สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบห้องน้ำสาธารณะหยอดเหรียญ

PUBLIC TOILET COIN SYSTEM

นายชัช เหลือบุญนุ่ม

นายปราโมทย์ ชัยชนะมงคล

เลขหมู่.....
เลขทะเบียน 50433
วัน,เดือน,ปี 13 พ.ค. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PUBLIC TOILET COIN SYSTEM

CHAT LUABUNNUM

PRAMOTE CHAICHANAMONGKHOL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ ระบบห้องน้ำสาธารณะหยอดเหรียญ

PUBLIC TOILET COIN SYSTEM

นักศึกษาผู้จัดทำ นายชัช เหลือบุญนุ่ม รหัสประจำตัว 43015506

นายปราโมทย์ ชัยชนะมงคล รหัสประจำตัว 43015518

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2545

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
อาจารย์ สาท คำมูล	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 22 เมษายน พ.ศ. 2546

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผ.ศ.ประสิทธ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบห้องน้ำสาธารณะหยอดเหรียญ
	PUBLIC TOILET COIN SYSTEM
นักศึกษาผู้จัดทำ	นายชัช เหลือบุญนุ่ม
	นายปราโมทย์ ชัยชนะมงคล
อาจารย์ที่ปรึกษา	อาจารย์ สาท คำมูล
ปีการศึกษา	2545

บทคัดย่อ

ปัจจุบันเทคโนโลยีได้ก้าวหน้าไปอย่างรวดเร็วไม่ว่าจะเป็นการโทรคมนาคม เครื่องมือสื่อสาร เครื่องอำนวยความสะดวกต่างๆ ซึ่งได้มีการนำระบบอัตโนมัติเข้ามาใช้งานแทนการทำงานของคน เช่นเครื่องขายน้ำอัดลม เครื่องขายตั๋วรถไฟฟ้าเป็นต้น รวมไปถึงงานบริการต่างๆ ที่ปกติจะใช้การทำงานของคน แต่ก็ยังมีงานอีกหลายๆ อย่างที่ยังใช้คนทำงานอยู่ และการให้บริการห้องน้ำสาธารณะก็เป็นอีกอย่างหนึ่งที่ยังคงใช้คนทำงานเป็นผู้เก็บเงินค่าบริการอยู่ ซึ่งการใช้คนเป็นผู้เก็บเงินค่าบริการนี้ จะต้องมีการคัดเลือกคนที่จะเข้ามาทำงาน เสียค่าจ้างแก่คนที่ทำงาน และยังเสี่ยงกับการทุจริตที่อาจเกิดขึ้น เช่นการยักยอกเงินค่าบริการ การนำสินค้ามาขายแทนเงินทอนที่ผู้บริการควรได้รับ ดังนั้นเราจึงคิดที่จะสร้างเครื่องเก็บเงินค่าบริการอัตโนมัติ โดยใช้การควบคุมจากไมโครคอนโทรลเลอร์ ซึ่งใช้งานง่าย และเป็นที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน โดยเครื่องที่ทำการสร้างขึ้นนั้นมีการทำงานที่เที่ยงตรง การทอนเหรียญและการคิดค่าบริการยุติธรรม และจะช่วยแก้ปัญหาการทุจริตได้ และยังช่วยลดค่าใช้จ่ายที่เกิดขึ้นจากค่าจ้างผู้เก็บค่าบริการด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Public Toilet Coin System
Authors	Mr.Chat Luabunnum
	Mr.Pramote Chaiahanamongkhol
Thesis Advisor	Mr.Sart Kummoon
Year	2002

ABSTRACT

At present, technology is moving very fast, as we can see from telecommunication communication equipment, witch used an automatic machine instead of human power, such as, soft drink selling machine, BTS ticket selling machine, etc. and include other services.

However, some services are still need human power, public toilet is one of example services that need to hire someone to collect money. This has to select someone to response this job, pay his/her salary and might be face a risk of cheating from his/her.

Therefore, we would like to create machine to automatically collect money by controlling from micro controller, which is easily and widely used.

The machine we created is working accurately with a fairly calculate return money and service charge, it also can reduce a problem of cheating and cost of employment.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดีเพราะได้รับความกรุณาจาก อาจารย์สาท คำมูล ที่ได้ให้คำแนะนำแก่ผู้จัดทำตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์ เครื่องมือต่างๆ รวมทั้งให้ใช้ห้องทดลองนิวแมติกส์ ในการทำปริญญานิพนธ์นี้ ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ได้ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

ขอบคุณนักศึกษาห้อง 3S/1 ที่ช่วยติชมปริญญานิพนธ์ฉบับนี้และช่วยเหลือเกี่ยวกับเครื่องมือต่างๆ และเป็นกำลังใจซึ่งกันและกัน

และที่ลืมมิได้คือ ขอกราบขอบพระคุณคุณพ่อ คุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์ที่ได้จากปริญญานิพนธ์ฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญภาพ.....VII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 ขั้นตอนการศึกษา.....2

บทที่ 2 ความรู้เบื้องต้น และหลักการทำงาน.....3

2.1 กล่าวนำเกี่ยวกับสเต็ปมอเตอร์.....3

สเต็ปมอเตอร์ในปัจจุบัน.....3

การพันขดลวดบนสเตเตอร์ของสเต็ปมอเตอร์.....4

การต่อวงจรจ่ายไฟให้สเต็ปมอเตอร์.....5

การควบคุมการหมุนของสเต็ปมอเตอร์.....6

2.2 กล่าวนำเกี่ยวกับไมโครคอนโทรลเลอร์.....11

ไมโครคอนโทรลเลอร์.....11

การเลือกใช้ไมโครคอนโทรลเลอร์.....13

โครงสร้างและสถาปัตยกรรม.....14

การจัดขาไมโครคอนโทรลเลอร์.....15

โครงสร้างและการทำงานของพอร์ต.....18

การจัดการหน่วยความจำของไมโครคอนโทรลเลอร์.....23

รีจิสเตอร์ฟังก์ชั่นพิเศษ.....26

การเข้าถึงข้อมูลไมโครคอนโทรลเลอร์.....31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

สแต็กในไมโครคอนโทรลเลอร์.....32
การอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์.....35
กระบวนการรีเซตในไมโครคอนโทรลเลอร์.....40

บทที่ 3 หลักการและการออกแบบ.....49
3.1 กล่าวนำ.....49
3.2 หลักการออกแบบส่วนคัดแยกเหรียญ.....50
3.3 หลักการออกแบบชุดทอนเหรียญ.....58

บทที่ 4 สรุปผลการวิจัยและข้อเสนอแนะ.....60
4.1 สรุป.....60
4.2 ข้อเสนอแนะและแนวทางในการพัฒนา.....61

บรรณานุกรม.....62

ภาคผนวก.....63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงลำดับการกระตุ้นขดลวดแบบเวฟ..... 6
2.2 แสดงลำดับเวลาการกระตุ้นของขดลวดแบบ 2 เฟส.....7
2.3 แสดงลำดับการกระตุ้นขดลวดแบบครึ่งเฟส.....8
2.4 แสดงหลักการคำนวณหาค่าพารามิเตอร์ของสเต็ปมอเตอร์.....8
2.5 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์.....13
2.6 แสดงหน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรเลอร์ MCS-51 แบบแฟลช.....18
2.7 แสดงการเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์R0-R7.....27
2.8 แสดงค่าของข้อมูลที่เกิดขึ้นของรีจีสเตอร์ในไมโครคอนโทรลเลอร์หลังการรีเซต.....42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 แสดงถึงการควบคุมและการพันขดลวด……. ……...5

2.2 การต่อวงจรไฟฟ้าให้กับสเต็ปมอเตอร์แบบไบโพล่า……. ……...5

2.3 การต่อวงจรไฟฟ้าให้กับสเต็ปมอเตอร์แบบยูนิโพล่า……. ……...6

2.4 แสดงวงจรสร้างลำดับการกระตุ้น…….. ……..9

2.5 แสดงสเต็ปมอเตอร์แบบชนิด 6 สาย……..10

2.6 แสดงส่วนประกอบของสเต็ปมอเตอร์……..10

2.7 แสดงโครงสร้างพื้นฐานของไมโครโปรเซสเซอร์……...11

2.8 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์……12

2.9 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช……..14

2.10 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรเลอร์ MCS-51 แบบแฟลช……. ……16

2.11 แสดงการจัดขามาตรฐานของไมโครคอนโทรลเลอร์……..17

2.12 แสดงวงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช……...19

2.13 แสดงไซเกิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช……..……...22

2.14 แสดงการจัดสรรหน่วยความจำโปรแกรมของMCS-51 แบบแฟลช……...23

2.15 แสดงการเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์……..24

2.16 แสดงการเชื่อมต่อหน่วยความจำข้อมูลภายนอกMCS-51 แบบแฟลช……24

2.17 แสดงไดอะแกรมของกระบวนการรีเซตภายในไมโครคอนโทรลเลอร์ MCS-51…….. ……41

2.18 ตัวอย่างโปรแกรมภาษาแอสเซมบลีที่มีส่วนประกอบหลักครบถ้วน…….. ……43

2.19 ตัวอย่างไฟล์นามสกุล .LST หรือลิสต์ไฟล์ที่ได้จากการแอสเซมเบลอร์……..47

3.1 แสดงระยะทางและระยะขจัด……...51

3.2 แสดงค่า ΔS เข้าใกล้ΔL เมื่อเวลาเปลี่ยนแปลงไป……..53

3.3 แสดงวัตถุที่เป็นไปตามกฎข้อที่ 1……..55

3.4 แสดงวัตถุที่เป็นไปตามกฎข้อ 2……..55

3.5 แสดงการออกแบบตัวคัดแยกเหรียญ……..56

3.6 แสดงวิธีให้เหรียญล้นออกมา……..57

3.7 แสดงการวางตำแหน่งของสเต็ปมอเตอร์……58

3.8 แสดงการปัดเหรียญ 5 บาท……59

3.9 แสดงการปัดเหรียญ 1 บาท……59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

เนื่องจากในปัจจุบันความเป็นอยู่ของมนุษย์ดีขึ้น มนุษย์รู้จักการรักษาสุขภาพอนามัยเพื่อให้ปราศจากโรคภัยต่างๆ ห้องน้ำจึงมีความจำเป็นต่อสุขภาพอนามัยของมนุษย์ และในปัจจุบันนี้เทคโนโลยีเข้ามามีบทบาทต่อมนุษย์มากมาย ทำให้มีความสะดวกสบายมากยิ่งขึ้น ดังนั้นวิธีการเก็บเงินเพื่อบำรุงรักษาห้องน้ำแบบเก่าคงมีความเหมาะสมที่น้อยลง เพราะวิธีการแบบนี้ต้องอาศัยมนุษย์เป็นผู้เก็บเงิน ซึ่งการคัดหาบุคคลนั้นจำเป็นต้องจ้างมา ทำให้อาจเจอปัญหาการยักยอกเงินที่ได้จากการเก็บค่าใช้บริการและอีกเหตุผลหนึ่งที่ผู้จัดทำเจอคือ การจ่ายเงินค่าบริการซึ่งค่าบริการเขียนไว้ 2 บาท เมื่อจ่ายเหรียญ 5 บาทไป ผู้เก็บเงินจะไม่ทอนเงินแต่จะให้เป็นกระดาษทิชชูมาซึ่งผู้จัดทำไม่มีความจำเป็นที่จะต้องใช้ และสังเกตผู้ใช้บริการคนอื่นๆก็เจอแบบเดียวกัน

ดั้งนั้นทางผู้จัดทำจึงคิดวิธีที่จะให้มีการจัดเก็บเงินแบบอัตโนมัติโดยไม่ใช้คนเก็บเงินซึ่งมีการใช้ ไมโครคอนโทรลเลอร์เข้ามาควบคุมการทำงานและผู้จัดทำยังทำให้มีการทอนเงินกลับมาให้ผู้ใช้บริการอีกด้วย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้เป็นการศึกษาและออกแบบสร้างตู้เก็บเงินค่าใช้บริการในการใช้ห้องน้ำให้สามารถนำไปใช้งานจริงในชีวิตประจำวันได้ เพื่อเป็นการประหยัดเวลา ประหยัดค่าใช้จ่ายในการสรรหาบุคลากรมาทำงาน และยังสามารถแก้ปัญหาการยักยอกเงินค่าใช้บริการ ซึ่งถ้าสามารถใช้งานได้ดีก็จะมีการปรับปรุงให้มีขีดความสามารถที่สูงขึ้น

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะกล่าวถึงหลักการพื้นฐานของตัวไมโครคอนโทรลเลอร์ หลักการของเครื่องตรวจเหรียญเกี่ยวกับการทำงานเมื่อเหรียญถูกหยอดลงไปจะผ่านตัวตรวจสอบโลหะและจะถูกนำไปประมวลผลว่าจะรับหรือปฏิเสธเหรียญที่หยอดลงไป หลักการตัวคัดแยกเหรียญเป็นตัวคัดแยกเหรียญที่หยอดลงไปมีมูลค่าเท่าใดและยังใช้เป็นตัวเก็บเหรียญที่ต้องการทอน หลักการของตัวทอนเหรียญเป็นการควบคุมโดยใช้ไมโครคอนโทรลเลอร์ควบคุมโดยตัวไมโครคอนโทรลเลอร์ต้องรู้มูลค่าของเหรียญที่ถูกหยอดเข้ามา และเป็นตัวสั่งการให้มีการทอนเหรียญออกตามมูลค่าที่ถูกต้อง หลักการทำงานของ สเต็ปมอเตอร์ เป็นการกล่าวถึงหลักการของสเต็ปมอเตอร์เบื้องต้นในเรื่องของการกระตุ้นขดลวด การควบคุมความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

การทำโครงการวิจัยในปริญญานิพนธ์ฉบับนี้ มีขั้นตอนการศึกษาเริ่มจากการตรวจสอบความเป็นไปได้ที่จะทำโครงการนี้โดยการสุ่มปัญหาที่จะสามารถเกิดขึ้นได้ ถ้านำโครงงานนี้ไปใช้เมื่อรู้ปัญหาที่เกิดขึ้นแล้วก็นำปัญหานี้ไปประเมินผลและคาดการว่าจะแก้ไขได้อย่างไร จากนั้นก็เริ่มศึกษาถึง การทำงานของตัวไมโครคอนโทรลเลอร์ที่จะนำมาควบคุม การทำงานของระบบทั้งหมดเมื่อรู้ขอบเขตการทำงานของตัวไมโครคอนโทรลเลอร์แล้ว ก็สามารถกำหนดว่าจะให้ตัวไมโครคอนโทรลเลอร์ทำอะไรบ้าง จากนั้นเริ่มทำแบบจำลองของโครงงานชิ้นนี้ขึ้นมาก่อน แล้วเริ่มแบ่งส่วนออกเป็น 4 ส่วน คือส่วนควบคุม ชุดตรวจสอบเหรียญ ชุดคัดเหรียญ ชุดทอนเหรียญ จากนั้นจึงทำการทดสอบการทำงานว่าทำงานได้ตามที่คาดหมายไว้หรือไม่แล้วทำการสรุปผลการทดลอง และตรวจสอบแก้ไขปัญหาที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ความรู้เบื้องต้น และหลักการทำงาน

2.1 กล่าวนำเกี่ยวกับสเต็ปมอเตอร์

สเต็ปมอเตอร์ เป็นมอเตอร์ที่หมุนเพียงเล็กน้อยตามแรงดันไฟฟ้าที่ป้อนเป็นพัลซ์ ซึ่งต่างจากมอเตอร์ ทั่วไปที่จะหมุนทันทีและตลอดเวลาเมื่อป้อนแรงดันไฟฟ้าข้อดีของสเต็ปมอเตอร์ คือ สามารถกำหนดตำแหน่งของการหมุน(องศาหรือระยะทาง) ได้อย่างละเอียดโดย ใช้คอมพิวเตอร์ หรือ ไมโครคอนโทรลเลอร์เป็น เครื่องกำหนดและจัดเก็บตัวเลข

โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ทำมาจากแผ่นเหล็กวงแหวนที่มีซี่ยื่นออกมาประกอบกันเป็นชั้นๆ โดยที่แต่ละซี่นั้นจะมีคอยล์(ขดลวด)พันสวมอยู่ เมื่อมีการป้อนกระแสผ่านคอยล์ทำให้เกิดสนามแม่ เหล็กไฟฟ้า(Electromagnetic) ดังรูปด้านล่าง จะแสดงถึงองค์ประกอบ ในที่นี้ซึ่งถ้าเราเพิ่มจำนวนของขั้วแม่เหล็กมากขึ้นจะเพิ่มจำนวนของสเต็ปต่อวงจรรอบมากขึ้นตามด้วย

ลักษณะการนำไปใช้งาน สเต็ปมอเตอร์ ใช้งานลักษณะ Open Loop System แปลเป็นภาษาไทย ระบบเปิด คือ สเต็ปมอเตอร์สามาทำงานได้โดยไม่ต้องมีการ ป้อนค่าพารามิเตอร์กลับมา (Feed back) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งที่แน่นอน จะต้องการป้อนกลับไปยังระบบและตัวบอก ตำแหน่งว่าถูกต้องหรือผิดผลาดให้รับทราบ

ดังเช่นวิธีที่ใช้กับสเต็ปมอเตอร์ คือนำลิมิทสวิตช์ ติดตามตำแหน่งที่จะตรวจจับ เมื่อสเต็ปมอเตอร์ เริ่มหมุนแล้วหมุนไปจนถึงตำแหน่งของสวิตช์ตรวจจับสัญญาณสวิตช์ทำงาน ก็จะป้อนกลับไปสู่ระบบซึ่งก็จะทำให้รู้การทำงานของสเต็ปมอเตอร์ตลอด ตัววงจรไมโครคอนโทรลเลอร์เองจะมีจุดอ้างอิง ไว้ให้เริ่มต้นการทำงานและอ้างอิงตำแหน่งได้ถูกต้อง โดยแนวทางสเต็ปมอเตอร์เป็นอุปกรณ์จำพวกเชิงกลทางไฟฟ้า โดยมีกรุ๊ปของไบนารี่โวลท์เตทเป็นอินพุตและการเคลื่อนที่ แบบเชิงมุมเป็นเอาท์พุต หรือว่าหมุนทีละสเต็ปซึ่งอยู่ระหว่าง 0.1 - 30 องศา อยู่ที่โครงสร้างของสเต็ปมอเตอร์ โดยตามสัญญาณพัลซ์ที่จ่ายให้กับขดสเตเตอร์ ทำให้เกิดแรงผลักแก่โรเตอร์หมุนไป สเต็ปมอเตอร์มีขดลวดหลายชุดในที่นี้เราเรียกว่า Phase (เฟส) ดังนั้นสัญญาณที่ต่อเนื่องเป็น Sequence(ซีเควน) ลักษณะของBinary(ไบนารี่) ซึ่งจะต้องไปผ่านวงจร Driver(ไดรเวอร์) ก็จะทำให้โรเตอร์หมุนไปอย่างต่อเนื่อง

สเต็ปมอเตอร์ที่พบในปัจจุบันมี 3 ลักษณะดังนี้

1. แบบแม่เหล็กถาวร(PERMANENT MAGNET_PM)

สเต็ปมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆโพลโดยมีโรเตอร์ (ROTOR) เป็นรูปทรง กระบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้อนไฟกระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้กับขดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักต่อโรเตอร์ ทำให้มอเตอร์หมุนมอเตอร์แบบ PM จะเกิดแรงฉุดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

2. แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE- VR)

สเต็ปมอเตอร์แบบVR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ทำจากสารเฟอร์โรแมกเนติก กำลังอ่อน มีลักษณะเป็นฟันเลื่อย รูปทรงกระบอกโดยจะมีความสัมพันธ์โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนท์ต่ำที่สุด ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างขดกันไป ก็ทำให้มอเตอร์ หมุนไปตำแหน่งต่างๆ กันโรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

3. แบบผสม(HYBRID-H)

สเต็ปมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดีทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงฉุดยึดโรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ

การพันขดลวดบนสเตเตอร์ของสเต็ปมอเตอร์

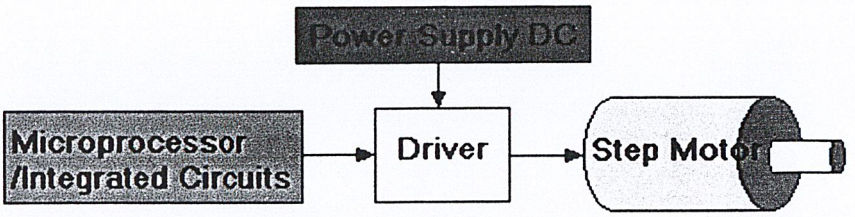
1. แบบ Bipolar

จะมีการพันขดลวดหนึ่งขด(จะกี่รอบก็แล้วแต่ สเป็กใช้งานนะครับ)ในแต่ขั้วแม่เหล็กของสเตเตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้น ที่สเตเตอร์จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้า ซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้เพียง การกลับทิศทางของการไหลในกระแสไฟฟ้า โดยมาจากการควบคุมของวงจรสวิทชิ่งให้กลับขั้วไฟฟ้า

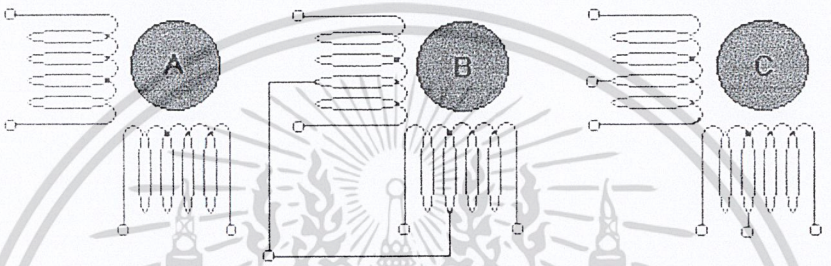
2. แบบ Unipolar

แบบนี้มี 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงกันข้าม เช่นกันครับการกลับทิศทางขั้วแม่เหล็กทำได้โดยใช้วงจรสวิทชิ่งให้สลับหนึ่งไปยังอีกขั้วหนึ่งแทนกันแค่ล่ะครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การควบคุมระบบสเต็ปมอเตอร์

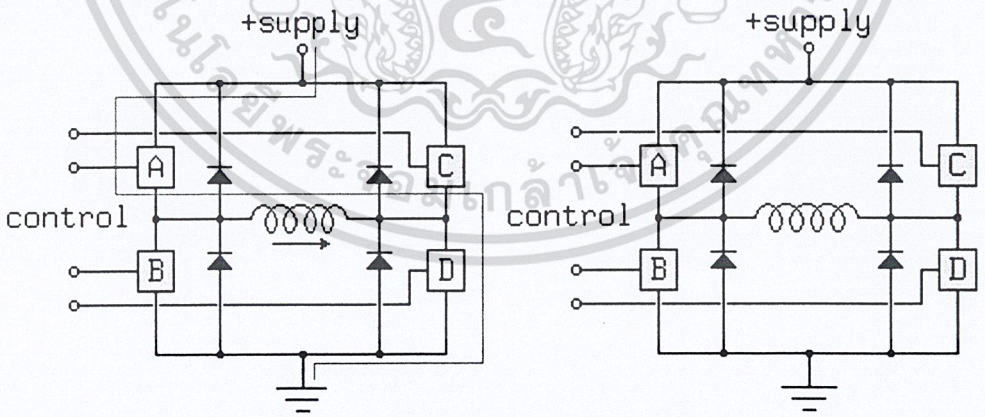


A) แบบไบโพล่าร์ B)แบบยูนิโพลาร์ 5 สาย C)แบบยูนิโพลาร์ชนิด 6 สาย

การพันขดลวดบนสเตเตอร์ของสเต็ปมอเตอร์

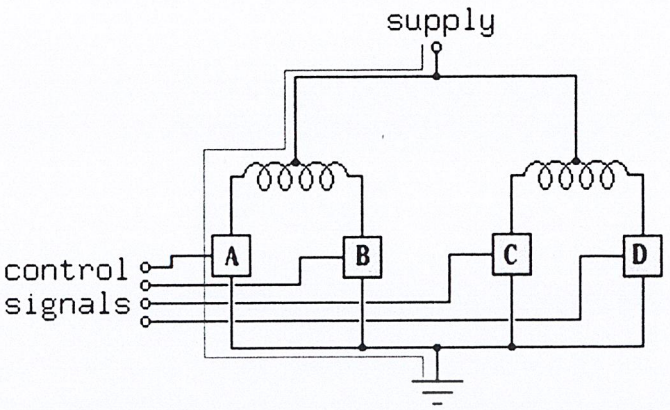
ภาพที่ 2.1 แสดงถึงการควบคุมและการพันขดลวด

การต่อวงจรการจ่ายไฟให้ สเต็ปมอเตอร์



ภาพที่ 2.2 การต่อวงจรจ่ายไฟให้กับสเต็ปมอเตอร์แบบไบโพล่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 การต่อวงจรจ่ายไฟให้กับสเต็ปมอเตอร์แบบยูนิโพล่า

การควบคุมการหมุนของสเต็ปมอเตอร์

การควบคุมและสั่งงานให้สเต็ปมอเตอร์ทำงาน ไปทีละสเต็ปสามารถทำได้โดยการจ่ายกำลังไฟไปยังขดลวด ในแต่ละขอบนสเตเตอร์ โดยการป้อนจะทำในลักษณะเป็นลำดับหรือเรียกว่าซีเควนเวียลในลูปที่ถูกต้อง ซึ่งจะแบบ ได้เป็น 3 รูปแแบ คือ แบบเวฟ(wave) แบบ 2 เฟส(2 phase) และแบบครึ่งสเต็ป (half step) ทั้ง 3 แบบนี้ก็จะมีข้อดีและข้อเสียต่างกันออกไป

1. แบบเวฟ (wave)

จะเป็นการกระตุ้นแบบที่ง่ายที่สุด ซึ่งจะทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆเรียงกันไป ตัวอย่างเช่น ขดที่ 1 , 2 , 3 , 4 ,1, 2 , 3 , 4 เป็นลำดับอย่างนี้ หรือ ขด 1 , 4 , 3 , 2 , 1 , 4 , 3 , 2 เป็นลำดับกันไป ทั้งนี้ขึ้นอยู่กับทิศทางที่เราต้องให้มอเตอร์หมุนไป วงจรที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่าและง่ายกว่า ดังในรูปของวงจรการจ่ายไฟ

ตารางที่ 2.1 แสดงลำดับการกระตุ้นขดลวดแบบเวฟ

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON			
2		ON		
3			ON	
4				ON
5	ON			
6		ON		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แบบสองเฟส (2 Phase)

แบบนี้ก็จะคล้ายกับการกระตุ้นในแบบเวฟแต่จะต่างกันตรงที่ แบบ 2 เฟส จะกระตุ้นทีละ 2 ขด ที่อยู่ใกล้กันใน เวลาเดียวกัน และจะเรียงลำดับกันไป ดังเช่นแบบเดียวกับแบบเวฟ จะยกตัวอย่างการกระตุ้นขดลวดในลักษณะ ซีเควนให้ดูดังนี้ 12,23,34.41,12,23,34,41 เรียงลำดับกันไปเรื่อยๆ หรือจะเป็น 14,43,32,21,14,43,32,21 เรียงกันไปเรื่อยๆเช่นกัน ถ้าจะมากล่าวถึงข้อดีข้อเสียของแบบ 2 เฟส แล้วมีดังนี้

ข้อดี การที่เราจะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่า แบบเวฟ ซึ่งโรเตอร์จะหมุนด้วยแรง ดึงแบบเต็มๆแรงจาก ทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน

ข้อเสีย แบบ 2 เฟส จะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบเวฟ

ตารางที่ 2.2 แสดงลำดับการกระตุ้นของขดลวดแบบ 2 เฟส

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON	ON		
2		ON	ON	
3			ON	ON
4	ON			ON
5	ON	ON		
6		ON	ON	

3. แบบครึ่งสเต็ป

แบบนี้แบบรูปแบบผสมผสานของการกระตุ้นระหว่าง แบบเวฟ กับ แบบ 2 เฟส เพื่อให้จำนวนรอบของสเต็ปให้ มากขึ้นเป็น 2 เท่า ซึ่งในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเรื่อยๆ เป็นลำดับดังจะยกตัวอย่างต่อไปนี้1,12,2,23,3,34,4,41,1,12,2,23,3,34,4,41,1 เป็นลำดับอยู่อย่างนี้ 1,41,4,43,3,32,2,21,1,41,4,43,3,32,2,21,1 เป็นลำดับ กันไปเราจะมาพูดกันถึงข้อดีและข้อเสียของการกระตุ้นแบบครึ่งสเต็ปกัน

ข้อดี การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้น เนื่องจากช่วงสเต็ปที่มีระยะสั้นลงอีกประการหนึ่งแต่ละ สเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกันเป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย

ข้อเสีย ก็คงจะเช่นเดียวกับแบบ 2 เฟสล่ะครับ ที่ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบเวฟ หรือจะใช้เท่ากับแบบ 2 เฟส นั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงดับการกระตุ้นขดลวดแบบครึ่งเฟส

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON			
2	ON	ON		
3		ON		
4		ON	ON	
5			ON	
6			ON	ON
7				ON
8	ON			ON
9	ON			
10	ON	ON		

ตารางที่ 2.4 แสดงหลักการคำนวณหาค่าพารามิเตอร์ของสเต็ปมอเตอร์

1. Tooth Pitch

Ps = Lenght of stator tooth pitch

Pr = Lenght of rotor tooth pitch

Ps = 360/Ns ; Ns =number of stator tooth.

Pr = 360/Nr ; Nr =number of rotor tooth.

2. Step Angle

fs = Pr/NP =360/NrNp ; Number of Phase.

= (1*Pr-Ps*1)x2 degree/step

3. Stepping Rate

Rs = 360/fs =NrNp step/round

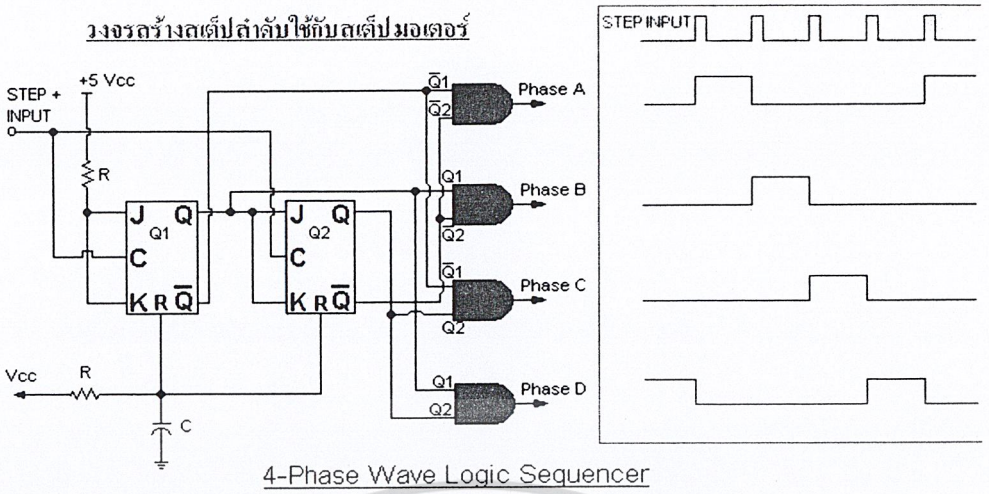
4. Speed of step motor(W)

W = 60f/Rs = 60f/NpNr =fsf/6 (rpm)

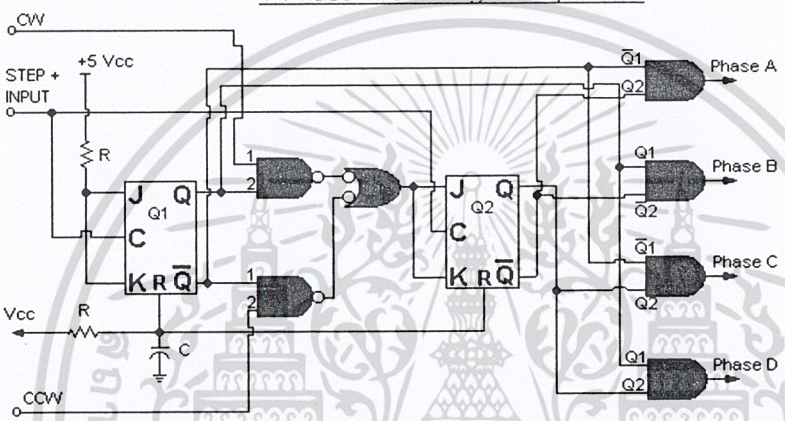
5. Number of stator poles per phase

X =Ns/Np = Rs/Np(Np+-1) =Nr/Np+-1

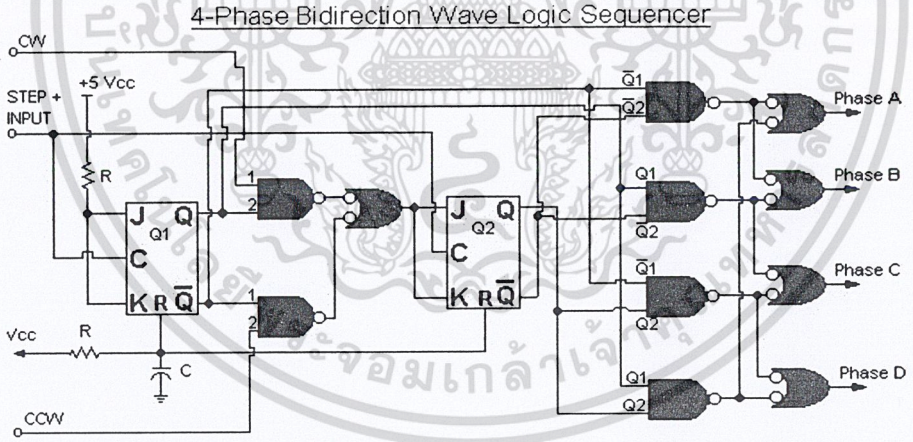
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



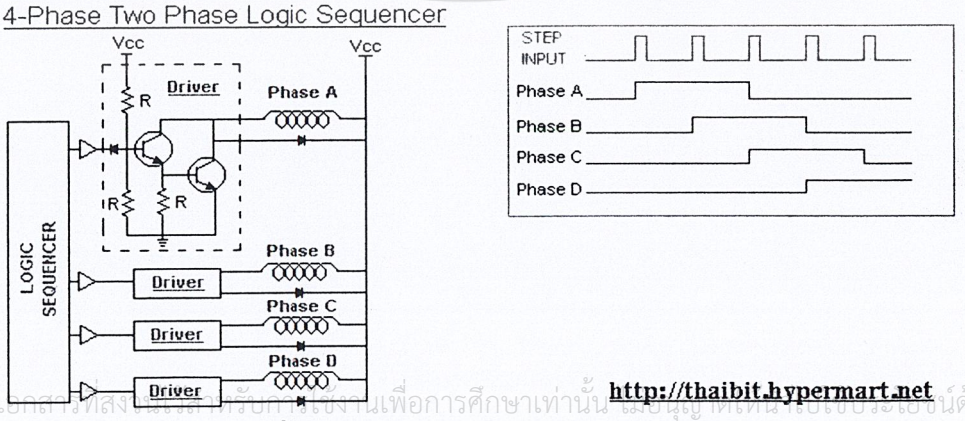
4-Phase Wave Logic Sequencer



4-Phase Bidirection Wave Logic Sequencer

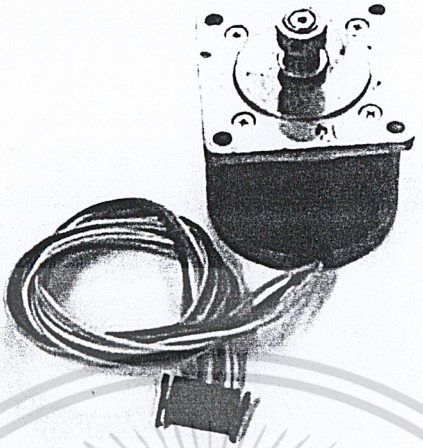


4-Phase Two Phase Logic Sequencer

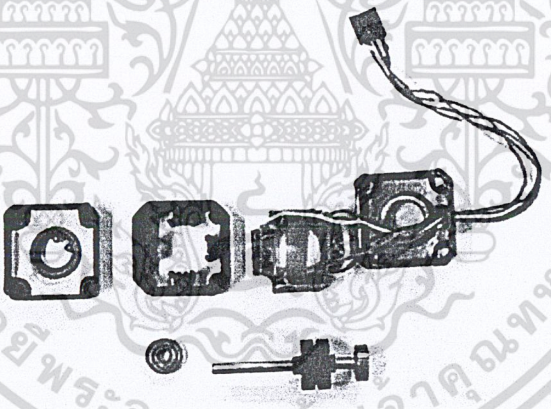


http://thaibit.hypermart.net
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.4 แสดงวงจรสร้างลำดับการกระตุ้น



ภาพที่ 2.5 แสดงสเต็ปมอเตอร์แบบชนิด 6 สาย

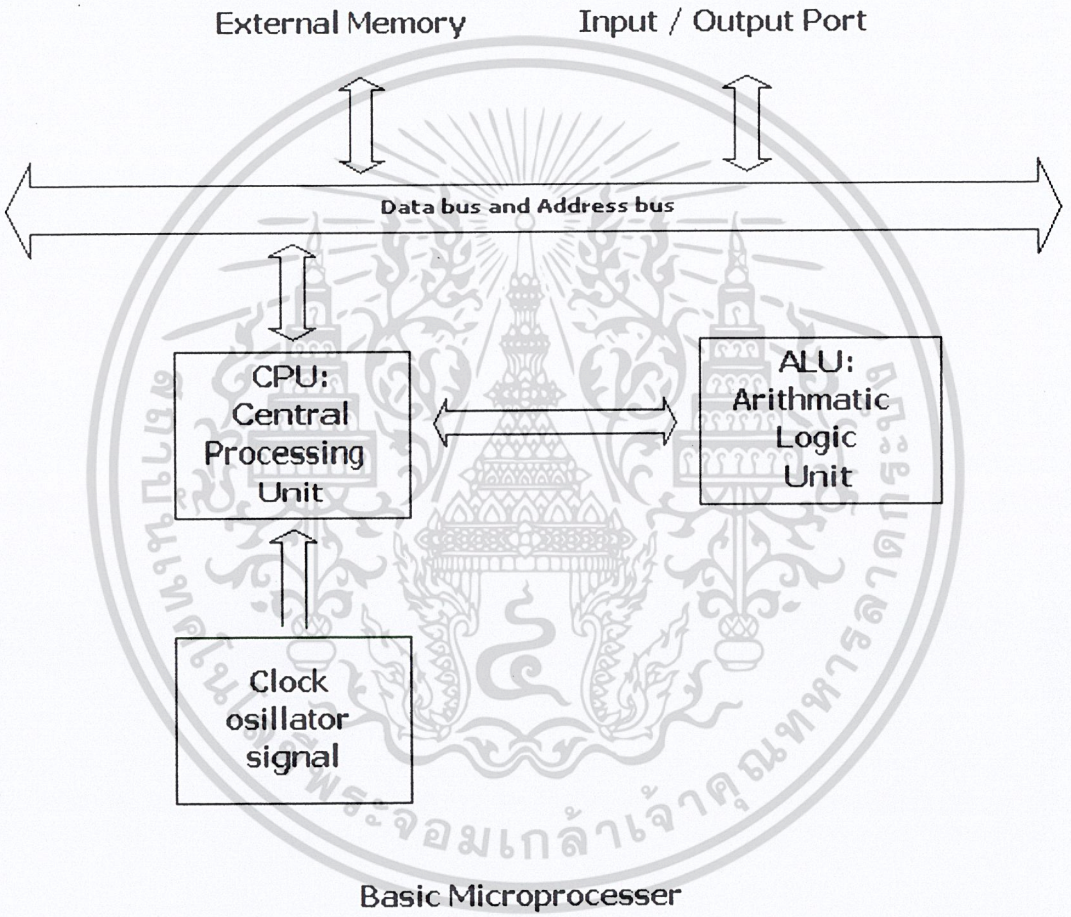


ภาพที่ 2.6 แสดงส่วนประกอบของสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 กล่าวนำเกี่ยวกับไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่บรรจุความสามารถมากมายไม่ว่าจะเป็นหน่วยประมวลผลหน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณออกทางเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกา ทำให้ไมโครคอนโทรลเลอร์สามารถที่จะนำไปประยุกต์ใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี โดยช่วยลดจำนวนของอุปกรณ์และขนาดของระบบลง ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม



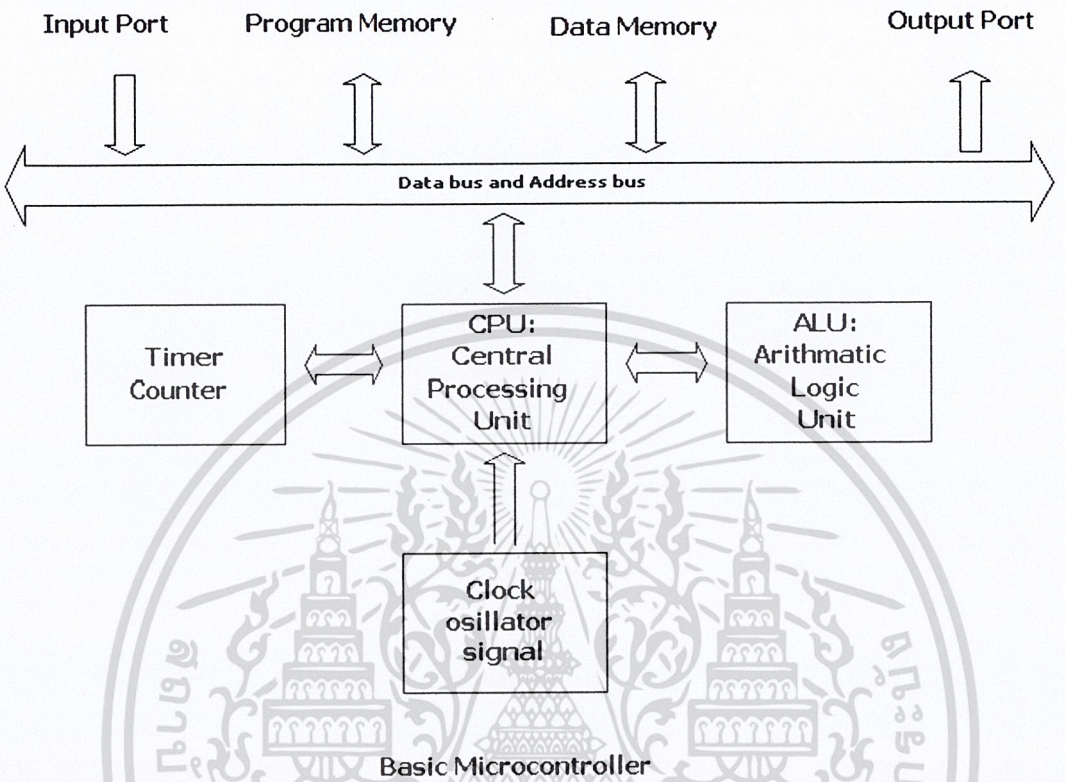
ภาพที่ 2.7 แสดงโครงสร้างพื้นฐานของไมโครโปรเซสเซอร์

ความแตกต่างของไมโครคอนโทรลเลอร์กับไมโครโปรเซสเซอร์

จากภาพที่ 2.7 จะแสดงโครงสร้างของไมโครโปรเซสเซอร์ ซึ่งมีหน่วยประมวลผลกลาง หน่วยคำนวณทางคณิตศาสตร์และลอจิก บัสข้อมูลและแอดเดรสสำหรับติดต่อกับหน่วยความจำภายนอก และวงจรกำเนิดสัญญาณนาฬิกา นั่นหมายความว่า การใช้งานไมโครโปรเซสเซอร์จะต้องเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอกและถ้าหากมีความต้องการเชื่อมต่อกับอุปกรณ์อินพุตและเอาต์พุต ต้องอาศัยอุปกรณ์ที่เรียกว่า ไอซีขยายพอร์ต (port expander) ทำให้การสร้างระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น เพื่อการนี้ต้องขออนุญาตเป็นลายลักษณ์อักษรจากจุฬาลงกรณ์มหาวิทยาลัยก่อนเท่านั้น

ควบคุมด้วยไมโครโปรเซสเซอร์จึงมีความต้องการอุปกรณ์จำนวนมาก ส่งผลให้ขนาดของระบบใหญ่พอสมควร



ภาพที่ 2.8 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

จากภาพที่ 2.8 จะแสดงโครงสร้างของไมโครคอนโทรลเลอร์ จะเห็นได้อย่างชัดเจนว่าภายในไมโครคอนโทรลเลอร์มีอุปกรณ์พื้นฐานเหมือนกับไมโครโปรเซสเซอร์ หากแต่จะบรรจุหน่วยความจำโปรแกรม หน่วยความจำข้อมูล และพอร์ตอินพุตและเอาต์พุตไว้ภายในพร้อมสรรพ ผู้ใช้งานจึงเพียงแต่เขียนโปรแกรมควบคุมลงบนตัวไมโครคอนโทรลเลอร์ แล้วต่ออุปกรณ์ที่ใช้ในการสร้างวงจรกำเนิดสัญญาณนาฬิกา อาทิ คริสตอล ตัวเก็บประจุเป็นต้น สุดท้ายเชื่อมต่ออุปกรณ์อินพุตและเอาต์พุตที่ต้องการเข้ากับขาพอร์ตของไมโครคอนโทรลเลอร์เพียงเท่านี้ก็สามารถใช้งานไมโครคอนโทรลเลอร์ได้แล้ว ส่งผลให้ขนาดและราคาของระบบลดลงอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์

คุณสมบัติ	ไมโครโปรเซสเซอร์	ไมโครคอนโทรลเลอร์
ขนาดของหน่วยประมวลผลกลาง	ไม่น้อยกว่า 8 บิต	ส่วนใหญ่จะมีขนาด 8 บิต
หน่วยคำนวณทางคณิตศาสตร์และลอจิก	มีอยู่ภายใน	มีอยู่ภายใน
วงจรกำเนิดสัญญาณนาฬิกา	มีอยู่ภายใน	มีอยู่ภายใน
การเชื่อมต่อกับหน่วยความจำโปรแกรม	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายในและภายนอก
การเชื่อมต่อกับหน่วยความจำข้อมูล	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายในและภายนอก
การเชื่อมต่อกับพอร์ตอินพุตเอาต์พุต	เชื่อมต่อภายนอกเท่านั้น	มีอยู่ภายในและสามารถขยายได้
ไทเมอร์ / เคาน์เตอร์	ไม่มีในซีพียูขนาดเล็ก	มีอย่างน้อย 1 ตัว ขนาด 8-16 บิต
วอตช์ด็อกไทเมอร์	ไม่มีในซีพียูขนาดเล็ก	มีอย่างน้อย 1 ตัว
จำนวนขาต่อใช้งาน	ไม่น้อยกว่า 40 ขา	มีตั้งแต่ 8 ขาขึ้นไป

จะเลือกใช้ไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์เมื่อใด

ถ้าเป็นการสร้างระบบควบคุมขนาด 8 บิต มีความต้องการเชื่อมต่อกับอุปกรณ์ภายนอกไม่มากนัก (น้อยกว่า 10 แบบ) ควรเลือกใช้ไมโครคอนโทรลเลอร์ หากต้องมีการประมวลผลข้อมูลจำนวนมากต้องใช้หน่วยความจำโปรแกรมสูงถึง 8 กิโลไบต์ ทั้งยังมีความต้องการเก็บรักษาข้อมูลลงหน่วยความจำข้อมูลมากเป็นกิโลไบต์ ควรออกแบบให้ไมโครคอนโทรลเลอร์ในระบบควบคุมนี้เชื่อมต่อกับหน่วยความจำภายนอก ถ้าหากต้องใช้งานกับข้อมูลมากกกว่า 8 บิต ตลอดเวลาใช้งานและต้องการความเร็วในการทำงานสูง ๆ สามารถติดต่อหน่วยความจำได้เป็นจำนวนมาก ๆ ติดต่อกับอุปกรณ์อินพุตและเอาต์พุตได้จำนวนมากมายในเวลาเดียวกัน ควรเลือกใช้ไมโครโปรเซสเซอร์ ดังจะเห็นได้จากในเครื่องคอมพิวเตอร์ที่ใช้ไมโครโปรเซสเซอร์ เป็นหัวใจหลักในการทำงาน ทำให้สามารถเชื่อมต่อกับหน่วยความจำได้มากเป็นหน่วยกิกะไบต์ ขนาดของข้อมูลสูงถึง 64 บิต ความเร็วสูงเป็นหลายร้อยเมกะเฮิรตซ์ เป็นต้น

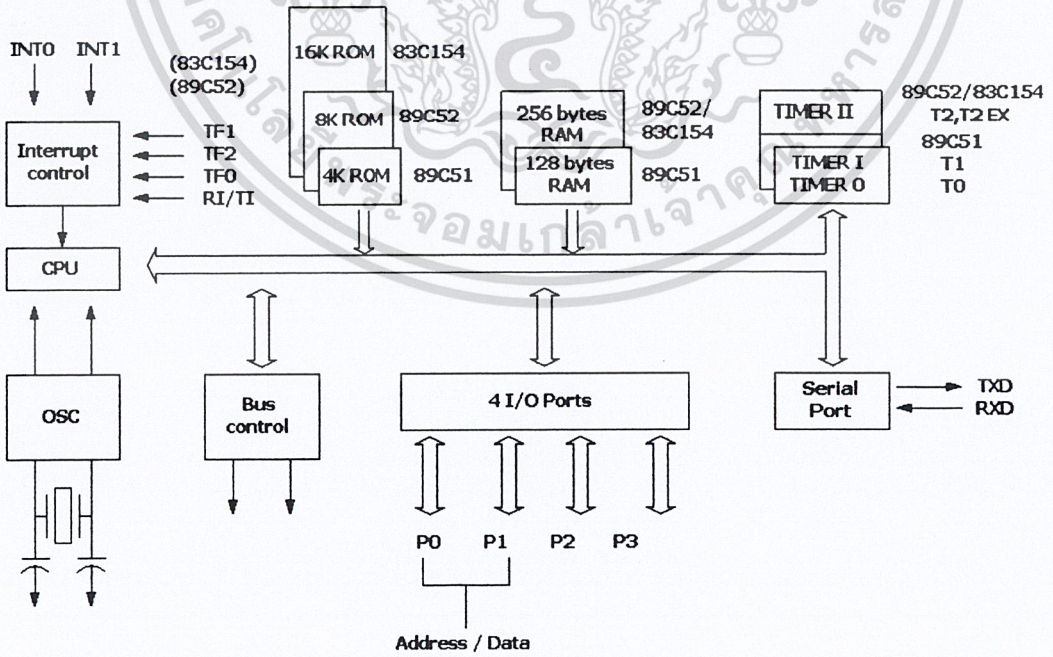
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น สรุปได้ว่า ไมโครคอนโทรลเลอร์เหมาะสำหรับการสร้างระบบควบคุม ในขณะที่ไมโครโปรเซสเซอร์เหมาะสำหรับการสร้างระบบประมวลผลข้อมูลความเร็วสูงและระบบควบคุมที่มีขนาดใหญ่

โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล MCS - 51 อนุกรม AT89xx

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์ / เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT899Sxx
- มีวอตซ์ด็อกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx



ภาพที่ 2.9 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.9 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx จะเห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐานหากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา หากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว

สำหรับโครงสร้างพื้นฐานของอนุกรม AT89Sxx จะเห็นได้ว่ามีสวนประกอบที่เพิ่มเติมแตกต่างจาก AT89Cxxx อยู่หลายส่วน อาทิเช่น วงจรเชื่อมต่อวงจรอนุกรมแบบ SPI ซึ่งจะทำให้ไมโครคอนโทรลเลอร์ อนุกรมนี้ใช้ในการเขียนข้อมูลในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิปออกไปจากระบบหรือเรียกว่าการโปรแกรมในวงจร ไทเมอร์ / เคาน์เตอร์ขนาด 16 บิต ที่เพิ่มเติมเข้ามาอีกหนึ่งตัวเป็นไทเมอร์ 2 และวงจรวอตซ์ด็อกที่ใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู

การจัดขาของไมโครคอนโทรลเลอร์ MCS – 51

ไมโครคอนโทรลเลอร์ MCS - 51 ทุกเบอร์จะมีสถาปัตยกรรม และขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปโดยมีรายละเอียดขั้นต้นดังนี้

ขา VCC ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต(P0.0 - P0.7) มี 8 ขา แต่ละขาสามารถกำหนดได้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไปถ้าหากต้องการหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอยใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0 - A7) และมีขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

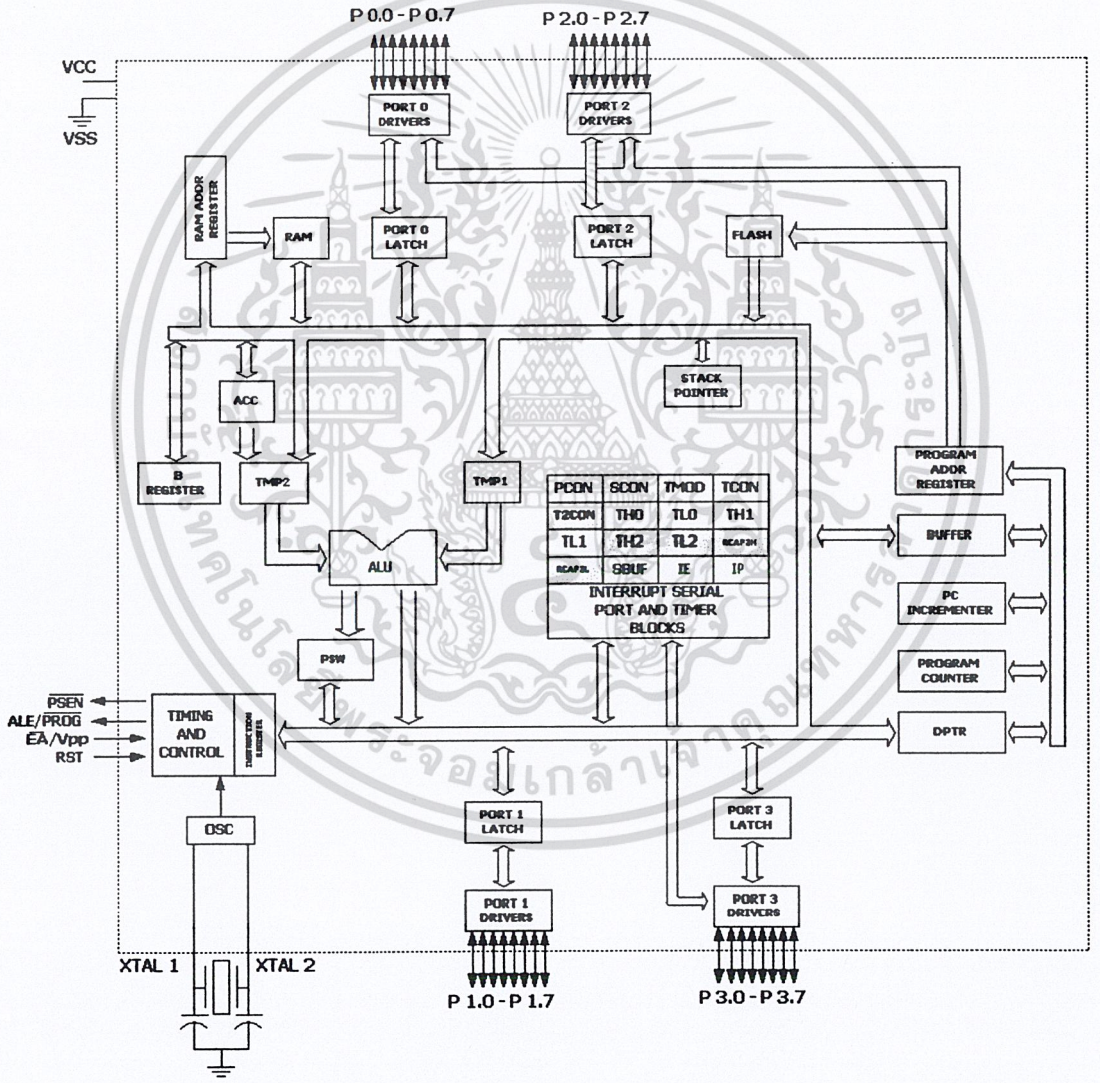
ขาพอร์ต(P1.0 - P1.7) มีขา 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำให้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะให้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริเกอร์ของไทเมอร์ 2 ใน ขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตดินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกัยขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

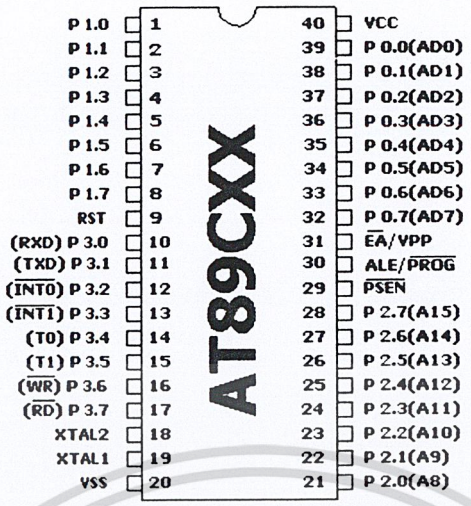
ขาพอร์ต 3 (P3.0 - P3.7) มี 8 ขา แต่ละขาสมารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังต่อไปนี้



ภาพที่ 2.10 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ภาพที่ 2.11 แสดงการจัดขามาตรฐานของไมโครคอนโทรลเลอร์

P3.0 ใช้เป็นขาอินพุตสำหรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 0 หรือขา INT0

P3.3 ใช้เป็นขาสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 0 หรือ ขาT0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซต สถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา ALE / PROG (Address Latch Enable / Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตซ์ ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS- 51 รุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์ จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง ในแต่ละแมชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใด ๆ ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50433

ขา EA / Vpp (External Access enable/ Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น 0 เป็นการเลือนให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น 1 เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ต คือ พอร์ต 0 ถึง พอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุต

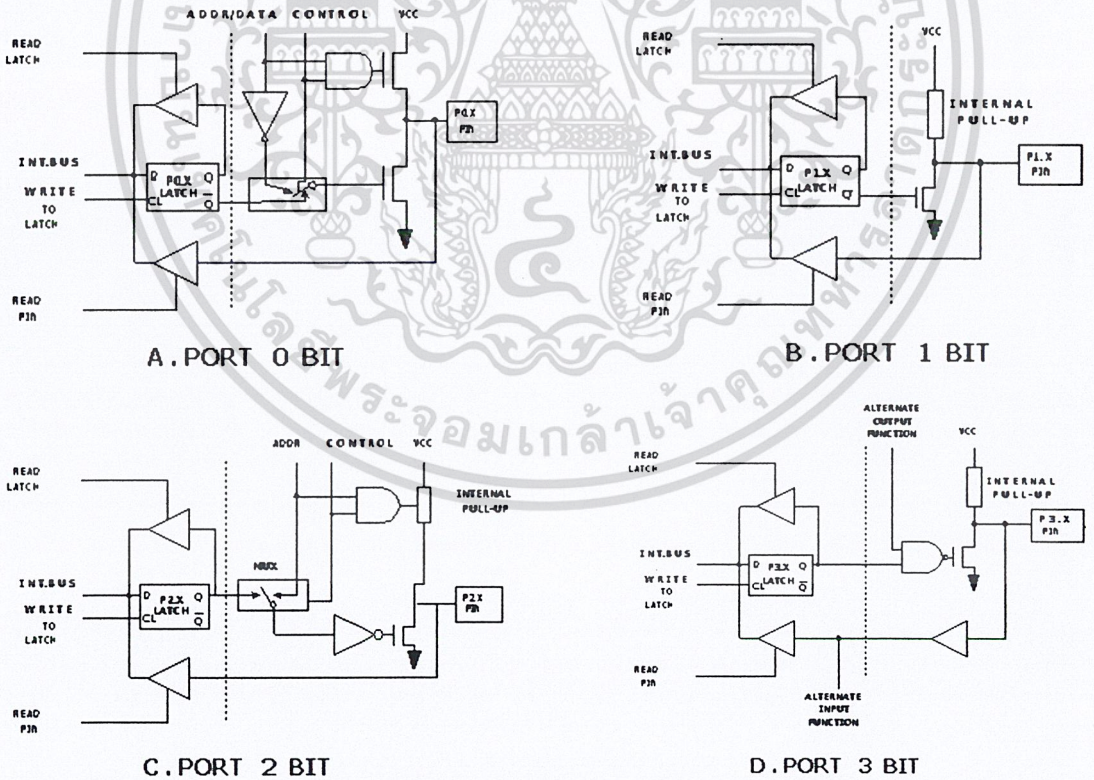
ตารางที่ 2.6 แสดงหน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรเลอร์ MCS-51 แบบแฟลช

ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.0	AT89C52/At89Sxx	ขา T2 เป็นขาอินพุตนับค่าของไทเมอร์/ เคาน์เตอร์ 2 และเป็นขาเอาต์พุตของการกำเนิดสัญญาณนาฬิกาโดยไทเมอร์ 2 (clock out)
P1.1	At89C52/At89Sxx	ขา T2EX เป็นขาอินพุตทริกเกอร์สำหรับการแคปเจอร์/ รีโหลดและควบคุมทิศทางของสัญญาณ
P1.4	AT89Sxx	ขา SS(Slave Select) เป็นขาเลือกการติดต่อในกรณีที่ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบการติกต่อแบ SPI
P1.5	AT89Sxx	ขา MOSI (Master data input,Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT89Sxx	ขา MISO (Master data input, Slave data output) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไปและใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางขานอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ใด ดังในตาราง

จากรูปแสดงวงจรภายในของพอร์ตของไทโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยในรูป (A) เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตคือวงจรฟลิปฟลอปการอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระต่อกันด้วยสัญญาณที่แยกจากกัน นั่นคือ สัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLLK ของดีฟลิปฟลอปในขณะที่ข้อมูลจะส่งผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิฟลอปที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่า ต้องการใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตปกติหรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย



ภาพที่ 2.12 แสดงวงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูป (B) เป็นวงจรภายในพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์เนื่องจากพอร์นี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน สำหรับรายละเอียดของวงจรพูลอัปแสดงในรูป

ในรูป (C) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรพูลอัปเพิ่มเติมเข้ามา ส่วนในรูป (D) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์ และวงจรอินพุต เอาต์พุต เมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้น จึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้น ๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้อุปกรณ์ภายนอกที่เชื่อมต่อพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS - 51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0" (ซึ่งในปัจจุบันอุปกรณ์อินพุตเชื่อมต่อไมโครคอนโทรลเลอร์แบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล "0" ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล "0" ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับเฟต ทำให้เฟตทำงานที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก "0" ขึ้นในทางตรงข้ามหากต้องการส่งข้อมูล "1" ออกไป ก็ให้เขียนข้อมูล "1" ไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเป็นพอร์ตเอาต์พุต แต่ละขา (หรือแต่ละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (soutce current) ได้สูงสุด 10 mA และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26mA สำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต 0 และ 15 mA สำหรับพอร์ต 1-3 ในกรณีที่ใช้งานทุกพอร์ตเอาต์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัยหาเกี่ยวกับความสามารถในการจ่ายกระแสจึงควรต่อวงจรบัฟเฟอร์ทางเอาต์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

การอ่านค่าลอจิกจากพอร์ต

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะ คือ อ่านจากขาพอร์ตโดยตรง และอ่านวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ต่อกับขาเบสทรานซิสเตอร์ชนิด NPN และขาอิมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล "1" ไปยังทรานซิสเตอร์ จะทำให้ทรานซิสเตอร์ทำงานสถานะลอจิกที่ขาพอร์ตจะเป็น "0" เนื่องจากเมื่อทรานซิสเตอร์ทำงานจะเสมือนว่าขาพอร์ตนั้นถูกต่อลงกราวด์ทำให้หากอ่านค่าลอจิกที่ขาพอร์ตจะได้ผลตรงข้ามกับที่วิ่งออกมา แต่ถ้าหากทำงานอ่านค่าลอจิกที่วงจรแลตช์จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริง ดังนั้น ในการค่าลอจิกพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

จังหวะการทำงานของไมโครคอนโทรเลอร์ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียูและลำดับขั้นตอนการประมวลผลคำสั่ง ในการประมวลผลคำสั่งของซีพียะมีขั้นตอนหลัก ๆ 2 ขั้นตอน คือ

กระบวนการเฟตช์ (fetch) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั้งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผล ขั้นต่อมาคือ

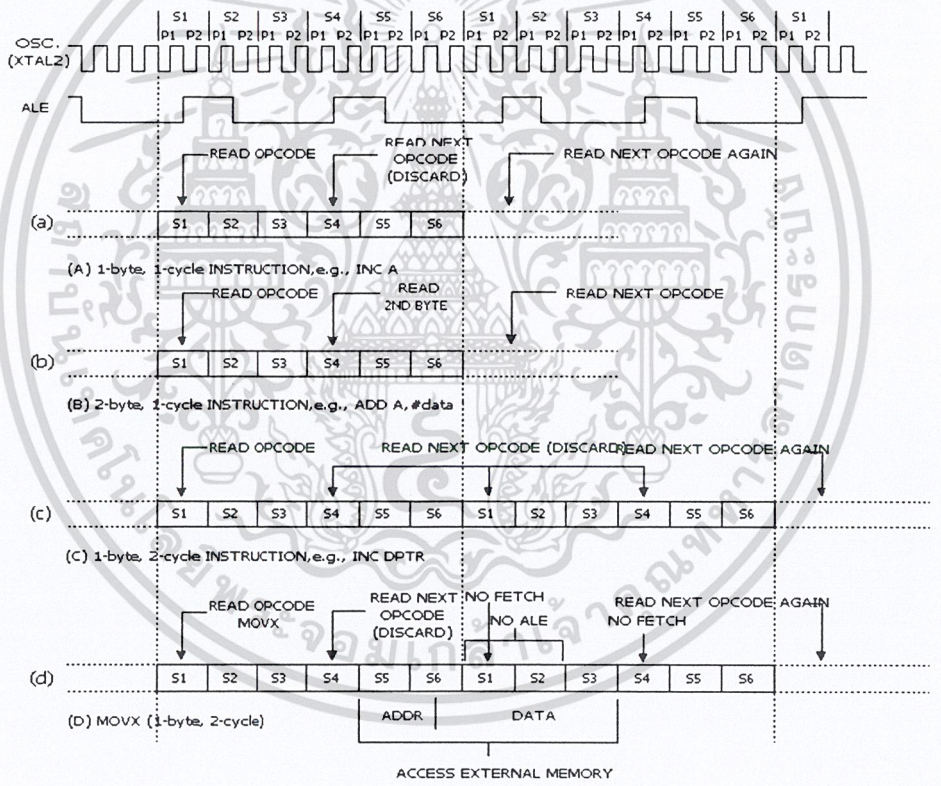
กระบวนการเอ็กซิคิวต์ (execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามเฟตช์ขึ้นมา โดยกระบวนการก่อนหน้านี้

เมื่อทำการเอ็กซิคิวต์คำสั่งเรียบร้อยแล้ว ก็จะไปเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์ MCS-51 แบบเฟลช จะเกิดการรีเซตในลักษณะที่เรียกว่า **เพาเวอร์ออนรีเซต (power - on reset)** ทำให้ซีพียูเริ่มต้นการทำงานที่แอดเดรส 0000H ของหน่วยความจำโปรแกรม จังหวะการทำงานของซีพียูจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจากการทำงานหรือแมชีนไซเกิล (machinecycle) ในรูป เป็นไดอะแกรมเวลาแสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 โดยใน 1 รอบการทำงานหรือแมชีนไซเกิลจะแบ่งย่อยออกเป็น 6 สเตต (state) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตตมีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกา ถ้าสัญญาณนาฬิกามีความถี่ 12MHz จะมีคาบเวลาเท่ากับ 1 ms คาบเวลาทั้งสองภายในหนึ่งสเตตจะเรียกว่า เฟส 1(phase 1) และเฟส (phase 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูป (ก) และ (ข) จะเป็นการเอ็กซิคิวต์คำสั่งที่ใช้เวลา 1 ไซเกิล เริ่มต้นที่สเตต 1 จะเป็นการอ่านค่าออปโค้ด อันเป็นกระบวนการแลตช์ค่าของออปโค้ดส่งไปให้รีจิสเตอร์คำสั่ง (Instruction Register : IR) การเฟตช์ครั้งที่สองจะเกิดขึ้นที่สเตต 4 ภายในแมชีนไซเกิลเดียวกัน ในกรณีที่เป็นคำสั่งไบต์เดียว การเฟตช์ครั้งที่ 2 ภายในแมชีนไซเกิลเดียวกันถูกตัดทิ้งไป ในคำสั่งที่มีใช้เวลา 1 ไซเกิล จะสิ้นสุดการทำงานลงในสเตต 6 ของแมชีนไซเกิลเดียวกัน

ในกรณีที่คำสั่งใช้เวลา 2 ไซเกิล การทำงานของคำสั่งนั้นจะสิ้นสุดลงสเตต 6 ของแมชีนไซเกิลที่สองดังในไดอะแกรมรูป (ค) สำหรับในการกระทำคำสั่ง MOVX ซึ่งเป็นคำสั่งขนาด 1 ไบต์ 2 ไซเกิล จะไม่มีการเฟตช์เกิดขึ้นในไซเกิล ที่สองของคำสั่ง MOVX นี้เนื่องจากซีพียูจะไปทำการติดต่อกับหน่วยความจำภายนอกดังแสดงในไดอะแกรมรูป (ง) จะเห็นได้ว่าเวลาในการเอ็กซิคิวต์จะไม่ได้ขึ้นอยู่กับว่าทำการติดต่อหน่วยความจำโปรแกรมภายในหรือภายนอก



ภาพที่ 2.13 แสดงไซเกิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

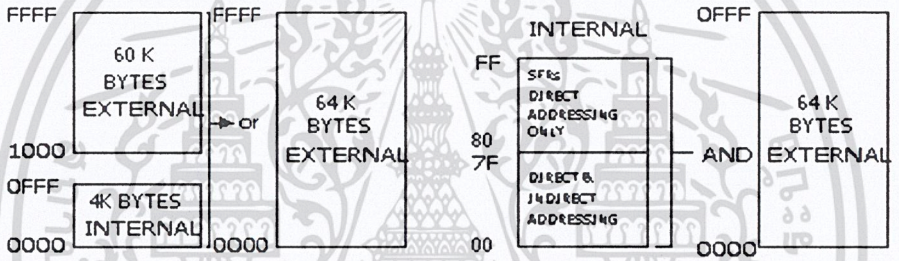
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

หน่วยความจำโปรแกรม (Program memory)

ในรูปแสดงการจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ในเบอร์ต่างๆ ที่นิยมใช้งาน อันประกอบด้วยเบอร์ AT89C51 และ AT89C52 จะเห็นได้ว่าทั้งสองเบอร์สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียว หรือรวมกับภายนอกหรือเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ ดังรูป (ก) โดยภายใน AT89C51 จะมีหน่วยความจำโปรแกรมภายใน 4 กิโลไบต์ ในขณะที่ AT89C52 จะมีขนาด 8 กิโลไบต์

ในกรณีที่ใช้หน่วยความจำภายในและภายนอกรวมกัน หากใช้ AT89C51 ก็จะสามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์ และถ้าใช้เบอร์ AT89C52 จะสามารถติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ 56 กิโลไบต์



ภาพที่ 2.14 แสดงการจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

หน่วยความจำโปรแกรมใช้เก็บข้อมูลของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์หรือที่เรียกว่า**โปรแกรมมอนิเตอร์ (monitor program)** หากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิดอีพรอม (EPROM: Erasable Programmable Read - only Memory) ซึ่งสามารถกระทำการอ่านได้เพียงอย่างเดียว

หน่วยความจำโปรแกรมมีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซตให้เริ่มต้นการทำงาน จะต้องมาเริ่มต้นที่แอดเดรส 0000H นี้เสมอ อย่างไรก็ตาม ในพื้นที่ของหน่วยความจำโปรแกรมไม่ว่าจะใช้งานภายในหรือภายนอกก็ตาม ต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H

พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH

พื้นที่สำหรับบริการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H

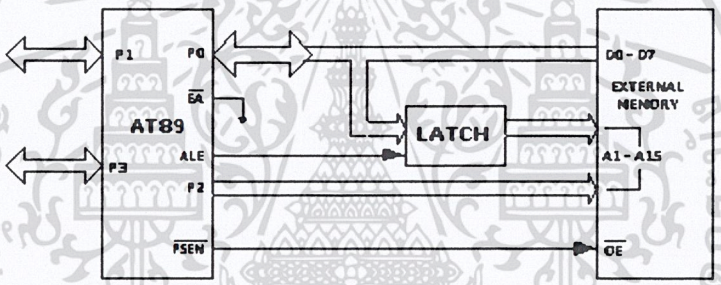
พื้นที่สำหรับการบริการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

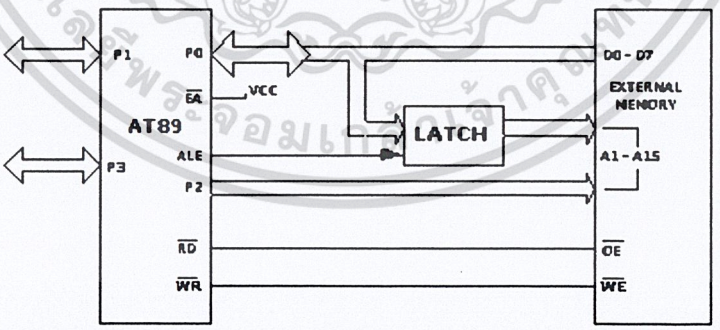
พื้นที่สำหรับบริการอินเตอร์รัปต์การสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 0023H
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH

กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ที่มีหน่วยความจำโปรแกรมภายในแต่ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกด้วย สามารถทำได้โดยต้องกำหนดแอดเดรสของหน่วยความจำโปรแกรมให้ ต่อจากแอดเดรสสุดท้ายของหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์

การต่อหน่วยความจำภายนอกแสดงดังรูปจะเห็นได้ว่าขาพอร์ต P0.0-P0.7 ถูกใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำ โดยผ่านวงจรแลตช์ ซึ่งปกติใช้ไอซีเบอร์ 74HC573 และใช้สัญญาณ ALE และ PSEN ในการเลือกว่า ต้องการใช้งานขา P0.0 - P0.7 เพื่อเป็นขาข้อมูลหรือขาแอดเดรสในขณะที่ขา P2.0-P2.7 ใช้ในการเชื่อมต่อกับขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์จะเหลือขาพอร์ตใช้งานเพียง 16 บิต คือที่ขาพอร์ต P1.0-P1.7 และ P3.0 - P3.7



ภาพที่ 2.15 แสดงการเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์



ภาพที่ 2.16 แสดงการเชื่อมต่อหน่วยความจำข้อมูลภายนอกไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำข้อมูล (Data memory)

มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายใน โดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ในอนุกรม AT89 สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อกับหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชแสดงในรูป จะเห็นได้ว่ามีลักษณะคล้ายกับการติดต่อหน่วยความโปรแกรมภายนอกแตกต่างกันที่มีสัญญาณที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ RD และ WR

สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ในอนุกรม AT89 ทุกเบอร์จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM : Randon Access Memory) โดยแต่ละเบอร์จะมีขนาดแตกต่างกันไป ในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ ในขณะที่เบอร์ AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วนคือ หน่วยความจำข้อมูลส่วนล่าง (lower) , ส่วนบน (upper) และรีจิสเตอร์ฟังก์พิเศษ (SFR: Special Function R egister) แต่ละส่วนมีขนาด 128 ไบต์ การจัดสรรหน่วยความจำข้อมูลส่วนล่าง หน่วยความจำ 32 ไบต์ ต่ำสุดที่แอดเดรส 00H-1FH แบ่งเป็น 4 กลุ่ม เรียกว่า 4 แบงก์ (bank) แต่ละแบงก์มีรีจิสเตอร์ 8 ตัวคือ R0-R7 การติดต่อกับหน่วยความจำในแบงก์ใดให้กำหนดที่รีจิสเตอร์ PSW (Program Status Word register)

จะเห็นได้ว่า หน่วยความจำข้อมูลส่วนบน และรีจิสเตอร์ฟังก์ชั่นพิเศษมีตำแหน่งทับซ้อนกัน แต่จะใช้การติดต่อที่แตกต่างกัน และในไมโครคอนโทรลเลอร์ MCS-51 บางเบอร์จะไม่มีหน่วยความจำข้อมูลส่วนบน

ขนาดของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยแท้จริงแล้วมีเพียง 256 ไบต์ แต่ด้วยการจัดการเข้าถึงที่แตกต่างกัน จึงดูเสมือนว่า ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H -7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อม สำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์ เช่นกัน มีแอดเดรสที่ 80H-FFH สามารถเข้าถึงแบบโดยอ้อมเท่านั้น ในขณะที่ริจิสเตอร์ SFR มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกับหน่วยความจำข้อมูลส่วนบน แต่สำหรับรีจิสเตอร์ SFR ใช้การเข้าถึงแบบโดยตรง

ดังนั้นเพื่อความสะดวกและง่าย ตลอดจนป้องกันความสับสนในการเขียนโปรแกรมสำหรับผู้เริ่มต้นจึงควรใช้หน่วยความจำข้อมูลภายในเพียง 128 ไบต์ จากหน่วยความจำข้อมูลส่วนล่างร่วมกับรีจิสเตอร์ SFR

หน่วยความจำข้อมูล 16 ไบต์ ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไป สามารถเข้าถึงได้ในระดับบิต (Bitaddressable) และหน่วยความจำข้อมูลที่เหลือ 80 ไบต์ จะต้องแบ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนหนึ่งสำรองไว้เป็นพื้นที่ของสแต็ก (stack : ที่พักข้อมูลชั่วคราวในกรณีที่ซีพียูมีการกระโดดไปทำงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำในส่วนนี้ต้องใช้การเข้าถึงในระดับไบต์

ในภาพแสดงโครงสร้างของหน่วยความจำข้อมูลส่วนบน ซึ่งจะมีลักษณะที่คล้ายกับหน่วยความจำข้อมูลส่วนล่าง หากแต่ใน 80 ไบต์ บนไม่จำเป็นต้องสำรองไว้สำหรับสแต็ก และต้องใช้การเข้าถึงในลักษณะโดยอ้อมเท่านั้น

รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register : SFR)

เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ทั้งหมดมีด้วยกัน 22 ตัว สำหรับในการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช AT89C51 และ 28 ตัวในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และอนุกรม AT89Sxx ทั้งนี้เนื่องจากใน AT89C52 และ AT89Sxx มีจำนวนไทเมอร์เคาน์เตอร์มากกว่า AT89C51

รีจิสเตอร์ SFR มีแอดเดรสอยู่ระหว่าง 80H-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบน สามารถเข้าถึงได้โดยตรง (direct addressing) ในรูปที่ 3-7 แสดงการจัดสรรพื้นที่ของรีจิสเตอร์ SFR แต่ละตัวในหน่วยความจำข้อมูลส่วนบน สำหรับรายละเอียดเบื้องต้นของรีจิสเตอร์ SFR มีดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
CY	AC	F0	RS1	RS0	OV	-	P

CY : แฟลกทด (Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์ และลอจิก แล้วค่าของแอกคิวมูลเลเตอร์เกิน 255 (ฐานสิบ) หรือ FFH

AC : แฟลกทดเสริม (Auxiliary Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์แล้วทำให้เกิดการทดข้ามจากบิต 3 มายังบิต 4 มักใช้บริการแปลงค่าเป็นเลขฐานสิบ (BCD operation)

F0 : แฟลกใช้งานทั่วไปเมื่อผู้เขียนโปรแกรมกำหนดค่าที่บิตนี้แล้ว ไม่ว่าจะกระทำคำสั่งใด ๆ ที่บิตนี้จะไม่มีการเปลี่ยนแปลง

RS1 : บิตเลือกรีจิสเตอร์แบงก์ (Register Select1) ใช้งานร่วมกับบิต RS0 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R77

RS0 : บิตเลือกรีจิสเตอร์แบงก์ (Register Select0) ใช้งานร่วมกับบิต RS1 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OV : บิตเกิน (Overflow) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว ทำให้เกิดการทดข้ามจากบิต 6 มายังบิต 7 ของแอกคิวมูลเตอร์ หรือแอกคิวมูเลเตอร์มีค่าเกิน127 (ฐานสิบ) นอกจากนั้นยังใช้เป็นการแสดงค่าลบด้วย

- : บิตนี้ผู้ใช้งานสามารถกำหนดใช้งานได้อย่างอิสระ

P : บิตพาริตี้ (Parity) ใช้ในการตรวจสอบจำนวนค่า "1" ภายในแอกคิวมูเลเตอร์ ถ้าหากในแอกคิวมูเลแตอร์มีจำนวนบิตที่เป็น "1" รวมกันเป็นเลขคู่ บิตนี้จะเป็น "0" ถ้ารวมเป็นเลขคี่ บิตนี้จะเป็น "1"

ตารางที่ 2.7 แสดงการเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์ R0-R7

RS1	RS0	แบงก์ของรีจิสเตอร์	ช่วงแอสเดรส
0	0	แบงก์ 0	00H-07H
0	1	แบงก์ 1	08H-0FH
1	0	แบงก์ 2	10H-17H
1	1	แบงก์ 3	18H-1FH

รีจิสเตอร์แสดงสถานะของโปรแกรม (Program Status Word : PSW)

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต นั่นหมายความว่า สามารถกระทำคำสั่งหรือกำหนดค่าในแต่ละบิต ของรีจิสเตอร์ตัวนี้ได้อย่างอิสระ มีแอดเดรสอยู่ที่ DOH เป็นรีจิสเตอร์ที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้น จะเรียกสถานะต่าง ๆ ของโปรแกรมว่า แฟลก (flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนแปลงสถานะขึ้น ผลของการเปลี่ยนแปลงนั้นจะปรากฏที่บิตต่าง ๆ ของรีจิสเตอร์ PSW รายละเอียดของแต่ละบิตในรีจิสเตอร์ PSW แสดงในรูป

จะเห็นได้ว่า นอกจากรีจิสเตอร์ PSW ถูกใช้ในการเก็บสถานะของโปรแกรมแล้ว ที่บิต RSO และ RSI ยังใช้ในการเลือกแบงก์ของหน่วยความจำส่วนล่าง ซึ่งเป็นพื้นที่ของรีจิสเตอร์ R0-R7 ด้วยดังมีรายละเอียดแสดงในตาราง โดยปกติแล้วในการใช้งานรีจิสเตอร์ R0-R7 มักนิยมเลือกใช้แบงก์ 0 เป็นลำดับแรก หากไม่เพียงพอจึงเลือกในแบงก์อื่น ๆ มาใช้ แต่ต้องระมัดระวังในการกำหนดค่าและลำดับการติดต่อให้ดี มิเช่นนั้น อาจทำให้การเขียนโปรแกรมเกิดความสับสน ดังนั้นสำหรับผู้เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ MCS-51 จึงควรเลือกใช้รีสเตอร์ R0-R7 ในแบงก์ 0 เพียงแบงก์เดียวให้ชำนาญเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น เพื่อการค้า

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

แอกคิวมูเลเตอร์ (Accumulator : ACC)

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งในการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะส่งข้อมูลหรือผลลัพธ์ ที่ได้นั้น ให้แก่ซีพียูเพื่อทำการประมวลผลต่อไป อาจเรียกรีจิสเตอร์แอกคิวมูเลเตอร์อย่างสั้น ๆ ว่า รีจิสเตอร์ A หรือ ACC

รีจิสเตอร์ A นี้สามารถเข้าถึงระดับบิตได้ นั่นหมายความว่า สามารถกระทำคำสั่งหรือกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตังนี้ได้อย่างอิสระ

รีจิสเตอร์ B

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ F00H มีหน้าที่พิเศษคือ หากมีความต้องการคูณหรือหารทางคณิตศาสตร์ จะต้องนำข้อมูลที่ต้องการหารหรือคูณนั้น มาเก็บไว้ในรีจิสเตอร์ B นี้ แล้วจึงกระทำคำสั่งการคูณหรือหารกับค่าในรีจิสเตอร์ A ต่อไป

ในกรณีที่ไม่ได้มีความต้องการคูณหรือการข้อมูล สามารถใช้รีจิสเตอร์ B นี้ในการเก็บข้อมูล ทั่วไปได้ เหมือนกับรีจิสเตอร์ปกติ และสามารถเข้าถึงในระดับบิตได้เช่นเดียวกับรีจิสเตอร์ A

โปรแกรมเคาน์เตอร์ ((Program Counter : PC)

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำ แหน่งถัดไปที่พีซียูจะต้องไปทำงาน รีจิสเตอร์ PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่จัดสรรไว้รวมกับรีจิสเตอร์ SFR ตัวอื่น ๆ การเปลี่ยนแปลงค่าของรีจิสเตอร์ PC จะขึ้นอยู่กับผลของการกระทำคำสั่งและคำสั่งภายในหน่วยความจำโปรแกรมที่ผู้เขียนโปรแกรมกำหนด

รีจิสเตอร์ PC มีความสำคัญมาก โดยเฉพาะอย่างยิ่งในการตรวจสอบการทำงานของโปรแกรมว่า คำเนินไปตามลำดับขั้นตอนที่กำหนดไว้หรือไม่

สแต็กพอยน์เตอร์ (Stack Pointer : SP)

หรือรีจิสเตอร์ตัวชี้สแต็ก มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 81H ใช้ในการเก็นค่าตำแหน่งของตัวชี้สแต็ก ซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระโดดไปทำงานที่โปรแกรมย่อย หรือกระโดดโปรแกรมย่อมกลับมายังโปรแกรมหลัก เมื่อมีการรีเซตเกิดขึ้น (รีเซต : การกระทำที่ส่งผลให้พีซียูต้องเริ่มต้นการทำงานใหม่ตั้งแต่ต้น) ค่าของรีจิสเตอร์ SP จะเท่ากับ 07H นั่นหมายความว่าตัวชี้สแต็ก มีค่า 07H แอดเดรสแรกของพื้นที่สำรองไว้ทำหน้าที่เป็นสแต็กจะเท่ากับ 08H

รีจิสเตอร์ชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data Pointer : DPTR)

มีขนาด 16 บิต โดยแบ่งเป็นรีจิสเตอร์ชี้ข้อมูลไบต์สูง (DPH) และรีจิสเตอร์ชี้ข้อมูลไบต์ต่ำ (DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H สำหรับ DPL และ 83H สำหรับ DPH รีจิสเตอร์

เอกสารนี้เป็นกรรมสิทธิ์ของผู้เขียน ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DPTR นี้ใช้ในการเก็บค่าแอดเดรสของหน่วยความจำหรืออุปกรณ์ภายนอกที่ไมโครคอนโทรลเลอร์ต้องการติดต่อด้วย

รีจิสเตอร์พอร์ต (Port register)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 มีด้วยกันทั้งสิ้น 4 ตัวคือ รีจิสเตอร์พอร์ต 0 หรือ P0 มีแอดเดรสอยู่ที่ 80H รีจิสเตอร์พอร์ต 1 หรือ P1 , มีแอดเดรสอยู่ที่ 90H รีจิสเตอร์พอร์ต 2 หรือ P2 มีแอดเดรสอยู่ที่ A0H และรีจิสเตอร์พอร์ต 3 หรือ P3 มีแอดเดรสอยู่ที่ B0H รีจิสเตอร์ทุกตัวสามารถเข้าถึงได้ในระดับบิต เมื่อต้องการอ่านหรือเขียนข้อมูลออกไปยังพอร์ตของไมโครคอนโทรลเลอร์ จะต้องกระทำผ่านรีจิสเตอร์นี้ทุกครั้ง

รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer : SBUF)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 99H ใช้ในการเก็บข้อมูลที่ทำการส่งออกหรือรับของวงจรสื่อสารอนุกรมที่มีอยู่ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยภายในรีจิสเตอร์ SBUF นี้จะแบ่งออกเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลจะถูกส่งผ่านไปยังทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปนังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51

รีจิสเตอร์ไทเมอร์ (Timer register)

เป็นรีจิสเตอร์ขนาด 16 บิต แต่จะจัดแย่งเป็นไบต์สูงและไบต์ต่ำเช่นเดียวกับรีจิสเตอร์ DPTR รีจิสเตอร์ไทเมอร์ใช้ในการเก็บค่าของตัวนับหรือเคาน์เตอร์ (counter) ภายในไมโครคอนโทรลเลอร์เพื่อใช้ในการสร้างฐานเวลา จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายใน บางทีเรียกรีจิสเตอร์ตัวนี้ว่า รีจิสเตอร์ไทเมอร์ / เคาน์เตอร์ ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C51 จะมีรีจิสเตอร์ไทเมอร์ / เคาน์เตอร์ 2 ตัว แบ่งเป็น T0 หรือ Timer 0 และ T1 หรือ Timer 1

ในรีจิสเตอร์ยังแบ่งเป็นรีจิสเตอร์ไทเมอร์ไบต์ต่ำ (TL) และรีจิสเตอร์ไทเมอร์ไบต์สูง (TH) เหมือนกัน โดนรีจิสเตอร์ TL0 มีแอดเดรสอยู่ที่ 8AH รีจิสเตอร์ TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH ตามลำดับ

สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89Sxx จะมีรีจิสเตอร์ไทเมอร์ / เคาน์เตอร์ถึง 3 ตัว โดยมีรีจิสเตอร์ TL2 และ TH2 ซึ่งมีแอดเดรสอยู่ที่ CCH และ CDH ตามลำดับเพิ่มเติมเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์แคปเจอร์ (Caputre register)

เป็นรีจิสเตอร์ขนาด 16 บิต มีเฉพาะในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้น เนื่องจากต้องใช้ร่วมกันไทเมอร์ / เคาน์เตอร์ 2 (Timer2) ซึ่งมีอยู่ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89Sxx โดยรีจิสเตอร์แคปเจอร์นี้มีชื่อเรียกอย่างย่อว่า รีจิสเตอร์ RCAP2 ซึ่งแบ่งออกเป็นไบต์ต่ำ คือ RCAP2L มีแอดเดรสอยู่ที่ CAH และไบต์สูงคือ RCAP2H มีแอดเดรสอยู่ที่ CBH

รีจิสเตอร์แคปเจอร์จะถูกใช้งานเมื่อกำหนดให้ไทเมอร์ 2 ทำงานในโหมดแคปเจอร์ ซึ่งเป็นโหมดที่กำหนดให้ไมโครคอนโทรลเลอร์ทำการตรวจจับการเปลี่ยนแปลงสถานะทางลอจิกที่ขา T2EX ทั้งนี้เพื่อใช้ประโยชน์ในการวัดคาบเวลา ความถี่ ตลอดจนการเลี่ยนแปลงของสัญญาณพัลส์ที่ที่ขา T2EX นี้

รีจิสเตอร์ควบคุม (Control register)

รีจิสเตอร์ SPR ที่ใช้ในการควบคุมการงานในส่วนต่าง ๆ ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ยังมีอีกหลายตัวประกอบด้วย

รีจิสเตอร์ PCON เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อสารอนุกรม และกำหนดการทำงานในโหมดประหยัดพลังงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของวงจรสื่อสารอนุกรมภายใน ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจิสเตอร์ TCON และ T2CON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไทเมอร์เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดย T2CON ใช้สำหรับไทเมอร์เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช AT89C52 และในอนุกรม AT89Sxx

รีจิสเตอร์ TMOD และ T2MOD เป็นรีจิสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทเมอร์ / เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดย T2MOD ใช้สำหรับไทเมอร์ / เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52

รีจิสเตอร์ IE และ IP เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการตอบสนองการอินเตอร์รัปต์ (interrupt : การขัดจังหวะการทำงานปกติของซีพียู) โดย IE เป็นรีจิสเตอร์สำหรับอีนาเบิลหรือใช้ในการกำหนดลักษณะของการตอบสนองการอินเตอร์รัปต์ ในขณะที่ IP เป็นรีจิสเตอร์สำหรับการกำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่า จะให้ซีพียูตอบสนองการเกิดอินเตอร์รัปต์ในลักษณะใดก่อนหรือหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าถึงข้อมูลไมโครคอนโทรลเลอร์ MCS-51

การเข้าถึงข้อมูล (addressing) เป็นวิธีการที่ไมโครคอนโทรลเลอร์ใช้ในการระบุตำแหน่งเพื่อเข้าถึงข้อมูลที่ต้องการดำเนินการ สำหรับไมโครคอนโทรลเลอร์ MCS-51 มีวิธีการเข้าถึงข้อมูลอยู่ 5 แบบคือ

แบบทันทีทันใด (immediate addressing mode)

แบบโดยตรง (direct addressing mode)

แบบผ่านรีจิสเตอร์ (register addressing mode)

แบบโดยอ้อม (indirect addressing mode)

แบบอินเด็กซ์ หรือแบบโดยอ้อมผ่านค่าดัชนี (indexed addressing mode)

การเข้าถึงข้อมูลแบบทันทีทันใด

เป็นวีการเข้าถึงข้อมูลที่มีความซับซ้อนน้อยที่สุด สามารถทำความเข้าใจได้ง่าย เนื่องจากสามารถโอนย้ายหรือเข้าถึงข้อมูลของรีจิสเตอร์ได้ด้วยค่าของข้อมูลตรง ๆ ที่มีรูปแบบตัวอย่างดังนี้ คำสั่ง รีจิสเตอร์หรือตำแหน่งของหน่วยความจำ ,# ข้อมูล (ฐานสอง , ฐานสิบ หรือ ฐานสิบหก) เป็นการนำค่าของข้อมูลที่ตามหลังเครื่องหมาย # ไปดำเนินการกับรีจิสเตอร์หรือตำแหน่งของหน่วยความจำปลายทาง ตัวอย่างการเขียนโปรแกรมที่ใช้การเข้าถึงแบบทันทีทันใดมีดังนี้

MOV RO, # 40 H : เป็นการนำค่า 40H ไปเก็บไว้ในรีจิสเตอร์ RO

การเข้าถึงข้อมูลแบบโดยตรง

เป็นการเข้าถึงข้อมูลโดยใช้หมายเลขตำแหน่งของหน่วยความจำภายในต้นทาง เพื่อเรียกหรือโอนย้ายข้อมูลได้โดยตรง แล้วนำข้อมูลจากหน่วยความจำภายในต้นทางนั้นไปเก็บยังรีจิสเตอร์หรือหน่วยความจำภายในปลายทาง การเข้าถึงแบบนี้จะใช้ได้กับหน่วยความจำข้อมูลภายในของไมโครคอนโทรลเลอร์ MCS-51 เท่านั้น สามารถอ้างแอดเดรสได้ 26 ตำแหน่ง ตั้งแต่ 00H-FFH ซึ่งก็คือ แอดเดรสของหน่วยความจำข้อมูลภายในนั่นเอง ดังตัวอย่าง

MOV A, 50 H : นำข้อมูลที่อยู่ในหน่วยความจำแอดเดรส 50H ไปเก็บไว้ใน A

การเข้าถึงข้อมูลแบบผ่านรีจิสเตอร์

การเข้าถึงข้อมูลแบบนี้จะคล้ายกับแบบโดยตรง ต่างกันตรงที่ข้อมูลต้นทางจะถูกเก็บไว้ในรีจิสเตอร์แบงก์ R0-R7 ส่วนรีจิสเตอร์ปลายทางต้องเป็นรีจิสเตอร์ A หรือ แอกคิวมูเลเตอร์ มีรูปแบบตัวอย่างดังนี้

MOV A, Rn : Rn เป็นรีจิสเตอร์แบงก์ใด ๆ (R0-R7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าถึงข้อมูลแบบโดยอ้อม

การเข้าถึงข้อมูลแบบนี้มีความซับซ้อนเพิ่มเติมจากแบบโดยตรง โดยค่าแอดเดรสของข้อมูลต้นทางจะถูกเก็บไว้ในรีจิสเตอร์ R0 หรือ R1 หรือ DPTR หรือใช้รีจิสเตอร์ R0, R1 และ DPTR เป็นตัวชี้แอดเดรสของหน่วยความจำแทน และต้องใส่สัญลักษณ์ @ หน้ารีจิสเตอร์ที่ใช้เป็นตัวชี้แอดเดรสต้นทางด้วย มีตัวอย่างดังนี้

MOV A, @Rn หรือ MOV A, @ DPTR

การเข้าถึงข้อมูลแบบอินเด็กซ์

วิธีการเข้าถึงข้อมูลแบบนี้มีความซับซ้อนมากที่สุด เนื่องจากต้องใช้รีจิสเตอร์ 2 ตัว ทำงานร่วมกัน แล้วใช้คำสั่งการโอนย้ายข้อมูล MOV A, @A+DPTR หรือ MOV A,@A+PC มาจัดการโอนย้ายข้อมูล

รีจิสเตอร์ DPTR และ PC ถูกนำมาใช้เป็นตัวชี้แอดเดรสเริ่มต้น และใช้ค่าที่เก็บอยู่ในรีจิสเตอร์ A เป็นตัวเลื่อนแอดดเดรสต้นทาง จากนั้นจึงทำการโอนย้ายข้อมูลจากแอดเดรสที่ถูกชี้ด้วยผลรวมของค่าในรีจิสเตอร์ A กับค่าของ DPTR มาเก็บไว้ในรีจิสเตอร์ A ในรูปที่ 5-2 แสดงกระบวนการเข้าถึงข้อมูลแบบนี้

การเข้าถึงข้อมูลแบบอินเด็กซ์หรือแบบโดยอ้อมผ่านค่าดัชนี ถึงแม้ว่าจะมีความซับซ้อนมากกว่าทุกวิธีที่กล่าวมาก่อนหน้านี้ แต่จะเป็นประโยชน์อย่างมากในการนำไปใช้เปิดตารางข้อมูลหรือ LooK up table

สแต็กในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

สแต็ก (Stack) คือ ชื่อเรียกพื้นที่บางส่วนของหน่วยความจำข้อมูลที่สำรองไว้สำหรับเก็บค่าของรีจิสเตอร์ เนื่องจากในการทำงานของไมโครคอนโทรลเลอร์บ่อยครั้งที่รีจิสเตอร์ตัวหนึ่งต้องทำงานกับข้อมูลหรือเงื่อนไขหลายอย่างภายในโปรแกรมควบคุมเดียวกัน และในขณะที่ทำงานอยู่อาจมีความจำเป็นต้องกระโดดหรือเปลี่ยนแปลงแอดเดรสในการทำงานชั่วคราว จะด้วยการขัดจังหวะหรือที่เรียกว่า อินเตอร์รัปต์ หรือการไปทำงานยังโปรแกรมย่อยเนื่องจากเงื่อนไขในการกระโดดเป็นจริง เป็นต้นข้อมูลสุดท้ายก่อนหน้าที่จะต้องกระโดดหรือถูกขัดจังหวะมีความจำเป็นต้องเก็บรักษาไว้ เนื่องจากเมื่อไปทำงานที่โปรแกรมย่อยอื่นหรือทำการบริการอินเตอร์รัปต์เรียบร้อยแล้ว จึงต้องกลับมาทำงานเดิมต่อไป ข้อมูลสุดท้ายจะต้องเรียกออกมาใช้เพื่อการทำงานยังคงความต่อเนื่องต่อไปได้

สแต็กคือ พื้นที่สำหรับเก็บข้อมูลชั่วคราวเหล่านั้น ดังนั้นในการจัดสรรหน่วยความจำข้อมูลเพื่อใช้งานจะต้อสำรองพื้นที่บางส่วนสำหรับจัดสรรเป็นสแต็กเสมอ สำหรับในไมโครคอนโทรลเลอร์ MCS-51 สามารถเลือกพื้นทีสำหรับการจัดสรรเป็นสแต็กได้ตลอดย่านของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำข้อมูลในกรณีที่ใช้หน่วยความข้อมูลภายในก็จะมีขอบเขตค่อนจำกัด แต่ถ้าใช้หน่วยความจำข้อมูลภายนอกพื้นที่ของสแต็กก็จะมีได้มากขึ้น

อย่างไรก็ตามเพื่อให้ทำงานเป็นไปอย่างรวดเร็วและไม่สิ้นเปลืองหน่วยความจำข้อมูลในการจัดสรรเป็นสแต็ก จึงมักมีการจัดสรรพื้นที่ไว้เก็บค่าของสแต็กไว้ในหน่วยความจำข้อมูลในไมโครคอนโทรลเลอร์ และสำรองไว้ประมาณ 10 ไบต์

จุดเริ่มต้นของสแต็ก

เริ่มที่ตำแหน่งใดก็ได้ในพื้นที่ของหน่วยความจำข้อมูลแต่ที่นิยมกันทั่วไปจะยึดจากรีจิสเตอร์ ชี้ตำแหน่งของสแต็ก หรือ สแต็กพอยน์เตอร์ (SP: Stack Pointer) ซึ่งเป็นรีจิสเตอร์ฟังก์ชั่นพิเศษตัวหนึ่งมีแอดเดรสสำหรับเก็บข้อมูลตำแหน่งของสแต็กอยู่ที่ 81H มีขนาด 8 บิต เมื่อเกิดการรีเซทขึ้นในระบบ ค่าของรีจิสเตอร์ SP จะเป็น 07H เสมอ ดังนั้น เพื่อให้จดจำได้ง่ายและสะดวกในการเขียนโปรแกรมที่นิยมกำหนดให้เพื้นที่สแต็กเริ่มต้นที่แอดเดรส 08H ของหน่วยความจำข้อมูล เนื่องจากที่แอดเดรส 07H ถูกใช้โดยรีจิสเตอร์ SP แล้ว

เมื่อเป็นเช่นนี้จึงสามารถกำหนดแอดเดรสของสแต็กได้ว่า ต้องเป็นแอดเดรสของหน่วยความจำข้อมูลที่อยู่ถัดจากแอดเดรสที่ถูกใช้สำหรับรีจิสเตอร์ SP

การเก็บข้อมูลลงในสแต็กและการอ่านข้อมูลออกจากสแต็ก

เพื่อให้เข้าใจถึงกระบวนการเก็บและอ่านข้อมูลกับสแต็ก ขอให้จำลองภาพว่า สแต็กคือกล่องที่มีช่องสำหรับนำของเข้าและออกได้เพียงทางเดียว ดังนั้นของชิ้นแรกที่ใส่ลงไปในกล่องจะอยู่ล่างสุดและของที่ใส่ลงไปในลำดับสุดท้ายจะอยู่บนสุด เมื่อต้องการนำของออกจากกล่องจะต้องนำของที่ใส่ลงไปในลำดับสุดท้ายออกมาก่อนเรียงลำดับไปเรื่อย ๆ ของชิ้นสุดท้ายที่นำออกจากกล่องก็คือ ของชิ้นแรกที่ใส่ลงไปนั่นเอง กระบวนการดังกล่าวเรียกว่า เข้าก่อนออกสุดท้าย หรือ FILO (First In Last Out)

การบรรจุข้อมูลลงในสแต็กเรียกว่า พุช (Push) เมื่อทำการพุชข้อมูลลงไป รีจิสเตอร์ SP ก็จะมีค่าเพิ่มขึ้น ดังนั้นรีจิสเตอร์ SP จึงเป็นตัวบอกถึงตำแหน่งบนสุดของสแต็กเสมอ ส่วนการนำข้อมูลออกจากสแต็กเรียกว่า ป๊อป (Pop) เมื่อทำการป๊อปข้อมูลออกไป ค่าของรีจิสเตอร์ SP จะลดลงดังนั้นผู้ใช้งานจึงต้องระมัดระวังในเรื่องของลำดับการเก็บและเรียกข้อมูลภายในสแต็ก โดยเฉพาะอย่างยิ่งเมื่อต้องการกระโดดไปทำงานในโปรแกรมย่อยและต้องกลับมาทำงานยังตำแหน่งแอดเดรสสุดท้ายก่อนการกระโดดเมื่อเสร็จสิ้นภาระกิจในโปรแกรมย่อยแล้ว

ตัวอย่างการเขียนโปรแกรมเพื่อใช้งานสแต็ก

โปรแกรมต่อไปนี้เป็นโปรแกรมตัวอย่างที่เขียนเพื่อแสดงให้เห็นถึงวิธีการใช้งานสแต็กตั้งแต่การกำหนดจุดเริ่มต้นของสแต็ก และการเรียกข้อมูลออกจากสแต็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ORG     000H
Sp      EQU     07H     : กำหนดตัวชี้สแต็ก
        MOV     A,#02H
        MOV     B,#05H
        ACALL   TEMP    : กระโดดไปโปรแกรมย่อย TEMP
        MUL     AB
        *
        *
TEMP    PUSH    A       : เก็บค่าของรีจิสเตอร์ A ลงในสแต็ก
        PUSH    B       : เก็บค่าของรีจิสเตอร์ B ลงในสแต็ก
        *
        *
        POP     B       : เรียกค่าของรีจิสเตอร์ B จากสแต็ก
        POP     A       : เรียกค่าของรีจิสเตอร์ A จากสแต็ก
        RET             : ออกจากโปรแกรมย่อย TEMP
        END
```

สรุป การใช้งานสแต็กมีลำดับขั้นตอนโดยสรุปดังนี้

1. กำหนดจุดเริ่มต้นของสแต็ก โดยการกำหนดค่าลงในรีจิสเตอร์ SP แอดเดรสที่อยู่ถัดจากค่าของ SP คือ แอดเดสแรกของสแต็กที่สามารถเก็บข้อมูลได้ โดยปกติจะใช้ค่า SP เท่ากับ 07H ดังนั้น แอดเดรสของสแต็กสำหรับเก็บข้อมูลคือ 08H
2. เก็บข้อมูลลงในสแต็กด้วยคำสั่ง PUSH
3. ทุกครั้งที่เก็บข้อมูลลงในสแต็กค่าของ SP จะเพิ่มขึ้น
4. สแต็กมีการลำดับการเข้าออกของข้อมูลแบบ FILO
5. เรียกข้อมูลออกจากสแต็กด้วยคำสั่ง POP
6. ไม่ควรสำรองพท้นที่ของสแต็กไว้มากเกินความจำเป็น เพราะจะทำให้จำนวนของหน่วยความจำข้อ มูลเหลือน้อยลง อาจไม่เพียงพอต่อการใช้งาน โดยเฉพาะอย่างยิ่งการใช้หน่วยความจำข้อมูลในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. จัดลำดับการเรียกข้อมูลออกจากสแต็กให้ถูกต้อง ข้อมูลที่เก็บเข้าไปลำดับสุดท้ายต้องเรียกออกมาก่อนเสมอ

อินเตอร์รัปต์ ในไมโครคอนโทรลเลอร์ MCS-51

การอินเตอร์รัปต์ (interrupt) เป็นชื่อเรียกกระบวนการที่เข้ามาขัดจังหวะการทำงานโดยปกติของไมโครคอนโทรลเลอร์ ในไมโครคอนโทรลเลอร์ MCS-51 สามารถตอบสนองการอินเตอร์รัปต์ที่เกิดขึ้นได้ จาก 5 แหล่ง กำเนิดสำหรับเบอร์ AT89C51 ประกอบด้วย การรับสัญญาณอินเตอร์รัปต์จากภายนอกผ่านทางขา INT0 และ INT1 สัญญาณอินเตอร์รัปต์จากไทเมอร์ / เคาน์เตอร์ T0 และ T1 และสัญญาณอินเตอร์รัปต์จากพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ในขณะที่ไมโครคอนโทรลเลอร์เบอร์ AT89C52 และในอนุกรม AT89Sxx สามารถตอบสนองต่อการอินเตอร์รัปต์ได้จาก 6 แหล่ง กำเนิด โดยเพิ่มการรับสัญญาณอินเตอร์รัปต์จากไทเมอร์ / เคาน์เตอร์ 2 อีกแหน่งแหล่งกำเนิด

การจัดการอินเตอร์รัปต์ไมโครคอนโทรลเลอร์ MCS-51

เมื่อมีการอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 เกิดขึ้น และมีการอีนาเบิลการตอบสนองการอินเตอร์รัปต์ไว้ กระบวนการหลังจากนั้นซีพียูจะทำกระการกระโดดไปยังแอดเดรสในหน่วยความจำที่กำหนดไว้ เรียกตำแหน่งแอดเดรสนี้ว่า แอดเดรสรัปต์เวเตอร์ (interrupt vector address) ดังนั้นจะต้องมีการเขียนโปรแกรมย่อยการบริการอินเตอร์รัปต์ไว้ที่แอดเดรสอินเตอร์รัปต์เวกเตอร์นี้ โดยค่าของแอดเดรสอินเตอร์รัปต์เวกเตอร์จะแตกต่างกันในการอินเตอร์รัปต์แบบต่างๆ ดังมีรายละเอียดต่อไปนี้

การอินเตอร์รัปต์ภายนอกที่ขา INT0 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0003H
การอินเตอร์รัปต์จากไทเมอร์ 0 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 000BHH
การอินเตอร์รัปต์ภายนอกที่ขา INT1 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0013H
การอินเตอร์รัปต์จากไทเมอร์ 1 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 001BH
การอิเนอตร์รัปต์จากพอร์ตอนุกรม มีค่าแอดเดสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0023H
การอินเตอร์รัปต์จากไทเมอร์ 2 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 002BH

การเขียนโปรแกรมย่อยบริการอินเตอร์รัปต์

มีหลักการโดยทั่ว ๆ ไปดังนี้

1. ต้องเริ่มต้นด้วยแอดเดรสอินเตอร์รัปต์เวกเตอร์เสมอ เพื่อให้การตรวจสอบการทำงานได้งานและแยกส่วนของโปรแกรมย่อยนี้ออกจากโปรแกรมฟลักหรือโปรแกรมย่อยอื่น ๆ อย่างชัดเจนด้วยคำสั่ง ORG xxxxH (ค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อเข้าสู่โปรแกรมย่อย ควรเก็บค่าของรีจิสเตอร์หรือแฟลกที่ใช้แสดงสถานะต่างๆ ซึ่งต้องมีการใช้งานในโปรแกรมบ่อยบริการอินเตอร์รัปต์นี้ไว้ในสแต็กเสียก่อน เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นต่อการทำงานของทั้งโปรแกรมบริการอินเตอร์รัปต์นี้และโปรแกรมหลักด้วยคำสั่ง PUSH
3. เมื่อเขียนโปรแกรมบริการอินเตอร์รัปต์เรียบร้อยแล้ว ให้ทำการคืนค่าของรีจิสเตอร์ที่นำมาใช้โปรแกรมบริการอินเตอร์รัปต์นี้ไปใช้งาน ซึ่งในทางปฏิบัติจริง ไม่พบมากนัก
4. ปิดท้ายโปรแกรมย่อยบริการอินเตอร์รัปต์ด้วยคำสั่ง RET1 เสมอ

ตัวอย่างโปรแกรมบริการอินเตอร์รัปต์ด้วยคำสั่งภายนอกที่ขา INT0 มีดังนี้

```
ORG     0003H
PUSH    A
PUSH    P1
PUSH    PSW
*
*               : ส่วนของโปรแกรมย่อยบริการอินเตอร์รัปต์
*
POP     PSW
POP     P1
POP     A
RETI
```

รีจิสเตอร์ที่เกี่ยวข้องกับการอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51

การอินเตอร์รัปต์ในไมโครโทรลเลอร์ MCS-51 แบบแฟลช มีรีจิสเตอร์ที่ต้องเกี่ยวข้องอยู่ 2 ตัวดังมีรายละเอียดต่อไปนี้

รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์ หรือ IE (interrupt Enable register)

มีแอดเดรสอยู่ที่ A8H ในพื้นที่ของริจิสเตอร์ฟังก์ชั่นพิเศษหรือ SFR มีขนาด 8 บิต สามารถเข้าถึงได้ระดับบิต ใช้ในการอีนาเบิลการตอบสนองการอินเตอร์รัปต์ในแบบต่างๆ มีรายละเอียดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
EA	-	ET 2	ES	ET 1	EX 1	ET 0	EX 0

EA (Global enable / disable interrupt) : ใช้ในการอีนาเบิลและดิสเอเบิลทั้งหมด

"0" ดิสเอเบิลการอินเตอร์รัปต์ นั่นคือ กำหนดให้ไมโครคอนโทรลเลอร์ไม่ตอบสนองการอินเตอร์รัปต์ ไม่ว่าจะมาจากแหล่งกำเนินใดก็ตาม

"1" อีนาเบิลการอินเตอร์รัปต์ นั่นคือ กำหนดให้ไมโครคอนโทรลเลอร์สามารถตอบสนองการอินเตอร์รัปต์จากแหล่งกำเนิดต่างๆ

นั่นคือ ถ้าต้องการให้ไมโครคอนโทรลเลอร์ตอบสนองการอินเตอร์รัปต์ไม่ว่าจะแหล่งกำเนิดใด จะต้องเซตบิต นี้ก่อนเสมอ สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ **ET2 (Timer 2 interrupt enable) :** ใช้ในการอีนาเบิลอินเตอร์รัปต์อันเนื่องจากการโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์ / เคาน์เตอร์ 2 จะมีเฉพาะในเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้น บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

ES (Serial port interrupt enable bit) : ใช้ในการอีนาเบิลการอินเตอร์รัปต์อันเนื่องจากการรับหรือส่งข้อมูลทางพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

ET1 (Timer 1 interrupt enable) : ใช้ในการอีนาเบิลการอินเตอร์รัปต์อันเนื่องจากการโอเวอร์โฟลวในไทเมอร์ / เคาน์เตอร์ 1 บิต นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ **EX1 (External interrupt 1 enable bit) :** ใช้ในการอีนาเบิลการอินเตอร์รัปต์อันเนื่องจากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT1 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

ET0 (Timer 0 interrupt enable) : ใช้ในการอีนาเบิลการอินเตอร์รัปต์อันเนื่องมากจากการโอเวอร์โฟลวในไทเมอร์ / เคาน์เตอร์ 0 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

EX0 (External interrupt 1 enable bit) : ใช้ในการอีนาเบิลการอินเตอร์รัปต์อันเนื่องจากมาจากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT0 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ สำหรับบิต 6 ของรีจิสเตอร์ IE ไม่มีการใช้งานต้องกำหนดให้เป็น "0" เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์จัดลำดับความสำคัญการตอบสนองการอินเตอร์รัปต์หรือ IP (Interrupt Priority register)

มีแอดเดรสอยู่ที่ B8H ในพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษหรือ SER มีขนาด 8 บิต สามารถเข้าถึงได้ระดับบิต ใช้ในการเลือกลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่า ต้องการให้ตอบสนองสัญญาณอินเตอร์รัปต์จากแหล่งกำเนินใดเป็นลำดับก่อนหลัง ถ้าต้องการให้การอินเตอร์รัปต์จากแหล่งดำเนินใดมีความสำคัญสูงสุด ให้กำหนดที่บิตนั้น เป็น "1" มีรายละเอียดของรีจิสเตอร์ IP ดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
-	-	PT2	PS	PT1	PX1	PT0	PX0

PT2 (Timer interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเตอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์ / เคาน์เตอร์ 2 จะมีเฉพาะในเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้น บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PS (Serial interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเตอร์รัปต์อันเนื่องมาจากการรับ หรือส่งข้อมูลทางพอร์ตอนุกรมภายในตัวของอุปกรณ์ไมโครคอนโทรลเลอร์ MCS-51สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PX1 (External interrupt 1 priority bit) : ใช้ในการกำหนดความสำคัญของการอินเตอร์รัปต์อันเนื่องจากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT1 บิต นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PT0 (Timer 0 interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเตอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวในไทเมอร์ / เคาน์เตอร์ 0 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

PX0 (External interrupt 0 priority bit) : ใช้ในการกำหนดความสำคัญของการอินเตอร์รัปต์อันเนื่องจากสัญญาณภายนอกที่ฟ้อนเข้ามายังขา INT0 บิตนี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ สำหรับบิต 6 และ 7 ของรีจิสเตอร์ IP ไม่มีการใช้งานต้องกำหนดให้เป็น "0" เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งกำเนินสัญญาณอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51

สัญญาณอินเตอร์รัปต์ภายนอก

เป็นการตรวจสอบสัญญาณที่ส่งเข้ามายังขา INT0 และ INT1 หากตรงตามเงื่อนไขที่กำหนดก็จะทำให้เกิดการอินเตอร์รัปต์ขึ้น โดยการอีนาเบิลการอินเตอร์รัปต์แบบนี้สามารถกระทำได้การกำหนดค่าในรีจิสเตอร์ IE ที่บิต EX0 สำหรับสัญญาณอินเตอร์รัปต์ขา INT0 และบิต EX1 สำหรับสัญญาณอินเตอร์รัปต์ที่ขา INT1 และทำการเลื่อนเงื่อนไขของการตรวจสอบสัญญาณในรีจิสเตอร์ TCON ที่บิต IE0 สำหรับสัญญาณอินเตอร์รัปต์ที่ขา INT0 และบิต IE1 สำหรับสัญญาณอินเตอร์รัปต์ที่ขา INT1

เงื่อนไขการตรวจสอบสัญญาณอินเตอร์รัปต์ที่ขา INT0 และ INT1 มีด้วยกัน 2 ลักษณะคือ

1. ตรวจสอบระดับลอจิก ถ้าหากบิต Iex ในรีจิสเตอร์ TCON เป็น "0" จะเกิดการอันเตอร์รัปต์จากภายนอกที่ขา INT หรือ INT1 ได้ก็ต่อเมื่อตรวจพบระดับลอจิกต่ำหรือ "0" เมื่อเกิดการอินเตอร์รัปต์แล้ว ให้ดำเนินการให้สัญญาณที่ขานี้กลับสู่ระดับลอจิก "1" ก่อนที่การบริการอินเตอร์รัปต์เสร็จสิ้น เพื่อป้องกันการเกิดอินเตอร์รัปต์ซ้อน

2. ตรวจสอบขอบขาของสัญญาณ ถ้าหากบิต Iex ในรีจิสเตอร์ TCON เป็น "1" จะเกิดการอินเตอร์รัปต์จากภายนอกที่ขา INT0 หรือ INT1 ได้ก็ต่อเมื่อตรวจพบการเปลี่ยนแปลงของสัญญาณที่ขา INT0 หรือ INT1 จาก "1" เป็น "0" หรือตรวจสอบพบขอบขาลงของสัญญาณที่ป้อมมายังขา INT0 หรือ INT1 และต้องมีการรักษาสถานะลอจิก "0" นี้เป็นเวลาอย่างน้อย 1 แมชีนไซเกิล จึงถือว่าเกิดการอินเตอร์รัปต์อย่างสมบูรณ์ เมื่อเกิดการอินเตอร์รัปต์ขึ้นซีพียูภายในไมโครคอนโทรลเลอร์จะกระโดดไปยังแอดเดรส 0003H สำหรับการ อินเตอร์รัปต์ที่ขา INT0 และ 0003H สำหรับการอินเตอร์รัปต์ที่ขา INT1

ลำดับความสำคัญของการอินเตอร์รัปต์ในโครคอนโทรลเลอร์ MCS-51

การกำเนิดสัญญาณอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 มีได้จาก 5-6 แหล่ง ดังนั้นจึงต้องมีการจัดลำดับความสำคัญ ในกรณีที่เกิดการอินเตอร์รัปต์ขึ้นพร้อมๆ จากหลายแหล่งกำเนิดโดยสามารถกำหนดความสำคัญได้ที่รีจิสเตอร์ IP

อย่างไรก็ตาม ลำดับความสำคัญของการอินเตอร์รัปต์โดยปกติหรือในกรณีกำหนดข้อมูลในรีจิสเตอร์ IP ให้เป็น "1" ทุกบิต (ยกเว้น 6 และ 7) จะเรียงลำดับจากความสำคัญสูงสุดไปจนถึงต่ำสุดดังนี้

1. อินเตอร์รัปต์ภายนอกที่ขา INT0 หรือการเซตของบิต IE0
2. อินเตอร์รัปต์จากไทเมอร์ 0 หรือการเซตบิตของบิต TF0
3. อินเตอร์รัปต์ภายนอกที่ขา INT1 หรือการเซตของบิต IE1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. อินเตอร์รัปต์จากไทเมอร์ 1 หรือการเซตของบิต TF1

5. อินเตอร์รัปต์จากพอร์ตอนุกรม หรือการเซตของบิต RI หรือ TI

6. อินเตอร์รัปต์จากไทเมอร์ 2 หรือการเซตของบิต TF2 หรือ EXF2

กระบวนการรีเซตในไมโครคอนโทรลเลอร์ MCS-51

การรีเซต (reset) เป็นกระบวนการที่ทำให้ซีพียูในไมโครคอนโทรลเลอร์กลับไปเริ่มต้นทำงานใหม่ที่แอดเดรส 0000H ในหน่วยความจำโปรแกรม พร้อมกันนั้นยังส่งผลให้เกิดการเคลียร์ค่าของข้อมูลในรีจิสเตอร์ภายในไมคครคอนโทรลเลอร์ และทำให้รีจิสเตอร์ทุกตัวกลับไปสู่สภาวะเริ่มต้นทำงานใหม่

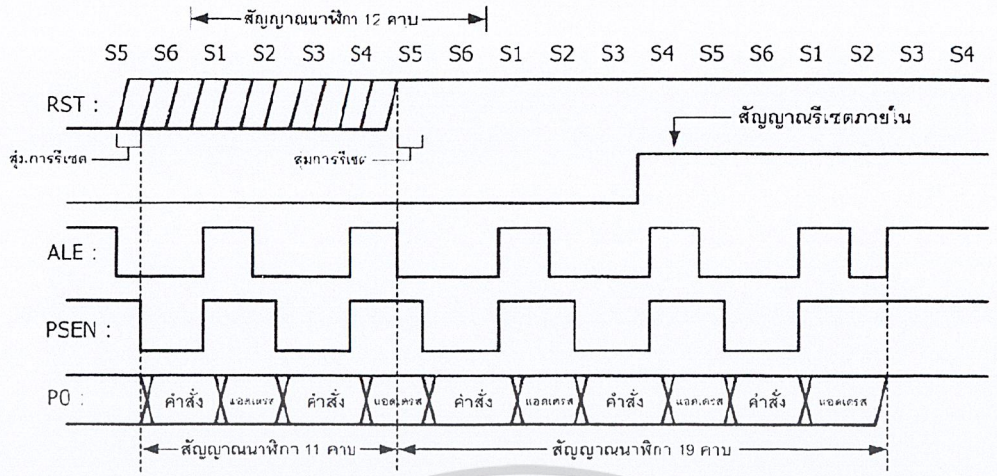
ดังนั้นจึงอาจพิจารณาการรีเซตเสมือนเป็นการอินเตอร์รัปต์แบบหนึ่งก็ได้ และค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์ คือ 0000H ที่แตกต่างกันคือ ในกระบวนการอินเตอร์รัปต์ค่าของรีจิสเตอร์ต่างๆ ก่อนหน้าการอินเตอร์รัปต์ จะยังคงอยู่ ไม่เปลี่ยนแปลงจนกว่าจะมีการกระทำคำสั่งในโปรแกรมย่อยบริการอินเตอร์รัปต์ แต่ใจการรีเซตของรีจิสเตอร์ส่วนใหญ่จะกลายเป็น 000H เว้นแต่รีจิสเตอร์พอร์ตที่จะกลายเป็น FFH

การรีเซตเกิดขึ้นอย่างไร

การเกิดรีเซตในไมโครคอนโทรลเลอร์ MCS-51 เกิดขึ้นได้เพียงกรณีเดียวคือ ที่ขา RST (ขา9) ได้รับลอจิก "1" อย่างน้อยเป็นเวลา 2 แมชีนไซเกิล หรือ 24 คาบเวลาของสัญญาณนาฬิกา โดยที่วงจรกำเนิดสัญญาณยังคงทำงานอยู่ เมื่อเกิดการรีเซตขึ้น ไมโครคอนโทรลเลอร์จะสร้างสัญญาณรีเซตภายในขึ้นเพื่อตอบสนองการรีเซตจากภายนอกดังแสดงไดอะแกรมของกระวนการรีเซตภายในไมโครคอนโทรเลอร์ MCS-51 ในรูป

สัญญาณรีเซตที่เกิดขึ้นจากภายนอกจะไม่มีความสอดคล้องกับสัญญาณนาฬิกาภายในของไมโครคอนโทรลเลอร์แต่อย่างใด เนื่องจากการรีเซตอาจเกิดขึ้นเมื่อใดก็ได้ โดยจะมีการสุ่มตรวจสอบสถานะลอจิกที่ขา RST ในสเตต 5 เฟส 2 ของทุกแมชีนไซเกิล เมื่อสามารถตรวจจับลอจิก "1" ที่ขา RST ได้และเมื่อเกิดสัญญาณรีเซตภายในขึ้น กระบวนการรีเซตของไมโครคอนโทรลเลอร์จะเริ่มต้นขึ้นอย่างเป็นทางการ ขาพอร์ตทั้งหมดยังคงรักษาข้อมูลปัจจุบันไว้ได้อีก 19 คาบ เวลาของสัญญาณนาฬิกาหลังจากที่ขา RST ได้รับลอจิก "1" จากนั้นเปลี่ยนแปลงค่ากลายเป็น FFH ทั้งหมด ของสัญญาณ ALE และ PSEN จะกลายเป็นลอจิกสูง เมื่อการรีเซตขึ้นอย่างสมบูรณ์ และเมื่อขา RST ถูกทำให้กลายเป็นลอจิกต่ำ อันเป็นการแจ้งให้ทราบว่า สามารถเริ่มต้นกระบวนการทำงานปกติได้แล้ว สัญญาณ ALE และ PSEN จะเกิดขึ้นใหม่อีกครั้งหลังจากที่ขา RST เป็นลอจิกต่ำ 1-2 แมชีนไซเกิล ทั้งนี้เพื่อให้อุปกรณ์ต่อร่วมกันอยู่ไมโครคอนโทรลเลอร์อยู่ในสภาวะพร้อมทำงาน และสามารถทำงานในจังหวะที่สอดคล้องกับไมโครคอนโทรลเลอร์ MCS-51 ได้อย่างเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.17 แสดงไดอะแกรมของกระบวนการรีเซตภายในไมโครคอนโทรลเลอร์ MCS-51

ผลของการรีเซตที่มีต่อรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51

เมื่อเกิดการรีเซตขึ้นและไมโครคอนโทรลเลอร์ตอบสนองการรีเซตจากภายนอกด้วยการสร้างสัญญาณรีเซตภายในให้เกิดขึ้น สิ่งที่เกิดขึ้นตามมาคือ มีการเขียนข้อมูล 0 ให้แก่รีจิสเตอร์ฟังก์ชั่นพิเศษทั้งหมด ยกเว้นรีจิสเตอร์พอร์ตทั้ง 3 ตัว รีจิสเตอร์ SP และ SBUF โดยค่าของรีจิสเตอร์พอร์ต P1-P3 จะหล่ายเป็น FFH ในขณะที่รีจิสเตอร์ SP มีค่าเท่ากับ 07H ส่วนค่าของรีจิสเตอร์ SBUF จะไม่แน่นอนและไม่สามารถกำหนดได้ (indetermine) ในตารางที่ 10-1 แสดงค่าของรีจิสเตอร์หลังจากเกิดการรีเซตอย่างสมบูรณ์

กลไกที่สำคัญอีกประการหนึ่งของกระบวนการรีเซตคือ ค่าของหน่วยความจำข้อมูลแรกภายในจะไม่ได้รับผลกระทบจากการรีเซตแต่อย่างใด ดังนั้น ในการเขียนโปรแกรมควบคุม ที่ส่วนต้นของโปรแกรมจึงควรมีการเคลียร์ค่าของหน่วยความจำข้อมูลภายในที่จะต้องใช้งานเสมอ เพื่อป้องกันการใช้ข้อมูลที่เกิดขึ้นก่อนการรีเซตมาประมวลผลเนื่องจากโอกาสในการรับรู้ว่าข้อมูลของหน่วยความจำการรีเซตมีค่าเท่าใดมีน้อยมาก

ตารางที่ 2.8 แสดงค่าของข้อมูลที่เกิดขึ้นของรีจีสเตอร์ในไมโครคอนโทรลเลอร์หลังการรีเซต

ชื่อรีจิสเตอร์ฟังก์ชั่นพิเศษ	ค่าที่เกิดขึ้นหลังการรีเซต
PC	0000H
ACC	00H
B	00H
PSW	00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อรีจิสเตอร์ฟังก์ชั่นพิเศษ	ค่าที่เกิดขึ้นหลังการรีเซต
SP	07H
DPTR	0000H
P0 - P3	FFH
IP (AT89C51)	XXX00000B
IP (AT89C52, AT89Sxx)	XX000000B
IE (AT89C51)	0XX00000B
IE (AT89C52, AT89Sxx)	0X000000B
TMOD	00H
T2MOD (AT89C52, AT89Sxx)	XXXXXX00B
TCON	00H
T2CON(AT89C52, AT89Sxx)	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2(AT89C52, AT89Sxx)	00H
TL2 (AT89C52, AT89Sxx)	00H
RCAP2H(AT89C52, AT89Sxx)	00H
RCAP2L (AT89C52, AT89Sxx)	00H
SCON	00H
SBUF	กำหนดไม่ได้
PCON	0XXX0000B

การเขียนโปรแกรมภาษาแอสเซมบลี

การเรียนรู้เพื่อใช้งานไมโครคอนโทรลเลอร์ไม่ว่าจะเป็นเบอร์ใดก็ตาม สิ่งที่มีความสำคัญในลำดับต่อมาหลังจากที่ทำความเข้าใจถึงโครงสร้างทางฮาร์ดแวร์แล้วก็คือ การเขียนโปรแกรมเพื่อกำหนดให้ไมโครคอนโทรลเลอร์ทำงาน ข้อมูลของโปรแกรมที่ไมโครคอนโทรลเลอร์ต้องการจะอยู่รูปของรหัสเลขฐานสิบหก หรือที่เรียกกันทั่วไปว่า ภาษาเครื่อง หรือแมชีนโค้ด (machine code)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของโปรแกรมภาษาแอสเซมบลี

1. ลาเบล (Label) ใช้ในการอ้างถึงบรรทัดใดบรรทัดหนึ่งของโปรแกรมที่ทำการเขียนขึ้น

2. รหัสนีโมนิก (Mnemoinc) เป็นส่วนแสดงคำสั่งของไมโครคอนโทรลเลอร์ ที่ต้องการให้กระทำ

3. โอเปอร์เรนด์ (Operand) เป็นส่วนที่แสดงถึงตัวกระทำหรือถูกกระทำและข้อมูลที่ใช่ในการกระทำตามคำสั่งที่กำหนดโดยรหัสนีโมนิกก่อนหน้านี้

4. คอมเมนต์ (Comment) เพื่อช่วยให้ผู้เขียนโปรแกรมสามารถตรวจสอบโปรแกรมที่เขียนขึ้นได้ง่าย รวมถึงเป็นประโยชน์ต่อผู้อื่นที่นำโปรแกรมที่เขียนขึ้นนี้ไปศึกษาอย่างมากด้วย

คำสั่งเทียม

ในการเขียนโปรแกรมภาษาแอสเซมบลีให้ได้ดีและสามารถตรวจสอบตลอดจนทำความเข้าใจได้ง่ายต่องอาศัยกลุ่มพิเศษอีกกลุ่มหนึ่งที่ไม่มีในชุดของไมโครคอนโทรลเลอร์จะเรียกกลุ่มคำสั่งนี้ว่า คำสั่งเทียม (Pseudo instruction)

เนื่องจากในโปรแกรมภาษาแอสเซมบลี จะมีการแบ่งส่วนเป็นโปรแกรมหลักและโปรแกรมย่อยมากมายการใช้คำสั่งเทียมจะช่วยให้ผู้เขียนโปรแกรมสามารถตรวจสอบโปรแกรมได้อย่างสะดวกขระเดียวกันยังช่วยให้การกำหนดชื่อหน้าที่ของพอร์ตหรือรีจิสเตอร์ต่างๆ สามารถกระทำได้อย่างชัดเจน และไม่ต้องกังวลเรื่องสิ้นเปลืองหน่วยความจำ เนื่องจากการใช้คำสั่งเทียมทุกคำสั่งจะไม่ใช้พื้นที่ของหน่วยความจำโปรแกรมแต่อย่างใด เนื่องจากคำสั่งเทียมนี้เป็นเพียงเครื่องมือที่ใช้ช่วยให้การอ้างถึงตำแหน่งต่างๆ หรือข้อมูลภายในโปรแกรมภาษาแอสเซมบลี เท่านั้น ซีพียูภายในไมโครคอนโทรลเลอร์มิได้กระทำคำสั่งเทียมนี้โดยตรง

```
ลาเบล     นีโมนิก   โอเปอร์แรนด์        คอมเมนต์
          ORG       0000h              ; Reset Vector
          AJMP      Initial            ; Jump to initial
INITIAL:  MOV       P0, #00000000B     ; Clear Databus
          MOV       P1, #11101111B     ; Clear Status all devices
          MOV       P2, #11111111B     ; Clear Status keypad
          MOV       P3, #00011111B     ; Clear status LCD
MAIN :    SETB      DRIVER_LE          ; Enable Driver Latch
          MOV       R0,#7              ; Set 7 times loop
          MOV       A, #00000001B      ; Initial ACC. Value
          END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.18 ตัวอย่างโปรแกรมภาษาแอสเซมบลีที่มีส่วนประกอบหลักครบถ้วน

คำสั่งเทียมของโปรแกรมแอสเซมเบอร์แต่ละตัวจะแตกต่างกันออกไป อย่างไรก็ตาม สามารถรูปคำสั่งเทียมที่มีการใช้งานกันโดยทั่วไป และโปรแกรมแอสเซมเบอร์ส่วนใหญ่สามารถตีความเข้าใจได้ ดังต่อไปนี้

EQU รูปแบบการใช้งาน ชื่อ ลาเบลหรือชื่อ ตัวแปร EQU แอดเดรสหรือค่าตัวเลขหรือชื่อตัวแปร มาจากคำว่า EQUAL ซึ่งแปลงว่า เท่ากับใช้ในการแทนตำแหน่งแอดเดรสหรือค่าคงที่ด้วยชื่อที่สามารถเข้าใจได้ง่าย เช่น

OUT_ PORT EQU PORT2

หมายความว่า กำหนดให้พอร์ต 2 มีชื่อว่า OUT_PORT ในกรณีที่ผู้เขียนโปรแกรมต้องการสื่อให้เห็นว่า พอร์ต 2 ของไมโครคอนโทรลเลอร์ได้รับการกำหนดให้เป็นพอร์ตเอาต์พุต

คำสั่ง EQU นี้ควรแทรกหรือกำหนดไว้ในตอนต้นของโปรแกรมเสมอ ซึ่งอาจต้องใช้หลายบรรทัด เมื่อใช้คำสั่ง EQU ในการแทนค่าแอดเดรสหรือข้อมูล หากมีความต้องการเปลี่ยนแปลงค่าแอดเดรสหรือข้อมูลนั้น แต่ยังคงต้องการใช้ชื่อตัวแปรเดิมอยู่ก็สามารถทำได้ง่ายดายด้วยการแก้ไขที่คำสั่ง EQU เพียงตำแหน่งเดียว เมื่อแก้ไขแล้ว ในโปรแกรมบรรทัดใดที่มีการอ้างถึงชื่อตัวแปรนั้นจะได้รับการแก้ไขค่าทั้งหมดโดยอัตโนมัติ เป็นการเพิ่มความสะดวกในการแก้ไขโปรแกรมให้แก่ผู้เขียนโปรแกรมเป็นอย่างมาก

ORG รูปแบบการใช้งาน ORG แอดเดรสเริ่มต้น หรือชื่อของลาเบลที่เป็นตำแหน่งเริ่มต้น มาจากคำว่า ORIGN หมายถึงจุดเริ่มต้นคำสั่งนี้ใช้ในการกำหนดจุดเริ่มต้นหรือแอดเดรสเริ่มต้นโปรแกรมหลัก โปรแกรมย่อยเริ่มต้นการทำงาน และโปรแกรมย่อยบริการอินเตอร์รัปต์ในหน่วยความจำโปรแกรม ทั้งนี้เนื่องจากในโปรแกรมเหล่านั้นจะมีตำแน่งเริ่มต้นที่แน่นอน

ในกรณีที่ใช้ชื่อลาเบลในการกำหนดตำแหน่งเริ่มต้นการทำงานต้องมีการกำหนดให้ทราบก่อนว่าลาเบลนั้นใช้อ้างถึงแอดเดรสใดด้วยคำสั่ง EQU ก่อนหน้าที่ใช้คำสั่ง ORG

DB (Define Byte) รูปแบบใช้งาน ชื่อลาเบล: DB รายการข้อมูล 1, รายการข้อมูล 2....., รายการข้อมูล n ใช้ในการกำหนดค่าให้แก่หน่วยความตำข้อมูลในแอดเดรสที่อ้างถึงด้วยชื่อของลาเบล มักจะใช้ประโยชน์ในรูปของตารางข้อมูล อาทิ ตารางข้อมูลการแสดงผลของ LED ตัวเลข 7 ส่วน ตารางข้อมูลของโน้ตดนตรี เป็นต้น

คำสั่ง DB นี้สามารถใช้กับข้อมูลขนาด 8 บิต และมักจะบรรจุอยู่ตอนท้ายของโปรแกรมเสมอ ตัวอย่างการใช้งานมีดังนี้

LED_ TABL : DB 3FH, O6H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH

จากตัวอย่างเป็นการกำหนดข้อมูลการแสดงผลของ LED ตัวเลข 7 ส่วน โดยที่ลาเบล LED_ TBL บรรจุข้อมูล สำหรับแสดงตัวเลข 0-9 เริ่มจากค่า 3FH เป็นข้อมูลสำหรับแสดงตัวเลข 0 ไล่ไปตามลำดับ ค่าสุดท้าย 6FH ก็จะเป็นข้อมูลแสดงเลข 9 ของ LED ตัวเลข 7 ส่วนชนิดแคโทด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร่วม ถ้าหากข้อมูลที่เรียกใช้มีเครื่องหมาย "....." (..... หมายถึง ข้อมูล) ข้อมูลดังกล่าวจะเป็นข้อมูลตัวอักษร ซึ่งแทนด้วยรหัสแอสกี้ของ ตัวอักษรตัวนั้น ๆ

DW (Define Word) รูปแบบการใช้งาน ชื่อลาเบล : DW รายการข้อมูล 1, รายการข้อมูล 2....., รายการข้อมูล n ใช้กำหนดค่าให้แก่หน่วยความจำข้อมูลในแอดเดรสที่อ้างถึงด้วยชื่อของลาเบล มีการใช้งานเหมือนกับคำสั่ง DB ต่างกันเพียงขนาดของข้อมูล โดยคำสั่ง DW นี้จะใช้ขนาดของข้อมูลเป็น 16 บิต หรือเรียกใช้ข้อมูลครั้งละ 2 ตัวอักษร

BIT รูปแบบใช้งาน ชื่อตัวแปร BIT บิตของหน่วยความจำหรือรีจิสเตอร์ที่เข้าถึงได้ระดับบิต เป็นคำสั่งที่มีลักษณะคล้ายกับคำสั่ง EQU หากแต่จะใช้เรียกแทนข้อมูลในระดับบิต ดังกล่าวดังต่อไปนี้

ตัวอย่างที่ 1

PORT_ ENB BIT P1.0

เป็นการกำหนดให้บิต 0 ของรีจิสเตอร์ P1 หรือรีจิสเตอร์พอร์ต 1 มีชื่อ PORT_ENB

ตัวอย่างที่ 2

INT_ENB BIT EA

เป็นการกำหนดให้บิต EA ของรีจิสเตอร์ IE มีชื่อว่า INT_ENB

ข้อควรระวังสำหรับการใช้คำสั่งเทียม BIT คือในโปรแกรมแอสเซมเบลอร์บางตัวอาจไม่รู้จักคำสั่งนี้

$ (string หรือ Dollar sign) สัญลักษณ์นี้จะมีใช้ในส่วนของโอเปอร์แรนด์เป็นการแทนค่าปัจจุบันของแอดเดรสที่บรรจุอยู่ในรีจิสเตอร์ PC หรือโปรแกรมเคาน์เตอร์ มักใช้ร่วมกับคำสั่งกระโดด เช่น

SJMP $

จากคำสั่งนี้หมายความว่า ให้พีซียูกระโดนไปยังแอดเดรสปัจจุบันที่เก็บอยู่ในโปรแกรมเคาน์เตอร์นั่นหมายความว่า ซีพียูจะทำงานอยู่ที่แอดเดรสนี้ไม่รู้จบหรือ endless loop อย่างไรก็ตามในโปรแกรมแอสเซมเบลอร์ บางตัวก็ไม่รู้จักเครื่องหมายนี้ จึงอาจแทนการใช้เครื่องหมาย $ ด้วยการเขียนโปรแกรมดังนี้

END_ LOOP : SJMP END_ LOOP

END เป็นคำสั่งที่ใช้แจ้งให้โปรแกรมแอสเซมเบลอร์ ทราบถึงบรรทัดสุดท้ายของโปรแกรมที่ต้องทำการแปลคำสั่ง ในโปรแกรมภาษาแอสเซมบลีทุกโปรแกรมต้องใส่คำสั่ง END นี้ไว้บรรทัดสุดท้ายเสมอ หากมีโปรแกรมต่อท้ายคำสั่ง END โปรแกรมแอสเซมเบลอร์จะไม่ทำการแปลหรือแอสเซมเบลอร์ โปรแกรมที่ต่อท้ายคำสั่ง END อย่างเด็ดขาด แม้ว่าจะเขียนถูกต้องไม่ผิดหลัก ไวยกรณ์หรือรูปแบบก็ตามในทางตรงข้ามก็จะไม่มีการแจ้งเตือนว่ามีการเขียนโปรแกรมผิดพลาดขึ้นในส่วนของโปรแกรมที่ต่อท้ายคำสั่ง END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตาม อาจมีโปรแกรมแอสเซมเบลอร์บางตัวที่สามารถแจ้งให้ผู้เขียนโปรแกรมทราบว่า มีการเขียนโปรแกรมต่อท้ายคำสั่ง END ขอให้ดำเนินการแก้ไขให้ถูกต่อไป

ข้อกำหนดที่สำคัญของการเขียนโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์

มีด้วยกัน 4 ประการคือ

1. การตีความระหว่างค่าของตัวเลขและตัวอักษร โดยปกติหากมีการเขียนตัวเลขลงในโปรแกรมความหมายของตัวเลขนั้นคือค่าจองตัวเลขฐานสิบ แต่ถ้าหากต้องการกำหนดให้เป็นตัวเลขฐานอื่นต้องมีการเขียนตัวอักษรต่อท้าย ดังนี้

B สำหรับเลขฐานสอง ค่าของตัวเลขมีได้เพียง 2 ตัว คือ 0 และ 1 เช่น 11001001B

O หรือ Q สำหรับเลขฐานแปด ค่าของตัวเลขมีตั้งแต่ 0-7 เช่น 1267Q หรือ 1267O

D สำหรับเลขฐานสิบ เช่น 4598

H สำหรับเลขฐานสิบหก ค่าของตัวเลขมีตั้งแต่ 0-F เช่น 123AH, 0FE7H จะเห็นได้ว่า ในกรณีที่เขียนข้อมูลเลขฐานสิบหกที่ขึ้นต้นด้วย A-F จะต้องมีเลขศูนย์ (0)ใส่อยู่ข้างหน้าทั้งนี้เพื่อแยกให้เห็นว่า ข้อมูลที่ปรากฏนี้เป็นข้อมูลตัวเลขฐานสิบหกหรือตัวอักษรชื่อลาเบลและชื่อตัวแปร

2. ในการกำหนดชื่อลาเบล ต้องต่อด้วยเครื่องหมาย : (โคลอน) เสมอ ในขณะที่หากใช้คำสั่ง EQU ที่ชื่อตัวแปรไม่ต้องมีเครื่องหมาย : ต่อท้าย

3. การเว้นช่องว่างระหว่างชื่อลาเบล รหัสคำสั่งนีโมนิก โอเปอร์แรนด์ และคอมเมนต์ ควรใช้การกดปุ่ม Tab

4. ในส่วนของคอมเมนต์ หรือส่วนของโปรแกรมที่ไม่ต้องการให้มีการแอสเซมเบลอร์ต้องใส่เครื่องหมาย : หรือเซมิโคลอนนำหน้าส่วนหรือข้อความนั้นเสมอ

การแอสเซมเบลอร์

เมื่อทำการเขียนโปรแกรมภาษาแอสเซมบลีเรียบร้อยแล้ว ขั้นตอนต่อมาคือ ทำการแปลความหายของภาษาแอสเซมบลี นี้เป็นภาษาเครื่องหรือแมชีนโค้ด ซึ่งต้องกระทำผ่านซอฟต์แวร์ โดยใช้โปรแกรมที่เรียกว่า **แอสเซมเบลอร์ (assembler)** ในปัจจุบันมีโปรแกรมแอสเซมเบลอร์ ให้เลือกให้หลายตัว ไม่ว่าจะเป็น CROSS - 32 ซึ่งสามารถใช้แอสเซมเบลอร์ไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์ ได้หลายเบอร์ หรือ SXA51 อันเป็นโปรแกรมแอสเซมเบลอร์ภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ MCS-51 ที่ได้รับความนิยมสูงในประเทศไทย เป็นต้น

หลังจากการแอสเซมเบลอร์ สิ่งที่เกิดขึ้นตามมาคือ ไฟล์ที่มีชื่อเดียวกับไฟล์แอสเซมบลี (นามสกุล .ASM) แต่มีนามสกุล .LST และ .HEX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนไฟล์นามสกุล .LST หรือเรียกว่า ลิสตไฟล์ (List file) จะแสดงถึงรายละเอียดของภาษาแอสเซมบลีและภาษาเครื่องคลอดจนลาเบลและคอมเมนต์ต่างๆ ไว้อย่างชัดเจน เพื่อประโยชน์ในการตรวจสอบและใช้ในการอ้างอิงได้เป็นอย่างดี การแอสเซมเบลอร์ไม่ว่าจะผ่านหรือไม่ ก็จะได้ไฟล์นามสกุล .LST และ .HEX ออกมา แต่ในโปรแกรมแอสเซมเบลอร์บางตัวจะให้ไฟล์นามสกุล .ERR ออกมาด้วยเพื่อแจ้งความผิดพลาด

```
8051 Cross Assembler(1.3)  (C)1987,1989 Binary Technology              page1
sample-1.ASM

                        1
;***************************************************************************
                        2;  Program         : EXAMPLE
                        3;  Description     : EXAMPLE
                        4;  For             : MCS 51 Microcontroller Text book
                        5;  Filename        : XAMPLE1.ASM
                        6;  Assembler       : SXA51
                        7;  Copyright (C)   1999 Innovative Experiment Co.,Ltd
                        8
;***************************************************************************
                        9
0094=                   10   DRIVER_LE     BIT     P1.4            ; Driver Latch Enable
                        11
                        12   ;**************
                        13   ; Main Program.
                        14   ;**************
                        15
0000                    16                  ORG     0000H           ; Reset Vector
0000 0108               17                  AJMP    INITIAL         ; Jump to initial
                        18
0002 758000             19   INITIAL:       MOV     P0,#00000000B   ; Clear Databus
0005 7590FF             20                  MOV     P1,#11_01_11B   ; Clear status
0008 75A0FF             21                  MOV     P2,#11111111B   ; Clear status
000B 75B01F             22                  MOV     P3,#00011111B   ; Clear LCD
                        23
000E D294               24   MAIN:          SETB    DRIVER_LE       ; Enable Driver Latch
0010 7807               25                  MOV     R0,#7           ; Set 7 times loop
0012 7401               26                  MOV     A,#00000001B    ; Initial ACC.
                        27
0014 F500               28   LOOP_LEFT:     MOV     P0,A            ; Out ACC. to DATABUS
0016 1133               29                  ACALL   DELAY_1s            ; Delay 1 s
0018 23                 30                  RL      A               ; Rotate ACC. Left
0019 D8FD               31                  DJNZ    R0,LOOP_LEFT    ; Do loop until
                        32
001B 7807               33                  MOV     R0,#7           ; Set 7 times loop
                        34
001D F500               35   LOOP_RIGHT:    MOV     P0,A            ; Out ACC. to DATABUS
001F 1133               36                  ACALL   DELAY_1s        ; Delay 1 s
0021 03                 37                  RR      A               ; Rotate ACC. Right
0022 D8F9               38                  DJNZ    R0,LOOP_RIGHT   ; Do loop until 7
                        39
0024 010E               40                  AJMP    MAIN            ; Jump to Main
                        41
0026-                   42                  END

driver_le = 0094  initial = 0002  loop_left = 0014  loop_right = 001D main = 000E
```

ภาพที่ 2.19 ตัวอย่างไฟล์นามสกุล .LST หรือลิสต์ไฟล์ที่ได้จากการแอสเซมเบลอร์

การแอสเซมเบลอร์ผ่าน หมายความว่า โปรแกรมภาษาแอสเซมบลีที่นำมาเขียนขึ้นนั้นไม่มีข้อผิดพลาดในหลักไวยกรณ์ไม่มีการใช้คำสั่งผิดรูปแบบ ตัวแปรที่อ้างถึงต่างๆ มีตัวเลขหรือค่าที่สามารถใช้อ้างได้จริง คำสั่งที่ใช้ไม่ผิดรูปแบบ แต่นั่นไม่ได้หมายความว่า โปรแกรมที่เขียนขึ้นสามารถใช้งานได้จริง จึงต้องนำไฟล์นามสกุล .HEX ไปเขียนลงในหน่วยความจำโปรแกรมแล้วทดลองรันให้งานดู จึงสามารถสรุปได้ว่า โปรแกรมที่เขียนขึ้นนั้นสามารถใช้งานได้จริงหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากการแอสเซมเบลอร์พบว่ามีการเขียนโปรแกรมผิดหลักไวยกรณ์หรือผิดรูปแบบโปรแกรม แอสเซมเบลอร์จะแจ้งให้ทราบในทันที และระบุตำแหน่งที่ผิดพลาด รวมทั้งลักษณะของการผิดพลาดด้วย โดยจะแสดงในลิสต์ไฟล์สำหรับโปรแกรมแอสเซมเบลอร์ที่ไม่มีการสร้างไฟล์นามสกุล .ERR ส่วนโปรแกรมแสเซมเบลอร์ที่สร้างไฟล์นามสกุล .ERR สามารถตรวจสอบความผิดพลาดได้ที่ไฟล์นามสกุล .ERR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการและการออกแบบ

3.1 กล่าวนำ

ในการออกแบบแบ่งออกเป็น 3 ส่วนด้วยกันคือ ส่วนควบคุม ส่วนคัดแยกเหรียญ และส่วนทอนเหรียญ โดยแต่ละส่วนที่ออกแบบต้องมีความสัมพันธ์กันและทำงานเข้ากันด้วยดี ในส่วนของการควบคุมนั้นต้องมีการคิดว่าต้องการจะสื่ออะไรออกมาบ้างซึ่งในที่นี้ต้องการให้ มีการควบคุมการแสดงผลเมื่อมีการหยอดเหรียญเข้าไป และสามารถแสดงจำนวนคนที่เข้าไปใช้บริการ มีการควบคุมการทอนเหรียญที่ถูกต้อง โดยมีการรับข้อมูลมาจากชุดตรวจสอบเหรียญแล้วสั่งการให้ชุดทอนทำงาน

ในส่วนของตัวคัดแยกเหรียญนั้นต้องการให้สามารถแยกเหรียญเป็นมูลค่าต่างๆ กันคือ 1 บาท 5 บาท 10 บาท ซึ่งจะใช้การบังคับเหรียญให้ไหลลงมาตามรางที่กำหนดไว้โดยการที่เหรียญจะถูกดักให้ตกลงมาตามช่องที่ได้เจาะรูไว้ซึ่งแบ่งไว้ 2 ขนาดคือ ช่องเหรียญ 1 บาท และช่องเหรียญ 5 บาท ส่วนของช่องเหรียญ 10 บาทนั้นไม่มีเพราะเหรียญ 10 มีขนาดที่ใหญ่ที่สุดจึงกำหนดให้ไหลผ่านช่องเหรียญ 1 บาท เหรียญ 5 บาท และอีกประการหนึ่งคือ เหรียญ 10 บาท ไม่จำเป็นต้องมีการสำรองเหรียญไว้ใช้ในการทอนเหรียญแก่ผู้ใช้บริการเพราะเหรียญ 10 บาทมีมูลค่าสูงสุดในการใช้งาน

ในส่วนของตัวทอนเหรียญ จะใช้สเต็ปมอเตอร์เป็นตัวจัดการในการทอนเหรียญโดยใช้การแกว่งของสเต็ปมอเตอร์ไปมาทั้ง 2 ข้างเป็นตัวปัดเหรียญที่ถูกสำรองรองไว้ในช่องเก็บเหรียญให้ตกลงมาที่ช่องรับเงินทอน โดยการควบคุมจากไมโครคอนโทรลเลอร์ที่ทำหน้าที่ประมวลผลและส่งผลที่ได้จากการนับพัลส์สัญญาณมาที่สั่งการให้สเต็ปมอเตอร์หมุนซ้ายหรือขวา เพื่อปัดให้เหรียญที่ต้องการทอนตกลงมาที่ช่องรับเงินทอน โดยสเต็ปเตอร์นี้จะใช้ชุดขับสเต็ปที่เป็นไอซีเบอร์ ULN 2003 ซึ่งมีความสามารถที่จะขับกระแสได้ถึงขาละ 500 มิลลิแอมป์ เป็นชุดขับสเต็ปมอเตอร์ สเต็ปมอเตอร์นี้มีขนาดแรงดันที่ใช้ที่ 24 โวลท์ ที่ไฟฟ้ากระแสตรง ในการทำชุดทอนเหรียญนี้ต้องทำให้มีความสัมพันธ์กับชุดที่มีการสำรองเหรียญไว้เพื่อทอนเหรียญแก่ผู้ใช้บริการ

จากทั้ง 3 ส่วนที่กล่าวมาแล้วต้องมีการวางแบบแผนที่สัมพันธ์กันโดยจะเห็นได้ว่าทุกส่วนมีความเกี่ยวเนื่องซึ่งกันและกันถ้าขาดส่วนใดส่วนหนึ่งไปหรือว่าส่วนที่ต้องทำต่อเนื่องนั้นไม่สามารถทำหน้าที่ตามที่ได้คาดหมายไว้ จะทำให้ส่วนที่ต่อเนื่องต่างๆ เกิดความผิดพลาด และล้มเหลว ดังตัวอย่างเช่น ถ้าส่วนที่ตรวจสอบเหรียญนั้นเกิดความผิดพลาดคือ มีการส่งสัญญาณออกมาอย่างไม่ถูกต้อง เช่นพัลส์ที่ออกมาไม่ครบทำให้ส่วนควบคุมที่ทำหน้าที่ตรวจนับพัลส์ทำการนับพัลส์สัญญาณไม่ถูกต้องก็จะส่งผลให้ส่วนแสดงผล แสดงผลออกมาผิดพลาดส่งผลไปให้ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมการทอนเหรียญเกิดความผิดพลาดตามมา เพราะว่าส่วนการทอนเหรียญนี้จะอ้างอิงการทำงานกับส่วนตัวตรวจนับสัญญาณพัลส์ ทำให้เกิดการทอนที่ผิดพลาด ซึ่งเป็นผลเสียต่อผู้ประกอบการ เพราะทำให้เสียผลประโยชน์ที่ควรจะได้ไป หรือถ้าส่วนตัวคัดขนาดเหรียญเกิดการผิดพลาดคือมีเหรียญไหลลงไปผิดตำแหน่งที่ควรจะเป็น โดยมากมักจะเกิดขึ้นจากความผิดพลาดที่เหรียญ 1 บาท กับเหรียญ 5 บาท เพราะเหรียญ 1 บาท มี อัตตาการไหลลงที่เร็วประกอบกับเหรียญมีขนาดเล็กอาจทำให้เหรียญ 1 บาท ไหลเลยช่องคัดขนาดเหรียญไหลลงไปรวม กับเหรียญ 5 บาทซึ่งผลที่ได้คือทำได้ความน่าเชื่อถือเกี่ยวกับตัวเครื่องนี้ลดลงถ้าผู้บริการเกิดเจอปัญหานี้ขึ้น แล้วจะทำให้เครื่องหยอดเหรียญสาธารณะนี้ ไม่สามารถทดแทนระบบที่ใช้บุคคลในการจัดการเก็บเงินได้ดังนั้นผู้ประกอบการก็จะไม่เลือกใช้เครื่องหยอดเหรียญนี้และจะหันกลับไปจัดสรรเลือกหาบุคลากรมาแล้วจะเกิดปัญหาการยักยอกค่าบริการ

3.2 หลักการออกแบบส่วนคัดแยกเหรียญ

หลักการที่ใช้ในการคัดแยกเหรียญนี้ใช้หลักการให้เหรียญที่หยอดลงมาตกลงมาตามรางที่ลาดเอียงซึ่งทำ มุม 20 องศาโดยรางที่ออกแบบนี้จะมีขนาดใหญ่กว่าเหรียญ 10 บาท เล็กน้อยซึ่งโดยปกติแล้วเหรียญ 10 บาท จะมีขนาดกว้าง 3.45 เซนติเมตรซึ่งจะเผื่อรางไว้ 1 มิลลิเมตร เหรียญที่ไหลลงมาตามรางจะถูกบังคับโดยรางไม่ให้ไหลไปทางอื่นอย่างอิสระ การทำเช่นนี้จะทำให้การบังคับเหรียญมีความแน่นอนยิ่งขึ้นทำให้แน่ใจได้ว่าเหรียญจะไม่มีทางไหลไปทางอื่นได้ ระหว่างทางนั้นก็จะมีช่องขนาดต่างๆ เพื่อคัดเลือกเหรียญ 1 บาท และเหรียญ 5 บาท โดยช่องเหรียญบาทจะมีขนาดใหญ่เท่ากับขนาดของเหรียญ 5 บาท คือมีขนาดเท่ากับ 2.60 เซนติเมตร การทำเช่นนี้เป็นการดักให้เหรียญ 1 บาท ตกลงมาในช่องนี้เพียงเหรียญเดียว ที่ว่าเพียงเหรียญเดียวเพราะว่าจากการเคลื่อนที่ของวัตถุในแนวเอียงนั้นจะมีแรงเฉื่อยเกิดขึ้นจะขอกล่าวถึงหลักการส่วนนี้ หลักการของ **"การเคลื่อนที่"** (แปลตรง ๆ จาก motion) ฟังดูแปลก ๆ แต่ก็ตรงตัว ถ้าจะเรียกให้ดูดีก็เรียกกันว่า **"กลศาสตร์"** (mechanics) ทั้งสองคำนี้คงไม่ต่างกันมากนัก การศึกษาเกี่ยวกับการ เคลื่อนที่นี้จะได้ศึกษาถึงหัวข้อพื้นฐานเช่น เรื่องของ ความเร่ง ความเร็ว เป็นต้น แต่ก็ยังมีบางเรื่องที่ซับซ้อนมากขึ้นไปที่ไม่ได้เฉกกล่าวถึงในกฎของนิวตัน เช่น เรื่องของ งาน แรงลัพธ์ ซึ่งจะช่วยให้รู้หลักการได้บ้าง

หากพิจารณาวัตถุสสารใด ๆ จะเห็นว่า บางวัตถุมีความหนาแน่นของเนื้อวัตถุมาก เช่น เหล็ก หิน บางวัตถุมีความหนาแน่นน้อย เช่น โฟม พลาสติก ทำให้มีน้ำหนักเบา **มวลสาร (Mass)** จึงเป็นปริมาณที่จะบอกคุณสมบัติของวัตถุ และถ้ามีแรงมากระทำต่อวัตถุพื้น ก็จะเกิดสภาพการต่อต้านสภาวะการเคลื่อนที่ เช่น ถ้าออกแรงผลักวัตถุที่มีมวลสารหนาแน่น ก็ต้องออกแรงมาก **แรง (Force)** คือปริมาณทางฟิสิกส์ที่กระทำต่อวัตถุ แล้วจะทำให้วัตถุนั้นเคลื่อนที่ด้วยความเร่งแรงเป็นเรื่องที่ใหญ่ที่สุดในฟิสิกส์ แรงทำให้เกิดพลังงาน แรงก็คืออะไรก็ตามที่มีผลกระทำต่อวัตถุ ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณคือลูกบอลอยู่ตรงนั้นแหละ แล้วมีคนมาเตะคุณ คุณก็จะกระเด็นออกไปเพราะมีแรงมากระทำต่อคุณ ในเรื่องของเวกเตอร์นั้น แรงทุก ๆ แรงสามารถถูกทำให้เห็นโดยการเขียนเวกเตอร์แทนได้ คราวนี้มาดูสมการกัน **"F = m * a"** สมการนี้สำคัญมา **"F"** คือปริมาณของ แรง **"m"** คือมวล และ "a" คือความเร่งที่เกิดขึ้น จะได้เป็นประโยคว่า "แรงที่กระทำ เท่ากับ มวล คูณด้วย ค่าของความเร่ง แล้วก็เหมือน ๆ กับเวกเตอร์ ถ้ามีแรงหลาย ๆ แรงก็บวกด้วยวิธีการเดียวกับการบวกเวกเตอร์ และค่าที่ใช้ก็คือ "Newton" เป็นปริมาณของแรง

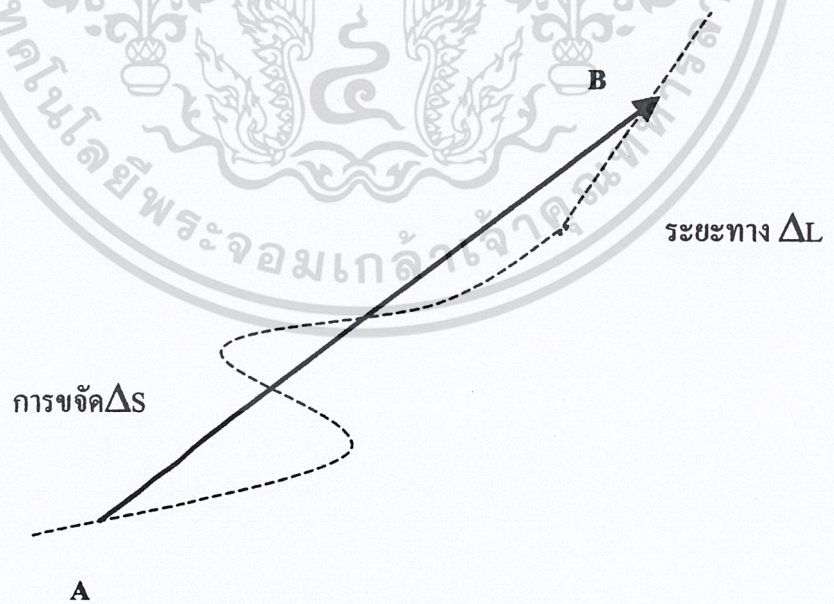
น้ำหนัก (weight) เมื่อวัตถุอยู่ภายใต้แรงดึงดูดของโลก จะมีแรงดึงดูดที่ทำให้วัตถุตกจากที่สูง และเคลื่อนที่เข้าสู่ศูนย์กลางของโลก แรงของโลกที่ดึงดูดมีค่าเท่ากับ W = mg

การเคลื่อนที่ คือ การย้ายที่อยู่จากที่หนึ่งไปยังอีกที่หนึ่ง ซึ่งมีผลทำให้เกิดการเคลื่อนที่ 2 ปริมาณคือ ระยะทาง และการขจัด

3.2.1 ระยะทางและการขจัด

ระยะทาง คือ เส้นทางที่วัตถุเคลื่อนที่ไม่ได้จริง ระยะทางเป็นปริมาณสเกลาร์ (scalar) มีแต่ขนาดไม่มีทิศทาง มีหน่วยเป็นเมตร

การขจัด คือ เส้นตรงที่ลากจากจุดเริ่มต้นไปยังจุดสุดท้ายของการเคลื่อนที่ การขจัดเป็นปริมาณเวกเตอร์ (vector) มีทั้งขนาดและทิศทาง โดยมีทิศที่พุ่งจากจุดเริ่มต้นไปยังจุดสุดท้ายของการเคลื่อนที่



ภาพที่ 3.1 แสดงระยะทางและระยะขจัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปวัตถุเคลื่อนที่ไปตามแนวเส้นประจาก A ไปยัง จะได้ว่า

ΔL = ระยะทางเส้นประจาก A ไป B

ΔS = การขจัดใช้เป็นเส้นตรง มีลูกศรชี้จุดสุดท้าย

3.2.2 ความเร็วและอัตราเร็ว

เมื่อมีการเคลื่อนที่จะเกิดปริมาณการวัดได้ทันที 3 ปริมาณ คือ ระยะทางΔL การขจัดΔS และช่วงเวลา Δt ที่ใช้ในการเคลื่อนที่นักฟิสิกส์กำหนดปริมาณใหม่ซึ่งเกิดจากการทำเวลา Δt ไปหาระยะทางและการขจัดดังนี้

อัตราเร็ว คือ ระยะทางต่อหน่วยเวลาหรือระยะทางที่วัตถุเคลื่อนที่ได้ในช่วงเวลา 1 วินาที เขียนเป็นสมการได้ว่า

อัตราเร็ว = ระยะทาง / เวลา

อัตราเร็วเป็น ปริมาณสเกลาร์มีหน่วยเป็น เมตร / วินาที (m/s) ทั้งนี้เพราะว่าระยะทางและเวลา เป็นสเกลาร์ผลหารจึงต้องเป็นสเกลาร์ด้วยและมีหน่วยตามระยะทางกับเวลา

ความเร็ว คือการขจัดต่อหนึ่งหน่วยเวลาหรือการขจัดที่วัตถุเคลื่อนที่ได้ในช่วงเวลา 1 วินาที ความเร็วก็เป็นพียงอีกคำหนึ่งของอัตราเร็ว เพียงแต่ว่ามันบอกทิศเราได้ด้วย ถ้าสมมุติว่าเดินไปข้างหน้า 30 กิโลเมตรใช้เวลาชั่วโมงหนึ่ง ก็แสดงว่า ผมมีความเร็ว 30 กิโลเมตรต่อชั่วโมง ซึ่งก็คือค่าของระยะทางหารด้วยเวลาที่ใช้ไปนั่นเอง ความเร็วจะคงที่ หรือเปลี่ยนแปลงได้ (ความเร่ง) มีคำอยู่หนึ่งคำครับ "ความเร็ว ณ ขณะใดขณะหนึ่ง" ก็คือความเร็วในชั่วเซี่ยวของวินาที คำว่า " ณ ขณะใดขณะหนึ่ง" นั้นเป็นการกำหนดช่วงเวลา จะใช้คำว่า "ลิมิต" (Limits แปลตรง ๆ ว่า ขอบเขตเห็นจะไม่ผิดนัก) เมื่อลิมิตของเวลาเข้าใกล้ศูนย์ (แต่ก็ไม่ใช่ศูนย์ แต่ก็น้อยมาก ๆ) จึงจะใช้วัดความเร็ว ณ ขณะใดขณะหนึ่งหนึ่งได้เขียนเป็นสมการได้ว่า

ความเร็ว = การขจัด / เวลา

ความเร็วเป็น ปริมาณเวกเตอร์มีหน่วยเป็น เมตร/วินาที (m/s) เพราะว่าการขจัดเป็นปริมาณเวกเตอร์ เมื่อหารด้วยเวลาที่เป็นปริมาณสเกลาร์ จะได้ผลลัพธ์ที่เป็นเวกเตอร์ มีทิศทางเดียวกับการขจัดในการคำนวณทางฟิสิกส์ ถ้าคิดการย้ายที่จากจุดหนึ่งไปยังอีกจุดหนึ่งให้ถือว่า ช่วงเวลาเคลื่อนที่ยาวนาน แต่ถ้าคิด ณ จุดหนึ่งบนทางเคลื่อนที่ก็ถือว่า ช่วงเวลาสั้นมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร่งเมื่อความเร็วเปลี่ยน เราก็จะได้ใช้คำว่าความเร่งก็คราวนี้แหละ ค่าของความเร่งที่เป็นบวกก็คือเวลาที่คุณเพิ่มความเร็ว และค่าของความเร่งที่เป็นลบก็แสดงว่าคุณลดความเร็วลงและก็เช่นกันกับคาวมเร็ว มีคำที่กล่าวว่า "ความเร่ง ณ ขณะใดขณะหนึ่ง" ด้วยความหมายลักษณะเดียวกัน

แรงเสียดทานก็คือแรงที่คอยต้านการเคลื่อนที่ของวัตถุ ก็แค่นั้นเอง มันมีอยู่ทั่วไปที่ ๆ วัตถุต่างสัมผัสกัน มันมีทิศที่ตรงกันข้ามกับทิศที่วัตถุจะเคลื่อนที่ไป ถ้าคุณวิ่ง ๆ ไปตามทางเท้าแล้วก็หยุดคุณคงพบว่ามนหยุดไม่ยากนักเพราะมีแรงเสียดทานระหว่างรองเท้าคุณกับพื้นแต่ถ้าวิ่งไปแล้วหยุดบนน้ำมันล่ะ แน่นอนแรงเสียดทางก็ยังคงอยู่แต่เพราะน้ำมันแรงนั้นก็น้อยมาก และมันคงยากขึ้นที่จะหยุด ลักษณะนี้ก็เกิดขึ้นกับรถเช่นเดียวกันเมื่อมีฝน และนี่คือเหตุผลที่ทำให้เกิดอุบัติเหตุมากมาย แรงเสียดทานในอากาศให้เดาว่ากำลังนึกถึงของแข็งเท่านั้น หยุดก่อนเพราะสิ่งที่คุณต้องการก็แค่พียงสสารเท่านั้นฉะนั้น สถานะไหนก็ได้ เช่นแรงลมก็ใช่ ถ้าคุณอยู่ในยานอวกาศแล้วกำลังเข้าสู่ชั้นบรรยากาศโลก ก้นของยานนั้นจะร้อนมาก ความร้อนนั้นเกิดขึ้นมาจากแรงเสียดทานระหว่างตัวยานกับอากาศ เพราะโมเลกุลของอากาศได้ถูไถไปตามยานทำให้เกิดความร้อนขึ้น

3.2.3 อัตราเร็วเฉลี่ยและความเร็วเฉลี่ย

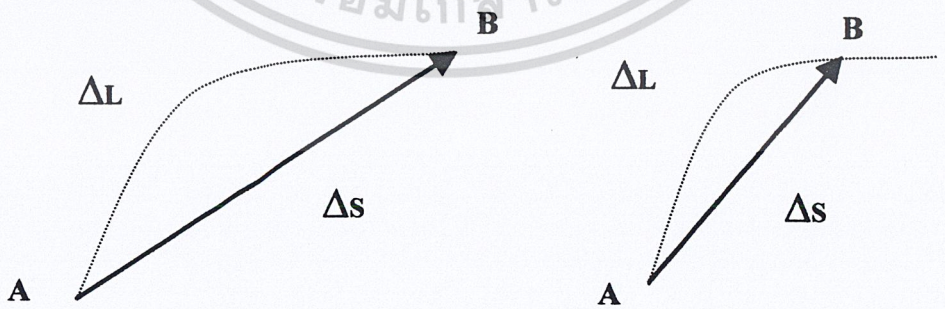
ถ้าวัตถุเคลื่อนที่ตามแนวเส้นประจาก A ไปยัง B ในช่วงเวลา t ที่ยาวนานดังนั้นเขียนเป็นสมการอัตราเร็วเฉลี่ย ความเร็วเฉลี่ยเขียนเป็นสมการได้ดังนี้

$$V_{av} = \Delta L / \Delta t$$

$$V_{av} = \Delta S / \Delta t$$

3.2.4 อัตราเร็วขณะใดขณะหนึ่งและความเร็วขณะใดขณะหนึ่ง

ดังที่ได้ล่าวแต่ตอนต้นว่า Δt เป็นช่วงเวลาที่ยาวนานที่วัตถุใช้เคลื่อนที่จาก AB ถ้าพิจารณาการเคลื่อนที่ในช่วงเวลา Δt สั้นลง ดังรูป



เวลาที่ยาวนาน เวลาสั้นลงทำΔS เข้าใกล้ΔL

ภาพที่ 3.2 แสดงค่า ΔS เข้าใกล้ΔL เมื่อเวลาเปลี่ยนแปลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 ความเร่ง

เมื่อนักฟิสิกส์กำหนดความเร็ว ณ จุดหนึ่งบนทางเคลื่อนที่ทำให้แนวคิดต่อไปว่า ความเร็ว ณ จุดถัดไปความเร็วเปลี่ยนไปอย่างไรจึงได้มีการกำหนดปริมาณใหม่ขึ้นมาว่า ความเร่ง ความเร่ง คือ ความเร็วที่เปลี่ยนไปต่อหนึ่งหน่วยเวลา หรือความเร็วที่เปลี่ยนไปในช่วงเวลา 1 วินาที เขียนเป็นสมการได้ว่า

ความเร่ง = ความเร็วที่เปลี่ยนไป / เวลา

ความเร่งเป็นปริมาณเวกเตอร์ มีทิศทางเดียวกับความเร็วที่เปลี่ยนไปความเร่งมีหน่วยเป็น เมตร / วินาที^{2} โดยถ้าความเร็วของวัตถุ ลดลง จะเรียกความเร่งนี้ว่า ความหน่วง แต่ถ้าความเร็วของวัตถุ เพิ่มขึ้น จะเรียกความเร่งนี้ว่า ความเร่ง เนื่องจากช่วงเวลาที่ใช้ในการเคลื่อนที่มี 2 แบบ คือ ช่วงเวลาที่ยาวนาน และช่วงเวลาที่สั้นมากจึงแบ่งความเร่งได้ 2 แบบ ตามช่วงเวลาเคลื่อนที่

3.2.6 สมการการเคลื่อนที่ในแนวเส้นตรง

สมมติว่าวัตถุเคลื่อนที่เป็นแนวเส้นตรงจาก A ไป B ด้วยความเร็วต้น u ความเร็วปลาย v และความเร่งคงที่สัมผัสเส้นทางเคลื่อนที่เท่ากับ $a_T = a$ ในช่วงเวลา t ได้การขจัด s จึงสรุปได้ว่า การเคลื่อนที่ในแนวเส้นตรงด้วยความเร่งคงที่ ขนาดคงที่ มีทิศที่สัมผัสทางเคลื่อนที่ตลอดทางนั้น มีสมการการเคลื่อนที่เหมือนกัน

$$v = u + at$$

$$s = ut + 0.5\,at^2$$

$$v^2 = u^2 + 2as$$

$$s_t = u + 0.5\,a(2t - 1)$$

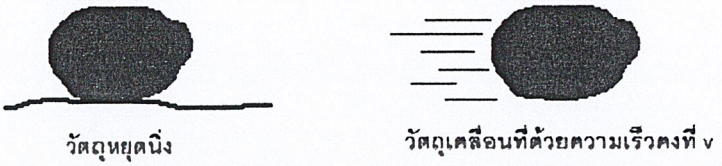
3.2.7 กฎการเคลื่อนที่ 3 ข้อของนิวตัน

เซอร์ไอแซคนิวตัน นักฟิสิกส์ชาวอังกฤษ ผู้มีผลงานโดดเด่นหลายเรื่อง โดยเฉพาะเรื่อง การคำนวณระหว่างแสงกับมวลสาร เขาได้อธิบายว่า ทำไมดวงจันทร์จึงโคจรรอบโลก และทำไม โลกจึงโคจรรอบดวงอาทิตย์ แรงที่กระทำต่อวัตถุที่เกิดจากแรงโน้มถ่วงของโลก และทำให้วัตถุ ตกจากที่สูงเคลื่อนที่อัตราเร่ง g นิวตันให้หลักการคำนวณที่เกิดจากแรงไว้ 3 ข้อ และเป็นกฎที่สำคัญ ในการอธิบายหลักทางฟิสิกส์ได้อย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎข้อที่หนึ่ง Law of Inertia (กฎของความเฉื่อย)

ถ้าวัตถุอยู่ในสภาพนิ่ง ก็จะรักษาสภาพนิ่ง ถ้าวัตถุอยู่ในสภาพการเคลื่อนที่ด้วยความเร็วคงที่ ก็จะเคลื่อนที่เช่นนั้น จนกว่าจะได้รับแรงจากภายนอกมากระทำต่อวัตถุนั้น



วัตถุอยู่ในสภาวะสมดุล

ภาพที่ 3.3 แสดงวัตถุที่เป็นไปตามกฎข้อที่ 1

กฎข้อที่สอง F = ma

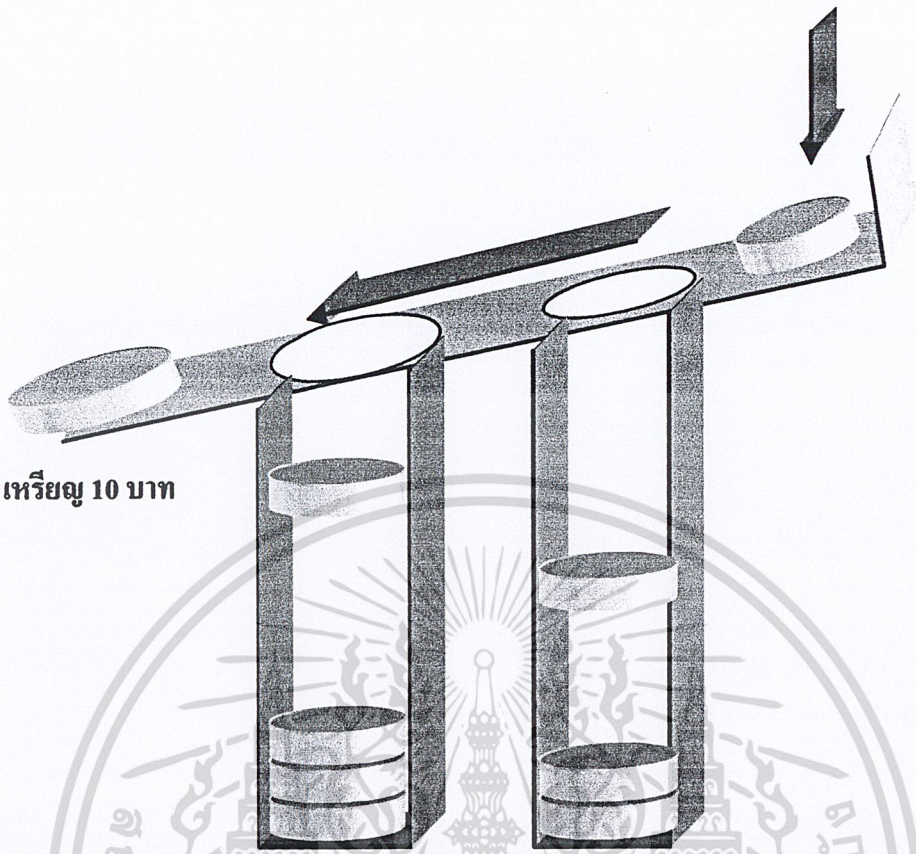
เมื่อมีแรงมากระทำต่อวัตถุ และผลรวมของแรงนั้นไม่เป็นศูนย์ จะทำให้วัตถุนั้นเคลื่อนที่ในทิศทางเดียวกับแรงลัพธ์ ความเร่งจะมีขนาดแปรผกผันกับมวลของวัตถุ

ภาพที่ 3.4 แสดงวัตถุที่เป็นไปตามกฎข้อ 2

กฎข้อที่สาม

เมื่อมีแรงกระทำเป็นแรงกริยาทุกแรงต้องมีแรงปฏิกิริยา ซึ่งมีขนาดเท่ากัน และทิศทางตรงข้าม ลองคิดถึงเกมพินบอล เมื่อลูกบอลไปชนกับ หลัก มันจะเด่งออกมาด้วยแรงในทิศตรงกันข้าม นำหลักการที่ได้ไปออกแบบทำตัวคัดแยกเหรียญโดยมีการเจาะช่องเพื่อใช้ในการแยกเหรียญ 1 บาท 5 บาท และ 10 บาท ให้ได้ตามต้องการ แล้วต้องมีการเผื่อขนาดไว้เล็กน้อยเพื่อลดความฝืดจากการที่เหรียญตกลงมาตามราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.5 แสดงการออกแบบตัวคัดแยกเหรียญ

จากรูปเราจะเห็นได้ว่าเหรียญที่ตกลงมาจะถูกบังคับให้ไหลตามรางลงมา เหรียญจะค่อยๆ ไหลลงมาตามรางที่กำหนดไว้ผ่านช่องคัดขนาดเหรียญที่มีขนาดที่แตกต่างกัน โดยเหรียญ 1 บาทจะถูกเก็บลงในช่องแรกโดยที่เหรียญ 1 บาท จะต้องไม่มีทางไหลลงมาที่ช่องเหรียญ 5 บาทโดยเป็นอันขาดเพราะจะทำให้เกิดการเก็บเหรียญที่ผิดพลาดได้แล้วจะทำให้เกิดการทอนที่ผิดพลาด ส่วนเหรียญ 5 บาท จะผ่านช่องเหรียญ 1 บาทมาแล้ว มาไหลลงตรงช่องเหรียญ 5 บาท และไหลลงมารวมกันและเก็บไว้เป็นตัวทอนเหรียญให้กับผู้ใช้บริการโดยมีการยอมให้ เหรียญ 5 บาทมีโอกาสที่จะตกเลยช่องเหรียญ 5 บาทได้เพราะว่าถ้าเลยช่องเหรียญ 5 บาทแล้วจะเป็นช่องเก็บเหรียญรวม คือมีทั้งเหรียญ 1 บาท เหรียญ 5 บาท และเหรียญ 10 บาทมารวมกันที่นี่ ส่วนเหรียญ 10 บาทนั้นจะต้องไหลตามรางลงมาโดยผ่านช่องคัดเหรียญ 1 บาท และเหรียญ 5 บาทลงมาโดยไม่ติดขัดใดๆ

แต่ปัญหาที่จะพบตามมาอีกอย่างคือ ว่าถ้าเกิดเหรียญล้นเต็มความจุของชุดสำรองเก็บเหรียญไว้ทอนจะทำอย่างไร เพราะเกิดปัญหาแน่นอนวิธีทางแก้ไขคือต้องสร้างชุดสำหรับให้เหรียญล้นออกมาซึ่งในการทำจะทำเพียง 1 ชุดเท่านั้นคือชุดสำหรับเหรียญ 1 บาท โดยสำหรับเหรียญ 5 บาทนั้นไม่ต้องทำเพราะเหรียญ 5 บาทเมื่อเต็มช่องที่ใส่เก็บสำรองเหรียญแล้ว เหรียญมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการใช้

จะล้นออกมาเองโดยไม่มีปัญหาใดๆ เกิดขึ้น แต่ถ้าเป็นเหรียญ 1 บาทแล้ว จะเกิดปัญหาอย่างแน่นอนที่สุดเพราะเมื่อเหรียญเต็มช่องแล้วเหรียญล้นเลยไปยังช่องเก็บเหรียญ 5 บาท ทันทีทำให้เกิดการผิดพลาดในการทอนได้ดังนั้นปัญหาอยู่ที่ช่องเหรียญ 1 บาท

วิธีการแก้ไขคือ ต้องเจาะช่องที่ใช้เก็บเหรียญให้มีช่องว่างเพื่อที่ว่าเมื่อเหรียญล้นขึ้นมาจนถึงระดับชิองที่ได้เจาะไว้แล้วนั้น เหรียญก็จะล้นผ่านช่องนั้นออกมาทำให้ไม่มีโอกาสที่เหรียญจะล้นขึ้นไปแล้วตกลงช่องใส่เหรียญ 5 บาท

ภาพที่ 3.6 แสดงวิธีให้เหรียญล้นออกมา

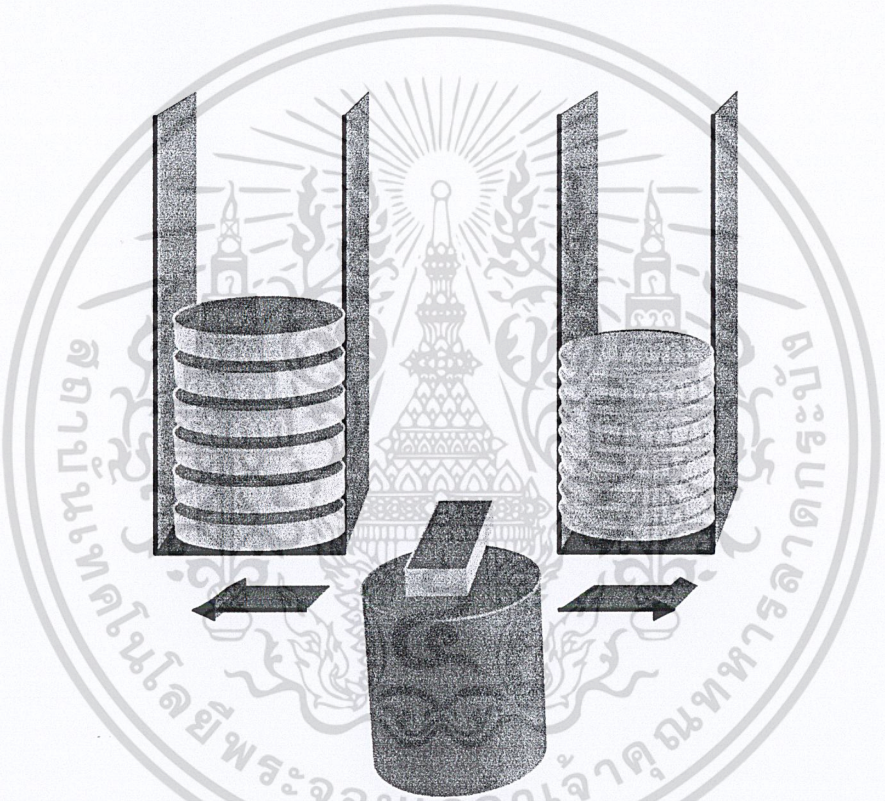
จากภาพเมื่อเหรียญเต็มช่องแล้วเหรียญจะกระทบกับมุมทำให้เหรียญที่เกินล้นออกมาข้างนอกไม่ตกรวมไปที่ช่องเหรียญ 5 บาท จึงเป็นการแก้ปัญหานี้ได้โดยไม่ส่งผลกระทบต่อระบบต่างๆ ที่ต่อเนื่องกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 หลักการออกแบบชุดทอนเหรียญ

การทอนเหรียญเป็นสิ่งที่มีความจำเป็นในปริญญานิพนธ์นี้ เพราะการที่รับเหรียญหลายๆ เหรียญเข้ามาได้จำเป็นที่จะต้องมีการทอนเหรียญให้แก่ผู้ใช้บริการ และที่สำคัญการทอนเหรียญให้ผู้ใช้บริการต้องไม่มีข้อผิดพลาด เพราะถ้าเกิดความผิดพลาดขึ้น ต้องต้องเกิดความสูญเสียระหว่างผู้ให้บริการและผู้ใช้บริการ

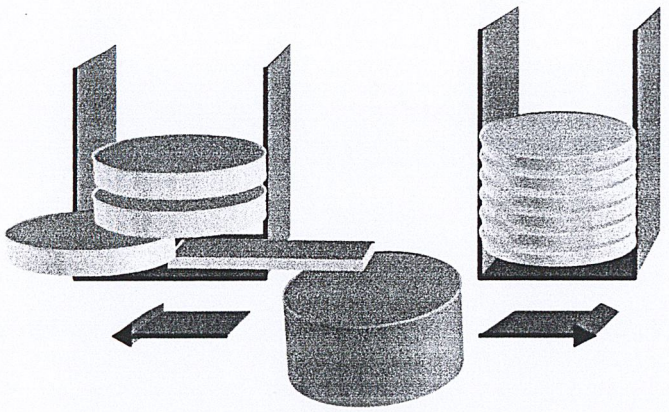
การทอนนี้จะใช้สเต็ปมอเตอร์เป็นหลักในการขับเคลื่อนให้เหรียญถูกปัดตกลงมาจากช่องเก็บสำรองเหรียญโดยใช้การแกว่งของแขนที่ต่อกับสเต็ปมอเตอร์แกว่งโดยถูกควบคุมด้วยไมโครคอนโทรลเลอร์ ทำทำตามที่ต้องการได้อย่างถูกต้องและแม่นยำ



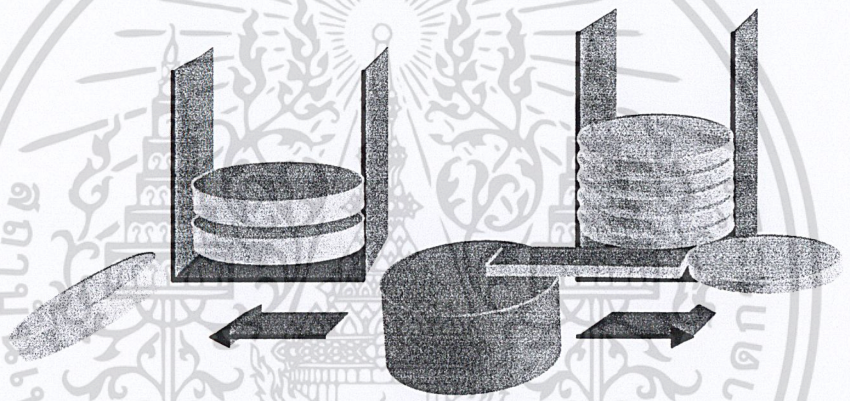
ภาพที่ 3.7 แสดงการวางตำแหน่งของสเต็ปมอเตอร์

จากภาพจะเห็นได้ว่ามีการวางตำแหน่งของสเต็ปมอเตอร์อยู่ตรงกลางระหว่าง ช่องเก็บเหรียญ 1 บาท และช่องเก็บเหรียญ 5 บาท สเต็ปมอเตอร์จะเป็นตัวหมุนให้แกนหมุนไปมาทางซ้ายและขวาปัดเหรียญให้ตกลงมาตามรางซึ่งขึ้นอยู่ว่าไมโครคอนโทรลเลอร์จะสั่งการควบคุมอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 แสดงการปัดเหรียญ 5 บาท



ภาพที่ 3.9 แสดงการปัดเหรียญ 1 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
สรุปผลการวิจัยและข้อเสนอแนะ

4.1 บทสรุป

การทำปริญญานิพนธ์นี้เป็นต้นแบบในการพัฒนาผลงานทำให้สามารถนำไปประยุกต์ใช้ในชีวิตประจำวัน โดยสามารถทดแทนทรัพยากรมนุษย์และต้องสามารถทำงานได้อย่างถูกต้อง แม่นยำ ไม่มีข้อผิดพลาด ซึ่งในสิ่งเหล่านี้มนุษย์อาจมีความผิดพลาดเกิดขึ้นได้ และในปัจจุบันนี้สิ่งอำนวยความสะดวกต่างๆ ก็เป็นปัจจัยหนึ่งที่มนุษย์จะขาดเสียมิได้

ปริญญานิพนธ์ฉบับนี้กล่าวถึงการทำงานของกลไก 3 ส่วนหลักคือ ส่วนควบคุม ส่วนตรวจสอบ และส่วนทอนเหรียญโดยทั้ง 3 ส่วนต้องทำงานสัมพันธ์กันจะมีส่วนใดส่วนหนึ่งไม่ทำงานมิได้ เพราะถ้าขาดส่วนใดส่วนหนึ่งไป ส่วนที่เหลือก็จะทำงานมิได้ ดังนั้นความถูกต้องแม่นยำเป็นสิ่งที่จำเป็น ในการทำปริญญานิพนธ์ฉบับนี้มีปัญหาต่างๆ มากมายหลายประการ ปัญหาบางอย่างก็แก้ไขได้อย่างง่าย แต่ปัญหาบางอย่างต้องใช้เวลาในการแก้ไข ไม่สามารถที่จะแก้ไขได้ทันที จึงถือว่าการทำปริญญานิพนธ์ฉบับนี้ให้หลายสิ่งหลายอย่างในการดำเนินชีวิตในยุคโลกไร้พรมแดนได้อย่างดี เนื่องจากโครงงานนี้ ต้องใช้ความคิดตั้งแต่ต้นในการวิเคราะห์ปัญหาที่อาจจะเกิดขึ้น แล้วนำปัญหาที่ได้ไปประเมินว่าควรหรือไม่ควรทำโครงงานนี้ จากนั้นก็นำไปสู่การออกแบบชิ้นงานให้มีความสอดคล้องและสัมพันธ์กัน จากนั้นก็มีการทดสอบการทำงานว่าสามารถทำงานได้ตามเป้าหมายที่กำหนดไว้ จากนั้นจึงสรุปผลที่ได้ออกมา หาสาเหตุของปัญหาที่เกิดขึ้นระหว่างกระบวนการต่างๆ พร้อมเสนอวิธีการแก้ไขปัญหาที่เกิดขึ้น จึงถือได้ว่าเป็นกระบวนการทางวิทยาศาสตร์ที่สมบูรณ์

การทำงานของตัวเครื่องเริ่มจากที่มีผู้ใช้บริการหยอดเหรียญเข้ามาในตัวระบบ ระบบก็จะแสดงผลมูลค่าของเหรียญออกมาเพื่อให้ผู้ใช้บริการรู้ว่าได้หยอดเหรียญมูลค่าเท่าไหร่ลงไป ซึ่งเป็นการเตือนความจำให้แก่ผู้ใช้บริการ จากนั้นเหรียญจะผ่านลงมายังรางรองรับเหรียญผ่านไปที่ตัวคัดแยกเหรียญเพื่อคัดขนาดเหรียญว่าเป็นเหรียญ 1 บาท เหรียญ 5 บาท และเหรียญ 10 บาท แล้วทำการเก็บเหรียญสำรองไว้เพื่อการทอน โดยเก็บเฉพาะเหรียญ 5 บาทและเหรียญ 1 บาท ส่วนเหรียญ 10 บาท จะถูกปล่อยให้ลงไปในช่องเก็บเงินรวม ในขณะเดียวกันนั้นไมโครคอนโทรลเลอร์ ก็จะตรวจว่าเหรียญที่ผ่านลงมามีมูลค่าเท่าใด แล้วสั่งให้สเต็ปมอเตอร์เป็นตัวทอนเหรียญออกมา ที่ช่องทอนเหรียญ ก็เป็นอันว่าเสร็จสิ้นกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ข้อเสนอแนะและแนวทางการพัฒนา

การทำปริญญานิพนธ์ครั้งนี้ยังมีปัญหาอีกมากมายในการทำงานคือ เรื่องของการหยอดเหรียญถ้ามีการหยอดเหรียญเร็วเกินไป จะทำให้ตัวนับจำนวนพัลส์เกิดการนับที่ซ้อนกันทำให้ตัวแสดงผล แสดงออกมาผิดพลาดและส่งผลให้ตัวทอนทำงานผิดพลาดด้วย ส่วนวิธีแก้ไขคือต้องเปลี่ยนจากการนับพัลส์มาเป็นการนับแบบสัญญาณอนาลอค แปลงมาเป็นดิจิตอล แล้วใช้การเปรียบเทียบเอาก็จะได้การนับที่เร็วละถูกต้องมากกว่า

อีกประการหนึ่งคือเรื่องของการคัดแยกเหรียญต้องทำให้มีการผิดพลาดน้อยกว่านี้ จะมีโอกาสเกิดความผิดพลาดน้อย หรือไม่มีเลย

และสุดท้ายนี้คิดว่าควรที่จะมีผู้สนใจพัฒนาโครงงานนี้เพื่อที่จะได้สามารถนำไปใช้งานจริงในชีวิตประจำวันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

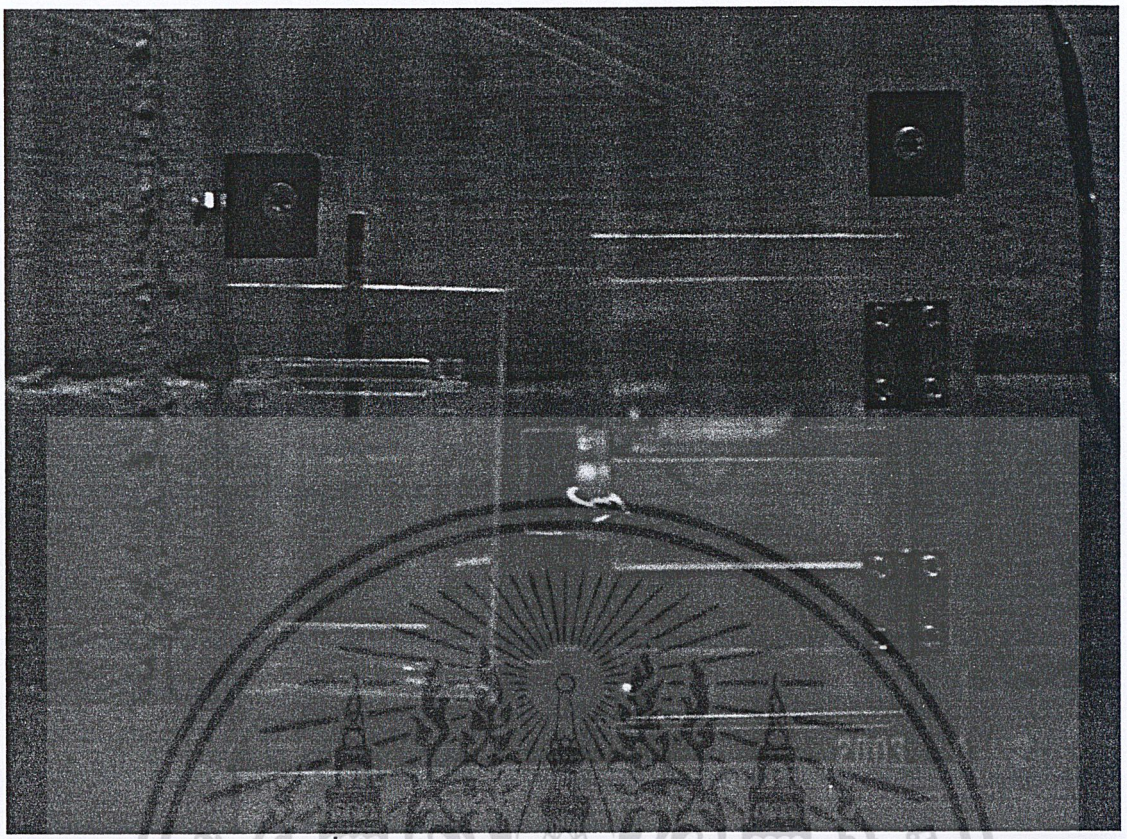
บรรณานุกรม

1. เรียนรู้และปฏิบัติการเรียนไมโครคอนโทรลเลอร์ตระกูล MCS51 , วรพจน์ กรแก้ววัฒนกุล และชัยวัฒน์ ลิ้มพรจิตรวิลัย , บริษัทอินโนเวตีฟเอ็กเพอริเมนต์ จำกัด
2. ฟิสิกส์ ม.4 , อาจารย์พันทิพา ถาวรรัตน์ อาจารย์สามารถ พงศ์ไพบูลย์ , บริษัทไอเอ็ดพับลิชชิ่ง จำกัด , 2533
3. แม็ค ม.ปลาย , บริษัทสำนักพิมพ์แม็ค จำกัด , ชวนพิมพ์ , 2533

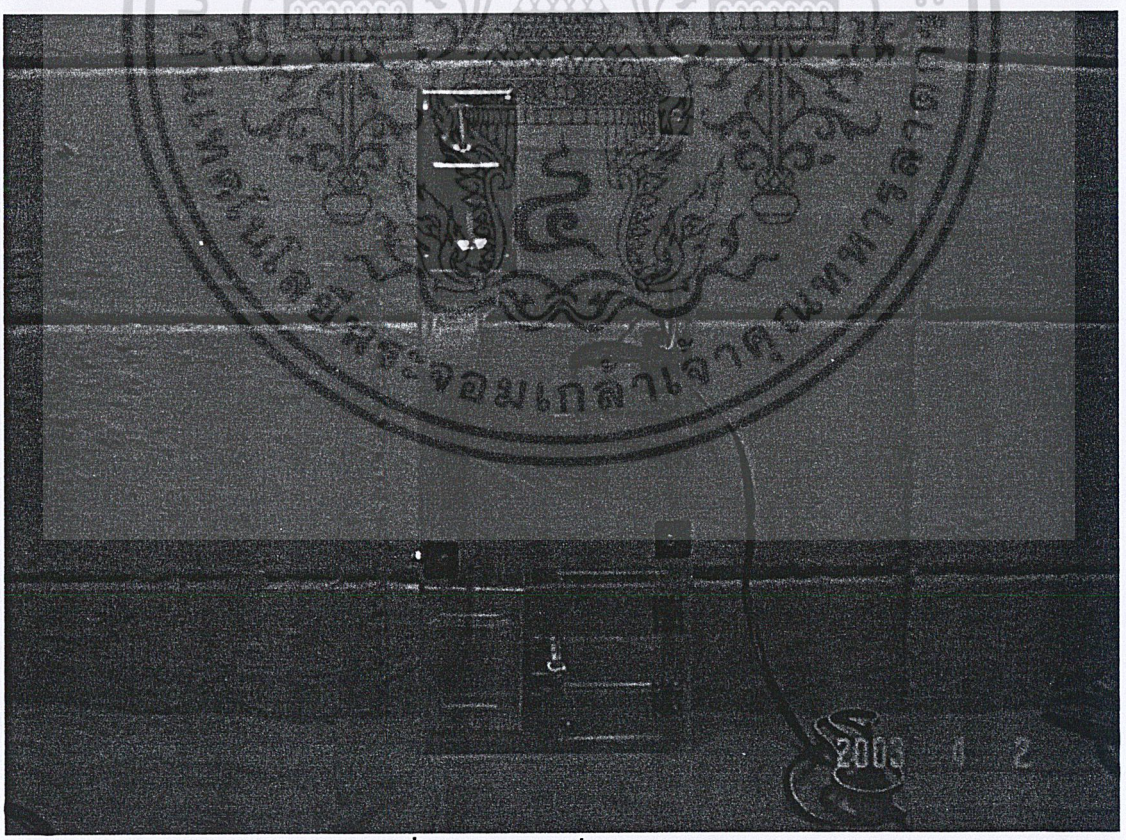
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

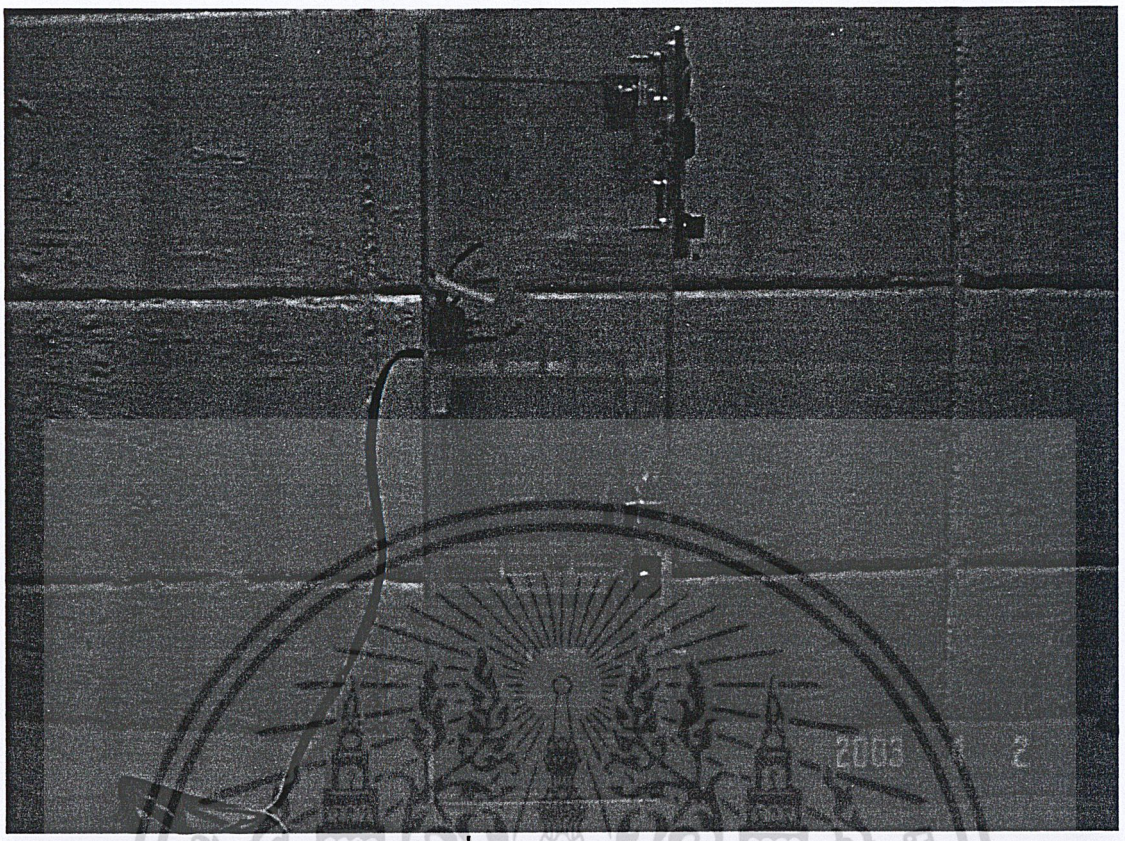


ภาพที่ 1 แสดงช่องเก็บเหรียญและช่องหยิบเงินทอน

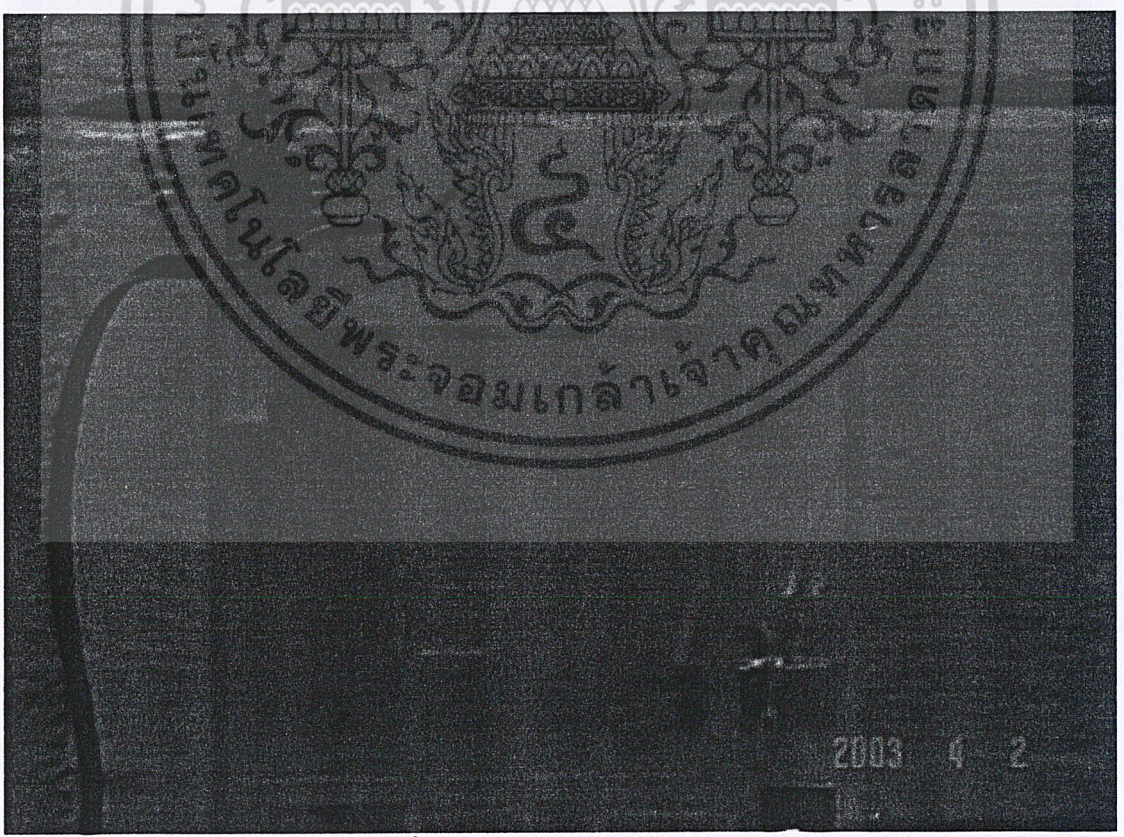


ภาพที่ 2 แสดงตัวเครื่องหยอดเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

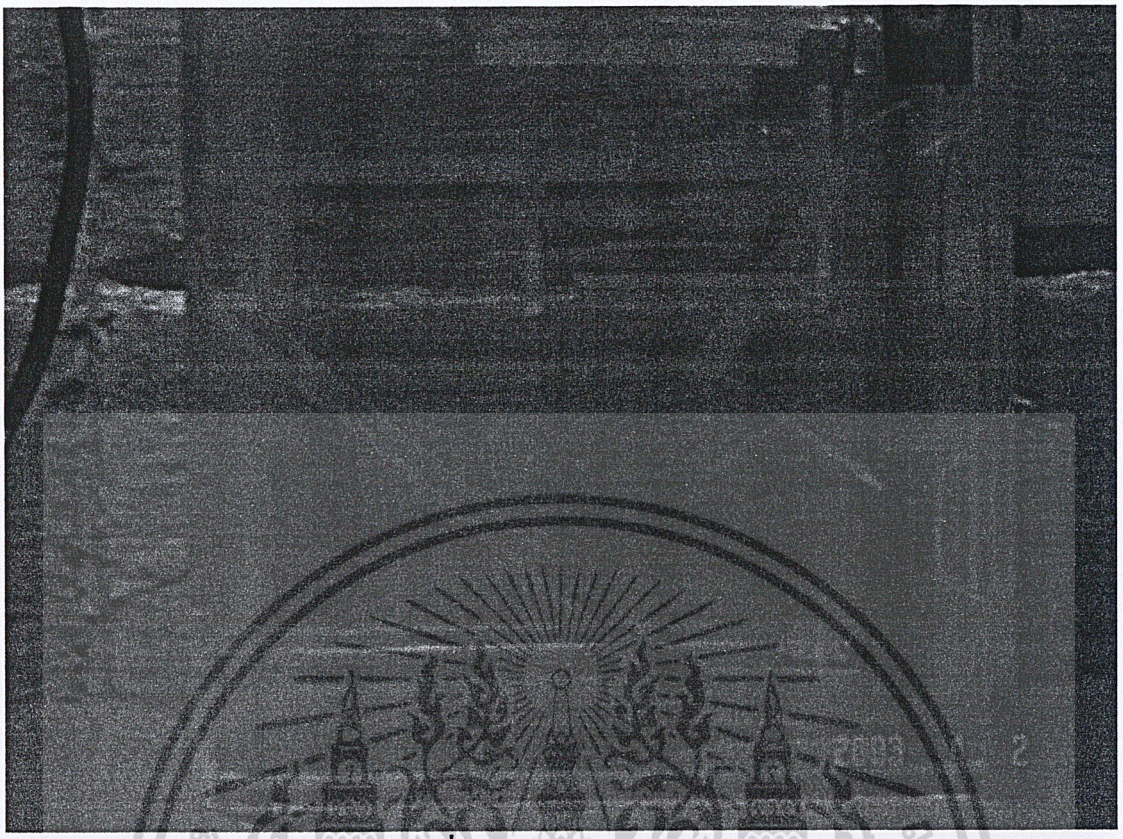


ภาพที่ 3 แสดงภาพด้านข้าง

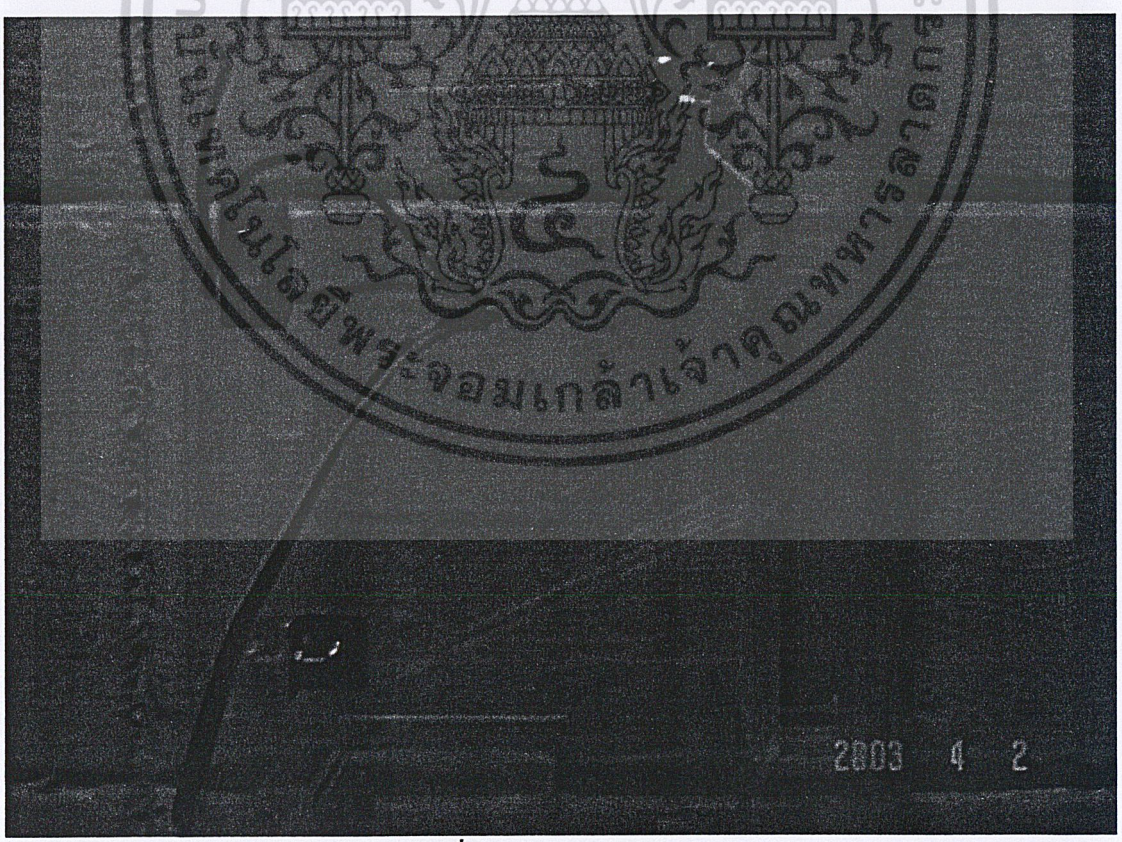


ภาพที่ 4 แสดงช่องเก็บสำรองเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

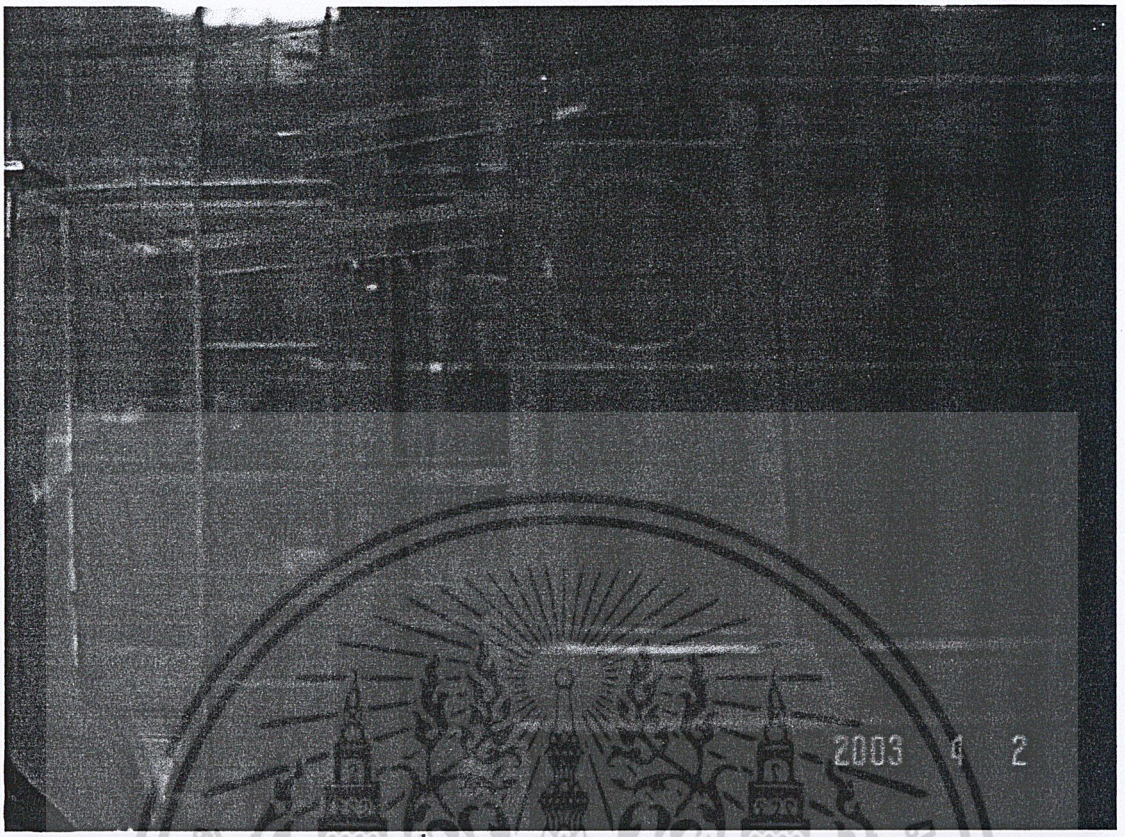


ภาพที่ 5 แสดงรางทอนเหรียญ

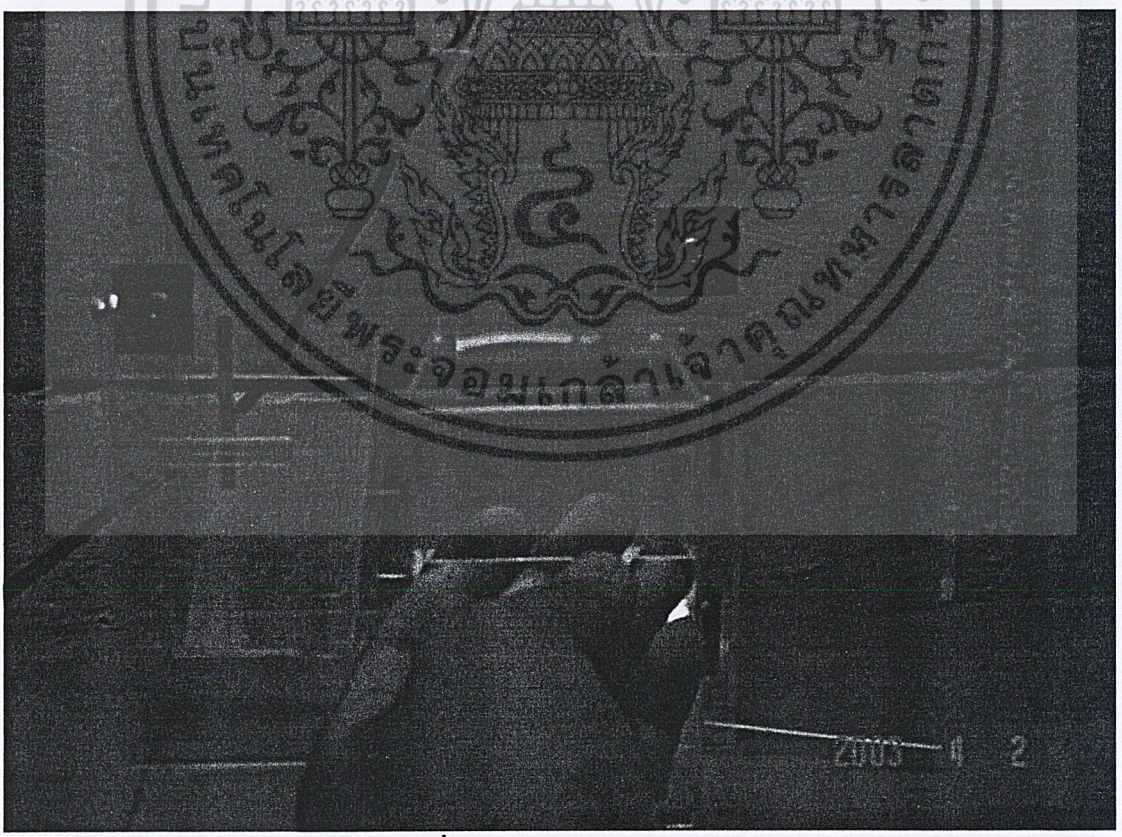


ภาพที่ 6 แสดงรางเหรียญล้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



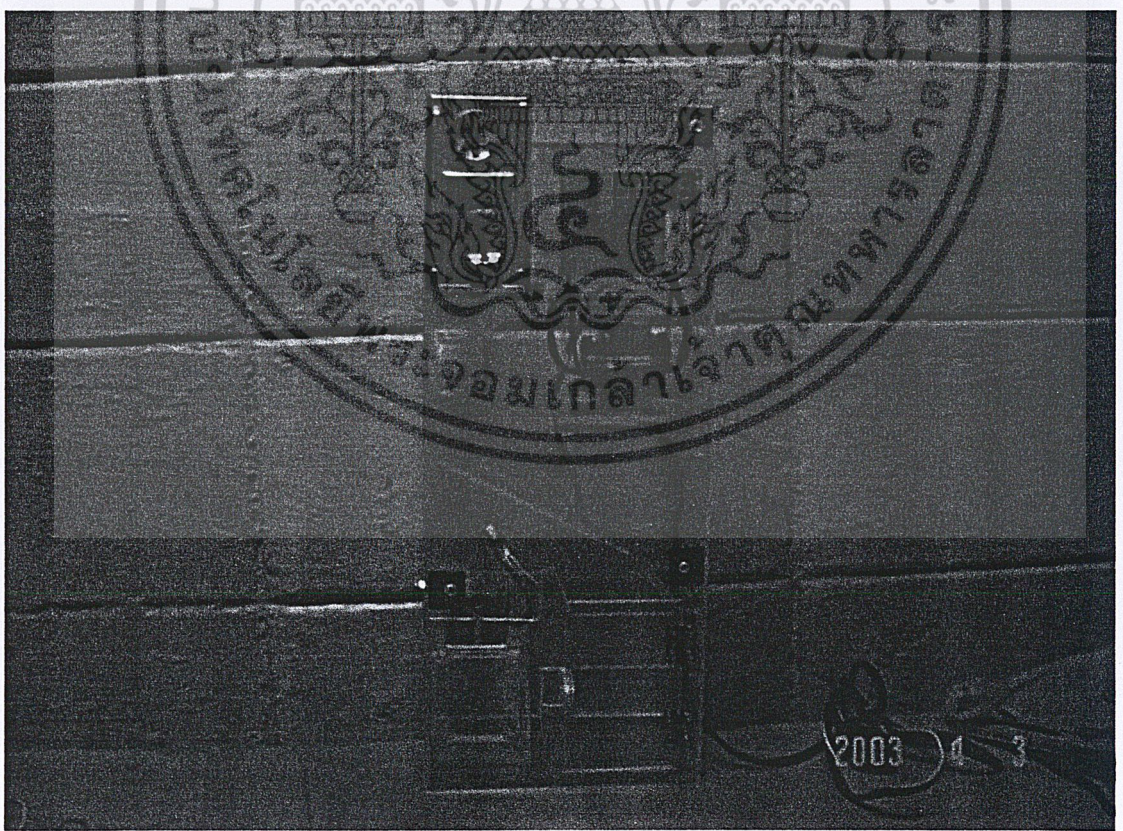
ภาพที่ 7 แสดงช่องคัดขนาดเหรียญ



ภาพที่ 8 แสดงกล่องเก็บเหรียญ

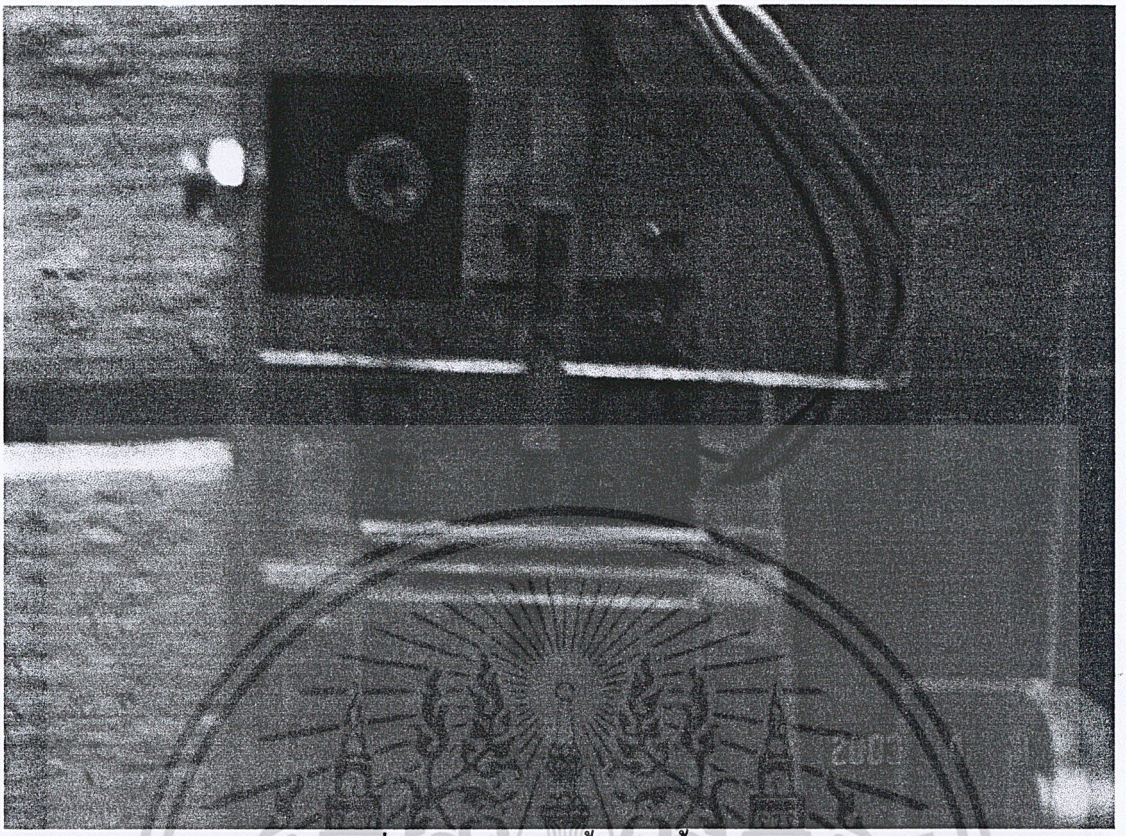
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 9 แสดงการหยิบเหรียญจากช่องทอน

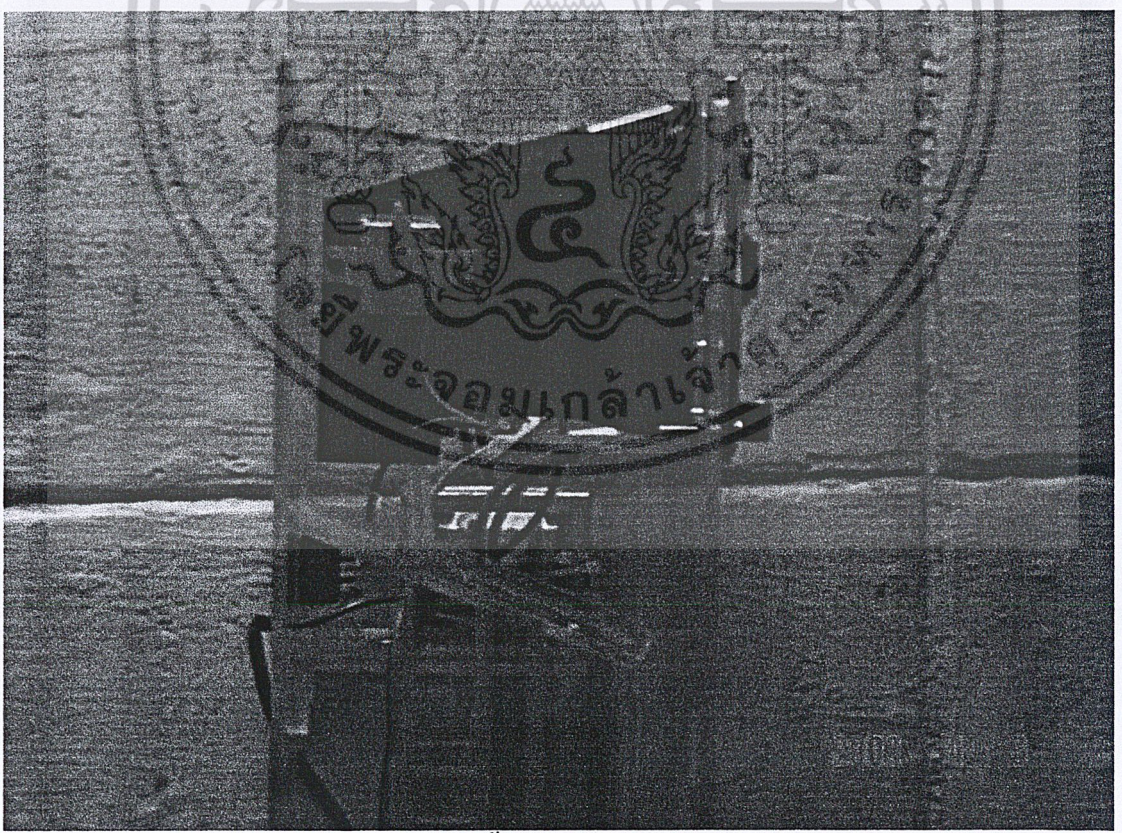


ภาพที่ 10 แสดงการติดตั้งอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

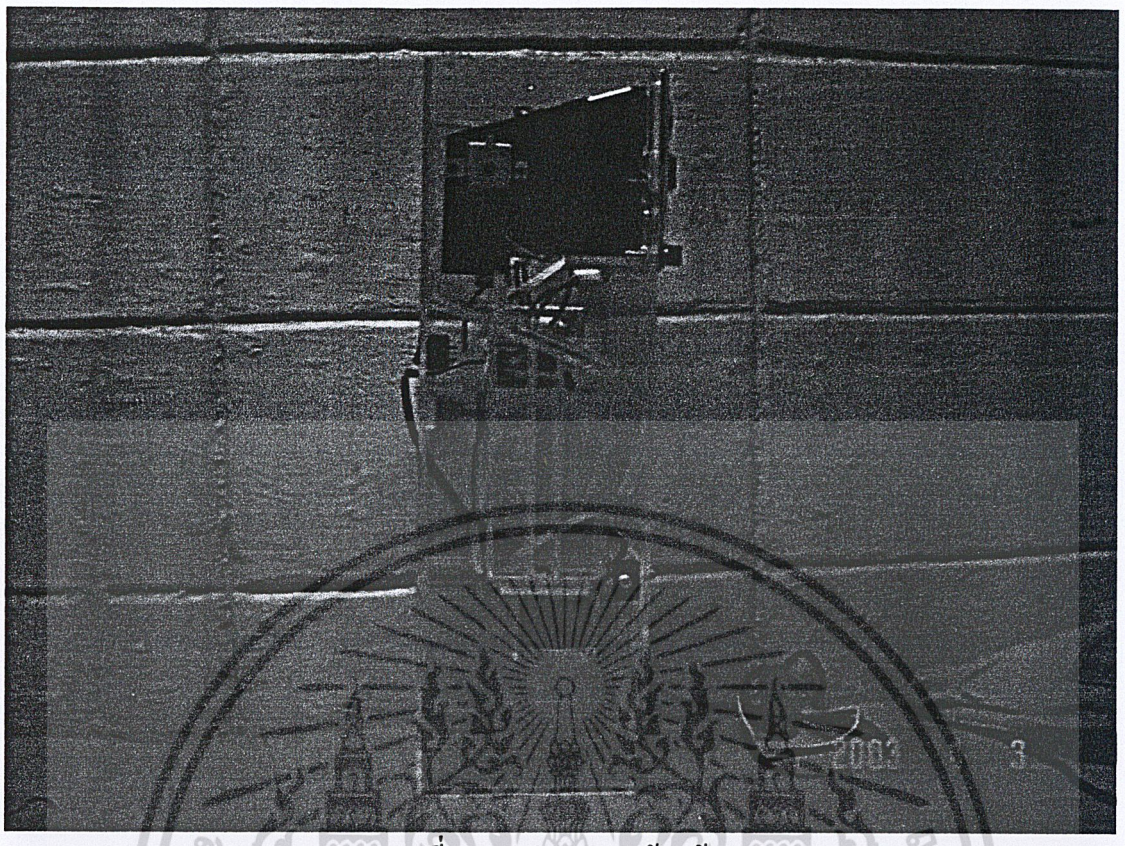


ภาพที่ 11 แสดงการติดตั้งสเต็ปปิ้งมอเตอร์

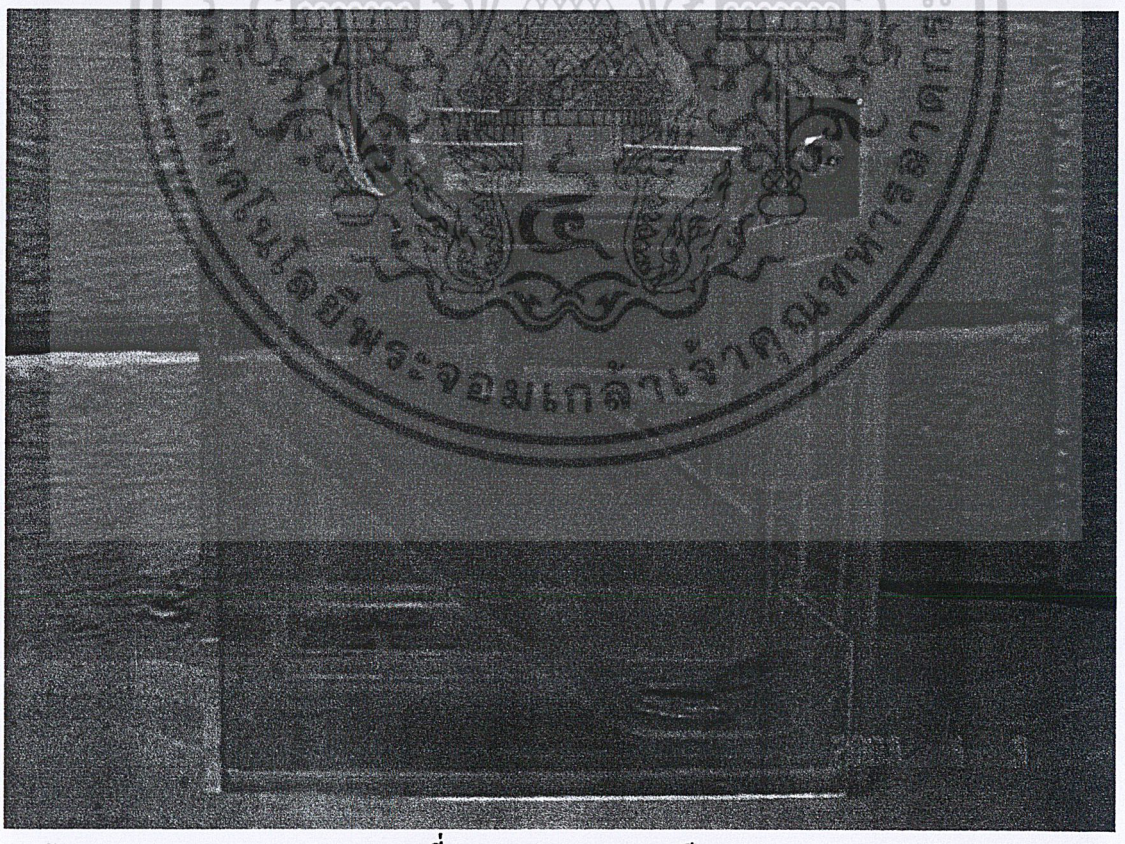


ภาพที่ 12 แสดงการติดตั้งหม้อแปลงและเครื่องตรวจเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

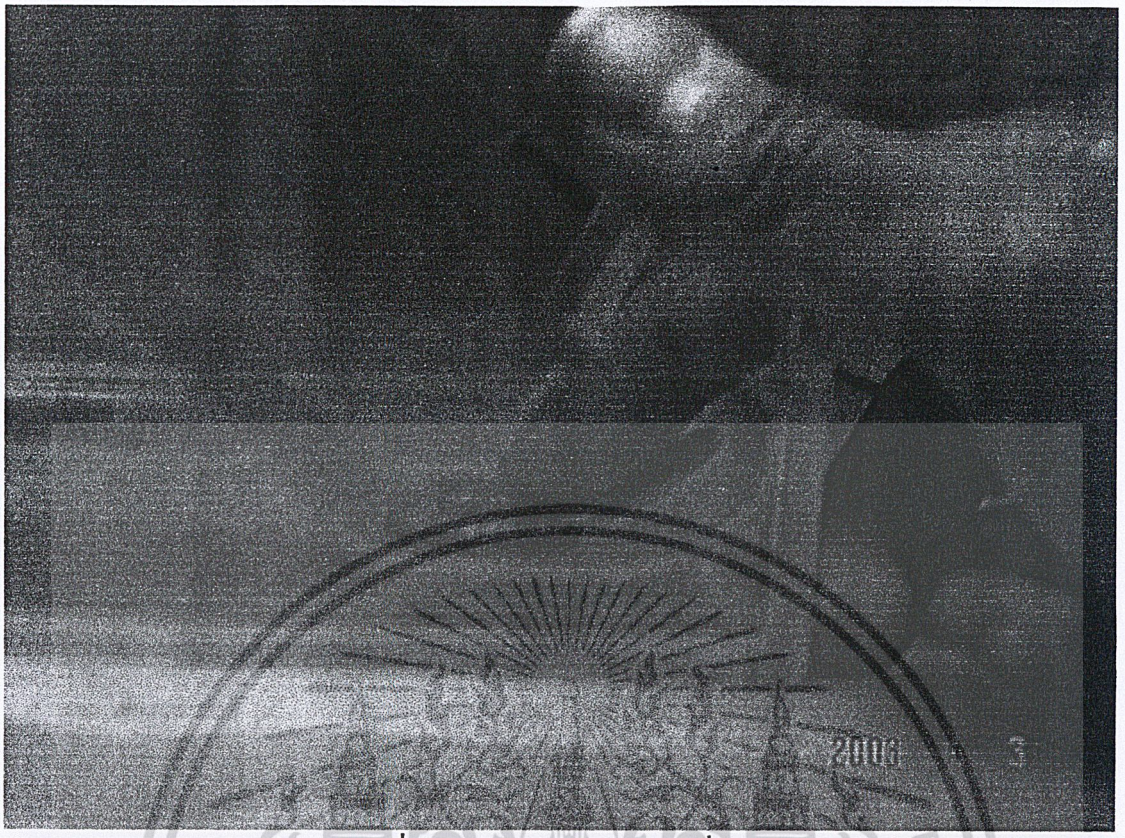


ภาพที่ 13 แสดงมุมมองด้านข้าง

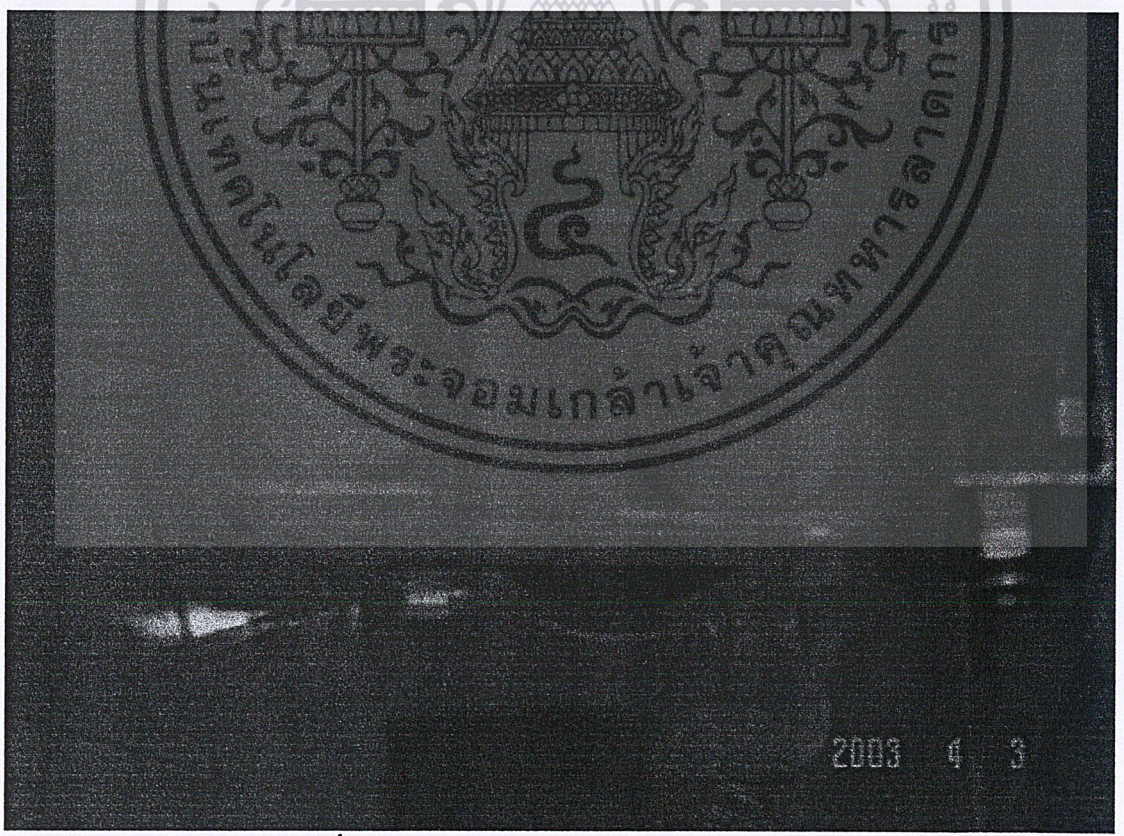


ภาพที่ 14 แสดงการทอนเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

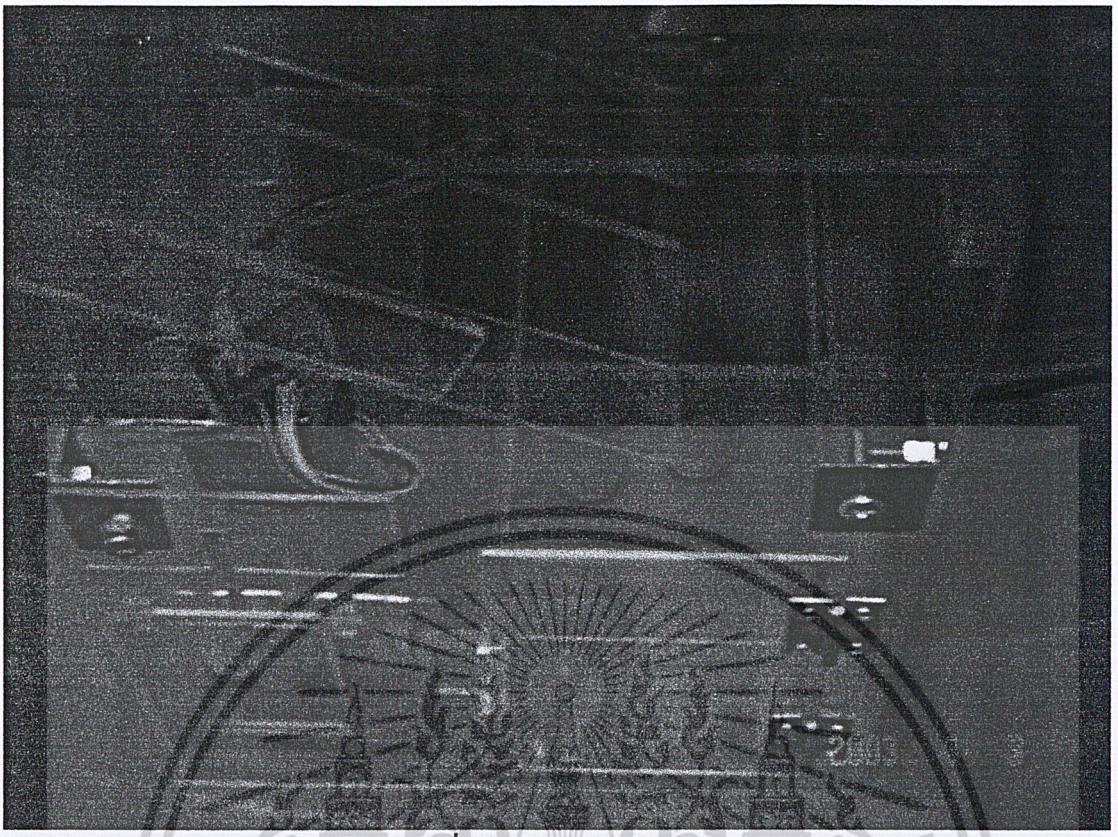


ภาพที่ 15 แสดงการหยิบเหรียญที่ทอน

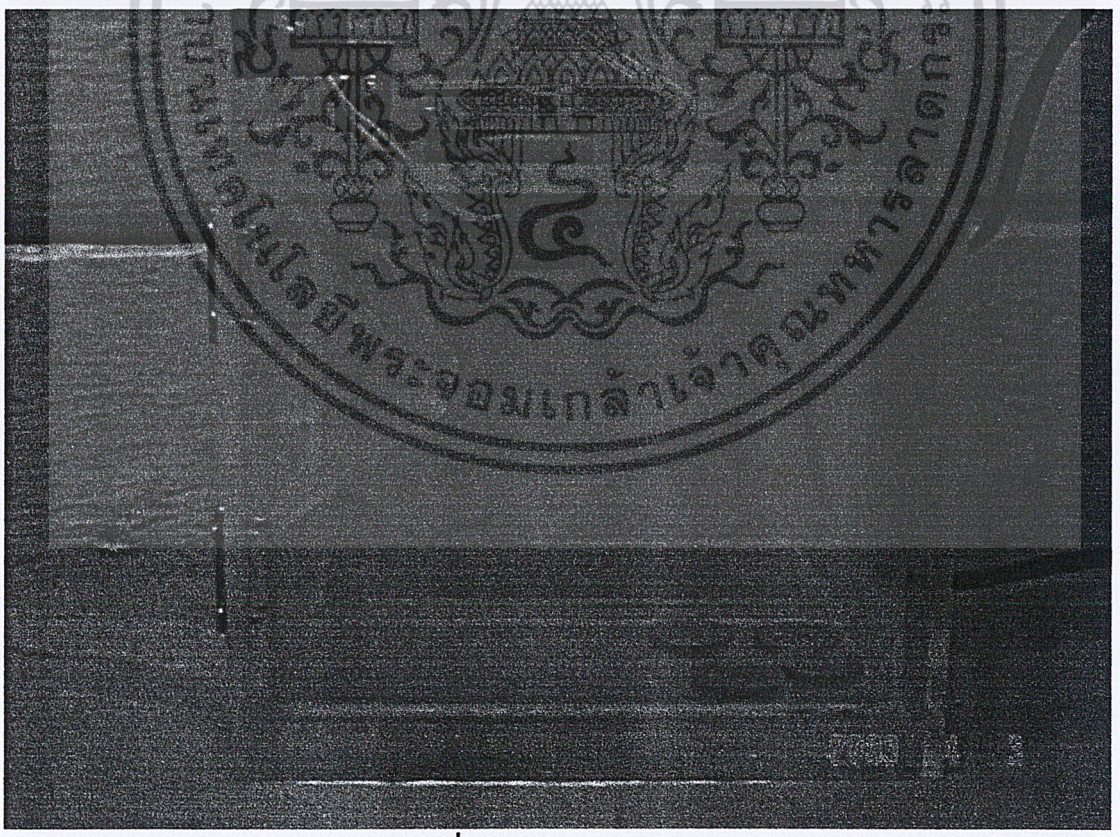


ภาพที่ 16 แสดงการเก็บเหรียญที่ช่องสำรองเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

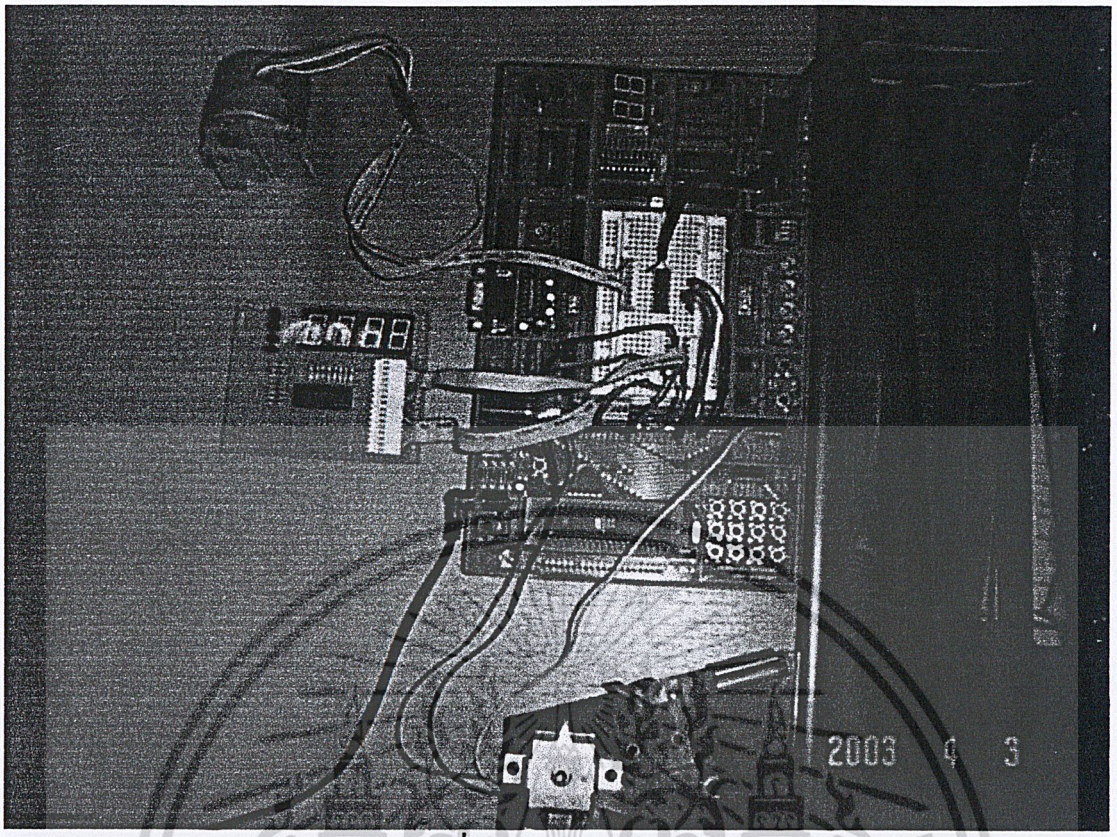


ภาพที่ 17 แสดงภาพโดยรวม

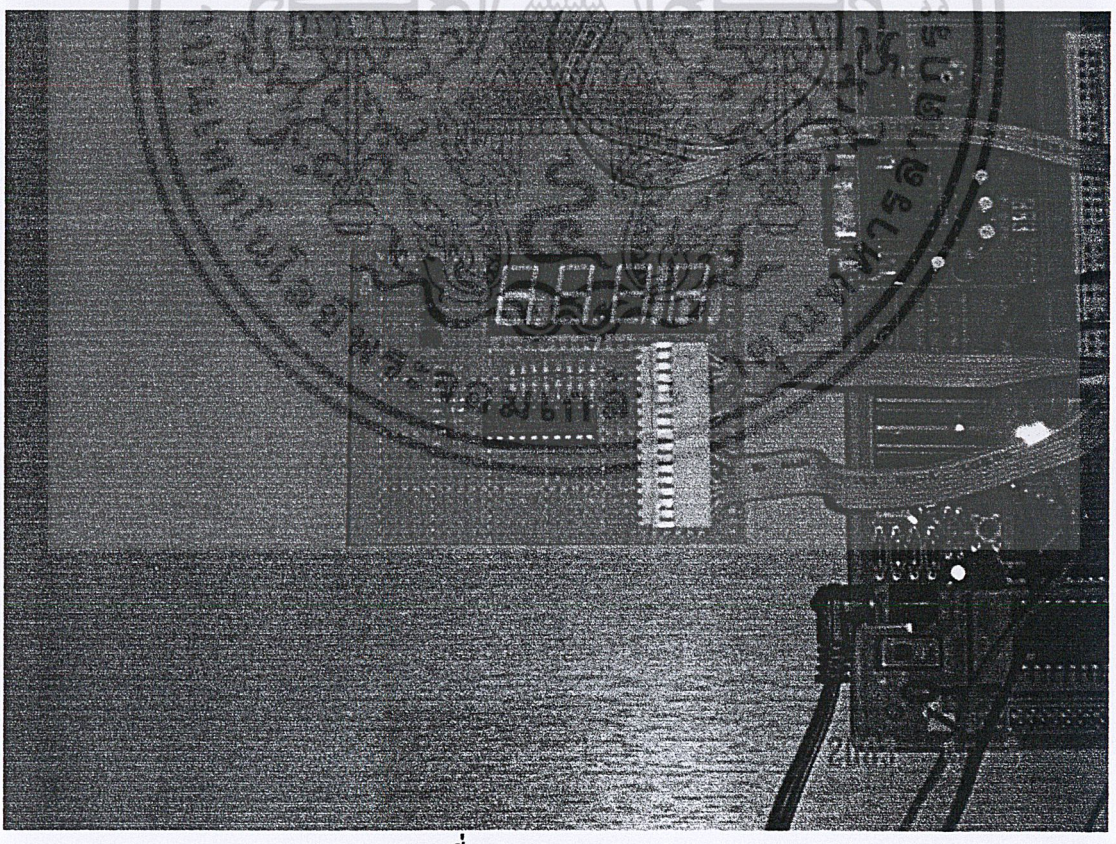


ภาพที่ 18 แสดงภาพด้านข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

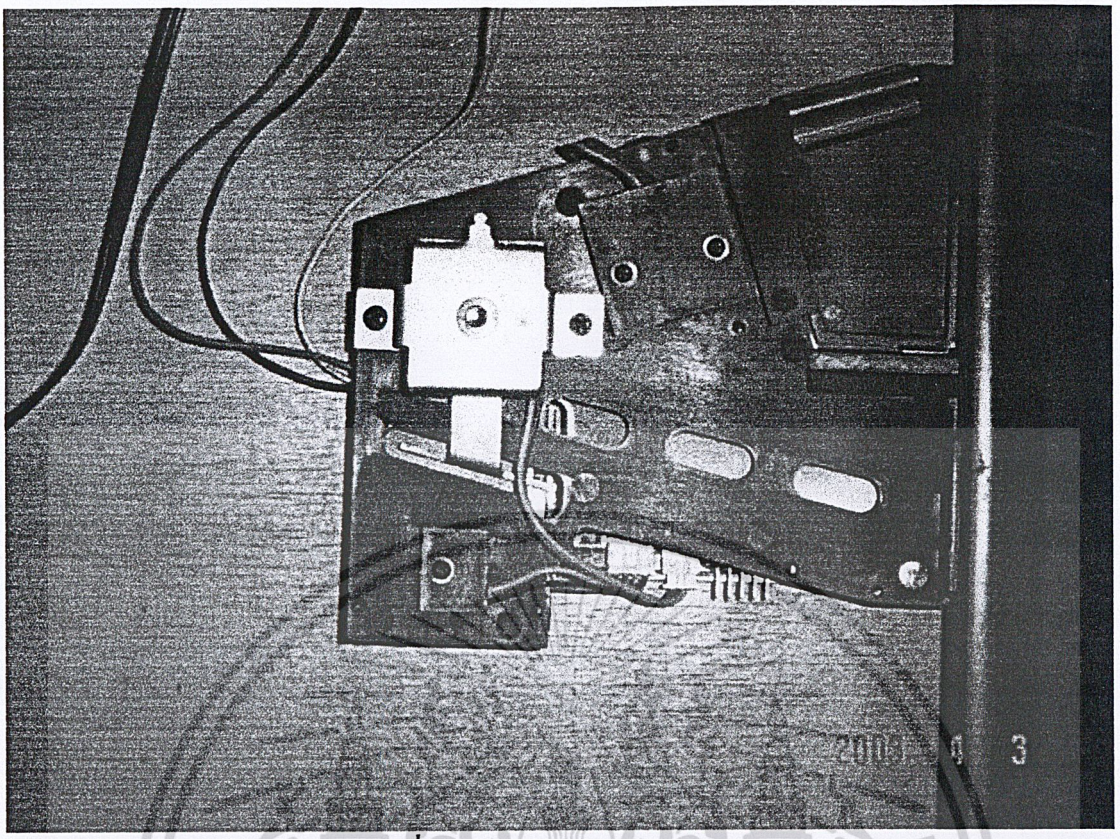


ภาพที่ 19 แสดงการทดลอง

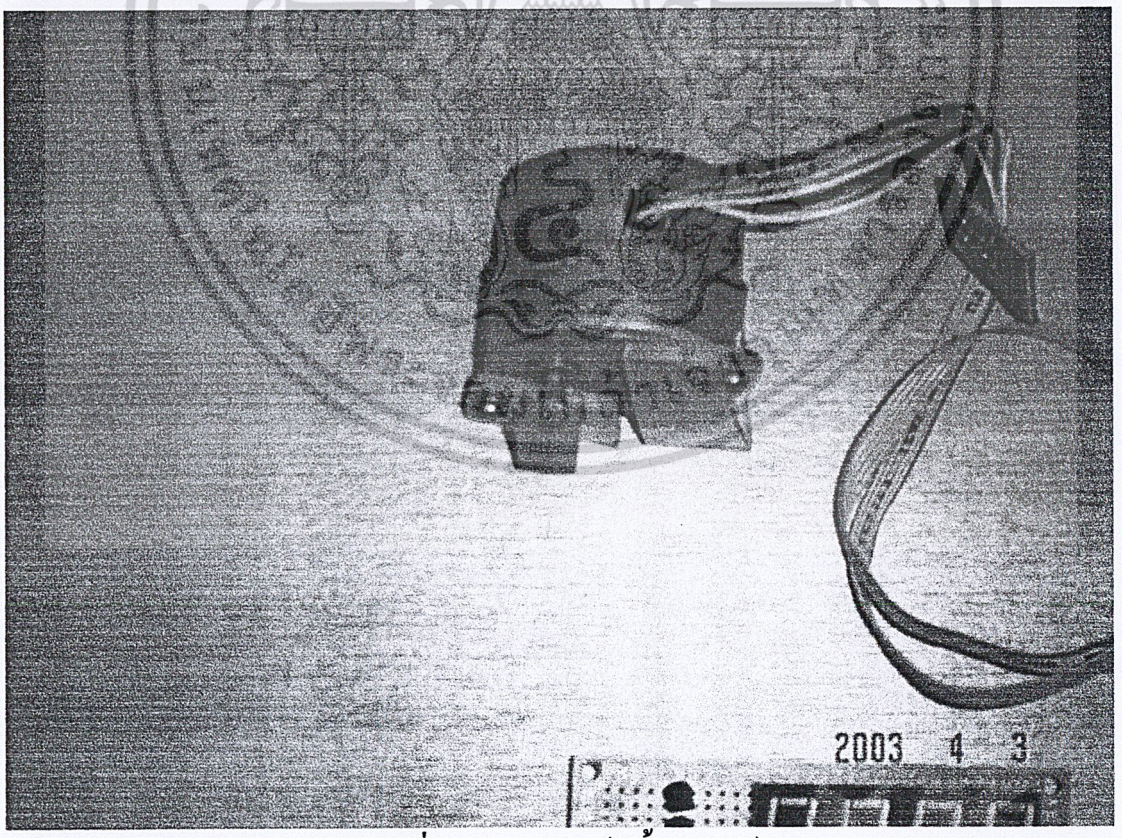


ภาพที่ 20 แสดงจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 21 แสดงชุดตรวจสอบเหรียญ



ภาพที่ 22 แสดงตัวสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้