# เครื่องชั่งน้ำหนักที่ควบคุมด้วยคอมพิวเตอร์และไมโครคอนโทรลเลอร์


นายจิรวัตน์ ศักดิ์รัตโน 5449136

นายวีระ ชุณหวรวรรณ 5449781

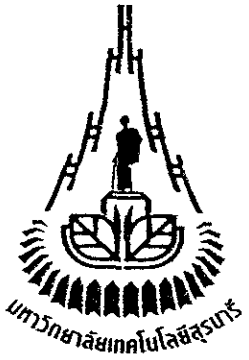

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2547

CONTRIBUTION




# เครื่องชั่งน้ำหนักน้ำนมวัวควบคุมโดยเซนเซอร์และไมโครคอนโทรลเลอร์

นาย ชัยวัฒน์ คำภีรธัมโม B4401308

นาย มิง คุณธรรมสกุล B4404781

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2547

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

**เครื่องชั่งน้ำหนักน้ำนมวัวควบคุมโดยเซ็นต์เซอร์และไมโครคอนโทรลเลอร์**

**ผู้ดำเนินการ** นาย ชัยวัฒน์ คำภีรธัมโม B4401308

นาย มิง คุณธรรมสกุล B4404781

**อาจารย์ที่ปรึกษา** อาจารย์ ดร. ชาญชัย ทองโสภา

**สาขาวิชา** โทรคมนาคม

**ภาคการศึกษา** 2547

---

**บทคัดย่อ**

จากการที่กลุ่มของพวกเราทราบมาว่าฟาร์ม ม.ท.ส. ของเราได้ทำการซื้อเครื่องชั่งน้ำหนักนมเพื่อทำการบันทึกปริมาณของน้ำนมวัวเพื่อเป็นฐานข้อมูลในการวิจัยและพัฒนาสูตรอาหารเพื่อเพิ่มน้ำนมวัว ซึ่งด้วยเครื่องและระบบมีราคาสูงมากแต่ปัจจุบันเครื่องดังกล่าวไม่สามารถได้งานตามวัตุประสงค์ที่ต้องการ ซึ่งแต่ละเครื่องก็มีความเสียหายและบกพร่องต่าง ๆ กันไป ซึ่งโดยรวมคือไม่สามารถทำการชั่งน้ำหนักและแสดงผลได้ตามต้องการ ซึ่งส่งผลให้ทางฟาร์มไม่มีฐานข้อมูลของน้ำนมวัว จึงทำให้ไม่สามารถวิจัยและพัฒนาสูตรอาหารเพื่อเพิ่มน้ำนมวัวได้ตามวัตถุประสงค์ เพราะไม่มีอะไรเป็นตัววัดผลของการทดลอง วิธีการแก้ไขปัญหานี้มีอยู่ 3 วิธีคือ 1.นำน้ำนมวัวใส่ภาชนะแล้วใช้เครื่องชั่งธรรมดาชั่ง จดบันทึกแล้วนำไปเทรวม 2.ซื้อเครื่องและระบบ 3.ซ่อมแซม วิธีที่ 1.ทำให้เกิดความยุ่งยากและความลำบากเพราะวัวมีจำนวนมาก วิธีที่ 2.เป็นไปได้เพราะตัวเครื่องมีราคาสูงและต้องสั่งซื้อจากต่างประเทศสิ้นเปลืองทั้งเวลาและงบประมาณ และได้ทราบมาจากทางช่างประจำฟาร์มว่าเครื่องชั่งนี้ได้เลิกผลิตแล้ว วิธีที่ 3.เป็นไปได้แต่การจะซ่อมแซมระบบเดิมนั้นทำไม่ได้เพราะยุ่งยากมากเราจึงมีความคิดว่าจะสร้างอุปกรณ์ขึ้นมาใหม่โดยทำงานร่วมกับอุปกรณ์เดิมและให้ผลตามที่ต้องการได้ซึ่งเป็นการประหยัดทั้งเวลาและงบประมาณ หลังจากโครงการเสร็จเรียบร้อยแล้วเครื่องชั่งจะสามารถชั่งน้ำนมวัวและแสดงผลได้ตามต้องการประโยชน์ที่ตามมาคือทางฟาร์ม ม.ท.ส. จะมีฐานข้อมูลของน้ำนมวัวเพื่อวิจัยและพัฒนาสูตรอาหารเพื่อเพิ่มน้ำนมวัวได้ตามวัตถุประสงค์เดิม

**Abstract**

We know that the SUT farm has been using the Strongko Milkers System from Denmark for measuring the milk volume and maintaining a data base of milk volume. But nowadays the Strangko Milkers System has broken. In this project, we improve and develop the Strangko Milkers. So that it is easy to use and the farmer can measure the milk volume to improve the formula.

# สารบัญ

**บทที่ 1 บทนำ** .....1

**บทที่ 2 ทฤษฎีและหลักการ** .....3

2.1โครงสร้างของระบบคอมพิวเตอร์.....3

2.2ความหมายของไมโครคอนโทรลเลอร์.....4

2.3ความแตกต่างของไมโครคอนโทรลเลอร์กับไมโครโปรเซสเซอร์.....5

2.4จะเลือกใช้ไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์เมื่อใด ?.....7

2.5โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช.....7

2.6การเชื่อมต่อกับหน่วยแสดงผล.....17

2.7การเชื่อมต่อสวิตช์เข้ากับระบบไมโครคอนโทรลเลอร์.....23

2.8รีเลย์(Relay).....28

**บทที่ 3 การคำนวนและการสร้าง**.....34

3.1ก) ส่วนแสดงผล.....34

3.1ข) ส่วนของเซนเซอร์.....36

3.1 ค)ส่วนควบคุมการทำงานของวาล์วในระบบเดิมของเครื่องชั่ง.....37

3.2 การออกแบบวงจรส่วนของ AT89S8252.....38

3.3 การโปรแกรมไมโครคอนโทรลเลอร์.....38

**บทที่ 4 การทดลองและผลการทดลอง**.....40

4.1การทดลองทางHard Ware ร่วมกับกล่องชั่งน้ำนม.....40

4.2การทดลองโดยการนำเครื่องควบคุมที่สร้างขึ้นไปติดตั้งเข้ากับระบบจริง.....42

**บทที่ 5 บทวิจารณ์และบทสรุป**.....52

**ข้อเสนอแนะ**.....53

## สารบัญ(ต่อ)

**ภาคผนวก**.....54

**กิตติกรรมประกาศ**.....55

**อ้างอิง**.....56

## สารบัญรูปภาพ


รูปที่ 1.1 แสดงการเชื่อมต่ออุปกรณ์ควบคุมในการทำงานของระบบที่สร้างขึ้น.....2
รูปที่ 2.1 ไดอะแกรมโครงสร้างของคอมพิวเตอร์.....3
รูปที่ 2.2 โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์.....5
รูปที่ 2.3 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์.....6
รูปที่ 2.4 โครงสร้างพื้นฐานของ MCS-51 แบบแฟลชในอนุกรม AT89CXX.....9
รูปที่ 2.5 โครงสร้างพื้นฐานของ MCS-51แบบแฟลชในอนุกรม AT89SXX.....10
รูปที่ 2.6 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชของ Atmel...12
รูปที่ 2.7 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5XX.....12
รูปที่ 2.8 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช.....16
รูปที่ 2.9 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช.....16
รูปที่ 2.10 แสดงโครงสร้างและสัญลักษณ์ของไดโอดเปล่งแสง.....17
รูปที่ 2.11 แสดงการเชื่อมต่อ 74LS373 กับ LED.....18
รูปที่ 2.12 แสดง LED แต่ละแบบ และการต่อเข้ากับพอร์ต.....18
รูปที่ 2.13 แสดง LED แต่ละแบบ และการต่อเข้ากับพอร์ต.....19
รูปที่ 2.14 แสดงสัญลักษณ์ของแต่ละ SEGMENT.....19
รูปที่ 2.15 แสดงการต่อ LED 7 ส่วนเข้ากับ 8255.....21
รูปที่ 2.16 แสดงการต่อ LED 7 ส่วนหลาย ๆ ตัวกับ 8255.....22
รูปที่ 2.17 แสดงการต่อ LED 7 ส่วน 4 ตัว.....22
รูปที่ 2.18แสดงการสร้างลอจิกจากสวิตช์.....23
รูปที่ 2.19 แสดงตัวอย่างสวิตช์ที่นิยมใช้กัน.....23
รูปที่ 2.20 แสดงสัญลักษณ์ของสวิตช์แบบต่าง ๆ.....24
รูปที่ 2.21 แสดงไบนารีสวิตช์.....24
รูปที่ 2.22 แสดงสัญญาณที่เกิดจากการสั่นของหน้าสัมผัสของสวิตช์.....24
รูปที่ 2.23 แสดงการนำแนนเกตมาต่อเป็น R-S ฟลิปฟลอป เพื่อแก้การสั่นของสวิตช์.....25
รูปที่ 2.24 การฟลิปฟลอปมาแก้ปัญหาสวิตช์.....25
รูปที่ 2.25 การแก้ปัญหาสวิตช์โดยใช้ Schmitt Trigger.....26
รูปที่ 2.26 แสดงการต่อสวิตช์เข้ากับพอร์ท P1...*.....26

## สารบัญรูปภาพ(ต่อ)


รูปที่ 2.27 แสดงไดอะแกรมการใช้อินเทอร์รัพท์.....27
รูปที่ 2.28 แสดงรูปร่างและสัญลักษณ์ของรีเลย์.....28
รูปที่ 2.29แสดงหลักการทำงานเบื้องต้นของรีเลย์.....28
รูปที่ 2.30 แสดงโครงสร้างและสัญลักษณ์ของชุดหน้าสัมผัสแบบ 4PST.....29
รูปที่ 2.31 แสดงหน้าสัมผัสแบบ SPDT.....29
รูปที่ 2.32 แสดงหน้าสัมผัสแบบ SPDT แบบ Break Make และ Make Break.....30
รูปที่ 2.33 แสดงรีเลย์ชนิดอาร์เมเจอร์.....30
รูปที่ 2.34 แสดงรีเลย์ชนิดรีดรีเลย์.....31
รูปที่ 2.35 แสดงรีเลย์ชนิดรีดรีเลย์.....31
รูปที่ 2.36 แสดงโซลิดสเตตรีเลย์.....32
รูปที่ 2.37แสดงการนำรีเลย์ไปต่อเป็นสวิตช์ในวงจรกันขโมย.....32
รูปที่ 2.38แสดงการนำรีเลย์มาต่อเป็นวงจรออสซิลเลเตอร์เพื่อทำเป็นไฟกระพริบ.....33
รูปที่ 2.39 แสดงการนำรีดสวิตช์ไปใช้ในวงจรกันขโมย.....33
รูป 3.1 แสดงวงจรการทำงานทั้งหมด.....35
รูป 3.2 แสดงส่วนของ7-SEGMENT แบบ COMMON CATHODE.....35
รูป 3.3 แสดงส่วนของ LED แทนการทำงานของสวิตช์ Start, Stop และ Sensor.....35
รูป 3.4 แสดงส่วนของ LED แทนการทำงานของ Sensor ทั้ง 3 ระดับ.....35
รูป 3.5 แสดงส่วนของ LED แทนการทำงานของSUPPLY ของระบบ.....35
รูป 3.6 แสดงส่วนของHard Ware ของ Sensor และแท่ง Sensor ที่จะนำเข้าไปในระบบเดิม.....36
รูป 3.7 แสดงส่วนของHard Ware ของ รีเลย์ และวาล์วที่จะควบคุมในระบบเดิม.....37
รูป 3.8 แสดงส่วนของวาล์วที่จะควบคุมในระบบเดิม.....37
รูป 3.9 แสดงส่วนของสวิตช์ต่างๆในการทำงาน.....38
รูป 4.1แสดงเครื่องชั่งน้ำนมจากระบบเดิม ด้านหน้า และด้านใต้.....40
รูป 4.2 แสดงการปรับปรุงส่วนของเครื่องควบคุมที่เราสร้างขึ้น.....41
รูป 4.3 แสดงเครื่องชั่งน้ำนมจากระบบเดิม เปิดฝาครอบด้านบนและด้านล่าง.....41
รูป 4.4 แสดงเครื่องชั่งน้ำนมจากระบบเดิม Sensor และวาล์ว.....41
รูป 4.5 แสดงระบบควบคุมใหม่ที่สร้างขึ้น.....41

## สารบัญรูปภาพ(ต่อ)

รูป 4.6 แสดงการติดตั้งระบบควบคุมใหม่ที่สร้างกับเครื่องชั่งน้ำนมและการติดตั้ง Sensor.....42
รูป 4.7 แสดงการแก้ปัญหาของ Sensor โดยทำการเปลี่ยนรูปแบบ.....42
รูป 4.8 แสดงการแก้ปัญหาของการแสดงผลทาง7-SEGMENT .....43
รูป 4.9 แสดงสถาพแวดล้อมของฟาร์ม มทส.....44
รูป 4.10 ก) แสดงภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง.....44
รูป 4.10 ข) แสดงภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง.....44
รูป 4.10 ค) แสดงสภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง.....45
รูป 4.10 ง) แสดงสภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง.....45
รูป 4.11 ถังรวมน้ำนมที่ได้ผ่านการชั่งน้ำหนักแล้ว.....45
รูป 4.12 ก) แสดงสภาพของระบบชั่งน้ำนม และเครื่องชั่งน้ำนมเดิม.....46
รูป 4.12 ข) แสดงสภาพของระบบชั่งน้ำนม และเครื่องชั่งน้ำนมเดิม.....46
รูป 4.13 แสดงเครื่องชั่งน้ำนมและเครื่องควบคุมของระบบเดิมซึ่งมีความเสียหาย.....47
รูป 4.14 ก) แสดงเครื่องควบคุมที่สร้างขึ้นติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิม.....47
รูป 4.14 ข)แสดงเครื่องควบคุมที่สร้างขึ้นติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิม.....48
รูป 4.14 ค) แสดงเครื่องควบคุมที่สร้างขึ้นติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิม.....48
รูป 4.14 ง) แสดงเครื่องควบคุมที่สร้างขึ้นติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิม.....49
รูป 4.15 ก) แสดงการรีดน้ำนมเพื่อทดสอบการทำงานร่วมกันของระบบ.....49
รูป 4.15 ข) แสดงการรีดน้ำนมเพื่อทดสอบการทำงานร่วมกันของระบบ.....50
รูป 4.16 แสดงการทำงานโดยมีอาจารย์และช่างฝ่ายเทคนิคของฟาร์มเป็นผู้ทดสอบ.....50
รูป 4.17 แสดงเครื่องชั่งระบบใหม่โดยเป็นระบบ Mechanic .....51

## สารบัญตาราง

ตารางที่ 2.1เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์.....6
ตารางที่ 2.2 รายละเอียดโดยสรุปบางส่วนของ MCS-51 แบบแฟลช.....10
ตารางที่ 2.3 หน้าที่พิเศษของพอร์ต 1 ใน MCS-51แบบแฟลชของ Atmel.....15
ตารางที่ 2.4 แสดงรายละเอียดของไดโอดเปล่งแสง.....17
ตารางที่ 2.5 แสดงตำแหน่งขาที่จะต่อกับพอร์ต.....20
ตารางที่ 2.6 แสดงข้อมูลที่ส่งให้ LED 7 ส่วนแสดงเป็นเลขต่าง ๆ.....20

# บทที่ 1
# บทนำ

เนื่องด้วยในปัจจุบันเทคโนโลยีขนาดเล็กต่าง ๆ ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์เรามากขึ้น ซึ่งทำให้อุปกรณ์ต่าง ๆ มีขนาดเล็กลงด้วยและมีกลไกลการทำงานที่ฉลาดมากขึ้น สิ่งที่เราจะกล่าวถึงก็คือ เทคโนโลยีของ “ไมโครคอนโทรลเลอร์” ซึ่ง ณ ปัจจุบันเทคโนโลยี ไมโครคอนโทรลเลอร์ ได้มีการนำไปใช้ในการควบคุมการทำงานของ อุปกรณ์ไฟฟ้า ประเภทต่าง ๆ มากมาย

เทคโนโลยี ไมโครคอนโทรลเลอร์จะประกอบไปด้วยในส่วนของ ซอฟต์แวร์และฮาร์ดแวร์ โดยที่เราจะใช้ในส่วนของซอฟต์แวร์ควบคุมการทำงานของฮาร์ดแวร์ ซึ่งหากเราทำให้ซอฟต์แวร์และฮาร์ดแวร์สามารถทำงานเชื่อมต่อกันได้ก็จะทำให้มีความสะดวก ประหยัดและปลอดภัย ซึ่งในที่นี้จะขอยกตัวอย่างในการประยุกต์ใช้งานในส่วนต่าง ๆ เช่น

**1.1 ทางด้านการแพทย์**

ในอุปกรณ์ต่าง ๆ ที่ใช้ในด้านการแพทย์ ซึ่งเป็นงานที่ต้องมีความแม่นยำและเที่ยงตรงอีกทั้งยังต้องมีความสะดวกในการใช้งาน ซึ่งในอุปกรณ์บางอย่าง หากมีมนุษย์เป็นผู้ควบคุมอาจทำให้เกิดข้อผิดพลาดได้

**1.2 ทางด้านความมั่นคงและการรักษาความปลอดภัย**

ใช้ในการควบคุมอุปกรณ์รักษาความปลอดภัยต่าง ๆ เช่น ควบคุม กล้องวงจรปิด ควบคุมการเข้าออกในสำนักงาน ควบคุมระบบSensor ในการรักษาความปลอดภัยต่าง ๆ

**1.3 ทางด้านธุรกิจและอุตสาหกรรม**

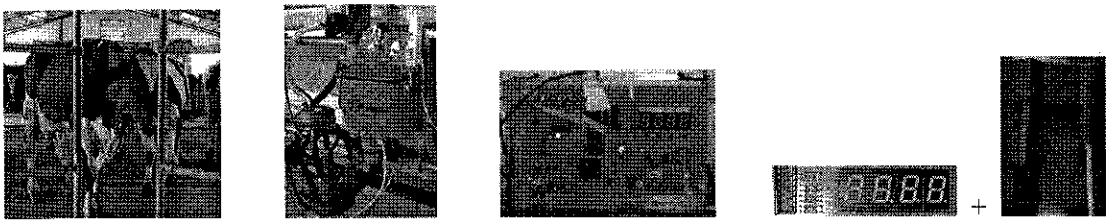
ใช้ในการควบคุมเครื่องจักรและอุปกรณ์เครื่องมือต่าง ๆ ภายในโรงงาน ในจุดที่มีความเสี่ยงสูง โดยไม่ต้องอาศัยมนุษย์เข้าไปทำงานในที่ที่มีความเสี่ยงเพื่อความปลอดภัยของพนักงานเอง อีกทั้งยังช่วยให้เครื่องมือต่าง ๆ ทำงานได้เองอย่างอัตโนมัต มีความแม่นยำในการทำงานสูงและเป็นการประหยัดค่าใช้จ่ายในการลงทุน

**1.4 ทางด้านสังคม**

ใช้ในการควบคุมสัญญาณไฟจราจร เพื่อให้การจราจรเป็นไปอย่างสะดวกหรือควบคุมกล้องวงจรปิด ในจุดที่เสี่ยงต่อการเกิดการโจรกรรม หรืออุบัติภัยสูง เพื่อเป็นการดูแลสวัสดิภาพของประชาชน

ในโครงงานนี้ได้นำเอาเทคโนโลยีไมโครคอนโทรลเลอร์ จากที่กล่าวในข้างต้นมาทำการศึกษาทั้งทางด้าน ฮาร์ดแวร์และซอฟต์แวร์ เพื่อนำมาสร้างเครื่องมือชุดนี้ขึ้นเพื่อให้ทำงานร่วมกับ

ระบบเดิมของฟาร์มได้รวมถึงยังทำให้เกิดความสะดวกมากขึ้นในการใช้งานและง่ายต่อการแก้ไข โดยหลักการทำงานอย่างคร่าว ๆ สามารถดูได้จากบล็อก รูปที่ 1.1 ดังต่อไปนี้



วัว---------- เครื่องชั่ง --------- ชุดควบคุม ---------- จอแสดงผล + วาล์ว

รูปที่ 1.1 แสดงการเชื่อมต่ออุปกรณ์ควบคุมในการทำงานของระบบที่สร้างขึ้น

จากบล็อกข้างต้นสามารถอธิบายได้ดังนี้

-เครื่องรีดนมวัว จะเป็นระบบเดิมที่มีอยู่แล้วซึ่งเป็นระบบสูญญากาศโดยจะรีดนมจากวัวที่เข้ามาในบล็อกที่กำหนดแล้วส่งไปยังเครื่องชั่ง

-เครื่องชั่ง Strangko Milkers เครื่องชั่งเป็นระบบสูญญากาศเช่นเดียวกับเครื่องรีดนม แต่ไม่สามารถทำงานได้ จึงเป็นส่วนที่เราต้องทำการแก้ไข โดยภายในประกอบด้วยตัวของเซนเซอร์และวาล์ว

-ชุดควบคุม เป็นส่วนที่เราได้สร้างขึ้น โดยจะรับสัญญาณจากเซนเซอร์ที่ติดตั้งเข้าไปในเครื่องชั่ง จะทำการประมวลผลถึงน้ำหนักที่ได้และส่งนมเข้าสู่ถังรวมโดยผ่านวาล์ว ซึ่งควบคุมจะทำการควบคุมการเปิดปิดของระบบวาล์วด้วย

-วาล์ว+จอแสดงผล ซึ่งจะถูกควบควบโดยชุดควบคุม และผลของน้ำหนักนมวัวที่ได้จะถูกแสดงออกทางจอแสดงผล

จากการทำงานเบื้องต้นที่กล่าวมา เราจะกล่าวถึงทฤษฎีและส่วนการออกแบบในบทต่อ ๆ ไป

# บทที่ 2
# ทฤษฎีและหลักการ

## 2.1 โครงสร้างของระบบคอมพิวเตอร์

หากพิจารณาถึงระบบคอมพิวเตอร์แล้ว จะไม่ใช่เพียงคอมพิวเตอร์ที่เราเห็นทั่วไปเท่านั้น เครื่องมืออิเล็กทรอนิกส์บางอย่างก็ถือว่าเป็นระบบคอมพิวเตอร์ได้ ถ้าจะพิจารณาว่าระบบที่เรากำลังศึกษาอยู่นี้เป็นระบบคอมพิวเตอร์หรือไม่ ให้พิจารณาจากโครงสร้างของมัน ซึ่งระบบคอมพิวเตอร์จะต้องมีส่วนประกอบดังนี้

1.หน่วยประมวลผลกลาง

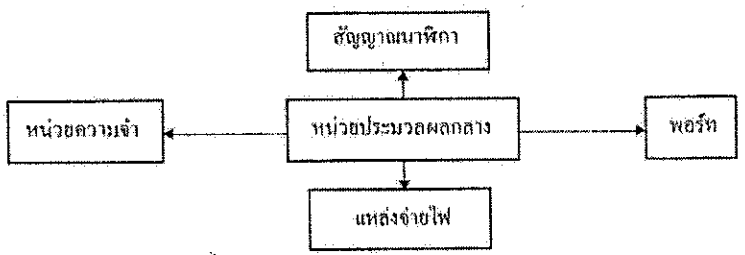
2.หน่วยความจำ

3.อินพุต

4.เอาต์พุต

5.โปรแกรม

ระบบคอมพิวเตอร์จะทำงานได้นอกจากต้องใช้พลังงานไฟฟ้าแล้ว ยังต้องใช้สัญญาณนาฬิกาอีกด้วยระบบคอมพิวเตอร์เบื้องต้นสามารถแสดงได้ดังบล็อกไดอะแกรมดังรูปที่ 2.1 ซึ่งมีรายละเอียดดังนี้




รูปที่ 2.1 ไดอะแกรมโครงสร้างของคอมพิวเตอร์

### 2.1.1 หน่วยประมวลผลกลาง(CPU : Central Processing unit)

เป็นส่วนที่สำคัญมากในระบบคอมพิวเตอร์การทำงานของคอมพิวเตอร์จะเกิดขึ้นได้เมื่อหน่วยประมวลผลกลางตีความคำสั่งที่อ่านจากหน่วยความจำ และกระทำตามคำสั่งนั้น ซึ่งสามารถเปรียบได้กับสมองของคน ภายในซีพียูยังมีหน่วยที่ทำงานทางด้านตรรกศาสตร์ และลอจิก (Arithmetic Logic Unit) หรือ ALU อีกด้วย

### 2.1.2 หน่วยความจำ(MEMORY)

ในระบบคอมพิวเตอร์ จะมีหน่วยความจำที่ใช้กันอยู่สองชนิดคือ หน่วยความจำที่เป็น ROM (Read Only Memory) ซึ่งเป็นหน่วยความจำแบบอ่านอย่างเดียวไม่มีการเขียนเกิดขึ้น และข้อมูลในหน่วยความจำจะไม่หายแม้ไม่มีไฟเลี้ยง หน่วยความจำชนิดนี้จะใช้เก็บโปรแกรมที่จะทำให้เกิดการทำงานในระบบคอมพิวเตอร์ หน่วยความจำอีกชนิดคือหน่วยความจำที่เป็น RAM (Random Access Memory) เป็นหน่วยความจำที่สามารถอ่านและเขียนข้อมูลได้ แต่ข้อมูลจะหายไปถ้าไม่มีไฟเลี้ยง หน่วยความจำชนิดนี้จะใช้เก็บข้อมูลที่เกิดจากการประมวลผลจากหน่วยประมวลผลกลาง

### 2.1.3 พอร์ตอินพุต/เอาต์พุต(I/O Port)

พอร์ตเป็นเส้นทางที่ระบบคอมพิวเตอร์ใช้ติดต่อกับอุปกรณ์ภายนอก เช่นเครื่องพิมพ์ คีย์บอร์ดเป็นต้น โดยพอร์ทจะเชื่อมต่อกับอุปกรณ์ภายนอกชนิดต่าง ๆ โดยวงจร Interface

คอมพิวเตอร์จะทำงานได้จะต้องมีโปรแกรมเป็นตัวกำหนดลักษณะการทำงาน เพื่อติดต่อกับภายนอกทางอินพุตและเอาต์พุต โดยมีหน่วยประมวลผลกลางเป็นตัวประมวลผลคำสั่งที่อ่านได้จากโปรแกรมหน่วยประมวลผลกลางจะเป็นตัวหลักของระบบคอมพิวเตอร์ซึ่งถูกออกแบบไว้ในชิพเพียงตัวเดียว เรียกว่า ไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเราสามารถโปรแกรมให้มันทำงานเฉพาะอย่างได้ ดังเช่นอุปกรณ์ทั่ว ๆ ไปที่ควบคุมด้วยระบบคอมพิวเตอร์

## 2.2 ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์(Microcontroller)เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรรับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

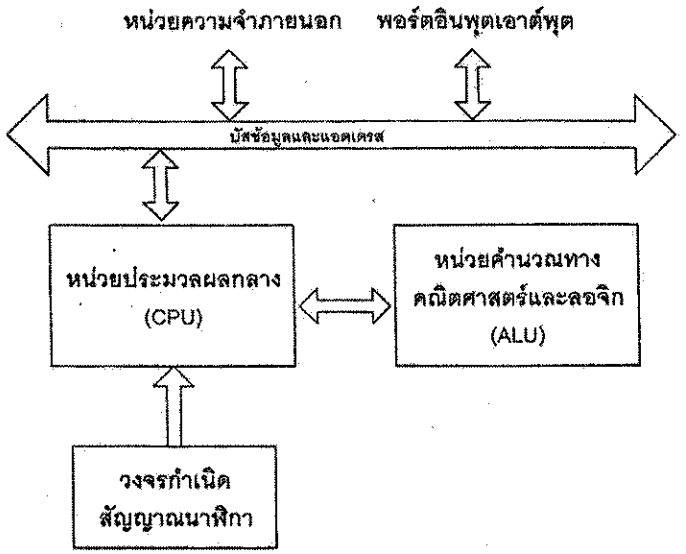
ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันคือ “ไมโคร” (Micro) ซึ่งหมายถึงไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็กภายในประกอบด้วย หน่วยประมวลผลกลางหรือ ซีพียู (CPU : Central Processing unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์” (Controller) หมายถึงอุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

**2.3 ความแตกต่างของไมโครคอนโทรลเลอร์กับไมโครโปรเซสเซอร์**

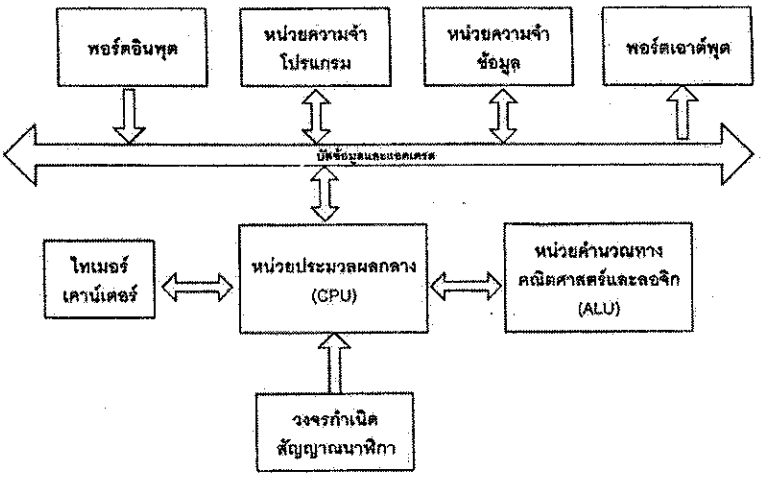
ในรูปที่ 2.2 แสดงโครงสร้างของไมโครโปรเซสเซอร์ ซึ่งประกอบด้วย หน่วยประมวลผลกลาง, หน่วยคำนวณทางคณิตศาสตร์และลอจิก บัสข้อมูลและแอดเดรสสำหรับติดต่อกับหน่วยความจำภายนอก และวงจรกำเนิดสัญญาณนาฬิกา นั่นหมายความว่า การใช้งานไมโครโปรเซสเซอร์จะต้องเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอก และถ้าหากต้องการเชื่อมต่ออุปกรณ์อินพุตและเอาต์พุต ต้องอาศัยอุปกรณ์ที่เรียกว่า ไอซีขยายพอร์ต (port expander) ทำให้การสร้างระบบควบคุมจึงต้องการอุปกรณ์จำนวนมาก ส่งผลให้ขนาดของระบบใหญ่พอสมควร

ในรูปที่ 2.3 แสดงโครงสร้างของไมโครคอนโทรลเลอร์ จะเห็นได้อย่างชัดเจนว่า ภายในไมโครคอนโทรลเลอร์มีอุปกรณ์พื้นฐานเหมือนกับไมโครโปรเซสเซอร์ หากแต่จะบรรจุหน่วยความจำโปรแกรม หน่วยความจำข้อมูล และพอร์ตอินพุตไว้ภายในพร้อมสรรพ ผู้ใช้งานจึงเพียงแต่เขียนโปรแกรมควบคุมลงบนตัวไมโครคอนโทรลเลอร์ แล้วต่ออุปกรณ์ที่ใช้ในการสร้างวงจรกำเนิดสัญญาณนาฬิกา อาทิ คริสตอล ตัวเก็บประจุเป็นต้น สุดท้ายเชื่อมต่ออุปกรณ์เอาต์พุตเข้ากับขาพอร์ต เพียงเท่านี้ก็สามารถใช้งานได้แล้ว

อย่างไรก็ตามไมโครคอนโทรลเลอร์ก็สามารถเชื่อมต่อกับหน่วยความจำภายนอกได้เช่นกัน โดยพิจารณาให้หน่วยความจำภายนอกนั้นเป็นอุปกรณ์อินพุตเอาต์พุตตัวหนึ่ง แล้วใช้ขาพอร์ตที่มีอยู่ติดต่อ ในตารางที่ 2.1 เป็นตารางสรุปความแตกต่างที่สำคัญระหว่างไมโครโปรเซสเซอร์และไมโครคอนโทรลเลอร์




รูปที่ 2.2 โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์




รูปที่ 2.3 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

| 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 |

ตาราง 2.1 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์

**2.4 จะเลือกใช้ไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์เมื่อใด ?**

ถ้าเป็นการสร้างระบบควบคุมขนาด 8 บิต มีความต้องการเชื่อมต่อกับอุปกรณ์ภายนอกไม่มากนัก(น้อยกว่า 10 แบบ)ควรเลือกใช้ไมโครคอนโทลเลอร์ และถ้าหากต้องมีการประมวลผลข้อมูลจำนวนมาก ต้องใช้หน่วยความจำโปรแกรมสูงถึง 8 กิโลไบต์ ทั้งยังมีความต้องการเก็บรักษาข้อมูลลงในหน่วยความจำข้อมูลมากเป็นกิโลไบต์ควรออกแบบให้ไมโครคอนโทรลเลอร์ ในระบบควบคุมนี้เชื่อมต่อกับหน่วยความจำภายนอก

ถ้าหากต้องใช้งานเก็บข้อมูลมากกว่า 8 บิตตลอดเวลา และต้องการความเร็วในการทำงานสูง ๆ สามารถติดต่อหน่วยความจำได้เป็นจำนวนมาก ๆ ติดต่อกับอุปกรณ์อินพุตเอาต์พุต ได้จำนวนมากมายในเวลาเดียวกันควรเลือกใช้ไมโครโปรเซสเซอร์ ดังนั้นจะเห็นได้จากเครื่องคอมพิวเตอร์ที่ใช้ไมโครคอนโปรเซสเซอร์เป็นหัวใจหลักในการทำงาน ทำให้สามาถเชื่อมต่อกับหน่วยความจำได้มากเป็นหน่วยกิกะไบต์ ขนาดของข้อมูลสูงถึง 64 บิต ความเร็วสูงเป็นหลายร้อยเมกะเฮริตซ์ เป็นต้น

ดังนั้นจึงพอสรุปได้ว่าไมโครคอนโทรลเลอร์เหมาะสำหรับการสร้างระบบควบคุม ในขณะที่ไมโครโปรเซสเซอร์เหมาะสำหรับการสร้างระบบประมวลผลข้อมูลความเร็วสูง และ ระบบควบคุมที่มีขนาดใหญ่มาก ๆ

**2.5 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช**

ไมโครคอนโทรลเลอร์ MCS-51 มีหน่วยความจำภายในเป็นแบบแฟรช (flash memory) ของ Atmel Corporation มีเบอร์ขึ้นต้นด้วย AT89 เหตุผลที่ใช้ไมโครคอนโทรเลอร์แบบนี้ในการเรียนรู้เพื่อใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มีกันด้วยหลายประการดังนี้

1.หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ เป็นแบบแฟลซ ทำให้สามารถลบและเขียนใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของ ไมโครคอนโทรเลอร์ชิปเดี่ยว ไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้ งานพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์

2.ต้นทุนและเวลาในการพัฒนาระบบไมไครคอนโทรลเลอร์ลดลงอย่างมากเนื่องจากไม่ต้องใช้เครื่องมือพัฒนาจำพวกอีมูเลเตอร์และเครื่องโปรแกรมอีพรอม

3.บริษัทผู้ผลิตได้ทำการผลิตไมโครคอนโทรลเลอร์ตระกูลนี้ออกมาหลายเบอร์ และมีความสามารถแตกต่างกันไป ทำให้มีทางเลือกในการใช้งานสูง

4.ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ ทำให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้เป็นอย่างดี

5.ในบางเบอร์ของไมโครคอนโทรลเลอร์ที่ผลิตโดย Atmel สามารถทำการโปรแกรมข้อมูลในหน่วยความจำโปรแกรมได้โดยที่ไม่ต้องถอดตัวไมโครคอนโทรลเลอร์ออกมาทำการโปรแกรมใหม่ หรือเรียกว่า การโปรแกรมในวงจร หรือ ในระบบ (In-system programming) ทำให้การพัฒนาหรือการซ่อมบำรุง ตลอดจนการปรับปรุงหรืออัปเกรดข้อมูลในหน่วยความจำโปรแกรมทำได้อย่างสะดวก ภายใต้งบประมาณที่ไม่สูงมากนัก

6.ชุดคำสั่งและสถาปัตยกรรมพื้นฐานเหมือนกับไมโครคอนโทรลเลอร์ MCS-51 ของผู้ผลิตอื่น ไม่ว่าจะเป็นอินเทล ซีเมนส์ หรือ ดัลลัส

**2.5.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51อนุกรม AT89xx**

-เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต

-ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง

-หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม

-ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต

-มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์

-ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว

-สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท

-สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์

-มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

-มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับอนุกรม AT89Sxx

-มีวอตซ์ด็อกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx

ในรูป 2.4 เป็นโครงสร้างพื้นฐานของ ไมโครคอนโทรลเลอร์ ตระกูล MCS-51อนุกรม AT89Cxx จะเห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา หากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว

สำหรับในรูปที่ 2.5 เป็นโครงสร้างพื้นฐานของอนุกรม AT89Sxx จะเห็นได้ว่า มีส่วนประกอบที่เพิ่มเติมแตกต่างจาก AT89Cxx อยู่หลายส่วน อาทิ วงจรเชื่อมต่ออนุกรมแบบ SPI ซึ่งในไมโครคอนโทรลเลอร์อนุกรมนี้ใช้ในการเขียนข้อมูลลงในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิปออกไปจากระบบหรือเรียกว่าการโปรแกรมในวงจร ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตที่เพิ่ม

เติมเข้ามาอีกหนึ่งตัวเป็นไทเมอร์ 2 และวงจรวอตว์ด็อกที่ใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู

ในตารางที่ 2.2 แสดงรายระเอียดบางส่วนของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แต่ละเบอร์ที่ Atmel ผลิตขึ้น และมีใช้งานอยู่ในปัจจุบัน

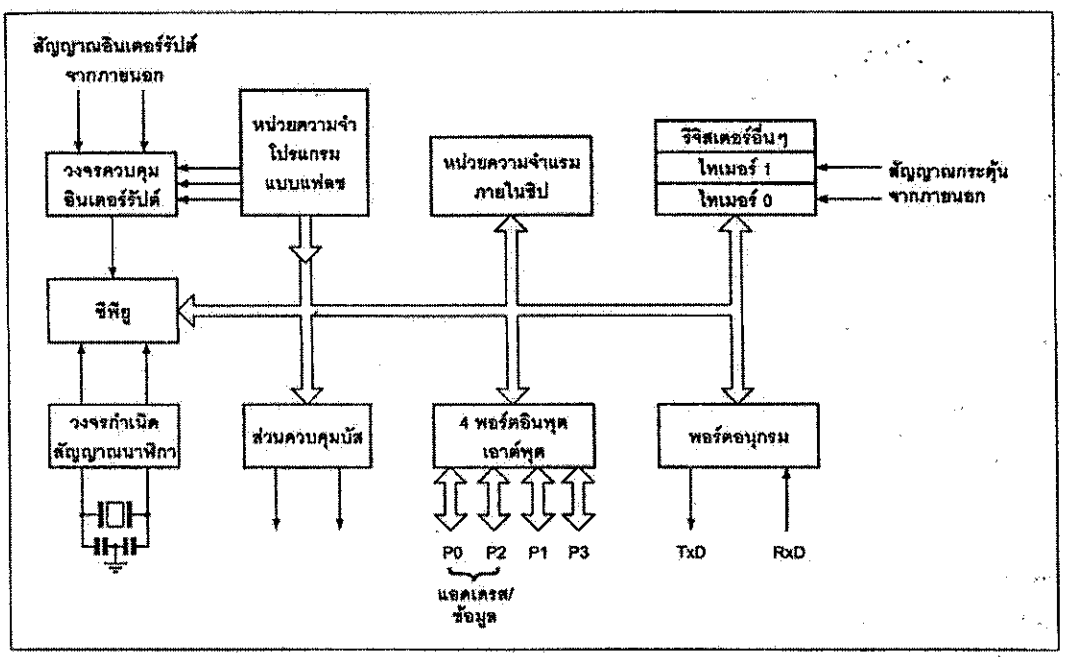
### 2.5.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรมและขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2-6 และ 2-7 โดยมีรายระเอียดขั้นต้นดังนี้

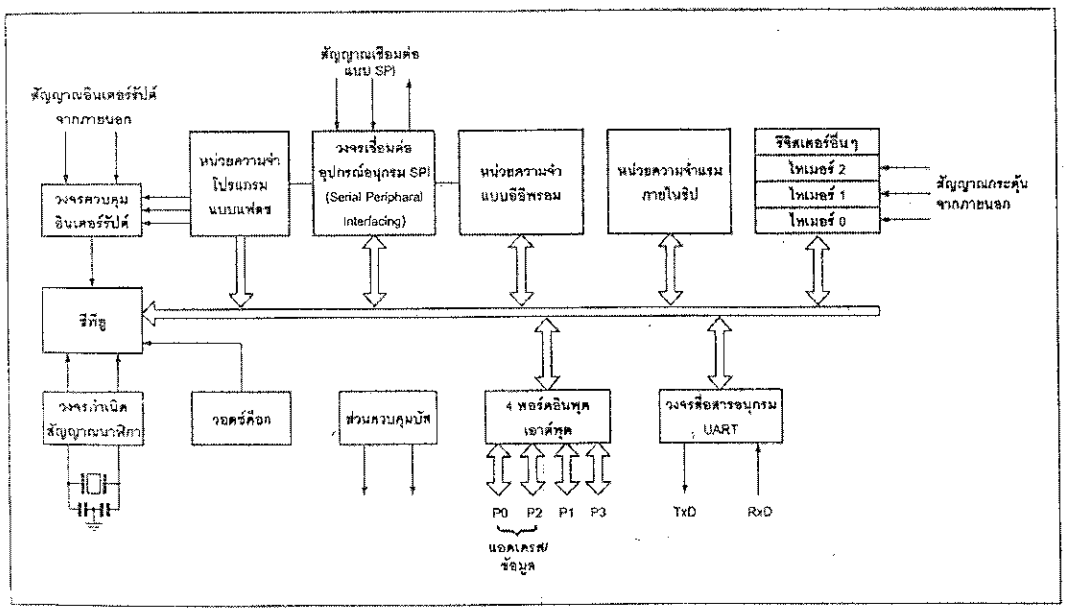
ขา Vcc ใช้สำหรับต่อไฟลี้ยง +5v

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0(p0.0-p0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก(A0-A7) และขาข้อมูล(D0-D7)โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล




รูปที่ 2.4 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89CXX




รูปที่ 2.5 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชในอนุกรม AT89SXX

| เบอร์ของไมโครคอนโทรลเลอร์ | หน่วยความจำโปรแกรม | หน่วยความจำข้อมูล | จำนวนไทเมอร์/เคาน์เตอร์ 16 บิต |
|---|---|---|---|
| AT89C1051 | แบบแฟลช ขนาด 1 กิโลไบต์ | แรม 64 ไบต์ | 1 |
| AT89C2051 | แบบแฟลช ขนาด 2 กิโลไบต์ | แรม 128 ไบต์ | 2 |
| AT89C51 | แบบแฟลช ขนาด 4 กิโลไบต์ | แรม 128 ไบต์ | 2 |
| AT89C52 | แบบแฟลช ขนาด 8 กิโลไบต์ | แรม 256 ไบต์ | 3 |
| AT89C55 | แบบแฟลช ขนาด 20 กิโลไบต์ | แรม 256 ไบต์ | 3 |
| AT89S8252 | แบบแฟลช ขนาด 8 กิโลไบต์ | แรม 256 ไบต์<br>อีอีพรอม 2 กิโลไบต์ | 3 |
| AT89S53 | แบบแฟลช ขนาด 12 กิโลไบต์ | แรม 256 ไบต์ | 3 |

ตาราง 2.2 รายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชที่ Atmel ผลิตขึ้นและใช้ในการอ้างอิง

ขาพอร์ต(P1.0-P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุตและเอาต์พุตสำหรับการใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

ขาพอร์ต(P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุตและเอาต์พุตสำหรับการใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย( float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานนานติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก(A8-A15)

ขาพอร์ต(P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุตและเอาต์พุตสำหรับการใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย( float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังมีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา txD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INT0

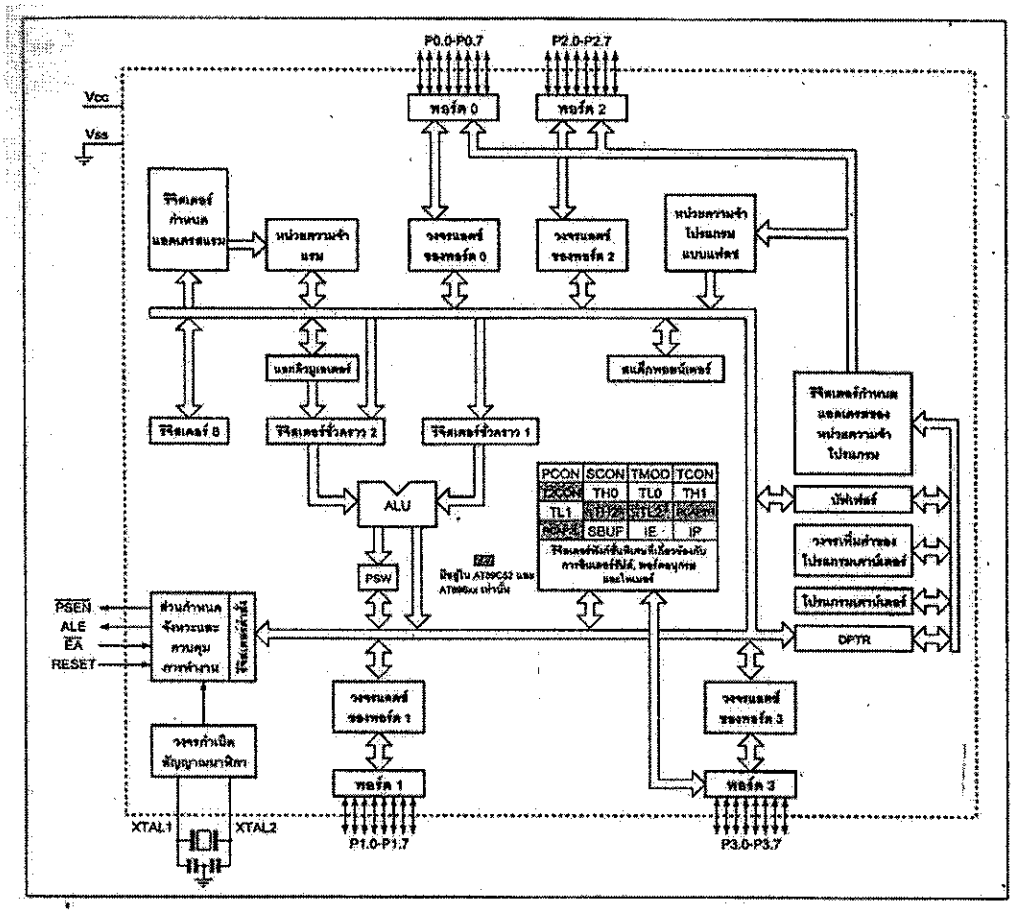
P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1

P3.4ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง0หรือขา T0

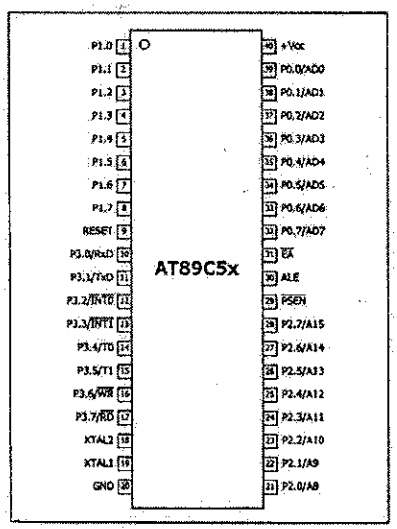
P3.5ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก




รูปที่ 2.6 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชของ Atmel




รูปที่ 2.7 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5XX

**ขารีเซต(Reset)**ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรูเซตอย่างน้อย 2 แมชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

**ขาALE/PROG(Address Latch Enable/Program pulse input)**เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้น ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

**ขา PSEN (Program Store Enable)** ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใด ๆ ออกมา

**ขา EA/Vpp (External Access enable/Programming voltage input)** ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ ไมโครคอนโทรลเลอร์ ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

**ขา XTAL1 และ XTAL2** เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

### 2.5.3 โครงสร้างการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิน 4 พอร์ตคือ พอร์ต 0-3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออกทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชมีวงจรแลตซ์และวงจรขับตลอดจนบัฟเฟอร์อินพุต ดังแสดงให้เห็นในสถาปัตยกรรมรูปที่ 2-6

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางขานอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์MCS-51แบบแฟลชเบอร์ใดสรุปได้ในตารางที่ 2-3

ในรูปที่ 2-8 แสดงวงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช โดยในรูปที่ 2-8(ก) เป็นวงจรของพอร์ต 0 วงจรแลตซ์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจร

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

ดีฟลิปฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตซ์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตซ์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิปฟลอป ในขณะที่ข้อมูลจะผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิปฟลอป

ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่า ต้องการใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตปกติหรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์

เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

ในรูปที่ 2-8(ข) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์ เนื่องจากพอร์ตนี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์นี้แทน สำหรับรายละเอียดของวงจรพูลอัปแสดงในรูปที่ 2-9

ในรูปที่ 2-8(ค) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรพูลอัปเพิ่มเข้ามา ส่วนในรูปที่ ในรูปที่ 2-8(ง) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์และวงจรอินพุตเอาต์พุตเมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

**2.5.4 การใช้งานเป็นพอร์ตอินพุต**

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์MCS-51แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์MCS-51แบบแฟลช

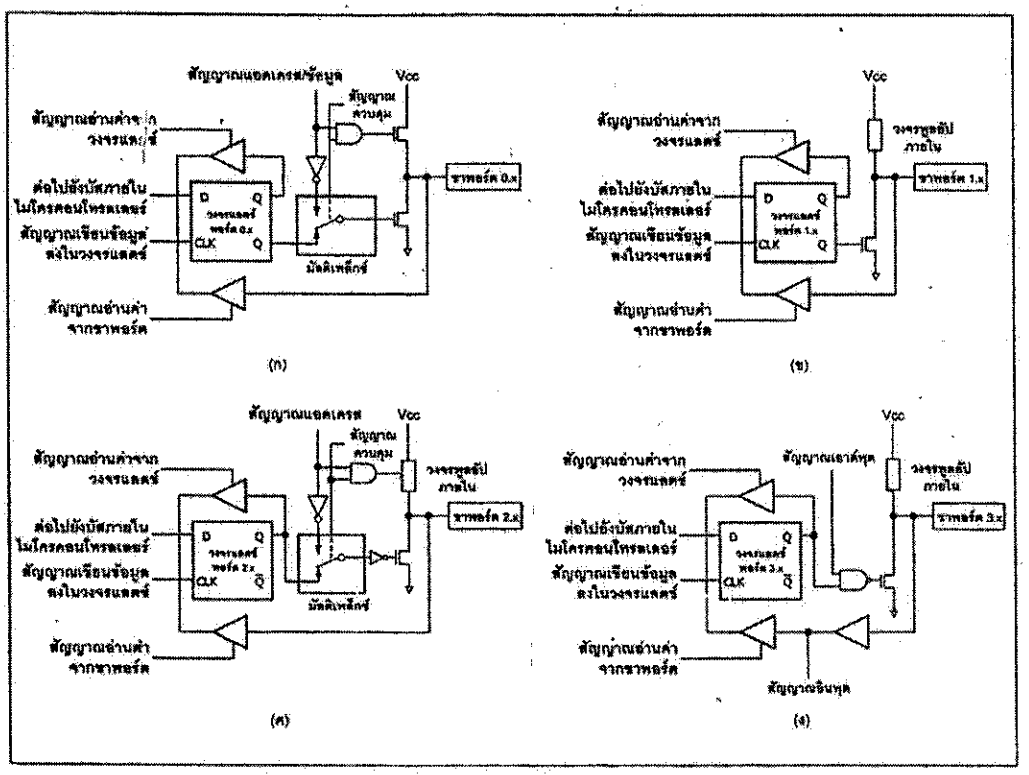
ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล “1” มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการจับสัญญาณเอาต์พุตของบิตนั้น ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรงส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น “1” สามารถรับสัญญาณลอจิก “0” จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์MCS-51แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก “0”

| ชื่อขา | เบอร์ของไมโครคอนโทรลเลอร์ | หน้าที่พิเศษ |
|---|---|---|
| P1.0 | AT89C52/AT89Sxx | ขา T2 เป็นขาอินพุตนับค่าของไทเมอร์/เคาน์เตอร์ 2 และเป็นขา |
| P1.1 | AT89C52/AT89Sxx | และควบคุมทิศทางของสัญญาณ |
| P1.4 | AT89Sxx | ขา $\overline{SS}$ (Slave Select) เป็นขาเลือกการติดต่อในกรณีที่ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบการติดต่อแบบ SPI |
| P1.5 | AT89Sxx | ขา MOSI (Master data output, 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 |

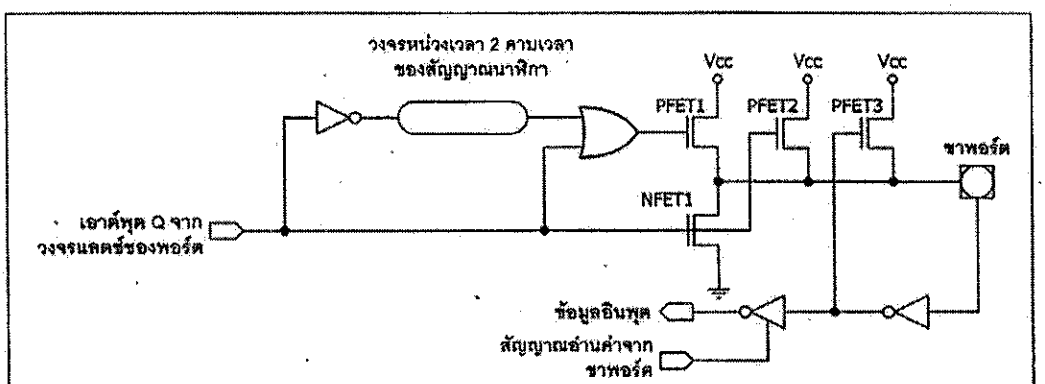
ตาราง 2.3 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชของ Atmel

### 2.5.5 การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล “0” ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล “0” ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับเฟต ทำให้เฟตทำงาน ที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก “0” ขึ้น ในทางตรงข้ามหากต้องการส่งข้อมูล “1” ออกไป ก็ให้เขียนข้อมูล “1” ไปยังวงจรแลตช์ วงจรขับก็หยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก “1” ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต




รูปที่ 2.8 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช




วงจรพูลอัปประกอบด้วยเฟ็ตชนิดพีแชนเนล 3 ตัวคือ PFET1-PFET3 โดย NFET1 จะทำงานเมื่อได้รับลอจิก "1" จากขา Q และหยุดทำงานเมื่อได้รับลอจิก "0" วงจรพูลอัปจะเริ่มต้นทำงานเมื่อ NFET1 ได้รับลอจิก "1" PFET1 จะทำงานนานประมาณ 2 คาบเวลาของสัญญาณนาฬิกาภายใน หลังจากที่เกิดการเปลี่ยนแปลงจากลอจิก "0" เป็นลอจิก "1" ในขณะที่ PFET1 ทำงาน จะทำให้ PFET3 ทำงานตามไปด้วย ทำให้เกิดการพูลอัปขาพอร์ต

รูปที่ 2.9 วงจรพูลอัปภายในพอร์ทของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช

**2.6 การเชื่อมต่อกับหน่วยแสดงผล**

เมื่อต้องการให้ MCS-51 หรือระบบคอมพิวเตอร์ทั่วไปแสดงผลข้อมูลสามารถทำได้หลายวิธี ตัวอย่างเช่น ถ้าต้องการแสดงระดับลอจิก "0" หรือ "1" อาจใช้หลอด LED ในการแสดงผลแบบติดดับถ้าต้องการแสดงเป็นตัวเลขอาจใช้หลอด LED แบบ 7 ส่วน (7 Segment)หากต้องการแสดงเป็นภาพอาจต้องใช้ LED แบบ DOT เมทริกซ์ เป็นต้น แต่ในโครงงานจะแสดงการเชื่อมต่อพอร์ตกับอุปกรณ์แสดงผลพื้นฐาน

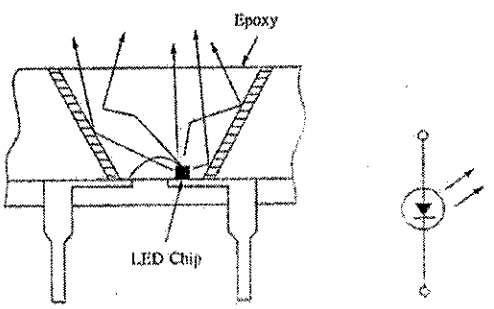
**2.6.1 การเชื่อมต่อกับไดโอดเปล่งแสง**

ไดโอดเปล่งแสงหรือ LED (Light Emitting Diode) สามารถนำมาเชื่อมต่อกับพอร์ตขนานของ MCS-51 ได้ เนื่องจาก LED เป็นอุปกรณ์พื้นฐานที่สามารถแสดงผลจากระบบไมโครคอมพิวเตอร์ได้คุณสมบัติต่าง ๆ ของ LED นั้นจะ Compatible กับระดับสัญญาณของ TTL Logic และยังกินกระแสต่ำเหมาะกับระบบที่ใช้พลังงานจาก Battery อีกด้วย คุณสมบัติต่าง ๆ ของ LED แสดงได้ดังตางรางที่ 2.4

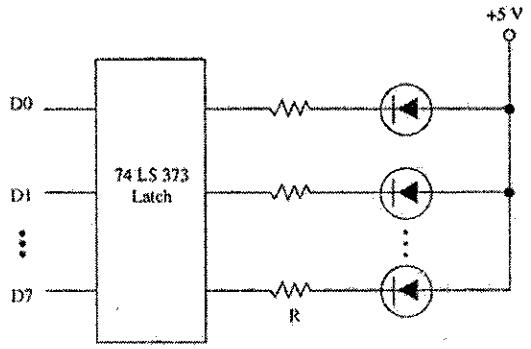
| | |
|---|---|
| ความต่างศักดิ์ | 1.5 - 3 |
| กระแส (mA) | 5 - 25 |
| ความสว่าง (mf) | $10^3$ |
| สี | แดง, เหลือง, เขียว |
| อายุ (h) | $10^5$ |

ตารางที่ 2.4

หลอด LED สร้างจากสารกึ่งตัวนำซึ่งสามารถให้แสงในย่านอุลตราไวโอเลต ถึงอินฟราเรด ได้แต่ในที่นี้เราสนใจในย่าน Visible Spectrum หรือ LED ที่ให้แสงสีในย่านที่ตามองเห็น โดยสีแดงจะสร้างจาก GaAsP สีเหลืองสร้างจาก GaAsP บน GaP สีเขียวสร้างจาก GaP โครงสร้างจของ LED จะแสดงดังรูปที่ 2.10



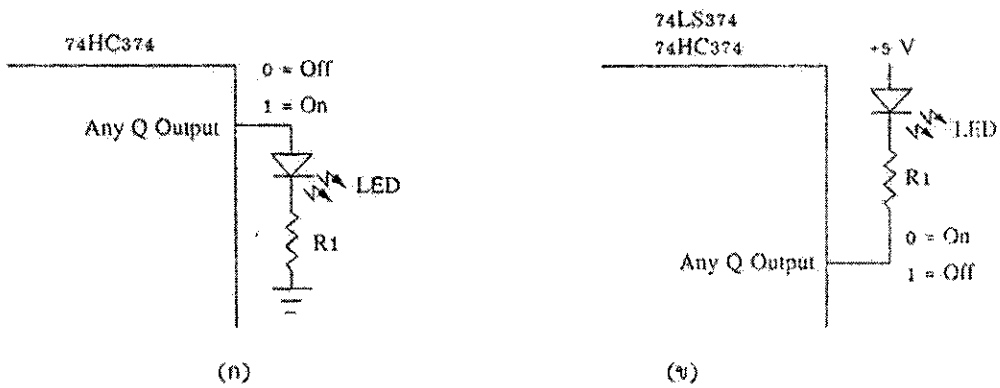

รูปที่ 2.10 แสดงโครงสร้างและสัญลักษณ์ของไดโอดเปล่งแสง



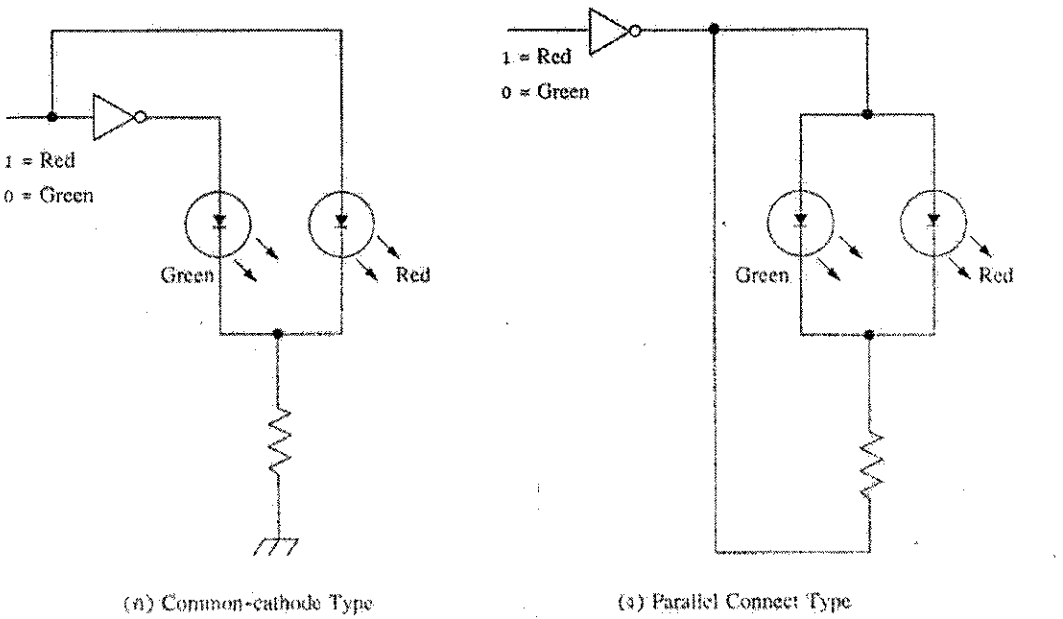

รูปที่ 2.11 แสดงการเชื่อมต่อ 74LS373 กับ LED

### 2.6.2 การเชื่อมต่อหลอด LED

ตัว MCS-51และไอซีพอร์ตขนานเบอร์ 8255 เอาต์พุตของมันไม่สามารถขับหลอด LED ได้ แต่ถ้าเราฝืนเอาไปต่อก็อาจใช้ได้แต่อาจทำให้ชิปพัง เพราะไอซีเหล่านั้นขับกระแสได้ไม่พอ แต่หลอด LED สามารถต่อกับไอซี TTL บางเบอร์ หรือไอซี 74LS373 ได้ โดยการต่อจะต้องมีตัวต้านทานต่ออนุกรมเพื่อให้ได้กระแสในย่านที่ต้องการ ในรูปที่ 12.4 แสดงการต่อพอร์ตกับหลอด LED โดยรูปที่ 12.4(ก) จะสว่างเมื่อเอาต์พุตเป็นลอจิก "1" รูปที่ 12.4(ข) จะสว่างเมื่อเอาต์พุตเป็นลอจิก "0" สำหรับรูปที่ 12.4 (ค) และ (ง) จะใช้บัฟเฟอร์มาช่วยเมื่อต่อกับพอร์ตของระบบไมโครคอนโทรลเลอร์ทั่ว ๆ ไป สำหรับค่าความต้านทานที่นิยมใช้สำหรับ LED แต่ละสีแสดงได้ดังตารางที่ 2.5



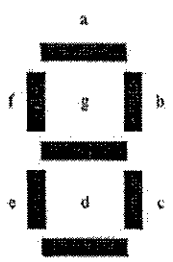

รูปที่ 2.12 แสดง LED แต่ละแบบ และการต่อเข้ากับพอร์ต




รูป 2.13

### 2.6.3 การเชื่อมต่อกับ LED 7 ส่วน

ต่อไปจะกล่าวถึงการเชื่อมกับ LED ที่แสดงผลแบบตัวเลขได้ที่เรียกว่า LED 7 ส่วน (7 segment Display) ซึ่งมีทั้งแบบคาโทดร่วม (Commom-cathode)และแบบอาโนดร่วม(Common-anode) LED7 ส่วนนี้จะเป็นการรวม LED7 หลอดประกอบกันให้สามารถแสดงเป็นตัวเลขได้ถ้าเป็นชนิดที่นำขาคาโทดของหลอด LED ทุกตัวมารวมกันเรียกว่าแบบคาโทดร่วม แบบอาโนดร่วมก็ทำนองเดียวกัน แต่ละเซกเมนต์จะมีชื่อขาดังรูปที่2.14




รูปที่ 2.14 แสดงสัญลักษณ์ของแต่ละเซกเมนต์

ถ้าเราเชื่อมต่อแต่ละขากับบัสข้อมูลของไมโครคอนโทรลเลอร์ โดยแต่ละบิตจะต่อกับขาของหลอด LED 7 ส่วนดังตารางที่ 12.3 ถ้าหากต้องการให้ LED 7 ส่วนแสดงตัวเลขและอักษรต่าง ๆ เราจะต้องส่งข้อมูลให้แต่ละเซกเมนต์สว่างหรือดับให้ประกอบเป็นอักษรต่าง ๆ ดังนั้นข้อมูลที่ส่ง

ไปที่พอร์ตจะเป็นตัวกำหนดตัวอักษรที่จะแสดงบน LED 7 ส่วน ตัวอักษรและค่าต่าง ๆ ที่ส่งออกมามีความสัมพันธ์กันดังแสดงในตารางที่ 2.5

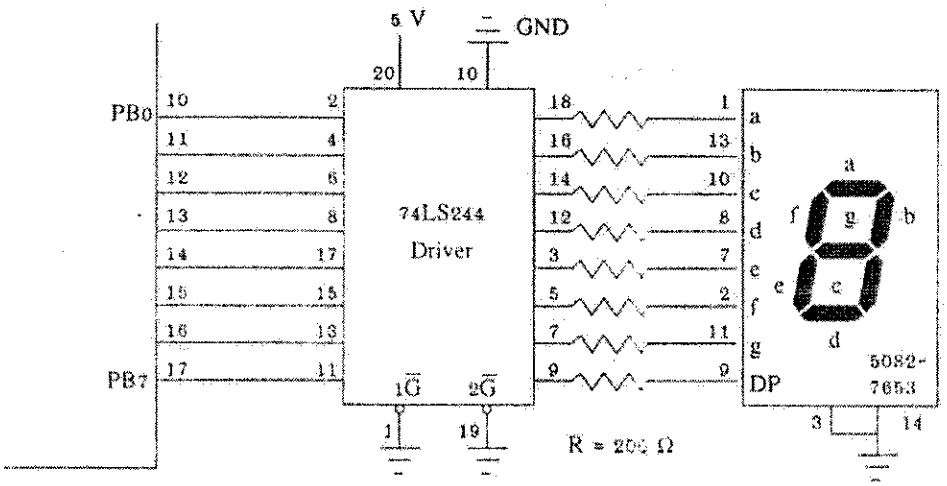
| ตำแหน่งบิต | ตำแหน่งเซกเมนต์ |
|---|---|
| 7 | - |
| 6 | g |
| 5 | f |
| 4 | e |
| 3 | d |
| 2 | c |
| 1 | b |
| 0 | a |

ตารางที่ 2.5 แสดงตำแหน่งขาที่จะต่อกับพอร์ต

| แสดงผล | อาโนดร่วม | คาโทดร่วม | แสดงผล | อาโนดร่วม | คาโทดร่วม |
|---|---|---|---|---|---|
| 0 | C0 | 3F | J | E1 | 1E |
| 1 | F9 | 06 | L | C7 | 38 |
| 2 | A4 | 5B | O | C0 | 3F |
| 3 | B0 | 4F | P | 8C | 73 |
| 4 | 99 | 66 | U | C1 | 3E |
| 5 | 92 | 6D | Y | 91 | 66 |
| 6 | 82 | 7D | b | 83 | 7C |
| 7 | F8 | 07 | c | A7 | 58 |
| 8 | 80 | 7F | d | A1 | 5E |
| 9 | 98 | 67 | h | 8B | 74 |
| A | 88 | 77 | n | AB | 54 |
| C | C6 | 39 | o | A3 | 5C |
| E | 86 | 79 | r | AF | 50 |
| F | 8E | 71 | u | E3 | 1C |
| G | 82 | 70 | - | BF | 40 |
| H | 89 | 76 | ? | AC | 53 |
| I | F9 | 06 | BLANK | FF | 00 |

ตารางที่ 2.6แสดงข้อมูลที่ส่งให้ LED 7 ส่วนแสดงเป็นเลขต่าง ๆ

พิจารณาวงจรรูปที่ 2.15 ถ้าหากต่อ LED 7 ส่วนกับพอร์ต A ของ 8255 ซึ่งถอดรหัสไว้ที่หมายเลข 0FC00H โดยจะส่งข้อมูลให้ LED 7 ส่วนกับพอร์ต A ส่วนขาคาโทดร่วมจะต่อกับบิต PC0 ซึ่งมีเกตบัฟเฟอร์ที่เอาต์พุตเป็นแบบ OC ต่ออยู่ด้วย ดังแสดงในรูปที่ 2.15

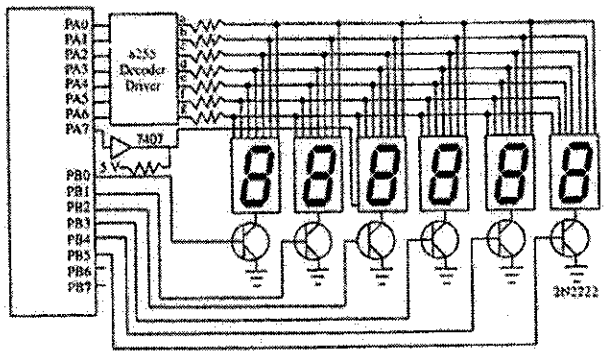



รูปที่ 2.15 แสดงการต่อ LED 7 ส่วนเข้ากับ 8255

ดังที่กล่าวมาแล้วการที่จะให้ LED 7 ส่วนแสดงเป็นตัวเลขต่าง ๆ จะต้องส่งข้อมูลออกมาให้ LED แต่ละส่วนสว่างเป็นเลขนั้น ในที่นี้จะเก็บข้อมูลของรูปแบบที่จะแสดงตัวเลข 0-9 ไว้ในตาราง และใช้การเขียนโปรแกรมแบบเปิดตาราง ขั้นแรกจะเป็นการโปรแกรมให้ 8255 ทำงานเป็นเอาต์พุตทุกพอร์ท จากนั้นให้รีจิสเตอร์ DPTR เก็บค่าตำแหน่งเริ่มต้นของตารางเอาไว้ จากนั้นถ้าจะแสดงตัวเลขอะไรให้เก็บค่านั้นไว้ในรีจิสเตอร์ A แล้วเขียนโปรแกรมให้นำค่าที่อยู่ในตำแหน่งที่ DPTR ชี้อยู่บวกกับค่าใน A ซึ่งจะได้รูปแบบของข้อมูลที่จะทำให้แสดงเป็นตัวเลขที่อยู่ใน A แล้วส่งค่าออกไปที่ LED 7 ส่วน

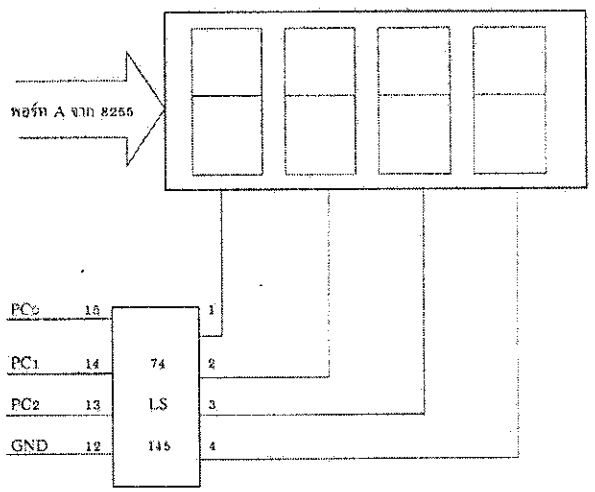
### 2.6.4 การแสดงผลกับ LED 7 ส่วนหลาย ๆ ตัว

ระบบไมโครคอนโทรลเลอร์หากต้องการแสดงผลเป็นตัวเลขหลาย ๆ หลัก จำเป็นต้องใช้ LED 7 ส่วนหลาย ๆ ตัว จากที่กล่าวมาข้างต้น LED 7 ส่วนหนึ่งตัว จะต้องใช้พอร์ตขนาด 8 บิตหนึ่งพอร์ต หากต้องการแสดงตัวเลข 3 หลักจะต้องใช้พอร์ตถึง 3 พอร์ต ซึ่งจะเห็นว่าถ้าจะแสดงหลายหลักจะต้องใช้พอร์ตหลายพอร์ต มีวิธีหนึ่งที่จะแสดงผลหลายหลักได้ โดยจะประหยัดจำนวนพอร์ตและเป็นวิธีที่ใช้กันโดยทั่ว ๆ ไปเรียกว่า Multiplexed Displays โดยต่อขาแต่ละSegment เข้าด้วยกัน จากนั้นจะใช้วิธีสแกนให้ LED ติดทีละหลัก โดยการสแกนแต่ละหลักจะต้องเร็วจนตาไม่สามารถมองการดับของ LED 7ส่วนได้ทัน จนดูเหมือนว่า LED 7 ส่วนทุกตัวติดพร้อมกัน พิจารณาวงจรตามรูปที่ 2.16




รูปที่ 2.16 แสดงการต่อ LED 7 ส่วนหลาย ๆ ตัวกับ 8255

วงจรตามรูปที่ 2.16 เรียกว่า Six-digit Multiplexing Displays ข้อมูลที่ส่งให้ LED แต่ละหลักจะให้พอร์ต A ของ 8255 ถ้า LED ทุกเซกเมนต์สว่างพร้อมกันในแต่ละหลัก(แสดงเลข 3)จะมีกระแสไหลที่ขาคาโทดร่วมประมาณ 60 mA ดังนั้นจะใช้ทรานซิสเตอร์ช่วยในการขับกระแสลงกราวด์ โดยต่อขา B กับพอร์ต B ของ 8255 ตรงนี้จะเรียก Digit Driver เมื่อ 8255 ส่งข้อมูลให้พอร์ต A LED ทุกหลักจะพร้อมที่จะแสดงผล ขึ้นกับว่าในพอร์ต B นั้นบิตใดเป็น “1” หลักนั้นก็จะแสดงผล ถ้าต้องการให้แสดงผลเป็นตัวเลข 123456 จะทำได้โดยขั้นแรกส่งข้อมูลที่แสดงเลข 1 ออกไป จากนั้นทำให้ PB0 เป็น “1” ต่อมาส่งข้อมูลที่ทำให้แสดงตัวเลข 2 ออกไป จากนั้นทำให้ PB1 เป็น “1” ทำไปจนถึงหลักที่ 6 แล้วเขียนโปรแกรมวนลูปโดยหน่วยเวลาให้เหมาะสมก็จะเห็นตัวเลข 123456 พิจารณาวงจรตามรูปที่ 2.17



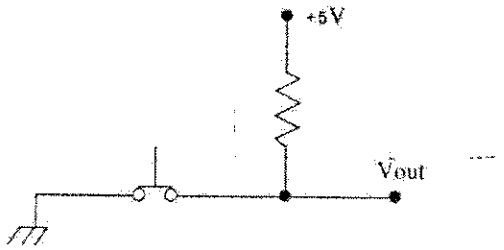

รูปที่ 2.17 แสดงการต่อ LED 7 ส่วน 4 ตัว

จากรูปที่ 12.9 เป็นการต่อ LED 7 ส่วน 4 ตัวเข้ากับพอร์ตเพื่อที่จะแสดงผลแบบ Multiplexed ในการต่อวงจรจะต่อพอร์ต A ของ 8255 เข้ากับทุกSegmentของ LED 7 ส่วน จากนั้นจะใช้พอร์ต C เป็นตัวสแกนทางหลัก โดยใช้ไอซีเบอร์ 74LS145 เป็นตัวถอดรหัส ซึ่ง IC เบอร์นี้

สามารถต่อกับ LED 7 ส่วนได้ถึง 10 ตัว และเอาต์พุตของ IC เบอร์นี้จะเป็นแบบ Open Collector ซึ่งสามารถรับกระแสเข้าได้มากเมื่อ LED ทุกเซกเมนต์ส่วางพร้อมกัน

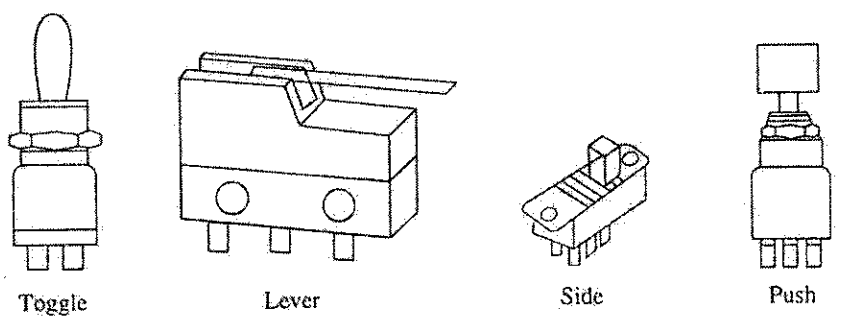
**2.7 การเชื่อมต่อสวิตช์เข้ากับระบบไมโครคอนโทรลเลอร์**

สวิตช์หรือ Keyboard เป็นอัปกรณ์อินพุตพื้นฐานที่ระบบไมโครคอนโทรลเลอร์สามารถรับข้อมูลได้ในบทนี้จะกล่าวถึงการต่อสวิตช์แบบต่าง ๆ ตั้งแต่มีจำนวนน้อยตัวจนถึงการต่อสวิตช์หลาย ๆ ตัวแบบเมทริกซ์การสร้างสวิตช์ให้กับระบบไมโครคอนโทรลเลอร์นั้นอาจทำได้ดังรูป 13.1 โดยการต่อเข้ากับแต่ละบิตของพอร์ทอินพุต ถ้าสวิตช์ ON จะให้ลอจิก "0" แต่ถ้าสวิตช์ OFF จะให้ลอจิก "1"

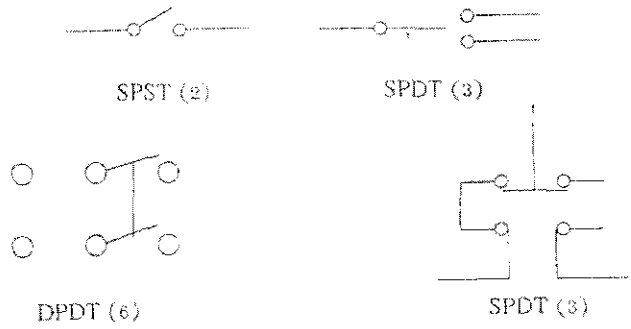



รูปที่ 2.18แสดงการสร้างลอจิกจากสวิตช์

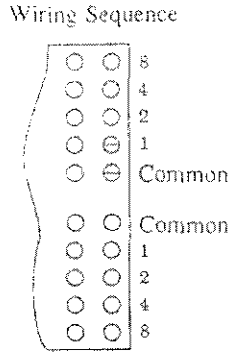
สวิตช์นิยมใช้กันมากมีหลายชนิดดังรูปที่ 2.19 แต่ละแบบอาจแบ่งได้จากจำนวนหน้าสัมผัสและจำนวนขั้วของตัวสวิตช์เอง ได้แก่แบบ Single-Pole/Single-Throw(SPST), Single-Pole/Double-Throw(SPDT), Double-Pole/Double-Throw(DPDT) เป็นต้น สัญลักษณ์แสดงได้ดังรูปที่ 2.20 นอกจากนี้ยังมีสวิตช์ที่มีโครงสร้างภายในเป็นแบบหมุนซึ่งจะให้สัญญาณออกมาเป็นรหัส BCD ได้เลย เรียกว่า Thumbwheel Switch การใช้สวิตช์แบบนี้จะต้องป้อนสัญญาณเข้าที่ขา Common ดังรูปที่ 2.21 ในโครงงานนี้จะกล่าวถึงการต่อสวิตช์แบบ ON-OFF เข้ากับระบบไมโครคอนโทรลเลอร์




รูปที่ 2.19 แสดงตัวอย่างสวิตช์ที่นิยมใช้กัน



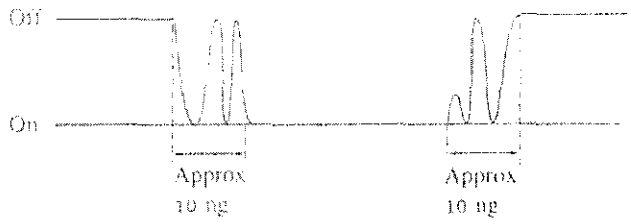

รูปที่ 2.20 แสดงสัญญลักษณ์ของสวิตช์แบบต่าง ๆ




| Wheel Printing | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | 4 | 0 | 0 | 0 | 0 | | | | | 0 | 0 |
| | 2 | 0 | 0 | | | 0 | 0 | | | 0 | 0 |
| | 1 | 0 | | 0 | | 0 | | 0 | | 0 | |
| | Common | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Common | x | x | x | x | x | x | x | x | x | x |
| | 1 | | x | | x | | x | | x | | x |
| | 2 | | | x | | x | | x | x | | |
| | 4 | | | | | x | x | x | x | | |
| | 8 | | | | | | | | | x | x |

รูปที่ 2.21 แสดงไบนารีสวิตช์

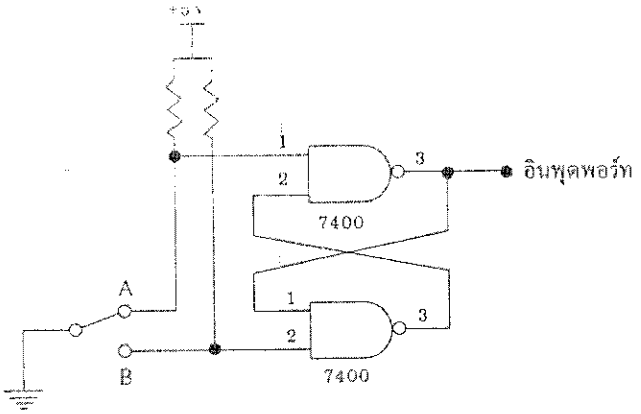
การต่อสวิตช์เข้ากับระบบไมโครคอนโทรลเลอร์จะต้องให้หน้าสัมผัสของสวิตช์สะอาดอยู่เสมอพิจารณาตามรูปที่ 2.18 ถ้าสวิตช์ไม่มีการกดสัญญาณที่ได้จะมีค่าเป็น “1” ถ้าสวิตช์มีการกดข้อมูลที่ได้จะมีค่าเป็น “0” แต่โดยทั่วไปแล้วการเปลี่ยนระดับสัญญาณจาก “1” เป็น “0” จะมีสัญญาณที่ไม่ต้องการเกิดขึ้นดังรูปที่ 2.22 ซึ่งเกิดจากการสั่นของหน้าสัมผัสของสวิตช์ ทำให้เกิดการแกว่งของสัญญาณซึ่งเรียกว่า บาวนซ์(Bounce) อยู่ชั่วระยะหนึ่ง โดยปกติจะมีเวลาประมาณ 5 ถึง 50 มิลลิวินาที ดังนั้นสิ่งที่ MCS-51 ได้รับจากการกดสวิตช์จะไม่ใช่สัญญาณหนึ่งลูก แต่เป็นสัญญาณหลาย ๆ ลูก ซึ่ง MCS-51 อาจได้รับข้อมูลผิดพลาด การแก้ปัญหานี้เรียกว่า การทำ Debounce ซึ่งอาจทำได้ 2 วิธีคือ Hardware Debounce และ Software Debounce




รูปที่ 2.22 แสดงสัญญาณที่เกิดจากการสั่นของหน้าสัมผัสของสวิตช์

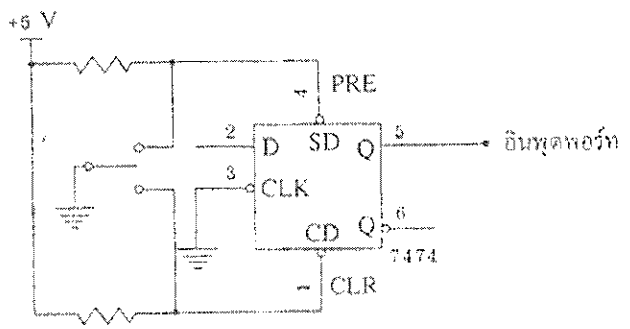
### 2.7.1 Hardware Debounce

การแก้โดยวิธีทางฮาร์ดแวร์วิธีง่าย ๆ วิธีหนึ่งคือการใช้ แนนเกต มาต่อเป็น R-S ฟลิปฟลอป หรืออาจใช้ไอซีเบอร์ 74LS279 ก็ได้ พิจารณารูปที่ 2.23 เมื่อสวิตช์ถูกโยกมาที่ตำแหน่ง A ขาอินพุตของแนนเกตตัวบนจะได้รับลอจิก "0" ซึ่งจะทำให้เอาต์พุตของเกตตัวบนมีลอจิกเป็น "1" เอาต์พุตนี้จะถูกนำไปใช้งาน ขณะเดียวกันก็จะป้อนกลับมาให้กับอินพุตของแนนเกตตัวล่าง ทำให้เอาต์พุตของแนนเกตตัวล่างมีค่าเป็น "0" เมื่อมีการแกว่งของสัญญาณที่จุด A แนนเกตตัวบนจะได้รับลอจิกอินพุตที่เปลี่ยนแปลงแต่ไม่มีผลทำให้เอาต์พุตเปลี่ยนแปลง



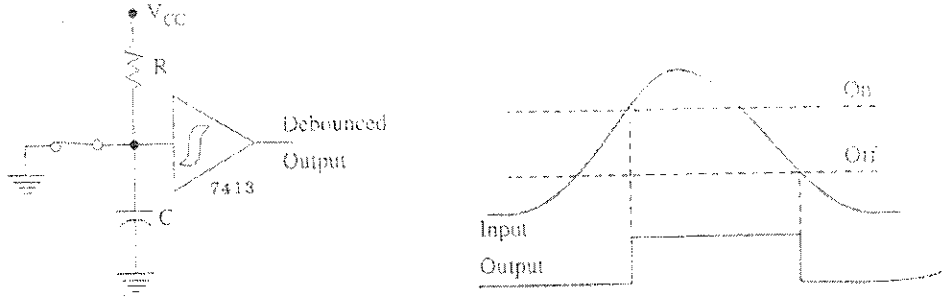

รูปที่ 2.23 แสดงการนำแนนเกตมาต่อเป็น R-S ฟลิปฟลอป เพื่อแก้การสั่นของสวิตช์

การแก้ปัญหาอีกวิธีหนึ่งอาจนำเอา D ฟลิปฟลอป มาช่วยดังแสดงในรูปที่ 2.24 โดยใช้ไอซีเบอร์7447 โดยให้อินพุตจากสวิตช์ต่อเข้าที่ขา Preset และ Clear ถ้าขา Preset เป็น "0" เอาต์พุตจะเป็น "1" ถ้ามีการแกว่งของสัญญาณเกิดขึ้นจะไม่มีผลต่อเอาต์พุต และเมื่อสวิตช์ถูกทำให้ Clear เป็น "0" เอาต์พุตที่ได้จะเป็นลอจิก "0"




รูปที่ 2.24 การฟลิปฟลอปมาแก้ปัญหาสวิตช์

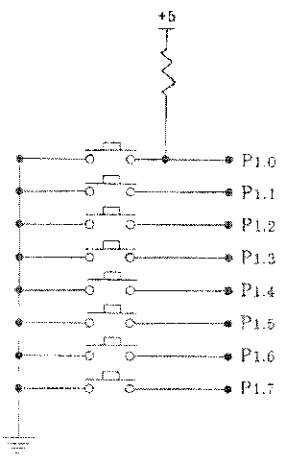
ถ้าใช้สวิตช์แบบ SPST อาจแก้ได้โดยใช้ D ฟลิปฟลอป โดยจะได้สัญญาณอินพุตเมื่อมี Clock เข้าไปหรืออาจใช้ Schmitt Trigger ถ้าสัญญาณอินพุตจากสวิตช์มีลักษณะไม่เหมาะกับระบบดิจิตอล ดังรูปที่ 2.25




รูปที่ 2.25 การแก้ปัญหาสวิตช์โดยใช้ Schmitt Trigger

### 2.7.2 การต่อสวิตช์เข้ากับพอร์ตของ MCS-51 โดยตรง

ตามที่ทราบมาแล้วว่า MCS-51 มีพอร์ตให้ใช้งานอยู่หลายพอร์ต ในที่นี้จะยกตัวอย่างการต่อสวิตช์ 8 ตัวเข้ากับพอร์ต 1 ดังแสดงในรูปที่ 2.26 ระบบนี้เป็นการต่อสวิตช์แบบง่ายที่สุดเหมาะสำหรับระบบที่ไม่ต้องการสวิตช์มากนัก สำหรับการเขียนโปรแกรมจะต้องอ่านค่าจากสวิตช์คืออ่านค่าจากพอร์ต 1 จากนั้นต้องเขียนโปรแกรมตรวจสอบการสั่นของสวิตช์ เมื่อได้ค่าจากการกดสวิตช์แน่นอนแล้วก็ทำการตรวจสอบว่าค่าที่อ่านได้นั้นเป็นค่าจากการกดสวิตช์ใด

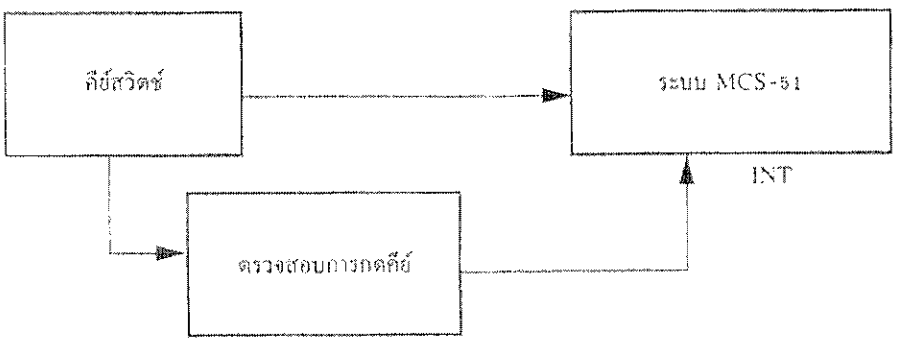



รูปที่ 2.26 แสดงการต่อสวิตช์เข้ากับพอร์ต P1

ต่อไปจะยกตัวอย่างการเชื่อมต่อสวิตช์เข้ากับพอร์ตของ 8255 โดยใช้สวิตช์แบบดิพสวิตช์เชื่อมต่อกับพอร์ต A ของ 8255 การเขียนโปรแกรมอย่างง่าย ๆ อาจทำได้โดยอ่านค่าทางพอร์ต A แล้วทำการตรวจสอบว่าสวิตช์ใด ON โดยนำค่าที่อ่านได้ส่งให้โปรแกรมประมวลผล ซึ่งการเขียนโปรแกรมอาจทำได้หลายวิธีในที่นี้จะยกตัวอย่างง่าย ๆ โดยอ่านค่าเข้ามาแล้วตรวจสอบว่า สวิตช์แรก ON หรือไม่ ถ้าไม่ใช้ตรวจสวิตช์ที่สอง ทำไปเรื่อย ๆ จนครบ สมมติว่าถ้าเรามีโปรแกรมต่าง ๆ หลาย ๆ โปรแกรมและเก็บโปรแกรมเหล่านั้นเป็นโปรแกรมย่อยจากนั้นจะเลือกทำโปรแกรมต่าง ๆ

โดยการกดสวิตช์ เราอาจเชื่อมต่อสวิตช์เข้ากับพอร์ทของ 8255 จากนั้นเขียนโปรแกรมให้อ่านค่าจากสวิตช์เพื่อเลือกทำโปรแกรมย่อยต่าง ๆ

การเชื่อมต่อสวิตช์และการเขียนโปรแกรมที่ผ่านมาเราต้องเขียนโปรแกรมวนลูปเพื่อตรวจสอบการกด Key ตลอดเวลา ซึ่งจะทำให้ระบบไมโครคอนโทรลเลอร์ต้องเสียเวลาส่วนใหญ่ในการตรวจสอบการกดเราอาจต่อสวิตช์อีกแบบได้โดยวิธีอินเทอร์รัพท์(Interrupt)ในภาวะปกติให้ไมโครคอนโทรลเลอร์ทำงานตามปกติไป เช่น อาจทำโปรแกรมแสดงผลต่าง ๆ ถ้ามีการกด Key จะมีสัญญาณไปอินเทอร์รัพท์ MSC-51 จากนั้น MSC-51 จะไปทำโปรแกรมตอบสนองการอินเทอร์รัพท์ต่าง ๆ ตามต้องการ ซึ่งอาจต่อวงจรได้ดังรูปที่ 2.27 ถ้ามีการกดสวิตช์จะมีสัญญาณมาอินเทอร์รัพท์ MSC-51 จากนั้น MSC-51 จะทำโปรแกรมตอบสนองการอินเทอร์รัพท์คือ อ่านค่าจากพอร์ตที่สวิตช์ต่ออยู่ จานั้นจะตรวจสอบว่าสวิตช์ใดกด จากนั้นไปทำโปรแกรมย่อยต่าง ๆ ต่อไป แต่อย่าลืมว่าถ้าเป็นสวิตช์แบบกดติดปล่อยดับจะต้องมีการแก้การสั่นของสวิตช์ด้วย

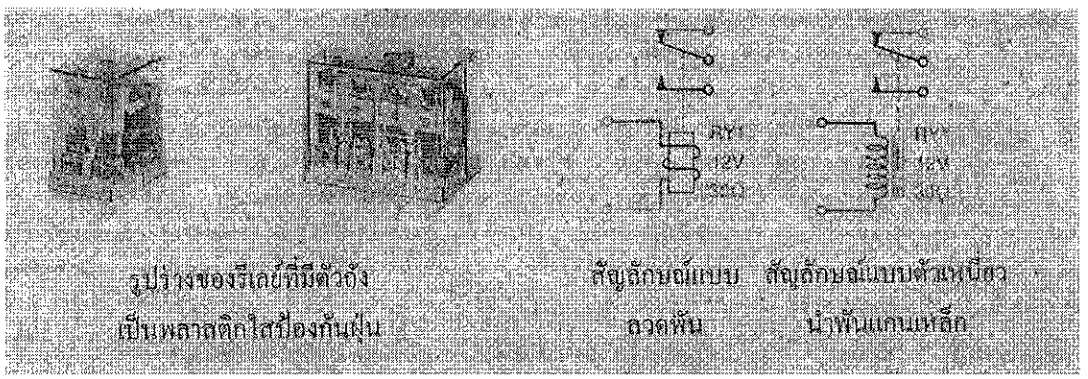



รูปที่ 2.27 แสดงไดอะแกรมการใช้อินเทอร์รัพท์

**2.8 รีเลย์(Relay)**

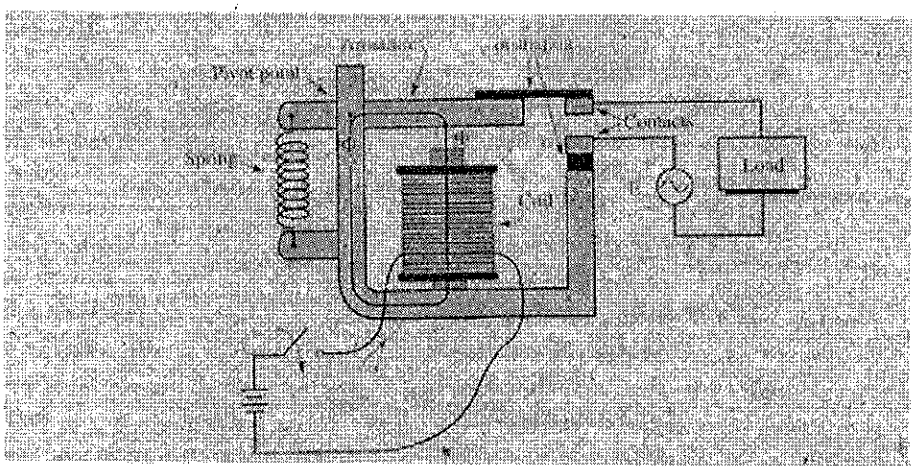
**2.8.1 หลักการเบื้องต้น**

รีเลย์เป็นอุปกรณ์ที่นิยมนำมาทำเป็นสวิตซ์ทางด้านอิเล็กทรอนิกส์ โดยจะต้องป้อนกระแสไฟฟ้าให้ไหลผ่านขดลวดจำนวนหนึ่งเพื่อนำไปควบคุมวงจรกำลังงานสูง ๆ ที่ต่ออยู่กับหน้าสัมผัสหรือคอนแทกต์ของรีเลย์ รูปที่ 2.28 แสดงรูปร่างและสัญลักษณ์ของรีเลย์



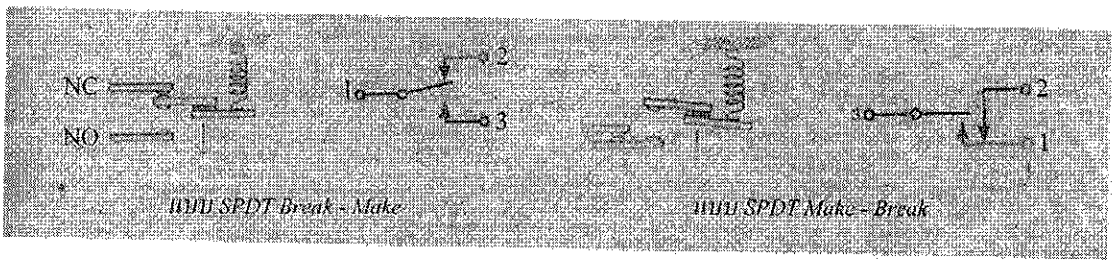

รูปที่ 2.28 แสดงรูปร่างและสัญลักษณ์ของรีเลย์

หลักการทำงานเบื้องต้นของรีเลย์แสดงดังรูปที่ 2.29 การทำงานเริ่มจากปิดสวิตช์กระแสให้กับขดลวด(Coil)โดยทั่วไปจะเป็นขดลวดพันรอบแกนเหล็ก ทำให้เกิดสนามแม่เหล็กไปดูดเหล็กอ่อนที่เรียกว่าอาร์เมเจอร์(Armature)ให้ต่ำลงมาที่ปลายของอาร์เมเจอร์ด้านหนึ่งมักยึดติดกับสปริง(Spring)และปลายอีกด้านหนึ่งยึดติดกับหน้าสัมผัส(Contacts)การเคลื่อนที่อาร์เมเจอร์จึงเป็นการควบคุมการเคลื่อนที่ของหน้าสัมผัส ให้แยกจากหรือแตะกับหน้าสัมผัสอีกอันหนึ่งซึ่งยึดติดอยู่กับที่ เมื่อเปิดสวิตช์อาร์เมเจอร์ก็จะกลับสู่ตำแหน่งเดิม เราสามารถนำหลักการนี้ไปควบคุมโหลด(Load)หรือวงจรอิเล็กทรอนิกส์ต่าง ๆ ได้ตามต้องการ




รูปที่ 2.29แสดงหลักการทำงานเบื้องต้นของรีเลย์

หากตรงข้ามกันคือ หน้าสัมผัส 1 และ 2 จะสัมผัสกันและจะไม่แยกจากกันจนกว่าหน้าสัมผัส 1 และ 3 จะสัมผัสกัน (Make-Break)




รูปที่ 2.32 แสดงหน้าสัมผัสแบบ SPDT แบบ Break Make และ Make Break

**2.8.3 ชนิดของรีเลย์**

รีเลย์ที่ผลิตในปัจจุบันมีอยู่มากมายหลายชนิด ผู้เรียบเรียงจะขอแนะนำรีเลย์ที่นิยมใช้งานและรู้จักกันแพร่หลาย 4 ชนิดเพื่อเป็นแนวทางในการศึกษาในระดับสูงต่อไป

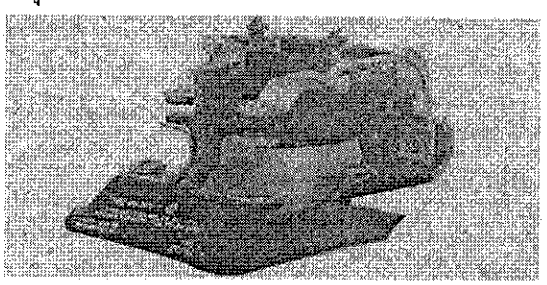
1.อาร์เมเจอร์รีเลย์(Armature Relay)

2.รีดรีเลย์(Reed Relay)

3.รีดสวิตช์(Reed Switch)

4.โซลิดสเตตรีเลย์(Solid-State Relay)

อาร์เมเจอร์รีเลย์(Armature Relay) คือรีเลย์ที่ได้อธิบายหลักการทำงานดังในรูปที่ 2.33 ซึ่งเป็นรีเลย์ที่นิยมใช้กันมากที่สุด บางครั้งเรียกรีเลย์แบบนี้ว่า รีเลย์ชนิดแคลปเปอร์(Clapper Relay)

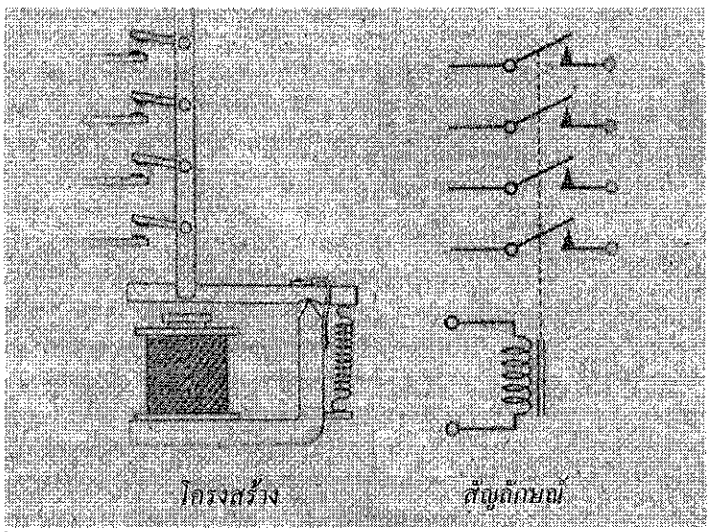


รูปที่ 2.33 แสดงรีเลย์ชนิดอาร์เมเจอร์

รีดรีเลย์(Reed Relay) เป็นรีเลย์ไฟฟ้าที่มีลักษณะเป็นแคปซูลขนาดเล็ก ในรูปที่ 2.34 แสดงภาพตัดขวางของรีเลย์ที่ประกอบด้วยส่วนที่เรียกว่ารีดแคปซูล ซึ่งมีคอยล์พันบนแกนบ๊อบบิ้นรีดแคปซูลจะเป็นหลอดแก้ว ภายในบรรจุก๊าซเฉื่อย หน้าสัมผัสเป็นโลหะผสมแผ่นบาง ๆ ปลายตัด 2 แผ่น วงจรซ้อนแต่ไม่สัมผัสกัน เป็นสวิตช์ชุดเดียวทางเดียวหน้าสัมผัสปกติเปิดวงจร(SPST-ON)

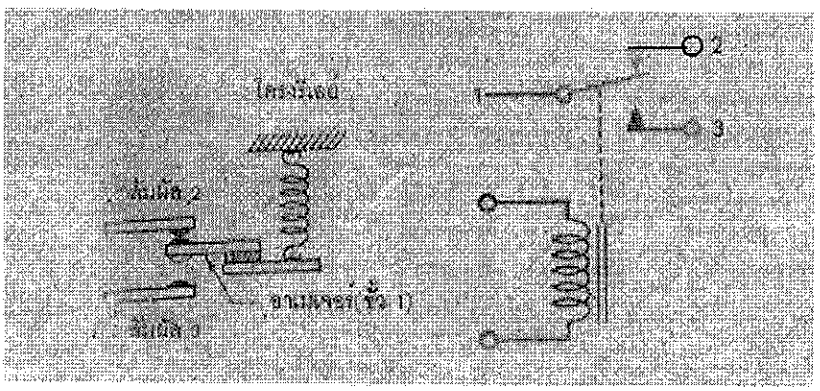
### 2.8.2 หน้าสัมผัสของรีเลย์

รูปที่ 2.29 แสดงรีเลย์ที่มีหน้าสัมผัสเพียงจุดเดียว ปัจจุบันรีเลย์ที่มีขดลวดชุดเดียวสามารถควบคุมหน้าสัมผัสได้หลายชุดดังรูปที่ 2.30 อาร์เมเจอร์อันเดียวถูกยึดอยู่กับหน้าสัมผัสที่เคลื่อนที่ได้ 4 ชุด ดังนั้นรีเลย์ตัวนี้จึงสามารถควบคุมการแตะหรือจากกันของหน้าสัมผัส ได้ถึง 4 ชุด



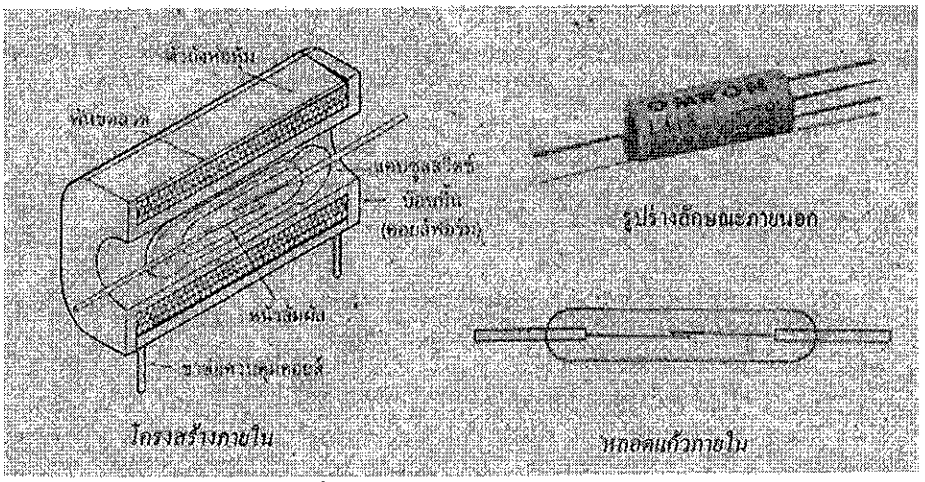

รูปที่ 2.30 แสดงโครงสร้างและสัญลักษณ์ของชุดหน้าสัมผัสแบบ 4PST

แต่ละหน้าสัมผัสที่เคลื่อนที่ได้มีชื่อเรียกว่าขั้ว(Pole)รีเลย์ในรูปที่ 9.3 มี 4 ขั้ว จึงเรียกหน้าสัมผัสแบบนี้ว่าเป็นแบบ 4PST (Four Pole Single Throw) ถ้าแต่ละขั้วที่เคลื่อนที่แล้วแยกจากหน้าสัมผัสอันหนึ่งไปแตะกับหน้าสัมผัสอีกอันหนึ่งเหมือนกับสวิตช์โยก โดยเป็นการเลือกหน้าสัมผัสที่ขนาบอยู่ทั้งสองด้านดังรูปที่ 2.31 หน้าสัมผัสแบบนี้มีชื่อว่า SPDT (Single Pole Double Throw)



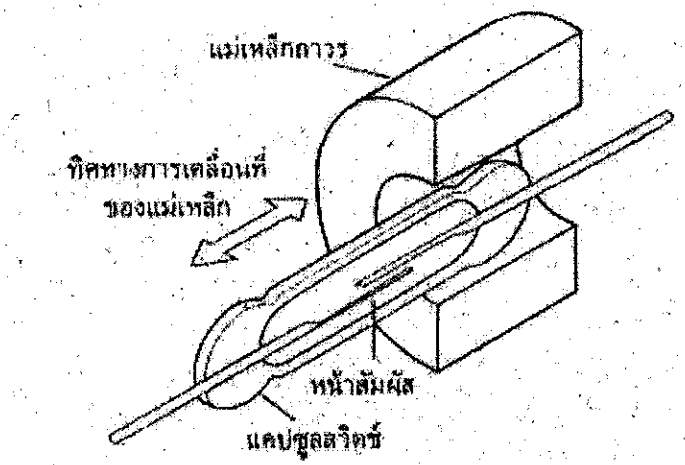

รูปที่ 2.31 แสดงหน้าสัมผัสแบบ SPDT

ในกรณีที่ไม่มีการป้อนกระแสไฟฟ้าเข้าขดลวดของรีเลย์ สภาวะ NO (Normally Open)คือสภาวะปกติหน้าสัมผัสกับขั้วแยกจากกัน ถ้าต้องการให้สัมผัสกันจะต้องป้อนกระแสไฟฟ้าเข้าขดลวด ส่วนสภาวะ NC (Normally-Closed) คือสภาวะปกติหน้าสัมผัสกับขั้วสัมผัสกัน ถ้าต้องการให้แยกกันจะต้องป้อนกระแสไฟฟ้าเข้าขดลวด นอกจากนี้ยังมีแบบแยกก่อนแล้วสัมผัส (Break-Make) หมายถึงหน้าสัมผัสระหว่าง 1 และ 2 จะแยกจากกันก่อนที่หน้าสัมผัส 1 และ 3 จะสัมผัสกันแต่ถ้า



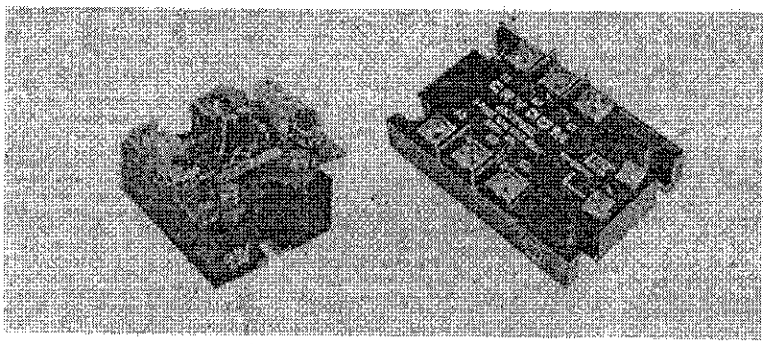

รูปที่ 2.34 แสดงรีเลย์ชนิดรีดรีเลย์

รีดสวิตช์ (Reed-Switch) เป็นรีเลย์อีกชนิดหนึ่งแต่ไม่มีชุดขดลวดสำหรับสร้างสนามแม่เหล็ก การควบคุมการปิดเปิดหน้าสัมผัสของสวิตช์จะใช้สนามแม่เหล็กจากภายนอกมาควบคุมหน้าสัมผัส โครงสร้างภายในของรีดสวิตช์แสดงดังรูปที่ 2.35




รูปที่ 2.35 แสดงรีเลย์ชนิดรีดรีเลย์

โซลิดสเตตรีเลย์(Solid-State Relay) เป็นรีเลย์ที่ไม่มีโครงสร้างทางกลอยู่ภายใน มีขั้วต่ออย่างละ 2 ขั้ว ขั้วอินพุตเป็นขั้วสำหรับป้อนสัญญาณควบคุมเพื่อบังคับให้ขั้วเอาต์พุตปิดหรือเปิดวงจร โดยจะมีการแยกกันทางไฟฟ้าระหว่างขั้วอินพุตและเอาต์พุต

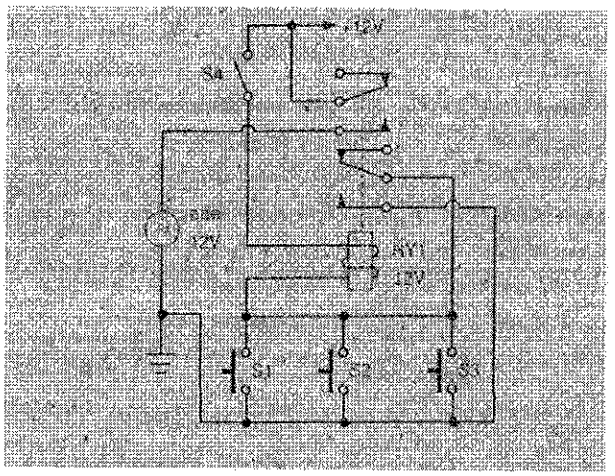


รูปที่ 2.36 แสดงโซลิดสเตตรีเลย์

### 2.8.4 การประยุกต์ใช้งานรีเลย์

ปัจจุบันได้มีการนำรีเลย์ไปใช้ในการทำเป็นสวิตช์ทางด้านอิเล็กทรอนิกส์ในวงจรต่าง ๆ มากมาย ผู้เรียบเรียงจะขอยกตัวอย่าง รายละเอียดและรูปร่างวงจรที่พอเป็นแนวทางในการศึกษาค้นคว้าต่อไปดังนี้

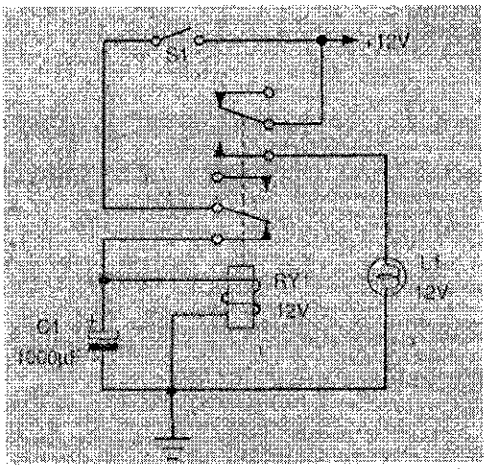
รูปที่ 2.37 เป็นการนำรีเลย์ที่มีหน้าสัมผัส 2 ชุดมาต่อเป็นวงจรกันขโมย โดยที่หน้าสัมผัสของสวิตช์ใช้แบบปกติเปิดวงจร (NO)เมื่อมีการกดสวิตช์ S1, S2 และ S3 ตัวใดตัวหนึ่งจะทำให้ออดส่งเสียงเตือนค้าง โดยมีสวิตช์ S4 ทำหน้าที่รีเซตวงจร




รูปที่ 2.37 แสดงการนำรีเลย์ไปต่อเป็นสวิตช์ในวงจรกันขโมย

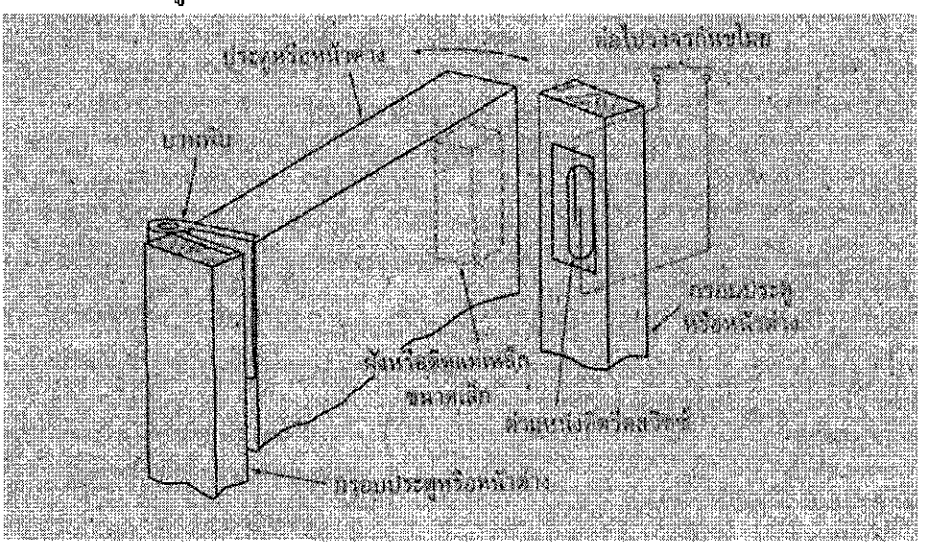
รูปที่ 2.38 แสดงการนำรีเลย์มาต่อเป็นวงจรออสซิลเลเตอร์เพื่อทำเป็นไฟกระพริบ ภายในวงจรใช้รีเลย์ขนาด 12 โวลท์ โดยที่หน้าสัมผัสจะจากกันเมื่อแรงดันต่ำกว่า 5 โวลท์ การทำงานของวงจรเริ่มจากการกดสวิตช์ S1 จะทำให้มีกระแสไหลครบวงจรผ่านขดลวดของรีเลย์ พร้อมทั้งชาร์จไฟเข้าที่ตัวเก็บประจุ C1 ซึ่งจะทำการประจุกระแส จนกระทั่งแรงดันตกคร่อมขดลวดของรีเลย์ RY1 ทำงาน ทำให้หน้าสัมผัสแบบ NC เปิดวงจรออก ตัวเก็บประจุ C1 หยุดการชาร์จ ในขณะเดียวกันก้จะทำให้หน้าสัมผัสซึ่งเป็นแบบ NO ปิดวงจรส่งผลให้หลอดไฟ L1 ส่วาง ขณะนี้ตัวเก็บประจุ C1 เริ่มคายประจุให้กับขดลวดแทน มีผลทำให้รีเลย์คงสภาวะการทำงานค้างไว้ จนกระทั่งแรงดันที่คาย

ออกจาก C1 ค่อย ๆ ลดลงจนถึงค่าที่ทำให้ขดลวดไม่สามารถดูดหน้าสัมผัสให้อยู่ได้ จึงทำให้รีเลย์กลับสู่สภาวะเริ่มต้นหรือสภาวะปกติอีกครั้ง ทำให้หน้าสัมผัสของรีเลย์เปิดปิดสลับกันไปตลอดทำให้ไฟติดและดับสลับกัน




รูปที่ 2.38แสดงการนำรีเลย์มาต่อเป็นวงจรออสซิลเลเตอร์เพื่อทำเป็นไฟกระพริบ

รูปที่ 2.39 แสดงการนำรีดสวิตช์ไปใช้ในวงจรกันขโมย โดยฝังสวิตช์ไว้ในกรอบประตูและฝังแม่เหล็กในบานประตู ขณะที่มีการเปิดประตูจะทำให้หน้าสัมผัสของรีดสวิตช์เปิดออกตามลักษณะการเปิดปิดประตู ถ้ามีขโมยเข้ามาก็จะทราบได้ทันที




รูปที่ 2.39 แสดงการนำรีดสวิตช์ไปใช้ในวงจรกันขโมย

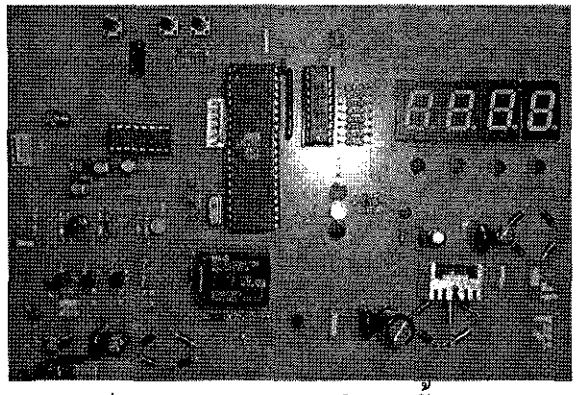
# บทที่ 3
# การคำนวนและการสร้าง

จากการศึกษาการทำงานของระบบชั่งน้ำนมเดิม(Strangko Milkers) พบว่าระบบจะเริ่มการทำงานจากเครื่องรีดน้ำนมที่รีดนมจากวัวที่เข้ามาในระบบ หลังจากนั้นน้ำนมจะถูกส่งไปตามท่อยางเข้าสู่กล่องชั่งน้ำนม ภายในกล่องชั่งน้ำนมจะประกอบไปด้วย Sensor และ วาล์ว โดย Sensor จะทำการส่งสัญญานเข้าสู่เครื่องควบคุมเมื่อระดับน้ำนมมาถึงระดับที่เราได้ตั้งไว้ เครื่องควบคุมจะทำการรับสัญญาณจาก Sensor และทำการประมวลผลส่งสัญญาณไปควบคุมการทำงานของวาล์ว จากนั้นจะทำการประมวลผลแสดงผลของน้ำหนักนมที่ผ่านการชั่งโดยระบบ สุดท้ายแล้วน้ำนมทั้งหมดที่ผ่านการชั่งโดยระบบจะถูกนำไปเก็บรวมที่ถังรวมน้ำนม จากการตรวจสอบและสังเกตุลักษณะการทำงาน พบว่ากล่องชั่งน้ำนมและเครื่องควบคุมแต่ละเครื่องภายในระบบนั้นมีความเสียหายและไม่สามารถทำงานได้ตามวัตถุประสงค์เดิม ซึ่งในแต่ละเครื่องก็มีส่วนที่เสียหายแตกต่างกันออกไป โดยกลุ่มของเราจะสร้างระบบควบคุมการชั่งน้ำนมขึ้นมาใหม่โดยอาศัยโครงสร้างของระบบเดิมที่มีอยู่เพื่อวัตถุประสงค์ที่ว่าสามารถทำงาน เข้ากับโครงสร้างของระบบเดิมได้โดยระบบสามารถชั่งปริมาณน้ำนมได้ดังเดิมและให้ระบบควบคุมใหม่ที่สร้างขึ้นนี้ง่ายต่อการใช้งานและการซ่อมแซมหากมีความชำรุดเสียหาย โดยในการทำงานของระบบที่เราจะสร้างขึ้นแบ่งหน้าที่การทำงานออกเป็น 3 ส่วนคือ ส่วนแสดงผล ส่วนของ Sensor และส่วนควบคุมการทำงานของวาล์วในระบบเดิมของเครื่องชั่ง โดยการทำงานของระบบจะเชื่อมการทำงานของทุกส่วนให้สามารถทำงานร่วมกันได้โดยจะใช้ไมโครคอนโทรลเลอร์ AT89S8252 เป็นตัวเชื่อมการทำงานและเป็นส่วนประมวลผลของระบบที่เราจะสร้างขึ้นด้วย

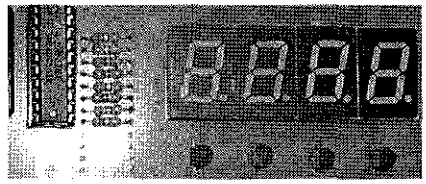
**3.1 ก) ส่วนแสดงผล**

ส่วนแสดงผลของระบบใหม่ที่เราจะสร้างขึ้นนี้จะประกอบไปด้วย LED กับ7-SEGMENT COMMON CATHODE โดยที่ LED จะมีทั้งหมด 3 ส่วน ส่วนแรกจะแสดงการทำงานของวงจร Supply เพื่อให้รู้ว่ามีไฟเข้าสู่ระบบโดยส่วนนี้จะทำงานเองโดยอัตโนมัติ ในส่วนที่สองจะแสดงการทำงานของ Sensor โดยจะแบ่งเป็น 3 ระดับ ตามระดับน้ำที่สูงขึ้นโดยจะทำงานเมื่อระดับน้ำมาถึง ส่วนที่สาม จะเป็นส่วนของการแสดงการทำงานของสวิตช์ Start, Stop และ Sensor โดยจะมีสีแตกต่างกัน โดยสวิตช์ Start จะแทนด้วยสีเขียว สวิตช์ Stop จะแทนด้วยสีแดง และสวิตช์ Sensor จะแทนด้วยสีเหลือง โดยการทำงานของ LED ทั้งสามตัวนี้ จะถูกควบคุมโดย ไมโครคอนโทรลเลอร์ โดย LED ของ Start จะถูกควบคุมโดยพอร์ต P2.5 ในส่วนของ LED ของ Stop จะถูกควบคุมโดยพอร์ต P2.6 และในส่วนของ LED ของ Sensor จะถูกควบคุมโดยพอร์ต P2.4

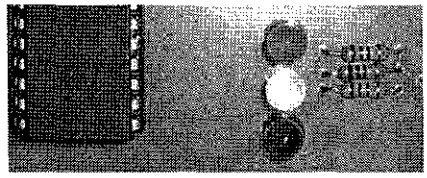
ในส่วนแสดงผลของ 7-SEGMENT โดยเราจะเลือกใช้ 7- SEGMENT แบบ COMMON CATHODE ซึ่งจะทำงานแบบ Active Lowโดยจะใช้ทั้งหมด 4 หลัก เพื่อแสดงผลของน้ำนมวัวที่ได้ และจะแสดงออกในหน่วย กิโลกรัม เพื่อให้ง่ายต่อการวิจัยปริมาณน้ำนมกับสูตรอาหาร 7-SEGMENT ทั้ง 4 ตัวนี้จะถูกควบคุมโดย ไมโครคอนโทรลเลอร์ ซึ่งจะใช้พอร์ต P2.0 P2.1 P2.2 และ P2.3 ตามลำดับ เป็นตัวควบคุม โดยไฟเลี้ยงของส่วนแสดงผลทั้งหมด จะใช้ไฟกระแสตรง +5 โวลท์



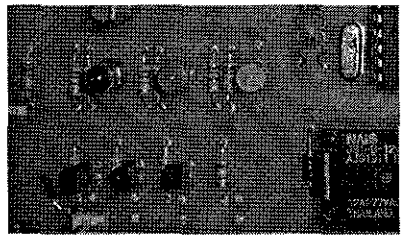
รูป 3.1 แสดงวงจรการทำงานทั้งหมด



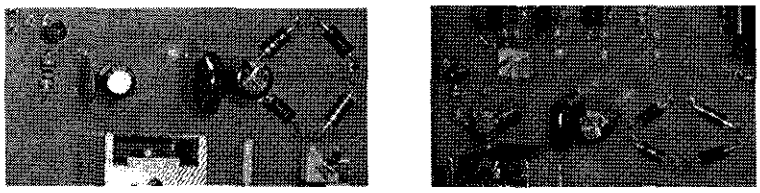
รูป 3.2 แสดงส่วนของ7- SEGMENT แบบ COMMON CATHODE



รูป 3.3 แสดงส่วนของ LED แทนการทำงานของสวิตช์ Start, Stop และ Sensor



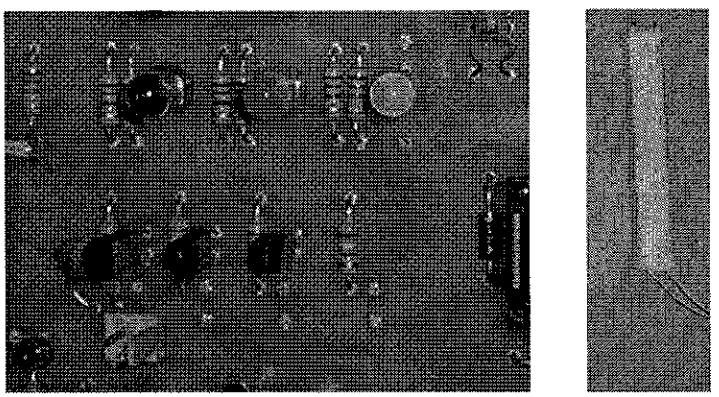
รูป 3.4 แสดงส่วนของ LED แทนการทำงานของ Sensor ทั้ง 3 ระดับ



รูป 3.5 แสดงส่วนของ LED แทนการทำงานของSUPPLY ของระบบ

**3.1 ข) ส่วนของ Sensor**

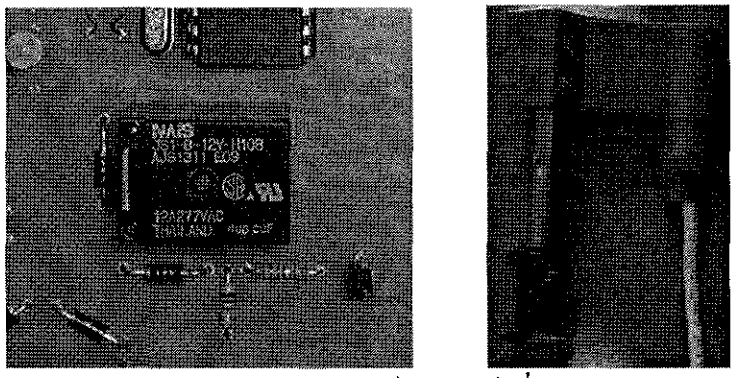
ในส่วนของ Sensor นี้เราไม่ได้ใช้เซนเซอร์ชนิดเดียวกับระบบเดิม แต่จะเป็น Sensor ที่เราสร้างขึ้นเองโดยประยุกต์จากวงจรวัดระดับน้ำโดยเราจะใส่ ความต้านทาน และต่อวงจรเพิ่มเข้าไปเพื่อให้เหมือนกับเป็นสวิตช์ โดยจะส่งผลเข้าไปสั่งการในส่วนของไมโครคอนโทรลเลอร์ โดยจะสั่งการไปยังพอร์ต P1.3 เพื่อให้ไมโครคอนโทรลเลอร์รู้ว่าระดับน้ำนมได้มาถึงในจุดที่มีน้ำหนักโดยประมาณ 0.2 กิโลกรัม ซึ่งเมื่อไมโครคอนโทรลเลอร์รับทราบคำสั่งแล้วก็จะทำงานในส่วนต่อไปตามที่ได้ออกแบบการทำงานไว้ ซึ่ง Sensor นี้จะอาศัยน้ำเป็นสื่อกลางในการทำงานก็ต่อเมื่อขาทั้งสองของ Sensor อยู่ในน้ำที่เชื่อมต่อกัน โดยในส่วนของ LED แสดงผลการทำงานของ Sensor ซึ่งมี 3 ตัว ประกอบด้วยสีแดง สีเหลือง และสีเขียวตามลำดับ โดยเมื่อน้ำนมถึงระดับใด สีของ LED ระดับนั้นก็จะส่วางขึ้น โดยสีเขียวจะเป็นจุดที่ต่ำที่สุด LED สีเหลืองจะเป็นจุดที่อยู่ในระดับกลาง และ LED สีแดงจะอยู่ในระดับสูงสุดซึ่งเป็นระดับที่เราต้องการเพราะเป็นระดับที่น้ำนมจะมีน้ำหนักประมาณ 0.2 กิโลกรัม แต่ในการทำงานจริงเราได้ตัดในส่วน LED แสดงผล สีเหลืองและสีเขียวออกไปคงเหลือแต่เพียงสีแดง เนื่องจากเราต้องการให้ขนาดของแท่ง Sensor ที่จะเข้าไปในกล่องชั่งน้ำนมมีขนาดเล็ก เพื่อที่ว่าระบบสูญญากาศ จะไม่รั่วไหลออกมา นอกจากนี้เรายังได้ทำการเพิ่ม ตัวเก็บประจุ 0.1 ไมโครฟาหรัด เข้าไปยังส่วนของฮาร์ดแวร์ เพื่อให้ Sensor ที่เราสร้างขึ้นไม่อ่อนไหวต่อการทำงานจนเกินไป



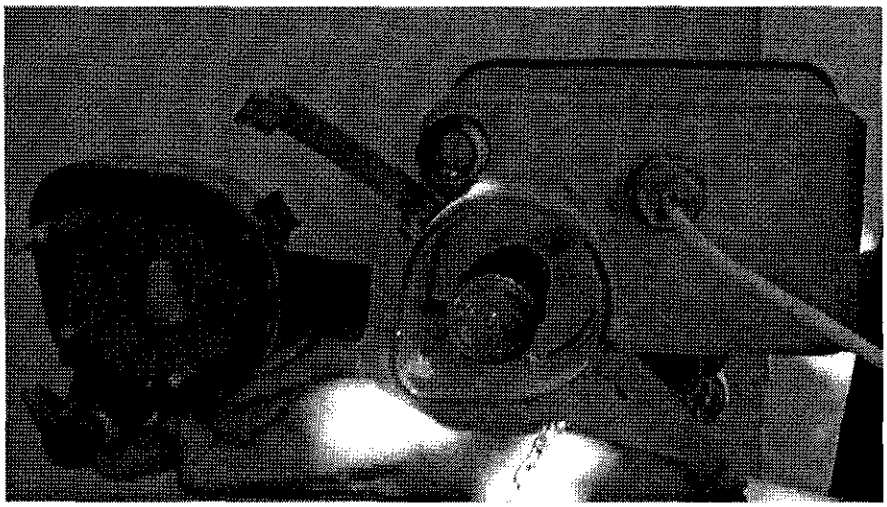
รูป 3.6 แสดงส่วนของHard Ware ของ Sensor และแท่ง Sensor ที่จะนำเข้าไปในระบบเดิม

**3.1 ค) ส่วนควบคุมการทำงานของวาล์วในระบบเดิมของเครื่องชั่ง**

ในส่วนนี้เราจะใช้ รีเลย์ 5 Contacts ขนาด 12 โวลท์ เป็นตัวป้อนกระแสไฟฟ้าให้ไหลผ่านขดลวดจำนวนหนึ่งเพื่อนำไปควบคุมวาล์วในระบบเดิม โดยจะส่งสัญญาณไฟขนาด 24 โวลท์ กระแสตรงเข้าไปยังขดลวดในวาล์วเพื่อเหนี่ยวนำให้เกิดสนามแม่เหล็กดูดแท่งเหล็กขึ้นมา วาล์วก็จะสามารถทำงานได้ ในส่วนของฮาร์ดแวร์เราจะต้องเพิ่ม ไดโอด เบอร์ IN4001 เข้าไปเพื่อป้องกันการเหนี่ยวนำกระแสย้อนกลับจากวาล์วเข้าสู่ไมโครคอนโทรลเลอร์ เพื่อป้องกันการรบกวนการทำงานของไมโครคอนโทรลเลอร์ เพราะหากถ้าเราไม่ป้องกันแล้วไมโครคอนโทรลเลอร์จะเกิดความผิดพลาดในการทำงานหรือเกิดการError ขึ้นได้โดย รีเลย์นี้จะถูกควบคุมโดยไมโครคอนโทรลเลอร์โดยจะใช้พอร์ท P3.7 เป็นตัวควบคุมการทำงาน



รูป 3.7 แสดงส่วนของHard Ware ของ รีเลย์ และวาล์วที่จะควบคุมในระบบเดิม



รูป 3.8 แสดงส่วนของวาล์วที่จะควบคุมในระบบเดิม

**3.2 การออกแบบวงจรส่วนของ AT89S8252**

ในส่วนของไมโครคอนโทรลเลอร์นี้ จะใช้คริสตอลความถี่ 11.0592 MHz เป็นตัวกำเนิดความถี่และใช้ไฟเลี้ยงขนาด +5 โวลท์ซึ่งเป็นกระแสตรง โดยขาต่าง ๆ ที่ใช้มีดังนี้

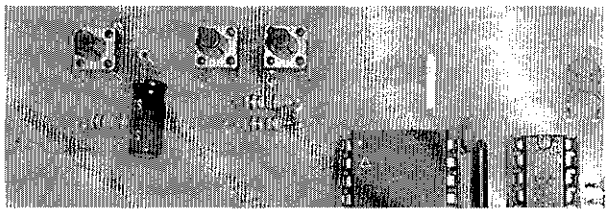
ก)พอร์ต0(P0.0-P0.7)ใช้ในการส่ง Data ไปยัง 7-SEGMENT ทั้ง 4 ตัว โดยจะเป็นตัวกำหนดตัวเลขต่าง ๆ ที่ต้องการแสดงผลออกยัง 7- SEGMENT

ข)พอร์ต1จะใช้ P1.0 เป็นสวิตช์ Stop P1.1 เป็นสวิตช์ Start P1.3 เป็นส่วนที่รับคำสั่งจาก Sensor เข้ามา และ RST(ขาที่ 9)จะเป็นสวิตช์ Reset

ค)พอร์ต2(P2.0-P2.3) จะเป็นส่วนที่ส่งสัญญาณไฟฟ้าไปให้กับทรานซิสเตอร์ BC557 เพื่อให้ทรานซิสเตอร์ ควบคุมการทำงานของ 7- SEGMENT ว่าจะให้หลักใดติดจะให้หลักใดดับ ส่วน P2.4 เป็นตัวควบคุมการทำงานของ LED แสดงผลว่ารับ Sensor เข้ามา P2.5 เป็นตัวควบคุมการทำงานของ LED แสดงผล Start และ P2.6 เป็นตัวควบคุมการทำงานของ LED แสดงผล Stop

ง)พอร์ต3(P3.7)เป็นตัวควบคุมการทำงานของรีเลย์เพื่อส่งสัญญาณไฟฟ้าไปควบคุมการทำงานเปิดปิดของวาล์วในระบบเดิม

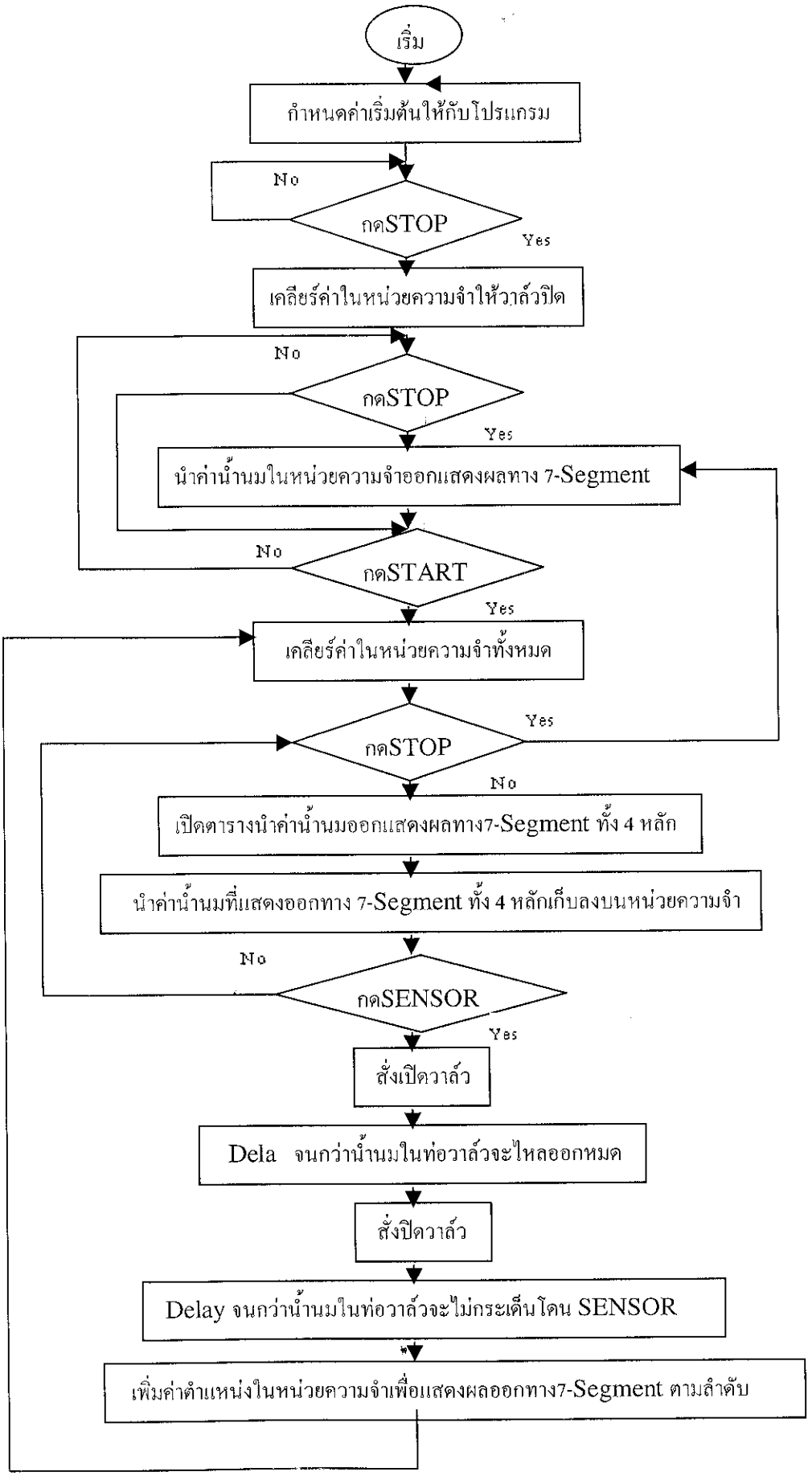
ซึ่งในโครงการนี้จะใช้หน้าที่พิเศษของพอร์ต 3 นี้ด้วยคือ P3.0 และ P3.1 โดยจะนำมาเป็นส่วนของการส่งข้อมูลระหว่าง ไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ ในลักษณะการสื่อสารแบบอนุกรม ตามแบบ RS-232 ซึ่งต้องเพิ่มส่วนของ Line Driver เพื่อให้แรงดันในการสื่อสารเป็นไปตาม RS-232 โดยเลือกใช้ IC เบอร์ MAX 232 แต่ในโครงการนี้เราไม่ได้ใช้ในส่วนนี้ แต่เราทำไว้เพื่อให้นำมาประยุกต์ใช้ในการทำ Data Base เพื่อเก็บข้อมูลเข้าสู่คอมพิวเตอร์ สำหรับผู้ที่ต้องการศึกษาต่อไป



รูป 3.9 แสดงส่วนของ Switch ต่างๆ ในการทำงาน

**3.3 การโปรแกรมไมโครคอนโทรลเลอร์**

สำหรับการโปรแกรมการควบคุมการทำงานและการประมวลผลของวงจร จะใช้ภาษา Assembly ซึ่งมีรายละเอียดการทำงานตามรูปของ Flow Chart ดังหน้าถัดไปโดยจะใช้โปรแกรม Notepad ในการเขียนโปรแกรมและใช้โปรแกรม Command prompt ในการ Command โปรแกรมให้เป็น HEX file จากนั้นจะใช้โปรแกรม WS8252V5 ในการโหลดโปรแกรมเข้าสู่ไมโครคอนโทรลเลอร์


เริ่ม
กำหนดค่าเริ่มต้นให้กับโปรแกรม
No
กดSTOP
Yes
เคลียร์ค่าในหน่วยความจำให้วาล์วปิด
No
กดSTOP
Yes
นำค่าน้ำนมในหน่วยความจำออกแสดงผลทาง 7-Segment
No
กดSTART
Yes
เคลียร์ค่าในหน่วยความจำทั้งหมด
Yes
กดSTOP
No
เปิดตารางนำค่าน้ำนมออกแสดงผลทาง7-Segment ทั้ง 4 หลัก
นำค่าน้ำนมที่แสดงออกทาง 7-Segment ทั้ง 4 หลักเก็บลงบนหน่วยความจำ
No
กดSENSOR
Yes
สั่งเปิดวาล์ว
Dela จนกว่าน้ำนมในท่อวาล์วจะไหลออกหมด
สั่งปิดวาล์ว
Delay จนกว่าน้ำนมในท่อวาล์วจะไม่กระเด็นโดน SENSOR
เพิ่มค่าตำแหน่งในหน่วยความจำเพื่อแสดงผลออกทาง7-Segment ตามลำดับ

# บทที่ 4

# การทดลองและผลการทดลอง

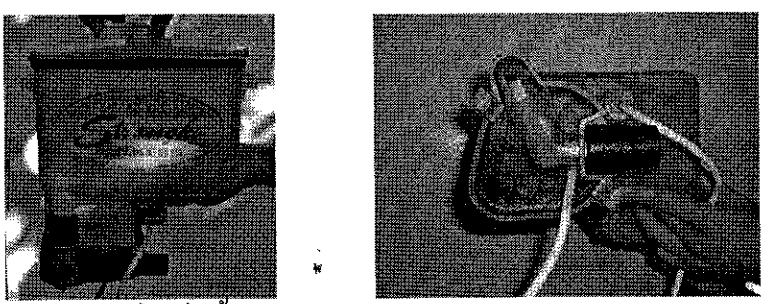
## 4.1 การทดลองทางHardware ร่วมกับกล่องชั่งน้ำนม

ในการทดลองนี้เริ่มต้นด้วยการทดลอง Hardware ก่อนโดยอาศัย Sensor ทำการส่งสัญญาณเข้ามายังไมโครคอนโทรลเลอร์และดูผลการทำงานของสวิตช์ต่างๆ ที่ได้กำหนดว่าสามารถทำตามวัตถุประสงค์ได้หรือไม่ และไมโครคอนโทรลเลอร์นั้นสามารถประมวลผลคำสั่งได้ตามที่กำหนดไว้หรือไม่

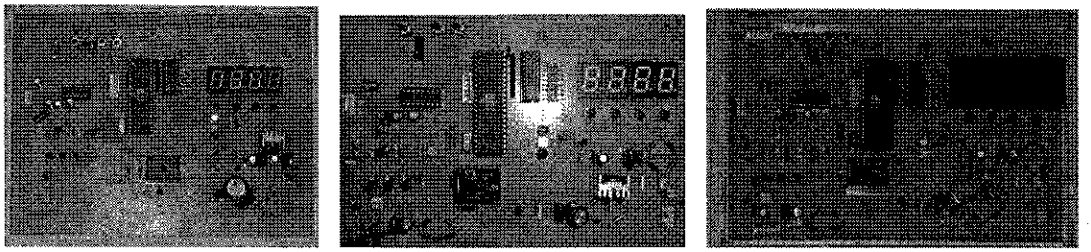
ผลจากการทดลองในขั้นนี้พบว่า Hardwareและ Software ที่ได้สร้างขึ้นสามารถทำงานได้ตรงตามวัตถุประสงค์ที่เรากำหนดคือ เมื่อกดสวิตช์ Start เครื่องควบคุมการชั่งที่เราได้สร้างขึ้นก็จะเริ่มทำงานโดยจะรอรับสัญญาณจาก Sensor เข้ามายังไมโครคอนโทรลเลอร์ทำการประมาลผล แล้วจึงส่งสัญญาณไฟฟ้าขนาด +24 V กระแสตรงไปควบคุมการทำงานของวาล์วและแสดงผลของน้ำหนักนมวัวผ่าน7-SEGMENT และ LED สีเขียว ส่วนทาง LED สีเหลืองจะแทนการทำงานของวาล์ว

เมื่อกดสวิตช์ Stop เครื่องควบคุมการชั่งที่เราได้สร้างขึ้นก็จะหยุดการทำงานโดยไม่สนใจสัญญาณจาก Sensor และแสดงผลการทำงานผ่านLED สีแดง เมื่อกดสวิตช์ Reset เครื่องควบคุมการชั่งที่เราได้สร้างขึ้นก็จะเริ่มทำงานใหม่ทั้งหมดโดยเริ่มต้นจากค่าน้ำหนักที่ 0 กิโลกรัม

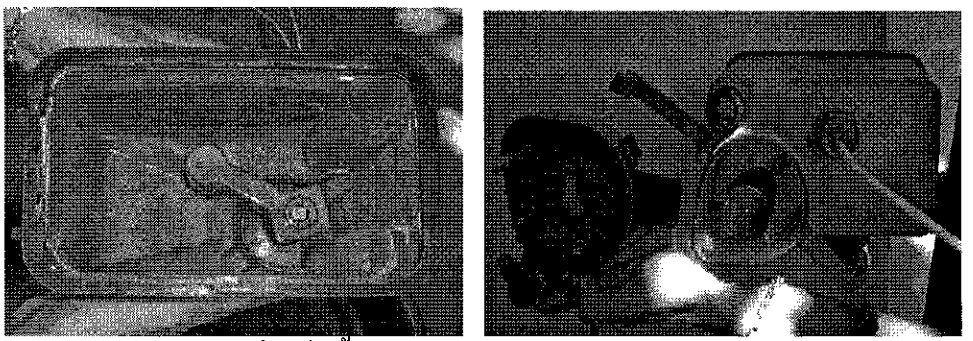
ส่วนปัญหาที่พบก็คือการกระเด็นของน้ำนมมาโดน Sensor ทำให้ Sensor เข้าใจว่าระดับน้ำนมได้มาถึงระดับที่กำหนดไว้ แต่จริงๆแล้วเป็นเพียงละอองน้ำนมที่กระเด็นมาโดนเท่านั้น ในการแก้ไขเราได้ทำการแก้ไขส่วนของโปรแกรมให้มี Delay ในการทำงานของ Sensor คือให้หยุดการทำงานในช่วงที่มีปัญหาการกระเด็นของน้ำนมรอจนกว่าปัญหานี้จะหายไปจากระบบแล้วจึงค่อยทำงานต่อตามปกติ นอกจากนี้แล้วสิ่งที่ต้องระวังเป็นพิเศษก็คือการกระเด็นของน้ำนมในระหว่างที่ทำการทดลองมาถูกระบบที่สร้างเช่น สวิตช์ จะทำให้เกิดการShort กันของสวิตช์ ซึ่งจะทำให้ระบบที่สร้างเข้าใจว่ามีการกด สวิตช์ ขึ้นดังนั้นจึงจำเป็นที่จะต้องสร้างกรอบใส่ระบบที่สร้าง โดยต้องทำการป้องกันการกระเด็นเข้ามาโดนของน้ำนม



รูป 4.1แสดงเครื่องชั่งน้ำนมจากระบบเดิม ด้านหน้า และด้านล่าง

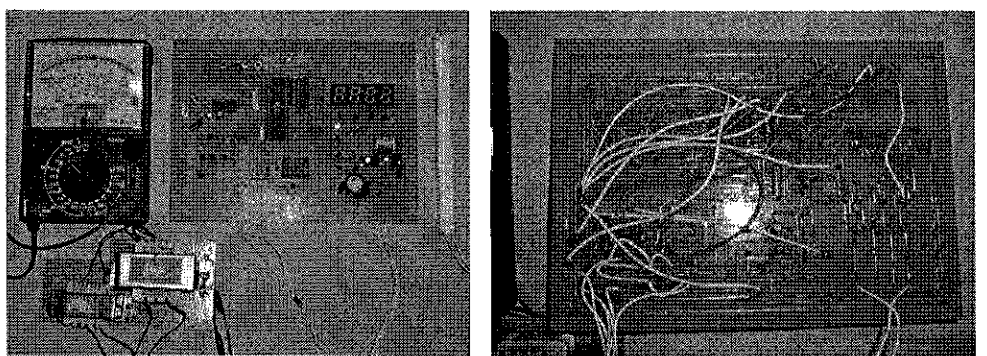


รูป 4.2 แสดงการปรับปรุงส่วนของเครื่องควบคุมที่เราสร้างขึ้น

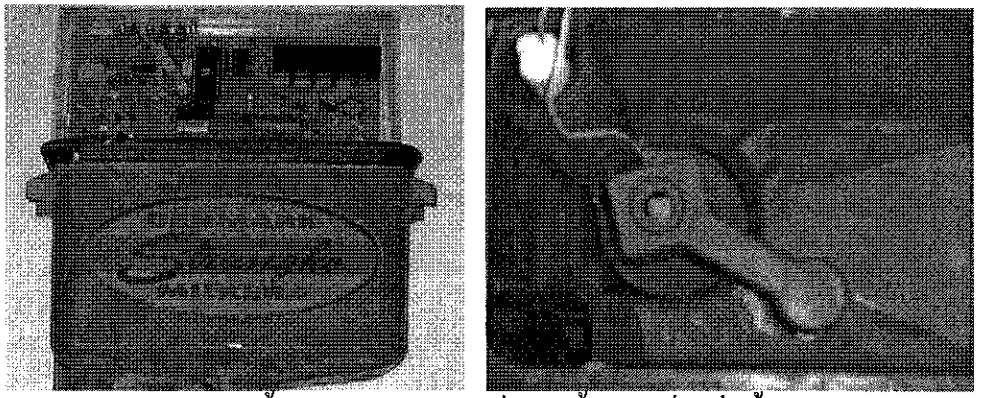


รูป 4.3 แสดงเครื่องชั่งน้ำนมจากระบบเดิม เปิดฝาครอบด้านบนและด้านล่าง

รูป 4.4 แสดงเครื่องชั่งน้ำนมจากระบบเดิม เซนเซอร์และวาล์ว



รูป 4.5 แสดงระบบควบคุมใหม่ที่สร้างขึ้น



รูป 4.6 แสดงการติดตั้งระบบควบคุมใหม่ที่สร้างขึ้นกับเครื่องชั่งน้ำนมจากระบบเดิมและการติดตั้ง Sensor

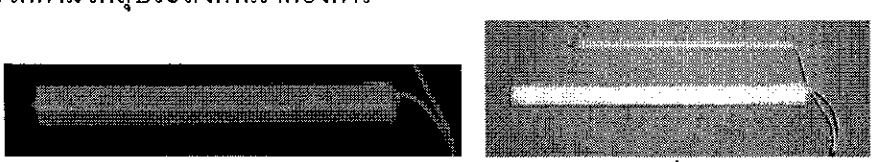
**4.2 การทดลองโดยการนำเครื่องควบคุมที่สร้างขึ้นไปติดตั้งเข้ากับระบบจริง**

ก)จากการทดลองดังข้อที่แล้วเราได้ทำการนำเครื่องควบคุมที่สร้างขึ้นไปติดตั้งเข้ากับระบบเดิมที่ฟาร์มมีอยู่เพื่อทดลองว่าเครื่องควบคุมที่เราสร้างขึ้นสามารถทำงานร่วมกับระบบของฟาร์มได้หรือไม่ ซึ่งจากการทดลองพบว่ามีความผิดพลาดในการทำงานเกิดขึ้นโดยเกิดการ Error ขึ้นที่เครื่องควบคุม เมื่อทำการติดตั้งกับระบบการทำงานจริง

จากการแก้ปัญหาพบว่าที่เครื่องควบคุมเกิดการ Error ขึ้นนั้น เป็นผลเนื่องมากจากสัญญาณรบกวนจากวาล์ว ที่เกิดการเหนี่ยวนำย้อนกลับเข้าสู่ไมโครคอนโทรลเลอร์ ทำให้เกิดการประมวลผลผิดพลาด ซึ่งทางเราก็ได้แก้ไขโดยใส่ไดโอด(IN4001)เพื่อทำการกั้นกระแสที่เกิดจากการเหนี่ยวนำของวาล์วไม่ให้ไหลเข้าสู่ไมโครคอนโทรลเลอร์

ข)หลังจากการที่เราได้แก้ปัญหาแล้วเราได้ทำการนำเครื่องควบคุมไปทดลองกับระบบเดิมอีกพบว่ามีความผิดพลาดเกิดขึ้นโดยความผิดพลาดที่เกิดขึ้นพบว่าเป็นผลมาจาก Sensor ที่เราสร้างขึ้นมีความแปรปรวนในการทำงาน

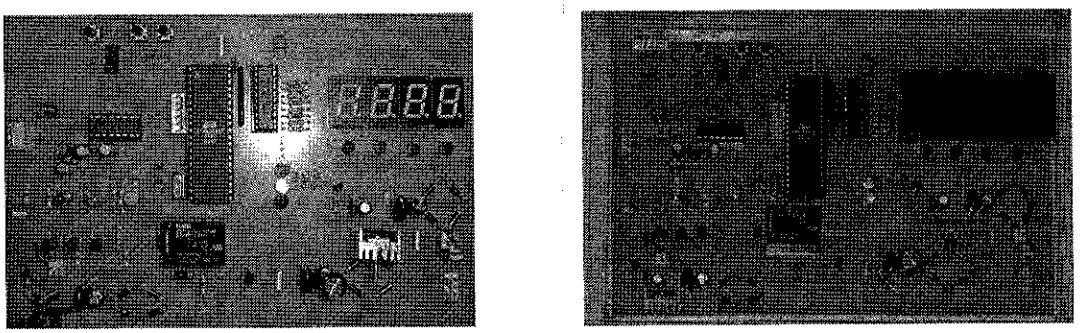
จากการแก้ปัญหาพบว่า Sensor ที่เราได้สร้างขึ้นนั้น ขาทั้งสองข้างอยู่ใกล้ชิดกันจนเกินไป ทำให้ง่ายต่อการรบกวน ของน้ำนมในการทำให้สัญญาณไฟของทั้งสองข้างเชื่อมต่อกัน ในการแก้ปัญหาเราได้ทำการแยกขาทั้งสองข้างของ Sensor ออกจากกันโดยมีผนังของกล่องชั่งน้ำนมเป็นตัวกั้นระหว่างขาทั้งสองของ Sensor เมื่อนำไปทดลองกับระบบเดิมก็พบว่าระบบของเครื่องชั่งก็สามารถทำงานได้ตามวัตถุประสงค์ที่เราต้องการ



รูป 4.7 แสดงการแก้ปัญหาของ Sensor โดยทำการเปลี่ยนรูปแบบ

ค)ปัญหาประการที่สามที่พบก็คือ จอแสดงผลทาง 7-SEGMENT มีความสว่างไม่เพียงพอต่อการมองเห็น คือมีสว่างเพียงพอต่อการมองเห็นในที่มืดได้เป็นอย่างดีแต่ในที่ส่วางมองเห็นได้ไม่ชัดเจนนัก

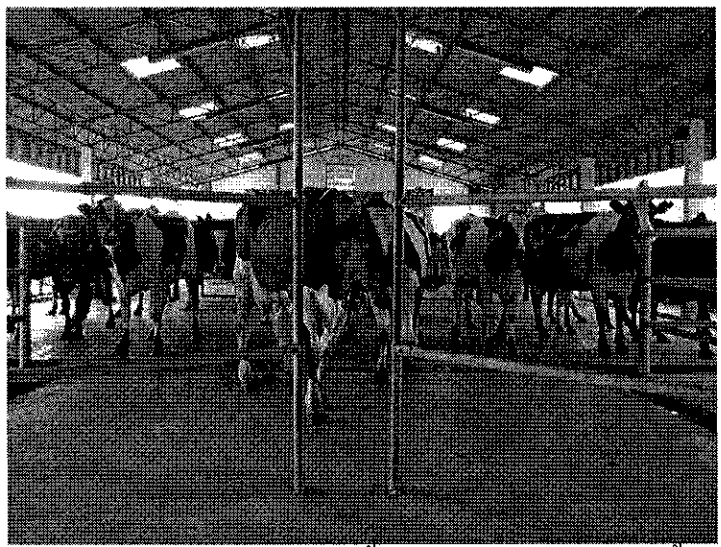
จากการแก้ปัญหาเราได้ทดลองแก้ทาง โปรแกรมโดยลดค่า Delay ของการแสดงผลแต่พบว่าไม่ได้ช่วยอะไรมากนัก เราจึงทำการถอดความต้านทานของการแสดงผลบางส่วนออก พบว่าสามารถทำให้ 7-SEGMENT เพิ่มความส่วางขึ้นมาได้แต่ไม่เพียงพอต่อการมองเห็นในที่สว่างมาก ๆ เราจึงได้ทำการ สร้างกล่องขึ้นมาคลุม 7-SEGMENT ก็พบว่าสามารถมองเห็นได้ชัดเจนทั้งในที่สว่างและที่มืด



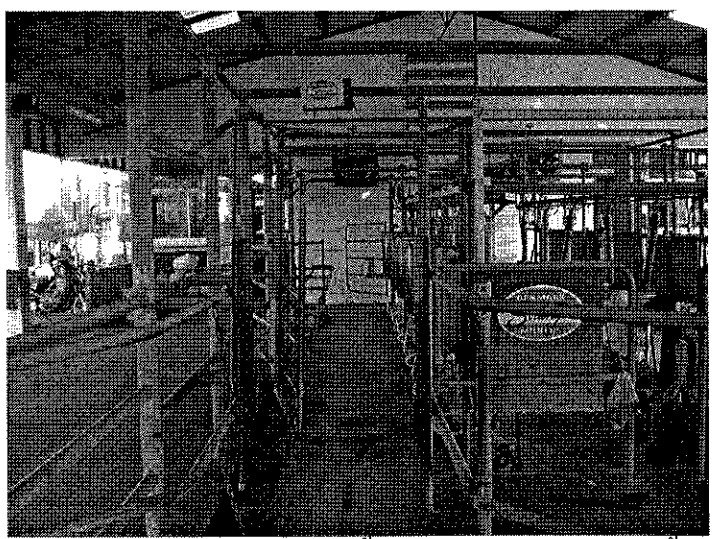
รูป 4.8 แสดงการแก้ปัญหาของการแสดงผลทาง7-SEGMENT

จากการแก้ปัญหาทั้งหมดที่กล่าวมาข้างต้น เมื่อทำการนำเครื่องควบคุมที่สร้างขึ้นที่ได้แก้ปัญหาแล้วไปติดตั้งเข้ากับระบบจริง สามารถทำงานร่วมกันได้เป็นอย่างดีและให้ผลการทดลองใกล้เคียงกับเครื่องชั่งชนิดใหม่ที่ทางฟาร์มได้จัดซื้อมาแต่เราไม่สามารถวัดผลการทดลองโดยตรงได้ เนื่องจาก ในระบบจริงจะเป็นระบบสูญญากาศ ซึ่งจะมีแรงดูดทำให้วาล์วปิดสนิทขึ้น น้ำนมเล็ดลอดผ่านได้น้อยมากแต่จะเป็นเช่นนี้เมื่อติดตั้งในระบบจริงเท่านั้นซึ่งน้ำนมที่ไหลผ่านทั้งหมดจะถูกส่งไปเก็บยังถังรวม ทำให้ไม่สามารถรู้ได้ว่าน้ำนมที่ผ่านเครื่องชั่งของเรามีปริมาณตรงกับน้ำนมที่วัดได้มากน้อยเพียงใดเพราะเมื่อเรามาทำการทดลองโดยไม่ติดตั้งเข้าสู่ระบบนั้น จะพบว่าวาล์วจะปิดไม่สนิททำให้มีน้ำนมเล็ดลอดผ่านช่องวาล์วออกมาได้ ทำให้ปริมาณของน้ำนมที่วัดได้มีความไม่แน่นอนซึ่งในการทดลองนั้นเราได้ทำการเทียบเคียงค่าผลการทดลองที่ได้กับเครื่องชั่งชนิดใหม่ที่ทางฟาร์มได้จัดซื้อมาซึ่งก็พบว่าให้ผลที่ใกล้เคียงกัน

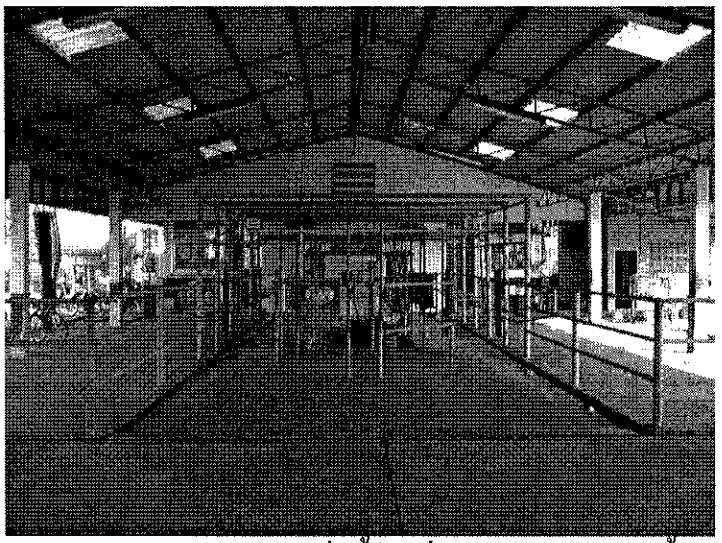
รูป 4.9 แสดงสภาพแวดล้อมของฟาร์ม มทส.



รูป 4.10 ก) แสดงสภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง

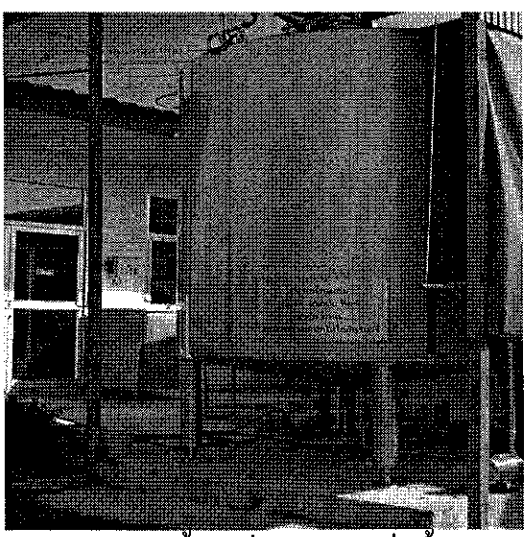


รูป 4.10 ข) แสดงสภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง



รูป 4.10 ค) แสดงสภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง

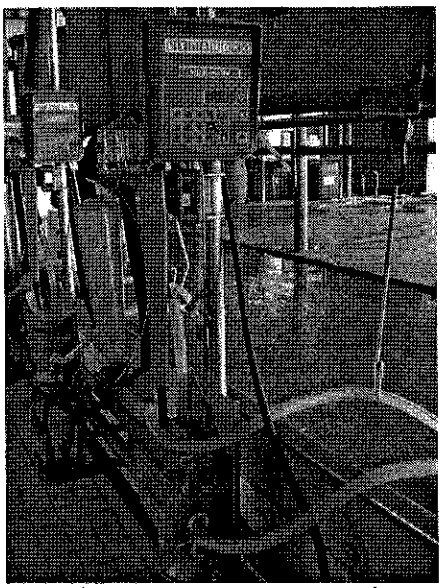
รูป 4.10 ง) แสดงสภาพของระบบชั่งน้ำนมที่ฟาร์ม มทส. ทำการซื้อมาติดตั้ง



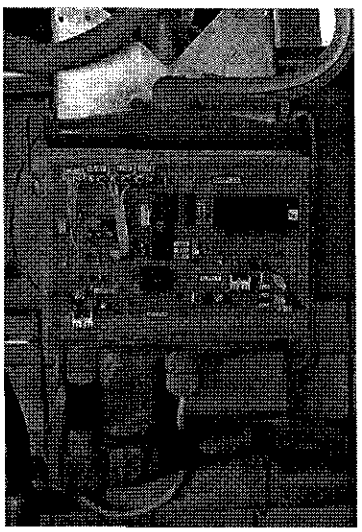
รูป 4.11 ถังรวมน้ำนมที่ได้ผ่านการชั่งน้ำหนักแล้ว

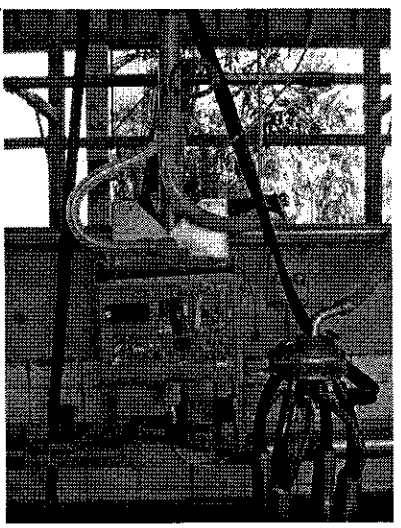
รูป 4.12 ก) แสดงสภาพของระบบชั่งน้ำนม และเครื่องชั่งน้ำนมเดิม

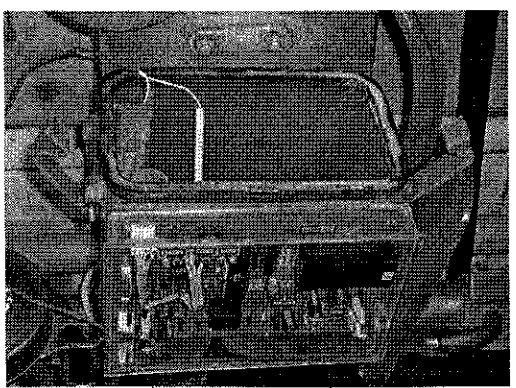
รูป 4.12 ข) แสดงสภาพของระบบชั่งน้ำนม และเครื่องชั่งน้ำนมเดิม



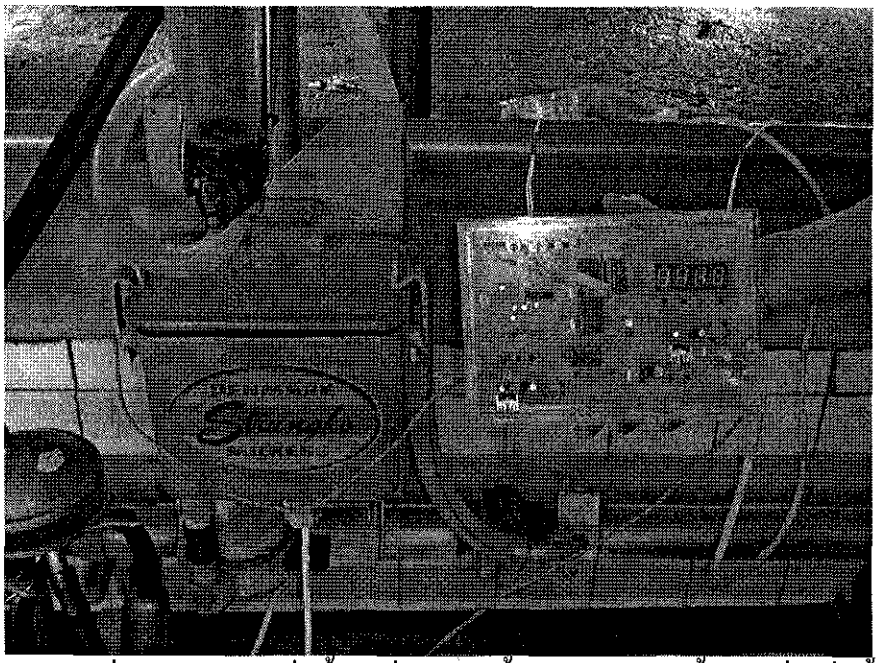
รูป 4.13 แสดงเครื่องชั่งน้ำนมและเครื่องควบคุมของระบบเดิมซึ่งมีความเสียหายและใช้การไม่ได้ในบางส่วน



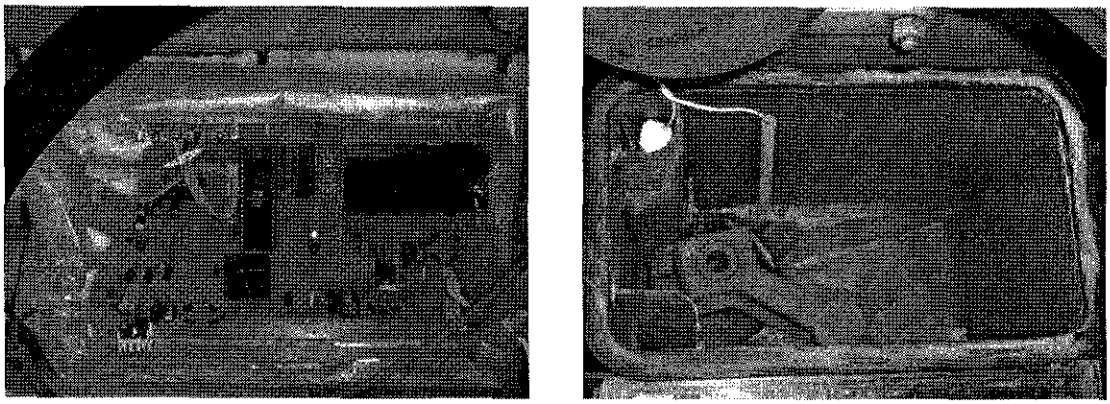


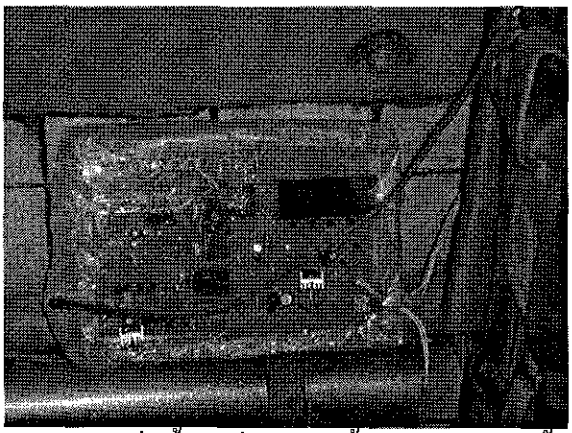


รูป 4.14 ก) แสดงเครื่องควบคุมการชั่งน้ำนมที่เราสร้างขึ้นโดยทำการติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิมเพื่อทดสอบความสามารถในการทำงานร่วมกัน

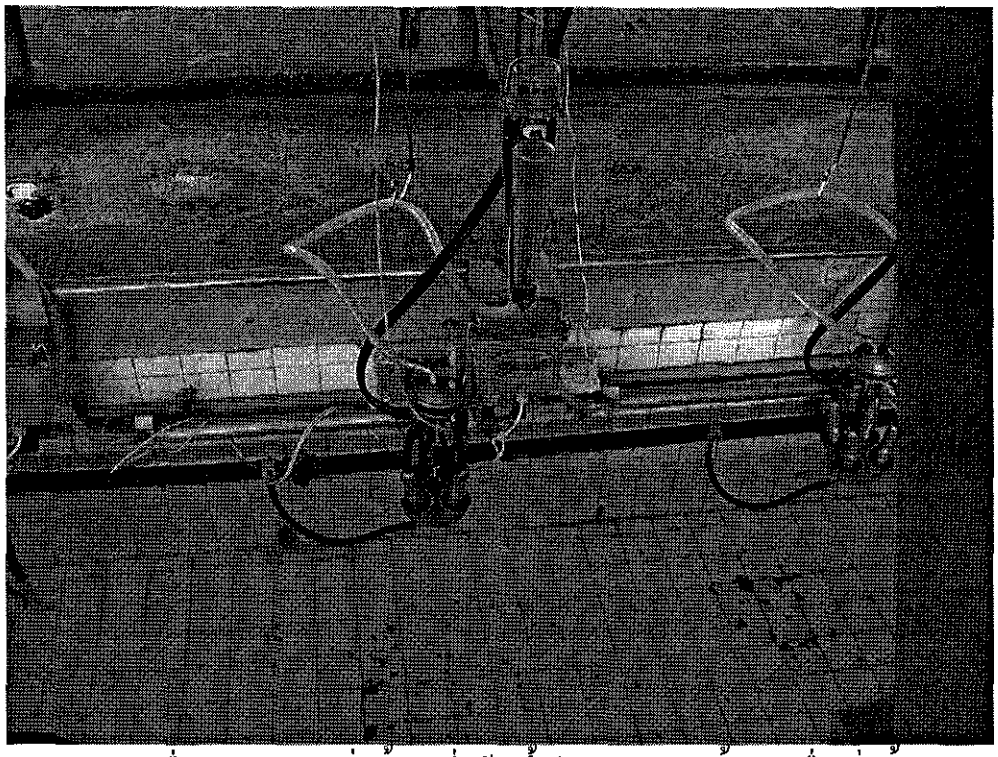


รูป 4.14 ข) แสดงเครื่องควบคุมการชั่งน้ำนมที่เราสร้างขึ้นโดยทำการติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิมเพื่อทดสอบความสามารถในการทำงานร่วมกัน



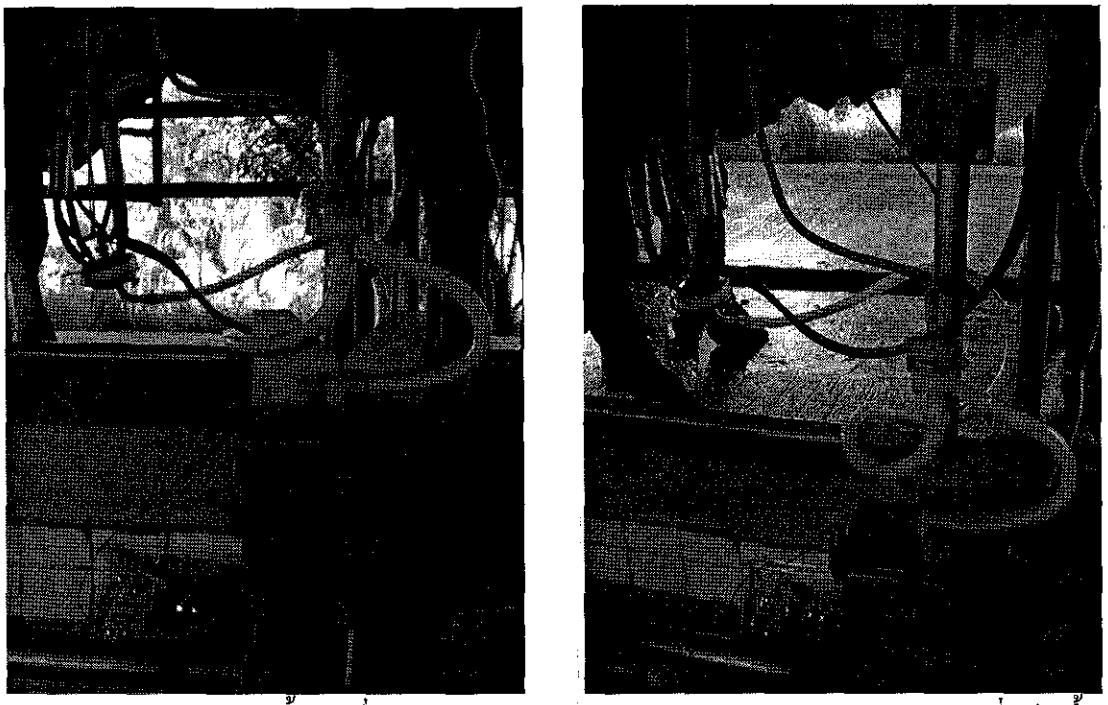


รูป 4.14 ค) แสดงเครื่องควบคุมการชั่งน้ำนมที่เรวสร้างขึ้นโดยทำการติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิมเพื่อทดสอบความสามารถในการทำงานร่วมกัน

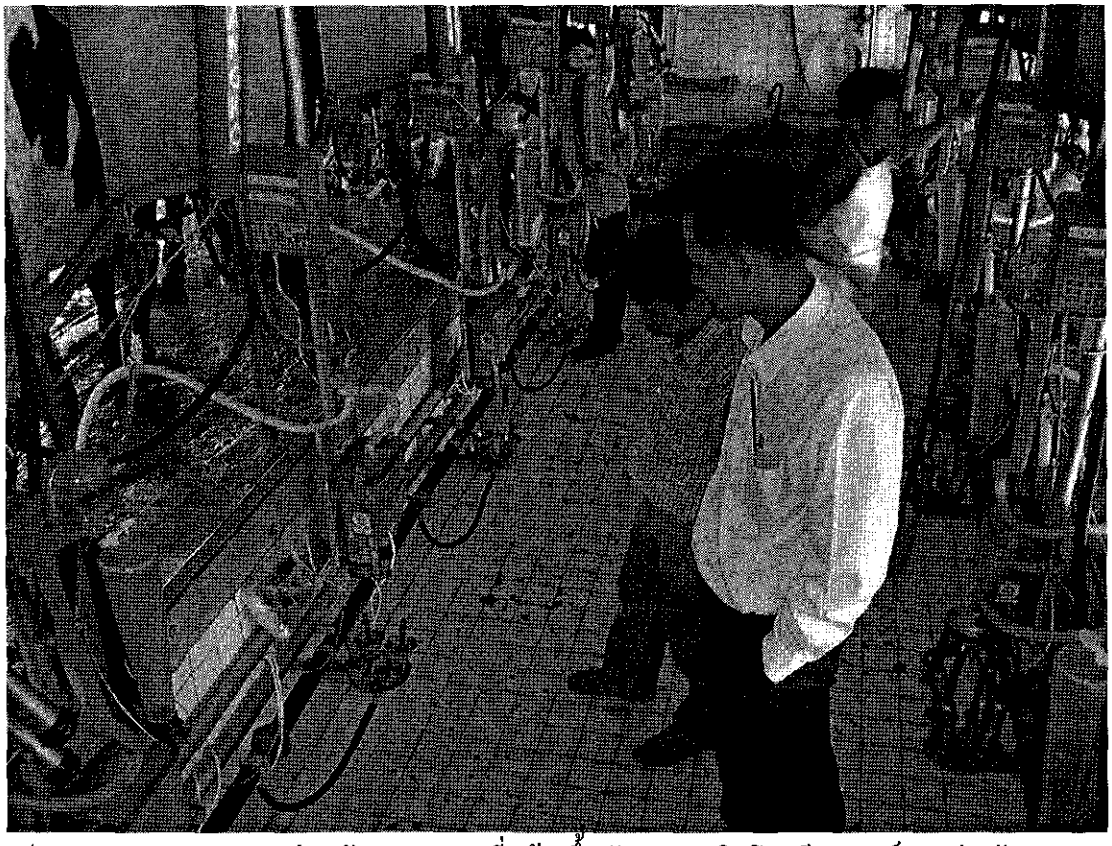


รูป 4.14 ง) แสดงเครื่องควบคุมการชั่งน้ำนมที่สร้างขึ้นโดยทำการติดตั้งกับเครื่องชั่งน้ำนมของระบบเดิมเพื่อทดสอบความสามารถในการทำงานร่วมกัน

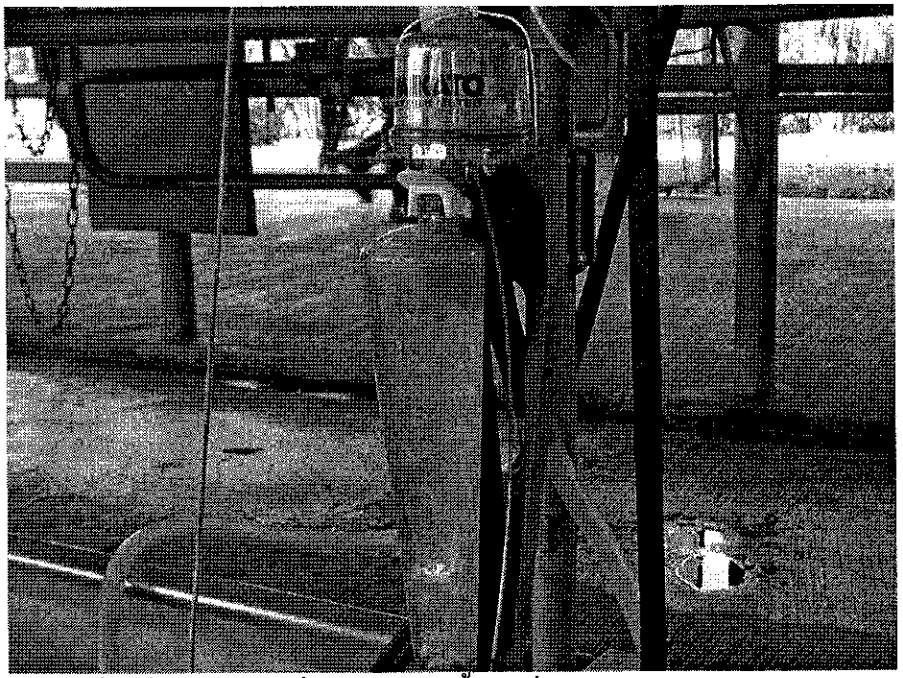
รูป 4.15 ก) แสดงการรีดน้ำนมเพื่อทดสอบความสามารถในการทำงานร่วมกันของระบบที่สร้างขึ้นกับระบบเดิม



รูป 4.15 ข) แสดงการรีดน้ำนมเพื่อทดสอบความสามารถในการทำงานร่วมกันของระบบที่สร้างขึ้นกับระบบเดิม



รูป 4.16 แสดงการทำงานร่วมกันของระบบที่สร้างขึ้นกับระบบเดิมโดยมีอาจารย์และช่างฝ่ายเทคนิคของฟาร์มเป็นผู้ทดสอบ



รูป 4.17 แสดงเครื่องชั่งระบบใหม่ที่ทางฟาร์มจัดซื้อมาเพื่อทดแทนระบบเดิม โดยเป็นระบบ Mechanic แทนที่ระบบเดิมที่เป็น Electronic

# บทที่ 5
# บทวิจารณ์และบทสรุป

จากการที่ได้ออกแบบสร้างและทดสอบตามกระบวนการทดลองแล้วทุก ๆ ขั้นตอนที่ได้ปฏิบัติไป จะพบว่าเกิดปัญหามากมายในการปฏิบัติงาน เนื่องจากโครงการนี้ เป็นโครงการที่เราต้องคิดและออกแบบการทำงานเองทั้งหมด โดยอาศัยการประยุกต์ความรู้ที่ได้มาจากแหล่งต่าง ๆ เช่น ความรู้ที่ได้จาก Internet หนังสือ และคำแนะนำบอกกล่าวจากอาจารย์ ท่านต่าง ๆ นอกจากนี้ระบบที่เราได้สร้างขึ้นนั้น ไม่สามารถทำงานด้วยตัวเองได้แต่ระบบที่เราสร้างขึ้นจำเป็นที่จะต้องทำงานร่วมกับระบบเดิมที่มีอยู่ ซึ่งในระบบเดิมนั้นมีหลายส่วนที่เกิดความเสียหายและพวกเราจำเป็นที่จะต้องสร้างขึ้นมาใหม่โดยวัสดุและอุปกรณ์ที่เลือกใช้ เป็นวัสดุอุปกรณ์ที่เราเลือกเอง ไม่ได้ยึดหลักกับอุปกรณ์เดิมที่เป็นเช่นนี้ก็เพราะว่าเราต้องการให้ระบบที่เราสร้างขึ้น มีต้นทุนที่ต่ำสะดวกต่อการใช้งาน ง่ายต่อการแก้ไขและมีความยืดหยุ่นต่อการทำงาน ด้วยเหตุนี้ทำให้ในการทดลองและผลการทดลองได้เกิดปัญหาต่าง ๆ ขึ้น แต่พวกเราก็สามารถแก้ปัญหาให้ลุล่วงมาได้

ขบวนการในการสร้างเริ่มจาก ชุดไมโครคอนโทรลเลอร์ซึ่งเราได้ประยุกต์ใช้ความรู้ที่ได้จากการศึกษา จากที่ได้กล่าวมาข้างต้น ในการสร้างชุดควบคุม เริ่มจากการซื้อวัสดุที่ใช้และใช้เวลาในการออกแบบและสร้างนานพอสมควร ต่อมาเป็นส่วนของการเขียนโปรแกรมเพื่อให้สอดคล้องกับฮาร์ดแวร์ที่เราได้สร้างขึ้นพบว่ามีปัญหามากมายที่เกิดขึ้นเพราะการที่จะทำให้ ซอฟท์แวร์และฮาร์ดแวร์ทำงานร่วมกันนั้น เป็นเรื่องที่ยากซึ่งต้องใช้เวลาในการปรับปรุงแก้ไขอยู่เป็นเวลานานพอสมควร หลังจากนั้นเราได้ทำการติดตั้งกับระบบจริงก็พบว่ามีปัญหาต่าง ๆ ที่เกิดขึ้นตามมามากมายในการทำงานให้สอดคล้องกันระหว่างระบบที่เราสร้างขึ้นและระบบเดิมซึ่งเราได้แก้ไขระบบที่เราสร้างขึ้นให้มีความยืดหยุ่นและสามารถทำงานร่วมกับระบบเดิมได้เป็นอย่างดี

สำหรับงานที่ทำเพิ่มในเทอมนี้ก็มีส่วนของการทำแผ่นวงจร การลงอุปกรณ์ และการประกอบลงกล่อง ส่วนในด้านซอฟต์แวร์เราได้ใช้ภาษา Assembly ในการออกแบบการทำงานของเราโดยได้มีการปรับปรุงให้สอดคล้องกับสิ่งที่เราต้องการ และได้สร้างสวิตช์เชื่อมต่อระหว่างฮาร์ดแวร์และตัวอุปกรณ์ เพื่อให้มีความสะดวกในการใช้งานขึ้น

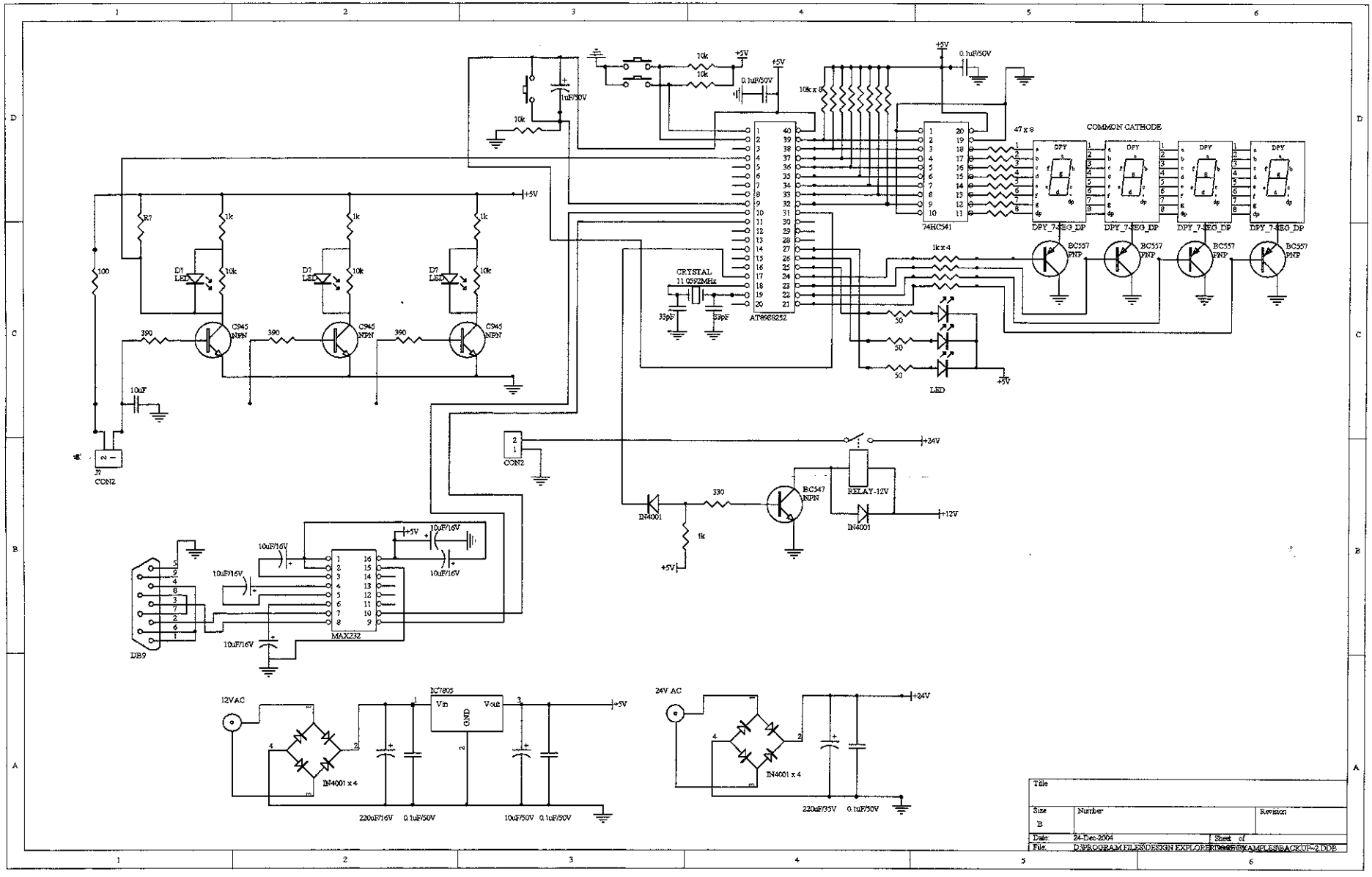
ทั้งผู้จัดทำโครงการนี้หวังว่าจะมีการพัฒนาโครงการนี้ต่อไป เนื่องจากระบบที่สร้างขึ้นสามารถลดต้นทุนในการสร้างได้หลายเท่ามาก นอกจากนี้ยังเป็นการประยุกต์ใช้ความรู้ที่มีอยู่แก้ปัญหาได้จริง เพื่อสร้างความมั่นใจและฝึกทักษะการทำงานเพื่อออกไปเผชิญกับโลกภายนอกได้ถึงแม้ว่าเทคโนโลยีที่เราได้นำมาสร้างขึ้นจะไม่เป็นเทคโนโลยีที่สูงมากและเป็นที่สนใจของตลาด

ด้านเทคโนโลยีในปัจจุบันแต่อย่างน้อย มันก็สามารถแก้ปัญหาและสร้างสิ่งต่าง ๆ ให้ทำงานตามวัตถุประสงค์ที่เราต้องการได้เป็นอย่างดี

**ข้อเสนอแนะ**

จากระบบที่สร้างขึ้นได้ทำการสร้างส่วนของการสื่อสารติดต่อเข้ากับคอมพิวเตอร์ โดยสร้างในลักษณะพอร์ตสื่อสารแบบอนุกรมตามแบบ RS-232 ซึ่งได้เพิ่มส่วนของ Line Driver เพื่อให้แรงดันในการสื่อสารเป็นไปตาม RS-232 โดยเลือกใช้ IC เบอร์ MAX 232 แต่ในโครงการนี้เราไม่ได้ใช้ในส่วนนี้ แต่เราทำไว้เพื่อให้มีการพัฒนาประสิทธิภาพการทำงานให้สูงขึ้นโดยการนำมาประยุกต์ใช้ในการทำ Data Base เพื่อเก็บข้อมูลเข้าสู่คอมพิวเตอร์ ซึ่งจะใช้การส่งปริมาณน้ำนมผ่านพอร์ตสื่อสารแบบอนุกรมเข้าคอมพิวเตอร์โดยอัตโนมัติแทนการจดข้อมูลซึ่งเป็นวิธีการที่ทางฟาร์มใช้อยู่ เพื่อเป็นการเพิ่มความสะดวกในการจัดเก็บข้อมูล การประมวลผลข้อมูล และเป็นการประหยัดคนงานในการที่จะต้องคอยจดบันทึกผลของน้ำหนักน้ำนมวัวที่ได้ ซึ่งโดยปกติแล้วจะต้องมีคนงานอย่างน้อย 1 คนที่จะต้องทำหน้าที่คอยจดบันทึกผลของน้ำนมที่ได้ นอกจากนี้แล้วยังอาจทำการพัฒนาไปสู่การควบคุมการทำงานของระบบแบบไร้สาย หรือควบคุมการทำงานโดยผ่านระบบ Internet ก็ได้

# ภาคผนวก


10k
+5V
0.1uF/50V
1uF/50V
10k x 8
74HC541
47 x 8
COMMON CATHODE
DPY_7-SEG_DP
BC557 PNP
1k x 4
CRYSTAL 11.0592MHz
33pF
AT89S8252
LED
R7
100
1k
D? LED
10k
C945 NPN
390
10uF
JP CON2
+24V
IN4001
330
BC547 NPN
RELAY-12V
+12V
10uF/16V
MAX232
DB9
12VAC
IC7805
Vin
Vout
GND
IN4001 x 4
220uF/16V
10uF/50V
24V AC
220uF/35V
Title
Size
B
Number
Revision
Date:
24-Dec-2004
Sheet of
File:
D:\PROGRAM FILES\DESIGN EXPLORER 99 SE\EXAMPLES\BACKUP-2.DDB

# โปรแกรม Assembly

```
        ORG   0000H

        SETB  P1.0
        SETB  P1.1
        SETB  P1.3
        JB    p1.0,$
        CALL  DELAY_50MS
        JB    p1.0,$
        MOV   20H,#00H
        MOV   25H,#00H
        MOV   28H,#00H
        MOV   29H,#00H
XX:     MOV   P3,#00000000H
        CLR   A
        JB    P1.0,ZZ          ;STOP
        CALL  DELAY_50MS
        JB    P1.0,ZZ

        MOV   P2,#00110000B

XXX:
        MOV   P2,#00111110B ;POINT 1
        MOV   A,29H
        MOV   P0,A
        CALL  DELAY_1MS

        MOV   P2,#00111101B ;POINT 2
        MOV   A,28H
        MOV   P0,A
        CALL  DELAY_1MS
```

```
        MOV   P2,#00111011B ;POINT 3
        MOV   A,25H
        MOV   P0,A
        CALL  DELAY_1MS

        MOV   P2,#00110111B ;POINT 4
        MOV   A,20H
        MOV   P0,A
        CALL  DELAY_1MS

        MOV   P2,#00110000B
        MOV   P0,#00000000B
        CALL  DELAY_1MS



        JB    P1.1,XXX              ;START
        CALL  DELAY_50MS
        JB    P1.1,XXX

ZZ:
        JB    P1.1,BB
        CALL  DELAY_50MS
        JB    P1.1,BB
        MOV   P2,#01011000B
        JMP   MAIN

BB:     LJMP  XX

MAIN:
        CLR   A
```

```
        MOV   R0,A
        MOV   R1,A
        MOV   R2,A
        MOV   R3,A
        MOV   R4,A
        MOV   R5,A
        MOV   R6,A
        MOV   R7,A
        JMP   A3

AA:     INC   R0
        CJNE  R0,#5,A1
        INC   R5
        CJNE  R5,#10,A2
        INC   R6
        CJNE  R6,#10,A3
        INC   R7
        CJNE  R7,#10,A4
        JMP   MAIN

A4:
        CLR   A
        MOV   R5,A
        MOV   R6,A
        MOV   R0,A

A3:
        CLR   A
        MOV   R5,A
        MOV   R0,A

A2:
```

```
        CLR   A
        MOV   R0,A

A1:
        JB    P1.0,A0           ;STOP
        CALL  DELAY_50MS
        JB    P1.0,A0
        JMP   XX

A0:     MOV   P2,#01010111B     ;POINT 1
        MOVX  @DPTR,A
        MOV   A,R7
        MOV   DPTR,#TABLE2
        MOVC  A,@A+DPTR
        MOV   DPTR,#P0
        MOVX  @DPTR,A
        MOV   P0,A
        MOV   20H,A
        CALL  DELAY_1MS

        MOV   P2,#01011011B     ;POINT 2
        MOVX  @DPTR,A
        MOV   A,R6
        MOV   DPTR,#TABLE2
        MOVC  A,@A+DPTR
        MOV   DPTR,#P0
        MOVX  @DPTR,A
        MOV   P0,A
```

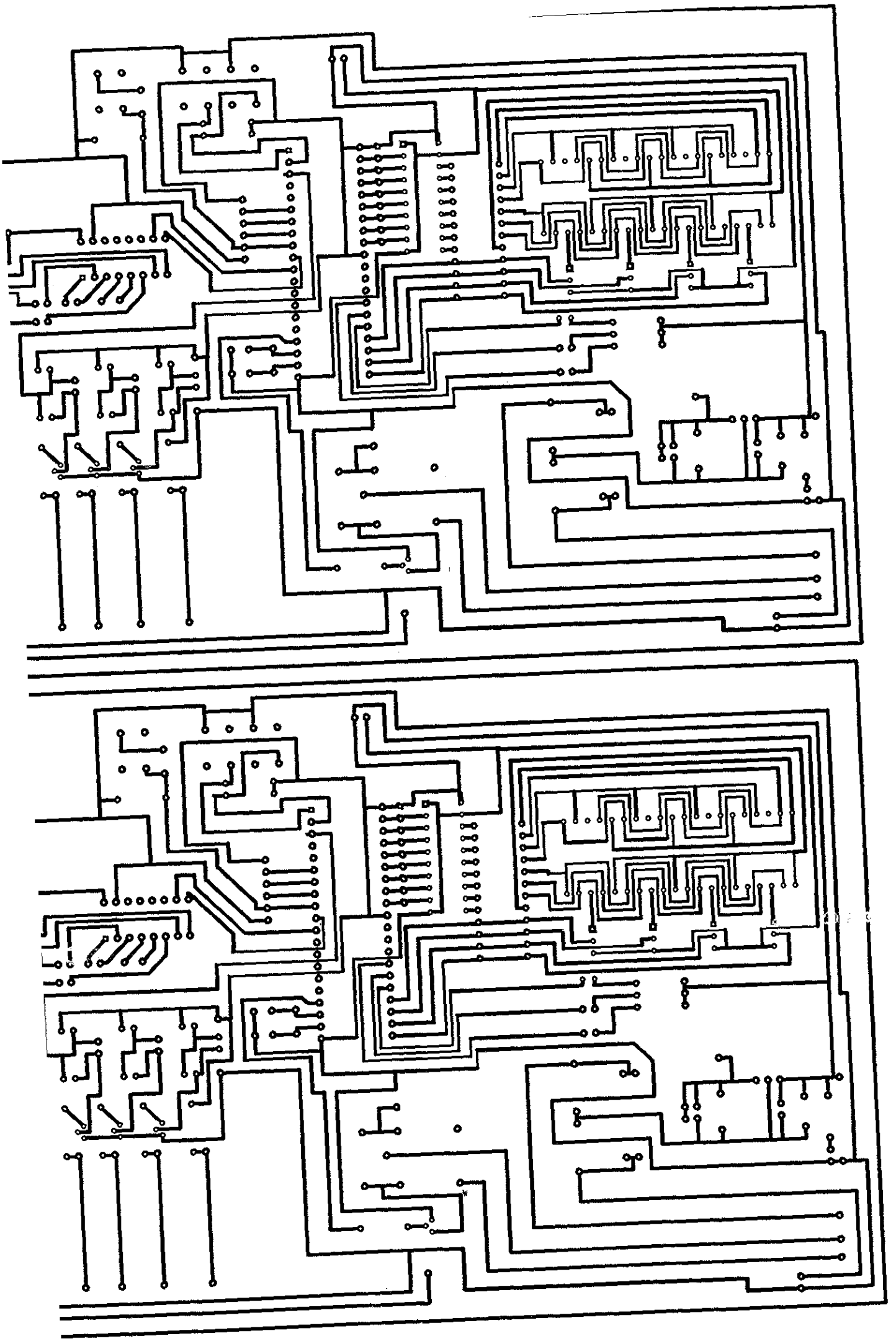
```
MOV   25H,A
CALL  DELAY_1MS


MOV   P2,#01011101B        ;POINT 3
MOVX  @DPTR,A
MOV   A,R5
MOV   DPTR,#TABLE2
MOVC  A,@A+DPTR
MOV   DPTR,#P0
MOVX  @DPTR,A
MOV   P0,A
MOV   28H,A
CALL  DELAY_1MS


MOV   P2,#01011110B        ;POINT 4
MOVX  @DPTR,A
MOV   A,R0
MOV   DPTR,#TABLE1
MOVC  A,@A+DPTR
MOV   DPTR,#P0
MOVX  @DPTR,A
MOV   P0,A
MOV   29H,A
CALL  DELAY_1MS


MOV   P2,#01010000B
MOV   P0,#00000000B
CALL  DELAY_1MS
```

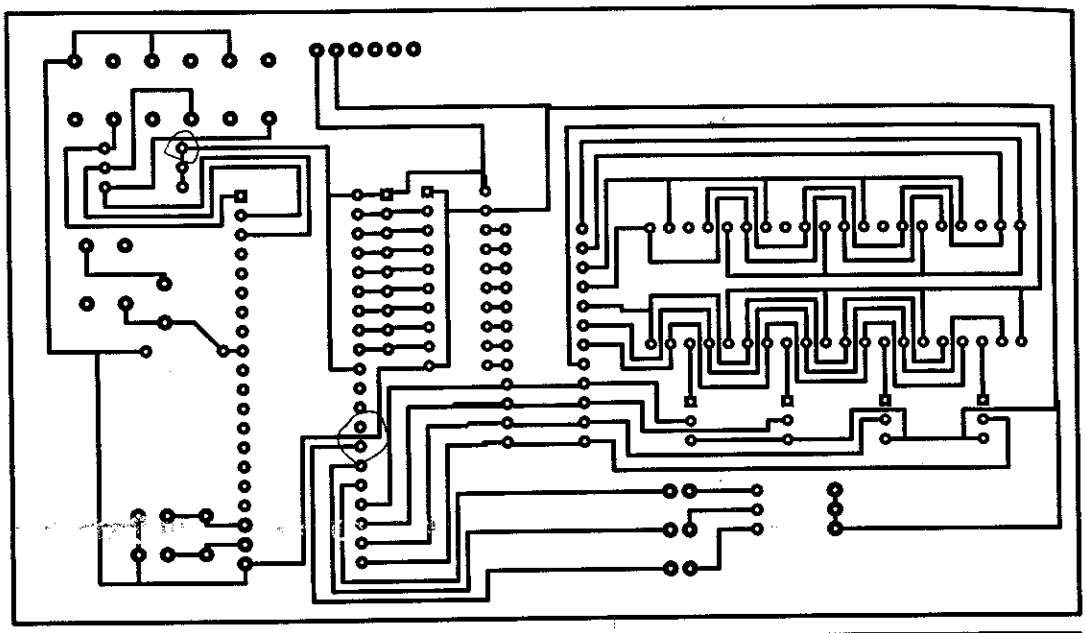
```
        JB      P1.3,CC             ;SENSOR
        CALL    DELAY_50MS
        JB      P1.3,CC
        JMP     DD

CC:
        LJMP    A1

DD:
        CALL    DELAY_500MS
        MOV     P2,#01001100B
        MOV     p3,#0FFH
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS

        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
        CALL    DELAY_500MS
```

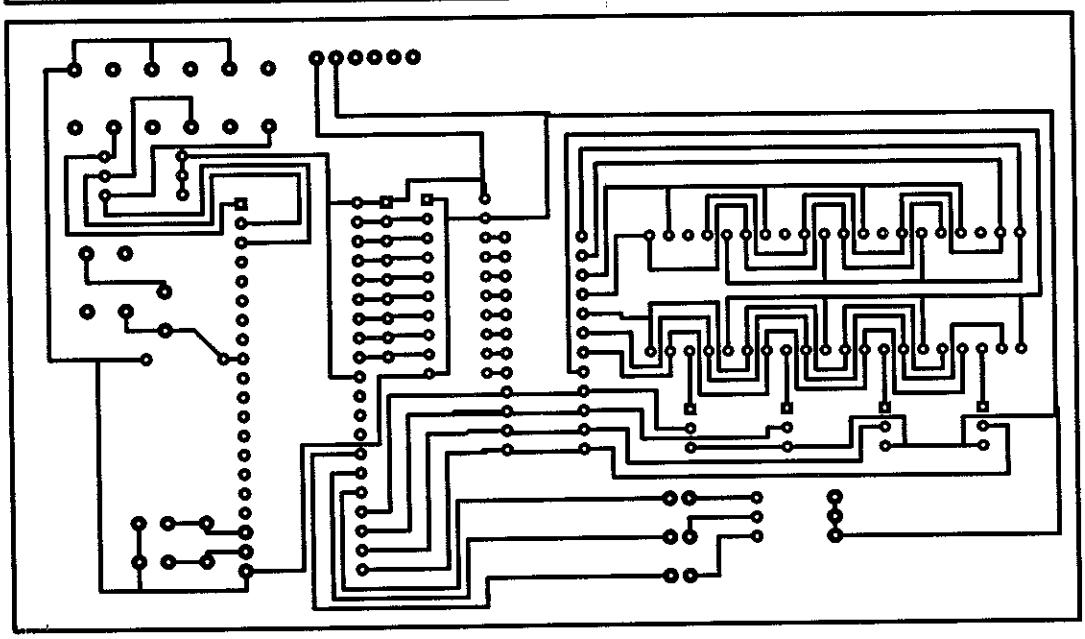
```
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS

MOV   P2,#01011100B
MOV   p3,#00H

CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
CALL  DELAY_500MS
```

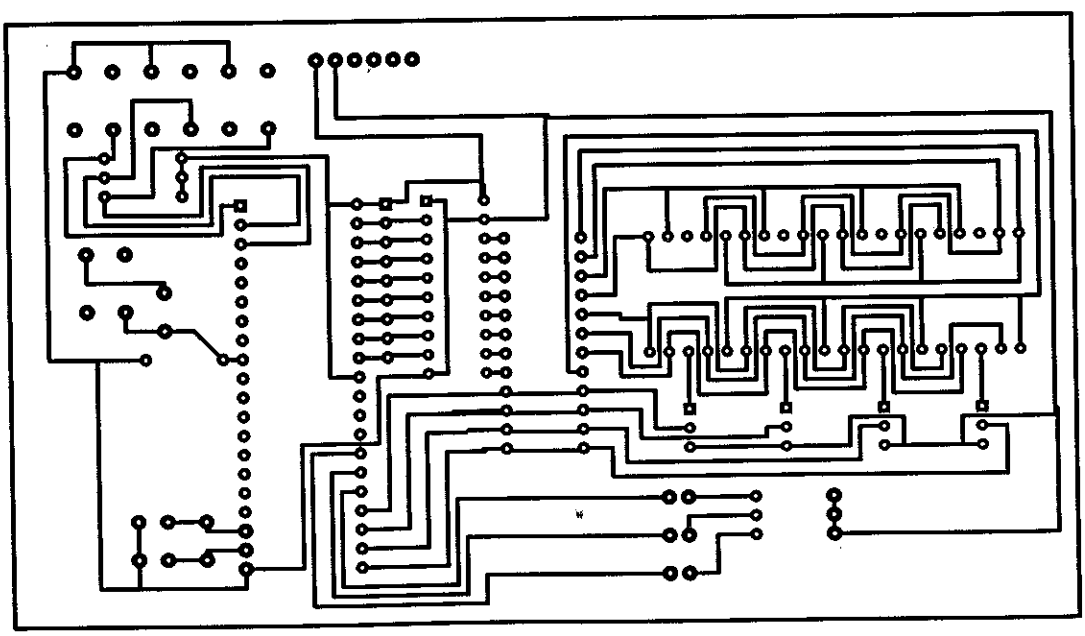
```
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        CALL  DELAY_500MS
        LJMP  AA



TABLE1:     DB    11110111B,11011011B,10101101B,11111110B,11111111B
TABLE2:     DB
        11110111B,10100001B,11011011B,11101011B,10101101B,11101110B,11111110B,10
100011B,11111111B,11101111B

DELAY_500MS:      PUSH  04H
                  MOV   R4,#750
NN:     CALL  DELAY_1MS
        CALL  DELAY_1MS
                  DJNZ  R4,NN
```

```
                POP     04H
                RET
DELAY_1MS:      PUSH    02H
                PUSH    03H
                MOV     R2,#5
YY:     MOV     R3,#16
        DJNZ    R3,$
        DJNZ    R2,YY
        POP     03H
        POP     02H
        RET

DELAY_50MS:     PUSH    04H     ;new
                MOV     R4,#250
jj:     CALL    DELAY_1MS
        CALL    DELAY_1MS
                DJNZ    R4,jj
                POP     04H
                RET


END
```









## กิตติกรรมประกาศ

โดยสาเหตุของการสนใจการทำโครงการนี้ขึ้นมาก็เนื่องจากทางกลุ่มผู้จัดทำ มีความสนใจเกี่ยวกับเทคโนโลยีด้านนี้อีกทั้งยังเป็นการแก้ปัญหาที่เกิดขึ้นจริงและเป็นประโยชน์ต่อฟาร์ม และท้ายที่สุดนี้ทางกลุ่มผู้จัดทำโครงการนี้ขอขอบพระคุณ อาจารย์ ชาญชัย ทองโสภา ที่ให้คำปรึกษาและให้ความช่วยเหลือในทุกด้าน และขอขอบคุณเพื่อนทุกคนที่มีส่วนช่วยให้โครงการนี้สำเร็จได้

# อ้างอิง


1. ธีรวัฒน์ ประกอบผล, "การประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์", แซทโฟร์ พริ้นติ้ง,2541
2. วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ่มพรจิตรวิไล, "ปฏิบัติการ ไมโครคอนโทรเลอร์ MCS-51 ฉบับ P98C51RD2", อินโนเวทีฟ เอ็กเพอริเมนต์, 191 หน้า
3. วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ่มพรจิตรวิไล, "เรียนรู้และปฏิบัติการ ไมโครคอนโทรเลอร์ MCS-51 แบบแฟลช ฉบับ AT98C5X ของ Atmel", อินโนเวทีฟ เอ็กเพอริเมนต์, 399 หน้า
4. บุญสืบ โพธิ์ศรี สมหมาย ปานเขียว และ โกมล ศิริสมบูรณ์เวช, "ไฟฟ้าและอิเล็กทรอนิกส์เบื้องต้น", ศูนย์ส่งเสริมอาชีวะ, 297 หน้า, 2546
5. บัณฑิต จามรภูติ, "คู่มือการใช้งาน Protel 99", บัณฑิต, 279 หน้า , 2544
6. http://www.thaiio.com


ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี