มา ติดตั้งในส่วนต่างๆของเครื่องจักร เพื่อจะตรวจจับปัญหาต่างๆที่จะเกิดกับเครื่องจักร อาจจะเป็น เซ็นเซอร์ที่ตรวจจับความร้อนหรืออาจจะเป็น Limit Switch หรือเป็น รีเลย์ ซึ่งวงจรเซ็นเซอร์เหล่านี้ จะส่งผลที่ได้ให้ไมโครคอนโทรลเลอร์รับค่าเซ็นเซอร์นั้นมาเพื่อที่จะส่งผ่าน RS-232 เพื่อจะให้ COMPUTER นำข้อมูลเหล่านั้นประมวลผลต่อไป และในส่วนของ SOFTWARE นั้นสามารถจะ เพิ่มขีดความสามารถการทำงานของโปรแกรมได้ ซึ่งในโครงการนี้นั้นเป็นการจำลองการเก็บ ข้อมูลของเครื่องจักรเพียงตัวเดียว ซึ่งจะสามารถที่จะพัฒนาเพิ่มการทำงานของระบบนี้กับ เครื่องจักรหลายๆตัวได้

บรรณานุกรม

1. เริ่มแรกกับคอมพิวเตอร์ เรียบเรียง จาก YOUR FIRST COMPUTER ของ Alan Simpson เรียบเรียงโดย จิระพล ฉายษุติศ
2. รู้จักการทำงานของ CP-AT32plus ของ บริษัท ETT จำกัด
3. เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ของ ATMEL เรียบเรียงโดย วรพจน์ กรแก้ววัฒนากุล, =ชัชวัฒน์ ถิมพรจิตรวิไล
4. ดร.ประสิทธิ์ ประพัฒน์มงคลการ, "หลักการระบบสื่อสาร", บริษัทซีเอ็ด ยูเคชั่น จำกัด,

2532

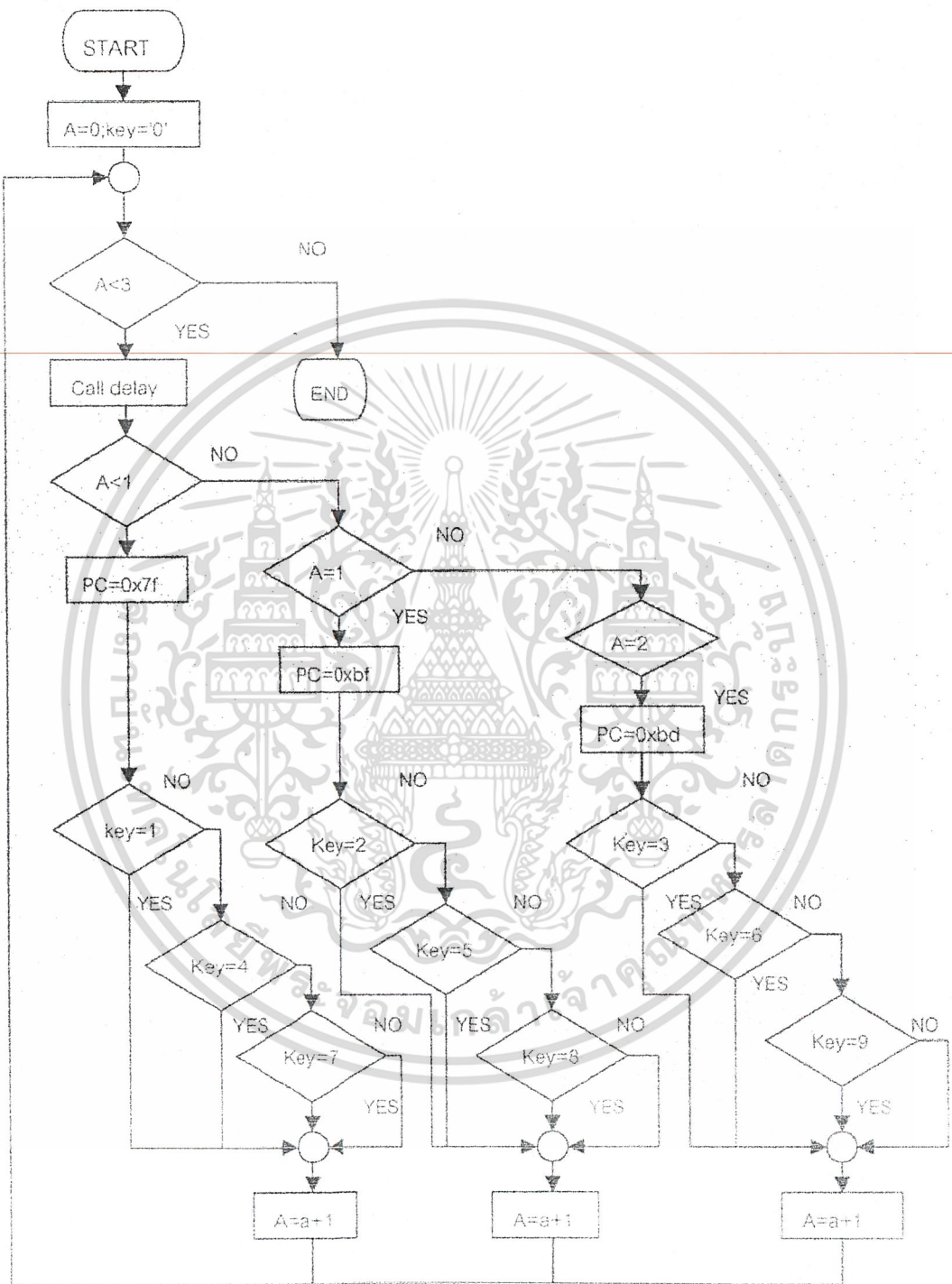


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

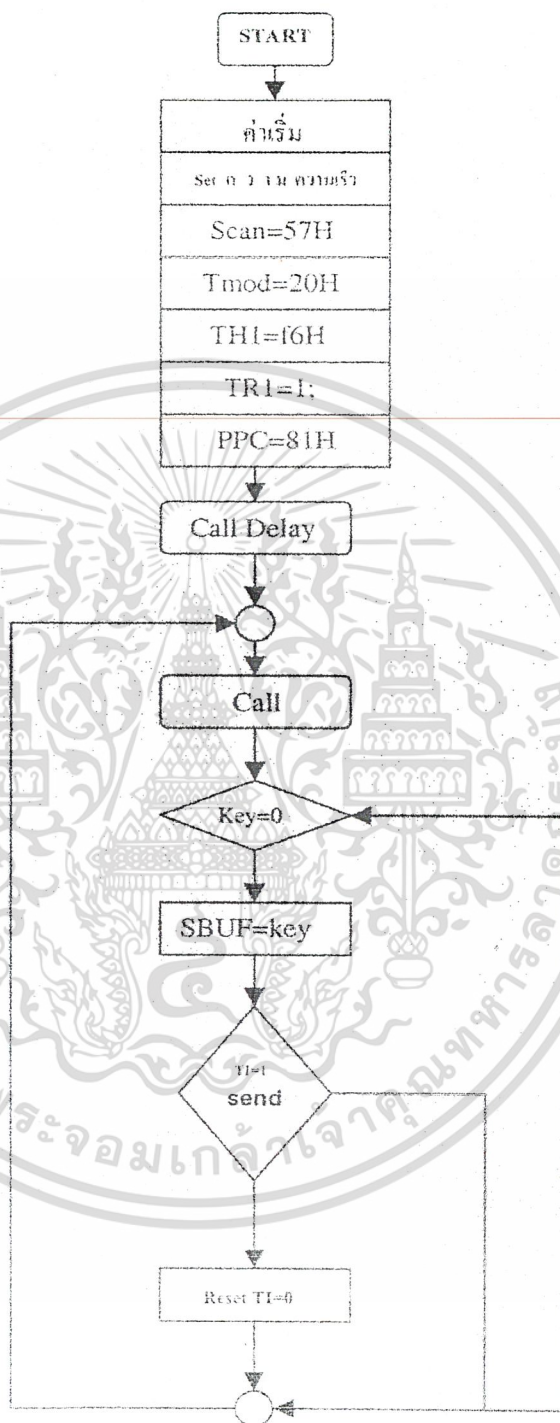


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOWCHART ของ โปรแกรมการ Scan key

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Flowchart ของ โปรแกรม ใน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; SOLINOI.SRC generated from: SOLINOI.C

\$NOMOD51

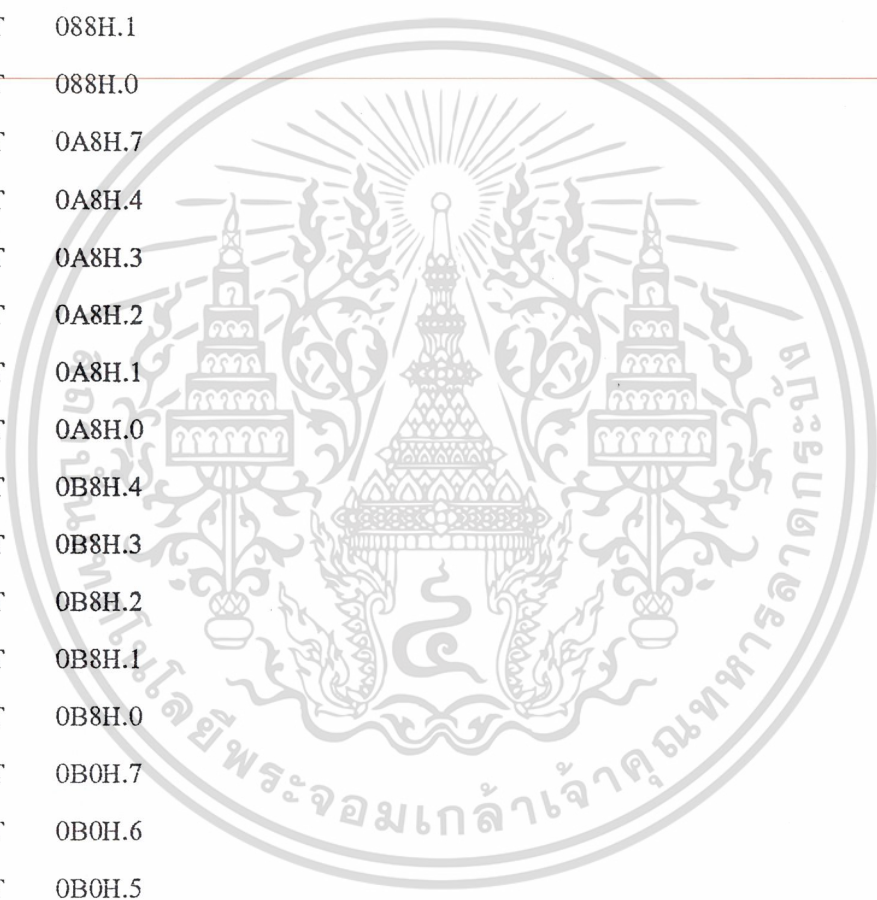
NAME SOLINOI

P0 DATA 080H
P1 DATA 090H
P2 DATA 0A0H
P3 DATA 0B0H
PSW DATA 0D0H
ACC DATA 0E0H
B DATA 0F0H
SP DATA 081H
DPL DATA 082H
DPH DATA 083H
PCON DATA 087H
TCON DATA 088H
TMOD DATA 089H
TL0 DATA 08AH
TL1 DATA 08BH
TH0 DATA 08CH
TH1 DATA 08DH
IE DATA 0A8H
IP DATA 0B8H
SCON DATA 098H
SBUF DATA 099H
CY BIT 0D0H.7
AC BIT 0D0H.6
F0 BIT 0D0H.5
RS1 BIT 0D0H.4
RS0 BIT 0D0H.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OV	BIT	0D0H.2
P	BIT	0D0H.0
TF1	BIT	088H.7
TR1	BIT	088H.6
TF0	BIT	088H.5
TR0	BIT	088H.4
IE1	BIT	088H.3
IT1	BIT	088H.2
IE0	BIT	088H.1
IT0	BIT	088H.0
EA	BIT	0A8H.7
ES	BIT	0A8H.4
ET1	BIT	0A8H.3
EX1	BIT	0A8H.2
ET0	BIT	0A8H.1
EX0	BIT	0A8H.0
PS	BIT	0B8H.4
PT1	BIT	0B8H.3
PX1	BIT	0B8H.2
PT0	BIT	0B8H.1
PX0	BIT	0B8H.0
RD	BIT	0B0H.7
WR	BIT	0B0H.6
T1	BIT	0B0H.5
T0	BIT	0B0H.4
INT1	BIT	0B0H.3
INT0	BIT	0B0H.2
TXD	BIT	0B0H.1
RXD	BIT	0B0H.0
SM0	BIT	098H.7
SM1	BIT	098H.6
SM2	BIT	098H.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REN	BIT	098H.4
TB8	BIT	098H.3
RB8	BIT	098H.2
TI	BIT	098H.1
RI	BIT	098H.0
P00	BIT	080H.0
P01	BIT	080H.1
P02	BIT	080H.2
P03	BIT	080H.3
P04	BIT	080H.4
P05	BIT	080H.5
P06	BIT	080H.6
P07	BIT	080H.7
P10	BIT	090H.0
P11	BIT	090H.1
P12	BIT	090H.2
P13	BIT	090H.3
P14	BIT	090H.4
P15	BIT	090H.5
P16	BIT	090H.6
P17	BIT	090H.7
P20	BIT	0A0H.0
P21	BIT	0A0H.1
P22	BIT	0A0H.2
P23	BIT	0A0H.3
P24	BIT	0A0H.4
P25	BIT	0A0H.5
P26	BIT	0A0H.6
P27	BIT	0A0H.7
P30	BIT	0B0H.0
P31	BIT	0B0H.1
P32	BIT	0B0H.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
P33 BIT 0B0H.3
P34 BIT 0B0H.4
P35 BIT 0B0H.5
P36 BIT 0B0H.6
P37 BIT 0B0H.7
```

```
?PR?main?SOLINOI SEGMENT CODE
```

```
EXTRN CODE (?C_STARTUP)
```

```
PUBLIC main
```

```
; #include <reg51.h>
```

```
;
```

```
; void main (void)
```

```
RSEG ?PR?main?SOLINOI
```

```
USING 0
```

```
main:
```

```
; SOURCE LINE # 3
```

```
;
```

```
; SOURCE LINE # 4
```

```
;
```

```
; SOURCE LINE # 5
```

```
RET
```

```
; END OF main
```

```
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <reg51.h> /* เรียก library reg51.h */
#include <absacc.h> /* เรียก library absacc.h */

#define PC XBYTE[0xA002] /* ทำ Memory Map ที่ตำแหน่ง 0A002 */
#define PPC XBYTE[0xA003] /* เรียก Memory Map ที่ตำแหน่ง 0A003 */

unsigned char key; /* กำหนดตัวแปรแบบ char ชื่อ Key */

void delay (int x) /* ทำ sub function ชื่อ delay */
{
    for(x=x;x>0;x--); /* ทำการวนลูป */
}

void scankey (void) /* ทำ sub function ชื่อ scankey */
{
    unsigned char a; /* กำหนดตัวแปรขนาด char ชื่อ a */

    a=0; /* กำหนดให้ a เท่ากับ 0 */
    key='0'; /* กำหนดให้ key มีค่า ascii "0" */
    while(a<3) /* กำหนดการวนลูป ถ้า a มีค่าน้อยกว่า 3 */
    {
        delay(1000); /* กระโดด ไป sub function delay */
        if(a<1) /* ตรวจสอบว่า a น้อยกว่า 1 หรือไม่ */
        {
            PC=0x7F; /* กำหนด ค่าของ PC เป็น 7FH */
            if(PC==0x77) /* ตรวจสอบว่า PC มีค่าเท่ากับ 77H หรือไม่ */
            {
                key='1'; /* ถ้าใช่ ให้ Key มีค่าเท่ากับ ascii "1" */
                delay(500); /* กระโดด ไป sub function delay */
                while(PC==0x77); /* วน ลูปถ้า PC มีค่าเท่ากับ 77H */
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    else

    if(PC==0x7b) /* ถ้า PC มีค่าเท่ากับ 7BH */
    {
        key='4'; /* ให้ Key มีค่ากับ ascii "4" */
        delay(500); /* กระโดด ไป sub funtion delay */
        while(PC==0x7b); /* วนลูปถ้า PC เท่ากับ 7BH */
    }

    else

    if(PC==0x7d) /* ถ้า PC มีค่าเท่ากับ 7DH */
    {
        key='7'; /* ให้ Key มีค่ากับ ascii "7" */
        delay(500); /* กระโดด ไป sub funtion delay */
        while(PC==0x7d); /* วนลูปถ้า PC เท่ากับ 7DH */
    }
    a++; /* เพิ่ม a ไปอีก 1 */
}

clsc

if(a==1) /* ถ้า a เท่ากับ 1 */
{
    PC=0xbf; /* กำหนดให้ PC มีค่าเท่ากับ 0DFH */
    if(PC==0xb7) /* ตรวจสอบว่า PC มีค่าเท่ากับ 0B7H หรือไม่ */
    {
        key='2'; /* ให้ Key มีค่ากับ ascii "2" */
        delay(500); /* กระโดด ไป sub funtion delay */
        while(PC==0xb7); /* วนลูปถ้า PC เท่ากับ 0B7H */
    }

    else

    if(PC==0xbb) /* ถ้า PC มีค่าเท่ากับ 0DDH */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

key='5'; /* ให้ Key มีค่ากับ ascii "5" */
    delay(500); /* กระโดด ไป sub funtion delay */
    while(PC==0xbb); /* วนลูปถ้า PC เท่ากับ 0BDH */
}

clsc

```

```

if(PC==0xbd) /* ตรวจสอบว่า PC มีค่าเท่ากับ 0BDH หรือไม่ */
{
key='8'; /* ให้ Key มีค่ากับ ascii "8" */
    delay(500); /* กระโดด ไป sub funtion delay */
    while(PC==0xbd); /* วนลูปถ้า PC เท่ากับ 0BDH */
}
a++; /* เพิ่ม a ไปอีก 1 */
else
if(a==2) /* ถ้า a มีค่าเท่ากับ 2 */
{
PC=0xdf; /* กำหนดให้ PC มีค่าเท่ากับ 0DFH */
if(PC==0xd7) /* ตรวจสอบว่า PC มีค่าเท่ากับ 0D7H หรือไม่ */
{
key='3'; /* ให้ Key มีค่ากับ ascii "3" */
    delay(500); /* กระโดด ไป sub funtion delay */
    while(PC==0xd7); /* วนลูปถ้า PC เท่ากับ 0D7H */
}
else
if(PC==0xdb) /* ตรวจสอบว่า PC มีค่าเท่ากับ 0DBH หรือไม่ */
{
key='6'; /* ให้ Key มีค่ากับ ascii "6" */
    delay(500); /* กระโดด ไป sub funtion delay */
    while(PC==0xdb); /* วนลูปถ้า PC เท่ากับ 0DBH */
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else
    if(PC==0xdd) /* ตรวจสอบว่า PC มีค่าเท่ากับ 0DDH หรือไม่ */
    {
    key='9'; /* ให้ Key มีค่ากับ ascii "9" */
    delay(500); /* ระยะเวลา ไป sub function delay */
    while(PC==0xdd); /* วนลูปถ้า PC เท่ากับ 0DDH */
    }
}
}
}

void main (void) /* function Main */
{
    unsigned char chksend;

    SCON=0x52; /* set SFR SCON มีค่าเท่ากับ 52H */
    TMOD=0x20; /* set SFR TMOD มีค่าเท่ากับ 20H */
    TH1=0xf6; /* SET TH1 มีค่าเท่ากับ 0F6H */
    TR1=1; /* กำหนดการ start Timer 1 */

    PPC=0x81; /* set 8255 M1M1 < out, out, in, out >
    delay(1000); /* ระยะเวลา ไป sub function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

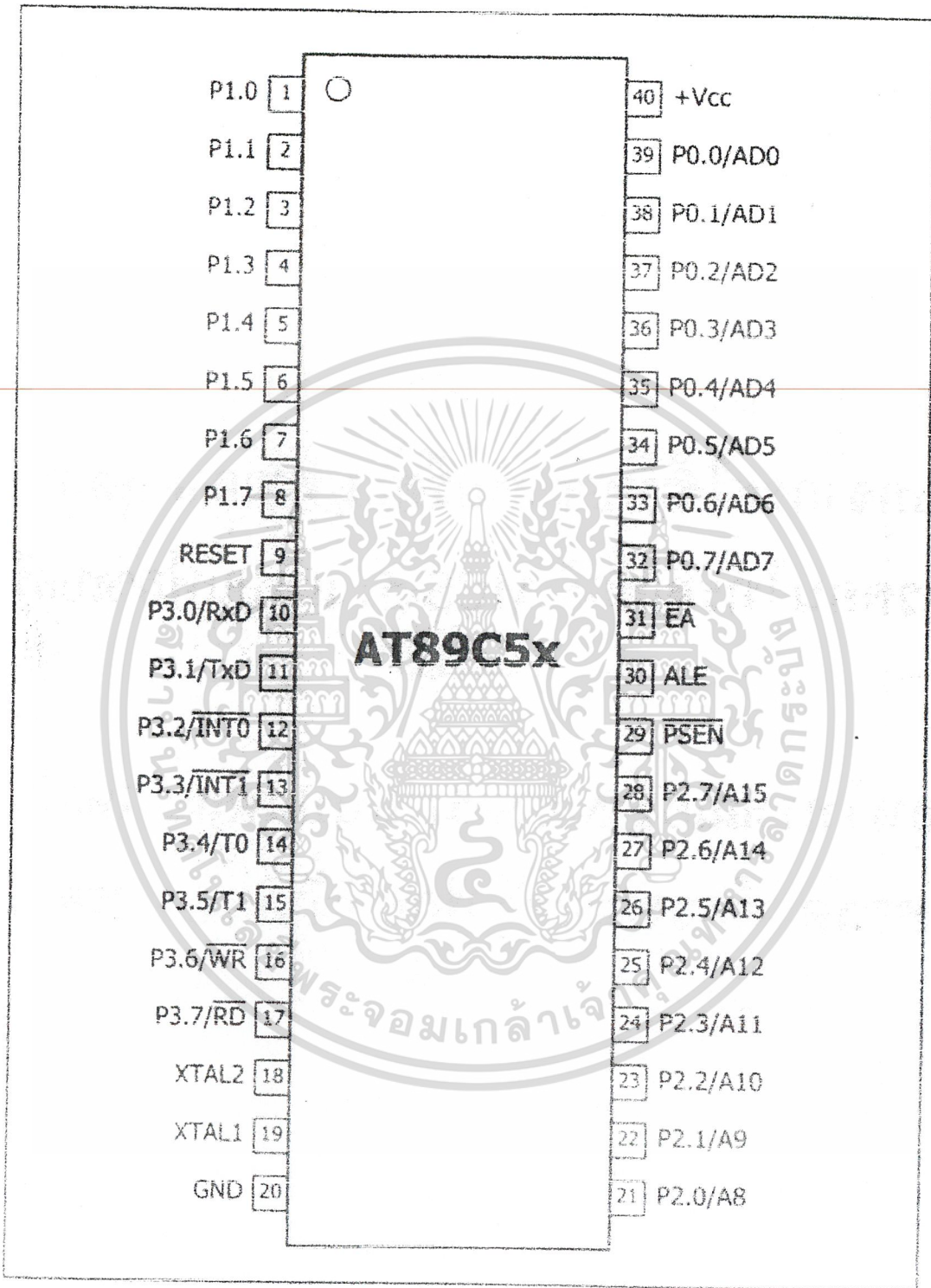
```

while(1) /* วนลูป */
{
    scankey(); /* กระโดด ไป sub funtuon scan key */
    if(key!='0') /* ถ้า Key มีค่าไม่เท่ากับ ascii "0" */
    {
        if(TI==1) /* ถ้า SFR TI มีค่าเท่ากับ 1 */
        {
            TI=0; /* set SFR TI ให้มีค่าเท่ากับ 0 */
            SBUF=key; /* Load ค่าตัวแปร Key เข้า SFR SBUF */
        }
    }
}
}
}

```

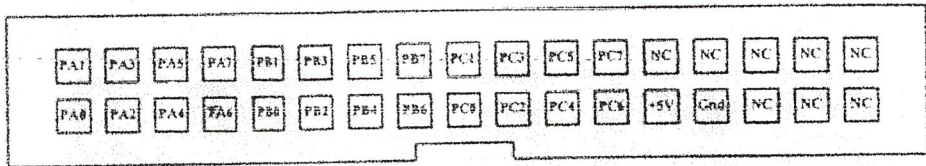


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

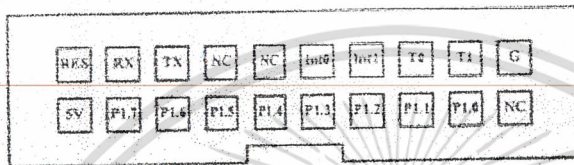


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

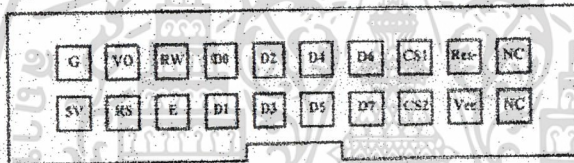
72IOZ80



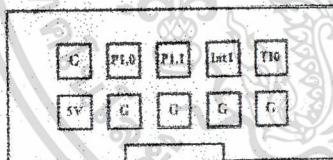
EXP1



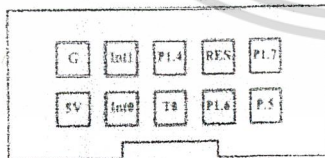
LCD



I2C



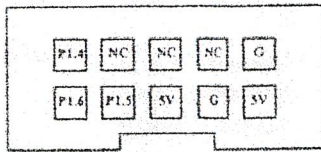
SPI



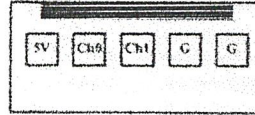
CONNECTER ต่างๆของ CP-AT32 plus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

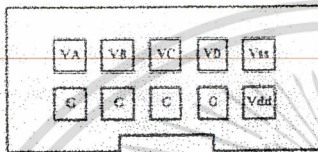
ET-SDP8



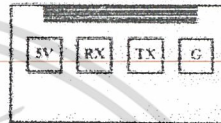
ADC



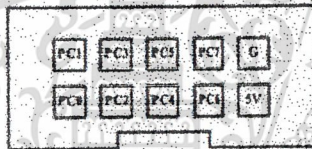
DAC



RS232



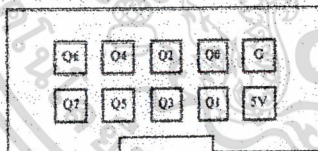
KEY



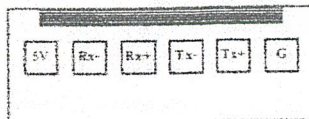
1-WIRE



EXP2

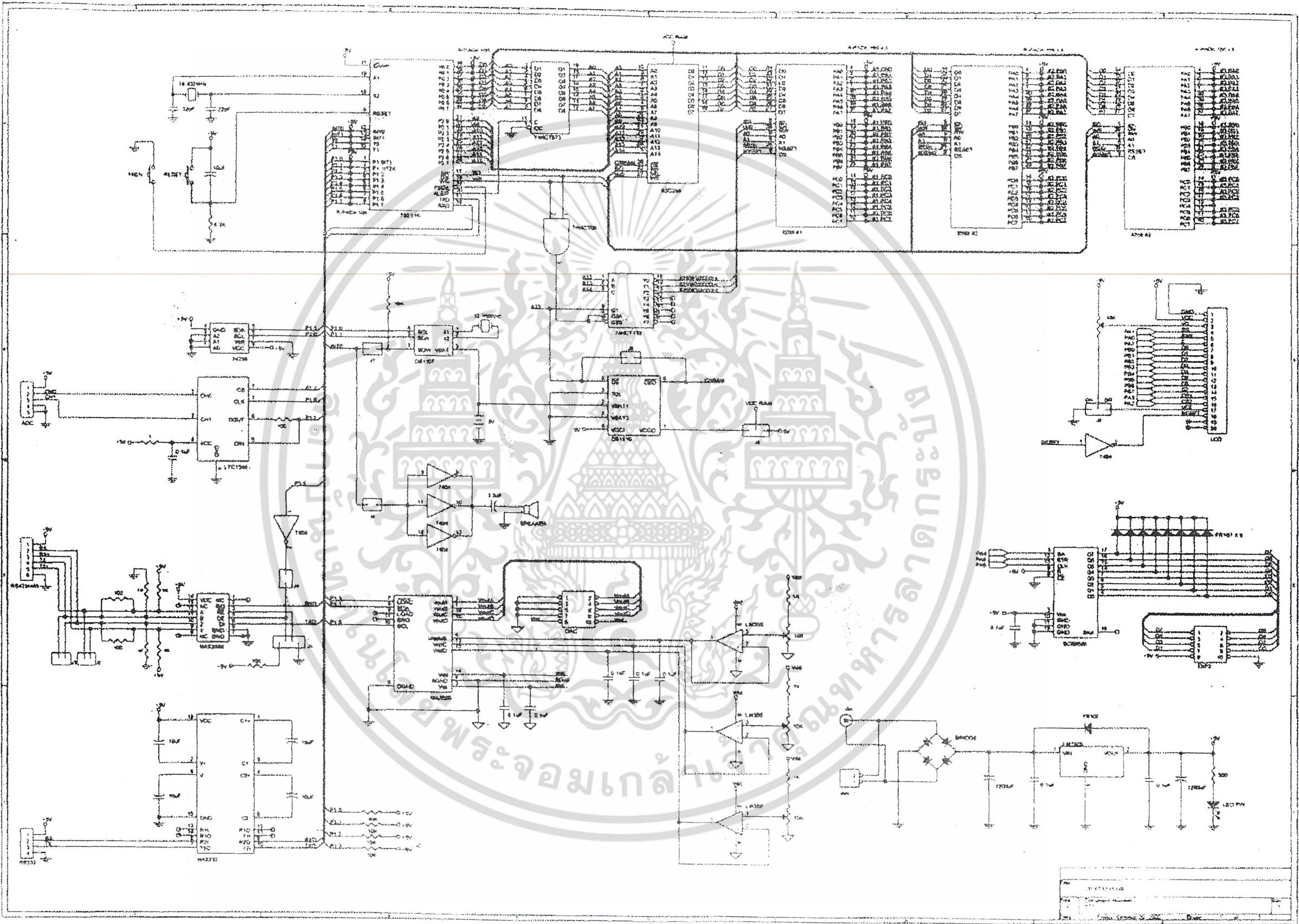


RS422/RS485



CONNECTER ต่างๆของ CP-AT32 plus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้