

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์

WIEGHT CONTROL MACHINE USING MICROCONTROLLER



26
ก.ค. 2551
11:00

เลขหมู่.....
เลขทะเบียน..... **83104**
วัน,เดือน,ปี..... - 5 ส.ค. 2551

b. 119 b31b3
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์

WIEGHT CONTROL MACHINE USING MICROCONTROLLER

ผู้จัดทำ

นางสาวกิริดา	มิเถาว์วัลย์	รหัสประจำตัว	48015281
นายชัยณรงค์	ทองอยู่	รหัสประจำตัว	48015284
นายทักษิณ	อ่อนแสง	รหัสประจำตัว	48015288

.....อาจารย์ที่ปรึกษา
(อาจารย์ทัตยา ปุคคละนนท์)

เครื่องข่งนำหนักควบคุมด้วยไมโครคอนโทรลเลอร์

โดย

นางสาวกิริดา	มิเถาวัลย์	48015281
นายชัยณรงค์	ทองอยู่	48015284
นายทักษิณ	อ่อนแสง	48015288

อาจารย์ที่ปรึกษา

อาจารย์ทัตยา ปุคละนันท์

ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอการออกแบบเครื่องข่งนำหนักควบคุมด้วยไมโครคอนโทรลเลอร์ โดยใช้โหลดเซลล์เป็นอุปกรณ์สำหรับตรวจวัดน้ำหนักของวัตถุที่ต้องการข่ง โครงสร้างหลักของเครื่องประกอบด้วยตัวถังบรรจุวัตถุ 3 ชนิด ชุดควบคุมการทำงานของมอเตอร์กระแสตรงเพื่อขับเกลียวควบคุมปริมาณการไหลของวัตถุให้ตกลงบนโหลดเซลล์ซึ่งถูกติดตั้งอยู่บนรางสไลด์ การทำงานของเครื่องเริ่มต้นจากการเลือกชนิดของวัตถุและระบุน้ำหนักที่ต้องการผ่านทางคีย์แพด (Keypad) โหลดเซลล์จะเคลื่อนที่ไปยังตำแหน่งของวัตถุที่ถูกเลือก และเมื่อข่งนำหนักได้ตามที่ถูกตั้งค่าไว้ โหลดเซลล์จะเคลื่อนที่กลับสู่ตำแหน่งเริ่มต้นเพื่อเข้าสู่กระบวนการบรรจุต่อไป สำหรับการควบคุมและประมวลผลในระบบจะใช้ไมโครคอนโทรลเลอร์ PIC16F877 โดยใช้โปรแกรม CCS PIC 'C' Compiler สำหรับการเขียนโปรแกรมภาษา C เพื่อควบคุมให้เครื่องทำงานตามต้องการ

WIEGHT CONTROL MACHINE USING MICROCONTROLLER

by

Miss Kirada Mithaowan

Mr. Chainarong Thongyoo

Mr. Thaksin Ornsang

Advisor

Miss Tattaya Pukkalanun

Academic Year 2007

ABSTRACT

A weight control machine using microcontroller is presented in this project. A load cell is employed for sensing the weight of substance. The main structure of the machine consists of a storage tower to fill 3 kinds of substance, a DC motor driver to run a thread of a spiral-shaped to carry the substance onto the loadcell, mounted on the sliding axis, and a keypad to set the type and weight of substance. The load cell will move to the position of selected substance and when the desired weight is achieved, the load cell will move back to its original position for substance to be conveyed to the packaging process afterward. Microcontroller PIC16F877 is used as a processor to implement the system. A C-program is written and compiled by CCS PIC 'C' Compiler to control the machine.

กิตติกรรมประกาศ

ในการทำโครงการปริญญาณิพนธ์เล่มนี้ หลายครั้งที่ประสบกับปัญหาต่างๆมากมาย แต่ก็ได้รับความช่วยเหลือจากอาจารย์หลายท่านที่ได้ให้คำแนะนำแนวทางในการแก้ไขและพัฒนาโครงการโดยตลอด โดยเฉพาะอย่างยิ่งอาจารย์ทัตยา ปุคคะสนันท์ ที่คอยให้คำแนะนำและให้คำปรึกษาในการทำงาน อีกทั้งยังเป็นแรงกระตุ้น และสร้างแรงผลักดันให้เกิดความคืบหน้าในการทำโครงการ ขอขอบคุณที่ธุรการภาค พิธีส โตร์ ที่ให้ความอนุเคราะห์ในการทำโครงการ ขอขอบพระคุณบิดา มารดาเป็นอย่างสูง ที่ให้การสนับสนุนด้านการศึกษาและให้กำลังใจเสมอมา และสุดท้ายนี้ขอขอบคุณท่านผู้อ่านทุกท่าน



ผู้จัดทำ

นางสาวกิริดา

มิเถาว์ลัย

นายชัยณรงค์

ทองอยู่

นายทักษิณ

อ่อนแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VIII
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์การจัดทำปริญญานิพนธ์	2
1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	
2.1 ไมโครคอนโทรลเลอร์ PIC16F877	4
2.1.2 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F877	6
2.1.3 การจัดสรรหน่วยความจำข้อมูล RAM ของ PIC16F877	7
2.1.4 การทำงานในโหมด PWM	10
2.1.4.1 หลักการสร้างสัญญาณ PWM	10
2.1.4.2 การกำหนดคาบเวลาของสัญญาณ PWM	10
2.1.4.3 กำหนดค่าดีวตี้ไซเคิลของสัญญาณ PWM	10
2.1.4.4 ความละเอียดของสัญญาณ PWM	11
2.1.5 โมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล PIC16F877	11
2.1.5.1 การทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล	13
2.1.5.2 การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณ	15
2.1.5.3 กระบวนการแปลงสัญญาณอนาลอกเป็นดิจิตอล	15
2.1.5.4 แรงดันอ้างอิงของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลใน PIC16F877	15
2.1.5.5 การใช้งานโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล	15
2.1.6 การเขียนโปรแกรมภาษา C	19
2.1.7 โปรแกรม CCS PIC ‘C’ คอมไพเลอร์	20

สารบัญ (ต่อ)

	หน้า
2.2 มอเตอร์ไฟฟ้ากระแสตรง	21
2.2.1 โครงสร้างการทำงานของมอเตอร์ไฟฟ้ากระแสตรง	21
2.2.2 การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง	22
2.3 การควบคุมมอเตอร์	23
2.3.1 IC L298	25
2.4 โหลดเซลล์ (Load Cell)	26
2.5 โมดูล LCD (Liquid Crystal Display Module)	27
2.6 Key pad	28
2.7 วงจรขยายสัญญาณแบบอินสตรูเมนต์	29
2.8 วงจรปรับค่าความชันและตำแหน่งศูนย์ (Zero and Span circuit)	30
บทที่ 3 โครงสร้างและการออกแบบ	
3.1 ขั้นตอนในการดำเนินงาน	33
3.2 โครงสร้างของเครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์	33
3.2.1 ส่วนการเก็บวัตถุดิบ	33
3.2.2 ส่วนการจ่ายวัตถุดิบ	34
3.2.3 ส่วนการรับวัตถุดิบ	34
3.2.4 ส่วนการควบคุม	34
3.3 แผนภูมิแสดงขั้นตอนการปฏิบัติงาน (Flow Chart)	37
บทที่ 4 การทดลองและผลการทดลอง	
4.1 การทดลองการทำงานของโหลดเซลล์	38
4.2 การทดลองควบคุมการไหลของวัตถุดิบ	39
4.3 การทดลองการชั่งน้ำหนัก	39
บทที่ 5 บทสรุปและวิจารณ์	
5.1 สรุปผลการทดลอง	42
5.2 ปัญหาที่พบและแนวทางแก้ไข	42
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	42
ภาคผนวก	44
โปรแกรมควบคุมการทำงาน	45
เอกสารอ้างอิง	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
1.1 บล็อกไดอะแกรมของระบบน้ำหนักควบคุมโดยใช้ไมโครคอนโทรลเลอร์	2
1.2 แสดงรายละเอียดของเครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์	3
2.1 ขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F877	5
2.2 การจัดสรรหน่วยความจำข้อมูล RAM และตำแหน่งรีจิสเตอร์ของ PIC16F877	9
2.3 ไดอะแกรมการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลใน PIC16F877	14
2.4 โมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลใน PIC16F877	17
2.5 โฟลวชาร์ตของโปรแกรมการใช้งานโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลใน PIC16F877	18
2.6 แสดงการแปลโปรแกรมภาษา C ให้เป็นภาษาเครื่อง	19
2.7 โปรแกรม CCS PIC 'C' คอมไพเลอร์	20
2.8 การทำงานเบื้องต้นของมอเตอร์	21
2.9 ดีซีมอเตอร์ (DC Motor)	22
2.10 ความกว้างของพัลส์ขนาดต่างๆ ของสัญญาณพัลส์ที่มีความถี่คงที่	23
2.11 แสดงสัญญาณพัลส์ วิตช์ มอเตอร์เกียร์ ที่มีดีวีดีไซเคิลต่างๆ	24
2.12 แสดงตำแหน่งขา IC เบอร์ L298	26
2.13 โหลดเซลล์ (Load Cell)	27
2.14 แสดงขาของโหลดเซลล์ (Load Cell)	27
2.15 โมดูล LCD 16x2	28
2.16 Key pad ป้อนค่า	28
2.17 วงจรขยายความแตกต่าง	29
2.18 วงจรขยายสัญญาณแบบอินสตรูเมนต์พื้นฐาน	29
2.19 วงจรขยายสัญญาณอินสตรูเมนต์	30
2.20 วงจรปรับความชันและตำแหน่งศูนย์และกราฟคุณสมบัติของวงจร	31
2.21 วงจรขยายสัญญาณอินสตรูเมนต์ต่อใช้งานร่วมกับวงจรปรับค่าความชันและตำแหน่งศูนย์	32
3.1 โครงสร้างเครื่องชั่งน้ำหนักด้วยไมโครคอนโทรลเลอร์	35
3.2 แสดงวงจรควบคุมระบบที่ออกแบบ	36
3.3 โฟลวชาร์ตการทำงานของเครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.1 แสดงกราฟเปรียบเทียบระหว่างน้ำหนัก และแรงดัน	39
4.2 Key pad ป้อนค่า	39
4.3 Display แสดงผลการเลือก TANK	40
4.4 Display แสดงผลการเลือกปริมาณน้ำหนัก	40
4.5 Display แสดงผลการกลับสู่ MENU	40



สารบัญตาราง

ตารางที่	หน้า
2.1 ค่า PCLATH<4:3> ในการเลือก Page ของ PIC16F877	6
2.2 ค่า RPO และ RP1 ในการเลือก Bank ของ PIC16F877	8
2.3 แสดงตารางการทำงานของ IC เบอร์ L298	26
4.1 แสดงตารางเปรียบเทียบระหว่างน้ำหนัก และแรงดัน	38
4.2 ตารางผลการทดลองการชั่งน้ำหนัก	41



บทที่ 1

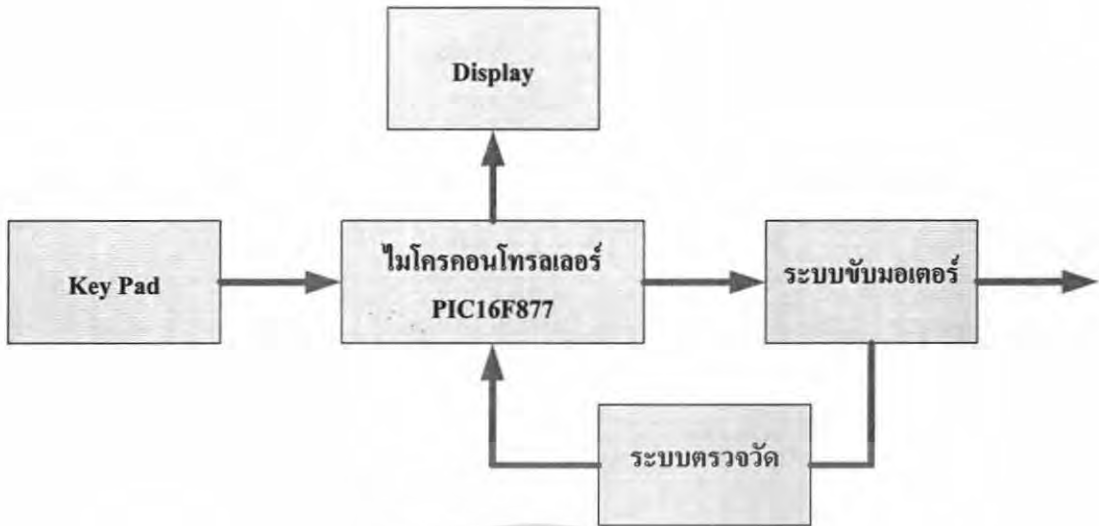
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันโลกกำลังพัฒนาเข้าสู่ยุคแห่งเทคโนโลยี ซึ่งเป็นยุคที่ระบบคอมพิวเตอร์และระบบไมโครคอนโทรลเลอร์มีความจำเป็นมากขึ้น อาจกล่าวได้ว่าในปัจจุบันระบบคอมพิวเตอร์และระบบไมโครคอนโทรลเลอร์ที่มีอยู่อย่างมากมายนั้น เป็นสิ่งที่อำนวยความสะดวกเพื่อตอบสนองความต้องการในด้านการทำงานและการใช้ชีวิตประจำวันของมนุษย์โดยเฉพาะด้านอุตสาหกรรม จะมีการใช้ระบบคอมพิวเตอร์หรือระบบไมโครคอนโทรลเลอร์มาใช้ในกระบวนการควบคุมอัตโนมัติ ยิ่งในอุตสาหกรรมที่ต้องมีการผสมวัตถุดิบ จำเป็นต้องมีการชั่ง ตวง วัด เพื่อให้ได้ผลิตภัณฑ์ตามที่ต้องการ เพื่อให้กระบวนการผลิตมีความประสิทธิภาพ ไม่เกิดความผิดพลาดใดๆ ในกระบวนการผลิต

ซึ่งในปัจจุบันก็ได้มีการผลิตอุปกรณ์ตรวจวัดให้มีความแม่นยำ และมีประสิทธิภาพมากขึ้น ทำให้การตรวจวัดมีความน่าเชื่อถือมากขึ้น จึงมีแนวคิดที่จะสร้างอุปกรณ์ในการชั่งตวงแบบอัตโนมัติขึ้น โดยสามารถชั่งตวงตามน้ำหนักที่ต้องการได้ โดยการป้อนปริมาณน้ำหนักที่ต้องการเพียงเท่านั้น เพื่อเพิ่มความแม่นยำในการชั่งตวง ซึ่งเกิดความผิดพลาดจากผู้ชั่งตวง และเพื่ออำนวยความสะดวกอีกในทางหนึ่ง

เครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์นี้ ใช้โหนดเซลล์เป็นตัวตรวจวัดน้ำหนัก ให้ไมโครคอนโทรลเลอร์ PIC เป็นตัวควบคุมกระบวนการชั่ง การทำงานของระบบได้มีการเขียนโปรแกรมไปเก็บไว้ในไมโครคอนโทรลเลอร์ PIC16F877 โดยใช้โปรแกรม CCS PIC “C” คอมไพเลอร์ เพื่อควบคุมให้ระบบทำงานตามที่กำหนด และใช้ IC ขั้วมอเตอร์เพื่อควบคุมการไหลของวัตถุดิบ นอกจากนี้ยังใช้รางสไลด์เพื่อรองรับวัตถุดิบ



รูปที่ 1.1 บล็อกโคอะแกรมของระบบน้ำหนักรควบคุมด้วยไมโครคอนโทรลเลอร์

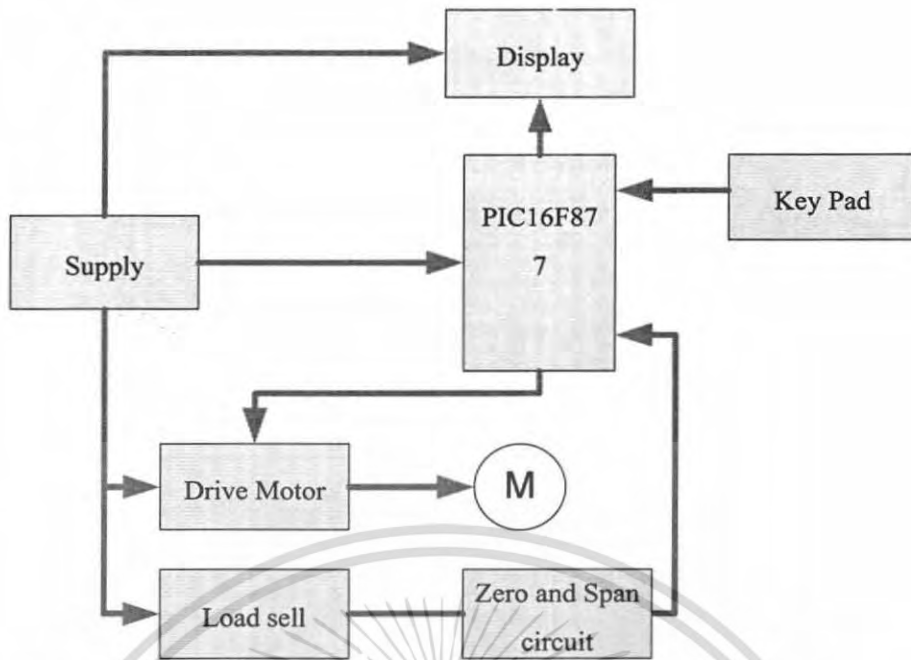
1.2 วัตถุประสงค์การจัดทำปริญญานิพนธ์

1. เพื่อศึกษาและเข้าใจการทำงานของโพลลเซลล์
2. เพื่อศึกษาการควบคุมระบบโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวประมวลผล
3. เป็นการศึกษาการควบคุมวงปิด
4. เพื่อให้ได้วงจรที่ใช้งานได้จริงที่มีความแม่นยำเที่ยงตรง
5. เพื่อนำความรู้และทฤษฎีที่ได้ศึกษามาประยุกต์ใช้งาน และสามารถแก้ไขปัญหาในการทำโครงการได้

1.3 ขอบเขตของปริญญานิพนธ์

1. ทำการออกแบบตัวถังไซโลที่ใช้บรรจุวัตถุดิบ ให้สามารถใช้งานได้
2. ศึกษาการทำงาน โครงสร้าง และคุณสมบัติของไมโครคอนโทรลเลอร์
3. ศึกษาหาข้อมูลการควบคุมการจ่ายวัตถุดิบ โดยใช้โพลลเซลล์เป็นอุปกรณ์ตรวจวัดน้ำหนักร
4. สามารถใช้คำสั่งเพื่อเลือกวัตถุดิบและปริมาณน้ำหนักรที่ต้องการ และสามารถตรวจวัดน้ำหนักรให้ตรงกับคำสั่ง ที่ถูกป้อนไป
5. สามารถรับ ส่งข้อมูลระหว่างอุปกรณ์ตรวจจับกับตัวควบคุม พร้อมทั้งสามารถปรับแต่งตัวควบคุมเพื่อให้ น้ำหนักรที่ได้ถูกต้องมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 แสดงรายละเอียดของเครื่องซึ่งนำหน้ควบคุมด้วยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

บทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับการทำโครงการ เพื่อปรับพื้นฐานให้สามารถทำความเข้าใจในเนื้อหาในส่วนอื่นๆ ต่อไป เนื้อหาและทฤษฎีที่จะกล่าวถึงต่อไปนี้จะเป็นส่วนสำคัญที่ใช้สนับสนุนแนวคิดในการทำการศึกษาคำโครงการ รวมทั้งกระบวนการวิเคราะห์และสรุปผลในประเด็นต่างๆ ตลอดจนเนื้อหาของการทำงานปริญญานิพนธ์ ซึ่งมีรายละเอียดที่สำคัญต่อระบบดังนี้

2.1 ไมโครคอนโทรลเลอร์ PIC16F877

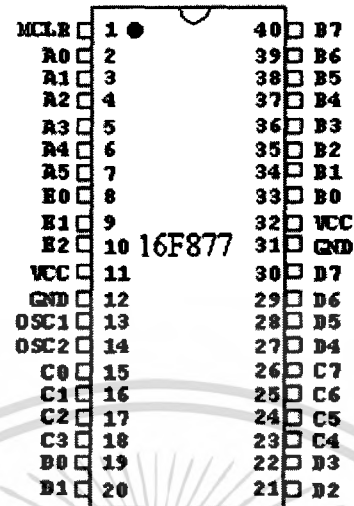
ไมโครคอนโทรลเลอร์ที่เลือกใช้จะเป็นไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877 เนื่องจากต้องการระบบที่มีขนาดเล็ก ทำงานได้แม่นยำสามารถควบคุมได้ง่าย และสามารถโปรแกรมเงื่อนไขการทำงาน รวมทั้งสั่งการได้โดยผ่านคอมพิวเตอร์ ไมโครคอนโทรลเลอร์ PIC16F877 มีโครงสร้าง 40 ขา มีพอร์ตใช้งานทั้งสิ้น 5 พอร์ต คือ พอร์ต A 6 บิต พอร์ต B 8 บิต พอร์ต C 8 บิต พอร์ต D 8 บิต และพอร์ต E 3 บิต เป็นพอร์ตแบบสองทิศทางสามารถเป็นได้ทั้งอินพุตและเอาต์พุตและยังเป็นพอร์ตที่สามารถแปลงสัญญาณอนาล็อกเป็นดิจิตอลได้อีกด้วย

คุณสมบัติของ PIC16F877 นี้

1. หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8k word (1 word = 14 บิต)
2. มีหน่วยความจำข้อมูล (Data Memory RAM) ขนาด 368 ไบต์
3. มีหน่วยความจำข้อมูลแบบ EEPROM ขนาด 256 ไบต์
4. ตอบสนองการอินเทอร์รัปต์ได้ทั้งหมด 14 แหล่ง
5. มี stack ให้ใช้ได้สูงสุด 8 ระดับ
6. สัญญาณนาฬิกา有多种ให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้
7. ทำงานที่ไฟเลี้ยง 2 VDC ถึง 5.5 VDC
8. Current Sink และ Current Source อยู่ที่ 25 mA
9. มี Timer/Counter 3 ตัว คือ Time 0 ขนาด 8 บิต, Time 1 ขนาด 16 บิต และ Time 2 ขนาด 8 บิต
10. มีโมดูล Capture/Compare/PWM (Pulse width Modulation) 2 ชุด
11. มีวงจรแปลงสัญญาณแอนาล็อกเป็นดิจิตอล (A/D Converter) ขนาด 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F877



รูปที่ 2.1 ขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F877

ไมโครคอนโทรลเลอร์ PIC16F877 มี 40 ขา มีขาสัญญาณต่างๆ ดังนี้

1. MCLR/V_{pp} : Master Clear (Reset) Input / Programming Voltage Input ทำหน้าที่เป็นขาสัญญาณ รีเซต (Reset) เมื่อขานี้ได้รับลอจิก 0 ไมโครคอนโทรลเลอร์จะถูกรีเซต และทำหน้าที่เป็นขาสัญญาณรับแรงดัน ขณะทำการบันทึกโปรแกรมลงหน่วยความจำของไมโครคอนโทรลเลอร์
2. VDD : Positive Supply (+2.00 V ถึง +5.5 V) ทำหน้าที่เป็นขาไฟเลี้ยงของไมโครคอนโทรลเลอร์
3. VSS : Ground ทำหน้าที่เป็นขากราวด์
4. OSC1/CLKIN : Oscillator Crystal Input / External Clock Source Input
5. OSC2/CLKOUT : Oscillator Crystal Output / External Clock Source Output ทั้งสองขาทำหน้าที่เป็นขาสัญญาณสำหรับต่อคริสตัล ในกรณีที่อยู่ในโหมดการใช้สัญญาณนาฬิกาจากภายนอก (Crystal Oscillator Mode)
6. RA0 – RA5 : พอร์ต A มีจำนวน 6 ขา เป็นพอร์ตแบบสองทิศทาง (Bi-directional I/O Port) คือ เป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ
7. RB0 – RB7 : พอร์ต B มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้บางขายังทำหน้าที่รับสัญญาณอินพุตจากการอินเทอร์รัปต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Interrupt) จากภายนอกด้วย

8. RC0 – RC7 : พอร์ต C มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ
9. RD0 – RD7 : พอร์ต A มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ
10. RE0 – RE7 : พอร์ต E มีจำนวน 3 ขา เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ

2.1.2 การจัดการหน่วยความจำโปรแกรมของ PIC16F877

PIC16F877 จะมีขนาด Program Memory ซึ่งสามารถอ้างได้ถึง 8 K byte โดย PIC16F877 จะมีขนาดความจำเท่ากับ $8\text{ K} \times 14\text{ bits}$ ซึ่งตำแหน่ง Reset Vector จะอยู่ที่ 0000h และ Interrupt Vector จะอยู่ที่ 0004h PIC จะแบ่ง Program Memory ออกเป็น Page ซึ่งแต่ละ Page ก็จะมีขนาด 2 Kbytes ซึ่งคำสั่ง CALL และ GOTO สามารถตั้งให้ Program Counter กระโดดไปมาในช่วง Page เท่านั้น แต่ถ้าเมื่อต้องกระโดดจาก Page หนึ่งไปยังอีก Page หนึ่ง ดังนั้นจึงต้องไปควบคุม PCLATH<4:3> (Bit Address ที่ 12 และ 13 ให้ชี้ไปยัง Page ที่ต้องการเสียก่อน หลังจากนั้นจึงเรียกคำสั่ง CALL หรือ GOTO ตามอีกที

การเลือก Page ของหน่วยความจำโปรแกรม จะต้องเลือกในรีจิสเตอร์ PCLATH โดยการระบุตำแหน่งที่บิต 3 และบิต 4

ตารางที่ 2.1 ค่า PCLATH<4:3> ในการเลือก Page ของ PIC16F877

PCLATH<4:3>	Page
00	0
01	1
10	2
11	3

เมื่อใช้คำสั่ง CALL ไปที่ Routine ใด Routine หนึ่งแล้ว แล้วจะใช้คำสั่ง RETURN ในการ RETURN กลับไปตำแหน่งเดิม การ กลับนั้นไม่จำเป็นต้องสั่ง PCLATH ให้ชี้ไปยัง Page ก่อนที่หน้าที่เรียก CALL เพราะค่า Address ดังกล่าวจะถูกเก็บไว้ใน STARCK อยู่แล้ว แต่สำหรับคำสั่ง GOTO เวลาข้าม Page จะต้องสั่งให้ PCLATH ชี้ไปยัง Page ที่ต้องการจะไปทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การจัดสรรหน่วยความจำข้อมูล RAM ของ PIC16F877

PIC16F877 มีหน่วยความจำข้อมูล RAM สำหรับใช้งานทั่วไป 368 bytes และมีรีจิสเตอร์ไฟล์ 8 bits 57 ตัว ดังรูปที่ 2-3 แต่ละแบงก์มีขนาดสูงสุด 128 bytes แต่มีการใช้งานได้จริงในแต่ละแบงก์ต่างกัน โดยในแต่ละแบงก์มีการจัดสรรพื้นที่ดังนี้

- Bank 0 มีช่วง Address 0x00-0x7F
 - Address 0x00-0x1F เป็นพื้นที่รีจิสเตอร์ไฟล์
 - Address 0x20-0x7F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 96 bytes
- Bank 1 มีช่วง Address 0x80-0xFF
 - Address 0x80-0x9F เป็นพื้นที่รีจิสเตอร์ไฟล์ แต่มีบาง Address ไม่ใช้งาน
 - Address 0xA0-0xEF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 bytes
 - Address 0xF0-0xFF บรรจุข้อมูลเหมือนกับใน Address 0x70-0x7F ใน Bank 0 เพื่อช่วยให้สามารถใช้ข้อมูลจาก Address 0x70-0x7F ได้ง่ายขึ้นโดยไม่ต้องเปลี่ยน Bank
- Bank 2 มีช่วง Address 0x100-0x17F
 - Address 0x100-0x10F เป็นพื้นที่รีจิสเตอร์ไฟล์ แต่มีบาง Address ไม่ใช้งาน
 - Address 0x110-0x11F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 bytes
 - Address 0x120-0x16F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 bytes
 - Address 0x170-0x17F บรรจุข้อมูลเหมือนกับใน Address 0x70-0x7F ใน Bank 0 เพื่อช่วยให้สามารถใช้ข้อมูลจาก Address 0x70-0x7F ได้ง่ายขึ้นโดยไม่ต้องเปลี่ยน Bank
- Bank 3 มีช่วง Address 0x180-0x1FF
 - Address 0x180-0x18F เป็นพื้นที่รีจิสเตอร์ไฟล์ แต่มีบาง Address ไม่ใช้งาน
 - Address 0x190-0x19F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 bytes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Address 0x1A0-0x1EF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 bytes
- Address 0x1F0-0x1FF บรรจุข้อมูลเหมือนกับใน Address 0x70-0x7F ใน Bank 0 เพื่อช่วยให้สามารถใช้ข้อมูลจาก Address 0x70-0x7F ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยน Bank

การเลือก Bank ของหน่วยความจำข้อมูล RAM จะต้องเลือกในรีจิสเตอร์ STATUS โดยการระบุตำแหน่งที่บิต 5 และบิต 6 (RPO, RP1)

ตารางที่ 2.2 ค่า RPO และ RP1 ในการเลือก Bank ของ PIC16F877

PR1:PR2	Bank
00	0
01	1
10	2
11	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾ 00h	Indirect addr. ⁽¹⁾ 80h	Indirect addr. ⁽¹⁾ 100h	Indirect addr. ⁽¹⁾ 180h
TMR0 01h	OPTION REG 81h	TMR0 101h	OPTION REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPAD0 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h	General Purpose Register 16 Bytes 117h	General Purpose Register 16 Bytes 197h
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCP2L 1Bh			
CCP2H 1Ch			
CCP2CON 1Dh			
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes 20h-7Fh	General Purpose Register 90 Bytes EFh-F7h	General Purpose Register 90 Bytes 120h-16Fh	General Purpose Register 90 Bytes 1ACh-1EFh
Bank 0	Bank 1	Bank 2	Bank 3

Unimplemented data memory locations, read as 0.
 * Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
 2: These registers are reserved, maintain these registers clear.

รูปที่ 2.2 การจัดสรรหน่วยความจำข้อมูล RAM และตำแหน่งรีจิสเตอร์ของ PIC16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การทำงานในโหมด PWM

โมดูล CCPx จะกำเนิดสัญญาณมอดูเลชันทางความกว้างพัลส์ (Pulse Width Modulation: PWM) ความละเอียด 10 บิต สัญญาณ PWM ที่สร้างขึ้นจะส่งออกทางขา RC2/CCP1 หรือ RC1/T10SI/CCP2

2.1.4.1 หลักการสร้างสัญญาณ PWM

1. กำหนดค่าให้เรจิสเตอร์ PR2 เพื่อกำหนดคาบเวลาของสัญญาณ PWM
2. กำหนดค่าควิต์ไซเคิล โดยเขียนข้อมูลลงในเรจิสเตอร์ CCPRxL ร่วมกับบิต 5-4 ของเรจิสเตอร์ CCRxCON
3. กำหนดให้ขาพอร์ต RC2/CCP1 หรือ RC1/T10SI/CCP2 เป็นเอาต์พุตเพื่อเป็นทางออกของสัญญาณ PWM โดยการเคลียร์บิตที่ 1 หรือ 2 ของเรจิสเตอร์ TRISC
4. กำหนดค่าปริสเกลเลอร์ของ TMR2 และเอ็นเนเบิลการทำงานของไทมเมอร์ 2
5. กำหนดให้โมดูล CCPx ทำงานในโหมด PWM

2.1.4.2 การกำหนดคาบเวลาของสัญญาณ PWM

ทำได้โดยการเขียนข้อมูลลงในเรจิสเตอร์ PR2 (เรจิสเตอร์คาบเวลาของไทมเมอร์ 2) แล้วนำค่าของ PR2 มาคำนวณหาคาบเวลาของสัญญาณ PWM

$$\text{คาบเวลาของสัญญาณ PWM} = (\text{ค่าในเรจิสเตอร์ PR2} + 1) \times 4 \times T_{\text{osc}} \times \text{ค่าปริสเกลเลอร์ของ TMR2} \quad (2.1)$$

โดยที่ T_{osc} คือ คาบเวลาของสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาที
ค่าในเรจิสเตอร์ทั้งหมดคำนวณในรูปของเลขฐานสิบ

$$\text{ความถี่ของสัญญาณ PWM} = 1/\text{คาบเวลาของสัญญาณ PWM} \quad (2.2)$$

2.1.4.3 กำหนดค่าควิต์ไซเคิลของสัญญาณ PWM

ทำได้โดยการเขียนข้อมูลลงในเรจิสเตอร์ CCPRxL ร่วมกับบิต 5-4 ของเรจิสเตอร์ CCRxCON ทำให้สามารถกำหนดรายละเอียดของสัญญาณ PWM ได้สูงสุดถึง 10 บิต โดย 8 บิตบนจะใช้ข้อมูลในเรจิสเตอร์ CCPRxL ส่วนใน 2 บิตล่างใช้ข้อมูลในบิต 5-4 ของเรจิสเตอร์ CCR1CON

$$\text{ควิตซ์ไซ้เกิดของสัญญาณ PWM} = (\text{CCPR1L: CCP1CON} \langle 5:4 \rangle)_{10\text{บิต}} \times T_{\text{osc}} \quad (2.3)$$

โดยที่ T_{osc} คือ คาบเวลาของสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาที
ค่าในรีจิสเตอร์ทั้งหมดคำนวณในรูปของเลขฐานสิบ

2.1.4.4 ความละเอียดของสัญญาณ PWM

จะขึ้นอยู่กับบิตสเกลเลอร์ของไทมเมอร์ 2 ในขณะที่ความละเอียดสูงสุดของสัญญาณ PWM จะสัมพันธ์กับความถี่ที่กำหนด ความถี่ของสัญญาณนาฬิกา และค่าปริสเกลเลอร์

$$\text{ความละเอียด (บิต)} = \frac{\log\left(\frac{f_{\text{osc}}}{f_{\text{PWM}} \times N}\right)}{\log 2} \quad (2.4)$$

โดยที่ f_{osc} คือ ความถี่ของสัญญาณนาฬิกาหลักมีหน่วยเป็น Hz

f_{PWM} คือ ความถี่ของสัญญาณ PWM มีหน่วยเป็น Hz

N คือ ค่าปริสเกลเลอร์ในไทมเมอร์ 2

สัญญาณ PWM จะมีความละเอียดสูงสุดเมื่อค่าของปริสเกลเลอร์เท่ากับ 1

2.1.5 โมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล PIC16F877

อีกหนึ่งโมดูลสำคัญที่ไมโครคอนโทรลเลอร์สมัยใหม่ต้องมี คือ โมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลและสำหรับไมโครคอนโทรลเลอร์ PIC16F877 มีความละเอียด 10 บิต 9.5 ช่องอินพุต สำหรับรุ่น 28 ขา และ 8 ช่องอินพุตสำหรับรุ่น 40 ขา โดยพอร์ตที่ใช้งานร่วมด้วยคือขาพอร์ต RA0 – RA3 – RA5 และ RE0-RE3 (เฉพาะในรุ่น 40 ขา) ในการทำงานเพื่อรองรับอินพุตจำนวนมากจะใช้วิธีการมัลติเพล็กซ์ ซึ่งควบคุมด้วยกระบวนการทางซอฟต์แวร์สำหรับการทำงานของ โมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลใน PIC16F877 เป็นแบบซิกเซสซีฟแอสซิงโครนัส รีจิสเตอร์ที่เกี่ยวข้อง

มีทั้งสิ้น 4 ตัว โดยแบ่งเป็นรีจิสเตอร์ควบคุมการทำงาน 2 ตัว คือ ACDON0 ซึ่งมีแอดเรสอยู่ที่ 0X1F และ ADCON1 มีแอดเรสอยู่ที่ 0X9F ส่วนอีก 2 ตัวคือ รีจิสเตอร์ ADRESH และ ADREST ซึ่งใช้ในการเก็บผลลัพธ์ของการแปลงสัญญาณโดย ADRESL มีแอดเรสอยู่ที่ 0X9F .ในแบงก์ 1 รีจิสเตอร์ทั้งสองมีขนาดตัวละ 8 บิต ต้องทำงานร่วมกันเพื่อรองรับข้อมูลดิจิตอล 10 บิต ที่ได้จากการแปลงสัญญาณอนาลอก

รีจิสเตอร์ ADCCON0

มีขนาด 8 บิต มีแอดเดรสที่ 0X1F ในแบงก์ เป็นรีจิสเตอร์หลักที่ใช้ควบคุมการทำงานของ โมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัล รายละเอียดการใช้งานดังนี้

บิต7	บิต 6	บิต 5	บิต 4	บิต 3	บิต2	บิต 1	บิต 0
ADCS1	ADCS2	CHS2	CHS1	CHS0	GO/DONE	-	ADON

ADCS1, ADCS0(A/D Conversion Clock Select bits) เลือกความถี่ของสัญญาณนาฬิกาที่ใช้ในการแปลงสัญญาณอนาลอกเป็นดิจิทัล

“00” ความถี่สัญญาณนาฬิกาหาร 2 (FOSC/2)

“01” ความถี่สัญญาณนาฬิกาหาร 8 (FOSC/8)

“10” ความถี่สัญญาณนาฬิกาหาร 32 (FOSC/32)

“11” ใช้ความถี่สัญญาณนาฬิกาจากวงจร RC (FRC)

นอกจากนั้นยังใช้ร่วมกับบิต ADCS2 (บิต 6 ในรีจิสเตอร์ ADCON !) เฉพาะในอนุกรมPIC16F877 เพื่อเลือกความถี่สัญญาณนาฬิกาได้มากขึ้น

CHS2, CHS1, CHS0, (Analog Channel bits บิต5, 4 และ3) บิตเลือกสัญญาณอนาลอก

“000” ช่อง 0 (AN0/RA0)

“001” ช่อง 1 (AN1/RA1)

“010” ช่อง 2 (AN2/RA2)

“011” ช่อง 3 (AN3/RA3)

“100” ช่อง 4 (AN4/RA4)

“101” ช่อง 5 (AN5/RE0) ไม่มีในรุ่น 28 ขา

“110” ช่อง 6 (AN6/RE1) ไม่มีในรุ่น 28 ขา

“111” ช่อง 7 (AN7/RE2) ไม่มีในรุ่น 28 ขา

GO/DON (A/D Conversion Status bit บิต 2) บิตแสดงสถานะการแปลงสัญญาณทำงานร่วมกับ ADON

กรณี ADON เป็น ...”1”

“0” การแปลงสัญญาณเสร็จสมบูรณ์ หรือยังไม่เริ่มการแปลงสัญญาณ

“1” ยังอยู่ในระหว่างการแปลงสัญญาณ

บิตนี้สามารถเคลียร์ได้ด้วยกระบวนการทางฮาร์ดแวร์ 2 ลักษณะคือ เมื่อการเปลี่ยนแปลงเสร็จ

สมบูรณ์จะเคลียร์เองอัตโนมัติ และเคลียร์เนื่องจากการเกิดเพาเวอร์ออนรีเซต

บิต 1 ไม่ใช้งาน กำหนดเป็น “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADON (A/D bit บิต 0) บิตเปิดการทำงานของโมดูลแสดงสัญญาณอนาลอกเป็นดิจิทัล

“0” ปิดการทำงาน

“1” เปิดการทำงาน

รีจิสเตอร์ ADCON 1

เป็นตัวรีจิสเตอร์ควบคุมการทำงานของโมดูลแปลงสัญญาณอนาลอก เป็นดิจิทัลที่ต้องทำงานร่วมกับADCON 0 มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 0x9F ในแบงก์ที่ 1 โดยรีจิสเตอร์ตัวนี้ใช้กำหนดการทำงานของขาพอร์ตที่เกี่ยวข้องกับโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลและใช้เลือกรูปแบบของข้อมูลผลลัพธ์ ที่ได้จากการแปลงสัญญาณมีรายละเอียดของการทำงาน ดังนี้ ADFM

(A/D Result Format Select bit บิต 7) บิตเลือกรูปแบบผลลัพธ์ของการแปลงสัญญาณ

“0” เลือกผลลัพธ์แบบชิดซ้าย (left justified)

“1” เลือกผลลัพธ์แบบชิดขวา (right justified)

ADCS 2 (A/D Conversion Clock Select bit – บิต 6) บิตเลือกความถี่ของสัญญาณนาฬิกาที่ใช้ในการแปลงสัญญาณอนาลอกเป็นดิจิทัลนี้มีเฉพาะในอนุกรม PIC16F87x ต้องใช้ร่วมกับบิตADCS1 และADCS0 ในรีจิสเตอร์ ADCON 0

2.1.5.1 การทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัล

โคออร์ดิเนตการทำงานของขาพอร์ตที่ใช้เป็นอินพุตอนาลอกจะต้องได้รับการกำหนดมาจากบิต PCFG3, PCFG0 ในรีจิสเตอร์ ADCON1 ว่าจะใช้ขาพอร์ตใดเป็นพอร์ตรับสัญญาณอนาลอกพอร์ตใดรับแรงดันอ้างอิงหรือพอร์ตใดเป็นพอร์ตดิจิทัล จากนั้นสัญญาณอนาลอกที่ต้องการแปลงจะถูกส่งผ่านวงจรเลือกของสัญญาณซึ่งควบคุมด้วยบิต CHS2, CHS0 ในรีจิสเตอร์ ADCON 0 ภาคอินพุตของวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัลภายใน PIC16F87x แสดงด้วยวงจรสมมูลประกอบด้วยวงจรสุ่มและเก็บสัญญาณ (Simple and hold) ที่มีตัวเก็บประจุ 120 พิโกฟารัดทำหน้าที่เก็บแรงดันทางอินพุตไว้ให้กับวงจรแปลงดิจิทัลเป็นอนาลอกและวงจรเปรียบเทียบภายนอกยังมีตัวทานอีก 3 ตัว ที่มีผลต่อเวลาที่ใช้ในการเก็บประจุนั้น คือ ความต้านทานของแหล่งจ่ายสัญญาณภายนอกและความต้านทานของสวิทช์สุ่มสัญญาณ หรือสมการถ้าตัวเก็บประจุมีค่าคงที่แล้ว ดังนั้นค่าคงที่ของเวลาจะขึ้นอยู่กับค่าของความต้านทานทั้งสามตัวโดยค่าสูงสุดนั้นมีค่าประมาณ (ข้อมูลจากผู้ผลิต) ค่าความต้านทานจะเปลี่ยนแปลงตามแรงดันที่ใช้ซึ่งสามารถหาค่าโดยประมาณได้จากกราฟ ส่วนค่าอื่นนั้นจะขึ้นอยู่กับค่าอิมพีแดนซ์ของแหล่งจ่ายแรงดันซึ่งไม่ควรเกินค่าเวลานี้มีความสำคัญเป็นอย่างมากต่อการใช้งาน โมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลเนื่องจากเป็นเวลาที่ตัวเก็บประจุจะต้องใช้ในการประจุแรงดันให้เต็มผลซึ่งมีผลต่อผลลัพธ์ของการแปลงสัญญาณ กล่าวคือ

พิจารณาได้ ดังนั้นเมื่อเขียน โปรแกรมให้ไมโครแปลงสัญญาณอนาลอกเป็นดิจิตอลทำงาน ต้องคำนึงถึงค่าเวลานี้ด้วย นั่นคือต้องหนดเวลาให้นานพอจนกระทั่งวงจรสุ่มและรักษาค่าสัญญาณเตรียมแรงดันสำหรับการแปลงสัญญาณมีค่าถูกต้องมากที่สุด

2.1.5.2 การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณ

การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณนี้ มีความสำคัญมากเนื่องจากมีผลต่อค่าเวลาในการแปลงสัญญาณ โดยในไมโครแปลงสัญญาณอนาลอกเป็นดิจิตอลของ PIC16F877 ต้องใช้เวลาแปลงสัญญาณอนาลอกเป็นข้อมูลดิจิตอล 10 บิตเท่ากับ $120 T_{AD}$ โดยค่าเวลา T_{AD} คือค่าเวลาที่ใช้ในการแปลงสัญญาณต่อบิต และค่าเวลา T_{AD} ที่น้อยที่สุดยังคงให้ผลการแปลงสัญญาณที่ถูกต้องคือ 1.6 ไมโครวินาที ดังนั้นการเลือกความถี่สัญญาณนาฬิกาต้องคำนึงถึงปัจจัยนี้ด้วยการเลือกความถี่ทำได้โดยการกำหนดข้อมูลในบิต ADCS 1 และ ADCS 0 ในอนุกรม PIC16F877 มาตรฐานและความสามาระใช้บิต ADCS 2 เพิ่มในกรณีไมโครคอนโทรลเลอร์อนุกรม PIC16F877 แสดงถึงความสัมพันธ์ของการเลือกคาบเวลาสัญญาณนาฬิกาของไมโครแปลงสัญญาณอนาลอกเป็นดิจิตอลกับความถี่สูงสุดของสัญญาณนาฬิกาหลักเพื่อให้ได้ค่า T_{AD} เท่ากับ 1.6 ไมโครวินาที

2.1.5.3 กระบวนการแปลงสัญญาณอนาลอกเป็นดิจิตอล

เมื่อสั่งให้ไมโครแปลงสัญญาณอนาลอกเป็นดิจิตอลทำงาน บิต GO/DONE ที่ใช้แสดงสถานะการทำงานจะเซต ถ้าหากมีการเคลียร์บิตเกิดขึ้น ในขณะที่การแปลงสัญญาณยังคงดำเนินอยู่ การแปลงสัญญาณถูกหยุดลงทันที แต่ข้อมูลของการแปลงสัญญาณที่ไม่สมบูรณ์จะไม่ได้รับการนำไปใช้งาน นั่นคือถ้าแปลงไม่สมบูรณ์ค่าในรีจิสเตอร์ ADRESH และ ADRESL จะยังเป็นค่าเดิมก่อนหน้านี้นี้แสดงกระบวนการแปลงสัญญาณที่สัมพันธ์กับค่าเวลาและบิตแสดงสถานะ

2.1.5.4 แรงดันอ้างอิงของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลใน PIC16F877

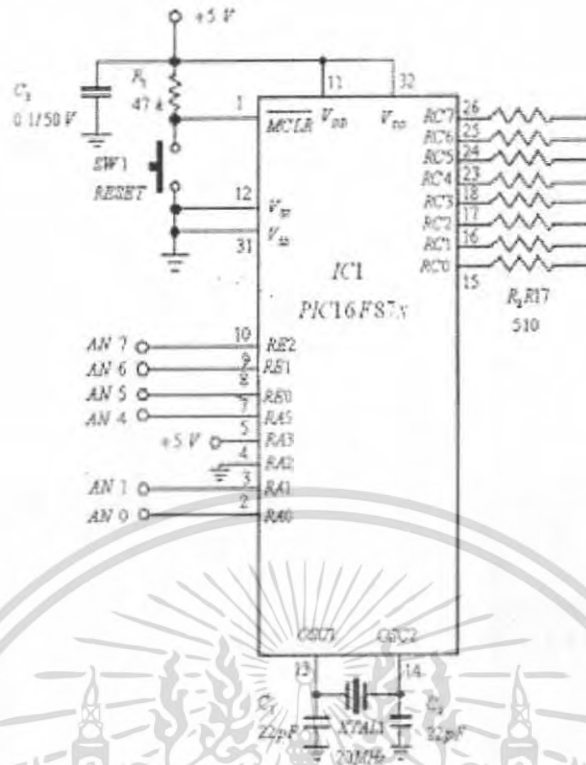
ในการใช้งาน ไมโครแปลงสัญญาณอนาลอกเป็นดิจิตอล ยังมีตัวแปรอีกตัวหนึ่งที่ต้องให้ความสนใจ คือ ค่าของแรงดันอ้างอิง ซึ่งต้องเลือกค่าแรงดันอ้างอิง V_{REF+} และ V_{REF-} ให้ถูกต้องและเหมาะสมด้วย โดยเลือกได้ที่บิต PCFG 3 : PCFG 0 ในรีจิสเตอร์ ADCON1 ซึ่งสามารถเลือกใช้แรงดันอ้างอิงได้จาก 2 แหล่งกำเนิด คือจากภายใน จะทำให้อินพุต V_{REF+} ถูกต่อเข้ากับขาไฟเลี้ยง V_{DD} และอินพุต V_{REF-} ถูกต่อเข้ากับขาราวด์หรือ V_{SS} : ซึ่งก็คือแรงดันไฟเลี้ยงนั่นเองแต่ถ้าใช้จากภายนอก อินพุต V_{REF+} จะต่อเข้ากับขาพอร์ต RA3 อินพุต V_{REF-} ต่อเข้ากับขาพอร์ต RA2 ดังนั้นเมื่อใช้แรงดันอ้างอิงภายนอกควรกำหนดให้ขาพอร์ตทั้งสอง (RA 3 และ RA 2) เป็นอินพุตเสมอ ประโยชน์ของการใช้แรงดันอ้างอิง V_{REF+} และ V_{REF-} คือ เพิ่มความละเอียดในการวัดแรงดันต่างๆ ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

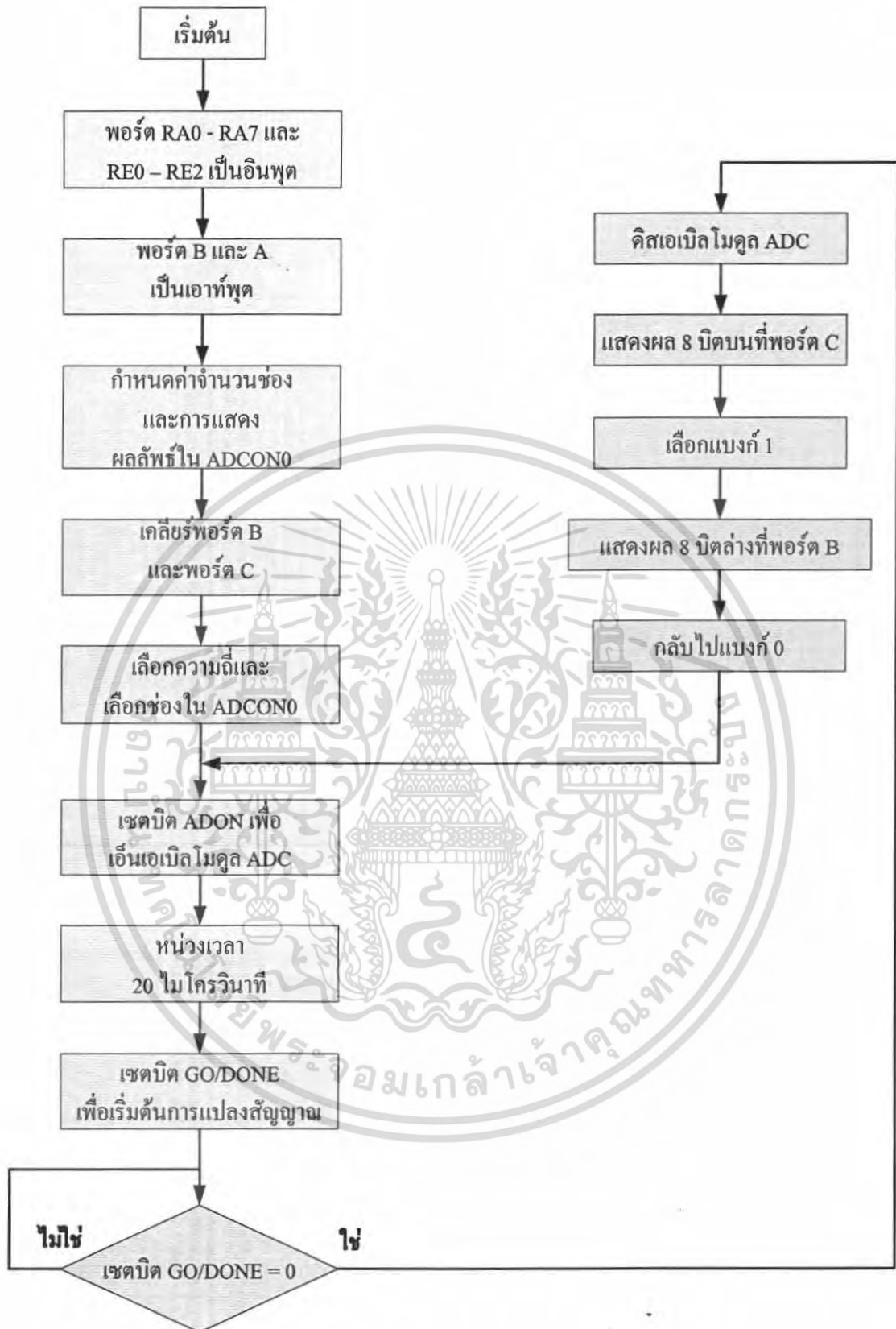
2.1.5.5 การใช้งานโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัล

ขั้นตอนการใช้งานโดยสรุปมีดังนี้

1. กำหนดค่าพารามิเตอร์เบื้องต้นต่างๆ
 - 1.1 กำหนดให้ขาพอร์ตที่ใช้งานเป็นอินพุตโดยเซตบิตในรีจิสเตอร์ TRISA หรือ TRISE
 - 1.2 เลือกจำนวนช่องและแหล่งจ่ายแรงดันอ้างอิงที่บิต PCFG3 : PCFG0 ในรีจิสเตอร์ ADCON 1
 - 1.3 เลือกรูปแบบผลลัพธ์จากบิต ADFM ในรีจิสเตอร์ ADCON1
 - 1.4 เลือกช่องสัญญาณผ่านทางรีจิสเตอร์ ADCON 1 และ ADCON 0
 - 1.5 เลือกสัญญาณนาฬิกาจากข้อมูลที่บิต ADCS 1 : ADCS 0 ในรีจิสเตอร์ ADCON 0
 - 1.6 เอ็นเอเบิลโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัลโดยเซตบิต ADON ในรีจิสเตอร์ ADCON0
2. ถ้ามีการใช้งานอินเทอร์รัปต์ให้กำหนดค่าต่าง ๆ ดังต่อไปนี้ (ถ้าไม่ใช้การอินเทอร์รัปต์ให้ข้ามข้อ 3.
 - 2.1 เคลียร์บิต ADIF ในรีจิสเตอร์ PIR1
 - 2.2 เซตบิต ADIE ในรีจิสเตอร์ PIE 1
 - 2.3 เอ็นเอเบิลการอินเทอร์รัปต์โดยการเซตบิต DIE ในรีจิสเตอร์ INTCON
3. หน่วงเวลาเพื่อให้พ้นช่วงเวลาแอกควิซิชัน (Acquisition time)
4. เริ่มต้นการแปลงสัญญาณด้วยการเซตบิต GO/DONE ในรีจิสเตอร์ ADCON 0 (ห้ามเซตบิตนี้ในคำสั่งเดียวกับที่เซตบิต ADON เพราะจะทำให้เกิดความผิดพลาดในการแปลงสัญญาณได้)
5. หน่วงเวลาเพื่อรอให้จบกระบวนการแปลงสัญญาณ ซึ่งทำได้โดย
 - 5.1 วนรอกำที่บิต GO/DONE ถ้าเป็น “0” หมายถึง เสร็จสิ้นการแปลงสัญญาณ
 - 5.2 รอให้เกิดอินเทอร์รัปต์ขึ้น
6. อ่านผลลัพธ์จากรีจิสเตอร์ ADRESH : ADRESL
7. ถ้าต้องการให้เกิดการแปลงสัญญาณอย่างต่อเนื่องให้หน่วงเวลาอย่างน้อย $2T_{AD}$ แล้วกลับไปทำให้ ข้อ 3. (ถ้ามีการใช้งานอินเทอร์รัปต์ให้กลับไปทำข้อ 2.1)



รูปที่ 2.4 โมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลใน PIC16F877



รูปที่ 2.5 โฟลวชาร์ตของ โปรแกรมการใช้งาน โมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลใน

PIC16F877

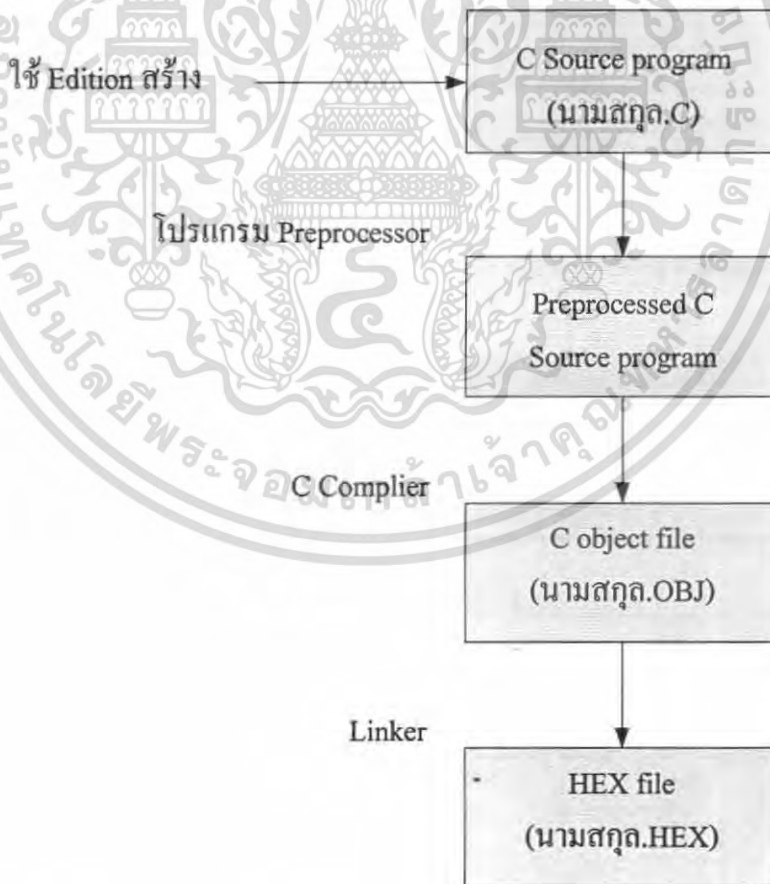
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การเขียนโปรแกรมภาษา C

ระบบไมโครคอนโทรลเลอร์นั้นจะทำงานด้วยวงจรทางไฟฟ้าและอุปกรณ์ภายนอกอย่างเดียวนั้นไม่ได้ จึงจำเป็นต้องมีชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ทำงานตามที่ต้องการ เก็บไว้ในหน่วยความจำสี่เหลี่ยมเก็บไว้ในโปรแกรม โดยชุดคำสั่งหรือโปรแกรมเหล่านี้จะทำการควบคุมระบบและการทำงานของอุปกรณ์ภายนอกที่เชื่อมต่อกับไมโครคอนโทรลเลอร์ ชุดคำสั่งที่เก็บไว้ในหน่วยความจำโปรแกรมจะอยู่ในรูปแบบของภาษาเครื่อง แต่ในการเขียนโปรแกรมเราสามารถเลือกเขียนโดยใช้ภาษาอื่น แล้วทำการแปล (Compiler) เป็นภาษาเครื่องก่อนส่งให้ไมโครคอนโทรลเลอร์ได้ ซึ่งสามารถเขียนได้หลายภาษา เช่น ภาษาแอสเซมบลี ภาษาซี การเขียนโปรแกรมด้วยภาษาซีจะช่วยให้การพัฒนาโปรแกรมสามารถทำได้เร็ว และง่ายกว่าการใช้ภาษาแอสเซมบลีและสามารถจำลองการทำงานของโปรแกรมได้ ดังนั้นจึงเลือกใช้ภาษา C ในการเขียนโปรแกรมควบคุมระบบ

ภาษา C จัดเป็นภาษาระดับสูงเพราะตัวคำสั่งต่างๆ ส่วนใหญ่ใกล้เคียงกับภาษามนุษย์ แต่ก็มีบางคำสั่งที่เป็นลักษณะคล้ายสัญลักษณ์ ดังนั้นเราอาจจัดได้ว่า ภาษา C เป็นภาษาระดับสูงที่อยู่ต่ำที่สุด

การแปลโปรแกรมภาษา C ให้เป็นภาษาเครื่อง



รูปที่ 2.6 แสดงการแปลโปรแกรมภาษา C ให้เป็นภาษาเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการสร้าง Source Program ภาษา C โดย Editor แล้ว ต้องทำกระบวนการ Preprocessor คือ คิวไฟล์ต่างๆ เข้ามารวมกับ Source Program ก่อนการ compiler จริงๆ ซึ่งกระบวนการนี้จะถูกทำโดยโปรแกรม Preprocessor แต่ Compiler ภาษา C ในปัจจุบันจะทำขั้นตอนนี้ให้โดยอัตโนมัติ ก่อนการ compiler จริงๆ ดังนั้นขั้นตอนนี้อาจเรียกว่า Compiler directive เมื่อทำการ compiler Source Program ภาษา C แล้วจะได้ object file ที่ได้เข้ากับ function ต่างๆ ที่ถูกเขียนไว้แล้วใน file libraries โดยใช้โปรแกรม Linker ก็จะได้ file ภาษาเครื่องที่สมบูรณ์ (นามสกุล.HEX)

2.1.7 โปรแกรม CCS PIC 'C' คอมไพเลอร์

ในโครงการนี้จะกล่าวถึงการเขียน โปรแกรมและพัฒนาโปรแกรมไมโครคอนโทรลเลอร์โดยใช้โปรแกรม CCS PIC 'C' คอมไพเลอร์ สำหรับการเขียนโปรแกรมจะใช้โครงสร้างภาษา C ซึ่งจะทำให้การพัฒนาโปรแกรมทำได้รวดเร็วขึ้นทั้งยังสามารถตรวจสอบและจำลองการทำงานของโปรแกรมโดยใช้โปรแกรม Proteus เพื่อตรวจสอบความถูกต้องและการทำงานของโปรแกรม CCS C เป็นซอฟต์แวร์สำหรับการเขียนโค้ดโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ PIC โดยแบ่งออกเป็น 2 เวอร์ชัน คือ หนึ่ง เวอร์ชันคอมไพเลอร์โปรแกรมผ่านคอมพิวเตอร์ (ในโหมดคอสต์ฟรี) และสอง เวอร์ชันคอมไพเลอร์โปรแกรมบนไมโครซอฟต์วินโดวส์ ซึ่งทำงานในรูปแบบ IDE (เขียนแล้วคอมไพล์โค้ดในตัว) ซึ่งเราใช้เวอร์ชันคอมไพเลอร์โปรแกรมบนไมโครซอฟต์วินโดวส์



รูปที่ 2.7 โปรแกรม CCS PIC 'C' คอมไพเลอร์

เครื่องมือช่วยในการใช้งาน PIC C Compiler

1. Valid Fuses

แสดงรายละเอียดของ #fuse ซึ่งใช้กำหนดคุณสมบัติเบื้องต้นการใช้งานไมโครคอนโทรลเลอร์สามารถดูได้โดยไปที่เมนู View เลือกเมนูย่อย Valid Fuses

2. Valid Interrupts

แสดงรายละเอียดของอินเทอร์รัปต์ที่มีใช้งานใน PIC16F877 ดูได้โดยไปที่เมนู View เลือกเมนูย่อย Valid Interrupts

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Serial Input/Output Monitor

ใช้แสดงข้อมูลที่ได้จากพอร์ตอนุกรม (RS-232) บนเครื่องคอมพิวเตอร์ เช่น เขียน โปรแกรมใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ แล้วต้องการส่งผลลัพธ์ผ่านพอร์ตอนุกรม (หน้าตาการทำงานในลักษณะเดียวกับโปรแกรม Hyper Terminal ของวินโดวส์) โดยไปที่เมนู Tools เลือกเมนูย่อย Serial Port Motor

4. Device Selection Tool

เครื่องมือเลือกเบอร์ไมโครคอนโทรลเลอร์ เพื่อดูรายละเอียดของไมโครคอนโทรลเลอร์ที่ต้องการใช้งาน

2.2 มอเตอร์ไฟฟ้ากระแสตรง

2.2.1 โครงสร้างการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์จะหมุนก็ต่อเมื่อได้รับแรงดันไฟฟ้า ซึ่งเมื่อป้อนแรงดันไฟตรงเข้าไปที่แปรงถ่าน ก็เกิดกระแสไหลผ่านขดลวดตัดสนามแม่เหล็ก ทำให้เกิดแรงผลักให้ขดลวดหมุน



รูปที่ 2.8 การทำงานเบื้องต้นของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดของแรงที่เกิดขึ้นนี้หาได้จาก

$$F = Bil \quad (2.6)$$

- เมื่อ F = แรงที่เกิดขึ้นบนตัวนำหนึ่งตัว
 B = ความหนาแน่นสนามแม่เหล็ก
 i = กระแสที่ไหลในตัวนำ
 l = ความยาวของตัวนำ

แรง F ที่เกิดขึ้น จะอยู่ในแนวที่ตั้งฉากกับสนามแม่เหล็ก และกระแสที่ผ่านในตัวนำนั้นๆ โดยคอมมิวเตเตอร์ (Commutator) เป็นตัวนำให้กระแสไหลผ่านตัวนำไปในทิศทางเดียวตลอดเวลา ภายใต้ขั้วแม่เหล็กแต่ละขั้ว ซึ่งช่วยทำให้เกิดแรงบิดไปในทิศทางเดียวกันอย่างต่อเนื่อง

ดีซีมอเตอร์ (DC Motor) แรงดัน 12 V ความเร็ว 300 rpm กระแสเมื่อจ่ายโหลดเต็มพิกัด 2 A และกระแสขณะไม่มีโหลด 200 mA



รูปที่ 2.9 ดีซีมอเตอร์ (DC Motor)

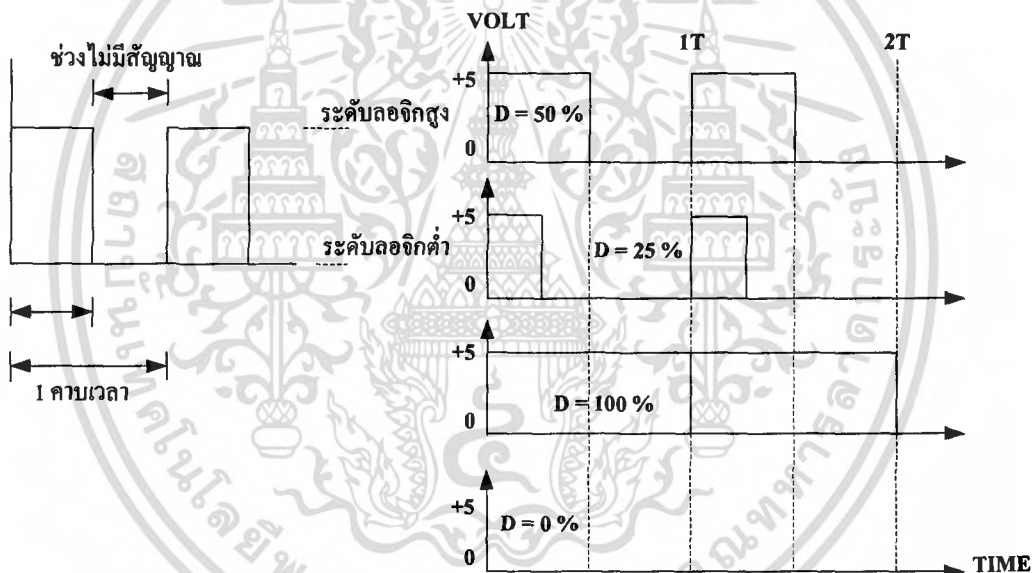
2.2.2 การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีควบคุมแบบพื้นฐาน เช่น การควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าได้ต่ออนุกรมกับมอเตอร์เพื่อลดแรงดันไฟฟ้าที่ป้อนให้กับมอเตอร์ หรือใช้วิธีการมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation) จะเป็นการปรับเปลี่ยนความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มีเปลี่ยนแปลงหรือเป็นการเปลี่ยนแปลงที่ค่าความถี่ไซเคิล (duty cycle) นั้นเอง ซึ่งค่าความถี่ไซเคิล คือช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าความถี่ไซเคิลมีค่าเท่ากับ 50 % ก็หมายถึงในหนึ่งรูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง ดังรูปที่ 3-4 และในทำนองเดียวกันถ้าหากคิดว่าค่าความถี่ไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าความถี่ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่า จะไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าความถี่ไซเคิลสามารถหาค่าได้จากค่าความสัมพันธ์ดังนี้

$$\text{ค่าความถี่ไซเคิล} = (\text{ช่วงของสัญญาณพัลส์} / \text{คาบเวลาทั้งหมดของสัญญาณ}) \times 100 \% \quad (2.7)$$



รูปที่ 2.10 ความกว้างของพัลส์ขนาดต่างๆ ของสัญญาณพัลส์ที่มีความถี่คงที่

2.3 การควบคุมมอเตอร์

ในส่วนการควบคุมมอเตอร์ ใช้ดีซีมอเตอร์ (DC Motor) เนื่องจากการควบคุมความเร็วและทิศทางการหมุนทำได้ง่าย คือผ่านการควบคุมโดยไมโครคอนโทรลเลอร์ PIC16F877 การทำงานโดยพื้นฐานของดีซีมอเตอร์ (DC Motor) ทิศทางการหมุนจะควบคุมได้โดยการควบคุมทิศทางกระแสที่จ่ายให้กับมอเตอร์ ส่วนความเร็วของมอเตอร์ควบคุมได้โดยการปรับ PWM ให้กับมอเตอร์ การควบคุมมอเตอร์โดยใช้ไมโครคอนโทรลเลอร์ PIC16F877 ก่อนที่สัญญาณควบคุมจาก

ไมโครคอนโทรลเลอร์จะไปควบคุมมอเตอร์ จะต้องมีการสำหรับขับมอเตอร์โดยใช้ IC เบอร์ L298

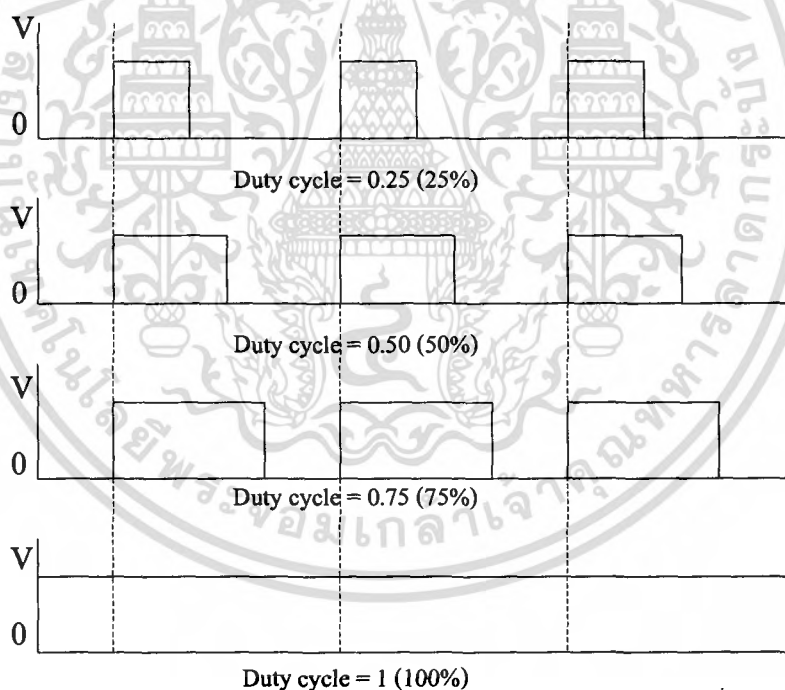
สัญญาณควบคุมแบบพัลส์ วิตช์ มอดดูเลชัน คือสัญญาณควบคุมที่มีความถี่คงที่ แต่ช่วงที่สัญญาณ ON และ OFF มีความสั้นยาวแตกต่างกันไปตามคิวดี้ไซเคิล (Duty cycle) ซึ่ง

$$\text{Duty cycle} = \frac{T_{ON}}{T} = \frac{T_{ON}}{T_{ON} + T_{OFF}} \quad (2.8)$$

ซึ่งเมื่อคิวดี้ไซเคิลเปลี่ยนไป แรงดันเฉลี่ยของสัญญาณก็จะเปลี่ยนด้วยโดย

$$V_{AV} = V \times \text{Duty cycle} \quad (2.9)$$

ตัวอย่างสัญญาณควบคุมแบบ PWM ที่มีคิวดี้ไซเคิล ต่างๆ



รูปที่ 2.11 แสดงสัญญาณพัลส์ วิตช์ มอดดูเลชัน ที่มีคิวดี้ไซเคิล ต่างๆ

เมื่อคิวดี้ไซเคิลของสัญญาณ PWM ที่จ่ายให้กับมอเตอร์มีค่ามากขึ้น ความเร็วของมอเตอร์ก็จะเพิ่มมากขึ้น

2.3.1 IC L298

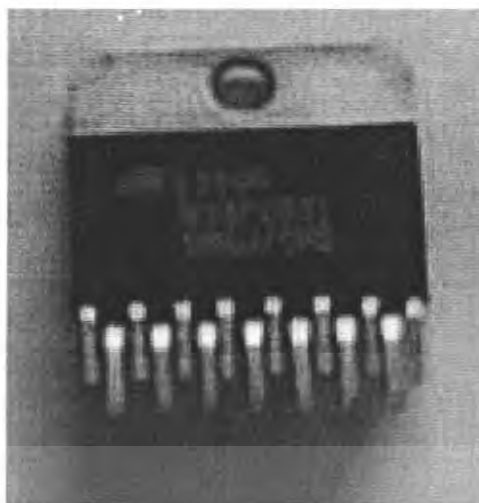
ทำหน้าที่รับสัญญาณควบคุมจากไมโครคอนโทรลเลอร์ PIC16F877 แล้วไปขับมอเตอร์ให้ทำงานตามคำสั่ง โดยปรกติ จะใช้ L298 ขับมอเตอร์สี่ขั้วสองทิศทาง สามารถจ่ายกระแสได้สูงถึง 2 A ต่อหนึ่งช่องสัญญาณออก

คุณสมบัติของ IC เบอร์ L298

- มีทรานซิสเตอร์ต่ออยู่ภายใน 8 ตัว
- ควบคุมการทำงานด้วยสภาวะลอจิก
- สามารถสร้างแหล่งจ่ายแรงดันได้ถึง 46 V
- กระแส DC สูงสุด 2A ต่อ หนึ่งช่องสัญญาณ
- สามารถตัดวงจรเมื่ออุณหภูมิสูงเกินขีดจำกัดได้

ซึ่ง IC เบอร์ L298 มีตำแหน่งขา และการทำงาน ดังนี้





รูปที่ 2.12 แสดงตำแหน่งขา IC เบอร์ L298

ตารางที่ 2.3 แสดงตารางการทำงานของ IC เบอร์ L298

INPUT		OUTPUT
ENA = VCC	IN1 = VCC	OUT 1 = VS
	IN1 = Low	OUT 1 = Low
	IN2 = VCC	OUT 2 = VS
	IN2 = Low	OUT 2 = Low
ENA = low	IN1 = X , IN2 = X	OUT 1,2 = High-Impedance
ENB = VCC	IN3 = VCC	OUT 3 = VS
	IN3 = Low	OUT 3 = Low
	IN4 = VCC	OUT 4 = VS
	IN4 = Low	OUT 4 = Low
ENB = low	IN3 = X , IN4 = X	OUT 3,4 = High-Impedance

VCC = แรงดันอ้างอิงลอจิกสูง VS = แรงดันที่แหล่งจ่าย Low = สภาวะลอจิกต่ำ หรือ กราวด์
 X = สภาวะลอจิกต่ำหรือสูง High-Impedance = สภาวะความต้านทานสูง

2.4 โหลดเซลล์ (Load Cell)

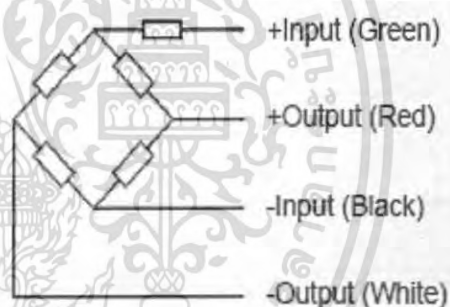
โหลดเซลล์เป็นอุปกรณ์ที่นำเอาสเตรนเกจมาประยุกต์ใช้ในการตรวจวัดแรงทางกลที่ไม่ทราบค่า (unknown force) หรือมวล บางครั้งนิยมเรียกแรงทางกลที่ไม่ทราบค่านี้ว่า “โหลด (load)” โครงสร้างทั่วไปของโหลดเซลล์ที่ต่อใช้งานร่วมกับสเตรนเกจ ประกอบด้วยสเตรนเกจจำนวนสี่ตัว ขีดติดอยู่กับเพลลาโหลด (load bearing) เมื่อมีแรงทางกลมากระทำกับปุ่มโหลด (load button) แล้วจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งผลทำให้เกิดความเครียด (stain) ขึ้นบนเพลาโหลด เป็นเหตุให้ค่าความต้านทานภายในตัวสเตรนเกจมีการเปลี่ยนแปลงไปตามขนาดของแรงที่มากระทำ และในส่วนของ การนำเอาสัญญาณเอาต์พุตที่ได้จากโหลดเซลล์นี้ไปใช้งานในรูปของสัญญาณไฟฟ้านั้นมักจะต่อร่วมกับวงจรบริดจ์แบบวีตสโตน เมื่อค่าความต้านทานของตัวสเตรนเกจในโหลดเซลล์มีค่าเปลี่ยนแปลงไปก็จะทำให้แรงดันไฟฟ้าเอาต์พุต V_o ที่ได้จากวงจรบริดจ์มีค่าเปลี่ยนแปลงตามไปด้วย จึงกล่าวได้ว่าขนาดของแรงดันไฟฟ้า V_o มีค่าแปรผันไปตามขนาดของแรงกระทำทางกลที่กำลังตรวจวัด

จากการทดสอบการตรวจวัดก่อนข้างแม่นยำ มีความแข็งแรงทนทาน ราคาไม่แพงนัก รวมทั้งยังสามารถทำการตรวจวัดค่าโหลดได้ในย่านที่กว้างอีกด้วย คุณสมบัติที่สำคัญประจำโหลดเซลล์ได้ดังนี้

- ความแม่นยำ ดีกว่า 0.25%
- ความเป็นเชิงเส้น มากกว่า 90%
- ความสามารถในการแสดงค่าซ้ำ ดีกว่า 0.25%
- ความสามารถในการกลับสู่สภาพเดิมเมื่อโอเวอร์โหลด มากกว่า 150%

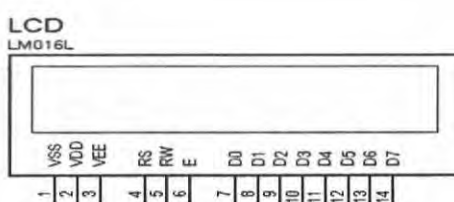


รูปที่ 2.13 โหลดเซลล์ (Load Cell)

รูปที่ 2.14 แสดงขาของโหลดเซลล์ (Load Cell)

2.5 โมดูล LCD (Liquid Crystal Display Module)

หน่วยแสดงโมดูล LCD (Liquid Crystal Display Module) หรือหน่วยแสดงผลแบบผลึกเหลว โดยโมดูล LCD จะมีอยู่สองชนิดด้วยกัน คือ แบบตัวอักษร และแบบกราฟิก ในที่นี้เราใช้โมดูล LCD แบบตัวอักษร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



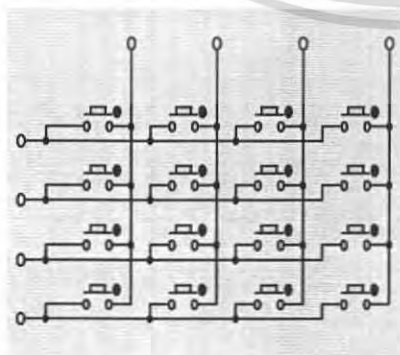
รูปที่ 2.15 โมดูล LCD 16x2

รายละเอียดขาสัญญาณของ โมดูล LCD มีดังนี้

- ขาที่ 1:** GND สำหรับต่อขากราวด์ของวงจร
- ขาที่ 2:** +Vcc ต่อกับไฟเลี้ยง
- ขาที่ 3:** Vo สำหรับปรับความสว่างของหน้าจอโมดูล LCD
- ขาที่ 4:** RS (Register Select) ขาเลือกการติดต่อกับรีจิสเตอร์คำสั่งหรือข้อมูล โดย “0” จะติดต่อกับคำสั่ง และ “1” จะติดต่อกับข้อมูล
- ขาที่ 5:** R/W (Read/Write control) ขาอ่านหรือเขียนข้อมูลกับโมดูล LCD
- ขาที่ 6:** E(Enable) ป้อนสัญญาณพัลส์ Enable ให้โมดูล LCD เริ่มทำงาน
- ขาที่ 7 ถึง 14:** D0-D7 (DATA) เป็นขาข้อมูล

2.6 Key pad

ทำหน้าที่เป็นตัวป้อนค่าที่ต้องการ ทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ PIC16F877 โดยจะต่อแบบเมตริกซ์เพื่อลดจำนวนขาในการเชื่อมต่อ เพราะ key pad จะเชื่อมต่อกับ CPU โดยตรงถ้าเรามีการใช้จำนวนปุ่มเยอะๆแล้ว หากไม่มีการจัดการกับการออกแบบ จะทำให้เปลืองจำนวนขาไปโดยเปล่าประโยชน์

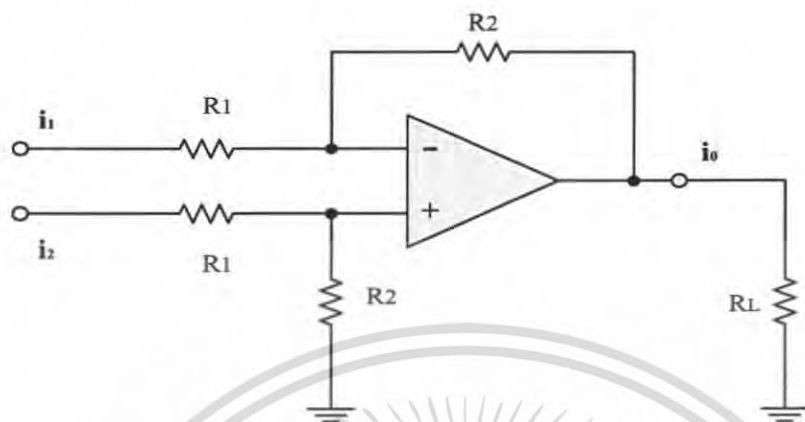


รูปที่ 2.16 Key pad ป้อนค่า

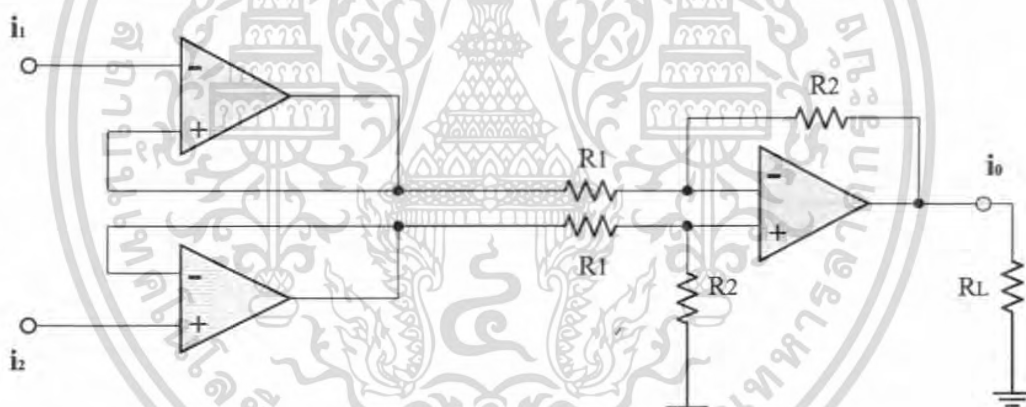
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 วงจรขยายสัญญาณแบบอินสตรูเมนต์

2.7.1 วงจรขยายสัญญาณแบบอินสตรูเมนต์แบบพื้นฐาน



รูปที่ 2.17 วงจรขยายความแตกต่าง



รูปที่ 2.18 วงจรขยายสัญญาณแบบอินสตรูเมนต์พื้นฐาน

จากสมการ $V_o = \frac{R_2}{R_1}(V_2 - V_1)$ จะได้ว่าแรงดันเอาต์พุต V_o จะขึ้นอยู่กับผลต่างของแรงดัน

อินพุต V_1 และ V_2 ถ้าแรงดันอินพุต V_1 และ V_2 มีค่าเท่ากันจะได้แรงดันเอาต์พุต V_o เท่ากับศูนย์ และจะเรียกสัญญาณแรงดันอินพุต V_1 และ V_2 นี้ว่าสัญญาณรวม (Common Mode Voltage) ความต้านทาน R_1 และ R_2 แต่ละในคircuit จะมีความสัมพันธ์กันมากเพื่อที่จะช่วยให้อัตราขยายของวงจรต่อสัญญาณรวมเท่ากับศูนย์ จากวงจรในรูปที่ 2.19 จะเป็นวงจรขยายความแตกต่างที่เพิ่มวงจรตาม

ศักดาหรือวงจรบัฟเฟอร์ (Buffer) เข้ามาบางส่วนหน้า ซึ่งวงจรนี้เรียกว่า วงจรขยายสัญญาณแบบอิน

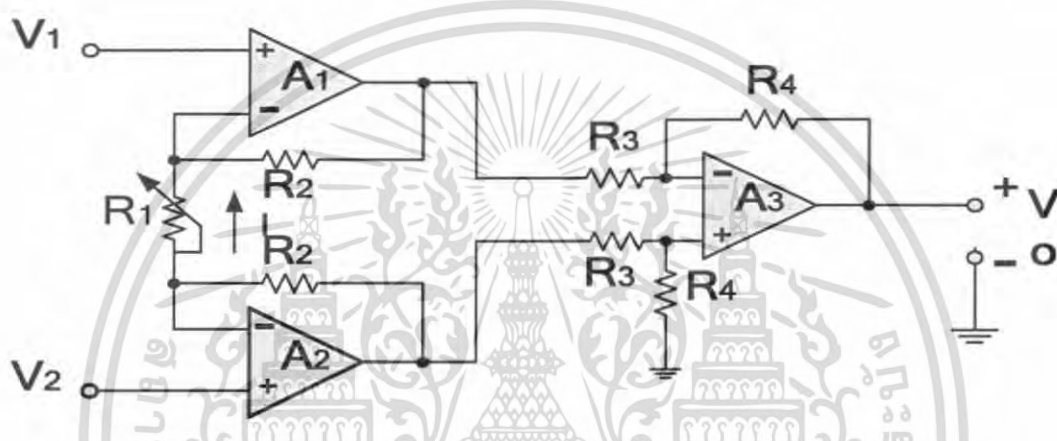
สตรูเมนต์พื้นฐาน (Basic Instrumentation Amplifier) โดยที่ความสัมพันธ์ระหว่างแรงดันอินพุต V_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ V_2 และค่าแรงดันเอาต์พุต V_0 จะเหมือนสมการ $V_0 = \frac{R_2}{R_1}(V_2 - V_1)$ แต่จะมีปัญหาในการปรับอัตราขยายของวงจรจะกระทำได้โดยการแปรค่าความต้านทาน R_2 ทั้ง 2 ตัวในวงจร ซึ่งการแปรค่าความต้านทาน R_2 ก็อาจจะส่งผลให้อัตราการขยายรวมของวงจรมีค่าไม่เท่ากับศูนย์ เนื่องจากความไม่สมนัยกันของค่าความต้านทานจึงได้มีการปรับปรุงวงจรขึ้นใหม่ ดังแสดงในรูปที่ 2.20 การแปรค่าอัตราขยายของวงจรจะแปรค่าความต้านทาน R_2 เพียงค่าเดียวโดยค่าอัตราขยายของวงจรมีค่า

$$V_0 = \left(1 + \frac{2R_1}{R_2}\right)(V_2 - V_1) \quad (2.10)$$

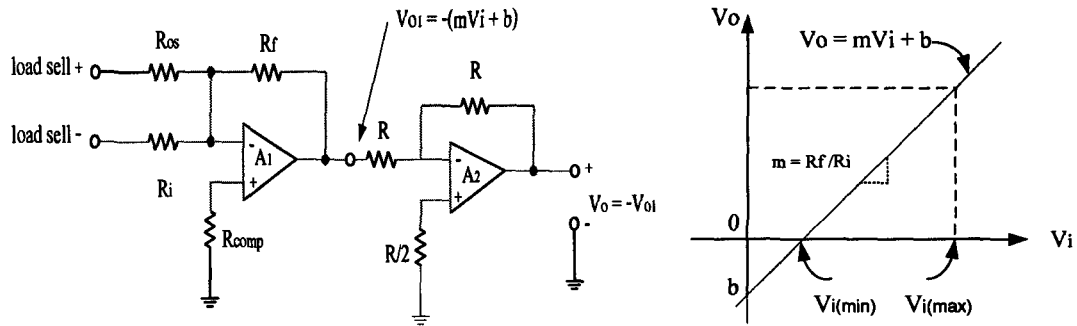


รูปที่ 2.19 วงจรขยายสัญญาณอินสตรูเมนต์

2.8 วงจรปรับค่าความชันและตำแหน่งศูนย์ (Zero and Span circuit)

จากตารางทดสอบโหลดเซลล์จะเห็นว่าแรงดันที่เอาต์พุตที่ออกมีค่าน้อยมาก แต่ไมโครคอนโทรลเลอร์ต้องการแรงดันในการประมวลผลอยู่ที่ $0-5\text{V}$ ดังนั้นจึงต้องใช้วงจรปรับค่าความชันและตำแหน่งศูนย์เป็นวงจรปรับสัญญาณให้อยู่ในช่วงที่ไมโครคอนโทรลเลอร์รับรู้ ซึ่งวงจร Zero and Span เป็นวงจรที่มีความสำคัญวงจรหนึ่งในระบบเครื่องมือวัดและคอมพิวเตอร์ โดยทำหน้าที่ส่งผ่าน (transmission) และปรับแต่งสัญญาณ (signal conditioner) ระหว่างสัญญาณเอาต์พุตกับอินพุตให้มีความสัมพันธ์เป็นไปตามเงื่อนไขที่ระบบต้องการ ซึ่งวงจรนี้จะทำหน้าที่เปลี่ยนแปลงความชัน (slope) และปรับแต่งตำแหน่งศูนย์ (zero) ของความสัมพันธ์ระหว่างสัญญาณเอาต์พุตกับอินพุต

วงจรปรับค่าความชันและตำแหน่งศูนย์และกราฟคุณสมบัติระหว่างแรงดันเอาต์พุต V_0 กับแรงดันอินพุต V_i ของวงจร แสดงในรูปที่ 2-20



รูปที่ 2.20 วงจรปรับความชันและตำแหน่งศูนย์และกราฟคุณสมบัติของวงจร

ซึ่งประกอบด้วยวงจรรวมสัญญาณ (summing amplifier) A1 และวงจรขยายสัญญาณแบบกลับเฟส (inverting amplifier) A2 ที่มีอัตราขยายเท่ากับ -1 ดังนั้นแรงดันเอาต์พุตของ A1 (V_{O1}) จึงมีค่าเท่ากับ

$$V_{O1} = -\left(\frac{R_f}{R_i} V_i + \frac{R_f}{R_{os}} V\right) \quad (2.11)$$

จากสัญญาณเอาต์พุตที่ได้จาก A1 จะถูกป้อนเป็นสัญญาณอินพุตให้กับ A2 ซึ่งต่อเป็นวงจรขยายสัญญาณที่มีอัตราขยายเท่ากับ -1 ทำให้แรงดันเอาต์พุตของ A2 (V_O) มีค่าเท่ากับ

$$V_o = V_{O2} = (-1)V_{O1} = -\left(\frac{R_f}{R_i} V_i + \frac{R_f}{R_{os}} V\right) \quad (2.12)$$

สมการ (2.12) แสดงให้เห็นว่าความสัมพันธ์ระหว่าง V_o กับ V_i ของวงจรมีอยู่ในรูปของสมการเส้นตรง และจากรูปแบบทั่วไปของสมการเส้นตรง

$$y = mx + b \quad (2.13)$$

ดังนั้น เมื่อทำการเปรียบเทียบสมการ (2.12) กับสมการ (2.13) แล้ว พบว่า

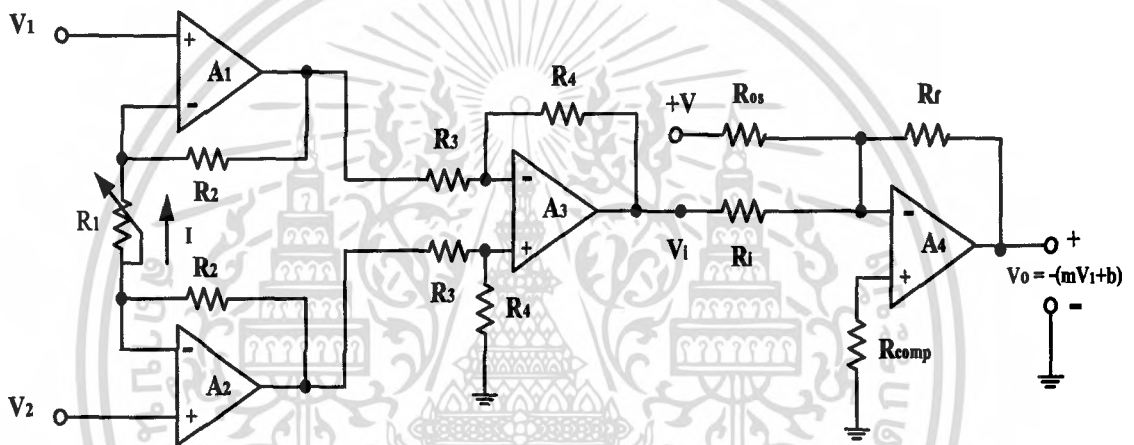
$m = \frac{R_f}{R_i}$ คือ ค่าความชันของกราฟ (slope) หรืออัตราขยาย (gain) ของวงจร

และ $b = \frac{R_f}{R_{os}} (\pm V)$ คือ จุดตัดแกน y ของกราฟ หรือตำแหน่งศูนย์

นั่นคือ การปรับค่าความชันสามารถทำการปรับแต่งได้จากค่าความต้านทาน R_f กับ R_i ของวงจร ในขณะที่ตำแหน่งศูนย์ของกราฟความสัมพันธ์จะปรับค่าได้ที่ความต้านทาน R_{os} หรือแรงดันอ้างอิง $+V$

ก่อนที่จะเข้าสู่ชุด Zero and Span circuit เราต้องทำการปรับค่าแรงดันที่ออกจากโพลเซลล์ไปเทียบกับกราวด์ก่อนมิฉะนั้นค่าที่ออกมาจะเกิดการผิดพลาดโดยได้เพิ่มวงจรขยายสัญญาณอินสตรูเมนต์ เข้าไป

ในทางปฏิบัติการปรับแต่งอัตราขยายแรงดัน A_v ของวงจรขยายสัญญาณอินสตรูเมนต์นี้สามารถกระทำได้โดยเลือกค่า R_1 , R_2 และ R_3 ให้คงที่ จากนั้นจึงทำการแปรค่า R_4 เพื่อให้ได้ A_v เป็นไปตามต้องการ



รูปที่ 2.21 วงจรขยายสัญญาณอินสตรูเมนต์ที่ทำงานร่วมกับวงจรปรับค่าความชันและตำแหน่งศูนย์

บทที่ 3

โครงสร้างและการออกแบบ

3.1 ขั้นตอนในการดำเนินงาน

ขั้นตอนในการแสดงผลกระบวนการซึ่งนำนักควบคุมด้วยไมโครคอนโทรลเลอร์ ประกอบด้วย

1. การสร้างตัวเครื่องซึ่งนำนักควบคุมด้วยไมโครคอนโทรลเลอร์ เป็นขั้นตอนในการออกแบบและสร้างตัวเครื่องซึ่งนำนักควบคุมด้วยไมโครคอนโทรลเลอร์ให้สามารถทำการซึ่งนำนักได้อย่างมีประสิทธิภาพ สามารถทำได้โดยการนำอุปกรณ์ต่างๆ ในส่วนของเครื่องซึ่งนำนักควบคุมด้วยไมโครคอนโทรลเลอร์มาต่อใช้งานร่วมกัน จากแบบที่กำหนด

2. การสร้างแผงควบคุม เป็นขั้นตอนในการออกแบบและสร้างแผงควบคุมที่ใช้ ติดต่อกับผู้ใช้งาน ซึ่งสามารถทำได้โดยการนำอุปกรณ์ต่างๆ ในส่วนของแผงควบคุมมาต่อใช้งานร่วมกัน จากแบบที่กำหนด

3. การเขียนโปรแกรมเพื่อควบคุมการซึ่งนำนัก เป็นขั้นตอนที่สำคัญและยากที่สุดในการที่จะออกแบบให้ทำงานได้ตามความต้องการ นอกจากนี้ยังจะต้องมีความสัมพันธ์กับอุปกรณ์อื่นๆ ในโครงงานอีกด้วย ซึ่งต้องอาศัยการศึกษาและประสบการณ์ ในการเขียนโปรแกรมอย่างมาก เพื่อให้มีการทำงานอย่างมีประสิทธิภาพ

3.2 โครงสร้างของเครื่องซึ่งนำนักควบคุมด้วยไมโครคอนโทรลเลอร์

หลังจากทำการศึกษาระบบต่างๆ มาแล้ว จึงได้ทำการวิเคราะห์และออกแบบระบบ เริ่มจากออกแบบโครงสร้างเครื่องตลอดจนรายละเอียดชิ้นส่วนของตัวเครื่อง เพื่อความเหมาะสมที่จะดำเนินการจัดสร้างรวมทั้งพัฒนาเครื่องจักรให้มีความสามารถในการทำงานตามที่ต้องการ โดยโครงสร้างหลักของเครื่องจะประกอบด้วย ส่วนการเก็บวัตถุดิบ ส่วนการจ่ายวัตถุดิบ ส่วนรับการวัตถุดิบ ส่วนการควบคุม รายละเอียดต่างๆ สามารถแสดงได้ดังต่อไปนี้

3.2.1 ส่วนการเก็บวัตถุดิบ

ทำหน้าที่เก็บวัตถุดิบที่ต้องการซึ่งนำนัก และส่งวัตถุดิบออกไปยังกระบวนการอื่นๆ ต่อไป ซึ่งมีส่วนประกอบที่สำคัญคือ ตัวถังไซโล (Silo) ทำมาจากแผ่นอะคริลิก ประกอบกันเป็นถังบรรจุวัตถุดิบ มีช่องใส่วัตถุดิบ 3 ช่อง แต่ละช่องมีขนาด 20×30 เซนติเมตร ด้านล่างของตัวถังไซโล

(Silo) แต่ละช่องได้เจาะรูเพื่อใช้เป็นช่องทางที่วัตถุดิบจะไหลลงไปในภาชนะที่รองรับบนโหลด เซลล์ที่ใช้ตรวจวัดน้ำหนัก

3.2.2 ส่วนการจ่ายวัตถุดิบ

ทำหน้าที่จ่ายวัตถุดิบ และส่งวัตถุดิบออกไปยังกระบวนการอื่นๆ ต่อไป ซึ่งมีส่วนประกอบที่สำคัญคือ ท่อเกลียว มีลักษณะเป็นท่อพลาสติกขนาด 40 เซนติเมตร มีเกลียวอยู่ภายในท่อ ปลายท่อพลาสติกด้านหน้าเจาะรูข้างล่างต่อกับท่อพลาสติกใสขนาด 10 เซนติเมตร เพื่อให้วัตถุดิบไหลลงสู่ภาชนะได้สะดวก ปลายท่ออีกด้านหนึ่งต่อกับมอเตอร์ที่จะมาหมุนเกลียวเพื่อควบคุมการไหลของวัตถุดิบ ในส่วนของมอเตอร์ขับเคลื่อนเลือกใช้เป็น DC มอเตอร์ มีความเร็วรอบ 300 รอบต่อนาที เพลาของมอเตอร์ต่ออยู่กับ แกนเกลียวเพื่อเป็นตัวขับเคลื่อนเกลียว โดยมีวงจรขับมอเตอร์เป็นตัวควบคุมการหมุนของมอเตอร์ ใช้ IC เบอร์ L298 มาขับให้มอเตอร์ทำงาน

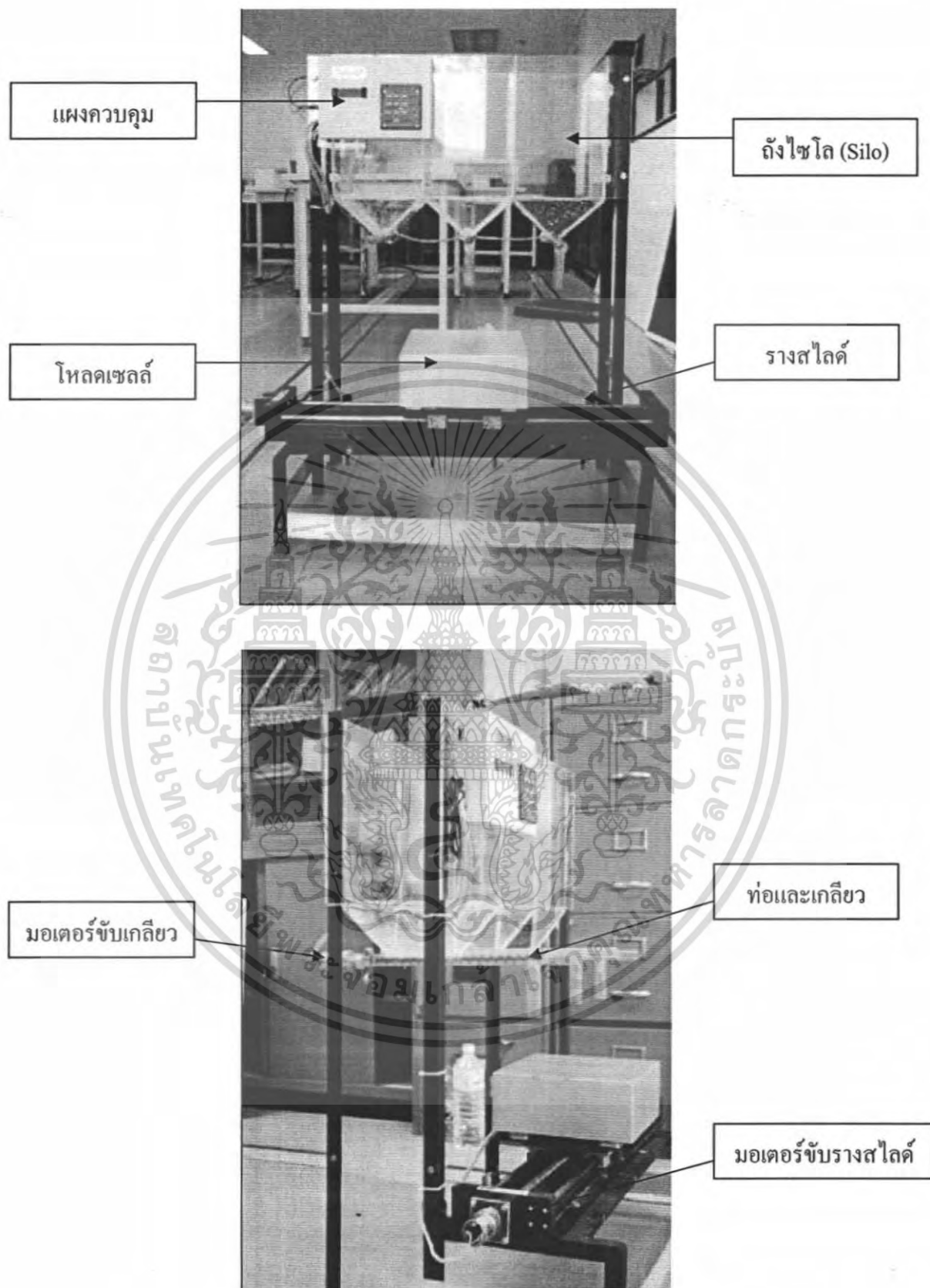
3.2.3 ส่วนการรับวัตถุดิบ

มีหน้าที่รับวัตถุดิบมาจากกระบวนการเก็บ และจ่ายวัตถุดิบ ทำหน้าที่เลื่อนไปรับวัตถุดิบที่ออกจากไซโล (silo) ทำการตรวจวัดน้ำหนัก แล้วส่งค่าไปยังส่วนควบคุมเพื่อให้ได้น้ำหนักตามที่ต้องการ ซึ่งมีส่วนประกอบที่สำคัญคือ รางสไลด์ ทำหน้าที่เลื่อนภาชนะไปรับวัตถุดิบ ยังจุดที่วัตถุดิบถูกปล่อยออกมา โดยใช้มอเตอร์เป็นตัวขับเคลื่อนในการทำงาน ภาชนะรองรับวัตถุดิบที่ถูกปล่อยออกมา และโหลดเซลล์ตรวจวัดน้ำหนัก ทำหน้าที่ชั่งน้ำหนักวัตถุดิบที่ถูกปล่อยลงมา แล้วส่งค่าไปยังส่วนการควบคุมเพื่อสั่งให้ปล่อย หรือหยุดวัตถุดิบ

3.2.4 ส่วนการควบคุม

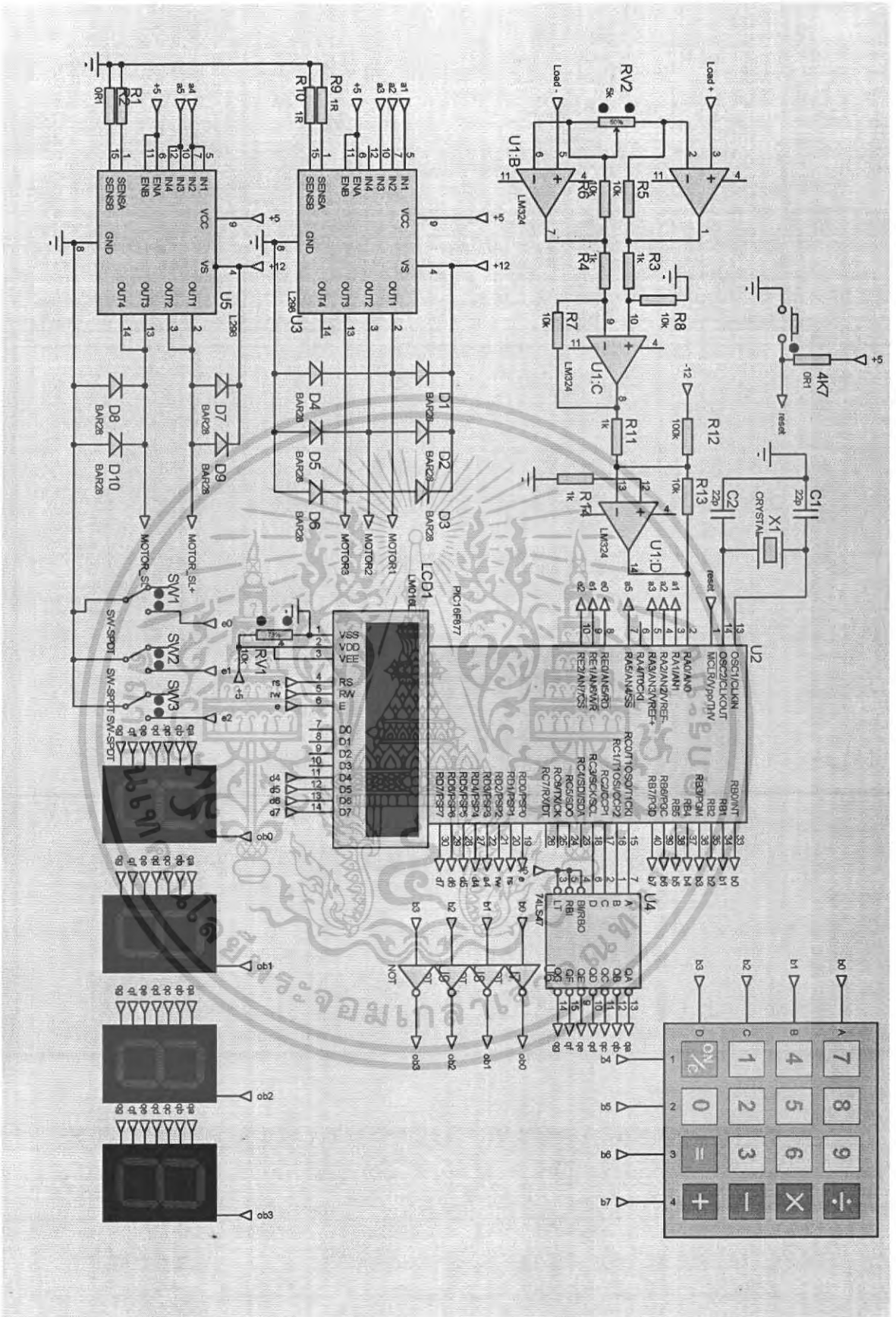
ในส่วนการควบคุมจะประกอบไปด้วยส่วนหลักๆ คือ ไมโครคอนโทรลเลอร์ PIC16F877 ทำการเขียนโปรแกรมควบคุมส่วนต่างๆ โดยใช้โปรแกรม CCS C COMPILER ใช้ Key Pad ในการป้อนค่า รับคำสั่งติดต่อกับผู้ใช้งานที่ต้องการทำการขายการ ส่วนการแสดงผลใช้จอ LCD ขนาด 16 ตัวอักษร 2 บรรทัด เพื่อแสดงเมนูต่างๆ และแสดงผลของปริมาณน้ำหนักที่ต้องการ การต่อวงจรในส่วนการควบคุมจะแสดงในรูปที่ 3.2

การทำงานของส่วนการควบคุมมีการทำงานหลักๆ คือ รับค่าปริมาณน้ำหนัก และชนิดของวัตถุดิบมาเช็คว่ามีน้ำหนักที่ได้ของวัตถุดิบได้หรือยัง ถ้ายังไม่ได้ไมโครคอนโทรลเลอร์ PIC16F877 ก็จะสั่งการให้มอเตอร์ทำงานต่อไป และถ้าได้น้ำหนักที่ต้องการแล้วมอเตอร์จะหยุดทำงาน สำหรับโปรแกรมที่เขียนให้กับไมโครคอนโทรลเลอร์ แสดงไว้ในภาคผนวก ข



รูปที่ 3.1 โครงสร้างเครื่องชั่งน้ำหนักด้วยไมโครคอนโทรลเลอร์

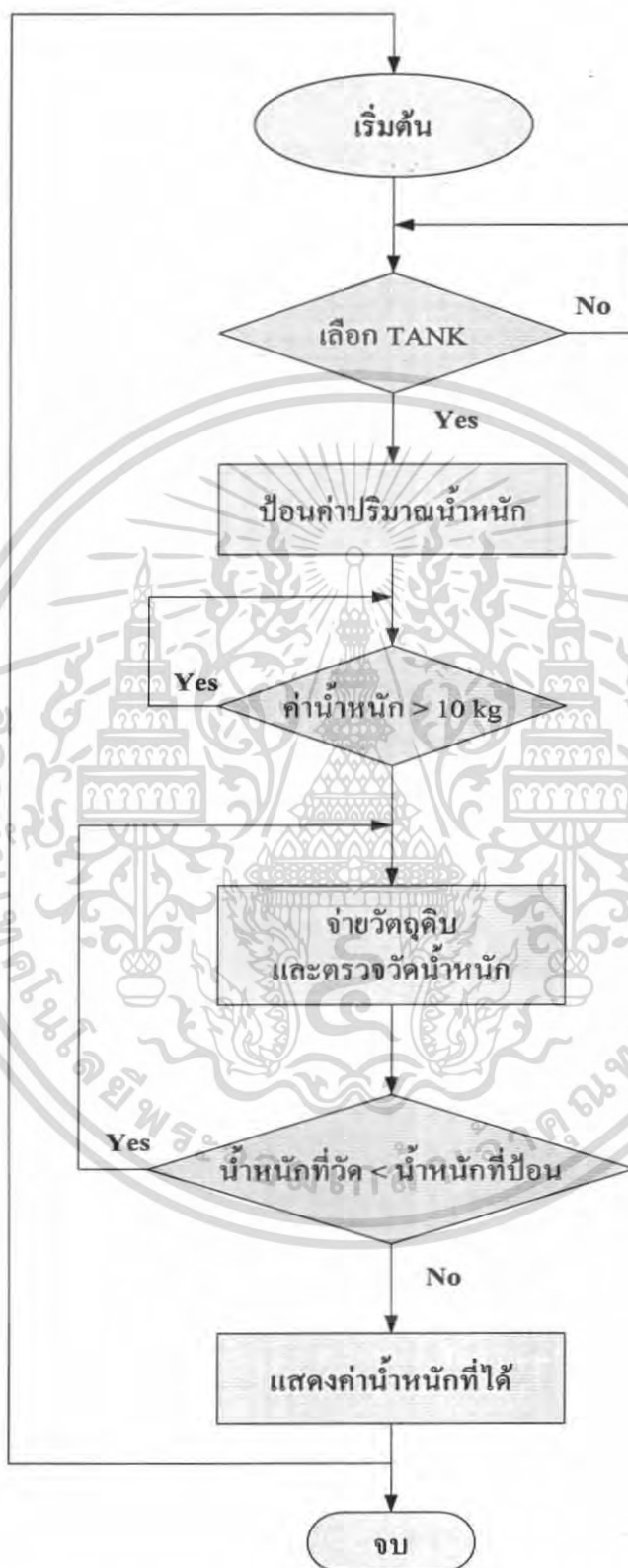
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงวงจรควบคุมระบบที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนภูมิแสดงขั้นตอนการปฏิบัติงาน (Flow Chart)



รูปที่ 3.3 โฟลวชาร์ตการทำงานของเครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

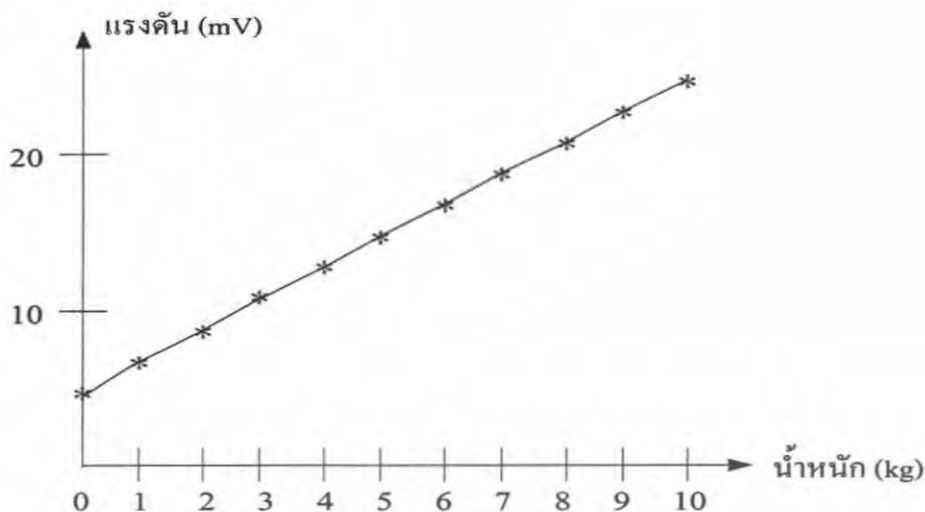
ในบทนี้จะเป็นการทดลองส่วนของระบบควบคุมที่สร้างขึ้นมาทดลองเพื่อทดสอบว่าสามารถควบคุมกระบวนการชั่งน้ำหนักได้หรือไม่ โดยผู้ใช้งานสามารถควบคุมกระบวนการชั่งน้ำหนักได้ด้วยตนเอง และทำการทดลองการชั่งน้ำหนัก การไหลของวัตถุ

4.1 การทดลองการทำงานของโพลีเซลล์

จากการทดสอบการทำงานของโพลีเซลล์จะได้ตารางเปรียบเทียบระหว่างน้ำหนัก (รวมแท่นชั่ง) และแรงดัน (เมื่อจ่ายแรงดันให้โพลีเซลล์ 5 V)

ตารางที่ 4.1 แสดงตารางเปรียบเทียบระหว่างน้ำหนัก และแรงดัน

น้ำหนัก (kg)	Voltage (mV)
0	2.35
1	3.35
2	4.35
3	5.20
4	6.35
5	7.40
6	8.40
7	9.40
8	10.40
9	11.40
10	12.40



รูปที่ 4.1 กราฟแสดงการเปรียบเทียบระหว่างน้ำหนัก และแรงดัน

4.2 การทดลองควบคุมการไหลของวัตถุคิบ

การควบคุมการไหลของวัตถุคิบในกระบวนการนั้น ทำการควบคุมโดยการควบคุมความเร็วของมอเตอร์ให้มีค่าคงที่ เมื่อได้ปริมาณน้ำหนักที่ต้องการแล้วไมโครคอนโทรลเลอร์จะสั่งการให้มอเตอร์หยุดทำงาน แต่ถ้ายังไม่ได้ปริมาณน้ำหนักที่ต้องการก็จะสั่งให้ทำงานต่อไป

4.3 การทดลองการชั่งน้ำหนัก

ขั้นตอนการทดลอง

1. กดปุ่ม **MENU** ที่ Key pad ป้อนค่า



รูปที่ 4.2 Key pad ป้อนค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กดปุ่ม ► ที่ Key pad ป้อนค่าเพื่อเลือก **TANK** เช่น เลือก **TANK >3<** เมื่อป้อนค่าเสร็จแล้วกดปุ่ม **ENTER**



รูปที่ 4.3 Display แสดงผลการเลือก TANK

3. กดปุ่ม ► ที่ Key pad ป้อนค่าเพื่อเลือกน้ำหนัก เช่น เลือกน้ำหนัก 2 kg เมื่อป้อนค่าเสร็จแล้วกดปุ่ม **ENTER**



รูปที่ 4.4 Display แสดงผลการเลือกปริมาณน้ำหนัก

4. เมื่อได้น้ำหนักที่ต้องการแล้ว **Display** แสดงผลจะยังคงแสดงผลค่าน้ำหนักที่เลือกค้างอยู่ หากต้องการทำรายการใหม่ ให้กดปุ่ม **MANU** อีกครั้ง แล้วทำตามขั้นตอนข้อ 1 - 3 ใหม่



รูปที่ 4.5 Display แสดงผลการกลับสู่ MENU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองการควบคุมน้ำหนัก จะได้ตารางผลการทดลองการชั่งน้ำหนัก ดังนี้

ตารางที่ 4.2 ตารางผลการทดลองการชั่งน้ำหนัก

น้ำหนัก (kg)	Vout ที่โหลดเซลล์ (mV)	Vout ที่วงจร Zero span (mV)	น้ำหนักที่ได้ (kg)
0	2.35	0	0.02
1	3.35	0.51	1.11
2	4.35	1.01	2.09
3	5.20	1.52	3.10
4	6.35	2.01	4.13
5	7.40	2.53	5.09
6	8.40	3.02	6.12
7	9.40	3.51	7.08
8	10.40	4.01	8.11
9	11.40	4.52	9.10
10	12.40	5.03	10.90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 สรุปผลการทดลอง

จากการทดลองสามารถควบคุมการไหลของวัตถุได้โดยให้มอเตอร์หมุนที่ความเร็วคงที่ แต่การสั่งจ่ายหรือหยุดวัตถุจากการ ON – OFF มอเตอร์ ทดสอบการทำงานของโพลีเซลล์พบว่าสามารถตรวจวัดน้ำหนักได้อย่างแม่นยำเที่ยงตรง และส่งค่าที่ได้ไปยังไมโครคอนโทรลเลอร์ได้ดี ไม่มีปัญหาเกิดขึ้น ในส่วนการควบคุมเครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์สามารถควบคุมการทำงานและเชื่อมต่อกับไมโครคอนโทรลเลอร์ PIC16F877 ได้ตามที่ต้องการ อีกทั้งยังสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ

5.2 ปัญหาที่พบและแนวทางแก้ไข

จากการศึกษาและทำโครงการนี้ในช่วงแรกพบปัญหาหลายอย่าง เนื่องจากโครงการนี้มีการใช้โปรแกรม CCS PIC ‘C’ คอมไพเลอร์ มาควบคุมไมโครคอนโทรลเลอร์ PIC16F877 เพื่อควบคุมกระบวนการชั่งน้ำหนัก จึงต้องใช้เวลาศึกษาโปรแกรมนาน ในการแสดงค่าและรับค่ายังมีปัญหาไม่สามารถรับค่าได้พร้อมกับแสดงค่าอย่างต่อเนื่อง ตรวจสอบพบว่ามีปัญหาในเรื่องของวงจรและโปรแกรม จึงทำการแก้ไขวงจรและโปรแกรมใหม่ เมื่อนำวงจรที่ได้ออกแบบมาต่อใช้งานจริงเมื่อทดลองแล้วพบว่ามีสัญญาณรบกวน แก้ปัญหาด้วยการนำค่าความต้านทาน และคาปาซิเตอร์มาต่อต่อแหล่งจ่าย ในส่วนของ Display แสดงผลก็มีตัวอักษรประหลาดเกิดขึ้น เมื่อตรวจสอบพบว่าโปรแกรมมีปัญหา ทำการแก้ค่าฟังก์ชันในโปรแกรมใหม่

ในส่วนของมอเตอร์ขับเคลื่อนรางสไลด์กินกระแสมาก แก้ด้วยการแยกแหล่งจ่ายออกจากชุดขับมอเตอร์ในการหมุนเกลิยว และใช้อุปกรณ์มาช่วยในการแยกแหล่งจ่าย และรางสไลด์ทำงานไม่เรียบเกิดการชะงัก เมื่อใช้รีเลย์ขับ เมื่อเปลี่ยนมาใช้ IC L298 ปรากฏว่ารางสไลด์ไม่ทำงานเลย ท่อพลาสติกและเกลิยวหัก อาจเป็นเพราะไม่ได้ถูกใช้งานมาเป็นเวลานาน ทำให้เกิดปัญหาการสึกหรอและเสื่อมสภาพของอุปกรณ์ แก้ไขด้วยการเปลี่ยนเกลิยวและท่อพลาสติกใหม่

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

กระบวนการชั่งน้ำหนักปัจจุบันมีการใช้อย่างงานอย่างแพร่หลายภายในระบบอุตสาหกรรม ส่งผลให้มีการพัฒนาระบบควบคุมการชั่งน้ำหนักอย่างต่อเนื่องโดยการเพิ่มขีดความสามารถในการใช้งานจากแบบ Manual ให้เป็นแบบ Automatic นอกจากการชั่งตวงน้ำหนักจะต้องมีความแม่นยำ

และเที่ยงตรงแล้ว ยังต้องมีฟังก์ชันการใช้งานที่หลากหลาย เช่น สามารถเลือกวัตถุบิดได้หลากหลาย น้ำหนักที่สามารถชั่งได้มากกว่า 10 kg หรืออาจใช้วิธีอื่นที่ไม่ใช่ไมโครคอนโทรลเลอร์ในการควบคุม น้ำหนัก หรือเลือกใช้วัสดุที่มีความเหมาะสมกับวัตถุบิดในการควบคุมการไหลของวัตถุบิดแทน เกลียวก็ได้

เนื่องจากค่าที่ได้จากเครื่องชั่งน้ำหนักควบคุมด้วยไมโครคอนโทรลเลอร์ นั้นยังมีค่าคลาดเคลื่อนไปอยู่มาก เนื่องจากขณะที่ทำการชั่งน้ำหนักจนเมื่อได้ค่าน้ำหนักที่ต้องการแล้วมอเตอร์ไม่สามารถที่จะหยุดการไหลของวัตถุบิดได้ทันที ดังนั้นจึงต้องปรับให้ความเร็วของมอเตอร์ทำการเปิดช้าๆ เพื่อป้องกันไม่ให้วัตถุบิดไหลลงมาเร็วจนเกินไป ทำให้การชั่งน้ำหนักคลาดเคลื่อนไปจากที่กำหนดไว้ และทำการเพิ่มความเร็วของมอเตอร์ในขณะที่ปิดให้มีความไวมากขึ้น เพื่อที่จะหยุดการไหลของวัตถุบิดได้ทันที และค่าที่ได้จากโพลดเซลล์จะมีค่าต่ำมากเป็นมิลลิโวลต์ ต้องทำการขยายขนาดของสัญญาณก่อน เพื่อให้ได้ค่าแรงดันที่มากขึ้นเป็นหน่วยโวลต์เพื่อให้สามารถส่งให้ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัลสามารถอ่านค่าได้ถูกต้อง โดยทำการขยายด้วยวงจรขยายสัญญาณอินสตรูเมนต์แอมป์รีไฟเออร์ก่อน หากวงจรอินสตรูเมนต์แอมป์รีไฟเออร์ ขยายสัญญาณไม่ค่อยมีความเสถียรภาพหรือทำให้เกิดสัญญาณรบกวนแทรกเข้าไป จะทำให้ค่าที่ออกมาจากวงจรอินสตรูเมนต์แอมป์รีไฟเออร์มีค่าผิดพลาดไปจากเดิม ดังนั้นควรใช้วงจรขยายสัญญาณอินสตรูเมนต์แอมป์รีไฟเออร์ที่มีความเสถียรภาพและอัตราขยายที่คงที่เพื่อให้แรงดันจากเอาต์พุตมีความแม่นยำ

เอกสารอ้างอิง

- [1] ชัยวัฒน์ ลิ้มพรจิตรวิไล , ัญญฐพล วงศ์สุนทรชัย , “**ปฏิบัติการไมโครคอนโทรลเลอร์ PIC16F877**” , อินโนเวทีฟ เอ็กเพอริเมนต์ , ISBN 974-90511-9-X , 324 หน้า , กรุงเทพฯ ฯ , 2521.
- [2] คอนสัน ปงผาบ , “**ไมโครคอนโทรลเลอร์ PIC และการประยุกต์ใช้งาน**” , สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) , ISBN 978-974-443-265-0 , 288 หน้า , กรุงเทพฯ ฯ , 2550.
- [3] ประจัน พลังสันติกุล , “**เรียนรู้และใช้งาน CCS C คอมไพเลอร์เขียนโปรแกรมภาษา C ความคุมไมโครคอนโทรลเลอร์ PIC**” , อินโนเวทีฟ เอ็กเพอริเมนต์ , ISBN 974-92022-9-5 , 312 หน้า , กรุงเทพฯ ฯ , 2521
- [4] ประจัน พลังสันติกุล , “**Example and C source code**” , แอพซอพท์แตก , ISBN 974-94645-7-5 , 303 หน้า , กรุงเทพฯ ฯ , 2537.
- [5] วรพงศ์ ตั้งศรีรัตน์ , “**เซ็นเซอร์และทรานสดิวเซอร์**” , สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) , ISBN 978-974-443-103-5 , 292 หน้า , กรุงเทพฯ ฯ , 2548.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงาน

```

#include <16F877.h>
#define adc=10
#define fuses HS,NOWDT,NOPROTECT,NOLVP
#define use delay(clock=18432000)
#define use rs232 (baud=9600,xmit=PIN_C6,rcv=PIN_C7,stream = pc_com)
#include <lcd.c>
#define use_portb_kbd
#define motor_1 pin_a1
#define motor_2 pin_a2
#define motor_3 pin_a3
#define motor_sl1 pin_d3
#define motor_sl2 pin_a5
int load[3][4]={{0,0,0,0},{0,0,0,0},{0,0,0,0}};
int16 adc;
int16 ad1=0;
int16 ad2=0;
float tank[3]=0,0,0 ;
int WEIGHT[4]=0,0,0,0;
int dis_1=1;
int dis_2=1;
char key=0;
char o_key=0;
int k=0;
int ad=0;
int loopDisplay3_1=0;
int loopDisplay3_2=0;
int loopDisplay3_3=0;
int loopDisplay3_4=0;
void motor_stop()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

output_bit(motor_1,0);
output_bit(motor_2,0);
output_bit(motor_3,0);
}
void motor_sl_stop()
{
output_bit(motor_sl1,0);
output_bit(motor_sl2,0);
}
void motor_sl_f()
{
output_bit(motor_sl1,1);
output_bit(motor_sl2,0);
}
void motor_sl_r()
{
output_bit(motor_sl1,0);
output_bit(motor_sl2,1);
}
void key_press()
{
adc=read_adc();
if(ad!=20)
{
ad1=ad1+adc;
ad=ad+1;
}
else if(ad==20)
{
ad1=ad1/20;
ad2=ad1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WEIGHT[0]=(ad1*0.0097751710654936461388074291300098)/10;
WEIGHT[1]=((ad1*0.0097751710654936461388074291300098)-
(WEIGHT[0]*10))/1;
WEIGHT[2]=(((ad1*0.0097751710654936461388074291300098)-
(WEIGHT[0]*10))-(WEIGHT[1]*1))/0.1;
WEIGHT[3]=((((ad1*0.0097751710654936461388074291300098)-
(WEIGHT[0]*10))-(WEIGHT[1]*1))-(WEIGHT[2]*0.1))/0.01;
ad=0;
ad1=0;
}
output_c(0x30);
output_low(pin_b0);
output_low(pin_b1);
output_low(pin_b2);
output_low(pin_b3);
if(input(pin_b4)&&input(pin_b5)&&input(pin_b6)&&input(pin_b7))
{
o_key=0;
}
output_c(0x30+WEIGHT[0]);
output_low(pin_b0);
output_high(pin_b1);
output_high(pin_b2);
output_high(pin_b3);
if(input(pin_b4)==0)
{
key='1';
k=1;
}
if(input(pin_b5)==0)
{
key='2';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    k=1;
}
if(input(pin_b6)==0)
{
    key='3';
    k=1;
}
if(input(pin_b7)==0)
{
    key='+';
    k=1;
}
delay_ms(5);
output_high(pin_b0);
output_c(0x30+WEIGHT[1]);
output_high(pin_b0);
output_low(pin_b1);
output_high(pin_b2);
output_high(pin_b3);
if(input(pin_b4)==0)
{
    key='4';
    k=1;
}
if(input(pin_b5)==0)
{
    key='5';
    k=1;
}
if(input(pin_b6)==0)
{
    key='6';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

k=1;
}
if(input(pin_b7)==0)
{
key='-';
k=1;
}
delay_ms(5);
output_high(pin_b1);
output_c(0x30+WEIGHT[2]);
output_high(pin_b0);
output_high(pin_b1);
output_low(pin_b2);
output_high(pin_b3);
if(input(pin_b4)==0)
{
key='7';
k=1;
}
if(input(pin_b5)==0)
{
key='8';
k=1;
}
if(input(pin_b6)==0)
{
key='9';
k=1;
}
if(input(pin_b7)==0)
{
key='f';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    k=1;
}
delay_ms(5);
output_high(pin_b2);
output_c(0x30+WEIGHT[3]);
output_high(pin_b0);
output_high(pin_b1);
output_high(pin_b2);
output_low(pin_b3);
if(input(pin_b4)==0)
{
    key='c';
    k=1;
}
if(input(pin_b5)==0)
{
    key='0';
    k=1;
}
if(input(pin_b6)==0)
{
    delay_ms(600);
}
if(input(pin_b7)==0)
{
    key='e';
    k=1;
}
delay_ms(5);
output_high(pin_b3);

```

```
output_c(0x30);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

output_high(pin_b0);
output_high(pin_b1);
output_high(pin_b2);
output_high(pin_b3);
if(key==o_key)
{
k=0;
}
}
void display2_1_1()
{
lcd_putc("\f TANK ?\n>1< 2 3 ");
}
void display2_1_2()
{
lcd_putc("\f TANK ?\n 1 >2< 3 ");
}
void display2_1_3()
{
lcd_putc("\f TANK ?\n 1 2 >3<");
}
void display3_1()
{
if(loopdisplay3_1<5)
{
printf(lcd_putc,"\f WEIGHT %d\n %d%d.%d%d Kg",dis_2,load[dis_2-1][0],load[dis_2-1][1],load[dis_2-1][2],load[dis_2-1][3] );
}
if(loopdisplay3_1>5)
{
printf(lcd_putc,"\f WEIGHT %d\n _%d.%d%d Kg",dis_2,load[dis_2-1][1],load[dis_2-1][2],load[dis_2-1][3] );
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(loopdisplay3_1==10)
{
loopdisplay3_1=0;
}
loopdisplay3_1++;
}
void display3_2()
{
if(loopdisplay3_2<5)
{
printf(lcd_putc,"\f WEIGHT %d\n %d%d.%d%d Kg",dis_2,load[dis_2-1][0],load[dis_2-1][1],load[dis_2-1][2],load[dis_2-1][3] );
}
if(loopdisplay3_2>5)
{
printf(lcd_putc,"\f WEIGHT %d\n %d_.%d%d Kg",dis_2,load[dis_2-1][0],load[dis_2-1][2],load[dis_2-1][3] );
}
if(loopdisplay3_2==10)
{
loopdisplay3_2=0;
}
loopdisplay3_2++;
}
void display3_3()
{
if(loopdisplay3_3<5)
{
printf(lcd_putc,"\f WEIGHT %d\n %d%d.%d%d Kg",dis_2,load[dis_2-1][0],load[dis_2-1][1],load[dis_2-1][2],load[dis_2-1][3] );
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(loopdisplay3_3>5)
{
printf lcd_putc, "\f WEIGHT %d\n  %d%d. %d Kg", dis_2, load[dis_2-1][0], load[dis_2-
1][1], load[dis_2-1][3] );
}
if(loopdisplay3_3==10)
{
loopdisplay3_3=0;
}
loopdisplay3_3++;
}
void display3_4()
{
if(loopdisplay3_4<5)
{
printf lcd_putc, "\f WEIGHT %d\n  %d%d.%d%d Kg", dis_2, load[dis_2-1][0], load[dis_2-
1][1], load[dis_2-1][2], load[dis_2-1][3] );
}
if(loopdisplay3_4>5)
{
printf lcd_putc, "\f WEIGHT %d\n  %d%d.%d Kg", dis_2, load[dis_2-1][0], load[dis_2-
1][1], load[dis_2-1][2] );
}
if(loopdisplay3_4==10)
{
loopdisplay3_4=0;
}
loopdisplay3_4++;
}
void display3_5()
{

```

```

printf(lcd_putc,"f WEIGHT %d\n %d%d.%d%d Kg",dis_2,load[dis_2-1][0],load[dis_2-1][1],load[dis_2-1][2],load[dis_2-1][3] );
}
void display4_1()
{
printf(lcd_putc," Tank 1 \n %d%d.%d%d Kg
",WEIGHT[0],WEIGHT[1],WEIGHT[2],WEIGHT[3]);
}
void display4_2()
{
printf(lcd_putc," Tank 2 \n %d%d.%d%d Kg
",WEIGHT[0],WEIGHT[1],WEIGHT[2],WEIGHT[3]);
}
void display4_3()
{
printf(lcd_putc," Tank 3 \n %d%d.%d%d Kg
",WEIGHT[0],WEIGHT[1],WEIGHT[2],WEIGHT[3]);
}
void display4_4()
{
printf(lcd_putc," TOTAL \n %d%d.%d%d Kg
",WEIGHT[0],WEIGHT[1],WEIGHT[2],WEIGHT[3]);
}
void main()
{
int type=0;
int key_con=0;
int en_k=0;
int sl=0;

port_b_pullups(true);
set_tris_a(0x01);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set_tris_d(0x00);
set_tris_c(0x00);
set_tris_e(0xff);
setup_adc_ports(RA0_ANALOG);
setup_adc(ADC_CLOCK_INTERNAL);
set_adc_channel(0);
lcd_init();
motor_stop();
motor_sl_stop();
while(input(pin_e0)&&input(pin_e1)&&input(pin_e2))
{
motor_sl_r();
key_press();
}
motor_sl_stop();
if(input(pin_e0)==0){sl=1;}
if(input(pin_e1)==0){sl=2;}
if(input(pin_e2)==0){sl=3;}

dis2_1:  type=0;
if(dis_2==1)
{
display2_1_1();
}
if(dis_2==2)
{
display2_1_2();
}
if(dis_2==3)
{
display2_1_3();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(type==0)
{
    key_press();
    if (k==1)
    {
        if(key=='+')
        {
            if(dis_2!=1)
            {dis_2=dis_2-1;o_key=key;key=0;}
            if(dis_2==1)
            {o_key=key;key=0;}
        }
        if(key=='-')
        {
            if(dis_2!=3)
            {dis_2=dis_2+1;o_key=key;key=0;}
            if(dis_2==3)
            {o_key=key;key=0;}
        }
        if(dis_2==1)
        {
            display2_1_1();
        }
        if(dis_2==2)
        {
            display2_1_2();
        }
        if(dis_2==3)
        {
            display2_1_3();
        }
        if(key=='e')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  type=1;
  o_key='e';key=0;
  key_con=0;
  goto dis3_1;
}
}
}
mor_sl:  if(sl==dis_2)
{
  motor_sl_stop();
  if(input(pin_e0)==0){sl=1;}
  if(input(pin_e1)==0){sl=2;}
  if(input(pin_e2)==0){sl=3;}
}
while(sl<dis_2)
{
  motor_sl_f();
  key_press();
  if(input(pin_e0)==0){sl=1;}
  if(input(pin_e1)==0){sl=2;}
  if(input(pin_e2)==0){sl=3;}
}
  motor_sl_stop();
  key=0;
  o_key='e';
while(sl>dis_2)
{
  motor_sl_r();
  key_press();
  if(input(pin_e0)==0){sl=1;}
  if(input(pin_e1)==0){sl=2;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(input(pin_e2)==0){sl=3;}
}
motor_sl_stop();
key=0;
o_key='e';
goto dis_end_1;

```

```

dis3_1:  if(key_con==0)
{
display3_1();
}
if(key_con==1)
{
display3_2();
}
if(key_con==2)
{
display3_3();
}
if(key_con==3)
{
display3_4();
}
if(key_con==4)
{
display3_5();
}
key_press();

if(k==1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
if((key_con<4)&&((key=='0')||(key=='1')||(key=='2')||(key=='3')||(key=='4')||(key=='5')||(key=='6')
|(key=='7')|(key=='8')|(key=='9'))
{
if(key=='0'){en_k=0; o_key=key;key=0;}
if(key=='1'){en_k=1; o_key=key;key=0;}
if(key=='2'){en_k=2; o_key=key;key=0;}
if(key=='3'){en_k=3; o_key=key;key=0;}
if(key=='4'){en_k=4; o_key=key;key=0;}
if(key=='5'){en_k=5; o_key=key;key=0;}
if(key=='6'){en_k=6; o_key=key;key=0;}
if(key=='7'){en_k=7; o_key=key;key=0;}
if(key=='8'){en_k=8; o_key=key;key=0;}
if(key=='9'){en_k=9; o_key=key;key=0;}
load[dis_2-1][key_con]=en_k;
key_con=key_con+1;
if(key_con==4)
{o_key=key;key=0;}
}
if(key=='c'){key_con=0; load[dis_2-1][0]=0; load[dis_2-1][1]=0; load[dis_2-1][2]=0;
load[dis_2-1][3]=0; o_key=key;key=0; goto dis3_1;}
if(key=='e'){o_key='e';key=0; goto mor_sl;}
}
goto dis3_1;

dis_end_1: tank[dis_2-1]=((load[dis_2-1][0]*10)+(load[dis_2-1][1]*1)+(load[dis_2-
1][2]*0.1)+(load[dis_2-1][3]*0.01))*102.3;

adc=read_adc();

if(ad!=20)
{
ad1=ad1+adc;
ad=ad+1;
}

```

```

else if(ad==20)
{
    ad2=ad1/20;
    ad=0;
    ad1=0;
}
printf(lcd_putc,"\f");
while(ad2<tank[dis_2-1])
{
    key_press();
    o_key=key;key=0;
    if(dis_2==1)
    {
        output_bit(motor_1,1);
        display4_10();
    }
    if(dis_2==2)
    {
        output_bit(motor_2,1);
        display4_20();
    }
    if(dis_2==3)
    {
        output_bit(motor_3,1);
        display4_30();
    }
}
    motor_stop();
    type=0;
    printf("\n\rTANK:%d WEIGHT>> %d%d.%d%d
kg",dis_2,WEIGHT[0],WEIGHT[1],WEIGHT[2],WEIGHT[3]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(type==0)
{
    key_press();

    if(dis_2==1)
    {
        display4_1();
    }
    if(dis_2==2)
    {
        display4_2();
    }
    if(dis_2==3)
    {
        display4_3();
    }
    if(key=='f')
    {
        o_key=key;key=0;
        goto dis2_1;
    }
    else{o_key=key;key=0;}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้