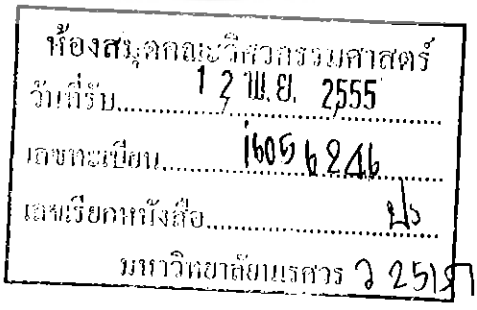
การประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง

APPLICATION OF DIGITAL FILTER FOR ESTIMATING MANGO WEIGHT ON BELT CONVEYOR

นายวรนารถ ศิริวัฒนานุรักษ์ รหัส 51383997

นางสาวสุนิศา เลื่อนชิด รหัส 51384079



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง
ผู้ดำเนินโครงงาน	นายวรนารถ ศิริวัฒนานุรักษ์ รหัส 51383997
	นางสาวสุนิศา เลื่อนชิด รหัส 51384079
อาจารย์ที่ปรึกษา	ผู้ช่วยศาตราจารย์ ดร.สุชาติ แย้มแม่น
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า

คณะกรรมการการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มแม่น)

.....กรรมการ

(ดร.มุฑิตา สงฆ์จันทร์)

.....กรรมการ

(ดร.พนัส นัถฤทธิ์)

หัวข้อโครงงาน	การประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง
ผู้ดำเนินโครงงาน	นายวรนารถ ศิริวัฒนานุรักษ์ รหัส 51383997
	นางสาวสุนิศา เลื่อนชิด รหัส 51384079
อาจารย์ที่ปรึกษา	ผู้ช่วยศาตราจารย์ ดร.สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

บทคัดย่อ

โครงงานนี้นำเสนอการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับการประมาณค่าน้ำหนักของผลมะม่วงบนระบบสายพานลำเลียง โดยกระบวนการจะทำการรับค่าสัญญาณทางไฟฟ้าจากโหลดเซลล์เข้าสู่วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล ด้วยไมโครคอนโทรลเลอร์ (dsPIC30F4011) และทำการกรองสัญญาณดิจิตอลเพื่อกำจัดสัญญาณรบกวนด้วยตัวกรองค่าเฉลี่ยแบบเคลื่อนที่ จากนั้นทำการแปลงสัญญาณดิจิตอลกลับด้วยวงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก เพื่อแสดงผลลัพธ์ของตัวกรอง จากการทดลอง พบว่า ความล่าช้าในการประมวลผลข้อมูลที่จำนวนข้อมูล 50, 100, 150 และ 200 ข้อมูล มีความล่าช้า 25, 50, 75 และ 100 มิลลิวินาที ตามลำดับ และจำนวนข้อมูลในการกรองมากขึ้นการแสดงผลช้าลง อย่างไรก็ตามเมื่อพิจารณากราฟสัญญาณที่ผ่านตัวกรองจากการเลือกใช้จำนวนข้อมูลในการกรองค่าต่างๆ พบว่า จำนวนข้อมูลการกรองมากขึ้นทำให้สัญญาณที่ได้มีความราบเรียบขึ้นและมีประสิทธิภาพในการลดสัญญาณรบกวนได้ดี

Project Title	Application of Digital Filter for Estimating Mango Weight on Belt Conveyor
Name	Mr. Voranart Siriwattananulak ID 51383997
	Miss Sunisa Lueanchit ID 51384079
Project Advisor	Assistant Professor Suchart Yammen, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic	2012

Abstract

This project presents the application of digital filters for estimating the weight of mangoes on a conveyor system. The microcontroller (DsPIC30F4011) receives the electrical signal from the load cell circuit in order to convert the analog signal into a digital signal. Noise of the received signal is also eliminated with the moving average filter. The denosing digital signal is then converted into the analog signal shown in the oscilloscope. From the experimental results, it was found that the output signal of the filter is delayed 25, 50, 75 and 100 ms in processing the data of 50, 100, 150 and 200 points, respectively. Also, the more number of the filter is, the less noise in the signal is.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยดี ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานและให้ความกรุณาในการตรวจทาน ปริญญานิพนธ์ คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบพระคุณ ดร.พนัส นัฉฤทธิ์ และ ดร.มุฑิตา สงฆ์จันทร์ ซึ่งเป็นคณะกรรมการในการสอบโครงงานที่ให้คำแนะนำ ชี้แนะแนวทาง และข้อคิดเห็นต่างๆที่เป็นประโยชน์ในโครงงานนี้ทำให้โครงงานออกมาสมบูรณ์แบบยิ่งขึ้น

ขอขอบพระคุณ ดร.นิพัทธ์ จันทรมินทร์ ซึ่งเป็นอาจารย์ผู้รับผิดชอบรายวิชาโครงงานวิศวกรรมไฟฟ้า ที่ให้คำแนะนำ คำปรึกษาในการพิมพ์รูปเล่มปริญญานิพนธ์ รวมถึงแก้ไขปรับปรุงให้รูปเล่มปริญญานิพนธ์ให้ถูกต้องตามหลักการพิมพ์และอื่นๆที่ทำให้ปริญญานิพนธ์มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณ คุณประทีป สังข์แป้น ที่กรุณาช่วยให้คำแนะนำเกี่ยวกับการออกแบบวงจรอิเล็กทรอนิกส์และการเขียนโปรแกรมไมโครคอนโทรลเลอร์ที่เป็นประโยชน์ต่อโครงงานนี้รวมถึงความรู้ใหม่ๆ

ขอขอบคุณคุณชลีดล อินยาศรี ที่ให้ความรู้เกี่ยวกับการประมวลผลสัญญาณดิจิตอลและอื่นๆ ที่เป็นประโยชน์ต่อการทำโครงงานนี้

นอกจากนี้ยังต้องขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ยืมอุปกรณ์และใช้ห้องปฏิบัติการ จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

เหนือสิ่งอื่นใด คณะผู้จัดทำโครงงานขอกราบขอบพระคุณคุณพ่อ คุณแม่ ผู้มอบความรักความเมตตา สติปัญญา เป็นที่ปรึกษาปัญหาในทุกๆ เรื่อง รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้และขอขอบคุณทุกๆ คนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นายวรนารถ ศิริวัฒนานุรักษ์ รหัส 51383997
นางสาวสุนิศา เลื่อนชิด รหัส 51384079

สารบัญ

หน้า

ใบรับรองโครงงานวิศวกรรม ..... ก

บทคัดย่อ ..... ข

กิตติกรรมประกาศ ..... ง

สารบัญ ..... จ

สารบัญรูป ..... ญ

สารบัญตาราง ..... ฐ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของโครงงาน 1

1.4 วิธีดำเนินการ 2

1.5 แผนการดำเนินงานโครงงาน 2

1.6 ผลที่คาดว่าจะได้รับ 3

1.7 งบประมาณของโครงงาน 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 4

2.1 หลักการพื้นฐานของเซนเซอร์ 4

2.1.1 โหลดเซลล์ 4

2.2 การออกแบบวงจรขยายสัญญาณและวงจรกรองความถี่ 7

2.2.1 ความรู้พื้นฐานเกี่ยวกับออปแอมป์ 7

2.2.2 ลักษณะของขาต่างๆของออปแอมป์ 8

สารบัญ (ต่อ)

หน้า

2.2.3 เทคนิคการเลือกใช้ออปแอมป์ 9

2.2.4 วงจรขยายอินสตรูเมนท์ 9

2.2.5 วงจรกรองสัญญาณ 10

2.2.6 วงจรผ่านต่ำบัตเตอร์เวิร์ท 11

2.2.7 วงจรลบสัญญาณ 12

2.2.8 สัญญาณรบกวน 13

2.2.9 เทคนิคการลดสัญญาณรบกวน 14

2.3 ประมวลผลสัญญาณดิจิตอล 15

2.3.1 การสุ่มตัวอย่าง 17

2.3.2 การออกแบบตัวกรองดิจิตอล 19

2.4 ไมโครคอนโทรลเลอร์ 22

2.4.1 คุณสมบัติ MPLAB C Compiler 22

2.4.2 การใช้งานพอร์ตอินพุตเอาท์พุตดิจิตอล 23

2.4.3 การทดลองใช้งานโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอลภายใน dsPIC เบื้องต้น .25

2.4.4 การใช้งานฟังก์ชันไลบรารี *DSP* 37

2.5 คุณสมบัติเด่นโดยรวมของ dsPIC 38

2.5.1 คุณสมบัติของซีพียู 38

2.5.2 คุณสมบัติด้านการประมวลสัญญาณดิจิตอล 39

2.5.3 คุณสมบัติของโมดูลฟังก์ชั่นพิเศษ 39

บทที่ 3 วิธีดำเนินโครงงาน 40

3.1 ศึกษาทฤษฎีและหลักการใช้งานของโหลดเซลล์ 40

สารบัญ (ต่อ)

หน้า

3.2 ศึกษาการออกแบบวงจรอิเล็กทรอนิกส์ 41
3.3 ศึกษาทฤษฎีประมวลผลสัญญาณดิจิตอล 45
3.4 ศึกษาไมโครคอนโทรลเลอร์ dsPIC30F4011 และการเขียนโปรแกรมภาษาซีเพื่อใช้ 45
ในการประมวลผลสัญญาณดิจิตอล 45
3.5 ออกแบบและทำการสร้างเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน 45
3.6 ออกแบบวงจรอิเล็กทรอนิกส์ 46
3.6.1 วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ 46
3.6.2 วงจรปรับปรุงสัญญาณ 49
3.6.3 วงจรปรับความเร็วมอเตอร์ 51
3.6.4 วงจรไมโครคอนโทรเลอร์ 53
3.7 ออกแบบวงจรกรองดิจิตอลและเขียนโปรแกรมประมวลผลสัญญาณ 55
3.7.1 ออกแบบวงจรกรองดิจิตอล 55
3.7.2 เขียนโปรแกรมประมวลผลสัญญาดิจิตอล 56
3.8 ทำการทดลองวงจรกรองดิจิตอลที่ได้ออกแบบไว้ในไมโครคอนโครลเลอร์โดยใช้สัญญาณ
จากเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน 61
3.9 ปรับปรุงและแก้ไขระบบ 64
3.10 จัดทำรูปเล่มรายงาน 64
บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง 65
4.1 การทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล 66
4.1.1 สัญญาณคลื่นไซน์ ความถี่ 50 เฮิรตซ์ 66
4.1.2 สัญญาณสี่เหลี่ยมความถี่ 50 เฮิรตซ์ 66

สารบัญ (ต่อ)

หน้า

4.1.3 สัญญาณสามเหลี่ยม ความถี่ 50 เฮิรตซ์ 67

4.1.4 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 0 เมตรต่อนาที 67

4.1.5 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 5 เมตรต่อนาที 68

4.1.6 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 10 เมตรต่อนาที 68

4.1.7 วิเคราะห์ผลการทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล 69

4.2 การทดลองวงจรกรองสัญญาณดิจิตอลโดยการใช้ตัวกรองเฉลี่ยแบบเคลื่อนที่ 69

4.2.1 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 50 ข้อมูล 69

4.2.2 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 100 ข้อมูล 71

4.2.1 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 150 ข้อมูล 72

4.2.1 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 200 ข้อมูล 74

4.2.5 วิเคราะห์ผลการทดลองวงจรกรองสัญญาณแบบดิจิตอล โดยวิธีการใช้ตัวกรองเฉลี่ยแบบเคลื่อนที่ 75

4.3 สรุปผลการทดลอง 76

บทที่ 5 สรุปผลและข้อเสนอแนะ 77

5.1 สรุปผลการดำเนินโครงงาน 77

5.2 ปัญหาและข้อเสนอแนะ 78

5.3 แนวทางในการพัฒนาโครงงาน 78

เอกสารอ้างอิง 79

ภาคผนวก 80

ภาคผนวก ก วิธีการติดตั้งโปรแกรม 81

สารบัญ (ต่อ)

หน้า

ภาคผนวก ข ขั้นตอนการสร้างโปรเจ็กต์ใน MPLAB C Compiler และการสร้างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่ 99

ภาคผนวก ค วิธีการทดลองโครงงานการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง 119

ภาคผนวก ง วิธีการทำ PCB โดยใช้ Dry film 125

ประวัติผู้เขียนโครงงาน 131

สารบัญรูป

หน้า

รูปที่ 2.1 การนำสเตรนเกจมาต่อเข้ากับวงจรดีเฟลคชั่นบริดจ์ 4
รูปที่ 2.2 โหลดเซลล์แบบลิงค์ 5
รูปที่ 2.3 โหลดเซลล์แบบคานประกอบด้วยชิ้นส่วนยืดหยุ่นกับสเตรนเกจ 5
รูปที่ 2.4 โหลดเซลล์แบบวงแหวน 6
รูปที่ 2.5 โหลดเซลล์ชนิดเชียร์-เวบ 6
รูปที่ 2.6 โครงสร้างพื้นฐานของออปแอมป์ 7
รูปที่ 2.7 ลักษณะของขาที่ใช้งานของออปแอมป์ 8
รูปที่ 2.8 วงจรรวม (Integrated Circuit: IC) สำเร็จรูปเบอร์ INA114 9
รูปที่ 2.9 วงจรขยายสัญญาณอินสตรูเมนท์ 9
รูปที่ 2.10 การกำหนดคุณลักษณะของวงจรกรองผ่านต่ำ 11
รูปที่ 2.11 ฟังก์ชันผลตอบสนองเชิงแอมพลิจูดของวงจรกรองผ่านต่ำบัตเตอร์เวิร์ท 12
รูปที่ 2.12 วงจรลบสัญญาณ 12
รูปที่ 2.13 แสดงกลไกการพัฒนาของสัญญาณรบกวนที่อาจเกิดขึ้นในวงจร 13
รูปที่ 2.14 โครงสร้างระบบประมวลสัญญาณแอนะล็อก 15
รูปที่ 2.15โครงสร้างระบบประมวลสัญญาณดิจิตอล 16
รูปที่ 2.16 การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลและลักษณะสัญญาณที่เกี่ยวข้อง 17
รูปที่ 2.17 สัญญาณอินพุตที่มีความยาวจำกัด $x[n]$ 19
รูปที่ 2.18 เอาท์พุตของระบบเฉลี่ยค่า $y[n]$ 20
รูปที่ 2.19 การคำนวณการเฉลี่ยค่าของฟิลเตอร์ที่เวลา n ใช้ค่าใน Sliding Window แถบสีเทาเข้ม 21
รูปที่ 2.20 ไดอะแกรมการทำงานอย่างง่ายของโมดูล ADC ในไมโครคอนโทรลเลอร์ 26
รูปที่ 2.21 รูปแบบของข้อมูลผลลัพธ์ที่ได้จากการแปลงสัญญาณของโมดูล ADC 27
รูปที่ 2.22 บล็อกไดอะแกรม ระบบการประมวลผลสัญญาณดิจิตอล 37
รูปที่ 3.1 แผนภาพขั้นตอนดำเนินงาน 40
รูปที่ 3.2 แสดงลักษณะของโหลดเซลล์ 41
รูปที่ 3.3 TL074CN 41
รูปที่ 3.4 ก. วงจรขยายสัญญาณอินสตรูเมนท์ (Instrument Amplifier Circuit) และ ข. วงจรลบ 42
รูปที่ 3.5 แสดงเครื่องชั่งนำหนักที่มีสายพานลำเลียงวิ่งอยู่ด้านบนที่สร้างเสร็จแล้ว 45
รูปที่ 3.6 วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ 46

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.7 ก. ลายวงจรพิมพ์วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ ข. ลักษณะการ 48
รูปที่ 3.8 วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ ที่สามารถใช้งานได้.....49
รูปที่ 3.9 วงจรปรับปรุงสัญญาณ.....49
รูปที่ 3.10 ก. ลายวงจรพิมพ์ของวงจรปรับปรุงสัญญาณ ข. ลักษณะการว่างอุปกรณ์ของ 50
รูปที่ 3.11 วงจรปรับปรุงสัญญาณที่อยู่บนแผ่นวงจรพิมพ์.....51
รูปที่ 3.12 วงจรปรับความเร็วมอเตอร์.....51
รูปที่ 3.13 ก. ลายวงจรพิมพ์วงจรปรับความเร็วมอเตอร์ ข. ลักษณะการว่างอุปกรณ์วงจร.....52
รูปที่ 3.14 วงจรปรับความเร็วมอเตอร์ที่สามารถใช้งานได้.....53
รูปที่ 3.15 วงจรไมโครคอนโทรเลอร์.....53
รูปที่ 3.16 ก. ลายวงจรพิมพ์วงจรวงจรไมโครคอนโทรเลอร์ ข. ลักษณะการว่างอุปกรณ์วงจร.....54
รูปที่ 3.17 วงจรไมโครคอนโทรลเลอร์.....55
รูปที่ 3.18 กระบวนการประมวลผลสัญญาณดิจิตอล.....56
รูปที่ 3.19 แสดงตำแหน่งการเลือกขาอินพุตแอนะล็อก.....56
รูปที่ 3.20 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก.....60
รูปที่ 3.21 (1) แสดงวงจรอิเล็กทรอนิกส์ต่างๆ ที่ได้ออกแบบไว้แล้ว.....62
รูปที่ 3.22 (2) แสดงวงจรอิเล็กทรอนิกส์ต่างๆ ที่ได้ออกแบบไว้แล้ว.....62
รูปที่ 3.23 ส่วนประกอบต่างๆ ที่ติดตั้งเสร็จ.....63
รูปที่ 3.24 แสดงอุปกรณ์ที่ใช้ทดลองในโครงงาน.....63
รูปที่ 4.1 สัญญาณอินพุตและเอาท์พุตของสัญญาณคลื่นไซน์ที่ความถี่ 50 เฮิรตซ์.....66
รูปที่ 4.2 สัญญาณอินพุตและเอาท์พุตของสัญญาณสี่เหลี่ยมที่ความถี่ 50 เฮิรตซ์.....66
รูปที่ 4.3 สัญญาณอินพุตและเอาท์พุตของสัญญาณสามเหลี่ยมที่ความถี่ 50 เฮิรตซ์.....67
รูปที่ 4.4 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 0 เมตร ต่อนาที.....67
รูปที่ 4.5 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 5 เมตร ต่อนาที.....68
รูปที่ 4.6 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 10 เมตร ต่อนาที.....68

สารบัญรูป (ต่อ)

หน้า

รูปที่ 4.7 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตร 69

รูปที่ 4.8 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตร 70

รูปที่ 4.9 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตร 70

รูปที่ 4.10 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตร 71

รูปที่ 4.11 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตร 71

รูปที่ 4.12 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 72

รูปที่ 4.13 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตร 72

รูปที่ 4.14 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตร 73

รูปที่ 4.15 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 73

รูปที่ 4.16 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตร 74

รูปที่ 4.17 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตร 74

รูปที่ 4.18 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 75

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงค่าผลลัพธ์ของ $y[n]$ 20
ตารางที่ 2.2 แสดงค่าผลลัพธ์ของ $y[n]$ 22
ตารางที่ 2.3 การกำหนดค่าให้กับรีจิสเตอร์ TRISB และการใช้งานรีจิสเตอร์ PORTB, LATB.....24
ตารางที่ 2.4 แสดงการกำหนดค่าเวลาที่ใช้ในการแปลงสัญญาณในโมดูล ADC 32

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ผลไม้นับเป็นพืชเศรษฐกิจที่สำคัญประเภทหนึ่งของประเทศไทย โดยผลไม้สามารถทำรายได้เข้าประเทศปีละหลายพันล้านบาท และผลไม้ไทยยังเป็นที่นิยมบริโภคกันทั่วไปทั้งในประเทศและต่างประเทศ นอกจากนี้ ความต้องการบริโภคผลไม้นับวันจะเพิ่มสูงขึ้น เนื่องมาจากสาเหตุหลัก คือ จำนวนประชากรที่เพิ่มมากขึ้นและความสนใจในสุขภาพก็มีมากขึ้นด้วย ในประเทศไทยได้มีการส่งออกผลไม้หลายชนิด ยกตัวอย่างเช่น มะม่วง มะม่วงเป็นผลไม้ไทยที่ทำรายได้ให้กับประเทศไทยปีละประมาณ 15 ล้านเหรียญสหรัฐ (สำนักบริการส่งออก)

ปัจจุบันการส่งออกมะม่วงอยู่ในรูปของมะม่วงสด มะม่วงกระป๋อง น้ำมะม่วง และมะม่วงแปรรูปอื่นๆ เช่น มะม่วงกวน อบแห้ง มะม่วงดอง เป็นต้น การส่งออกมะม่วงแต่ละครั้งจะต้องทำการคัดแยกมะม่วงออกเป็นขนาด ปริมาณและคุณภาพ ซึ่งการส่งออกมะม่วงสู่ต่างประเทศนั้นต้องผ่านกระบวนการคัดแยกมะม่วง เพื่อแบ่งออกเป็นขนาดต่างๆ ที่ต้องการ โดยการใช้น้ำหนักเป็นเกณฑ์วัด แต่เนื่องจากการน้ำหนักเป็นเกณฑ์วัดมีปัญญาเรื่องของความไม่ถูกต้อง มีความผิดพลาดจากสัญญาณรบกวน สัญญาณรบกวนนี้อาจเกิดจากตัวเครื่องคัดมะม่วง เช่น การสั่นสะเทือนของเครื่องที่เกิดจากการสั่นของมอเตอร์ขณะทำงาน หรือเกิดจากสัญญาณไฟบ้าน 50 เฮิรตซ์ ด้วยเหตุนี้จึงทำให้การประมาณค่าน้ำหนักมะม่วงเกิดความผิดพลาดไปจากความเป็นจริงมาก

จากปัญหาที่พบจึงได้ทำการศึกษาและประยุกต์ใช้วงจรกรองดิจิตอล เพื่อกรองสัญญาณรบกวนออกจากสัญญาณที่ต้องการ อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์

ออกแบบวงจรกรองดิจิตอลเพื่อกรองสัญญาณรบกวนออกจากเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน

1.3 ขอบเขตของโครงงาน

สร้างเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบนและออกแบบวงจรอิเล็กทรอนิกส์เพื่อให้ได้สัญญาณที่ต้องการและทำการวัดสัญญาณจากการประมวลผลสัญญาณดิจิตอลโดยการใช้ไมโครคอนโทรลเลอร์ dsPIC30F4011 ที่ใช้ภาษาซีในการเขียนโปรแกรม

1.4 วิธีดำเนินการ

1.4.1 ศึกษาทฤษฎีและหลักการใช้งานของโหลดเซลล์

1.4.2 ศึกษาการออกแบบวงจรอิเล็กทรอนิกส์

1.4.3 ศึกษาทฤษฎีประมวลผลสัญญาณดิจิตอล

1.4.4 ศึกษาไมโครคอนโทรลเลอร์ dsPIC30F4011 และหลักการเขียนโปรแกรมภาษาซีเพื่อใช้ในการเขียนโปรแกรมประมวลผลสัญญาณดิจิตอล

1.4.5 ออกแบบและทำการสร้างเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน

1.4.6 ออกแบบวงจรอิเล็กทรอนิกส์และวงจรไมโครคอนโทรลเลอร์

1.4.7 ออกแบบวงจรกรองดิจิตอลและเขียนโปรแกรมประมวลผลสัญญาณดิจิตอล

1.4.8 ทดลองวงจรกรองดิจิตอลที่ได้ออกแบบไว้ในไมโครคอนโทรลเลอร์โดยใช้สัญญาณจากเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน

1.4.9 ปรับปรุงและแก้ไข

1.4.10 จัดทำรูปเล่มรายงาน

1.5 แผนการดำเนินงานโครงงาน

กิจกรรม	ระยะเวลาดำเนินงาน							
	มิ.ย.	ก.ค	ส.ค.-ก.ย.	ต.ค.-พ.ย.	ธ.ค	ม.ค	ก.พ	มี.ค
1. ศึกษาทฤษฎีและหลักการใช้งานของโหลดเซลล์								
2. ศึกษาการออกแบบวงจรอิเล็กทรอนิกส์								
3. ศึกษาทฤษฎีประมวลผลสัญญาณดิจิตอล								
4. ศึกษาไมโครคอนโทรลเลอร์ dsPIC30F4011 และหลักการเขียนโปรแกรมภาษาซีเพื่อใช้ในการเขียนโปรแกรมประมวลผลสัญญาณดิจิตอล								
5. ออกแบบและทำการสร้างเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่								

ด้านบน							
6. ออกแบบวงจรอิเล็กทรอนิกส์และวงจรไมโครคอนโทรลเลอร์							
7. ออกแบบวงจรกรองดิจิตอลและเขียนโปรแกรมประมวลผลสัญญาดิจิตอล							
8. ทำการทดลองวงจรกรองดิจิตอลที่ได้ออกแบบไว้ในไมโครคอนโทรลเลอร์โดยใช้สัญญาณจากเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน							
9. ปรับปรุงและแก้ไขระบบ							
10. จัดทำรูปเล่มรายงาน							

1.6 ผลที่คาดว่าจะได้รับ

วงจรกรองดิจิตอลที่ออกแบบสามารถกรองสัญญาณรบกวนออกจากสัญญาณน้ำหนักของมะม่วงที่ออกมาจากการชั่งบนเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงอยู่ด้านบนได้

1.7 งบประมาณของโครงงาน

1.6.1 ค่าวัสดุอุปกรณ์	1,000 บาท
1.6.2 ค่าเอกสาร	500 บาท
1.6.3 อื่น ๆ	500 บาท
รวมเป็นเงินทั้งสิ้น	2,000 บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

การประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียงนั้นจำเป็นที่จะต้องมีความรู้พื้นฐานหลายส่วนมาประกอบกันเพื่อให้การประมวลผลมีประสิทธิภาพและมีความถูกต้องมากที่สุด จึงต้องมีความรู้พื้นฐานเกี่ยวกับหัวข้อต่างๆ ดังนี้

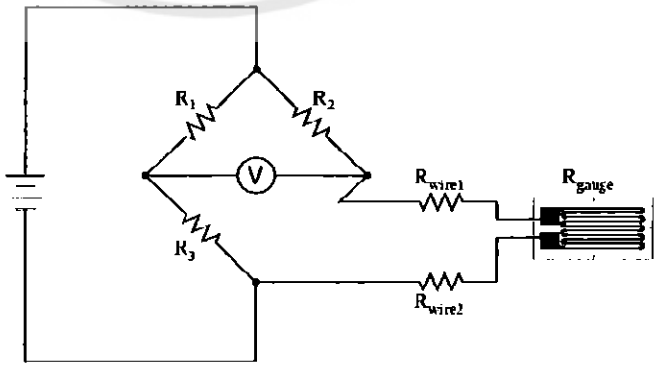
2.1 หลักการพื้นฐานของเซนเซอร์

ระบบเซนเซอร์ (Sensor System) หมายถึง ระบบที่รับค่าเปลี่ยนแปลงทางกล แล้วนำมาแปลงค่าเป็นค่าทางไฟฟ้า (กระแสและแรงดัน) เพื่อที่จะนำค่าทางไฟฟ้าไปใช้งานต่างๆ ต่อไป เช่น นำสัญญาณไฟฟ้าไปเป็นสัญญาณอินพุตของระบบควบคุมหรือแปลงค่าไฟฟ้าที่ได้เป็นสัญญาณตัวเลขตัวอย่างของระบบเซนเซอร์ เช่น ระบบโหลดเซลล์ (Load Cell) เป็นต้น

2.1.1 โหลดเซลล์

โหลดเซลล์เป็นระบบเซนเซอร์ที่แปลงค่าน้ำหนักทางกลของสิ่งของให้เป็นปริมาณทางไฟฟ้าซึ่งประกอบด้วยสเตรนเกจ (Strain Gage) เป็นส่วนตัวจับซึ่งจะคอยเปลี่ยนค่าความเครียดทางกลอันเนื่องจากน้ำหนักของวัตถุเป็นค่าความต้านทานไฟฟ้า

จากรูปที่ 2.1 เมื่อนำค่าความต้านทานที่ได้จากสเตรนเกจต่อเข้ากับวงจรดีเฟลคชั่นบริดจ์ (Deflection Bridge) ซึ่งต่อเข้ากับแหล่งจ่ายแรงดันกระแสตรงก็จะสามารถหาค่าเอาต์พุตของน้ำหนักวัตถุที่เป็นการเปลี่ยนแปลงของค่าแรงดันไฟฟ้าและนำสัญญาณไฟฟ้าที่ได้ไปประมวลผลเพื่อแสดงผลค่าน้ำหนักออกมาเป็นตัวเลขได้

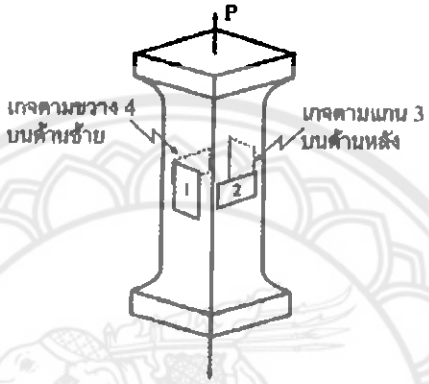


รูปที่ 2.1 การนำสเตรนเกจมาต่อเข้ากับวงจรดีเฟลคชั่นบริดจ์

โหลดเซลล์แบ่งออกเป็น 4 แบบ ดังนี้

2.1.1.1 โหลดเซลล์แบบลิงค์

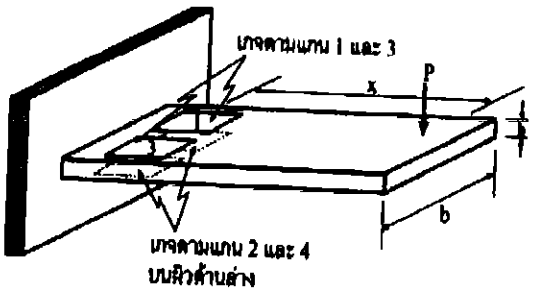
โหลดเซลล์แบบลิงค์ (Link-type load cell) ประกอบด้วยลิงค์และสเตรนเกจ ดังรูปที่ 2.2 ภาระ (Load) จุด P สามารถเป็นไปได้ทั้งแรงดึงหรือแรงอัด สเตรนเกจทั้ง 4 อัน จะต่อเป็นวงจรวีตสโตนบริดจ์โดยเกจที่อยู่ในแนวแกนต่ออยู่ในแขน 1 และ 3 ส่วนเกจที่อยู่ในแนวขวางต่ออยู่กับแขน 2 และ 4



รูปที่ 2.2 โหลดเซลล์แบบลิงค์

2.1.1.2 โหลดเซลล์แบบคาน

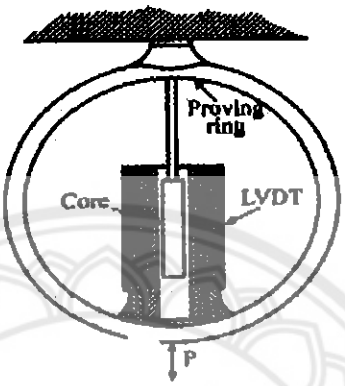
โหลดเซลล์แบบคาน (Beam-type load cell) ใช้วัดภาระในกรณีที่ใช้โหลดเซลล์แบบลิงค์ไม่ได้ จากรูปที่ 2.3 ใช้คานยื่นทำหน้าที่เป็นชิ้นส่วนยืดหยุ่นซึ่งมีสเตรนเกจ 2 อันติดอยู่ที่ผิวด้านบนและอีก 2 อัน ติดที่ผิวด้านล่าง (ทั้งหมดติดอยู่ในแนวขนานกับแกนของคาน) ทำหน้าที่เป็นตัวรับรู้



รูปที่ 2.3 โหลดเซลล์แบบคานประกอบด้วยชิ้นส่วนยืดหยุ่นกับสเตรนเกจ

2.1.1.3 โหลดเซลล์แบบวงแหวน

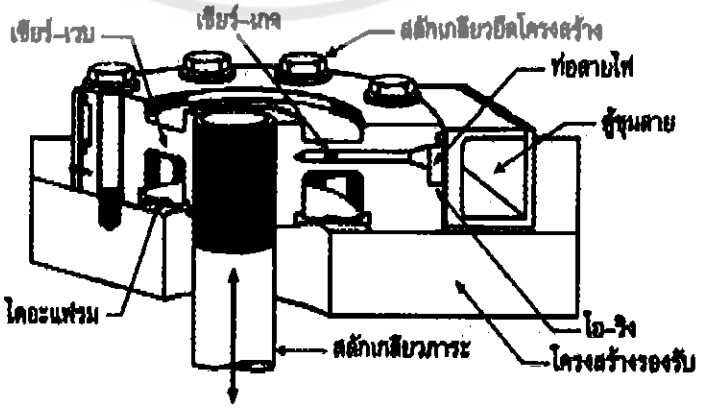
จากรูปที่ 2.4 แสดงส่วนประกอบของโหลดเซลล์แบบวงแหวน (Ring-type load cell) มีวงแหวนพรูฟวิง (Proving ring) เป็นส่วนยืดหยุ่นและมีตัวรับรู้ซึ่งสามารถใช้ได้ทั้งสเตรนเกจและ LVDT ดังแสดงใน



รูปที่ 2.4 โหลดเซลล์แบบวงแหวน

2.1.1.4 โหลดเซลล์ชนิดเชียร์-เวบ

โหลดเซลล์ชนิดเชียร์-เวบ (Shear-web-type load cell) เรียกอีกอย่างหนึ่งว่า โหลดเซลล์แบบแฟลต (Flat load cell) มีประโยชน์สำหรับการใช้งานเมื่อที่ว่างในแนวภาระกระทำมีจำกัด จากรูปที่ 2.5 โหลดเซลล์ชนิดเชียร์-เวบประกอบด้วย ดุมรับภาระข้างใน (Inner loading hub) และหน้าจานรองรับข้างนอก (Outer supporting flange) ซึ่งต่อกันด้วยเชียร์-เวบ สเตรนเกจชนิดรับแรงเฉือนติดตั้งอยู่ในรูเล็ก ๆ ที่ถูกเจาะเข้าไปในเชียร์-เวบ

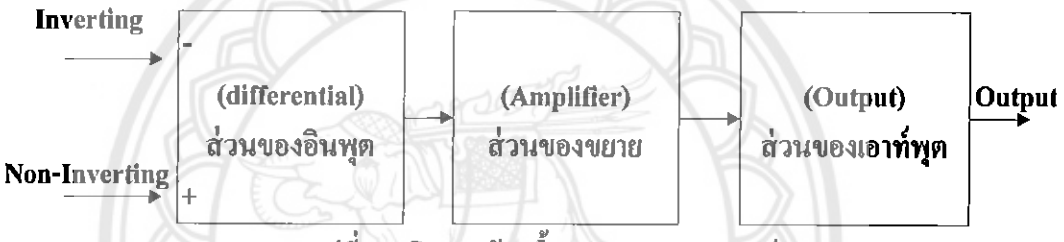


รูปที่ 2.5 โหลดเซลล์ชนิดเชียร์-เวบ

2.2 การออกแบบวงจรขยายสัญญาณและวงจรกรองความถี่

2.2.1 ความรู้พื้นฐานเกี่ยวกับออปแอมป์

ออปแอมป์ (Op-Amp) เป็นวงจรขยายสัญญาณที่อยู่ในรูปของตัวไอซี โดยจะมีอยู่ 2 แบบ คือ ออปแอมป์ แบบ 8 ขา และ 14 ขา ซึ่งจะมีลักษณะเป็นโลหะและจะเป็นแบบพลาสติก โดยจะมีการจัดวางขาเป็นแบบ (Dual In-Line Package: DIP) ซึ่งภายในตัวของออปแอมป์นั้นจะมีในของอุปกรณ์ทางอิเล็กทรอนิกส์ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ที่มีคุณสมบัติต่างๆ หลายตัวด้วยกัน คือ ทรานซิสเตอร์ (BJT), มอสเฟส (MOSFET), ไดโอด (Diode) และตัวต้านทาน (Resistor) มาประกอบรวมกันเป็นออปแอมป์ และออปแอมป์มีโครงสร้างที่สำคัญ 3 ส่วนก็คือ 1.ส่วนของสัญญาณเข้า (Differential) 2.ส่วนของการขยายสัญญาณ (Amplifier) 3.ส่วนของสัญญาณออก (Output) ดังในรูปที่ 2.6



รูปที่ 2.6 โครงสร้างพื้นฐานของออปแอมป์

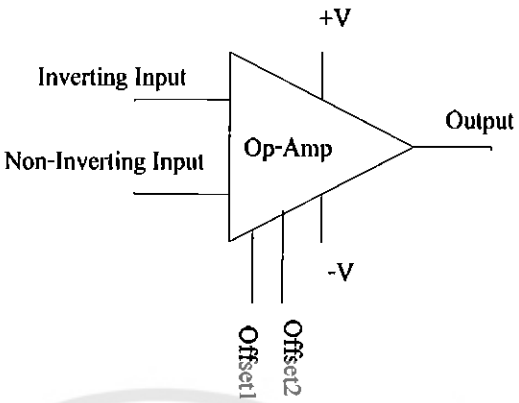
จากรูปที่ 2.6 เป็นโครงสร้างพื้นฐานของออปแอมป์ซึ่งประกอบด้วย

ส่วนของสัญญาณเข้า หรือที่เรียกว่า ส่วนผลต่าง นั้นจะมีลักษณะเป็นวงจรขยายสัญญาณผลต่าง ซึ่งถ้าดูในส่วนของอัตราการขยายสัญญาณอินพุตจะพบว่ามีอัตราการขยายที่สูง แต่ในทางกลับกันในส่วนของอัตราขยายสัญญาณอินพุตร่วมจะมีค่าที่ต่ำ และในส่วนของอิมพีแดนซ์ขาเข้าของวงจรจะมีค่าที่สูงมาก ซึ่งโดยทั่วไปจะมีค่าประมาณ 1 เมกะโอห์ม

ส่วนของการขยายสัญญาณ หรือที่เรียกว่า แอมพลิฟายเออร์ (Amplifier) นั้นก็จะทำหน้าที่ขยายสัญญาณทั้งแรงดันและกระแส เพื่อที่จะทำให้อัตราขยายโดยรวมนั้นมีค่าที่สูงขึ้น

ส่วนของสัญญาณออก หรือที่เรียกว่า เอาท์พุต ซึ่งในส่วนนี้ถ้ามาพิจารณาดูโครงสร้างของวงจรภายในก็จะทราบว่าจะเป็นวงจรขยายสัญญาณแบบอีมิตเตอร์ร่วม เพื่อที่จะทำให้อิมพีแดนซ์ที่ขาออกของวงจรมีค่าที่ต่ำและเพื่อที่จะสามารถขับกระแสให้กับโหลดที่ต่อได้

2.2.2 ลักษณะของขาต่างๆของออปแอมป์



รูปที่ 2.7 ลักษณะของขาที่ใช้งานของออปแอมป์

จากรูปที่ 2.7 จะเห็นถึงลักษณะโครงสร้างพื้นฐานของอุปกรณ์ออปแอมป์ ซึ่งคุณสมบัติโดยทั่วไปของออปแอมป์ จะมีคุณสมบัติพื้นฐานดังนี้ คือ อินพุตอิมพีแดนซ์จะมีค่าที่สูง เอาท์พุตอิมพีแดนซ์จะมีค่าต่ำและอัตราขยายแรงดันจะมีค่าที่สูงมาก ลักษณะโครงสร้างพื้นฐานของอุปกรณ์ออปแอมป์ ประกอบด้วยขาที่จะนำมาใช้งานดังนี้

ขาอินเวอร์ติง (Inverting) ซึ่งเป็นขาอินพุต (Input) ของออปแอมป์ โดยมีไว้เพื่อป้อนสัญญาณที่เป็นได้ทั้งสัญญาณไฟกระแสตรงและสัญญาณไฟกระแสสลับซึ่งสัญญาณที่ถูกป้อนเข้าไปที่ขาอินเวอร์ติง ก็จะได้สัญญาณกลับเฟสที่จะออกมาทางขาเอาท์พุต

ขานอนอินเวอร์ติง (Non-Inverting) ซึ่งเป็นขาอินพุตของออปแอมป์ โดยมีไว้เพื่อป้อนสัญญาณ ที่เป็นได้ทั้งสัญญาณไฟกระแสตรงและสัญญาณไฟกระแสสลับ ซึ่งสัญญาณที่ถูกป้อนเข้าไปที่ขานอนอินเวอร์ติง ก็จะได้สัญญาณที่ไม่กลับเฟสออกมาทางขาเอาท์พุต

ขาเอาท์พุต ซึ่งเป็นขาเอาท์พุตของออปแอมป์ โดยมีไว้เพื่อบอกถึงสภาวะการทำงานของออปแอมป์ ซึ่งจะเกิดขึ้นจากการที่ทำการป้อนสัญญาณเข้าที่ขาอินพุต ทั้งสองของออปแอมป์

ขาป้อนไฟบวก (+V) โดยจะมีค่าแรงดันไฟประมาณ 9 โวลต์ ถึง 18 โวลต์ ซึ่งโดยทั่วไปแล้วจะใช้แรงดันไฟประมาณ 15 โวลต์

ขาป้อนไฟลบ (-V) โดยจะมีค่าแรงดันไฟประมาณ -9 โวลต์ ถึง -18 โวลต์ ซึ่งโดยทั่วไปแล้วจะใช้แรงดันไฟประมาณ -15 โวลต์

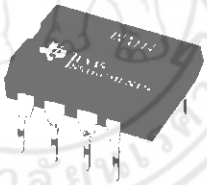
ขาออฟเซทหนึ่ง (Offset1) และ ขาออฟเซทสอง (Offset2) โดยทั้งสองขานี้จะใช้ต่ออุปกรณ์เพื่อที่จะเป็นการป้องกันการเกิดออสซิเลตของออปแอมป์

2.2.3 เทคนิคการเลือกใช้ออปแอมป์

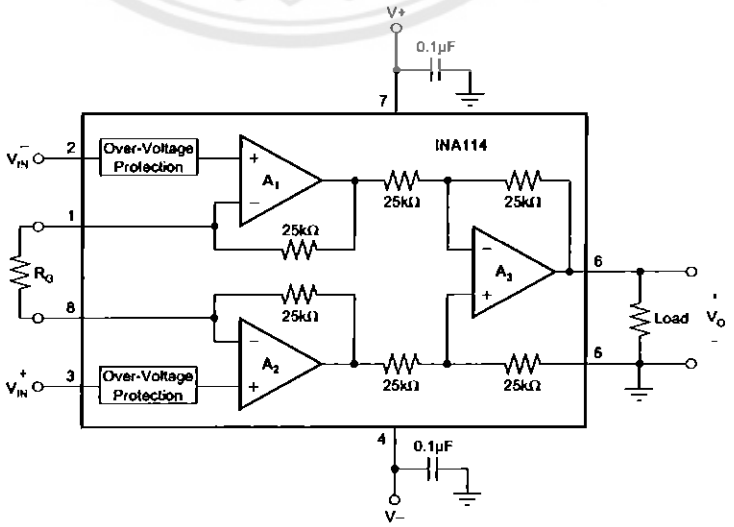
ออปแอมป์ เป็นอุปกรณ์ที่จะพบเห็นได้ทั่วไปในส่วนวงจรแอนะล็อกของหลายๆ ระบบ ออปแอมป์ ในปัจจุบันมีให้เลือกใช้มากมาย เมื่อจะเลือกออปแอมป์มาใช้ ก็ต้องลดจำนวนตัวเลือกลงโดยพิจารณาจากเกณฑ์หรือคุณสมบัติบางประการคือ การกินกำลังไฟต่ำ (Low-Power) แบนวิธ (Bandwidth) ระดับแรงดันตั้งแต่ระดับไฟเลี้ยงด้านลบจนถึงระดับไฟเลี้ยงด้านบวก (Rail-to-rail op amps) การทำงานที่แรงดันต่ำ (Low-voltage operation) ความแม่นยำ (Precision) และสัญญาณรบกวนต่ำ (Low noise)

2.2.4 วงจรขยายอินสตรูเมนท์

วงจรขยายอินสตรูเมนท์ (Instrument Amplifier Circuit) เป็นวงจรขยายสัญญาณในการวัดทางอุตสาหกรรมจะมีวงจรออปแอมป์ที่ใช้ในการขยายสัญญาณที่ได้จากการวัดซึ่งใช้ในงานอุตสาหกรรมทั่ว ๆ ไป ในการออกแบบวงจรจะใช้รูปแบบของวงจรขยายอินสตรูเมนท์ ดังรูปที่ 2.8 เป็นวงจรขยายอินสตรูเมนท์ ซึ่งเป็นวงจรรวม (Integrated Circuit: IC) สำเร็จรูปเบอร์ INA114 โดยมีความสามารถในการกำจัดแรงดันออฟเซททั้งที่เป็นบวกและลบอันเนื่องมาจากร่างกายได้ โดยการใช้เทคนิคการป้อนกลับทางแสงจากเอาต์พุตมายังอินพุตของวงจร แรงดันออฟเซทนี้หากปะปนเข้าไปในวงจรจะทำให้วงจรไม่สามารถขยายสัญญาณได้ทั้งนี้เพราะวงจรเกิดการอิ่มตัว



รูปที่ 2.8 วงจรรวม (Integrated Circuit: IC) สำเร็จรูปเบอร์ INA114



รูปที่ 2.9 วงจรขยายสัญญาณอินสตรูเมนท์

จากรูปที่ 2.9 เป็นวงจรขยายสัญญาณอินสตรูเมนท์ ที่ออกแบบอยู่ในวงจรรวม สำเร็จรูป เบอร์ INA114 ซึ่งประกอบไปด้วยออปแอมป์และตัวต้านทาน ร่วมถึงวิธีการต่อเมื่อนำไปใช้งาน

ในวงจรนี้สามารถกำหนดอัตราขยายได้ด้วยสมการที่ 2.1

$$G = 1 + \frac{50k\Omega}{R_G} \tag{2.1}$$

จากสมการที่ 2.1 เป็นคุณสมบัติของวงจรขยายอินสตรูเมนท์ จะเห็นได้ว่า R_G เป็นตัวกำหนดอัตราขยาย ยกตัวอย่างเช่น ต้องการอัตราอันตราขยายที่ 300 เท่า

จะได้ว่า

$$\begin{aligned} R_G &= \frac{50k\Omega}{G-1} \\ &= \frac{50k\Omega}{300-1} \\ &= 167.22\Omega \end{aligned}$$

ดังนั้นเลือกใช้ R ปรับค่าได้ ที่ 167 โอห์ม อัตราขยาย 300.40 เท่า

2.2.5 วงจรกรองสัญญาณ

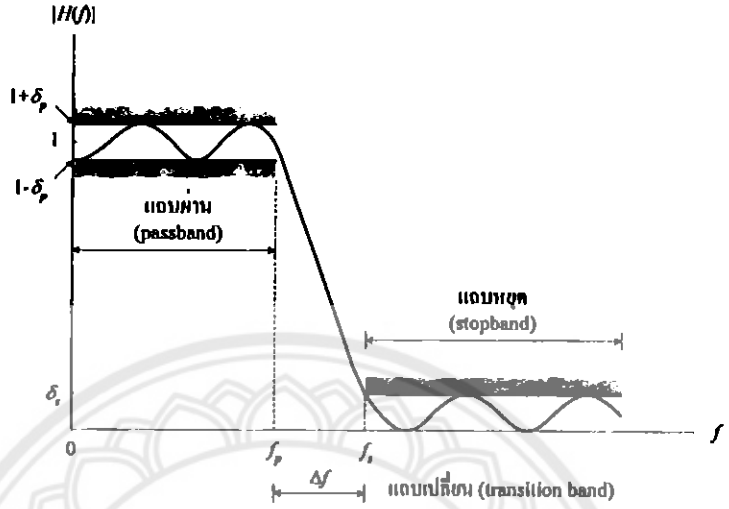
2.2.5.1 วงจรกรองผ่านต่ำ

วงจรกรองผ่านต่ำ (Lowpass Filter) มีคุณสมบัติในการยอมให้ความถี่ต่ำสามารถผ่านได้ การกำหนดคุณลักษณะของผลตอบสนองเชิงขนาดสำหรับวงจรกรองผ่านต่ำ ดังรูปที่ 2.10 มีการแบ่งคุณสมบัติทางความถี่เป็น 3 ส่วน ได้แก่ แถบผ่าน (Passband) แถบเปลี่ยน (Transition Band) และแถบหยุด (Stopband) ช่วงแถบผ่านจะถูกกำหนดด้วยพารามิเตอร์สองตัว ได้แก่ δ_p และ f_p คือ ความถี่ตั้งแต่กระแสตรงไปจนถึงความถี่ f_p จะได้รับการส่งผ่านโดยมีค่าผลตอบสนองเชิงขนาด $|H(f)|$ อยู่ในช่วง $1-\delta_p$ และ $1+\delta_p$ กล่าวคือ

$$1-\delta_p \le |H(f)| \le 1+\delta_p \text{ สำหรับ } |f| \le f_p \tag{2.2}$$

ช่วงแถบเปลี่ยนมีพารามิเตอร์สองตัว ได้แก่ f_p และ f_s เป็นตัวกำหนดช่วงความกว้างของความถี่สูงสุดที่วงจรกรองจะเปลี่ยนจากการทำงานในช่วงของแถบผ่านไปสู่ช่วงของแถบหยุด ซึ่งวงจรกรองที่ดีจะต้องกำหนดให้ช่วงเปลี่ยน Δf มีขนาดเล็ก และช่วงสุดท้าย ช่วงแถบหยุดมีพารามิเตอร์สองตัว ได้แก่ f_s และ δ_s พารามิเตอร์ f_s เป็นตัวกำหนดค่าความถี่แรกที่วงจรกรองจะต้อง

ทำงานในช่วงของแถบหยุด ส่วนพารามิเตอร์ δ_s เป็นค่าผลตอบสนองเชิงขนาด $|H(f)|$ สูงสุดที่วงจรกรองจะยอมรับได้สำหรับองค์ประกอบความถี่ของสัญญาณขาเข้าที่สูงกว่า f_s



รูปที่ 2.10 การกำหนดคุณลักษณะของวงจรกรองผ่านต่ำ

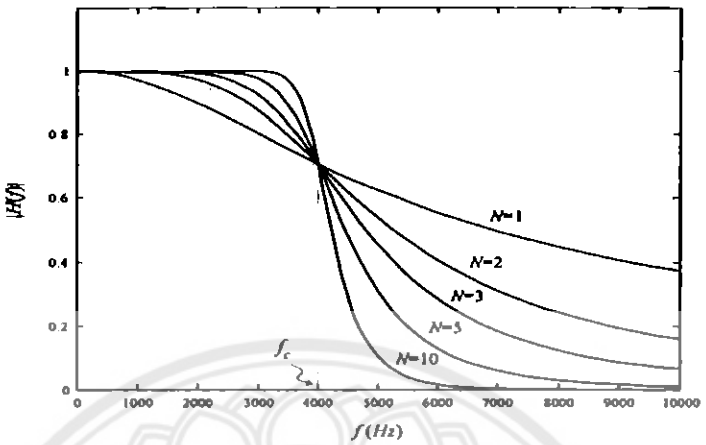
2.2.6 วงจรผ่านต่ำบัตเทอร์เวิร์ท

วงจรกรองผ่านต่ำบัตเตอร์เวิร์ท อันดับที่ N (Nth Order Butterworth Lowpass Filter) มีผลตอบสนองเชิงแอมพลิจูด (amplitude response) ดังสมการที่ 2.3

$$|H(f)| = \frac{G}{\sqrt{1+\left(f/f_c\right)^{2N}}} \tag{2.3}$$

โดยที่ G คือ อันตราขยายแถบผ่าน ซึ่งในกรณีของวงจรผ่านต่ำบัตเตอร์เวิร์ทกำหนดให้มีค่าเท่ากับอัตราขยายที่ความถี่เท่ากับ 0 เฮิรตซ์ กล่าวคือ $G = |H(0)|$ สำหรับ f_c คือ ความถี่ตัด (Cutoff Frequency) ของวงจรกรอง ณ ที่ความถี่ดังกล่าวผลตอบสนองเชิงแอมพลิจูดจะลดลงเหลือเท่ากับ $|H(f_c)| = G/\sqrt{2}$ นั่นคือมีอัตราขยายลดลง $1/\sqrt{2}$ เท่ากับอัตราขยายแถบผ่าน โดยทั่วไปจะกำหนดให้ $G=1$ พิจารณากราฟดังรูปที่ 2.11 แสดงการเปรียบเทียบผลตอบสนองเชิงแอมพลิจูดของวงจรกรองผ่านต่ำบัตเตอร์เวิร์ทอันดับต่าง ๆ โดยกำหนดที่ความถี่ตัดที่เดียวกันคือ 4,000 เฮิรตซ์ สังเกตว่าเมื่อวงจรมีอันดับเพิ่มขึ้นอัตราขยายมีการเปลี่ยนแปลงจากแถบผ่านไปแถบหยุดได้อย่างรวดเร็วมากขึ้น นั่นคือวงจรมีความแคบมากขึ้นนอกจากนี้ในช่วงแถบผ่านหรือแถบหยุด มีการแกว่งตัวขึ้นลงของอัตราขยายอย่างใด (Maximally Flat) เนื่องจากผลตอบสนองเชิงแอม

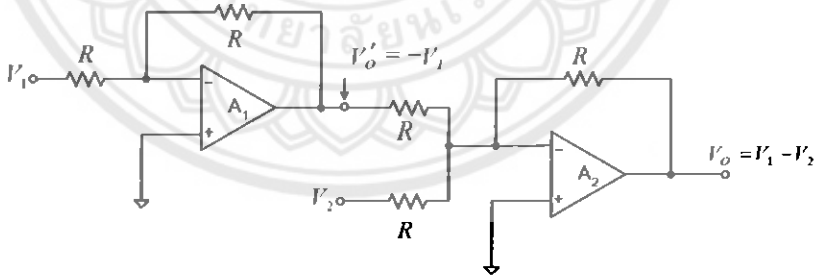
พลิจูดมีค่าลดลงตลอดตามการเพิ่มขึ้นของความถี่ จึงจัดว่าเป็นคุณลักษณะของฟังก์ชันโมโนโทนิก (Monotonic Function) และเป็นคุณสมบัติเฉพาะตัวของวงจรกรองผ่านต่ำบัตเตอร์เวิร์ท



รูปที่ 2.11 ฟังก์ชันผลตอบสนองเชิงแอมพลิจูดของวงจรกรองผ่านต่ำบัตเตอร์เวิร์ท

2.2.7 วงจรลบสัญญาณ

สามารถสร้างวงจรลบสัญญาณ (Substractor) ได้โดยการใช้วงจรขยายแบบกลับเฟสที่มีอัตราขยายแรงดันเป็น 1 เพื่อเปลี่ยนเครื่องหมายของสัญญาณอินพุต จากนั้นจึงนำ มาผ่านวงจรบวกสัญญาณ ดังวงจรในรูปที่ 2.12



รูปที่ 2.12 วงจรลบสัญญาณ

จากรูปที่ 2.12 พบว่าออปแอมป์ A_1 ต่อเป็นวงจรขยายแบบกลับเฟสที่มีอัตราขยายแรงดันเป็น 1 ดังนั้น $V_0' = -V_1$ ส่วนออปแอมป์ A_2 ต่อเป็นวงจรบวกสัญญาณที่มีอัตราขยายเป็น 1 เท่านั้น ดังนั้น

$$V_0 = -\left[-V_1 + V_2\right] \tag{2.4}$$

$$V_0 = V_i - V_2 \tag{2.5}$$

โดยที่ V_i สัญญาณอินพุต (Input) และ V_2 สัญญาณลบ

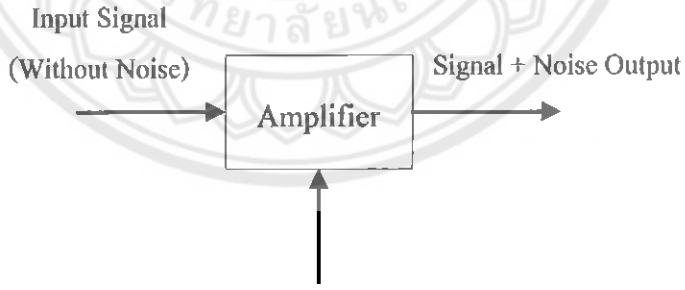
2.2.8 สัญญาณรบกวน

สัญญาณรบกวน (Noise) หมายถึงกระแสหรือแรงดันจากภายนอกที่เข้ามาข้องเกี่ยวกับวงจรไฟฟ้าหรืออิเล็กทรอนิกส์จนทำให้เกิดผลผิดพลาดได้ การเกิดของสัญญาณรบกวนอาจมาจากวงจรภายนอกและเข้ามาในวงจรในทิศทางใดทิศทางหนึ่งหรืออาจเกิดขึ้นจากวงจรภายในเองก็ได้

แหล่งกำเนิดของสัญญาณรบกวนสัญญาณรบกวนสามารถนิยามได้ด้วยค่าที่ไม่ต้องการซึ่งเป็นค่าการเบี่ยงเบนใดๆ ที่ได้จากการเปรียบเทียบกับค่าที่คาดหวังต่อค่าสัญญาณที่สนใจ ความสำคัญของสัญญาณรบกวนจะมีความสัมพันธ์กับขนาดแอมพลิจูด (Amplitudes) ของค่าที่ไม่ต้องการต่อค่าสัญญาณที่สนใจถ้าค่าที่ไม่ต้องการมีขนาดเล็กอัตราส่วนของสัญญาณเมื่อเทียบกับสัญญาณรบกวนจะมีค่ามาก (Large) หมายความว่าสัญญาณรบกวนก็จะไม่มีความสำคัญ

ในกระบวนการวัดสามารถกำหนดระดับการยอมรับได้ของอัตราส่วนความสัมพันธ์ระหว่างสัญญาณต่อสัญญาณรบกวนที่มีค่ามากให้คงไว้ได้ด้วยการเพิ่มระดับสัญญาณให้สูงขึ้นโดยปราศจากการเพิ่มขึ้นของสัญญาณรบกวนหรืออาศัยเทคนิควิธีการลดสัญญาณรบกวน

ตัวกำหนดคุณลักษณะของสัญญาณรบกวนซึ่งเป็นส่วนประกอบที่อาจเกิดขึ้นจากภายในตัวขยายสัญญาณจากวงจรที่ทำงานเชิงกล 3 อย่าง สามารถพิจารณาได้จากรูปที่ 2.13



รูปที่ 2.13 แสดงกลไกการพัฒนาของสัญญาณรบกวนที่อาจเกิดขึ้นในวงจร

จากรูปที่ 2.13 กำหนดให้สัญญาณอินพุตไม่มีสัญญาณรบกวน ส่วนขยายสัญญาณมีแหล่งจ่ายพลังงานไฟฟ้าสัญญาณเอาต์พุตโดยประมาณมีค่าเป็นหลายเท่าของสัญญาณอินพุตที่รวมกับสัญญาณรบกวนแหล่งของสัญญาณรบกวนที่เป็นไปได้อาจเกิดจากอุปกรณ์ในระบบการขยายสัญญาณ เช่น ตัวต้านทาน ตัวเก็บประจุ ทรานซิสเตอร์ และอื่นๆ เป็นต้น ซึ่งเป็นการผลิตขึ้นภายในตัวขยายสัญญาณและเรียกว่าสัญญาณรบกวนจากการผลิต (Generated Noise) แหล่งจ่ายพลังงาน

ไฟฟ้าอาจเป็นแหล่งของสัญญาณรบกวนได้หากรูปคลื่นมีลักษณะที่ไม่เป็นปกติ เช่น คลื่นที่มีลักษณะยอดแหลม คลื่นที่มาเป็นระลอกหรือเป็นช่วง ๆ และคลื่นที่มีการเบี่ยงเบนไม่แน่นอน เป็นต้น และถูกนำเข้าไปสู่ตัวขยายสัญญาณโดยผ่านสายไฟฟ้ากำลังหรือสายต่อเครื่องมือวัด สัญญาณรบกวนในลักษณะนี้เรียกว่า สัญญาณรบกวนจากตัวนำ (Conducted Noise) สัญญาณรบกวนจากแม่เหล็กไฟฟ้าเป็นการแผ่รัศมีไปในบริเวณโดยรอบของตัวขยายสัญญาณอาจเกิดจากไฟฟ้าสนามแม่เหล็ก หรือสิ่งรบกวนต่าง ๆ โดยทั่วไปมักเป็นส่วนประกอบของแม่เหล็กไฟฟ้าชั่วขณะ เช่น การเกิดประกายไฟที่เต้ารับและการผลิตสัญญาณรบกวนในภาครับวิทยุ เป็นต้น สัญญาณรบกวนจะแผ่รัศมีไปคล้องเกี่ยวกับเสาอากาศวิทยุและนำไปสู่การทำงานของวงจรอิเล็กทรอนิกส์ภายในหรืออาจเป็นการนำไปสู่การทำงานของวงจรโดยผ่านสายไฟฟ้ากำลังก็ได้

2.2.9 เทคนิคการลดสัญญาณรบกวน

ในการขจัดหรือลดสัญญาณรบกวนให้อยู่ในระดับที่น่าพอใจนั้นลำดับแรกต้องบอกให้ได้ว่าสัญญาณรบกวนเข้ามาในระบบได้อย่างไร อาจเกิดขึ้นจากการผลิต การนำหรือการแผ่รัศมี จากนั้นจึงนำเข้าสู่ขั้นตอนความเป็นไปได้ที่จะแก้ปัญหาหากมีสัญญาณรบกวนจากการผลิตภายในจากกลไกต่าง ๆ ตัวต้านทานอาจเป็นแหล่งหนึ่งเนื่องจากการนำของตัวต้านทานนั้นประกอบด้วยกลุ่มของอะตอมที่ต้องคงสภาพของความนำเอาไว้ อะตอมเหล่านี้จะถูกส่ง หรือนำอิเล็กตรอนไปทำให้เกิดกระแสไหลแต่เนื่องจากความเร็วที่เกิดจากการสั่นสะเทือนภายในอะตอมนั้น ทำให้เกิดความร้อนหรืออุณหภูมิการสั่นของอะตอมจะส่งต่อไปยังอิเล็กตรอนซึ่งอาจเกิดขึ้นร่วมกันกับกระแสที่ไม่ต้องการหรือสัญญาณรบกวน เมื่อสัญญาณรบกวนขึ้นอยู่กับอุณหภูมิและเพิ่มขึ้นด้วยค่าสูญเสียในลักษณะของความร้อน หรือในลักษณะสภาพแวดล้อมที่มีอุณหภูมิเพิ่มขึ้น สัญญาณรบกวนที่เกิดโดยกลไกนี้เรียกว่า สัญญาณรบกวนแบบจอห์นสัน (Johnson noise) หากการสั่นสะเทือนภายในอะตอมตัวต้านทานขึ้นกับช่วงของความถี่แล้วสัญญาณรบกวนที่เกิดขึ้นก็จะเป็นส่วนประกอบอันเนื่องมาจากแถบกว้างความถี่ บางครั้งเรียกว่าสัญญาณรบกวนสีขาว (white noise) เช่น สัญญาณรบกวนที่เกิดขึ้นจากเครื่องบินไอพ่น เป็นต้น

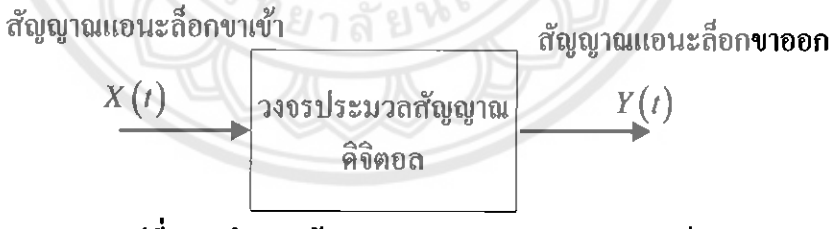
ความถี่เป็นส่วนประกอบอันหนึ่งที่ทำให้เกิดสัญญาณรบกวนได้ เช่น แหล่งจ่ายไฟฟ้าจ่ายสัญญาณที่ความถี่ 60 เฮิรตซ์ นำเข้าสู่ระบบโดยผ่านสายตัวนำไฟฟ้าและส่งถ่ายเข้าสู่อุปกรณ์อื่นในระบบต่อไปหรืออาจแผ่รัศมีไปสู่ระบบโดยผ่านสายตัวนำและจากหม้อแปลงการลดสัญญาณรบกวนนี้สามารถทำได้แต่ก็ยากมากเนื่องจากเป็นการเข้ามาด้วยการแผ่รัศมีหรือด้วยการนำสัญญาณถ้าเกิดจากการนำสัญญาณอาจแก้ได้ด้วยการใช้ตัวกรองที่วางที่ตัวนำสำหรับดักสัญญาณรบกวนแต่ถ้าเกิดจากการแผ่รัศมีอาจแก้ได้ด้วยการวางแนวตัวนำป้องกัน (Shielding) เพื่อลดสัญญาณที่จะแทรกเข้ามาได้การวางแนวป้องกันจะมีความยุ่งยากสูงเพราะต้องป้องกันที่ตัวนำสายทดสอบสายแหล่งจ่ายไฟฟ้า หรืออาจทำทั้งหมดภายในวงจร

2.3 ประมวลผลสัญญาณดิจิตอล

ปัจจุบันการประมวลสัญญาณเกิดขึ้นมากมายในระบบสื่อสารโทรคมนาคม ระบบวัดและการควบคุมเครื่องจักรในโรงงานอุตสาหกรรม การบังคับหุ่นยนต์ ระบบนำวิถีจรวดในวงการทหาร การตรวจจับเครื่องบินด้วยเรดาร์ การพยากรณ์อากาศจากภาพถ่ายดาวเทียม การวินิจฉัยโรคจากภาพถ่ายรังสีเอกซ์ การบีบอัดข้อมูลเพื่อการบันทึกจัดเก็บ และอื่น ๆ อีกมากมาย กรรมวิธีการประมวลสัญญาณสามารถแบ่งแยกออกได้เป็น 2 ประเภทหลัก ได้แก่

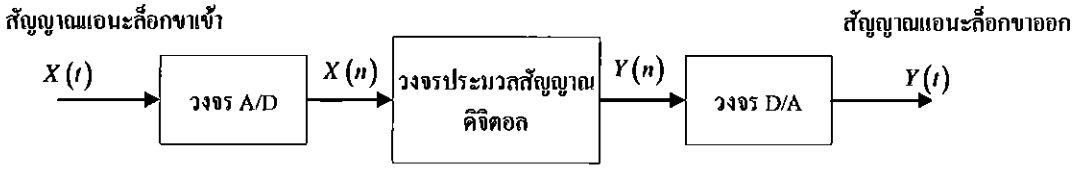
- กรรมวิธีสัญญาณแอนะล็อก (analog signal processing)
- กรรมวิธีสัญญาณดิจิตอล (digital signal processing)

ในอดีตการประมวลสัญญาณแทบทั้งหมดจะอาศัยกรรมวิธีทางแอนะล็อกเป็นหลัก ดูโครงสร้างการทำงานโดยรวมของระบบที่ใช้การประมวลสัญญาณแอนะล็อกในรูปที่ 2.14 จากรูปสัญญาณแอนะล็อกที่ป้อนเข้าสู่ระบบจะได้รับการประมวลผลโดยตรงในรูปของสัญญาณที่มีค่าต่อเนื่องทางเวลา โดยอุปกรณ์ทางไฟฟ้าที่ใช้ในกระบวนการประมวลผลสัญญาณแอนะล็อกที่สำคัญประกอบด้วย ตัวต้านทาน ตัวเก็บประจุ ตัวเหนี่ยวนำ วงจรขยาย และไดโอด สัญญาณที่ได้ทางขาออกมักจะอยู่ในรูปของสัญญาณแอนะล็อกด้วยเช่นกัน ข้อดีของการประมวลสัญญาณในลักษณะนี้คือ อุปกรณ์ต่าง ๆ ที่ใช้ในการประมวลผลสามารถแทนสมการอนุพันธ์สำหรับจำลองการทำงานของระบบจริงได้โดยไม่ต้องมีการประมาณค่า นอกจากนี้ระบบสามารถทำงานแบบเรียลไทม์ (real-time) ได้โดยไม่ขึ้นกับความถี่ของสัญญาณแอนะล็อกขาเข้า



รูปที่ 2.14 โครงสร้างระบบประมวลสัญญาณแอนะล็อก

และปัจจุบันความก้าวหน้าของเทคโนโลยีคอมพิวเตอร์ได้ส่งผลให้การประมวลสัญญาณดิจิตอลมีบทบาทเพิ่มมากขึ้นและได้เข้ามาแทนที่การประมวลสัญญาณแอนะล็อกลักษณะการทำงานของกรรมวิธีประมวลสัญญาณดิจิตอลสามารถแบ่งแยกออกได้เป็นสามส่วนหลักได้แก่ วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล (A/D converter) วงจรประมวลสัญญาณดิจิตอลและวงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก (D/A converter) ดังแสดงในรูปที่ 2.15



รูปที่ 2.15 โครงสร้างระบบประมวลสัญญาณดิจิตอล

จากรูปที่ 2.15 แปลงสัญญาณขาเข้า $X(t)$ ให้เป็นสัญญาณดิจิตอล $X(n)$ ซึ่งเป็นสัญญาณที่มีค่าดิสครีตทางเวลา (discrete time signal) ก่อนจะนำไปผ่านการประมวลผล กรรมวิธีการประมวลสัญญาณดิจิตอลมีความต่างไปจากกรณีของการประมวลสัญญาณแอนะล็อก โดยอุปกรณ์หลักที่ใช้ในการประมวลสัญญาณดิจิตอลประกอบด้วย วงจรบวก วงจรลบ และหน่วยความจำ ซึ่งก็คือการคำนวณตัวเลขในทางคณิตศาสตร์ผลที่ได้จากการประมวลผลจะยังคงอยู่ในรูปของสัญญาณดิจิตอล ดังนั้นในขั้นตอนท้ายสุดจึงต้องมีกระบวนการแปลงกลับให้ได้เป็นสัญญาณแอนะล็อกเหมือนเดิมโดยใช้วงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก (D/A)

ในระบบการประมวลผลสัญญาณดิจิตอลต้องมีการเพิ่มขั้นตอนขึ้นสองส่วนเพื่อแปลงสัญญาณแอนะล็อกไปเป็นดิจิตอลก่อนการประมวลผลและแปลงกลับให้มีรูปแบบเดิมขั้นตอนเหล่านี้ทำให้มีข้อจำกัดเกิดขึ้น กล่าวคือในบางสถานการณ์ระบบไม่สามารถทำงานแบบเรียลไทม์ได้โดยเฉพาะเมื่อต้องประมวลสัญญาณที่มีความถี่สูงมาก ๆ อย่างไรก็ตาม ระบบการประมวลสัญญาณดิจิตอลมีข้อดีหลายประการดังนี้คือ

ประการแรกคือ ระบบการประมวลสัญญาณดิจิตอลมีความยืดหยุ่นสูงมาก สามารถปรับเปลี่ยนไปตามความต้องการของผู้ออกแบบได้อย่างรวดเร็วและมีประสิทธิภาพ ยกตัวอย่างเช่นในการออกแบบวงจรกรองผ่านต่ำสำหรับใช้งานในระบบสื่อสารหนึ่ง เมื่อได้มีการระบุข้อกำหนดของวงจรไว้แล้ว สามารถเลือกออกแบบโดยใช้วงจรประมวลสัญญาณแอนะล็อกหรือวงจรประมวลสัญญาณดิจิตอลก็ได้และก็มักจะให้ผลที่ใกล้เคียงกัน แต่อย่างไรก็ตามพิจาณาหากเกิดสถานการณ์ที่มีความต้องการให้ปรับข้อกำหนดของวงจรกรองใหม่แม้เพียงเล็กน้อย การออกแบบโดยใช้วงจรประมวลสัญญาณแอนะล็อกจะต้องมีการปรับแก้วงจรใหม่หรือเปลี่ยนอุปกรณ์บางส่วนออก ในขณะที่ถ้าเป็นการออกแบบโดยใช้การประมวลสัญญาณดิจิตอล ผู้ออกแบบเพียงแต่ปรับแก้ค่าตัวเลขในโปรแกรมใหม่บางส่วนเท่านั้น ไม่ต้องมีการเปลี่ยนอุปกรณ์ใด ๆ ในระบบ นอกจากนี้หากต้องการออกแบบวงจรกรองรูปแบบอื่น เช่น วงจรกรองผ่านสูง แน่นอนว่าการใช้วงจรประมวลสัญญาณแอนะล็อกจะต้องมีการออกแบบใหม่ทั้งหมดและใช้อุปกรณ์ที่ต่างไปจากเดิม แต่สำหรับการออกแบบโดยใช้การประมวลสัญญาณดิจิตอลสามารถใช้อุปกรณ์ชุดเดิมได้ เพียงแต่ต้องมีการปรับเปลี่ยนโปรแกรมในวงจรใหม่เท่านั้น

ข้อได้เปรียบอีกประการของการประมวลสัญญาณดิจิตอลคือขีดความสามารถในการปรับตัวได้ในขณะใช้งาน เนื่องจากคุณลักษณะของระบบประมวลสัญญาณดิจิตอลสามารถปรับเปลี่ยนได้โดยง่ายจากโปรแกรมคอมพิวเตอร์ที่เป็นเพียงการคำนวณตัวเลข ทำให้สามารถสร้างระบบที่สามารถปรับเปลี่ยนคุณลักษณะการทำงานระหว่างการใช้งานจริงได้ เพื่อให้สอดคล้องกับสภาพแวดล้อมที่มีการเปลี่ยนแปลงไป ปัจจุบันความสามารถในการปรับตัวได้นี้มีการใช้งานอย่างกว้างขวางในทางปฏิบัติ และเป็นวิทยาการที่เรียกโดยรวมว่า วงจรกรองปรับตัวได้ (adaptive filter)

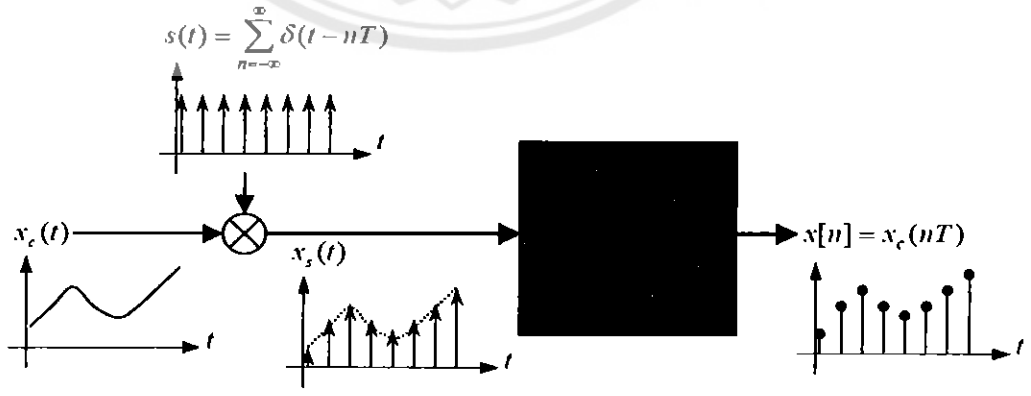
2.3.1 การสุ่มตัวอย่าง

2.3.1.1 การสุ่มตัวอย่างรายคาบ

โดยทั่วไปการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล จะใช้วิธีการสุ่มตัวอย่างรายคาบ (Periodic Sampling) ที่คงที่ซึ่งเป็นวิธีมาตรฐาน เป็นไปได้เหมือนกันที่จะใช้วิธีการสุ่มตัวอย่างรายคาบแบบไม่คงที่เพื่อที่จะปรับตัวตามธรรมชาติของสัญญาณมีการเปลี่ยนแปลงตลอดเวลา แต่วิธีนี้ไม่เป็นที่นิยมในปัจจุบัน เนื่องจากระบบควบคุมการเปลี่ยนรายคาบตามลักษณะของสัญญาณเข้ามีความซับซ้อนสูงและทำให้ระบบการประมวลสัญญาณต้องปรับตัวตามไปด้วย สมการการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลสามารถเขียนได้ดังนี้

$$x(n) = x_c(nT) \qquad , \ -\infty < n < \infty \tag{2.6}$$

โดยที่ T คือคาบการสุ่มตัวอย่าง (Sampling Period) และ $f_s = \frac{1}{T}$ คือความถี่การสุ่มตัวอย่าง (Sampling Frequency) โครงสร้างบล็อกพร้อมลักษณะของสัญญาณในแต่ละจุดได้ ดังรูปที่ 2.16



รูปที่ 2.16 การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลและลักษณะสัญญาณที่เกี่ยวข้อง
ที่มา: วุฒิพงศ์ (2548)

จากรูปที่ 2.16 จะเห็นได้ว่า จากสัญญาณ $x_c(t)$ ซึ่งเป็นสัญญาณแอนะล็อกหรือสัญญาณต่อเนื่อง จะถูกนำมาคูณกับสัญญาณสุ่มความถี่และคาบคงที่ $s(t)$ จะทำให้กลายเป็นสัญญาณ $x_s(t)$ คือสัญญาณไม่ต่อเนื่อง ดังต่อไปนี้

$$x_s(t) = x_c(t)s(t) \tag{2.7}$$

แทนค่าสัญญาณสุ่ม $s(t)$ ด้วยการเขียนในลักษณะอนุกรมของสัญญาณอิมพัลส์

$$= x_c(t)\sum_{n=-\infty}^{\infty}\delta(t-nT) \tag{2.8}$$

สัญญาณ $x_c(t)$ ไม่เกี่ยวข้องกับผลรวมอนุกรม จึงสามารถดึงเข้าไปอยู่ร่วมในอนุกรมได้

$$= \sum_{n=-\infty}^{\infty} x_c(nT)\delta(t-nT) \tag{2.9}$$

เพื่อความเข้าใจการสุ่มตัวอย่างต้องพิจารณาสัญญาณในเชิงความถี่ด้วยพิจารณาการแทนการแปลงฟูริเยร์ของสัญญาณสุ่ม $s(t)$

$$s(t) \xleftrightarrow{F} S(j\Omega) \tag{2.10}$$

$$\sum_{n=-\infty}^{\infty}\delta(t-nt) \xleftrightarrow{F} \frac{2\pi}{T}\sum_{k=-n}^{\infty}\delta(\Omega-k\Omega_s) \tag{2.11}$$

โดยที่ $\Omega_s = \frac{2\pi}{T}$ radian/second การคูณทางเวลาเทียบเท่ากับการคอนโวลูชันทางความถี่ แต่สัญญาณ $S(j\Omega)$ เป็นลักษณะอิมพัลส์ การคอนโวลูชันกับอิมพัลส์เดี่ยวจะทำให้ได้สัญญาณเหมือนเดิม

$$X_s(j\Omega) = \frac{1}{2\pi}X_c(j\Omega) * S(j\Omega) \tag{2.12}$$

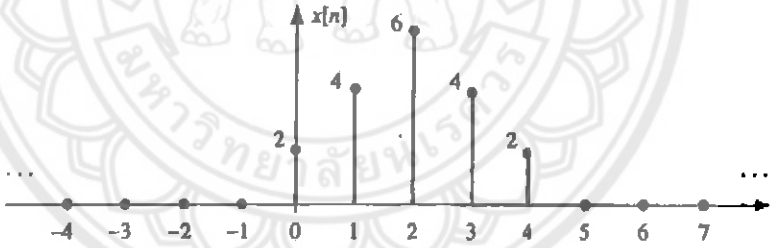
$$= \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j\Omega - k\Omega_s) \tag{2.13}$$

2.3.2 การออกแบบตัวกรองดิจิตอล

2.3.2.2 กรองเฉลี่ยแบบเคลื่อนที่ (Moving Average Filter)

การทรานสฟอร์มของสัญญาณดีสครีตไทม์ที่ง่ายแต่มีประโยชน์ คือการทำการคำนวณค่าเฉลี่ยแบบเคลื่อนที่ (Moving Average หรือ Running Average) ของจำนวนตัวเลขสองจำนวนหรือมากกว่าที่อยู่ติดกันในซีเควนซ์ และทำการสร้างซีเควนซ์ของค่าเฉลี่ยออกมา เอฟไออาร์ฟิลเตอร์คือกรณีที่กว้างขึ้นของแนวคิดของการหาค่าเฉลี่ยแบบเคลื่อนที่นี้ การทำการหาค่าเฉลี่ยนี้ใช้บ่อยเมื่อข้อมูลมีการเปลี่ยนแปลงค่าอย่างรวดเร็วและจะต้องถูกทำให้สัญญาณเรียบก่อนที่จะนำไปประมวลผลต่อไป ตัวอย่างเช่น ราคาหุ้นโดยทั่วไปจะมีการเปลี่ยนแปลงที่เร็วในหนึ่งชั่วโมงหรือหนึ่งวัน ดังนั้นอาจต้องทำการหาค่าราคาเฉลี่ยในช่วงหลายวันก่อนที่จะหาแนวโน้มของราคา

ในการที่จะพูดถึงนิยามทั่วไปของระบบแบบเอฟไออาร์ มาเริ่มต้นในการพิจารณาระบบเฉลี่ยค่าแบบเคลื่อนที่ เป็นตัวอย่างของระบบที่ประมวลผลค่าอินพุตซีเควนซ์เพื่อทำการสร้างเอาท์พุตซีเควนซ์ เพื่อให้เจาะจงมากขึ้น พิจารณาวิธีการหาค่าเฉลี่ยของค่าสามค่า (3-Point Average) ซึ่งก็คือแต่ละค่าเอาท์พุตซีเควนซ์คือผลรวมของสามค่าอินพุตซีเควนซ์ที่ติดกันแล้วหารด้วยสาม ถ้าใช้อัลกอริทึมนี้กับซีเควนซ์ที่เป็นรูปร่างสามเหลี่ยมดังแสดงในรูปที่ 2.18 สามารถคำนวณซีเควนซ์ใหม่ขึ้นมาเรียกว่า $y[n]$ ซึ่งก็คือเอาท์พุตของตัวดำเนินการที่ทำการหาค่าเฉลี่ย



รูปที่ 2.17 สัญญาณอินพุตที่มีความยาวจำกัด $x[n]$

ซีเควนซ์ในรูปที่ 2.17 นี้เป็นตัวอย่างสัญญาณที่มีความยาวจำกัด (Filter-Length Signal) ค่าเฉลี่ยของสามค่าของ $\{x[0], x[2]\} = \{2,4,6\}$ ให้ผลลัพธ์ $\frac{1}{3}(2+4+6) = 4$ ซึ่งเป็นค่าหนึ่งของเอาท์พุตซีเควนซ์ ค่าต่อไปได้จากการหาค่าเฉลี่ย $\{x[1], x[2], x[3]\} = \{4,6,4\}$ ให้ผลลัพธ์ 14/3 ก่อนที่จะทำต่อไปจากจุดนี้เราควรเลือกการใช้อินเด็กซ์ที่เหมาะสมสำหรับเอาท์พุต เช่น ค่า 4 และ 14/3 สามารถที่จะกำหนดให้เป็นค่าของ $y[0]$ และ $y[1]$ ได้ แต่การกำหนดในลักษณะนี้เป็นเพียงหนึ่งวิธี ในความเป็นไปได้มากมายหลายวิธี ถ้าใช้การกำหนดในลักษณะนี้สมการสำหรับการคำนวณเอาท์พุตจากอินพุตเป็นดังนี้คือ

$$y[0]=\frac{1}{3}\big(x[0]+x[1]+x[2]\big)$$
$$y[1]=\frac{1}{3}\big(x[1]+x[2]+x[3]\big)$$

ซึ่งสามารถเขียนให้อยู่ในรูปทั่วไปของสมการอินพุต-เอาท์พุตได้ดังนี้คือ

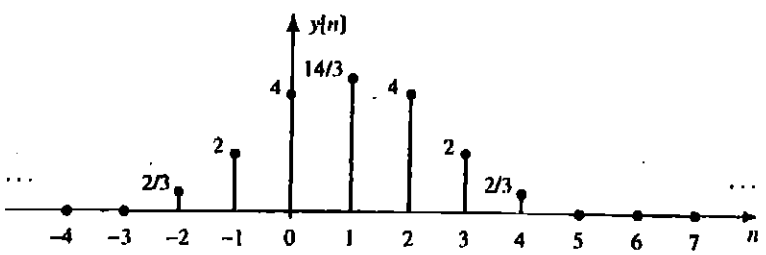
$$y[n]=\frac{1}{3}\big(x[n]+x[n+1]+x[n+2]\big) \qquad (2.49)$$

สมการ (2.49) เรียกว่า สมการดิฟเฟอเรนซ์ (Difference Equation) **ซึ่งเป็นการระบุการทำงานที่สมบูรณ์ของระบบแบบเอฟไออาร์ เนื่องจากสามารถใช้สมการ (2.49) ในการคำนวณหาสัญญาณเอาท์พุตทั้งหมดได้สำหรับค่าอินเด็กซ์ทุกๆ ค่า** $-\infty < n < \infty$ **สำหรับอินพุตแบบ** Triangular ดังแสดงในรูปที่ 2.17 จะได้ผลลัพธ์ $y[n]$ ดังแสดงในตารางดังนี้คือ

n	$n<-2$	-2	-1	0	1	2	3	4	5	$n>5$
$x[n]$	0	0	0	2	4	6	4	2	0	0
$y[n]$	0	$\frac{2}{3}$	2	4	$\frac{14}{3}$	4	2	$\frac{2}{3}$	0	0

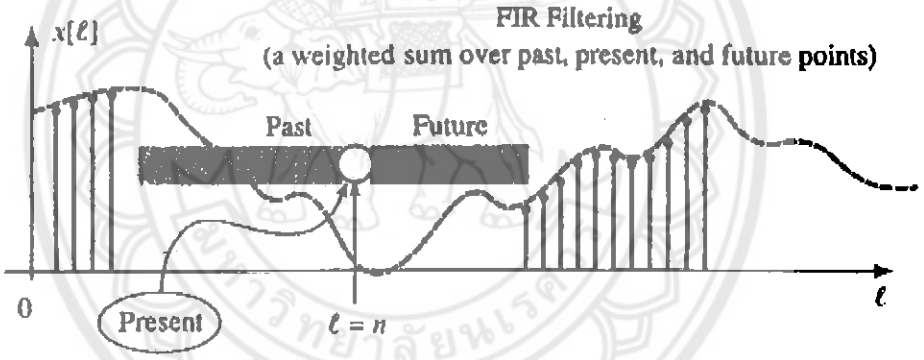
ตารางที่ 2.1 แสดงค่าผลลัพธ์ของ $y[n]$

สังเกตว่าค่าตัวเลขที่เป็นตัวหนาในแถวของ $x[n]$ **เป็นตัวเลขที่เกี่ยวข้องกับการคำนวณของ** $y[2]$ **และสังเกตด้วยว่า** $y[n]=0$ นอกช่วง $-2\le n\le 4$ **นั่นคือเอาท์พุตมีลักษณะ** Finite Support **เอาท์พุตซีเควนซ์แสดงในรูปที่ 2.18 สังเกตว่าเอาท์พุตซีเควนซ์ยาวกว่าอินพุตซีเควนซ์ และเอาท์พุตมีลักษณะสัญญาณที่มีการเปลี่ยนแปลงที่นุ่มนวลกว่าอินพุตซีเควนซ์ ซึ่งเป็นคุณลักษณะของเอฟไออาร์ฟิลเตอร์แบบเฉลี่ยค่าแบบเคลื่อนที่**



รูปที่ 2.18 เอาท์พุตของระบบเฉลี่ยค่า $y[n]$

การเลือกการกำหนดหมายเลขกำกับหรืออินเด็กซ์ของเอาท์พุตสามารถกำหนดอย่างไรก็ได้ แต่จะมีผลเมื่อพิจารณาถึงคุณสมบัติของฟิลเตอร์ ตัวอย่างเช่น ฟิลเตอร์ที่ถูกกำหนดโดยสมการที่ (2.49) จะมีคุณสมบัติที่ค่าเอาท์พุตจะเริ่มต้น (หรือมีค่าไม่ใช่ศูนย์) ก่อนการเริ่มต้นของอินพุต ลักษณะเช่นนี้เป็นลักษณะที่ไม่พึงประสงค์ถ้าสัญญาณอินพุตมาโดยตรงจากการแปลงสัญญาณแอนะล็อกเป็นดิจิตอล (A-to-D) ซึ่งเป็นลักษณะทั่วไป ในการประยุกต์ใช้งานทางด้านการประมวลผลสัญญาณเสียง ในกรณีนี้ n จะหมายถึงแกนของเวลา และสามารถแปลความหมาย $y[n]$ ในสมการ (2.49) เป็นการคำนวณของค่าเอาท์พุตปัจจุบันโดยการใช้ค่าอินพุตสามตัว เนื่องจากอินพุตเหล่านี้ถูกกำหนดค่าอินเด็กซ์ด้วย n, $n+1$ และ $n+2$ ค่าอินเด็กซ์ที่มีค่า $n+1$ และ $n+2$ นี้เป็นค่าอนาคต โดยทั่วไปแล้ว การคำนวณอาจใช้ค่าที่มาจากอดีตหรืออนาคต ดังแสดงในรูปที่ 2.19 ในกรณีของระบบเฉลี่ยค่าแบบเคลื่อนที่แบบ 3 พอยต์ นี้จะเห็นได้ว่าจะมี สไลดิงวินโดว์ (Sliding Window) ขนาดสามแซมเปิลเป็นตัวกำหนดว่าสามแซมเปิลใดที่จะถูกนำมาใช้ในการคำนวณ $y[n]$



รูปที่ 2.19 การคำนวณการเฉลี่ยค่าของฟิลเตอร์ที่เวลา n ใช้ค่าใน Sliding Window แถบสีเทาเข้มแสดงค่าอนาคต $(l > n)$ ส่วนแถบสีเทาอ่อนแสดงค่าอดีต $(l < n)$

ฟิลเตอร์ที่ใช้เฉพาะค่าอินพุตปัจจุบันและอดีตเรียกว่าฟิลเตอร์ชนิดคอซอล (Causal) ซึ่งมีความหมายว่า ผลลัพธ์ (Effect) จะไม่นำหน้าเหตุ (Cause) และฟิลเตอร์ที่ใช้ค่าอนาคตของอินพุต เรียกว่า นอนคอซอล (Noncausal) ระบบแบบนอนคอซอลจะไม่สามารถสร้างให้ทำงานได้ในการประยุกต์ใช้งานแบบเรียลไทม์ เนื่องจากค่าอินพุตยังไม่มีเมื่อจะต้องคำนวณค่าเอาท์พุต ส่วนในกรณีอื่นๆ เมื่อข้อมูลถูกเก็บไว้ในหน่วยความจำในคอมพิวเตอร์ล่วงหน้าแล้ว การที่ระบบจะเป็นแบบคอซอลหรือไม่คอซอล จึงไม่ใช่เรื่องสำคัญ

การกำหนดค่าอินเด็กซ์อีกรูปแบบหนึ่งที่จะทำให้ฟิลเตอร์เฉลี่ยค่าแบบ 3 พอยต์นี้เป็นแบบคอซอลคือการที่เอาท์พุต $y[n]$ เป็นการเฉลี่ยค่าอินพุตที่ n (ค่าปัจจุบัน) ที่ $n-1$ (ที่

แซมเปิลก่อนหน้า 1 แซมเปิล) และที่ $n-2$ (ที่แซมเปิลก่อนหน้า 2 แซมเปิล) สมการดิฟเฟอเรนซ์สำหรับฟิลเตอร์นี้คือ

$$y[n]=\frac{1}{3}\left(x[n]+x[n-1]+x[n-2]\right) \tag{2.50}$$

รูปแบบในสมการ (2.50) เป็นการเฉลี่ยค่าแบบเคลื่อนที่แบบคอซอลหรือเรียกอีกชื่อหนึ่งว่า การเฉลี่ยค่าแบบกลับหลัง (Backward Average) การใช้สมการดิฟเฟอร์เรนซ์ในสมการที่ (2.50) สามารถสร้างตารางของค่าเอาท์พุตทั้งหมดที่อยู่ในช่วง $-\infty < n < \infty$ สังเกตว่าค่าของ $x[n]$ ที่เป็นตัวหนา ถูกใช้ในการคำนวณ $y[4]$ แทนที่จะเป็น $y[2]$

n	$n<-2$	-2	-1	0	1	2	3	4	5	6	7	$n>7$
$x[n]$	0	0	0	2	4	6	4	2	0	0	0	0
$y[n]$	0	0	0	$\frac{2}{3}$	2	4	$\frac{14}{3}$	4	2	$\frac{2}{3}$	0	0

ตารางที่ 2.2 แสดงค่าผลลัพธ์ของ $y[n]$

ผลลัพธ์ $y[n]$ มีค่าเหมือนกันกับกรณีก่อนหน้านี้ แต่ช่วงค่าซัพพอร์ตในกรณีนี้อยู่ในช่วง $0 \le n \le 6$ สังเกตว่า เอาท์พุตของฟิลเตอร์แบบคอซอล มีลักษณะเหมือนกับผลลัพธ์เอาท์พุตที่ถูกเลื่อนทางเวลาของฟิลเตอร์นอนคอซอลก่อนหน้านี้ ดังนั้นค่าเอาท์พุตจะไม่เปลี่ยนไปจากศูนย์ก่อนที่อินพุตจะเปลี่ยนไปจากศูนย์

2.4 ไมโครคอนโทรลเลอร์

2.4.1 คุณสมบัติ MPLAB C Compiler

- เป็นไปตามมาตรฐานของ ANSI C ซึ่งประกอบด้วยไลบรารีที่มาตรฐาน เช่น ไลบรารีคณิตศาสตร์, หน่วยความจำ, การแปลงข้อมูล เป็นต้น
- รองรับการสร้างโค้ดในรูปแบบโมดูล เพื่อให้สามารถนำโค้ดกลับมาใช้ได้ใหม่
- สามารถออปติไมซ์โค้ดได้มากกว่า 30% ทำให้โค้ดโปรแกรมมีขนาดเล็ก
- สนับสนุนการเขียนโค้ดโปรแกรมแบบ In-line assembly คือสามารถแทรกคำสั่งภาษาแอสแซมบลีลงไปในโค้ดภาษาซี ได้ตามรูปแบบที่คอมไพเลอร์กำหนดไว้ เมื่อต้องการควบคุมชุดคำสั่งด้วยภาษาแอสแซมบลีโดยตรง
- มีไลบรารีสำหรับโมดูลต่างๆ ภายในตัวของ ไมโครคอนโทรลเลอร์ dsPIC เพื่อความรวดเร็วในการพัฒนาโปรแกรม

- แยกหน่วยความจำโค้ดและข้อมูลออกจากกันในตำแหน่งที่เฉพาะเจาะจงในหน่วยความจำหลัก (แอดเดรสหน่วยความจำจริง (Absolute Addresse))

- สนับสนุนรีจิสเตอร์แอกคูมิวเลเตอร์และมีฟังก์ชัน DSP ที่เขียนด้วยภาษาแอสแซมบลี ซึ่งสามารถเรียกใช้ผ่านทางภาษาซี ได้โดยตรง ทำให้ทำงานได้อย่างรวดเร็วเทียบเท่ากับภาษาแอสแซมบลี

- มีเวอร์ชันให้ทดลองใช้งานคอมไพเลอร์ได้ฟรี (Student Edition)

2.4.2 การใช้งานพอร์ตอินพุตเอาท์พุตดิจิตอล

ขาพอร์ตไมโครคอนโทรลเลอร์ dsPIC ทุกขา (ยกเว้น VDD, VSS, MCLR และ OSC1/CLKI) สามารถทำงานได้ 2 รูปแบบคือ

1. ขาพอร์ตโมดูลฟังก์ชันสำหรับการติดต่อกับอุปกรณ์ภายนอก (Peripherals) ทำหน้าที่เป็นขาพอร์ตอินพุตเอาท์พุตให้กับโมดูลฟังก์ชันที่เปิดใช้งาน เมื่อขาพอร์ตถูกโมดูลฟังก์ชันใช้งานแล้วจะไม่สามารถเป็นขาพอร์ต I/O ปกติได้ เช่น โมดูล SPI, I^2C, DCI, UART, CAN เป็นต้น

2. ขาพอร์ตสำหรับอินพุตเอาท์พุต (I/O Ports) ทำหน้าที่เป็นขาพอร์ตสำหรับรับและส่งสัญญาณ รวมถึงควบคุมการทำงานของอุปกรณ์ทั่วไปเป็นขาพอร์ตอินพุตเอาท์พุตดิจิตอล ซึ่งเป็นรูปแบบการทำงานพื้นฐานที่สุด เนื่องจากควบคุมการใช้งานขาพอร์ตเพียงแค่ 2 สถานะ คือ 1. ระดับสัญญาณลอจิก "1" หรือ High และ 2. ลอจิก "0" หรือ Low

2.4.2.1 รีจิสเตอร์ควบคุมการทำงานพอร์ต I/O

การควบคุมการทำงานของขาพอร์ตจะทำผ่านรีจิสเตอร์ 3 ตัวด้วยกันคือ รีจิสเตอร์

- TRISx (Data Direction register)
- PORTx (I/O Port register)
- LATx (I/O Latch register)

โดยรายละเอียดของรีจิสเตอร์เหล่านี้ได้มีการกำหนดไว้แล้วในเฮดเดอร์ไฟล์

ก. รีจิสเตอร์ TRIS

รีจิสเตอร์ TRISx ใช้ในการกำหนดทิศทางการทำงานของขาพอร์ตให้เป็นอินพุตหรือเอาท์พุตพอร์ต หากกำหนดค่า '1' ในบิตใดของรีจิสเตอร์ TRISx จะทำให้บิตที่ตำแหน่งเดียวกันของรีจิสเตอร์ PORTx และ LATx เป็นอินพุต และหากกำหนดค่า '0' จะเป็นเอาท์พุต รีจิสเตอร์ TRISx เช่น TRISA, TRISB, TRISC, TRISD เป็นต้น ตัวอย่างการกำหนดค่ารีจิสเตอร์ TRIS ดังตารางที่ 2.3 ดังนี้

SFR Name	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISB	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
ค่าที่กำหนดให้กับ TRISB	1	1	0	0	1	1
PORTB	RB5	RB4	RB3	RB2	RB1	RB0
LATB	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
สถานะของพอร์ต	Input	Input	Output	Output	Input	Input

ตารางที่ 2.3 การกำหนดค่าให้กับรีจิสเตอร์ TRISB และการใช้งานรีจิสเตอร์ PORTB, LATB

หมายเหตุ ข้อสังเกต 1 เทียบได้กับ I ซึ่งเท่ากับ Input และ 0 เทียบได้กับ O ซึ่งเท่ากับ Output

ข. รีจิสเตอร์ PORT

รีจิสเตอร์ PORTx ใช้ในการอ่านเขียนข้อมูลกับขาพอร์ต **การอ่านข้อมูลด้วยวิธีรีจิส**เตอร์ PORTx จะเป็นการอ่านข้อมูลจากขาพอร์ตโดยตรง (I/O pin) และ**หากเป็นการเขียนข้อมูลด้วย**รีจิสเตอร์ PORTx จะเป็นการเขียนข้อมูลไปที่แลตช์พอร์ตข้อมูล (port data latch) **รีจิสเตอร์** PORTx เช่น PORTA, PORTB, PORTC, PORTD ในบางชุดคำสั่งของ dsPIC เช่น**คำสั่ง** BSET และ BCLR เมื่อนำมาใช้กับรีจิสเตอร์ PORT ในโหมดการอ่านเขียนข้อมูล**ต่อเนื่อง อาจทำให้เกิดความผิดพลาดได้** เพราะการนำรีจิสเตอร์ PORT มาใช้ในกระบวนการ อ่าน-**แก้ไข-เขียน** (read-modify-write) นั้น ขาพอร์ตจะถูกกำหนดเป็นอินพุตและเอาท์พุตในเวลาที่**กระชั้นชิดเกินไป ส่งผลให้เกิดความผิดพลาดของข้อมูลได้**

ค. รีจิสเตอร์ LAT

รีจิสเตอร์ LATx ถูกนำมาใช้ ในการแก้ไขปัญหาของ**รีจิสเตอร์** PORTx ใน**กระบวนการ** อ่าน-แก้ไข-เขียนข้อมูลแบบต่อเนื่องกระชั้นชิด โดยการอ่าน**ข้อมูลด้วยรีจิสเตอร์** LATx **จะเป็นการอ่านข้อมูลที่ค้างอยู่ในแลตช์พอร์ตเอาท์พุต** (port output latches) **และการเขียนข้อมูลจะเป็นไปในรูปแบบเดียวกับรีจิสเตอร์** PORTx **คือ เขียนข้อมูลไปที่แลตช์พอร์ตข้อมูล** (port data latch) **ด้วยวิธีการนี้ทำให้ไม่เกิดความผิดพลาดในการอ่านเขียนข้อมูล**

สรุปความแตกต่างระหว่างรีจิสเตอร์ PORT **และ** LAT **มีรายละเอียดดังนี้**

- **การเขียนข้อมูลไปที่รีจิสเตอร์** PORTx **จะเขียนข้อมูลไปที่แลตช์พอร์ต**
- **การเขียนข้อมูลไปที่รีจิสเตอร์** LATx **จะเขียนข้อมูลไปที่แลตช์พอร์ต**
- **การอ่านข้อมูลจากรีจิสเตอร์** PORTx **จะเป็นการอ่านข้อมูลจาก** I/O **พอร์ตโดยตรง**
- **การอ่านข้อมูลจากรีจิสเตอร์** LATx **เป็นการอ่านข้อมูลซึ่งค้างอยู่ในแลตช์พอร์ต**

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

2.4.2.2 การใช้งานพอร์ต

เนื่องจากพอร์ตของ dsPIC สามารถทำหน้าที่ได้มากกว่าหนึ่งหน้าที่ ก่อนการใช้งานจึงต้องตรวจสอบว่าสถานะเริ่มต้นเมื่อไมโครคอนโทรลเลอร์เกิดการรีเซตนั้น แต่ละขาพอร์ตถูกกำหนดให้ทำหน้าที่ใดบ้าง และต้องไม่ลืมที่จะกำหนดสถานะพอร์ตว่าจะใช้งานเป็นอินพุตหรือเอาท์พุตด้วย โดยกำหนดผ่านทางรีจิสเตอร์ TRIS ก่อนที่จะใช้งาน สำหรับการกำหนดหน้าที่ใช้งานเป็นพอร์ตอินพุตเอาท์พุตดิจิตอลแนะนำดังนี้

1. กำหนดขาพอร์ตอินพุตหรือเอาท์พุตด้วยรีจิสเตอร์ TRIS
2. อ่านค่าข้อมูลจากพอร์ตหรือเป็นอินพุตพอร์ตให้ใช้รีจิสเตอร์ PORT
3. เขียนส่งค่าข้อมูลไปที่พอร์ตหรือเอาท์พุตพอร์ตให้ใช้รีจิสเตอร์ LAT หรือ PORT (แนะนำให้ใช้ LAT)
4. พอร์ตแอนะล็อกเมื่อต้องการใช้งาน เป็นพอร์ตอินพุตดิจิตอล ต้องทำการเซตบิตในรีจิสเตอร์ ADPCFG (ADC Port Configuration Register) โดยการเซตค่า '1' ที่รีจิสเตอร์ เพื่อปิด อินพุตแอนะล็อก ควบคู่กับการกำหนดรีจิสเตอร์ TRIS
5. หากมีการเปิดใช้งานโมดูลที่เกี่ยวข้องกับขาพอร์ต ขาพอร์ตนั้นจะถูกควบคุมด้วยโมดูล และไม่สามารถที่จะนำมาใช้งานเป็นอินพุตเอาท์พุตดิจิตอลได้

2.4.3 การทดลองใช้งานโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอลภายใน dsPIC เบื้องต้น

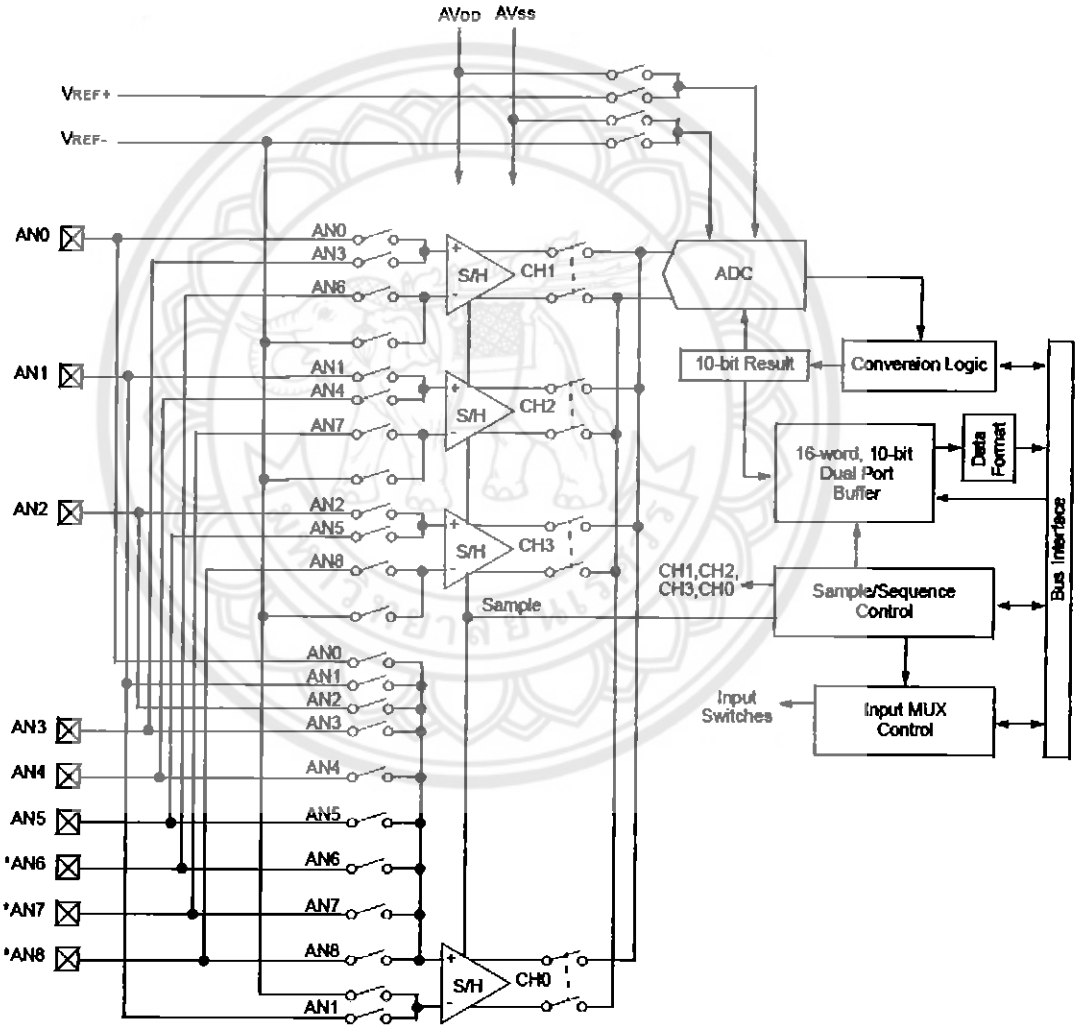
การเขียนโปรแกรมภาษา C เพื่อทดลองใช้งานโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอล (ADC) ภายในไมโครคอนโทรลเลอร์ dsPIC30F4011 ซึ่งมีการบรรจุโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่มีความละเอียด 10 บิตจำนวน 6 ช่อง ที่ความละเอียดของการแปลงข้อมูล 10 บิต ทำให้ได้ข้อมูลดิจิตอลมีค่าตั้งแต่ 0 ถึง 1023 (2^{10} หรือ 1,024 ค่า)

2.4.3.1 คุณสมบัติโดยสรุปของโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

- เป็นโมดูลที่ใช้แปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่มีความละเอียด 10 บิต จำนวน 6 ช่อง
- ใช้วิธีการแปลงสัญญาณแบบประมาณค่าหรือซักเซสซีฟ แอ็ปพร็อกซิเมชั่น (Successive Approximation)
- มีอัตราเร็วในการสุ่มสัญญาณสูงสุดที่ 500 กิโลแซมเปิลต่อวินาที (ksps) หรือ 500,000 จุดตัวอย่างต่อวินาที
- สามารถกำหนดให้ทำงานได้ขณะเข้าสู่โหมดสลีป (Sleep mode)
- สามารถกำหนดระดับแรงดันอ้างอิง ที่ได้ทั้งจากภายในผ่านทางขา AV_{DD} กับ AV_{SS} และภายนอกผ่านทางขา V_{REF+} และ V_{REF-}

2.4.3.2 การทำงานเบื้องต้นของโมดูล ADC ใน dsPIC30F4011

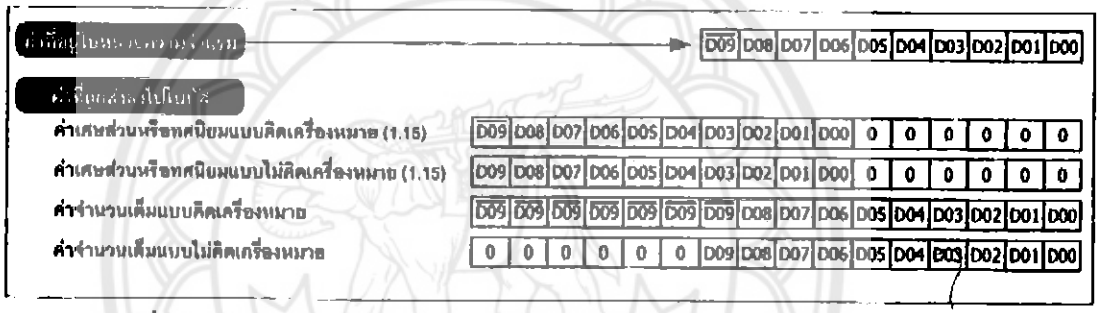
การทำงานของโมดูล ADC ใน dsPIC30F4011 ซึ่งมีขาพอร์ตอินพุตแอนะล็อกทั้งสิ้น 9 ขาคือ AN0-AN8 ดังรูปที่ 2.20 เป็นไดอะแกรม โดยมี 2 ขาที่สามารถใช้รับแรงดันอ้างอิงเพื่อขยายย่านของแรงดันอินพุต ภายในโมดูลมีวงจรสุ่มและเก็บค่าสัญญาณ (Sample and Hold : S/H) จำนวน 4 ชุด โดยทำงานร่วมกับส่วนควบคุมการมัลติเพล็กซ์สัญญาณอินพุต ทำให้สามารถจัดสรรวงจรสุ่มและเก็บค่าสัญญาณให้สามารถรองรับกับสัญญาณอินพุตแอนะล็อกทั้ง 6 ช่องได้ด้วยความเร็วสูงสุด



รูปที่ 2.20 ไดอะแกรมการทำงานอย่างง่ายของโมดูล ADC ในไมโครคอนโทรลเลอร์ dsPIC30F2010

สัญญาณที่ผ่านจากวงจรสุ่มและเก็บค่าสัญญาณ (Sample and Hold : S/H) จะถูกป้อนเข้าสู่วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอลแบบซักเซสซีฟ แอ็ปพรอกซิเมชั่น ขนาด 10 บิต ข้อมูลที่ได้จากการแปลงจะถูกพักไว้ในหน่วยความจำแรม จากนั้นจะได้รับการจัดรูปแบบตามที่ผู้พัฒนาโปรแกรมกำหนด ดังแสดงในรูปที่ 2.21 จากนั้นข้อมูลจะถูกถ่ายทอดลงบนบัสข้อมูลเพื่อส่งไปยังซีพียูต่อไป

อีกองค์ประกอบหนึ่งที่ทำให้โมดูล ADC สามารถแปลงสัญญาณได้รวดเร็วคือภายในโมดูล ADC มีบัฟเฟอร์ความจุ 16 เวิร์ด นั่นคือ สามารถรองรับข้อมูลที่ได้จากการแปลงสูงสุด 16 ชุดข้อมูล ดังนั้นเมื่อแปลงสัญญาณครั้งหนึ่งก็นำมาเก็บไว้ที่บัฟเฟอร์ หากบัฟเฟอร์ยังไม่เต็มก็สามารถกลับไปแปลงสัญญาณต่อได้ทันที โดยไม่ต้องรอให้การถ่ายทอดข้อมูลไปยังรีจิสเตอร์ที่ใช้เก็บค่าการแปลงเสร็จสิ้น



รูปที่ 2.21 รูปแบบของข้อมูลผลลัพธ์ที่ได้จากการแปลงสัญญาณของโมดูล ADC

1. รีจิสเตอร์หลักที่ใช้ในโมดูล ADC

ADCON1 (A/D Control Register 1) : รีจิสเตอร์ควบคุมโมดูล ADC ตัวที่ 1

บิต 15	บิต 14	บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8
AON	-	ASIDL	-	-	-	FORM1	FORM0
R/W -0	U -0	R/W -0	U -0	U -0	U -0	R/W -0	R/W -0
บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SSRC2	SSRC1	SSRC0	-	SIMSAM	ASAM	SAMP	DONE
R/W -0	R/W -0	R/W -0	U -0	R/W -0	R/W -0	R/W -0	R/C -0

บิต 15 – ADON (A/D Operating Mode bit): บิตเลือกให้โมดูล ADC ทำงาน

"1" = เลือกให้โมดูล ADC ทำงาน

"0" = ปิดการทำงานของโมดูล ADC

บิต 14 ไม่ใช้งาน กำหนดเป็น "0"

บิต 13 – ADSIDL (Stop in IDLE Mode bit): บิตกำหนดให้โมดูล ADC หยุดทำงานในโหมดไอเดิล

บิต 12 ถึง 10 ไม่ใช้งาน กำหนดเป็น "0"

บิต 9 และ 8 – FROM1 และ FROM0 (Data Output Format bits): บิตเลือกรูปแบบข้อมูลเอาท์พุต

บิต 7 ถึง 5 – SSRC2 ถึง SSRC0 (Conversion Trigger Source Select bits): บิตเลือกแหล่งกำเนิดสัญญาณกระตุ้นให้โมดูล ADC แปลงสัญญาณ

"000" = เลือกให้กระตุ้นเมื่อเกิดการเคลียร์บิต SAMP

"001" = เลือกให้กระตุ้นเมื่อเกิดการเปลี่ยนแปลงสัญญาณที่ขา INT0

"010" = เลือกให้กระตุ้นเมื่อการเปรียบเทียบข้อมูลในไทเมอร์ 3 เสร็จสิ้น

"011" = เลือกให้กระตุ้นเมื่อเสร็จสิ้นการสุ่มสัญญาณจากโมดูล PWM ควบคุมมอเตอร์

"100" = สำรองไว้

"101" = สำรองไว้

"110" = สำรองไว้

"111" = เลือกให้กระตุ้นเมื่อตัวนับค่าภายในเสร็จสิ้นการสุ่มสัญญาณและเริ่มต้นการแปลงสัญญาณ เป็นการกำหนดให้แปลงสัญญาณโดยอัตโนมัติ

บิต 4 ไม่ใช้งาน กำหนดเป็น "0"

บิต 3 – SIMSAM (Simultaneous Sample Select bit): บิตเลือกการสุ่มสัญญาณแบบทันทีทันใด จะใช้ก็ต่อเมื่อบิต CHPS = "01", "10", หรือ "11"

"0" = สุ่มสัญญาณเรียงลำดับตามหมายเลขช่องสัญญาณ

ถ้าบิต CHPS = "10" หรือ "11"

"1" = เลือกสุ่มสัญญาณจากวงจรสุ่มและเก็บค่าสัญญาณช่อง CH0, CH1, CH2, CH3 ทันที

ถ้าบิต CHPS = "01"

"1" = เลือกสุ่มสัญญาณจากวงจรสุ่มและเก็บค่าสัญญาณช่อง CH0 และ CH1 ทันที

บิต 2 – ASAM (A/D Sample Auto-Start bit): บิตกำหนดการเริ่มต้นสุ่มสัญญาณอัตโนมัติ

"0" = เริ่มสุ่มสัญญาณเมื่อบิต SAMP ถูกเซตเป็น "1"

"1" = เริ่มสุ่มสัญญาณทันทีที่การแปลงสัญญาณครั้งล่าสุดเสร็จสิ้นลง ทำให้บิต SAMP เซตอัตโนมัติ

บิต 1 – SAMP (A/D Sample Enable bit): บิตเอ็นเอเบิลการสุ่มสัญญาณของโมดูล ADC

"0" = เลือกให้พักการทำงานของวงจรสุ่มและเก็บค่าสัญญาณหรือดิสเอเบิลการสุ่มสัญญาณ

"1" = เลือกให้วงจรสุ่มและเก็บค่าสัญญาณอย่างน้อยหนึ่งวงจรทำการสุ่มสัญญาณหรือเอ็นเอเบิลให้เกิดการสุ่มสัญญาณ

- เมื่อบิต ASAM = "0" การเขียนข้อมูล "1" มาที่บิตนี้จะเป็นการกระตุ้นให้เริ่มสุ่มสัญญาณ

- เมื่อบิต SSRC = "000" คือการเขียนข้อมูล "0" มาที่บิตนี้ ซึ่งจะเป็นการกำหนดให้หยุดสุ่มสัญญาณแล้วเริ่มต้นกระบวนการแปลงสัญญาณ

บิต 0 – DONE (A/D Conversion Status bit): บิตแสดงสถานะ การแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

"0" = การแปลงสัญญาณยังไม่เสร็จสิ้น

"1" = การแปลงสัญญาณเสร็จสิ้น

บิตนี้สามารถเคลียร์ได้ด้วยกระบวนการทางซอฟต์แวร์ หรือเมื่อเริ่มต้นการแปลงสัญญาณในรอบใหม่ขึ้น นอกจากนั้นการเคลียร์บิตนี้จะสามารถไม่กระทบต่อการทำงานของวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอลแต่อย่างใด

ADCON2 (A/D Control Register 2): รีจิสเตอร์ควบคุมโมดูล ADC ตัวที่ 2

บิต 15	บิต 14	บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8
VDFG2	VCFG1	VCFG0	-	-	CSCNA	CHPS1	CHPS0
R/W -0	U -0	R/W -0	U -0	U -0	R/W -0	R/W -0	R/W -0
บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
BUFS	-	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
R/W -0	U -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0

บิต 15 ถึง 13 – VCFG2 ถึง VCFG0 (Voltage Reference Configuration bits): บิตกำหนดแรงดันอ้างอิงสำหรับใช้ในโมดูล ADC

ค่าของข้อมูล	แรงดันอ้างอิงด้านสูง V_{REFH}	แรงดันอ้างอิงด้านต่ำ V_{REFL}
"000"	จากขา AVDD (ปกติมีค่า +5V)	จากขา AVSS (ปกติต่อลงกราวด์)
"001"	จากภายนอกผ่านขา VREF+	จากขา AVSS (ปกติต่อลงกราวด์)
"010"	จากขา AVDD (ปกติมีค่า +5V)	จากภายนอกผ่านขา VREF-
"011"	จากภายนอกผ่านขา VREF+	จากภายนอกผ่านขา VREF-
"1xx"	จากขา AVDD (ปกติมีค่า +5V)	จากขา AVSS (ปกติต่อลงกราวด์)

บิต 12 และ 11 ไม่ใช้งาน กำหนดเป็น "0"

บิต 10 – CSCNA (Scan Input Selections for CH0+ S/H Input for MUX A Input Multiplexer Setting bit): บิตเลือกการสแกนช่องสัญญาณของวงจรสุ่มและเก็บค่าสัญญาณผ่านมัลติเพล็กเซอร์ A

"0" = ไม่มีการสแกนอินพุต

"1" = กำหนดให้สแกนอินพุต

บิต 9 และ 8 – CHPS1 และ CHPS0 (Selects Channels Utilized bits): บิตเลือกกลุ่มของช่องสัญญาณผ่านทางวงจรสุ่มและเก็บค่าสัญญาณ

"00" = ต้องการแปลงวงจรสุ่มและเก็บค่าสัญญาณช่อง CH0

"01" = ต้องการแปลงวงจรสุ่มและเก็บค่าสัญญาณช่อง CH0 และ CH1

"1x" = ต้องการแปลงวงจรสุ่มและเก็บค่าสัญญาณช่อง CH0, CH1, CH2 และ CH3

เมื่อบิต SIMSAM (บิต 3 ในรีจิสเตอร์ ADCON1) = "0" จะมีการสุ่มสัญญาณจากทุกช่อง

เมื่อบิต SMSAM = "1" การเลือกช่องสัญญาณจะกระทำผ่านบิต CHPS1 และ CHPS0

บิต 7 – BUFS (Buffer Fill Status bit): บิตแสดงสถานะบัฟเฟอร์

จะมีการแสดงผลเกิดขึ้นเมื่อบิต BUFM ถูกเซตเป็น "1" (รีจิสเตอร์ ADRES แบ่งเป็น 2 ส่วน ส่วนละ 8 เวิร์ด)

"0" = แจ้งว่า ขณะนี้โมดูล ADC กำลังเขียนข้อมูลลงในบัฟเฟอร์ที่ตำแหน่ง 0x0-0x7 ผู้พัฒนาโปรแกรมสามารถเข้าถึงข้อมูลในบัฟเฟอร์ที่ตำแหน่ง 0x8-0xF ได้

"1" = แจ้งว่า ขณะนี้โมดูล ADC กำลังเขียนข้อมูลลงในบัฟเฟอร์ที่ตำแหน่ง 0x8-0xF ผู้พัฒนาโปรแกรมสามารถเข้าถึงข้อมูลในบัฟเฟอร์ที่ตำแหน่ง 0x0-0x7 ได้

บิต 6 ไม่ใช้งาน กำหนดเป็น "0"

บิต 5 ถึง 2 – SMPI3 ถึง SMPI0 (Sample/Convert Sequences Per Interrupt Selection bits): บิตเลือกการเกิดอินเตอร์รัปต์ในกระบวนการสุ่มและแปลงสัญญาณในโมดูล ADC

"0000" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกลำดับของกระบวนการสุ่มและแปลงสัญญาณ

"0001" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 2 ของกระบวนการสุ่มและแปลงสัญญาณ

"0010" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 3 ของกระบวนการสุ่มและแปลงสัญญาณ

"0011" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 4 ของกระบวนการสุ่มและแปลงสัญญาณ

"0100" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 5 ของกระบวนการสุ่มและแปลงสัญญาณ

"0101" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 6 ของกระบวนการสุ่มและแปลงสัญญาณ

"0110" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 7 ของกระบวนการสุ่มและแปลงสัญญาณ

"0111" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 8 ของกระบวนการสุ่มและแปลงสัญญาณ

"1000" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 9 ของกระบวนการสุ่มและแปลงสัญญาณ

"1001" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 10 ของกระบวนการสุ่มและแปลงสัญญาณ

"1010" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 11 ของกระบวนการสุ่มและแปลงสัญญาณ

"1011" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 12 ของกระบวนการสุ่มและแปลงสัญญาณ

"1100" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 13 ของกระบวนการสุ่มและแปลงสัญญาณ

"1101" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 14 ของกระบวนการสุ่มและแปลงสัญญาณ

"1110" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 15 ของกระบวนการสุ่มและแปลงสัญญาณ

"1111" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกๆ ลำดับที่ 16 ของกระบวนการสุ่มและแปลงสัญญาณ

บิต 1 – BUFM (Buffer Mode Select bit): บิตเลือกโหมดของบัฟเฟอร์

"0" = กำหนดให้บัฟเฟอร์มีความจุ 16 เวิร์ด มีชื่อเป็น ADCBUF0 ถึง ADCBUF15

"1" = แบ่งบัฟเฟอร์เป็น 2 ส่วน ส่วนละ 8 เวิร์ด คือ รีจิสเตอร์ ADCBUF8 ถึง 15 และรีจิสเตอร์ ADCBUF7 ถึง ADCBUF0

บิต 0 – ALTS (Alternate Input Sample Mode Select bit): บิตเลือกโหมดการทำงานของอินพุตมัลติเพล็กเซอร์

"1" = เลือกใช้อินพุต A สำหรับการสุ่มสัญญาณครั้งแรก จากนั้นสลับกันระหว่างอินพุต A และ B

"0" = เลือกใช้อินพุต A ตลอดการทำงาน

ADCON3 (A/D Control Register 3): รีจิสเตอร์ควบคุมโมดูล ADC ตัวที่ 3

บิต 15	บิต 14	บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8
-	-	-	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
U -0	U -0	U -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
บิต 7	**บิต 6**	**บิต 5**	**บิต 4**	**บิต 3**	**บิต 2**	**บิต 1**	**บิต 0**
ADRC	-	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
R/W -0	U -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0

บิต 15 ถึง 13 ไม่ใช้งาน กำหนดเป็น "0"

บิต 12 ถึง 8 – SAMC4 ถึง SAMC0 (Auto-Sample Time bits): บิตเลือกค่าเวลาในการสุ่มสัญญาณอัตโนมัติ

"00000" = 0 T_{AD} (กำหนดได้ในกรณีที่เลือกใช้วงจร S/H มากกว่าหนึ่งวงจร)

"00001" = 1 T_{AD}

"00010" = 2 T_{AD}

.....

"11111" = 31 T_{AD}

บิต 7 – ADRC (A/D Conversion Clock Source bit): บิตเลือกแหล่งกำเนิดสัญญาณนาฬิกาสำหรับการแปลงสัญญาณ

"0" = ใช้จากสัญญาณนาฬิกาหลักของระบบ

"1" = ใช้จากวงจร RC ภายในโมดูล ADC

บิต 6 ไม่ใช้งาน กำหนดเป็น "0"

บิต 5 ถึง 0 – ADCS5 ถึง ADCS0 (A/D Conversion Clock Select bits): บิตเลือกค่าเวลาที่ใช้ในการแปลงสัญญาณ

"000000" = 0.5T_{CY} x (000000+1) = 0.5T_{CY}

"000001" = 0.5T_{CY} x (000001+1) = T_{CY}

.....

"111111" = 0.5T_{CY} x (111111+1) = 32 x 0.5T_{CY}

สัญญาณนาฬิกาของ A/D	บิตเลือกสัญญาณนาฬิกา		ค่าเวลาในการแปลงสัญญาณของโมดูล ADC (T_{AD}) ที่ความถี่สัญญาณนาฬิกาภายในค่าต่างๆ				
	ADRC	ADCS5-ADCS0	F_{CY} = 30MHz	F_{CY} = 30MHz	F_{CY} = 12.5MHz	F_{CY} = 6.25MHz	F_{CY} = 1MHz
$T_{CY}/2$	0	000000	16.67 นาโนวินาที	20 นาโนวินาที	40 นาโนวินาที	80 นาโนวินาที	500นาโนวินาที
$T_{CY}/2$	0	000001	33.33 นาโนวินาที	40 นาโนวินาที	80 นาโนวินาที	160 นาโนวินาที	1 ไมโครวินาที
$T_{CY}/2$	0	000011	66.66 นาโนวินาที	80 นาโนวินาที	160 นาโนวินาที	320 นาโนวินาที	2 ไมโครวินาที
$T_{CY}/2$	0	000111	133.32 นาโนวินาที	160 นาโนวินาที	320 นาโนวินาที	640 นาโนวินาที	4 ไมโครวินาที
$T_{CY}/2$	0	001111	266.64 นาโนวินาที	320 นาโนวินาที	640 นาโนวินาที	1.28ไมโครวินาที	8 ไมโครวินาที
$T_{CY}/2$	0	011111	533.28 นาโนวินาที	640 นาโนวินาที	1.28 ไมโครวินาที	2.56ไมโครวินาที	16 ไมโครวินาที
$T_{CY}/2$	0	111111	1066.56นาโนวินาที	1280 นาโนวินาที	2.56 ไมโครวินาที	5.12ไมโครวินาที	32 ไมโครวินาที
$T_{CY}/2$	0	xxxxxx	200-400นาโนวินาที	200-400นาโนวินาที	200-400นาโนวินาที	200-400นาโนวินาที	200-400นาโนวินาที

ตารางที่ 2.4 แสดงการกำหนดค่าเวลาที่ใช้ในการแปลงสัญญาณในโมดูล ADC

ADCHS (A/D Input Select Register): รีจิสเตอร์เลือกช่องของวงจร S/H ที่ต่อกับขาพอร์ตอินพุตแอนะล็อกที่ต้องการแปลงสัญญาณ

บิต 15	บิต 14	บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8
CH123NB1	CH123NB0	CH123SB	CH0NB	CH0SB3	CH0SB2	CH0SB1	CH0SB0
R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
CH123NA1	CH123NA0	CH123SA	CH0NA	CH0SA3	CH0SA2	CH0SA1	CH0SA0
R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0

บิต 15 และ 14 – CH123NB1 และ CH123NB0 (Channel 1, 2, 3 Negative Input Select for MUX B Multiplexer Setting bits): บิตเลือกช่องอินพุตลบของวงจร S/H ช่อง CH1, CH2 และ CH3 สำหรับมัลติเพล็กเซอร์ B

"11" = อินพุตลบของ CH1 ต่อกับอินพุต AN9, อินพุตลบของ CH2 ต่อกับอินพุต AN10, อินพุตลบของ CH3 ต่อกับอินพุต AN11

"10" = อินพุตลบของ CH1 ต่อกับอินพุต AN6, อินพุตลบของ CH2 ต่อกับอินพุต AN7, อินพุตลบของ CH3 ต่อกับอินพุต AN8

"00" และ "01" = อินพุตลบของ CH1, CH2 และ CH3 ต่อกับ VREF- ให้ใช้รูปที่ 8-1 ประกอบ

บิต 13 – CH123SB (Channel 1, 2, 3 Positive Input Select for MUX B Multiplexer Setting bit): บิตเลือกช่องอินพุตบวกของวงจร S/H ช่อง CH1, CH2 และ CH3 สำหรับมัลติเพล็กเซอร์ B

"1" = CH1 positive input is AN3, CH2 positive input is AN4, CH3 positive input is AN5

"0" = อินพุตบวกของ CH1 ต่อกับอินพุต AN0, อินพุตบวกของ CH2 ต่อกับอินพุต AN1, อินพุตบวกของ CH3 ต่อกับอินพุต AN2

บิต 12 – CH0NB (Channel 0 Negative Input Select for MUX B Multiplexer Setting bit): บิตเลือกช่องอินพุตลบของวงจร S&H ช่อง CH0 สำหรับมัลติเพล็กเซอร์ B

"0" = อินพุตลบของ CH0 ต่อกับ VREF-

"1" = อินพุตลบของ CH0 ต่อกับ AN1

บิต 11 ถึง 8 CH0SB3 ถึง CH0SB0 (Channel 0 Positive Input Select for MUX B Multiplexer Setting bits): บิตเลือกช่องอินพุตบวกของวงจร S/H ช่อง CH0 สำหรับมัลติเพล็กเซอร์ B

"0000" = อินพุตบวกของ CH0 ต่อกับขา AN0

"0001" = อินพุตบวกของ CH0 ต่อกับขา AN1

.....

"1110" = อินพุตบวกของ CH0 ต่อกับขา AN14

"1111" = อินพุตบวกของ CH0 ต่อกับขา AN15

บิต 7 และ 6 – CH123NA1 และ CH123NA0 (Channel 1, 2, 3 Negative Input Select for MUX A Multiplexer Setting bits): บิตเลือกช่องอินพุตลบของวงจร S&H ช่อง CH1, CH2 และ CH3 สำหรับมัลติเพล็กเซอร์ A

"11" = อินพุตลบของ CH1 ต่อกับอินพุต AN9, อินพุตลบของ CH2 ต่อกับอินพุต AN10, อินพุตลบของ CH3 ต่อกับอินพุต AN11

"10" = อินพุตลบของ CH1 ต่อกับอินพุต AN6, อินพุตลบของ CH2 ต่อกับอินพุต AN7, อินพุตลบของ CH3 ต่อกับอินพุต AN8

"00" และ "01" = อินพุตลบของ CH1, CH2 และ CH3 ต่อกับ VREF-

บิต 5 – CH123SA (Channel 1, 2, 3 Positive Input Select for MUX A Multiplexer Setting bit): บิตเลือกช่องอินพุตบวกของวงจร S&H ช่อง CH1, CH2 และ CH3 สำหรับมัลติเพล็กเซอร์ A

"1" = CH1 positive input is AN3, CH2 positive input is AN4, CH3 positive input is AN5

"0" = อินพุตบวกของ CH1 ต่อกับอินพุต AN0, อินพุตบวกของ CH2 ต่อกับอินพุต AN1, อินพุตบวกของ CH3 ต่อกับอินพุต AN2

บิต 4 – CH0NA (Channel 0 Negative Input Select for MUX A Multiplexer Setting bit): บิตเลือกช่องอินพุตลบของวงจร S&H ช่อง CH0 สำหรับมัลติเพล็กเซอร์ A

"0" = อินพุตลบของ CH0 ต่อกับ VREF-

"1" = อินพุตลบของ CH0 ต่อกับ AN1

บิต 3 ถึง 0 – CH0SA3 ถึง CH0SA0 (Channel 0 Positive Input Select for MUX A Multiplexer Setting bits): บิตเลือกช่องอินพุตบวกของวงจร S&H ช่อง CH0 สำหรับมัลติเพล็กเซอร์ A

"0000" = อินพุตบวกของ CH0 ต่อกับขา AN0

"0001" = อินพุตบวกของ CH0 ต่อกับขา AN1

.....

"1101" = อินพุตบวกของ CH0 ต่อกับขา AN13

"1110" = อินพุตบวกของ CH0 ต่อกับขา AN14

"1111" = อินพุตบวกของ CH0 ต่อกับขา AN15

หมายเหตุ ส่วนอินพุตมัลติเพล็กเซอร์สามารถกำหนดการทำงานทางอินพุตได้ 2 รูปแบบ คือ มัลติเพล็กเซอร์ A (MUX A) และมัลติเพล็กเซอร์ B (MUX B) โดยบิต 15 ถึง 8 ของรีจิสเตอร์ ADCHS ใช้ตั้งค่าสำหรับ MUX B ส่วนบิต 7 ถึง 0 ของรีจิสเตอร์ ADCHS ใช้ตั้งค่าสำหรับ MUX A

ADPCFG (A/D Port Configutation Register): รีจิสเตอร์กำหนดค่าทางฮาร์ดแวร์ของพอร์ตอินพุตแอนะล็อก

บิต 15	บิต 14	บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0

รีจิสเตอร์ตัวนี้สามารถรองรับการกำหนดค่าของพอร์ตอินพุตแอนะล็อกได้ครบทั้ง 16 ช่อง (กรณีใช้ dsPIC เบอร์ใหญ่ที่มีอินพุตแอนะล็อกครบ 16 ช่อง) โดยการกำหนดนี้จะอย่างกันอย่างอิสระ เริ่มจากบิต 15 (PCFG15) สำหรับกำหนดค่าของอินพุต AN15 ไล่ไปตามลำดับจนถึงบิต 0 (PCFG0) สำหรับกำหนดค่าของอินพุต AN0 สำหรับ dsPIC30F2010 มี 6 ช่อง จึงใช้งานเพียง 6 บิตคือ PCFG0-PCFG5

"0" = กำหนดให้อินพุตแอนะล็อกทำงานในโหมดแอนะล็อก

"1" = กำหนดให้อินพุตแอนะบ็อกทำงานในโหมดดิจิตอล

ADCSSL (A/D Input Scan Select Register): รีจิสเตอร์เลือกช่องอินพุตที่ต้องการแปลงสัญญาณ

บิต 15	บิต 14	บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0

รีจิสเตอร์ตัวนี้ใช้กำหนดให้วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอลมาอ่านค่าจากอินพุตแบบเรียงลำดับอัตโนมัติ ซึ่งสามารถเลือกได้ว่า ต้องการให้อ่านค่าจากช่องใดบ้าง การกำหนดค่าของรีจิสเตอร์ตัวนี้สามารถรองรับอินพุตแอนะล็อกได้ครบทั้ง 16 ช่อง (กรณีใช้ dsPIC เบอร์ใหญ่ที่มีอินพุตแอนะล็อกครบ 16 ช่อง) โดยการกำหนดนี้จะอย่างกันอย่างอิสระ เริ่มจากบิต 15 (CSSL15) สำหรับกำหนดค่าของอินพุต AN15 ไล่ไปตามลำดับจนถึงบิต 0 (CSSL0) สำหรับกำหนดค่าของอินพุต AN0 สำหรับ dsPIC30F2010 มี 6 ช่อง จึงใช้งานเพียง 6 บิต คือ CSSL0-CSSL5

"0" = ไม่อ่านค่าจากอินพุตนี้

"1" = เลือกให้อ่านค่าอินพุต

2. บัฟเฟอร์เก็บผลลัพธ์จากการแปลงสัญญาณของโมดูล ADC

มีชื่อเรียกว่า ADCBUF เป็นหน่วยความจำแรมขนาด 16 บิต มีทั้งสิ้น 16 ตัว หรืออาจกล่าวได้ว่า เป็นบัฟเฟอร์ 16 เวิร์ด จึงสามารถกำหนดชื่อเรียกได้เป็น ADCBUF0, ADCBUF1, ADCBUF2, ..., ADCBUFE,

ADCBUFF ใช้สำหรับเก็บค่าผลลัพธ์ที่ได้จากการแปลงสัญญาณของวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอลชั่วคราว ก่อนที่จะส่งต่อไปจัดรูปแบบข้อมูล แล้วถ่ายทอดต่อไปยังบัสของระบบต่อไป

3. การกำหนดค่าเพื่อใช้งานโมดูล ADC

มีขั้นตอนโดยสรุปดังนี้

1. ตั้งค่าของโมดูล ADC

1.1 เลือกขาพอร์ตให้ทำงานเป็นอินพุตแอนะล็อกที่รีจิสเตอร์ ADPCFG

1.2 เลือกแหล่งจ่ายแรงดันอ้างอิงให้เหมาะสมกับย่านแรงดันแอนะล็อกทางอินพุตที่บิต 15 ถึง 13 ของรีจิสเตอร์ ADCON2

1.3 เลือกสัญญาณนาฬิกาสำหรับการใช้แปลงสัญญาณที่บิต 5 ถึง 0 ของรีจิสเตอร์ ADCON3

1.4 กำหนดจำนวนช่องของวงจรสุ่มสัญญาณและเก็บค่าของสัญญาณ ที่ต้องใช้ที่บิต 9 และบิต 8 ของรีจิสเตอร์ ADCON2

1.5 กำหนดวิธีการสุ่มสัญญาณของบิตที่ 3 ของรีจิสเตอร์ ADCON1 และรีจิสเตอร์ ADCSSL

1.6 กำหนดจำนวนอินพุตที่จะต้องใช้ทำงานร่วมกับวงจรสุ่มและเก็บค่าของสัญญาณ (S&H) ที่รีจิสเตอร์ ADCHS

1.7 เลือกลำดับการสุ่มสัญญาณและแปลงสัญญาณที่บิต 7 ถึงบิต 0 ของรีจิสเตอร์ ADCON1 และ บิต 12 ถึง 8 ของรีจิสเตอร์ ADCON3

1.8 เลือกรูปแบบของผลลัพธ์ที่ต้องการที่บิต 9 และบิต 8 ของรีจิสเตอร์ ADCON1

1.9 เลือกการอินเตอร์รัปต์ที่บิต 9 ถึง 5 ของรีจิสเตอร์ ADCON2

1.10 เปิดการทำงานของโมดูล ADC ที่บิต 15 ของรีจิสเตอร์ ADCON1

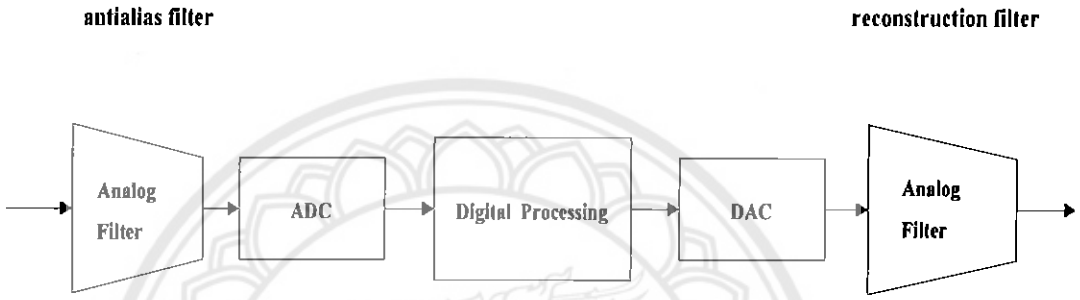
2. กำหนดการอินเตอร์รัปต์ (ถ้าต้องการ)

2.1 เคลียร์บิต ADIF

2.2 เลือกระดับความสำคัญของการอินเตอร์รัปต์

2.4.4 การใช้งานฟังก์ชันไลบรารี DSP

คุณสมบัติของไมโครคอนโทรลเลอร์ dsPIC ที่แตกต่างจากไมโครคอนโทรลเลอร์ PIC ทั่วไป นอกจากจะมีขนาด 16 บิตแล้ว คือ ความสามารถทางด้านการประมวลผลสัญญาณดิจิตอล (Digital Signal Processing) หรือที่เรียกว่า DSP จากรูปที่ 2.22 แสดงการใช้งานหน่วยประมวลผลสัญญาณดิจิตอลในรูปแบบหนึ่งที่ประกอบไปด้วย ภาคแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ภาคประมวลผลสัญญาณดิจิตอล (Digital Processing) และภาคแปลงสัญญาณดิจิตอลเป็นแอนะล็อก



รูปที่ 2.22 บล็อกไดอะแกรม ระบบการประมวลผลสัญญาณดิจิตอล

จากรูปที่ 2.22 ภาคแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ประกอบไปด้วย ชุดกรองสัญญาณ (Analog Filter) โดยจะส่งต่อสัญญาณให้ส่วนแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ที่จะทำหน้าที่ในการรับและแปลงสัญญาณแอนะล็อกเป็นดิจิตอล จากนั้นจะส่งข้อมูลเข้าสู่ ภาคประมวลผล ซึ่งจะทำหน้าที่ในการประมวลผลสัญญาณดิจิตอลตามความต้องการที่ระบบได้สร้างไว้แล้ว หลังจากที่ประมวลผลเสร็จสิ้น จะส่งข้อมูลไปที่ภาคแปลงสัญญาดิจิตอลเป็นแอนะล็อก แล้วผ่านการกรองสัญญาณอีกครั้ง จะได้สัญญาณแอนะล็อกในรูปแบบใหม่ตามความต้องการของระบบที่เกิดจากการประมวลผลสัญญาณดิจิตอล

ไมโครคอนโทรลเลอร์ dsPIC ได้เตรียมคุณสมบัติการประมวลผลสัญญาณดิจิตอลทั้งทางด้านฮาร์ดแวร์และทางด้านซอฟต์แวร์ไว้ให้พร้อมแล้ว ในด้านซอฟต์แวร์นั้น เครื่องมือพัฒนาโปรแกรม MPLAB + MPLAB C30 ได้เตรียมไลบรารีที่เกี่ยวกับการประมวลผลสัญญาณดิจิตอลไว้แล้วใน "DSP Library" ทั้งในส่วนของภาษาซีและภาษาแอสแซมบลี โดยประกอบไปด้วยฟังก์ชันต่อไปนี้

2.4.2.1 ฟังก์ชันเวกเตอร์

ฟังก์ชันเวกเตอร์ (Vector Functions) (เวกเตอร์ คือ ปริมาณที่มีทั้งขนาดและทิศทาง) รวบรวมฟังก์ชันที่ใช้ในการคำนวณเกี่ยวกับเวกเตอร์ เช่น การบวก, ลบ, คูณ, หาร, การคอนโวลูชัน (convolution), การคอเรเลชัน (correlation) ของเวกเตอร์ เป็นต้น

2.4.2.2 ฟังก์ชันแบบหน้าต่าง

ฟังก์ชันแบบหน้าต่าง (Window Function) เป็นการวิเคราะห์สัญญาณพื้นฐานวิธีหนึ่งที่ใช้ในการหาค่าสัมประสิทธิ์ของตัวกรอง (filter) โดยกำหนดกรอบสัญญาณเป็นช่วงสั้น ๆ ที่เท่ากัน โดยการคูณสัญญาณด้วยฟังก์ชันหน้าต่าง สัญญาณช่วงสั้นที่ได้ สามารถนำไปวิเคราะห์ต่อได้ เช่น ฟังก์ชันหน้าต่างบาเรต (Barlett window), แบล็กแมน (Blackman window), แฮมมิ่ง (Hamming), ฮานนิ่ง (Hanning window) และไคเซอร์ (Kaiser window) เป็นต้น

2.4.2.3 ฟังก์ชันเมตริกซ์

ฟังก์ชันเมตริกซ์ (Matrix Functions) (เมตริกซ์ คือ การจัดการข้อมูลในลักษณะแถวและหลัก) รวบรวมฟังก์ชันที่ใช้ในการคำนวณเกี่ยวกับเมตริกซ์ เช่น การบวก, ลบ, คูณ, ทรานสโพส (Transposes), อินเวอร์ส (Inverse) เมตริกซ์ เป็นต้น

2.4.2.4 ฟังก์ชันการกรองสัญญาณ

ฟังก์ชันการกรองสัญญาณ (Filtering Functions) ถูกใช้งานด้วยจุดประสงค์ 2 ประการหลัก คือ ใช้ในการแยก (Separation) สัญญาณดิจิตอลที่มีการรวมกันอยู่และแก้ไข (Restoration) สัญญาณดิจิตอลที่มีความผิดเพี้ยนในทางใดทางหนึ่ง ฟังก์ชันกรองสัญญาณ เช่น ผลตอบสนองอิมพัลส์จำนวนจำกัด (Finite Impulse Response (FIR)), ผลตอบสนองอิมพัลส์จำนวนไม่จำกัด (Infinite Impulse Response (IIR)) เป็นต้น

2.4.2.5 ฟังก์ชันการแปลงสัญญาณ

ใช้ในการแปลงสัญญาณ (Transform Functions) ในเชิงเวลาและสัญญาณในเชิงความถี่ ฟังก์ชันการแปลงสัญญาณ เช่น การแปลงโคไซน์ไม่ต่อเนื่อง (Discrete Cosine Transform (DCT)), การแปลงฟูริเยร์แบบเร็ว (Fast Fourier Transform (FFT)), การแปลงผกผันฟูริเยร์แบบเร็ว (Inverse Fast Fourier Transform (IFFT)) เป็นต้น

2.5 คุณสมบัติเด่นโดยรวมของ dsPIC

2.5.1 คุณสมบัติของซีพียู

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูแบบ RISC
- ความเร็วในการทำงานสูงถึง 30 ล้านคำต่อวินาที
- มีจำนวน 84 คำสั่งภาษาแอสแซมบลีมาตรฐาน ซึ่งรองรับรูปแบบการอ้างแอดเดรสได้อย่างอิสระ
- ชุดคำสั่งมีขนาด 24 บิต สามารถประมวลผลข้อมูลได้ 16 บิต

- มีหน่วยความจำโปรแกรมเป็นแบบแฟลช ที่สามารถลบและเขียนใหม่ได้ไม่น้อยกว่า 100,000 ครั้ง สามารถป้องกันการอ่านได้ และสามารถโปรแกรมตัวเอง โดยใช้กระบวนการทางซอฟต์แวร์

- มีหน่วยความจำข้อมูลอีอีพรอม ที่สามารถลบและเขียนใหม่ได้ไม่น้อยกว่า 1,000,000 ครั้ง

- มีอินเตอร์รัปต์เวกเตอร์จำนวนมากจึงรองรับการตอบสนองสัญญาณอินเตอร์รัปต์ได้ดี

- มีวงจรตรวจจับแรงดันไฟเลี้ยงต่ำกว่ากำหนดแบบโปรแกรมได้

- มีเพาเวอร์-ออนรีเซต, เพาเวอร์-อัปไทเมอร์ และออสซิลเลเตอร์สตาร์ต-อัปไทเมอร์

- มีวอตช์ด็อกไทเมอร์แบบโปรแกรมได้

- มีวงจรตรวจสอบการทำงานของวงจรกำเนิดสัญญาณนาฬิกา

- รองรับการโปรแกรมในวงจรแบบอนุกรม (ICSP : In-Circuit Serial Programming)

- สามารถเลือกโหมดการใช้พลังงานได้

2.5.2 คุณสมบัติด้านการประมวลสัญญาณดิจิตอล

- มีแอกคิวมูเลเตอร์ขนาด 40 บิตจำนวน 2 ตัว ที่รองรับการประมวลผลทางคณิตศาสตร์ได้เป็นอย่างดี

- มีหน่วยประมวลผลทางด้านการคูณและการหารเลข 17 บิต ในรูปของฮาร์ดแวร์จึงทำให้สามารถคูณและหารเลขได้อย่างรวดเร็ว

- ทำการคูณเลข 16 บิตได้ภายในสัญญาณนาฬิกาเพียง 1 ไซเคิล

- มีตัวเลื่อนข้อมูลบาร์เรล 40 สเตจ ช่วยให้การประมวลผลข้อมูลที่จำนวนบิตมากๆ สามารถทำได้อย่างรวดเร็ว

- มีวงจรเฟตช์ข้อมูลคู่ จึงทำให้สามารถประมวลผลข้อมูลได้อย่างรวดเร็ว

2.5.3 คุณสมบัติของโมดูลฟังก์ชั่นพิเศษ

- สามารถจ่ายกระแสออกทางขาพอร์ตได้ 25 mA ทั้งแบบกระแสซิงก์และซอร์ส

- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตไม่น้อยกว่า 3 ตัวต่อใช้งานร่วมกันเป็นไทเมอร์32 บิต

- มีโมดูลตรวจจับและเปรียบเทียบสัญญาณดิจิตอล

- มีส่วนเชื่อมต่ออุปกรณ์อนุกรมทั้งแบบ SPI และผ่านระบบบัส I2C

- มีโมดูลสื่อสารข้อมูลอนุกรม UART พร้อมบัฟเฟอร์แบบ FIFO

- มีวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ความละเอียด 10 หรือ 12 บิต

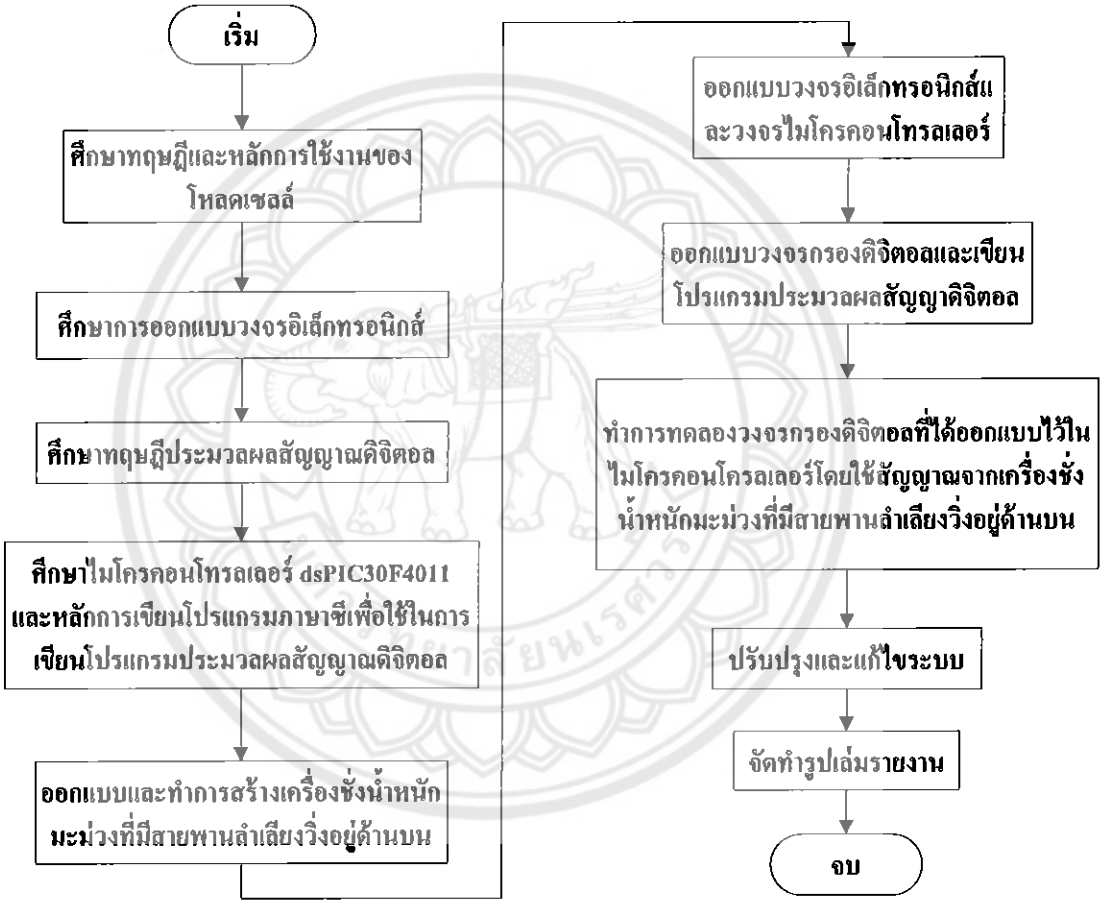
- มีโมดูลสร้างสัญญาณ PWM สำหรับควบคุมมอเตอร์

- มีโมดูลเชื่อมต่อตัวเข้ารหัสแบบควอดราเจอร์

บทที่ 3

วิธีดำเนินโครงงาน

จากการที่ได้ศึกษาทฤษฎีบทที่ 2 มาแล้วในบทนี้จะกล่าวถึงขั้นตอนการดำเนินงานในการออกแบบวงจรอิเล็กทรอนิกส์และวงจรกรองดิจิตอลพร้อมทั้งสร้างเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน โดยมีขั้นตอนดังนี้



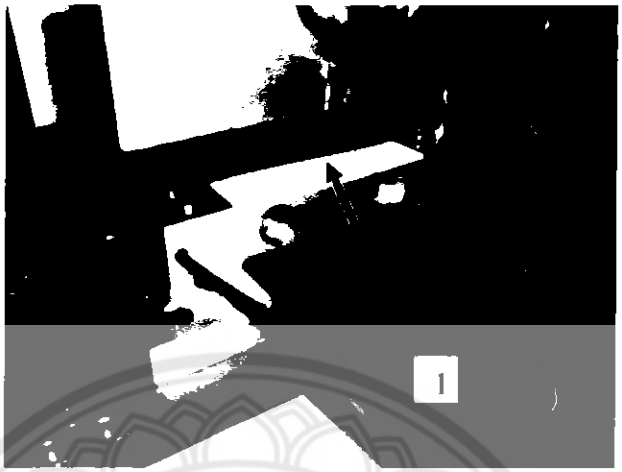
รูปที่ 3.1 แผนภาพขั้นตอนดำเนินงาน

จากรูปที่ 3.1 แสดงขั้นการดำเนินงานทั้งหมดสามารถอธิบายรายละเอียดได้ ดังนี้

3.1 ศึกษาทฤษฎีและหลักการใช้งานของโหลดเซลล์

เป็นการศึกษาถึงทฤษฎีที่เกี่ยวข้องกับโหลดเซลล์ (Load Cell) แต่ละชนิดอย่างเช่นโหลดเซลล์แบบคาน (Beam-type load cell) ที่ใช้คานยื่นทำหน้าที่เป็นชิ้นส่วนยืดหยุ่นซึ่งมีสเตรนเกจ 2

อันติดอยู่ที่ผิวด้านบนและอีก 2 อัน ติดที่ผิวด้านล่างทั้งหมดติดอยู่ในแนวขนานกับแกนของคานทำหน้าที่เป็นตัวรับรู้ ดังรูปที่ 3.2

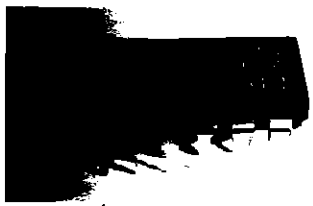


รูปที่ 3.2 แสดงลักษณะของโหลดเซลล์

" 1 " คือ ตำแหน่งของสเตรนเกจ โดยติดที่ผิวด้านบนและติดที่ผิวด้านล่าง ทั้งหมดติดอยู่ในแนวขนานกับแกนของคานและมีสารสีขาวเคลือบอยู่เพื่อป้องกันแรงกระแทก การขีดคว่นและป้องกันสัญญาณรบกวนมีขนาดแรงดันที่ออกมามีค่าเท่ากับ 2 มิลลิโวลต์ต่อโวลต์ หมายถึง ป้อนแรงให้กับโหลดเซลล์ 10 โวลต์ แรงดันที่ออกมาก็จะได้ 20 มิลลิโวลต์

3.2 ศึกษาการออกแบบวงจรอิเล็กทรอนิกส์

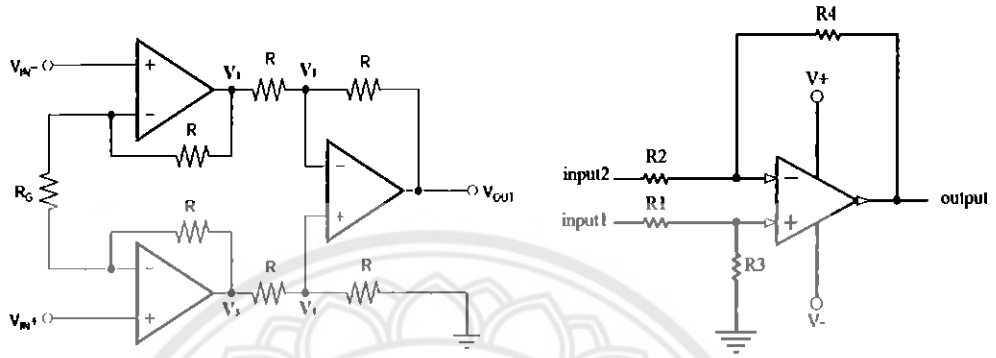
ศึกษาอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ที่ใช้ในการออกแบบ เช่น ออปแอมป์เบอร์ TL074CN ดังรูปที่ 3.3



รูปที่ 3.3 TL074CN

จากรูปที่ 3.3 แสดงลักษณะของออปแอมป์เบอร์ TL074CN ซึ่งรับไฟเลี้ยงไม่เกิน +18 โวลต์ อินพุตไม่เกิน +15 โวลต์ จะมีอัตราขยาย 200,000 เท่า รายละเอียดนอกจากนี้สามารถดูได้จากภาคผนวก

และศึกษาการออกแบบวงจรอิเล็กทรอนิกส์เพื่อใช้ในการปรับปรุงสัญญาณที่ได้จากโหลดเซลล์ ซึ่งเป็นสัญญาณแอนะล็อก (Analog) เช่น ศึกษาการออกแบบวงจรขยายสัญญาณอินสตรูเมนท์ (Instrument Amplifier Circuit) วงจรลบสัญญาณ ดังรูปที่ 3.4 นอกจากนี้ยังมีวงจรอื่นอีกแต่จะขอกล่าวในหัวขอปรับระดับแรงดันและวงจรแหล่งจ่ายไฟตรง (DC Power Supply) ดังรูปที่ 3.4



ก. ข.

รูปที่ 3.4 ก. วงจรขยายสัญญาณอินสตรูเมนท์ (Instrument Amplifier Circuit) และ ข. วงจรลบสัญญาณ (Subtractor Circuit)

จากรูปที่ 3.4 ก. วงจรขยายสัญญาณอินสตรูเมนท์ เป็นวงจรขยายความแตกต่างของ 2 สัญญาณ สามารถหาอัตราขยายได้ดังนี้

กำหนดให้

$$V_{IN^+} = V_a$$

$$V_{IN^-} = V_b$$

โนด V_b :

$$\frac{V_b - V_a}{R_G} + \frac{V_b - V_1}{R} = 0$$

$$\frac{V_b(R+R_G)}{R} - \frac{V_1 R_G}{R} = V_a \qquad (3.1)$$

โนด V_a :

$$\frac{V_a - V_b}{R_G} + \frac{V_a - V_3}{R} = 0$$

$$\frac{V_a(R+R_G)}{R} - \frac{V_3 R_G}{R} = V_b$$

$$V_a(1+\frac{R_G}{R}) - \frac{V_3 R_G}{R} = V_b \qquad (3.2)$$

โนด V_2 :

$$\frac{V_2 - V_1}{R} + \frac{V_2 - V_0}{R} = 0$$

$$2V_2 - V_0 = V_1 \tag{3.3}$$

$$\frac{V_2 - V_3}{R} + \frac{V_2 - 0}{R} = 0$$

$$2V_2 = V_3 \tag{3.4}$$

นำสมการที่ (3.4) แทนในสมการที่ (3.3) จะได้

$$V_3 - V_0 = V_1 \tag{3.5}$$

นำสมการที่ (3.5) แทนในสมการที่ (3.1) จะได้

$$V_b(1+\frac{R_G}{R}) - (V_3 - V_0)\frac{R_G}{R} = V_a \tag{3.6}$$

นำสมการที่ (3.6) ลบกับสมการที่ (3.2) จะได้

$$V_a - V_b = \left[V_b(1+\frac{R_G}{R}) - (V_3 - V_0)\frac{R_G}{R}\right] - \left[V_a(1+\frac{R_G}{R}) - \frac{V_3 R_G}{R}\right]$$

$$V_a - V_b = V_b(1+\frac{R_G}{R}) - \frac{V_3 R_G}{R} + \frac{V_0 R_G}{R} - V_a(1+\frac{R_G}{R}) + \frac{V_3 R_G}{R}$$

$$V_a - V_b = V_b(1+\frac{R_G}{R}) - V_a(1+\frac{R_G}{R}) + \frac{V_0 R_G}{R}$$

$$V_a\left[1+(1+\frac{R_G}{R})\right] - V_b\left[1+(1+\frac{R_G}{R})\right] = \frac{V_0 R_G}{R}$$

$$(V_a - V_b)\left(2+\frac{R_G}{R}\right) = \frac{V_0 R_G}{R}$$

$$A_v = \frac{V_0}{(V_a - V_b)} = \frac{R}{R_G}(2+\frac{R_G}{R})$$

ดังนั้นจะได้อัตราขยายคือ

$$A_v = \frac{V_0}{(V_{IN^+} - V_{IN^-})} = 1 + \frac{2R}{R_G} \tag{3.7}$$

และจากรูปที่ 3.4 ข. วงจรลบสัญญาณ (Subtractor Circuit) ทำหน้าที่ลบสัญญาณที่ได้จากการขยายสัญญาณความแตกต่างของวงจรขยายสัญญาณอินสตรูเมนท์ (Instrument Amplifier Circuit) ให้อยู่ประมาณ 0 โวลต์ เนื่องจากมีน้ำหนักของฐานของรับน้ำหนัก มอเตอร์ และส่วนประกอบอื่นๆที่อยู่ด้านบนโหลดเซลล์ ซึ่งเป็นน้ำหนักที่เราไม่ต้องการ สามารถหาสมาการได้ดังนี้

$$\frac{Va - input1}{R1} + \frac{Va - 0}{R3} = 0$$

$$Va(R3) - input1(R3) - Va(R1) = 0$$

$$Va(R3 - R1) = input1(R3)$$

$$Va = input1(\frac{R3}{R3 + R1}) \tag{3.8}$$

$$\frac{Va - input2}{R2} + \frac{Va - output}{R4} = 0$$

$$Va(R4) - input1(R4) + Va(R2) - output(R2) = 0$$

$$Va(R4 + R2) - input2(R4) = output(R2) \tag{3.9}$$

นำสมการที่ (3.8) แทนในสมการที่ (3.9) จะได้

$$input1(\frac{R3}{R3 + R1})(R4 + R2) - input2(R4) = output(R2)$$

$$input1(\frac{R3}{R3 + R1})(R4 + R2) - input2(R4) = output(R2)$$

ให้ $R1 = R2 = R3 = R4$ จะได้

$$input1 - input2 = output$$

ดังนั้นจะได้สมาการคือ

$$output = input1 - input2 \tag{3.10}$$

3.3 ศึกษาทฤษฎีประมวลผลสัญญาณดิจิตอล

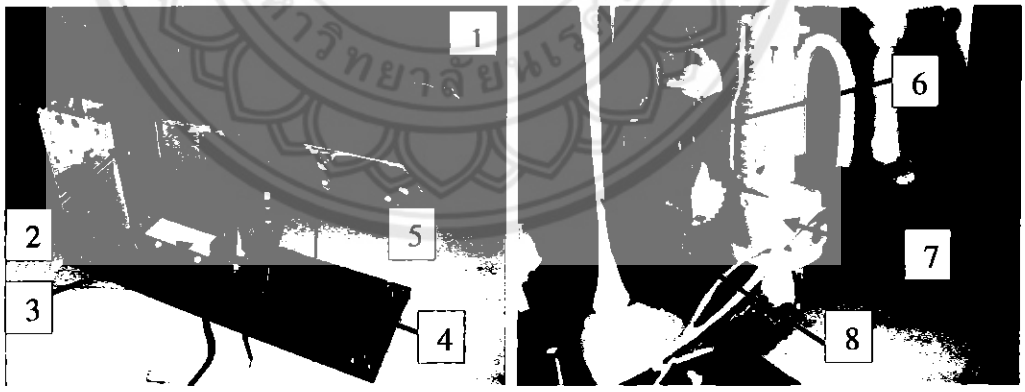
ศึกษาทฤษฎีเกี่ยวกับกรรมวิธีสัญญาณดิจิตอลต่าง ๆ เพื่อใช้ในการออกแบบวงจรกรองสัญญาณดิจิตอล เช่น สัญญาณและระบบ การชุ่มตัวอย่าง และการออกแบบตัวกรองดิจิตอล เป็นต้น

3.4 ศึกษาไมโครคอนโทรลเลอร์ dsPIC30F4011 และการเขียนโปรแกรมภาษาซีเพื่อใช้ในการประมวลผลสัญญาณดิจิตอล

เป็นการศึกษาถึงคุณสมบัติและวิธีการใช้งานฟังก์ชันไลบรารีต่าง ๆ ที่เกี่ยวกับการประมวลผลสัญญาณ (DSP) ของไมโครคอนโทรลเลอร์ dsPIC30F4011 (สามารถศึกษาข้อมูลและคุณสมบัติต่างของ dsPIC30F4011 ได้จากภาคผนวก) รวมถึงศึกษาคำสั่งและเงื่อนไขการเขียนโปรแกรมภาษาซีที่ใช้ในไมโครคอนโทรลเลอร์ dsPIC30F4011 เพื่อทำการประมวลผลสัญญาณดิจิตอล (โปรแกรมที่จำเป็นในการทดลองและวิธีการลงโปรแกรมร่วมถึงรายละเอียดที่จำเป็นสามารถดูได้จากภาคผนวก)

3.5 ออกแบบและทำการสร้างเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน

ออกแบบและทำการสร้างเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน ซึ่งมีส่วนประกอบที่สำคัญดังรูปที่ 3.5



รูปที่ 3.5 แสดงเครื่องชั่งน้ำหนักที่มีสายพานลำเลียงวิ่งอยู่ด้านบนที่สร้างเสร็จแล้ว

จากรูปมีส่วนประกอบดังนี้

" 1 " คือ สายพานลำเลียงซึ่งทำจากผ้าทั่วไป โดยมีความกว้างประมาณ 11 เซนติเมตร และมีความยาวประมาณ 110 เซนติเมตร ใช้สำหรับการลำเลียงมะม่วง

" 2 " คือ ชุดล้อลูกปืนมีทั้งหมด 4 ชุด อยู่บริเวณหัวท้ายอย่างละ 1 ชุด ทำหน้าที่ลดแรง

เสียดทานเพื่อช่วยให้สายพานวิ่งไปได้ง่ายขึ้นและอยู่ด้านล้าง 2 ชุด ทำหน้าที่เหมือกับ 2 ชุดแรกและยังทำหน้าที่ในการปรับความดึงของสายพานเมื่อสายพานย่อนเกินไปหรือตึงเกินไป

" 3 " คือ โหลดเซลล์แบบคาน ทำหน้าที่เปลี่ยนแรงทางกลให้เป็นแรงดันไฟฟ้า

" 4 " คือ ฐานรองรับของโหลดเซลล์ โดยมีความกว้างประมาณ 13 เซนติเมตร และมีความยาวประมาณ 35 เซนติเมตร

" 5 " คือ ฐานรองรับน้ำหนัก ซึ่งทำหน้าที่ในการรับน้ำหนักของมะม่วงขณะที่วิ่งอยู่ โดยจะมีสายพานลำเลียงวิ่งอยู่ด้านบน

" 6 " คือ ชุดเฟืองทด ทำหน้าที่เป็นตัวกลางในการส่งแรงจากมอเตอร์ไปยังสายพานลำเลียง โดยมีการทดเกียร์เพื่อเป็นการผ่อนแรงจึงทำให้มอเตอร์แรงน้อยสามารถขับโหลดที่มีค่ามากกว่าได้ แต่ความเร็วของสายพานจะต่ำจึงทำให้สายพานวิ่งช้า

" 7 " คือ มอเตอร์กระแสตรง 12 โวลต์ ทำหน้าที่ในการขับสายพานลำเลียง

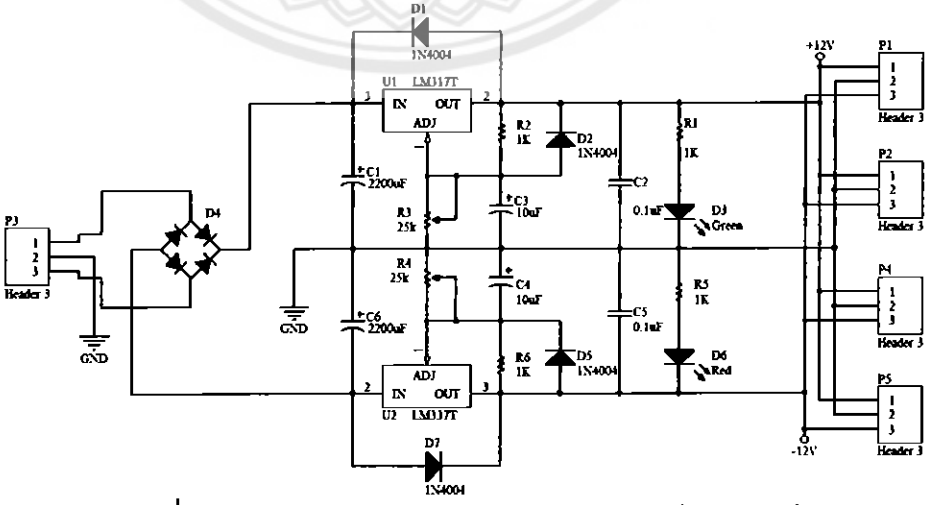
" 8 " คือ แท่งเหล็กโหลดเซลล์ที่อยู่ด้านล้างกับฐานรับน้ำหนักที่อยู่ด้านบน

3.6 ออกแบบวงจรอิเล็กทรอนิกส์

ทำการออกแบบวงจรอิเล็กทรอนิกส์เพื่อใช้สำหรับประมวลผลสัญญาณ โดยมีวงจรดังนี้

1. วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์
2. วงจรปรับปรุงสัญญาณ
3. วงจรปรับความเร็วมอเตอร์
4. วงจรไมโครคอนโทรเลอร์

3.6.1 วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์



รูปที่ 3.6 วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์

วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ จากรูปที่ 3.6 มีหน้าที่จ่ายไฟฟ้าให้กับวงจรต่างๆ ดังนี้

1. จ่ายเป็นไฟเลี้ยงให้กับออปแอมป์ ในภาคของวงจรปรับปรุงสัญญาณ (วงจรปรับปรุงสัญญาณจะขอกล่าวในหัวขอถัดไป)

2. จ่ายไฟเลี้ยงให้กับออปแอมป์ ในภาควงจรไมโครคอนโทรลเลอร์ส่วนของการแปลงสัญญาณดิจิตอลเป็นแอนะล็อก (D to A) (วงจรไมโครคอนโทรลเลอร์จะขอกล่าวในหัวข้อถัดไป)

3. จ่ายไฟให้กับโหลดเซลล์ +12 โวลต์

4. จ่ายไฟให้กับพัดลมระบายอากาศ 24 โวลต์ โดยการจ่ายไฟ +12 โวลต์และ-12 โวลต์อย่างละขั้วของพัดลม

วงจรวงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ จากรูปที่ 3.6 มีการทำงานโดยรับไฟฟ้ากระแสสลับจากไฟบ้านผ่านทาง P3 ไปยังส่วนของไดโอด D4 เพื่อทำหน้าที่ในการแปลงไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรง ไฟฟ้ากระแสตรงที่ได้นี้จะมีลักษณะของครึ่งคลื่นไซน์อยู่ยังไม่สามารถใช้งานได้จึงต้องนำไปกรองออกโดยการนำไปผ่าน C1 จะทำให้สัญญาณที่ได้มีลักษณะเป็นเส้นตรงมากขึ้นแต่ยังไม่มีประสิทธิภาพพอ ดังนั้นจึงต้องมีการใช้ U1 (LM317T) จะช่วยในการคุมแรงดันทำให้สัญญาณที่ได้เป็นเส้นตรงและมีประสิทธิภาพมากขึ้นและยังสามารถปรับแรงดันตามความต้องการได้ สามารถปรับแรงดันตามสมการที่ (3.11)

$$V_{out} = 1.25\left(1+\frac{R3}{R2}\right) + I_{ADJ}(R3) \qquad (3.11)$$

โดยที่ I_{ADJ} เท่ากับ 50×10^{-6} แอมป์

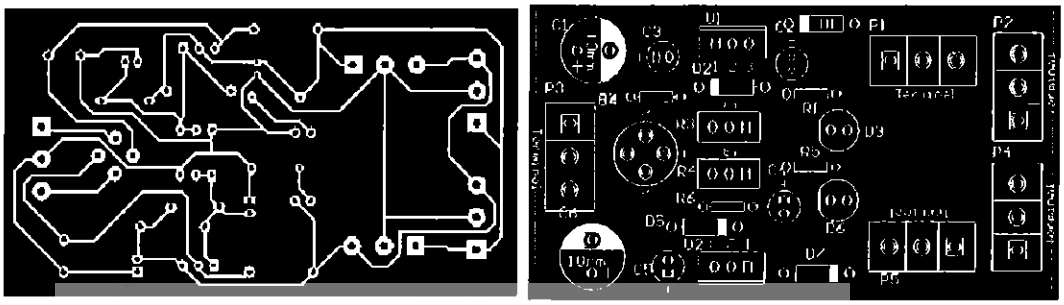
ส่วนวงจรด้านไฟลบก็เช่นเดียวกันกับวงจรด้านไฟบวก แต่เปลี่ยน U1 (LM317T) เป็น U2 (LM337T) สามารถปรับแรงดันตามสมาการที่ (3.12)

$$-V_{out} = \left|-1.25\left(1+\frac{R4}{R2}\right) - \left(I_{ADJ}(R4)\right)\right| \qquad (3.12)$$

โดยที่ I_{ADJ} เท่ากับ $65\text{x}10^{-6}$ แอมป์

ไดโอด D1, D2, D5 และ D7 ทำหน้าที่ป้องกันกระแสที่ไหลย้อนกลับเนื่องจากการปรับค่าแรงดันของวงจรที่นำมาต่อเพื่อใช้งาน C3 และ C4 ทำหน้าที่กรองสัญญาณรบกวนจากการปรับค่า R3 และ R4 D3 และ D6 มีไว้แสดงสถานะไฟบวกและไฟลบ ตามลำดับ

วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ จากรูปที่ 3.6 นำมาทำลายวงจรพิมพ์บนแผ่นวงจรพิมพ์ (Printed Circuit Board: PCB) โดยใช้โปรแกรม Altium Designer จะแสดงไว้ดังรูปที่ 3.7



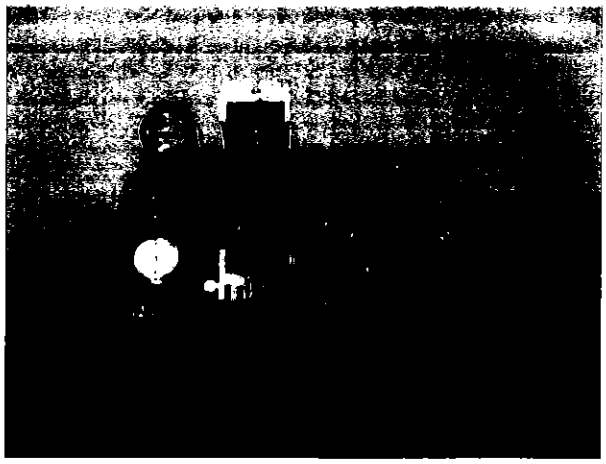
ก. ข.

รูปที่ 3.7 ก. ลายวงจรพิมพ์วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ ข. ลักษณะการว่าง

จากรูปที่ 3.7 ข. เป็นการวางตำแหน่งของอุปกรณ์ที่ได้ออกแบบไว้โดยมีอุปกรณ์ดังนี้

- U1 คือ ไอซี LM317T จำนวน 1 ตัว
- U2 คือ ไอซี LM337T จำนวน 1 ตัว
- C1, C6 = 2200 ไมโครฟารัด จำนวน 2 ตัว
- C3, C4 = 10 ไมโครฟารัด จำนวน 2 ตัว
- C2, C5 = 0.22 ไมโครฟารัด จำนวน 2 ตัว
- ความต้านทานปรับค่าได้ R3, R4 = 25 กิโลโอห์ม จำนวน 2 ตัว
- ความต้านทาน R1, R2, R5, R6 = 1 กิโลโอห์ม จำนวน 4 ตัว
- D4 (W04M) จำนวน 1 ตัว
- ไดโอด IN4004 D1, D2, D5, D7 จำนวน 4 ตัว
- ไดโอดเปล่งแสง (LED) D6 จำนวน 1 ตัว
- ไดโอดเปล่งแสง (Green) D3 จำนวน 1 ตัว

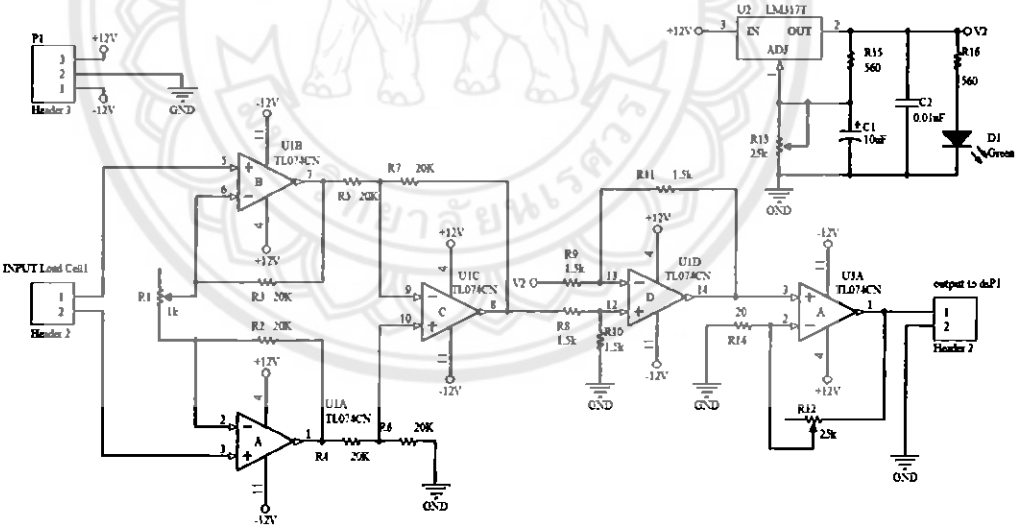
ดังนั้นจะได้วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ ที่อยู่บนแผ่นวงจรพิมพ์ดังรูปที่ 3.8



รูปที่ 3.8 วงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ ที่สามารถใช้งานได้

3.6.2 วงจรปรับปรุงสัญญาณ

วงจรปรับปรุงสัญญาณจะเป็นส่วนที่กระทำกับสัญญาณที่ได้จากโหลดเซลล์ให้มีขนาดที่เหมาะสมกับการประมวลผลสัญญาณดิจิตอลต่อไปโดยประกอบไปด้วย วงจรขยายแบบสัญญาณอินสตรูเมนท์ ดังรูปที่ 3.9



รูปที่ 3.9 วงจรปรับปรุงสัญญาณ

จากรูปที่ 3.9 เป็นวงจรขยายความแตกต่างของสัญญาณที่ออกมาจากโหลดเซลล์ สัญญาณที่ออกมาจากโหลดเซลล์จะมีขนาดน้อยมากอยู่ในระดับมิลลิแอมป์จึงไม่สามารถนำมาประมวลผลได้ จึงต้องนำวงจรขยายความแตกต่างอย่างวงจรขยายอินสตรูเมนท์มาขยายสัญญาณเพื่อให้ได้สัญญาณที่เหมาะสม จากนั้นเมื่อขยายสัญญาณแล้วก็จะใช้วงจรลบสัญญาณ เพื่อทำการลบสัญญาณที่ไม่ต้องการที่เกิดจากน้ำหนักของส่วนต่างๆ ที่โหลดเซลล์รับน้ำหนักอยู่ เช่น น้ำหนักจาก

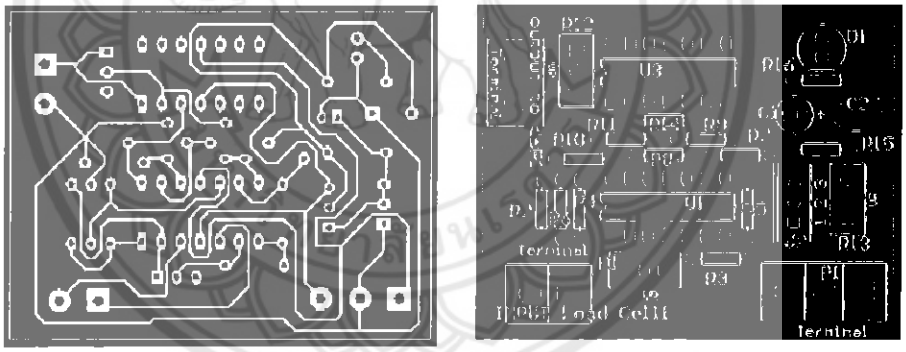
มอเตอร์ ฐานรับน้ำหนักที่มีสายพานวิ่งอยู่ด้านบน เป็นต้น วงจรสุดท้ายของส่วนปรับปรุงสัญญาณคือวงจรขยายสัญญาณแบบธรรมดา มีหน้าที่ในการขยายสัญญาณอีกทีเพื่อให้ได้สัญญาณตามความต้องการและยังช่วยจำกัดสัญญาณอยู่ในช่วงประมาณ 0 โวลต์ ถึง 5.5โวลต์ ถ้าแรงดันที่ได้มีค่าเกิน 5.5 โวลต์ จะทำให้วงจรไมโครคอนโทรลเลอร์เสียหายได้ เนื่องจากไมโครคอนโทรลเลอร์ dsPIC30F4011 รับอินพุตสูงสุดได้แค่ 5.5 โวลต์

วงจรปรับปรุงสัญญาณจากรูปที่ 3.9 สามารถปรับอัตราขยายได้จาก R1 ซึ่งเป็นความต้านทานปรับค่าได้มีค่าเท่ากับ 0-1000 โอห์ม ได้จากสมการที่ (3.7) ดังนี้

$$A_v = \frac{V_0}{(V_{IN^+} - V_{IN^-})} = 1 + \frac{2R}{R_G}$$

โดยที่ R_Gเท่ากับ R1

วงจรปรับปรุงสัญญาณจากรูปที่ 3.9 นำมาทำลายวงจรพิมพ์บนแผ่น**วงจรพิมพ์** (Printed Circuit Board: PCB) โดยใช้โปรแกรม Altium Designer จะได้ดังรูปที่ 3.10



ก. ข.

รูปที่ 3.10 ก. ลายวงจรพิมพ์ของวงจรปรับปรุงสัญญาณ ข. ลักษณะการว่างอุปกรณ์ของวงจรปรับปรงสัณฌาณ

จากรูปที่ 3.10 ข. มีอุปกรณ์อิเล็กทรอนิกส์ดังนี้

- U1, U3 คือ แอปแอมป์เบอร์ TL074CT จำนวน 2 ตัว
- U2 คือ ไอซี LM317T จำนวน 1 ตัว
- C1 = 10 ไมโครฟารัด จำนวน 1 ตัว
- C2 = 0.01 ไมโครฟารัด จำนวน 1 ตัว
- ความต้านทานปรับค่าได้ R13, R12 = 25 กิโลโอห์ม จำนวน 2 ตัว

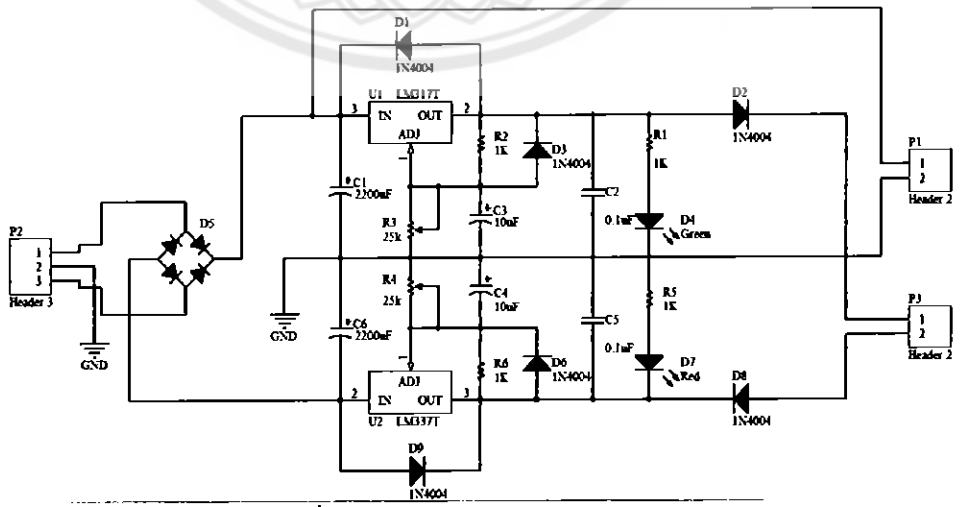
- ความต้านทานปรับค่าได้ R1 = 1 กิโลโอห์ม จำนวน 1 ตัว
- ไดโอดเปล่งแสง (Green) D1 จำนวน 1 ตัว
- ความต้านทาน R2, R3, R4, R5 R6, R7 = 20 กิโลโอห์ม จำนวน 6 ตัว
- ความต้านทาน R8, R9, R10, R11 = 1.5 กิโลโอห์ม จำนวน 4 ตัว
- ความต้านทาน R14 = 20 โอห์ม จำนวน 1 ตัว
- ความต้านทาน R15, R16 = 560 โอห์ม จำนวน 2 ตัว

ดังนั้นจะได้วงจรปรับปรุสัญญาณที่อยู่บนแผ่นวงจรพิมพ์ ดังรูปที่ 3.11

รูปที่ 3.11 วงจรปรับปรุงสัญญาณที่อยู่บนแผ่นวงจรพิมพ์

3.6.3 วงจรปรับความเร็วมอเตอร์

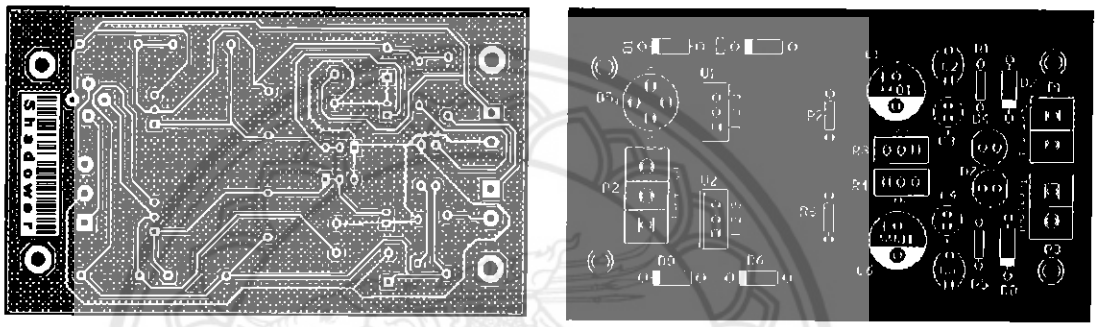
นำวงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ มาประยุกต์ใช้เป็นวงจรปรับความเร็วมอเตอร์ ดังรูปที่ 3.12



รูปที่ 3.12 วงจรปรับความเร็วมอเตอร์

วงจรปรับความเร็วมอเตอร์ จากรูปที่ 3.12 การทำงานของวงจรปรับความเร็วมอเตอร์จะเหมือนกลับวงจรแหล่งจ่ายไฟกระแสตรง +12 โวลต์ -12 โวลต์ โดยปรับความเร็วจากการปรับค่าความต้านทานปรับค่าได้ R3, R4 โดยใช้ตัวต้านทานปรับค่าได้แบบสองชั้นหมุนด้วยมือ แทน นอกจากปรับความเร็วมอเตอร์แล้วยังนำแรงดันที่ได้จากการแปลงไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรงไปใช้ เพื่อสร้างแหล่งให้กับวงจรไมโครคอนโทรลเลอร์

วงจรปรับความเร็วมอเตอร์ จากรูปที่ 3.12 นำมาทำลายวงจรพิมพ์บนแผ่นวงจรพิมพ์ (Printed Circuit Board: PCB) โดยใช้โปรแกรม Altium Designer จะได้ดังรูปที่ 3.13



ก ข.

รูปที่ 3.13 ก. ลายวงจรพิมพ์วงจรปรับความเร็วมอเตอร์ ข. ลักษณะการว่างอุปกรณ์วงจรปรับความเร็วมอเตอร์

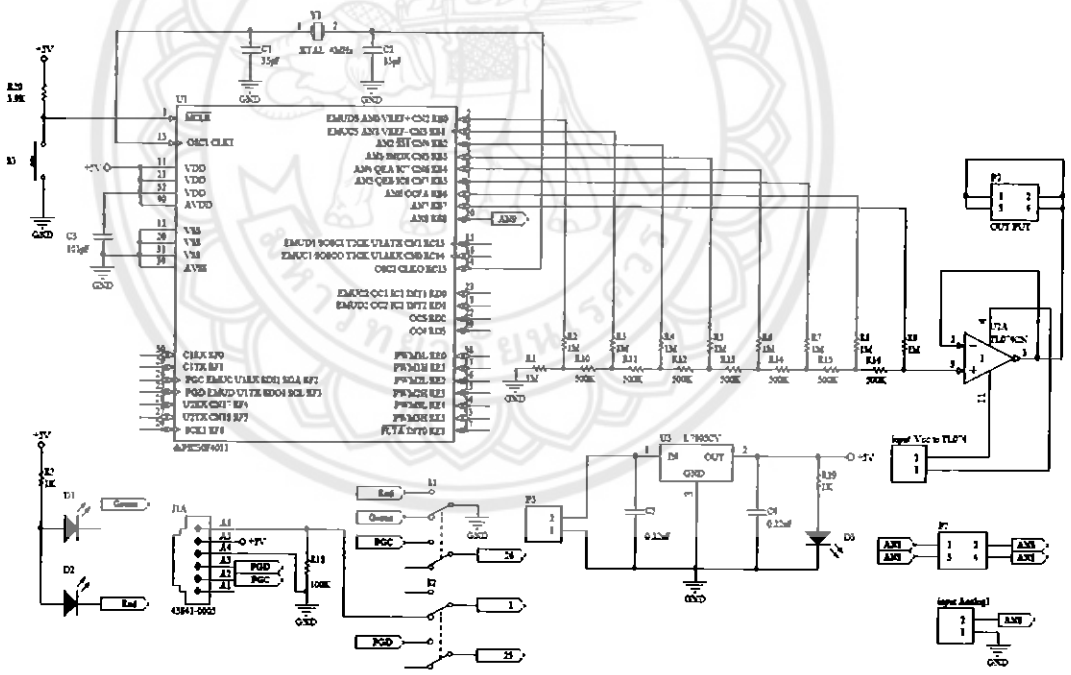
จากรูปที่ 3.13 ข. เป็นการวางตำแหน่งของอุปกรณ์ที่ได้ออกแบบไว้ โดยมีอุปกรณ์ดังนี้

- U1 คือ ไอซี LM317T จำนวน 1 ตัว
- U2 คือ ไอซี LM337T จำนวน 1 ตัว
- C1, C6 = 2200 ไมโครฟารัด จำนวน 2 ตัว
- C3, C4 = 10 ไมโครฟารัด จำนวน 2 ตัว
- C2, C5 = 0.22 ไมโครฟารัด จำนวน 2 ตัว
- ความต้านทานปรับค่าได้ โดยใช้ความต้านทางสองชั้น 6 ขา หมุนด้วยมือได้ แทน R3, R4 = 20 กิโลโอห์ม จำนวน 1 ตัว
- ความต้านทาน R1, R2, R5, R6 = 1 กิโลโอห์ม จำนวน 4 ตัว
- D5 (W04M) จำนวน 1 ตัว
- ไดโอด IN4004 D1, D2, D5, D7 จำนวน 4 ตัว
- ไดโอดเปล่งแสง (LED) D4 จำนวน 1 ตัว
- ไดโอดเปล่งแสง (Green) D7 จำนวน 1 ตัว

ดังนั้นจะได้วงจรปรับความเร็วมอเตอร์ที่อยู่บนแผ่นวงจรพิมพ์ ดังรูปที่ 3.14

รูปที่ 3.14 วงจรปรับความเร็วมอเตอร์ที่สามารถใช้งานได้

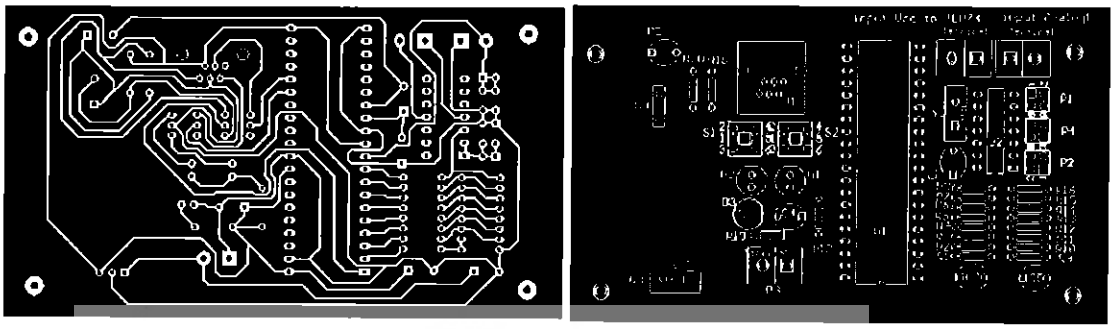
3.6.4 วงจรไมโครคอนโทรเลอร์



รูปที่ 3.15 วงจรไมโครคอนโทรเลอร์

วงจรไมโครคอนโทรเลอร์ จากรูปที่ 3.15 มีหน้าที่ในการประมวลผลสัญญาณดิจิตอลจากสัญญาณแอนะล็อกของวงจรปรับปรุงสัญญาณโดยใช้ไมโครคอนโทรเลอร์ dsPIC30F4011 ในการประมวลผลสัญญาณจากนั้นทำการแปลงสัญญาณดิจิตอลเป็นแอนะล็อกเพื่อวัดสัญญาณต่อไป (การประมวลผลสัญญาณจะขอกล่าวในหัวข้อถัดไป) วงจรไมโครคอนโทรเลอร์ที่รูปที่ 3.15

สามารถนำมาทำลายวงจรพิมพ์บนแผ่นวงจรพิมพ์ (Printed Circuit Board: PCB) โดยใช้โปรแกรม Altium Designer จะได้ดังรูปที่ 3.16



ก. ข.

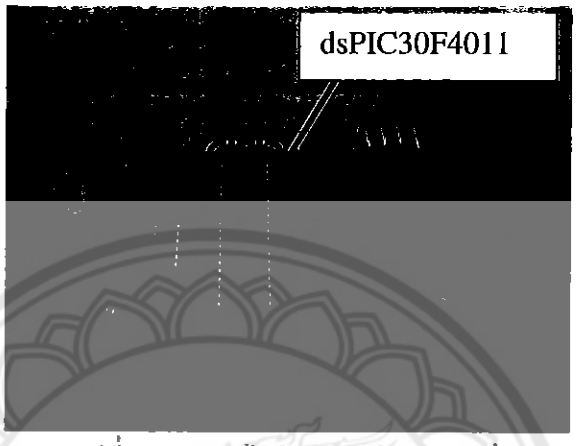
รูปที่ 3.16 ก. ลายวงจรพิมพ์วงจรวงจรไมโครคอนโทรเลอร์ ข. ลักษณะการว่างอุปกรณ์วงจรไมโครคอนโทรเลอร์

จากรูปที่ 3.16 ข. เป็นการวางตำแหน่งของอุปกรณ์ที่ได้ออกแบบไว้โดยมีอุปกรณ์ดังนี้

- U1 คือ ไมโครคอนโทรเลอร์ dsPIC30F4011 จำนวน 1 ตัว
- U2 คือ ไอซีแอปแอมป์ TL074CN จำนวน 1 ตัว
- U3 คือ ไอซี L7805CV จำนวน 1 ตัว
- J1A คือ โมดูล่าแจ็ค 6 ขา จำนวน 1 ตัว
- ความต้านทาน R1, R2, R3, R4, R5, R6, R7, R8, R9 = 1 เมกะโอห์ม ใช้งาน จำนวน 9 ตัว
- ความต้านทาน R10, R11, R12, R13, R14, R15, R16 = 500 กิโลโอห์ม จำนวน 7 ตัว
- ความต้านทาน R17 = 560 โอห์ม จำนวน 1ตัว
- ความต้านทาน R18 = 100 กิโลโอห์ม จำนวน 1 ตัว
- ความต้านทาน R19 = 1 กิโลโอห์ม จำนวน 1 ตัว
- Y1 คือ คลิสตอล 4 เมกะเฮิรตซ์ จำนวน 1 ตัว
- ตัวเก็บประจุ C1, C2 = 33 พิโคฟารัด จำนวน 2 ตัว
- ตัวเก็บประจุ C3 = 101 พิโคฟารัด จำนวน 1 ตัว
- ตัวเก็บประจุ C4, C5 = 0.22 ไมโครฟารัด จำนวน 2 ตัว
- สวิตซ์ Push switch S1, S2 ชนิด 6 ขา แบบ Lock type

- สวิตซ์รีเซต S3 จำนวน 1 ตัว
- ไฟแสดงสถานะสีเขียว D1 = LED (Red) จำนวน 1 ตัว
- ไฟแสดงสถานะสีแดง D2, D3 = LED (Green) จำนวน 2 ตัว

ดังนั้นจะได้วงจรไมโครคอนโทรเลอร์ที่อยู่บนแผ่นวงจรพิมพ์ ดังรูปที่ 3.17



รูปที่ 3.17 วงจรไมโครคอนโทรลเลอร์

3.7 ออกแบบวงจรกรองดิจิตอลและเขียนโปรแกรมประมวลผลสัญญาณ

ออกแบบวงจรกรองดิจิตอลและเขียนโปรแกรมประมวลผลสัญญาณเป็นการกระทำในส่วนของการเขียนโปรแกรมไมโครคอนโทรเลอร์เพื่อกรองสัญญาณรบกวนที่ไม่ต้องการออกไป ก่อนที่จะเขียนโปรแกรมกรองสัญญาณดิจิตอล ต้องทำการออกแบบวงจรกรองเสียก่อน ในโครงงานนี้เลือกวงจรกรองเฉลี่ยแบบเคลื่อนที่ (Moving Average Filter)

3.7.1 ออกแบบวงจรกรองดิจิตอล

การออกแบบตัวกรองดิจิตอลเลือกใช้วิธีกรองเฉลี่ยแบบเคลื่อนที่ เนื่องจากมีกระบวนการที่ง่ายและมีประโยชน์ สามารถนำไปประยุกต์ใช้โดยการเขียนโปรแกรมเพื่อกรองสัญญาณรบกวนออกได้ไม่ซับซ้อนมากนักและเหมาะสำหรับสัญญาณที่มีการเปลี่ยนแปลงค่าอย่างรวดเร็ว เช่น สัญญาณที่ได้จากโหลดเซลล์ ที่มีสัญญาณรบกวนจำนวนมากที่เกิดจากสัญญาณภายนอกและสัญญาณจากการสั่นของมอเตอร์ขณะที่กำลังขับสายพาน

โดยวิธีกรองเฉลี่ยแบบเคลื่อนที่โดยเป็นแบบคอซอลคือการที่เอาต์พุต $y[n]$ เป็นการเฉลี่ยค่าอินพุตที่ n (ค่าปัจจุบัน) ที่ $n-1$ (ที่แซมเปิลก่อนหน้า 1 แซมเปิล) และที่ $n-2$ (ที่แซมเปิลก่อนหน้า 2 แซมเปิล) สมาการดิฟเฟอเรนซ์สำหรับฟิวเตอร์จำนวน 3 ข้อมูล คือ

$$y[n]=\frac{1}{3}\left(x[n]+x[n-1]+x[n-2]\right) \tag{3.13}$$

สังเกตว่า สามารถที่ (13) ทำให้เป็นรูปทั่วไปดังนี้คือ

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{3.14}$$

โดยที่ $M = 2$ และ $b_k = 1/3$ สำหรับ $k = 0,1,2$

3.7.2 เขียนโปรแกรมประมวลผลสัญญาดิจิตอล

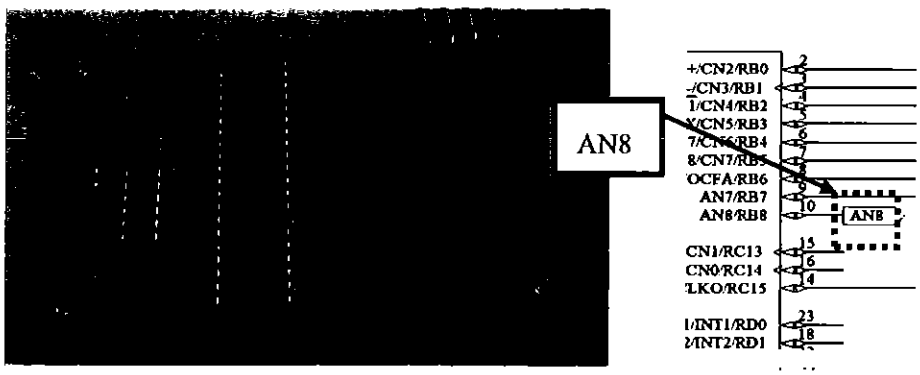
การประมวลผลสัญญาณดิจิตอล แสดงดังรูปที่ 3.18

รูปที่ 3.18 กระบวนการประมวลผลสัญญาณดิจิตอล

จากรูปที่ 3.18 จะแสดงกระบวนการประมวลผลสัญญาณดิจิตอล เริ่มต้นโดยการรับสัญญาณอินพุตที่เป็นสัญญาณแอนะล็อกจากโหลดเซลล์เพื่อทำการประมวลผลสัญญาณดิจิตอลโดยมีกระบวนการดังนี้

1. วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

การแปลงสัญญาณแอนะล็อกเป็นดิจิตอลโดยการเลือกใช้โมดูแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่มีอยู่ในตัวไมโครคอนโทรเลอร์ เนื่องจากสะดวกและรวดเร็วกว่า สามารถกำหนดความเร็วของอัตราการสุ่มตัวอย่างได้ง่าย เริ่มต้นโดยการเลือกขาอินพุตแอนะล็อกของไมโครคอนโทรเลอร์ dsPIC30F4011 ดังรูปที่ 3.19



รูปที่ 3.19 แสดงตำแหน่งการเลือกขาอินพุตแอนะล็อก

เมื่อเลือกขาได้แล้วจากนั้นทำการเขียนโปรแกรมแปลงสัญญาณแอนะล็อกเป็นดิจิตอลโดยมีขั้นตอนดังนี้

1.1 ตั้งค่าของโมดูล ADC

1.1.1 เลือกขาพอร์ตให้ทำงานเป็นอินพุตแอนะล็อกที่รีจิสเตอร์ ADPCFG

1.1.2 เลือกแหล่งจ่ายแรงดันอ้างอิงให้เหมาะสมกับย่านแรงดันแอนะล็อก ทางอินพุตที่บิต 15 ถึงบิต 13 ของรีจิสเตอร์ ADCON2

1.1.3 เลือกสัญญาณนาฬิกาที่ใช้สำหรับการแปลงสัญญาณที่บิต 5 ถึงบิต 0 ของรีจิสเตอร์ ADCON3

1.1.4 กำหนดจำนวนช่องของวงจรสุ่มสัญญาณและเก็บค่าของสัญญาณ ที่ต้องใช้ที่บิต 9 และบิต 8 ของรีจิสเตอร์ ADCON2

1.1.5 กำหนดวิธีการที่จะใช้สุ่มสัญญาณที่ บิต 3 ของรีจิสเตอร์ ADCON1 และของรีจิสเตอร์ ADCSSL

1.1.6 กำหนดจำนวนอินพุต เพื่อใช้ทำงานร่วมกับวงจรสุ่มสัญญาณและเก็บค่าสัญญาณที่รีจิสเตอร์ ADCHS

1.1.7 เลือกลำดับการสุ่มและการแปลงสัญญาณที่บิต 7 ถึงที่บิต 0 ของรีจิสเตอร์ ADCON1 และ บิต 12 ถึง 8 ของรีจิสเตอร์ ADCON3

1.1.8 เลือกรูปแบบของผลลัพธ์ที่ต้องการใช้ ที่บิต 9 และที่บิต 8 ของรีจิสเตอร์ ADCON1

1.1.9 เลือกการอินเตอร์รัปต์ที่บิต 9 ถึง 5 ของรีจิสเตอร์ ADCON2

1.1.10 เปิดการทำงานของโมดูล ADC ที่บิต 15 ของรีจิสเตอร์ ADCON1

1.2 กำหนดการอินเตอร์รัปต์ (ถ้าต้องการ)

1.2.1 เคลียร์บิต ADIF

1.2.2 เลือกระดับความสำคัญของการอินเตอร์รัปต์

จากขั้นตอนดังกล่าวสามารถดูรายละเอียดของแต่ละบิตของแต่ละส่วนของการตั่งค่าในบทที่ 2 หัวข้อการทำงานเบื้องต้นของโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ใน dsPIC30F4011 เมื่อทำตามขั้นตอนเสร็จก็จะได้ค่าต่างๆ ที่จำเป็นต่อการใช้โมดูแปลงสัญญาณแอนะล็อกเป็นดิจิตอลแล้วทำการเขียนฟังก์ชั่นการแปลงสัญญาณขึ้นมาดังนี้ (ทำการแอดแฮดเดอร์ไฟล์ชื่อ adc10.h เข้ามาร่วมด้วย วิธีการแอดไฟล์ดูได้จากภาคผนวก)

```
void ADC10_Init(void)                    // ฟังก์ชั่น ADC_Init()
{
 unsigned int config1, config2, config3;  // กำหนดตัวแปลให้กับ ADCON1 ADCON2 ADCON3
 unsigned int configport, configscan;     // กำหนดตัวแปลให้กับ ADPCFG  ADCSSL
```

unsigned int channel; // กำหนดตัวแปลให้กับ ADCHS

CloseADC10(); // เรียกใช้ฟังก์ชั่น CloseADC10(); เพื่อปิดการทำงานโมดูแปลงสัญญาณ

config1 = 0x00E6; // ADCON1>> 0000 0000 1110 0110 (บิต 7 ถึง 5 – SSRC2 ถึง SSRC0 = "111"เลือกให้กระตุ้นเมื่อตัวนับค่าภายในเสร็จสิ้นการสุ่มสัญญาณและเริ่มต้นการแปลงสัญญาณ เป็นการกำหนดให้แปลงสัญญาณโดยอัตโนมัติ

บิต 2 – ASAM = "1" เริ่มสุ่มสัญญาณทันทีที่การแปลงสัญญาณครั้งล่าสุดเสร็จสิ้นลง ทำให้บิต SAMP เซตอัตโนมัติ

บิต 1 – SAMP = "1" เลือกให้วงจรสุ่มและเก็บค่าสัญญาณอย่างน้อยหนึ่งวงจรทำการสุ่มสัญญาณหรือเอ็น

เอเบิลให้เกิดการสุ่มสัญญาณ)

config2 = 0x0000; // ADCON2>> 0000 0000 0000 0000 (บิต 15 ถึง 13 – VCFG2 ถึง VCFG0 = "000" จากขา AVDD (ปกติมีค่า +5V) จากขา AVSS (ปกติต่อลงกราวด์)

บิต 10 – CSCNA = "0" ไม่มีการสแกนอินพุต

บิต 9 และ 8 – CHPS1 และ CHPS0 = "00" ต้องการแปลงวงจรสุ่มและเก็บค่าสัญญาณช่อง CH0

บิต 7 – BUFS = "0" = แจ้งว่า ขณะนี้โมดูล ADC กำลังเขียนข้อมูลลงในบัฟเฟอร์ที่ตำแหน่ง 0x0-0x7 ผู้พัฒนาโปรแกรมสามารถเข้าถึงข้อมูลในบัฟเฟอร์ที่ตำแหน่ง 0x8-0xF ได้

บิต 5 ถึง 2 – SMPI3 ถึง SMPI0 = "0000" = เกิดอินเตอร์รัปต์เมื่อการแปลงสัญญาณเสร็จสิ้นทุกลำดับของกระบวนการสุ่มและแปลงสัญญาณ

บิต 1 – BUFM = "0" = กำหนดให้บัฟเฟอร์มีความจุ 16 เวิร์ด มีชื่อเป็น ADCBUF0 ถึง ADCBUF15

บิต 0 – ALTS = "0" = เลือกใช้อินพุต A ตลอดการทำงาน)

config3 = 0x0101; // ADCON3>> 0000 0001 0000 0001 (บิต 12 ถึง 8 – SAMC4 ถึง SAMC0 = "00001" 1 T_{AD} เลือกความเร็วสุ่มสัญญาณสูงสุด

บิต 5 ถึง 0 – ADCS5 ถึง ADCS0 = "000001" $0.5T_{CY}$ x (000001+1) = T_{CY} เลือกเวลาในการทำงานน้อยสุด)

channel = 0x0008; // ADCHS >> 0000 0000 0000 1000 (บิต 3 ถึง 0 – CH0SA3 ถึง CH0SA0 = "1000" = อินพุตบวกของ CH0 ต่อกับขา AN8)

configport = 0xFEFF; //ADPCFG >> 1111 1110 1111 1111 (กำหนดให้ AN8 = "0" กำหนดให้อินพุตแอนะล็อกทำงานในโหมดแอนะล็อก)

configscan = 0x0100; // ADCSSL>> 0000 0001 0000 0000 (บิต8 (AN8) = "1" เลือกให้อ่านค่าอินพุต)

OpenADC10(config1, config2, config3, configport, configscan); // เรียกใช้ฟังก์ชั่น OpenADC10(); เพื่อเริ่มต้นการทำงานโมดูแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

SetChanADC10(channel); //เรียนกใช้ฟังก์ฟัน SetChanADC10(); เพื่อทำการเลือกขาอินพุต AN8 แปลงสัญญาณแอนะล็อกเป็นดิจิตอล

}

2. วงจรกรองสัญญาณดิจิตอล

เมื่อกำหนดค่าต่างๆในโมดูแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลแล้วต่อไปจะเป็นการเขียนโปรแกรมกรองสัญญาณดิจิตอลโดยการยกตัวอย่างสามารที่ 3.13 การกรองสัญญาณโดยใช้ข้อมูล 3 ข้อมูล ดังนี้

$$y[n]=\frac{1}{3}(x[n]+x[n-1]+x[n-2])$$

เมื่อนำสามาการข้างต้นมาเขียนโปรแกรมจะได้ดังนี้

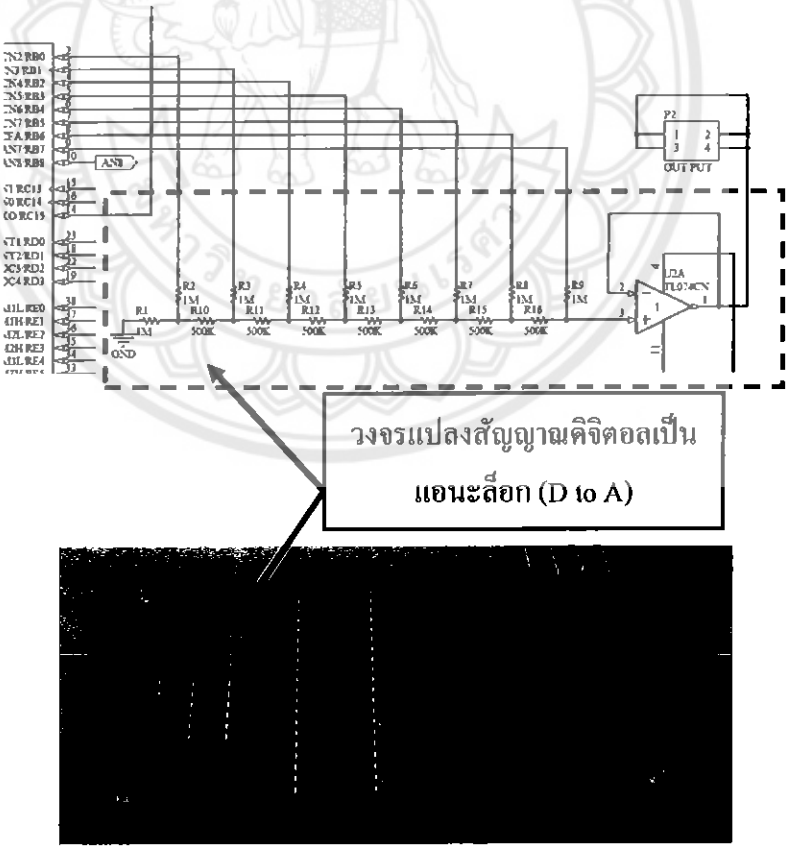
```
int main(void) ;                          // โปรแกรมหลัก
{
  TRISBbits.TRISB0= 0;                    // กำหนดให้ขา RB0 เป็นขาเอาท์พุต
  TRISBbits.TRISB1= 0;                    // กำหนดให้ขา RB1 เป็นขาเอาท์พุต
  TRISBbits.TRISB2= 0;                    // กำหนดให้ขา RB2 เป็นขาเอาท์พุต
  TRISBbits.TRISB3= 0;                    // กำหนดให้ขา RB3 เป็นขาเอาท์พุต
  TRISBbits.TRISB4= 0;                    // กำหนดให้ขา RB4 เป็นขาเอาท์พุต
  TRISBbits.TRISB5= 0;                    // กำหนดให้ขา RB5 เป็นขาเอาท์พุต
  TRISBbits.TRISB6= 0;                    // กำหนดให้ขา RB6 เป็นขาเอาท์พุต
  TRISBbits.TRISB7= 0;                    // กำหนดให้ขา RB7 เป็นขาเอาท์พุต

int y1=0;                    //กำหนดตัวแปรชนิดจำนวนเต็มและกำหนดค่าเริ่มต้นให้กับ
                             //y1=0
unsigned int x1=0,x2=0,x3=0; //กำหนดตัวแปรชนิดจำนวนเต็มแบบไม่คิดเครื่องหมายและ
                             //กำหนดค่าเริ่มต้นให้กับ x1=0, x2=0, x3=0
   while(1)                  //วนลูปไม่รู้จบ
{
```

```
        x1=ReadADC10(0)/4;    //อ่านค่าข้อมูลใส่ในตัวแปร x1
        y1=(x1+x2+x3)/3;      //หาค่าเฉลี่ยแล้วเก็บไว้ในตัวแปร y1
        LATB=y1;              //แสดงค่า y1 ออกทางพอร์ต RB0 – RB7
x3=x2;x2=x1;                  //ข้อมูล x1>>x2>>x3
}
        return 0;             //ไม่มีการคืนค่ากลับ
}
```

3. วงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก

วงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก 8 บิต โดยการใช้ความต้านทานต่อเป็นลำดับของสัญญาณทั้งหมด 8 ลำดับ โดยแต่ละลำดับรับค่าแรงดันที่ออกมาจากขาของไมโครคอนโทรเลอร์ RB0, RB1, RB2, RB3, RB4, RB5, RB6 และ RB7 และใช้แอปแอมป์ เบอร์ TL074CN ทำหน้าที่ในการรวมสัญญาณทั้งหมด 8 สัญญาณ ออกทางขา 1 ของออปแอมป์ จากรูปที่ 3.20



รูปที่ 3.20 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก

จากรูปที่ 3.20 เมื่อออกแบบวงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อกที่อยู่ภายนอก แล้วจากนั้นทำการเขียนโปรแกรมเพื่อแปลงค่าดิจิตอลที่ได้จากการกรองสัญญาณออกทางขาเอาท์พุตที่ได้ออกแบบไว้ทั้ง 8 ขา

วิธีการสั่งค่าที่ได้ออกทางขา RB0 – RB7 โดยใช้คำสั่งดังนี้

```
int main(void) ;                          // โปรแกรมหลัก
{
int y1=0;
TRISBbits.TRISB0= 0;                      // กำหนดให้ขา RB0 เป็นขาเอาท์พุต
TRISBbits.TRISB1= 0;                      // กำหนดให้ขา RB1 เป็นขาเอาท์พุต
TRISBbits.TRISB2= 0;                      // กำหนดให้ขา RB2 เป็นขาเอาท์พุต
TRISBbits.TRISB3= 0;                      // กำหนดให้ขา RB3 เป็นขาเอาท์พุต
TRISBbits.TRISB4= 0;                      // กำหนดให้ขา RB4 เป็นขาเอาท์พุต
TRISBbits.TRISB5= 0;                      // กำหนดให้ขา RB5 เป็นขาเอาท์พุต
TRISBbits.TRISB6= 0;                      // กำหนดให้ขา RB6 เป็นขาเอาท์พุต
TRISBbits.TRISB7= 0;                      // กำหนดให้ขา RB7 เป็นขาเอาท์พุต
      While
      {
     y1=ReadADC10(0)/4;                   //อ่านค่าจากการแปลงแอนะล็อกเป็นดิจิตอลใส่
                                          //ในตัวแปร y1ขนาด 8 บิต
      LATB = y1;                          //ส่งค่าในตัวแปร y1 ออกทางขา RB0 – RB7
     }
      return 0;                           //ไม่มีการคืนค่ากลับ
}
```

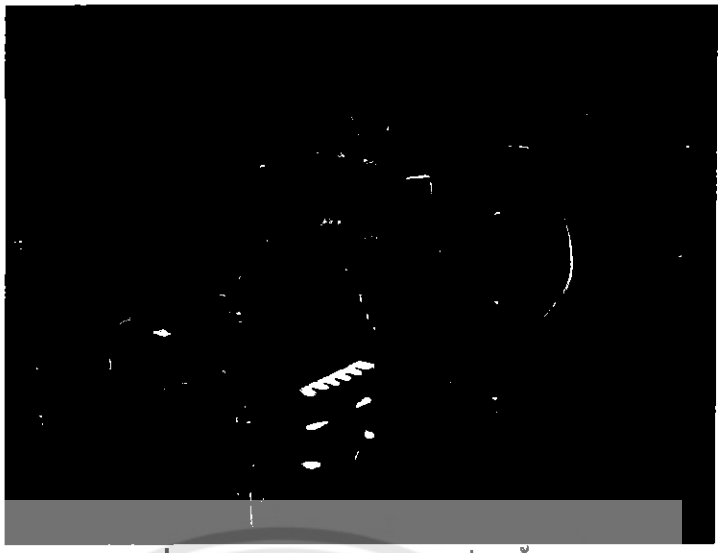
3.8 ทำการทดลองวงจรกรองดิจิตอลที่ได้ออกแบบไว้ในไมโครคอนโครลเลอร์โดยใช้สัญญาณจากเครื่องชั่งน้ำหนักมะม่วงที่มีสายพานลำเลียงวิ่งอยู่ด้านบน

หลังจากออกแบบวงจรอิเล็กทรอนิกส์ต่างๆแล้วนำมาประกอบเป็นชุดทดลองเพื่อทำการทดลองดังรูปที่ 3.21

รูปที่ 3.21 (1) แสดงวงจรอิเล็กทรอนิกส์ต่างๆที่ได้ออกแบบไว้แล้ว

รูปที่ 3.22 (2) แสดงวงจรอิเล็กทรอนิกส์ต่างๆที่ได้ออกแบบไว้แล้ว

จากรูปที่ 3.21 (1) (2) แสดงการนำเอาส่วนประกอบต่างๆ ได้แก่ วงจรของไมโครคอนโทรลเลอร์ หม้อแปลงขนาด 12 โวลต์ จำนวน 2 ตัว วงจรปรับปรุงสัญญาณ วงจรแหล่งจ่ายไฟกระแสตรง วงจรปรับความเร็วมอเตอร์ พัดลมระบายอากาศ และส่วนประกอบอื่นๆ เช่น สวิตช์ เป็นต้น มาจัดเรียงบนแผ่นอะคริลิค



รูปที่ 3.23 ส่วนประกอบต่างๆ ที่ติดตั้งเสร็จ

จากรูปที่ 3.22 แสดงการติดตั้งส่วนประกอบต่างๆ บนแผ่นอะคริลิคเรียบร้อย จากรูปจะเห็นได้ว่าทำการติดตั้งวงจรทั้งหมดและส่วนประกอบอื่นๆ แบ่งออกเป็นสองชั้น คือ ชั้นบนได้ติดตั้งวงจรไมโครคอนโทรลเลอร์และสวิตช์ต่างๆ ส่วนชั้นล่างได้ติดตั้ง หม้อแปลง วงจรปรับปรุงสัญญาณ วงจรแหล่งจ่ายไฟกระแสตรง วงจรปรับความเร็วมอเตอร์ และพัดลมระบายอากาศ จากรูปซึ่งได้เชื่อมต่อวงจรทั้งหมดเรียบร้อยพร้อมที่จะนำไปใช้งาน

รูปที่ 3.24 แสดงอุปกรณ์ที่ใช้ทดลองในโครงงาน

จากรูปที่ 3.23 ได้แสดงอุปกรณ์ที่ใช้ในการทดลองของโครงงานนี้ ซึ่งประกอบไปด้วย ชุดประมวลผลสัญญาณ ที่ได้ประกอบบนแผ่นอะคริลิค เครื่องชั่งโหลดเซลล์ที่มีสายพานลำเลียงวิ่งอยู่ด้านบน เครื่องออสซิลโลสโครป (Oscilloscope) ฟังก์ชั่นเจนเนอเรเตอร์ (Sweep Function Generator) สายโพรบสำหรับออสซิลโลสโคป และสายสัญญาณสำหรับฟังก์ชั่นเจน

เมื่ออุปกรณ์ต่างๆครบแล้ว จากนั้นทำการทดลองตามที่ได้ออกแบบดังนี้

1. การทดลองวงจรแปลงแอนะล็อกเป็นดิจิตอล (A to D) และการวงจรแปลงดิจิตอลเป็นแอนะล็อก (D to A) เป็นการทดสอบตัวแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่ความถี่อัตราการสุ่มสัญญาณ 2000 เฮิร์ต และแปลงดิจิตอลเป็นแอนะล็อก เพื่อวัดสัญญาณเอาท์พุต

2. การทดลองวงจรกรองสัญญาณดิจิตอลโดยการใช้ตัวกรองเฉลี่ยแบบเคลื่อนที่ (Moving Average Filter) เป็นการทดสอบตัวกรองที่ใช้จำนวนข้อมูลที่ต่างกันที่ได้ออกแบบไว้คือ จำนวนข้อมูล 50, 100, 150 และ 200 ข้อมูล ตามลำดับ

รายละเอียดการทดลองจะข้อกล่าวในบทที่ 4 ต่อไป

3.9 ปรับปรุงและแก้ไขระบบ

ทำการปรับปรุงและแก้ไขระบบให้มีความถูกต้องในการประมวลผลสัญญาณให้มีประสิทธิภาพตามที่ต้องการ

3.10 จัดทำรูปเล่มรายงาน

ทำการวิเคราะห์ข้อมูลและขั้นตอนวิธีการต่างๆ ตั้งแต่การดำเนินงาน การทดลอง ผลการทดลองและวิเคราะห์ผลการทดลองรวมถึงทฤษฎีที่ใช้ในโครงงานนี้

บทที่ 4

ผลการทดลองและวิเคราะห์ผลการทดลอง

หลังจากการออกแบบและสร้างชุดการทดลองวงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง โดยมีเครื่องชั่งน้ำหนักที่มีสายพานลำเลียงอยู่ด้านบนและชุดประมวลผลสัญญาณที่ประกอบด้วยวงจรปรับปรุงสัญญาณและวงจรไมโครคอนโทรเลอร์ dsPIC30F4011 แล้ว ในบทนี้จะกล่าวถึงการทดลอง ผลการทดลองและวิเคราะห์ผลการทดลอง รวมถึงสรุปผลการทดลอง ซึ่งมีวิธีการทดลองที่ได้ออกแบบไว้แล้วดังนี้

1. การทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล (A to D) โดยการป้อนสัญญาณต่างๆ เพื่อทดสอบ จากนั้นทำการแปลงกลับโดยใช้วงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก (D to A) โดยจะใช้สัญญาณที่ได้กำหนดไว้ดังนี้

1.1 สัญญาณคลื่นไซน์ (Sine wave) ที่ความถี่ 50 เฮิรตซ์

1.2 สัญญาณสี่เหลี่ยม (Square wave) ที่ความถี่ 50 เฮิรตซ์

1.3 สัญญาณสามเหลี่ยม (Triangle wave) ที่ความถี่ 50 เฮิรตซ์

1.4 สัญญาณจากโหลดเซลล์ (Load Call Signal) ที่ความเร็วของสายพาน 0 เมตรต่อนาที

1.5 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 5 เมตรต่อนาที

1.6 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 10 เมตรต่อนาที

2. การทดลองวงจรกรองสัญญาณดิจิตอลโดยการใช้ตัวกรองเฉลี่ยแบบเคลื่อนที่ (Moving Average Filter) ที่จำนวนข้อมูล 50, 100, 150 และ 200 ข้อมูล โดยการป้อนสัญญาณจากโหลดเซลล์ที่ใช้น้ำหนักของมะม่วงที่ 390 กรัม ที่ความเร็วต่างๆ ดังนี้

2.1 สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที

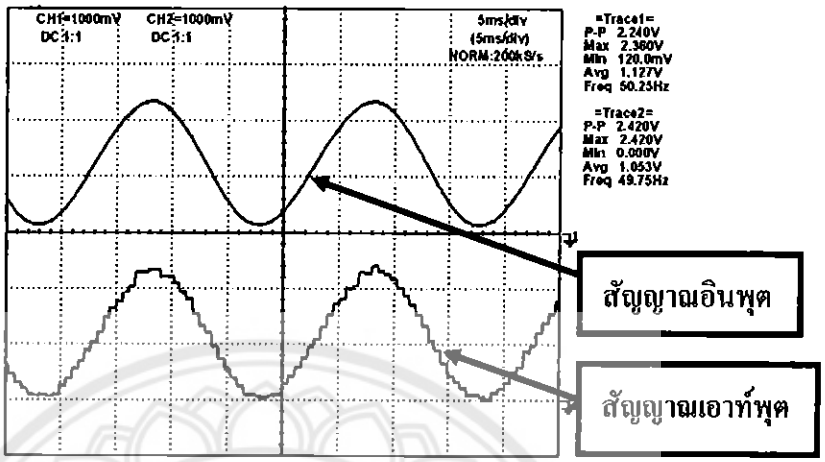
2.2 สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที

2.3 สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่มีความเร็วของสายพาน 10 เมตรต่อนาที

จากหัวข้อที่กล่าวมาข้างต้นสามารถอธิบายอย่างละเอียดได้ดังต่อไปนี้

4.1 การทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

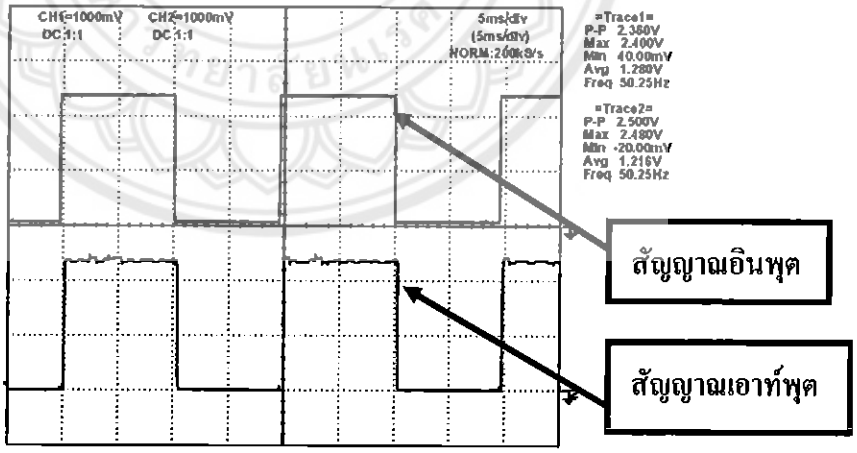
4.1.1 สัญญาณคลื่นไซน์ ความถี่ 50 เฮิรตซ์



รูปที่ 4.1 สัญญาณอินพุตและเอาท์พุตของสัญญาณคลื่นไซน์ที่ความถี่ 50 เฮิรตซ์

จากรูปที่ 4.1 ทำการป้อนสัญญาณคลื่นไซน์ ที่ความถี่ 50 เฮิรตซ์ เพื่อดูประสิทธิภาพของการแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ที่อัตราการสุ่มสัญญาณ 2000 เฮิรตซ์ จากนั้นทำการแปลงกลับจากสัญญาณดิจิตอลเป็นแอนะล็อกเพื่อวัดสัญญาณเอาท์พุตที่ได้

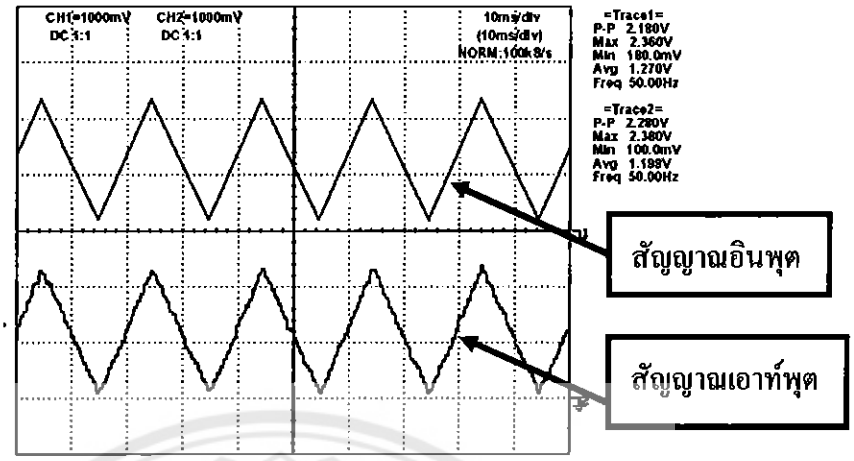
4.1.2 สัญญาณสี่เหลี่ยมความถี่ 50 เฮิรตซ์



รูปที่ 4.2 สัญญาณอินพุตและเอาท์พุตของสัญญาณสี่เหลี่ยมที่ความถี่ 50 เฮิรตซ์

จากรูปที่ 4.2 ทำการป้อนสัญญาณคลื่นสี่เหลี่ยม ที่ความถี่ 50 เฮิรตซ์ เพื่อดูประสิทธิภาพการแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่อัตราการสุ่มสัญญาณ 2000 เฮิรตซ์ จากนั้นทำการแปลงกลับจากสัญญาณดิจิตอลเป็นแอนะล็อกเพื่อวัดสัญญาณเอาท์พุตที่ได้

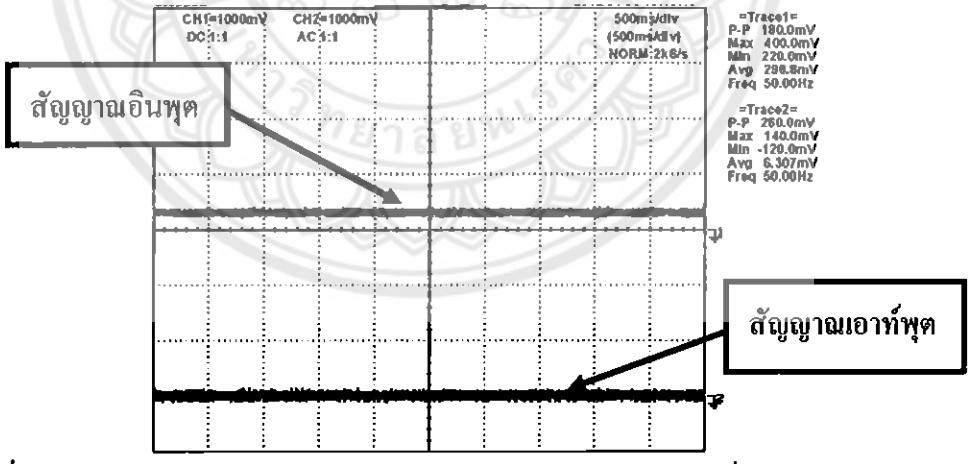
4.1.3 สัญญาณสามเหลี่ยม ความถี่ 50 เฮิรตซ์



รูปที่ 4.3 สัญญาณอินพุตและเอาท์พุตของสัญญาณสามเหลี่ยมที่ความถี่ 50 เฮิรตซ์

จากรูปที่ 4.3 ทำการป้อนสัญญาณคลื่นสามเหลี่ยม ที่มีขนาดความถี่ 50 เฮิรตซ์ เพื่อดูประสิทธิภาพการแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่อัตราการสุ่มสัญญาณ 2000 เฮิรตซ์ จากนั้นทำการแปลงกลับจากสัญญาณดิจิตอลเป็นแอนะล็อกเพื่อวัดสัญญาณเอาท์พุตที่ได้

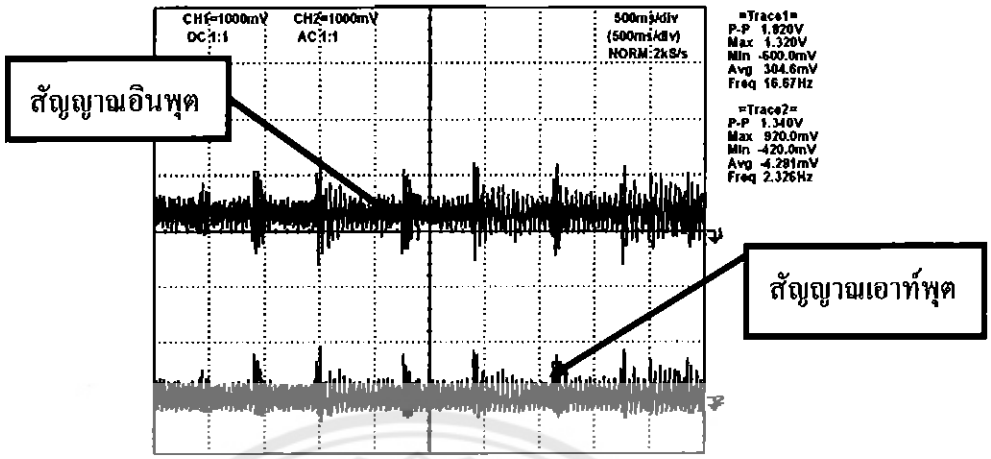
4.1.4 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 0 เมตรต่อนาที



รูปที่ 4.4 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 0 เมตรต่อนาที

จากรูปที่ 4.4 ทำการป้อนสัญญาณจากโหลดเซลล์ที่สายพานมีความเร็ว 0 เมตรต่อนาที เพื่อดูประสิทธิภาพการแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่อัตราการสุ่มสัญญาณ 2000 เฮิรตซ์ ที่มีสัญญาณรบกวน และทำการแปลงกลับจากดิจิตอลเป็นแอนะล็อกเพื่อวัดสัญญาณเอาท์พุตที่ได้

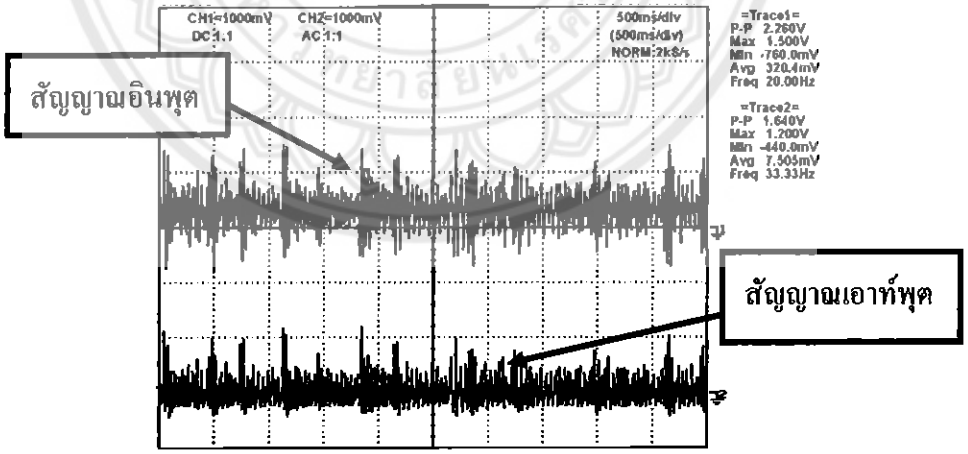
4.1.5 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 5 เมตรต่อนาที



รูปที่ 4.5 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 5 เมตรต่อนาที

จากรูปที่ 4.5 ทำการป้อนสัญญาณจากโหลดเซลล์ที่สายพานมีความเร็ว 5 เมตรต่อนาที เพื่อดูประสิทธิภาพการแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่อัตราการสุ่มสัญญาณ 2000 เฮิรตซ์ ที่มีสัญญาณรบกวน และทำการแปลงกลับจากดิจิตอลเป็นแอนะล็อกเพื่อวัดสัญญาณเอาท์พุตที่ได้

4.1.6 สัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 10 เมตรต่อนาที



รูปที่ 4.6 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่ความเร็วของสายพาน 10 เมตรต่อนาที

จากรูปที่ 4.6 ทำการป้อนสัญญาณจากโหลดเซลล์ที่สายพานมีความเร็ว 10 เมตรต่อนาที เพื่อดูประสิทธิภาพการแปลงสัญญาณแอนะล็อกเป็นดิจิตอลที่อัตราการสุ่มสัญญาณ 2000 เฮิรตซ์ ที่มีสัญญาณรบกวน และทำการแปลงกลับจากดิจิตอลเป็นแอนะล็อกเพื่อวัดสัญญาณเอาท์พุตที่ได้

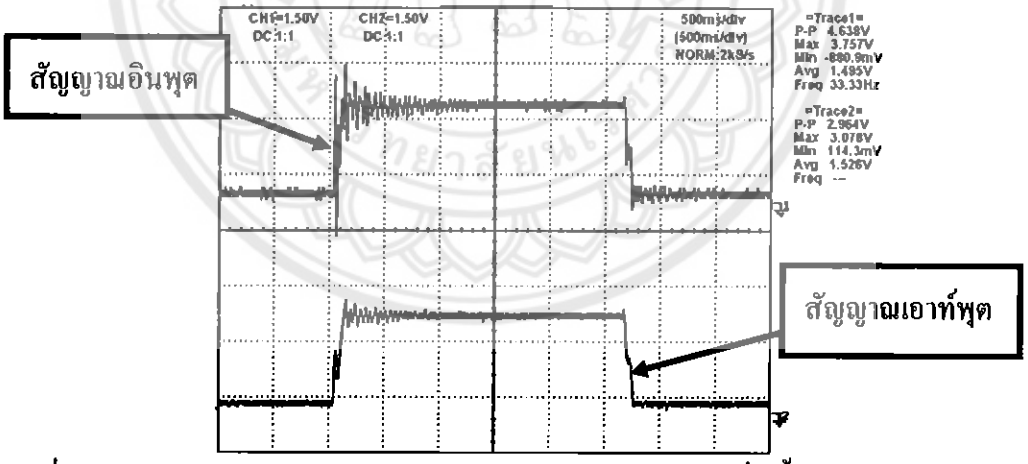
4.1.7 วิเคราะห์ผลการทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

จากการทดลองเมื่อป้อนคลื่นไซน์ความถี่ 50 เฮิรตซ์ จากรูปที่ 4.1 พบว่าเอาท์พุตที่ออกมาจะมีลักษณะเป็นคลื่นไซน์ที่เป็นขั้นบันได เกิดจากการสุ่มสัญญาณที่มีค่าน้อย มีค่าอัตราการสุ่มประมาณ 2000 เฮิรตซ์ จึงทำให้คลื่นไซน์ ที่ความถี่ 50 เฮิรตซ์ ใน 1 ลูกคลื่นจะมีข้อมูลแค่ 40 ข้อมูล จึงทำให้เห็นเป็นลักษณะขั้นบันได สัญญาณสี่เหลี่ยมและสัญญาณสามเหลี่ยม จากรูปที่ 4.2 และ 4.3 ก็เช่นเดียวกัน และเมื่อนำสัญญาณจากโหลดเซลล์ที่ความเร็วต่างๆ จากรูปที่ 4.4 ถึง 4.6 จะพบว่ามีสัญญาณรบกวนมาก ยิ่งความเร็วของสายพานมากสัญญาณรบกวนก็มากขึ้นตามลำดับ และเมื่อนำสัญญาณมาแปลงกลับเป็นสัญญาณแอนะล็อกที่จะทำให้สัญญาณรบกวนถูกกรองออกไปบางส่วนเนื่องจากการเลือกใช้อัตราการสุ่มสัญญาณที่น้อยคือ 2000 เฮิรตซ์ ทำให้สัญญาณที่ได้มาไม่ครบทุกความถี่

4.2 การทดลองวงจรกรองสัญญาณดิจิตอลโดยการใช้ตัวกรองเฉลี่ยแบบเคลื่อนที่

4.2.1 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 50 ข้อมูล

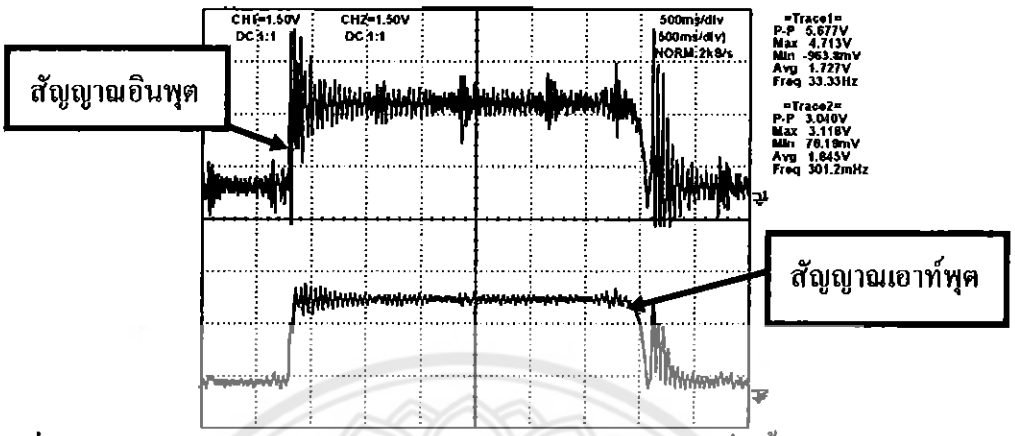
1. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที



รูปที่ 4.7 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที

จากรูปที่ 4.7 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 50 ข้อมูล จากนั้นใช้ออสซิลโลสโครป (Oscilloscope) ทำการวัดสัญญาณเอาท์พุตที่ได้

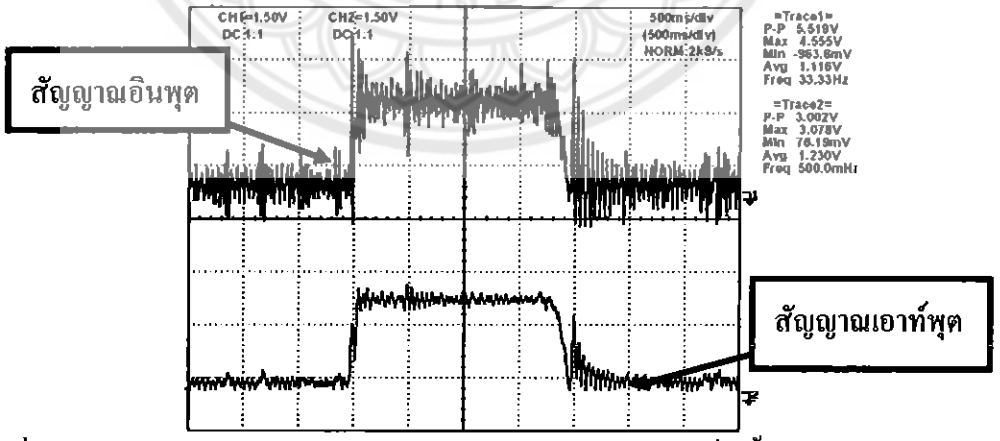
2. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที



รูปที่ 4.8 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที

จากรูปที่ 4.8 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพานที่ 5 เมตรต่อนาที โดยจะใช้วงจรกรองที่จำนวนข้อมูล 50 ข้อมูล จากนั้นใช้ออสซิลโลสโครป (Oscilloscope) ทำการวัดสัญญาณเอาท์พุตที่ได้

3. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที

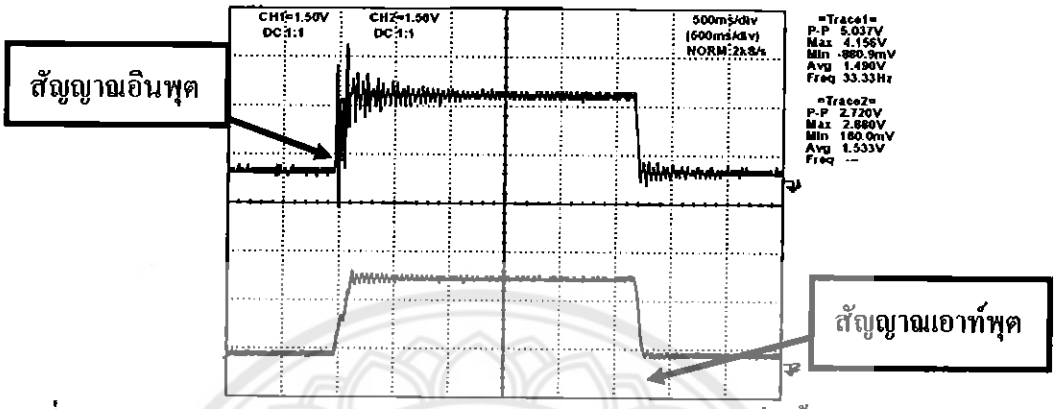


รูปที่ 4.9 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที

จากรูปที่ 4.9 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 50

4.2.2 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 100 ข้อมูล

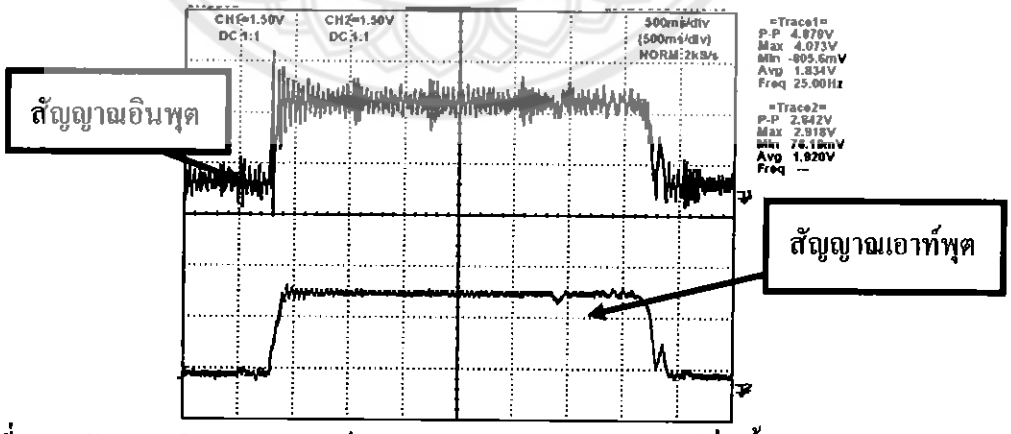
1. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที



รูปที่ 4.10 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที

จากรูปที่ 4.10 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 100 ข้อมูล จากนั้นใช้ออสซิลโลสโครป (Oscilloscope) ทำการวัดสัญญาณเอาท์พุตที่ได้

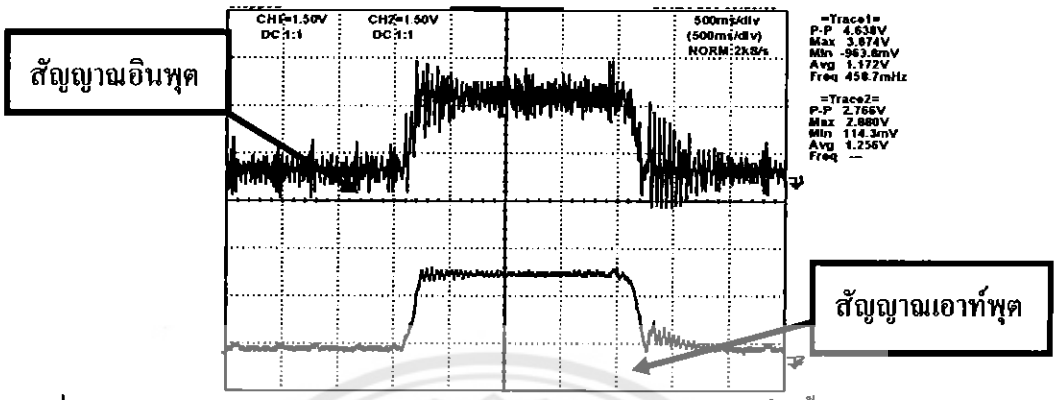
2. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที



รูปที่ 4.11 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที

จากรูปที่ 4.11 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 100 ข้อมูล

3. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที

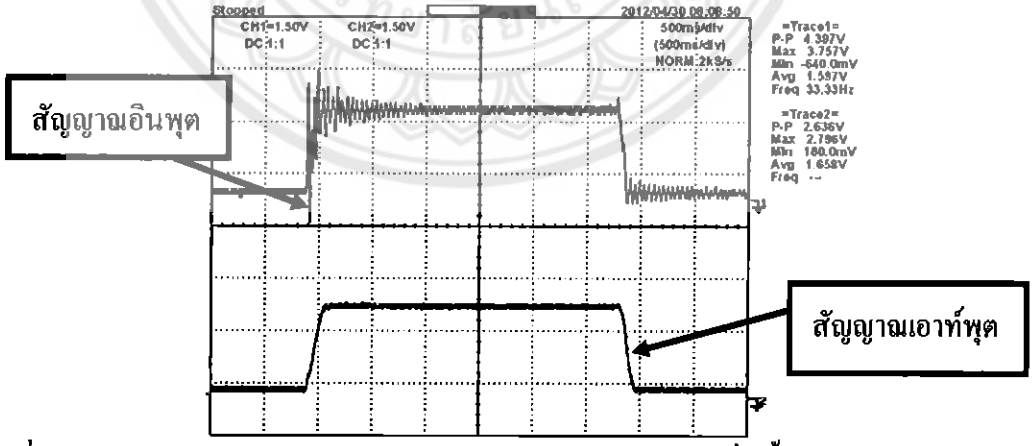


รูปที่ 4.12 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที

จากรูปที่ 4.12 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 100 ข้อมูล

4.2.1 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 150 ข้อมูล

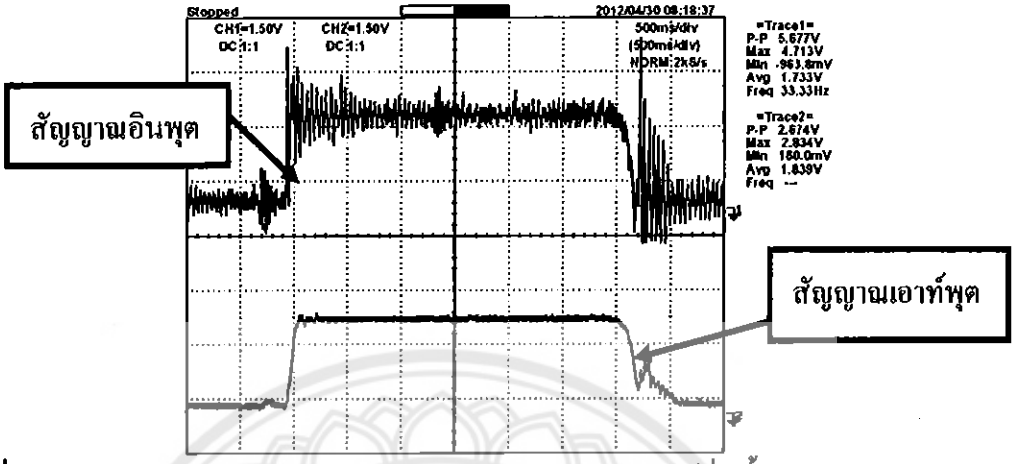
1. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที



รูปที่ 4.13 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที

จากรูปที่ 4.13 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 150 ข้อมูล

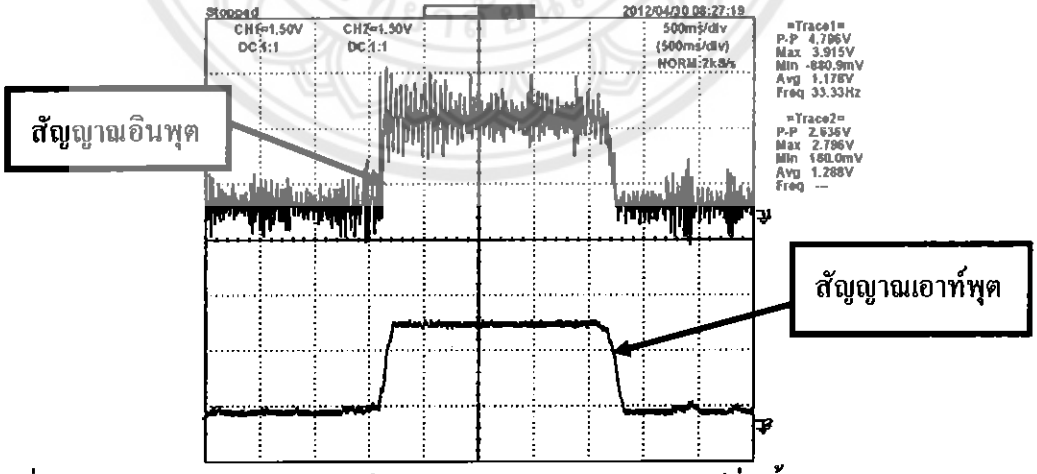
2. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที



รูปที่ 4.14 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที

จากรูปที่ 4.14 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 150 ข้อมูล

3. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที

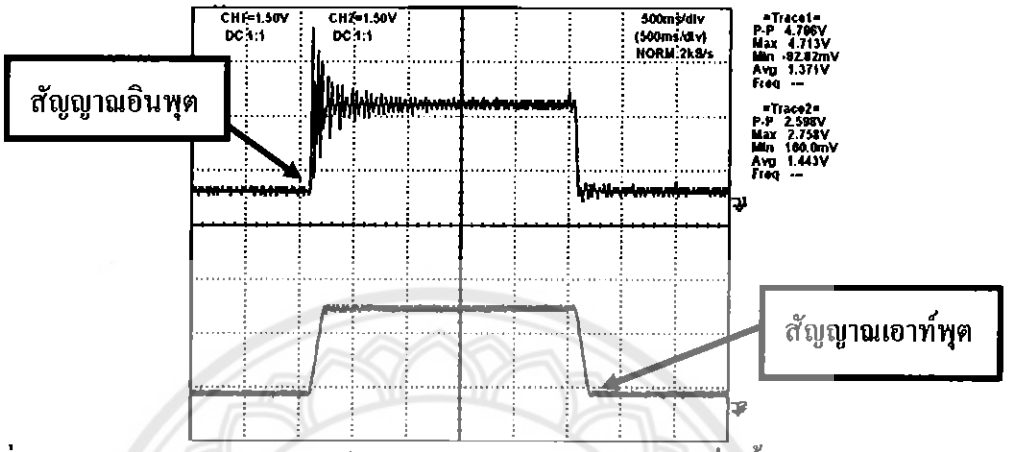


รูปที่ 4.15 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที

จากรูปที่ 4.15 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 150 ข้อมูล

4.2.1 ตัวกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 200 ข้อมูล

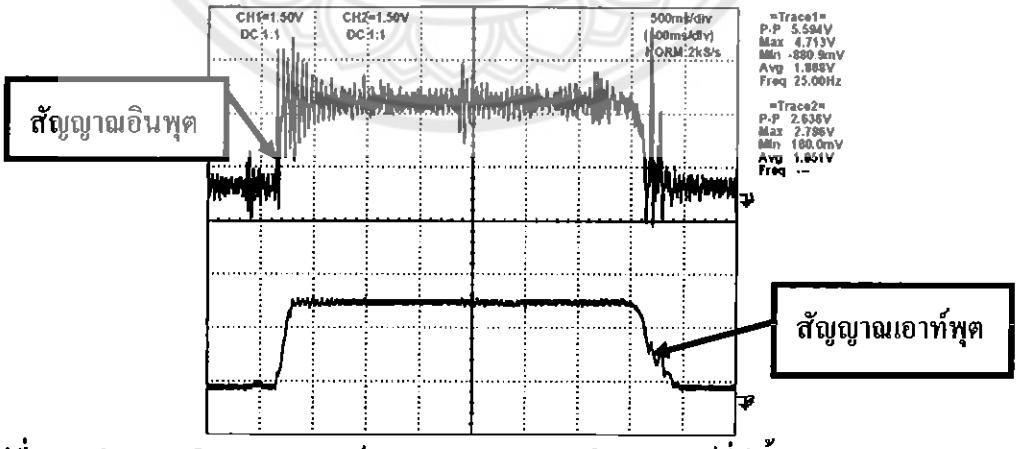
1. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที



รูปที่ 4.16 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที

จากรูปที่ 4.16 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 0 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 200 ข้อมูล

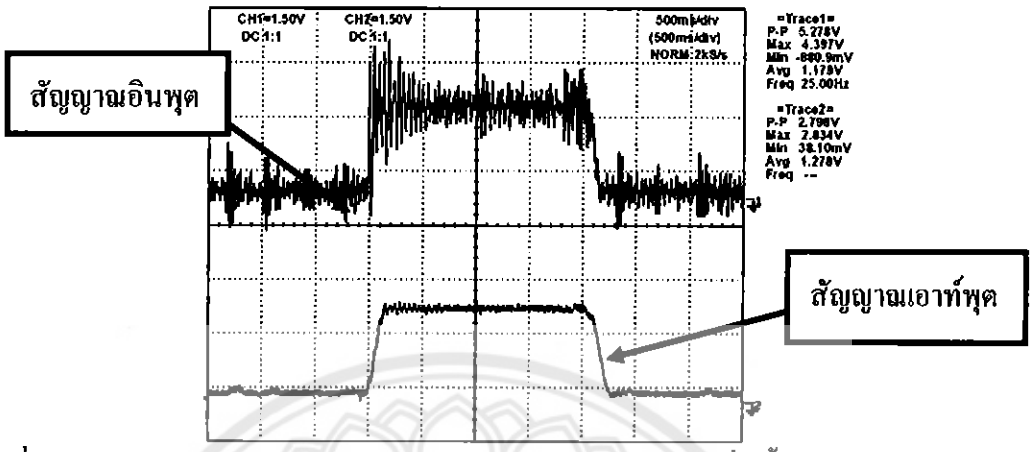
2. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที



รูปที่ 4.17 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที

จากรูปที่ 4.17 ทำการป้อนสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 5 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 200 ข้อมูล

3. สัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที



รูปที่ 4.18 สัญญาณอินพุตและเอาท์พุตของสัญญาณจากโหลดเซลล์ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที

จากรูปที่ 4.18 ทำการป้อนสัญญาณจากโหลดเซลล์ ที่มีน้ำหนักของมะม่วง 390 กรัม ที่ความเร็วของสายพาน 10 เมตรต่อนาที โดยใช้วงจรกรองที่จำนวนข้อมูล 200 ข้อมูล

4.2.5 วิเคราะห์ผลการทดลองวงจรกรองสัญญาณแบบดิจิตอล โดยวิธีการใช้ตัวกรองเฉลี่ยแบบเคลื่อนที่

จากการทดลองวงจรกรองที่จำนวนข้อมูล 50 ข้อมูล ดังรูปที่ 4.7 ถึง 4.9 พบว่าสามารถกรองสัญญาณรบกวนได้เพียงเล็กน้อย ทำให้ยังมีสัญญาณรบกวนเหลืออยู่ และสายพานที่ความเร็วเพิ่มมากขึ้นจะทำให้สัญญาณที่ไม่ได้ถูกกรองออกไปมีค่ามากขึ้น

จากการทดลองวงจรกรองที่จำนวนข้อมูล 100 ข้อมูล ดังรูปที่ 4.10 ถึง 4.12 พบว่าสามารถกรองสัญญาณรบกวนได้บางส่วน ทำให้ยังมีสัญญาณรบกวนเหลืออยู่แต่เมื่อเทียบกับการกรองที่จำนวนข้อมูล 50 ข้อมูล ถือว่ากรองสัญญาณได้ดีกว่า และสายพานที่ความเร็วเพิ่มมากขึ้นจะทำให้สัญญาณที่ไม่ได้ถูกกรองออกไปมีค่ามากขึ้น

จากการทดลองวงจรกรองที่จำนวนข้อมูล 150 ข้อมูล ดังรูปที่ 4.13 ถึง 4.15 พบว่าสามารถกรองสัญญาณรบกวนได้เรียบกว่าการกรองที่ผ่านมา ทำให้ไม่ค่อยเห็นสัญญาณรบกวนเหลืออยู่ แต่เมื่อเทียบกับการกรองที่จำนวนข้อมูล 50 และ 100 ข้อมูล ถือว่ากรองสัญญาณดีกว่ามาก แต่เมื่อเปิดมอเตอร์เพื่อขับสายพานที่ความเร็วมากขึ้น จะสังเกตว่าจะมีรูปสัญญาณรบกวนบางส่วนที่เกิดจากการสั่นของมอเตอร์เหลือเพียงเล็กน้อย

จากการทดลองวงจรกรองที่จำนวนข้อมูล 200 ข้อมูล ดังรูปที่ 4.16 ถึง 4.18 พบว่าสามารถกรองสัญญาณรบกวนได้ใกล้เคียงกับการกรองที่จำนวนข้อมูล 150 ทำให้ไม่ค่อยเห็น

สัญญาณรบกวนเหลืออยู่ แต่เมื่อเทียบกับการกรองที่จำนวนข้อมูล 50 และ 100 ข้อมูล ถือว่ากรองสัญญาณดีกว่ามาก และสามารถกรองสัญญาณรบกวนที่เกิดจากการสั่นของมอเตอร์ได้มากกว่าเมื่อเทียบกับการกรองที่จำนวนข้อมูล 150 ข้อมูล

4.3 สรุปผลการทดลอง

จากผลการทดลองการกรองสัญญาณที่จำนวนข้อมูลต่างๆ นั้นพบว่า ยิ่งจำนวนข้อมูลมากขึ้นการกรองสัญญาณรบกวนก็จะยิ่งดี แต่มีข้อเสียคือยิ่งใช้จำนวนข้อมูลมากความล่าช้าก็จะมากตามอย่างเช่นที่จำนวนข้อมูลที่ 50 ข้อมูล จะมีความล่าช้าอยู่ที่ 25 มิลลิวินาที ที่จำนวนข้อมูลที่ 100 ข้อมูลจะมีความล่าช้าอยู่ที่ 50 มิลลิวินาที ที่จำนวนข้อมูลที่ 150 ข้อมูล จะมีความล่าช้าอยู่ที่ 75 มิลลิวินาที และที่จำนวนข้อมูล 200 ข้อมูล จะมีความล่าช้าอยู่ที่ 100 มิลลิวินาที

ดังนั้นควรเลือกใช้จำนวนข้อมูลที่เหมาะสม โดยเลือกจำนวนข้อมูลที่ 150 ข้อมูล ถือว่าเหมาะสมกับโครงงานนี้ เพราะการกรองสัญญาณที่จำนวนข้อมูลที่ 150 ข้อมูล สามารถกรองสัญญาณรบกวนได้ดีกว่าการใช้จำนวนข้อมูลที่ 50, 100 ข้อมูล และดีพอๆ กับการกรองสัญญาณที่จำนวนข้อมูลที่ 200 ข้อมูล อีกทั้งความล่าช้าก็น้อยกว่า ประมาณ 25 มิลลิวินาที

บทที่ 5

สรุปผลและข้อเสนอแนะ

ในบทนี้ จะกล่าวถึงสรุปผลการทดลองการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง พร้อมทั้งนำเสนอปัญหาและข้อเสนอแนะต่างๆ ดังนี้

5.1 สรุปผลการดำเนินโครงงาน

โครงงานนี้เป็นการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียงโดยการรับสัญญาณจากโหลดเซลล์ (Load cell) จากนั้นทำการประมวลผลสัญญาณแอนะล็อกเพื่อให้ได้สัญญาณตามที่ต้องการเริ่มจากการรับสัญญาณจากโหลดเซลล์ที่มีค่าอยู่ในระดับมิลลิโวลต์ประมาณ 14 มิลลิโวลต์ ซึ่งไม่สามารถนำไปประมวลผลทางดิจิตอลได้ จึงทำการปรับสัญญาณให้ได้ในระดับสัญญาณที่เหมาะสมโดยการใช้วงจรปรับปรุงสัญญาณที่ประกอบไปด้วยวงจร 3 วงจรคือ วงจรขยายโดยการต่อวงจรแบบอินสตรูเมนท์ (Instrument Circuit) เพราะเป็นวงจรขยายความแตกต่างที่มีประสิทธิภาพสูงสามารถขยายได้หลายเท่า จากนั้นจะได้สัญญาณที่มีขนาดตามที่เหมาะสม แต่สัญญาณที่ได้จะมีขนาดของสัญญาณที่เกินมาเป็นผลมาจากค่าน้ำหนักของมอเตอร์ ชุดเฟือง และฐานรับน้ำหนักของมะม่วงหรืออื่นๆ ที่เกี่ยวข้อง จากสาเหตุข้างต้นสามารถแก้ไขโดยการใช้วงจรลบสัญญาณ (Substractor) เพื่อให้ขนาดของสัญญาณที่ไม่มีน้ำหนักของมะม่วงเริ่มต้นที่ประมาณ 0 โวลต์ เสร็จแล้วทำการขยายสัญญาณอีกทีโดยการใช้วงจรขยายแบบธรรมดาเพื่อเป็นการจำกัดสัญญาณให้มีขนาดประมาณ 0 – 5.5 โวลต์ เพราะว่าถ้าหากขนาดของสัญญาณเกิน 5.5 โวลต์ จะทำให้ไมโครคอนโทรลเลอร์ dsPIC30F4011 เสียหายได้ จากกระบวนการข้างต้นจะทำให้ได้สัญญาณมีขนาดตามที่ต้องการ เมื่อได้สัญญาณที่ต้องการแล้วจะเข้าสู่กระบวนการประมวลผลสัญญาณดิจิตอล การประมวลผลสัญญาณดิจิตอลจะเป็นขบวนการทางด้านซอฟต์แวร์โดยจะเป็นการเขียนโปรแกรมประมวลผลสัญญาณลงในไมโครคอนโทรลเลอร์ dsPIC30F4011 เริ่มจากการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล (A to D) จากนั้นทำการกรองสัญญาณดิจิตอลโดยใช้การกรองเฉลี่ยแบบเคลื่อนที่ (Moving Average Filter) ที่ได้ออกแบบไว้มีจำนวนข้อมูลคือ 50, 100, 150 และ 200 ข้อมูล ตามลำดับ จากนั้นสุดท้ายจะทำการแปลงสัญญาณดิจิตอลกลับเป็นสัญญาณแอนะล็อกโดยใช้วงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อก (D to A) เพื่อวัดสัญญาณที่ได้ต่อไป

5.2 ปัญหาและข้อเสนอแนะ

1. การสร้างตัวประมาณค่าน้ำหนักของมะม่วงขณะมีสายพานลำเลียงนั้นต้องใช้มอเตอร์ ซึ่งในการใช้มอเตอร์นั้น เมื่อทำการเดินเครื่องพบว่ามีสัญญาณรบกวนจากมอเตอร์ จะทำให้มีผลกระทบต่อระบบ

แนวทางแก้ไขคือนำตัวเก็บประจุมาต่อขนานระหว่างขั้วต่อกับตัวถังของมอเตอร์ทั้งสองด้าน เพื่อทำการกำจัดสัญญาณรบกวนที่เกิดจากมอเตอร์

2. การสร้างเครื่องชั่งน้ำหนักที่มีสายพานวิ่งอยู่ด้านบนบางจุดไม่แน่นและอาจเกิดการสัญญาณรบกวนที่เกิดจากเครื่องชั่งน้ำหนักในขณะที่ทำการทดลองมีสัญญาณรบกวนมากเช่นชุดเฟืองและล้อลูกปืนที่มีลักษณะหลวม

แนวทางแก้ไขคือใส่น็อตยึดให้กับชุดเฟืองเพื่อไม่ให้เกิดการสั่นขณะเดินมอเตอร์และใส่น้ำมันหล่อลื่นให้กับล้อลูกปืนเพื่อลดแรงเสียดทาน

5.3 แนวทางในการพัฒนาโครงงาน

จากการทดลอง ผู้ดำเนินโครงงานพบว่าการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับการประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียงโดยเลือกใช้ตัวกรองเฉลี่ยแบบเคลื่อนที่ ซึ่งสามารถกรองสัญญาณรบกวนได้อย่างมีประสิทธิภาพ

จากการที่ได้จัดทำของโครงงานนี้ ผู้ดำเนินโครงงานเห็นว่าสามารถนำไปประยุกต์ใช้กับตัวกรองแบบอื่นได้อีกเช่นวงจรกรองแบบ FIR, IIR และยังสามารถนำหลักการนี้ไปประยุกต์ใช้เพื่อสร้างตัวประมาณค่าน้ำหนักของวัตถุอื่นๆ เช่น การชั่งรถยนต์ขณะที่ติดเครื่องยนต์อยู่ ทำให้มีสัญญาณรบกวนเกิดขึ้นเนื่องจากการสั่นสะเทือนของเครื่องยนต์ เป็นต้น

ในโครงงานนี้ยังสามารถที่จะนำไปพัฒนาเพื่อแสดงผลเป็นค่าน้ำหนักบนจอแสดงผล โดยจะแสดงค่าน้ำหนักเป็นตัวเลขได้ เพื่อง่ายต่อการอ่านค่าน้ำหนัก

เอกสารอ้างอิง

[1] ประจิน พลังสันติกุล. (2551). **การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ dsPIC30F ด้วยคอมไพเลอร์ MPLAB C.** กรุงเทพฯ: บริษัท แอพซอฟต์เทค จำกัด.

[2] ทีมงานสมาร์ทเลิร์นนิ่ง.(2551). **เรียนรู้การใช้งาน Protel DXP Altium Designer 6.** กรุงเทพฯ: ห้างหุ้นส่วนสามัญสาร์ทเลิร์นนิ่ง.

[3] นคร ภักคีชาติ และชัยวัฒน์ ลิ้มพรจิตรวิไล. **คู่มือการทดลอง dsPIC Microcontroller เบื้องต้นด้วยโปรแกรมภาษา C กับ MPLAB C30.** กรุงเทพฯ: บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

[4] ลัญฉกร วุฒิสิทธิกุลกิจ. (2547). **พื้นฐานกรรมวิธีสัญญาณดิจิทัล.** กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

[5] นภัทร วัจนเทพินทร์. (2547). **วงจรไอซีและการประยุกต์ใช้งาน.** บริษัท สกายบุ๊กส์ จำกัด.

ภาคผนวก

ภาคผนวก ก
วิธีการติดตั้งโปรแกรม

วิธีการติดตั้งโปรแกรม

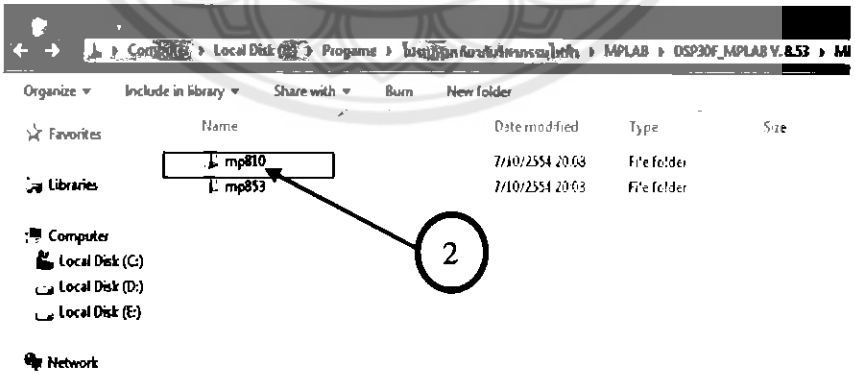
ในหัวข้อนี้จะกล่าวถึงวิธีการติดตั้งโปรแกรมที่ต้องใช้ในการทดลองโครงงานการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง ได้แก่ โปรแกรม MPLAB IDE V. 8.53 โปรแกรม MPLab_C30_3.01 และโปรแกรม PICkit 2 v2.61.00

1.1 วิธีการติดตั้งโปรแกรม MPLAB IDE V. 8.53

1. เริ่มต้นโดยการไปที่หน้าต่างที่มีโปรแกรมพร้อมทุกโปรแกรมที่ได้เตรียมไว้ให้ ดังรูปที่ 1.1 จากนั้นดับเบิลคลิกที่แฟ้มชื่อ "MPLAB V.8.53"

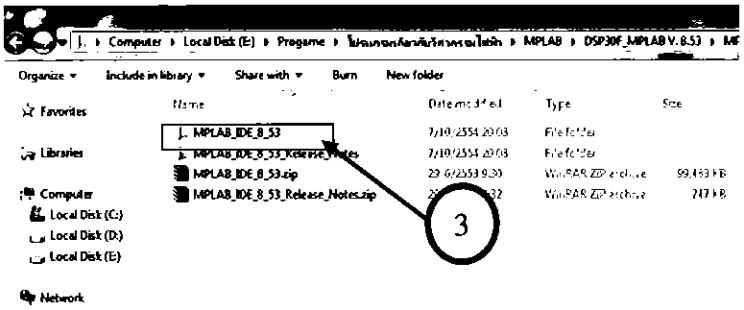
รูปที่ 1.1 โปรแกรมที่ต้องใช้ในการทดลองโครงงานการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง

2. ดับเบิลคลิกที่แฟ้ม (folder) ชื่อ "mp853" ดังรูปที่ 1.2



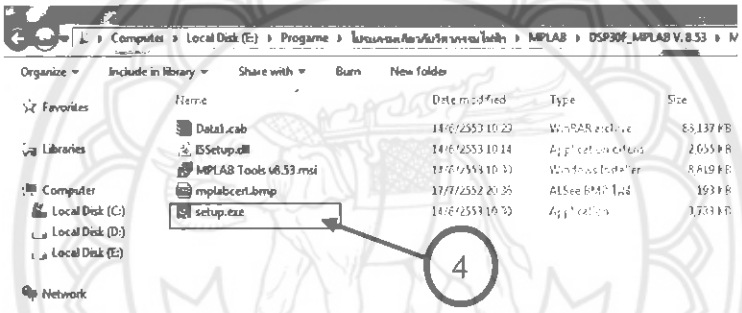
รูปที่ 1.2 แฟ้มชื่อ "mp853"

3. ดับเบิลคลิกที่ “MPLAB_IDE_8_53” ดังรูปที่ 1.3



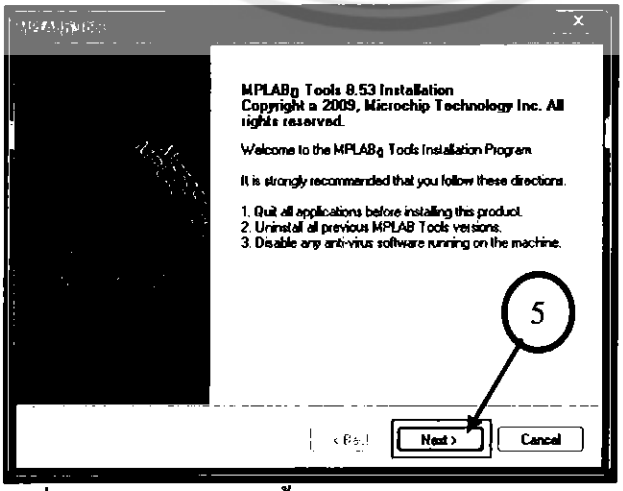
รูปที่ 1.3 โฟลเดอร์ “MPLAB_IDE_8_53”

4. ดับเบิลคลิกที่ “setup.exe” ดังรูปที่ 1.4



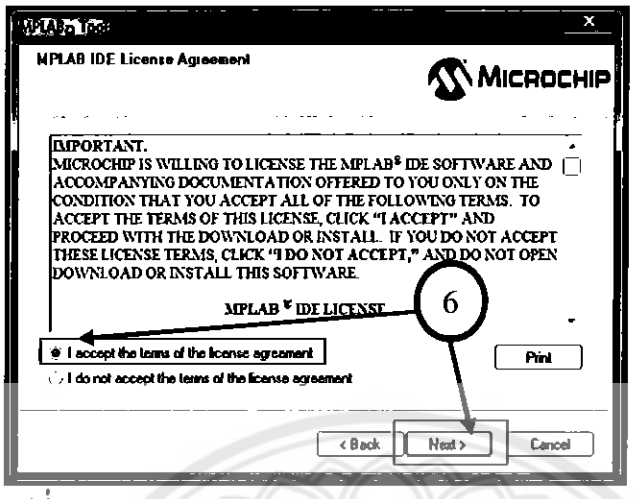
รูปที่ 1.4 “setup.exe”

5. หลังจากดับเบิลคลิกที่ “setup.exe” แล้วจะได้หน้าต่างการติดตั้ง ดังรูปที่ 1.5 จากนั้นคลิกปุ่ม “Next”



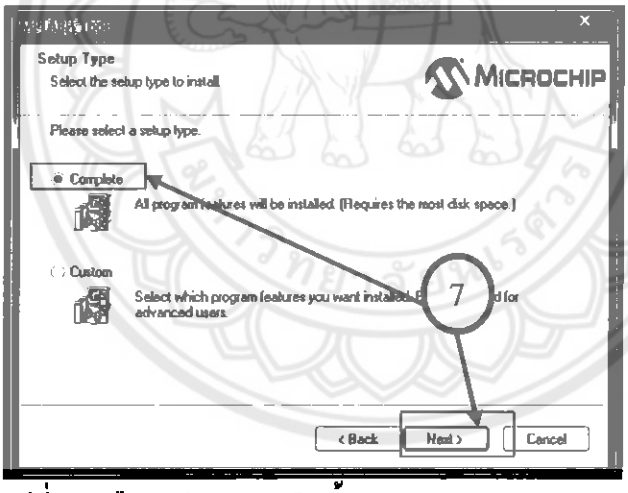
รูปที่ 1.5 หน้าต่างการติดตั้ง

6. คลิกเลือกที่ "I accept.." แล้วคลิกปุ่ม "Next" ดังรูปที่ 1.6



รูปที่ 1.6 "MPLAB IDE LICENSE"

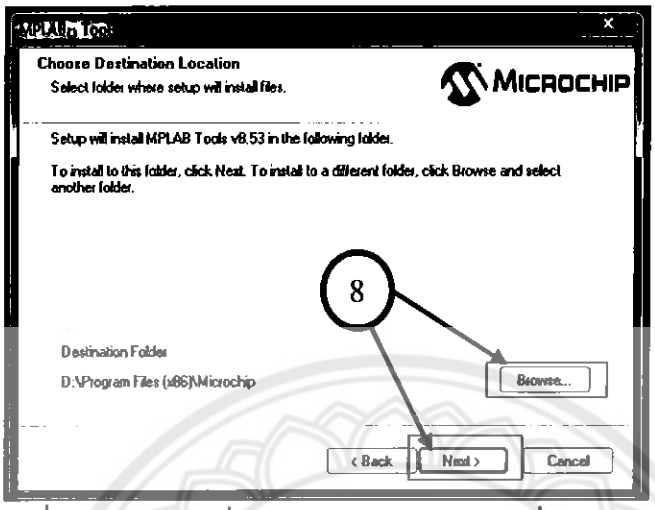
7. คลิกเลือกที่ "Complete" แล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.7



รูปที่ 1.7 เลือกรูปแบบการติดตั้งแบบ "Complete"

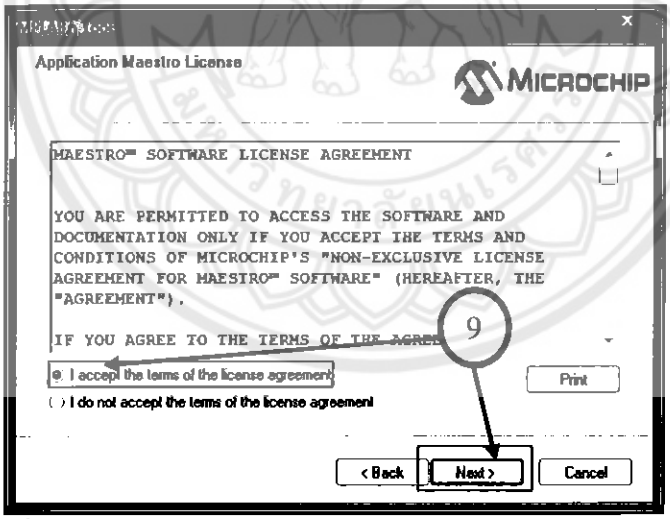
จากรูปที่ 1.7 สามารถเลือก การติดตั้งแบบ "Complete" เพื่อติดตั้งโปรแกรมที่มีมาพร้อมกับ "MPLAB Tools" หรือการติดตั้งแบบ "Custom" เพื่อเลือกโปรแกรมที่จะติดตั้งด้วยตัวเอง (แนะนำให้เลือกแบบการติดตั้งแบบ "Complete")

8. กำหนดที่อยู่แฟ้ม (folder) ของโปรแกรม ที่จะทำการติดตั้งโดยการเลือก “Browse…” เสร็จแล้วคลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.8



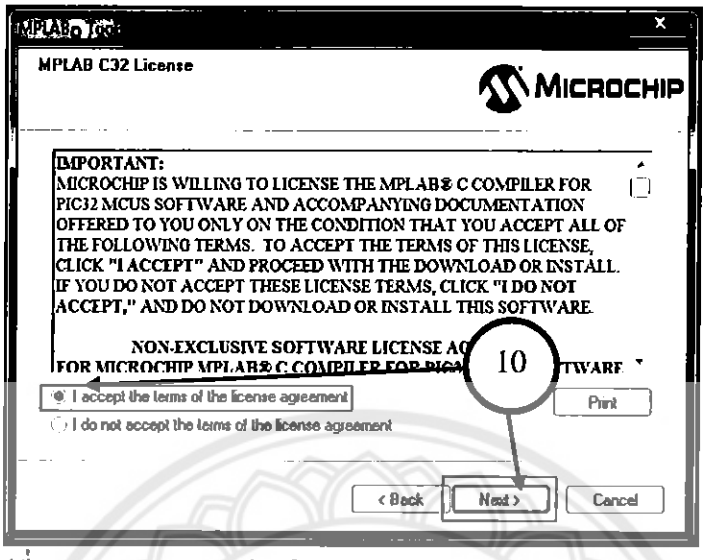
รูปที่ 1.8 . กำหนดที่อยู่ของแฟ้มของโปรแกรมที่จะทำการติดตั้ง

9. คลิกเลือกที่ “I accept..” แล้วคลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.9



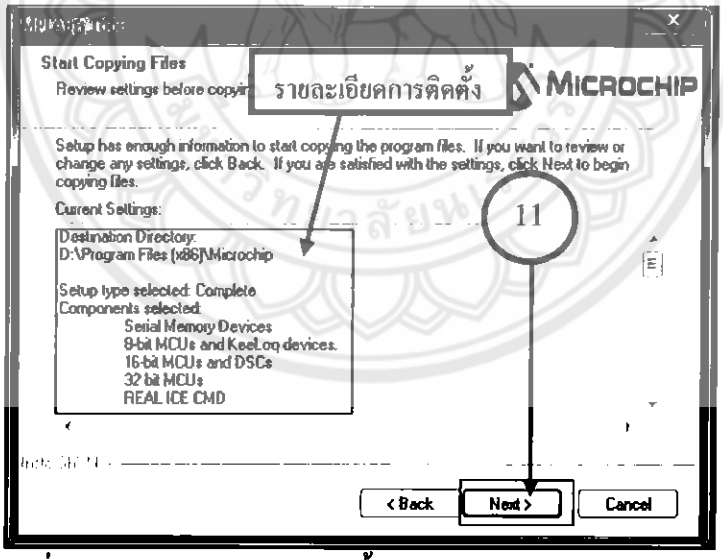
รูปที่ 1.9 แสดงรายละเอียดโปรแกรม “Maestro License”

10. คลิกเลือกที่ "I accept.." แล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.10



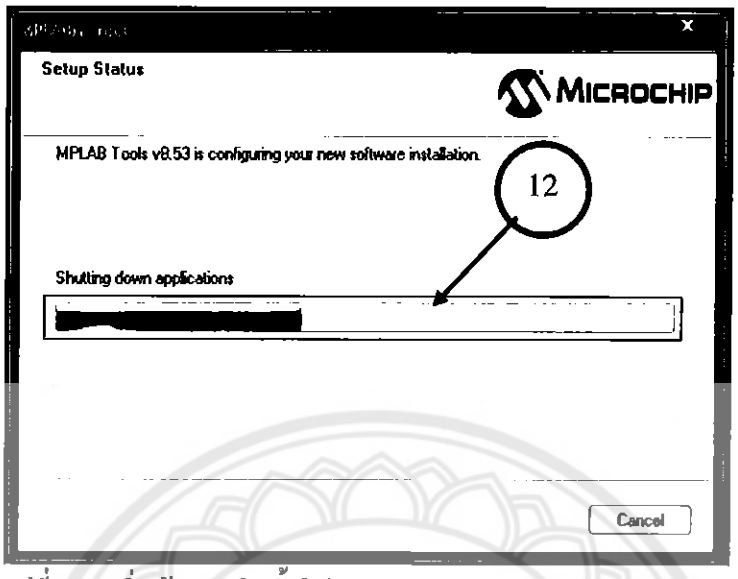
รูปที่ 1.10 แสดงรายละเอียดโปรแกรม "MPLAB C32 License"

11. แสดงรายละเอียดในการติดตั้ง แล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.11



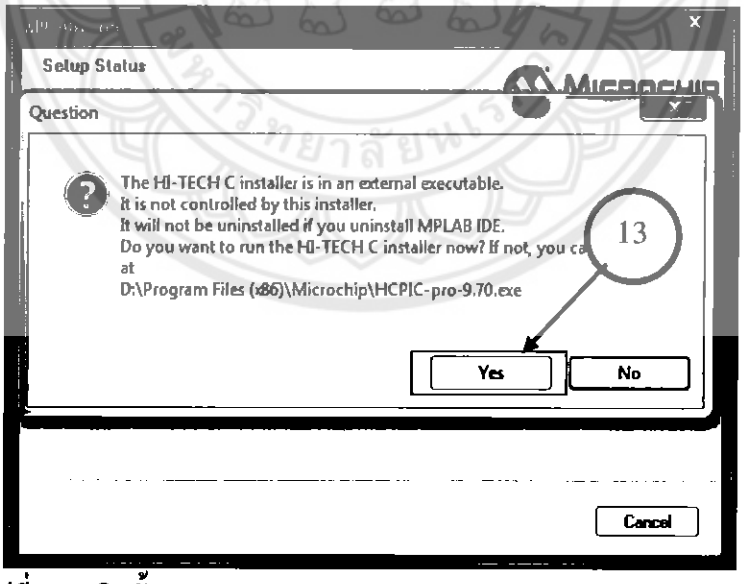
รูปที่ 1.11 รายละเอียดในการติดตั้ง

12. เริ่มต้นการติดตั้งโปรแกรม ดังรูปที่ 1.12



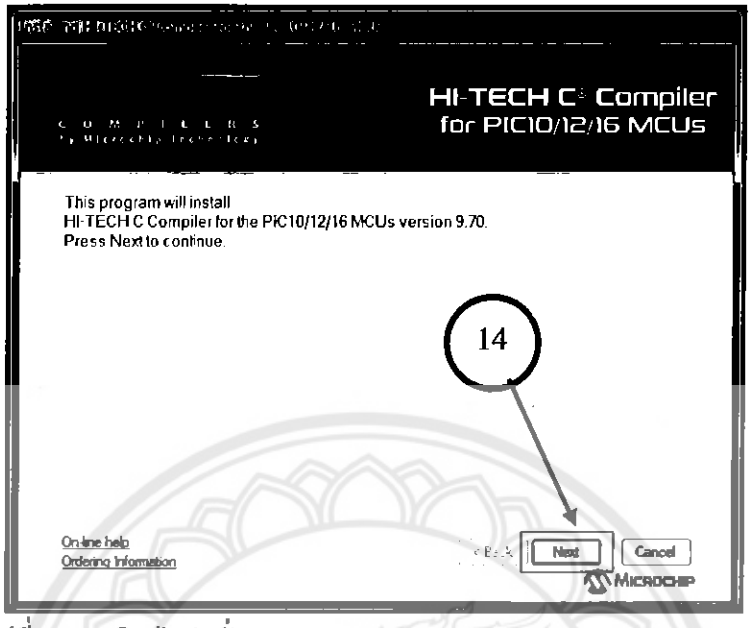
รูปที่ 1.12 เริ่มต้นการติดตั้งโปรแกรม

13. คลิกปุ่มคำสั่ง "Yes" เพื่อเริ่มการติดตั้ง "HI-TECH C" ดังรูปที่ 1.13



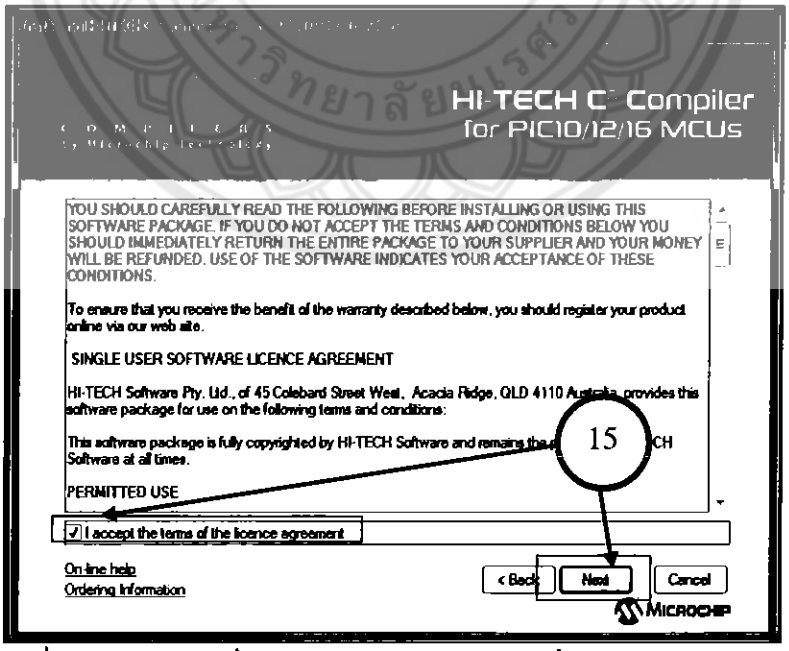
รูปที่ 1.13 ติดตั้ง "HI-TECH C"

14. คลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.14



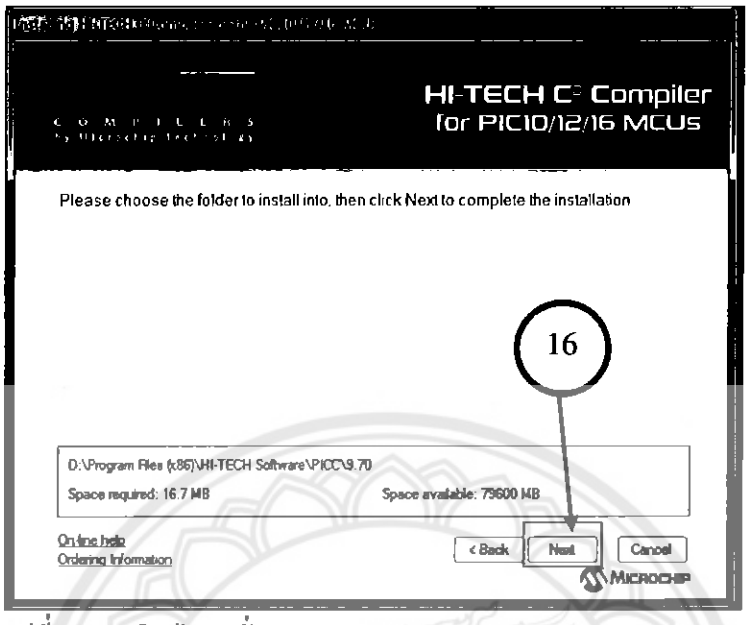
รูปที่ 1.14 คลิกปุ่มคำสั่ง “Next”

15. คลิกเลือกที่ “I accept..” แล้วคลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.15



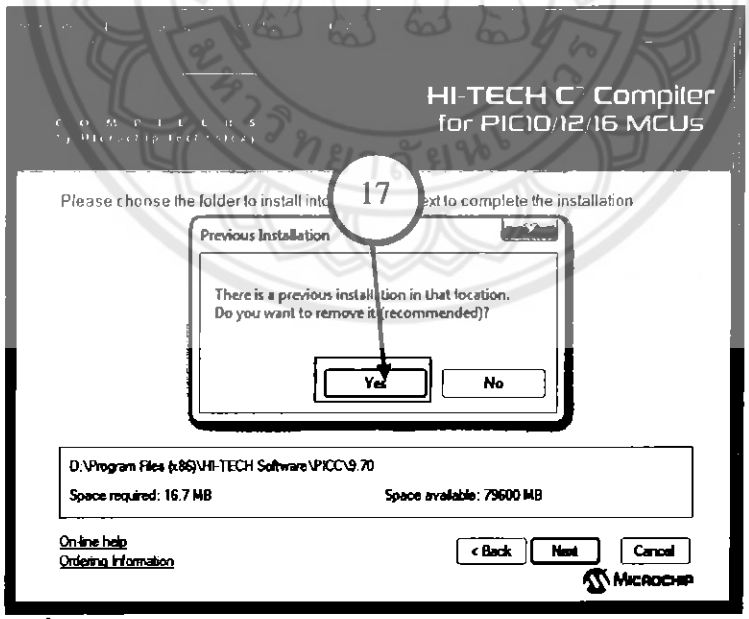
รูปที่ 1.15 คลิกเลือกที่ “I accept..” แล้วคลิกปุ่มคำสั่ง “Next”

16. คลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.16



รูปที่ 1.16 คลิกปุ่มคำสั่ง “Next”

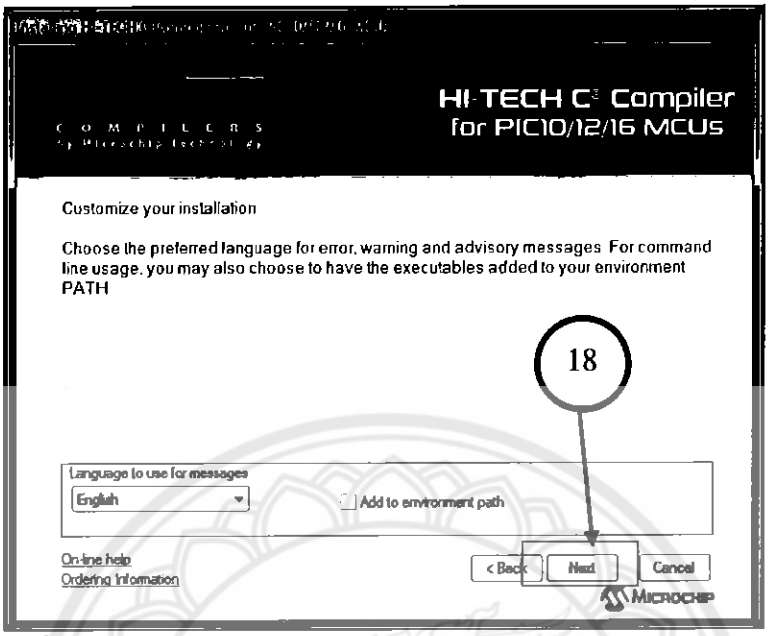
17. คลิกปุ่มคำสั่ง “Yes” ดังรูปที่ 1.17



รูปที่ 1.17 คลิกปุ่มคำสั่ง “Yes”

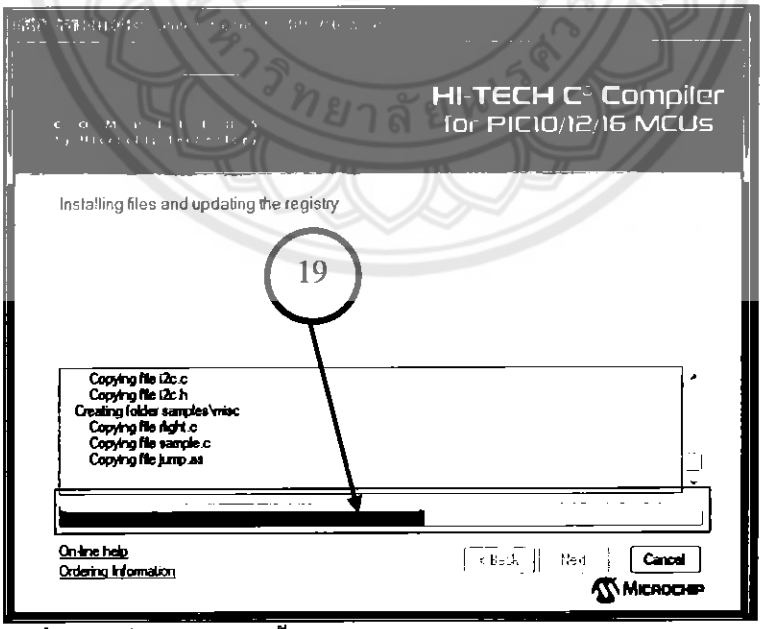
จากรูปที่ 1.17 ถ้าไม่มีหน้าต่างนี้ขึ้นให้ข้ามขั้นตอนนี้ไปได้เลย

18. คลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.18



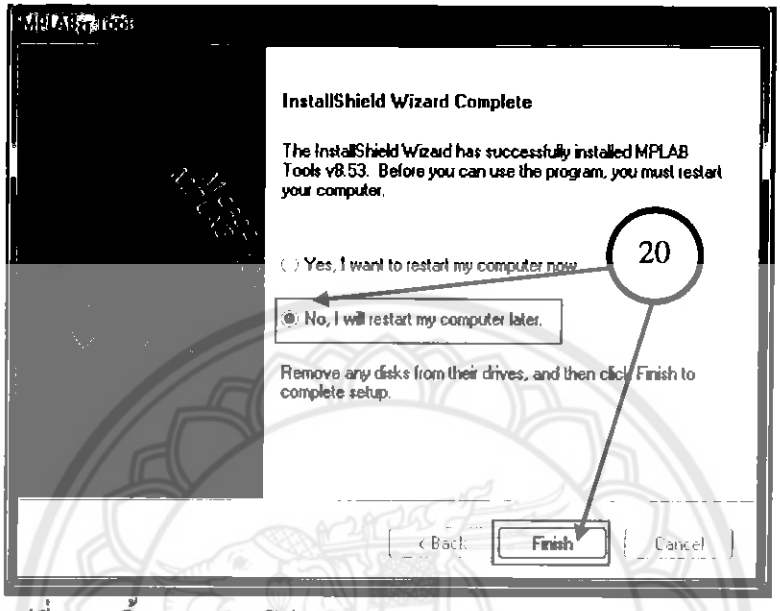
รูปที่ 1.18 คลิกปุ่มคำสั่ง "Next"

19. เริ่มต้นการติดตั้งโปรแกรม ดังรูปที่ 1.19



รูปที่ 1.19 เริ่มต้นการติดตั้งโปรแกรม

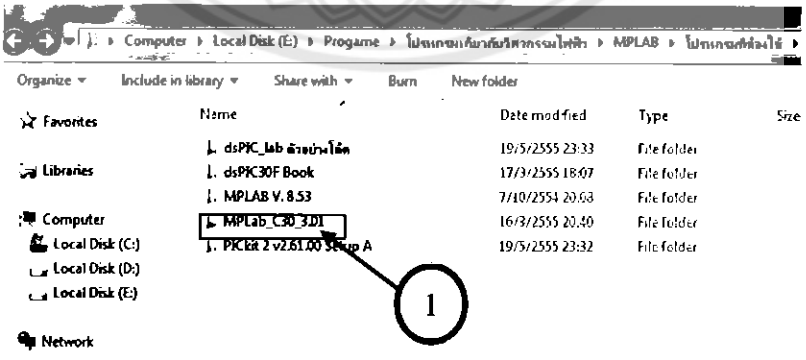
20. เลือก "NO" เนื่องจากจะทำการลงโปรแกรมทั้งสามโปรแกรมเสร็จก่อนแล้วค่อย "Restart" ทีเดียว จากนั้นคลิกเลือกปุ่มคำสั่ง "Finish" เพื่อสิ้นสุดการลงโปรแกรม "MPLAB IDE V8.53" ดังรูปที่ 1.20



รูปที่ 1.20 สิ้นสุดการลงโปรแกรม "MPLAB IDE V8.53"

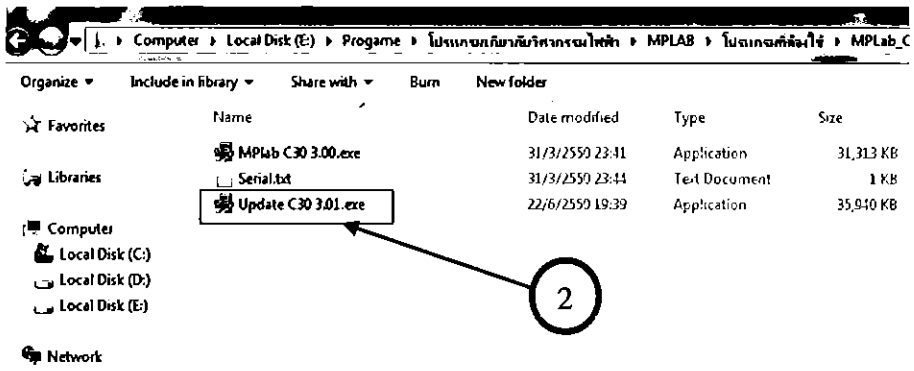
1.2 วิธีการติดตั้งโปรแกรม MPlab_C30_3.01

1. เริ่มต้นโดยการไปที่หน้าต่าง ที่มีโปรแกรมพร้อมทุกๆ โปรแกรม ที่ได้เตรียมไว้ให้ ดังรูปที่ 1.21 จากนั้นทำการดับเบิลคลิกที่แฟ้ม (folder) ชื่อ "MPlab_C30_3.01"



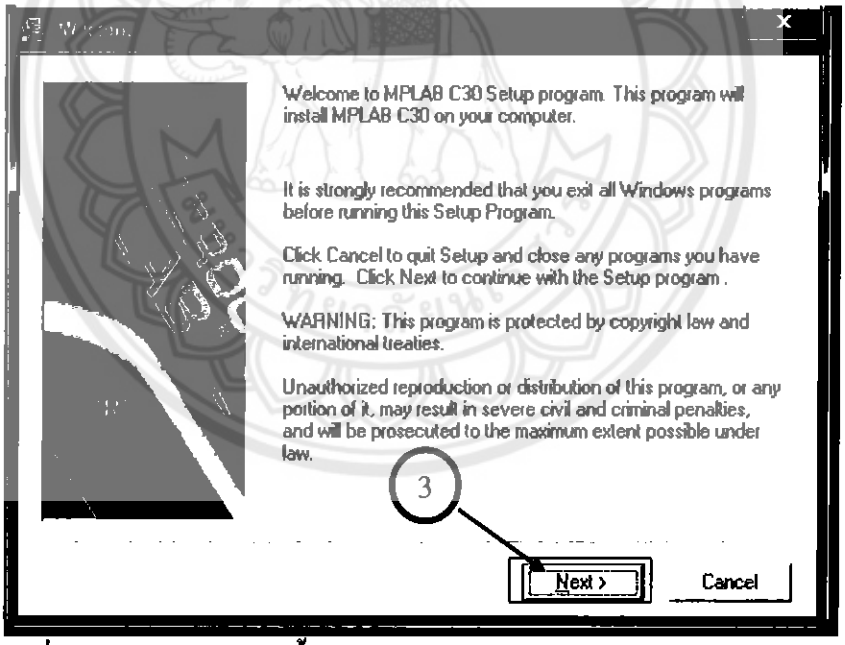
รูปที่ 1.21 แฟ้ม "MPlab_C30_3.01"

2. ดับเบิลคลิกที่แฟ้ม (folder) ชื่อ "Update C30 3.01.exe" ดังรูปที่ 1.22



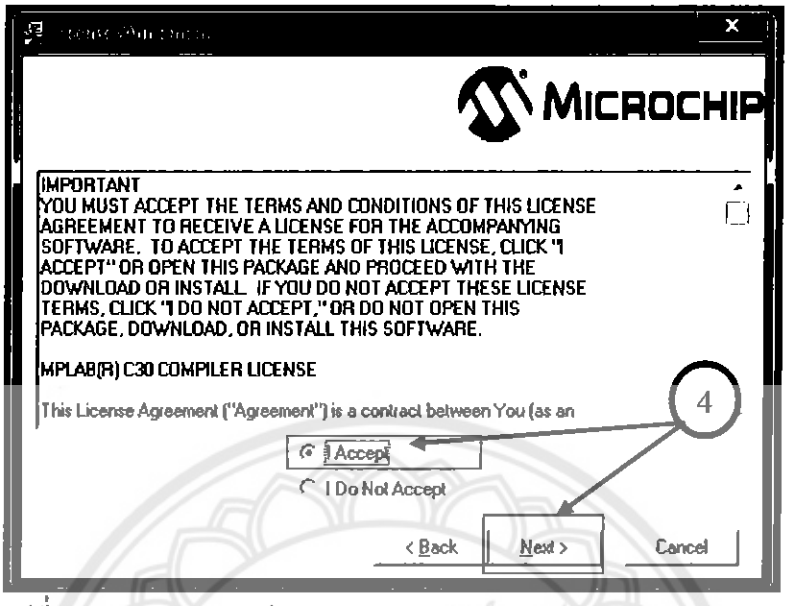
รูปที่ 1.22 แฟ้มชื่อ "Update C30 3.01.exe"

3. หลังจากดับเบิลคลิกที่ "Update C30 3.01.exe" แล้วจะได้หน้าต่างการติดตั้ง จากนั้นคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.23



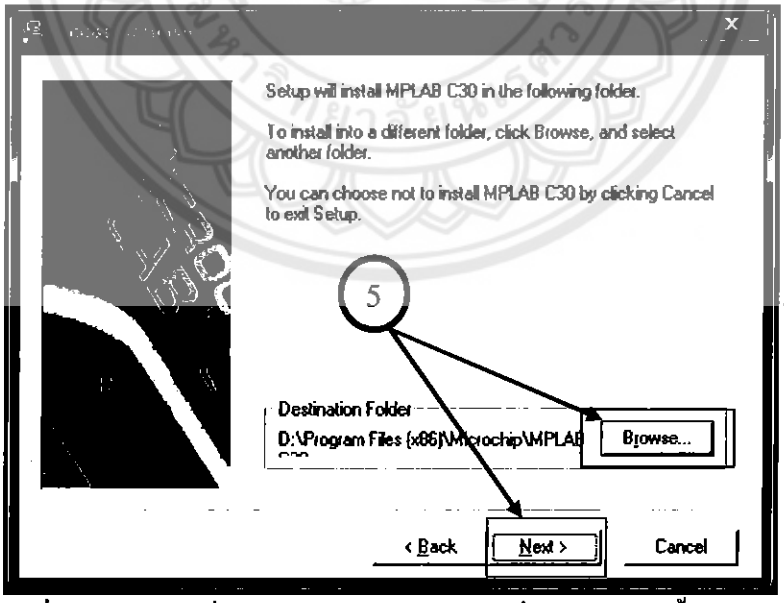
รูปที่ 1.23 หน้าต่างการติดตั้ง

4. คลิกเลือกที่ "I accept.." แล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.24



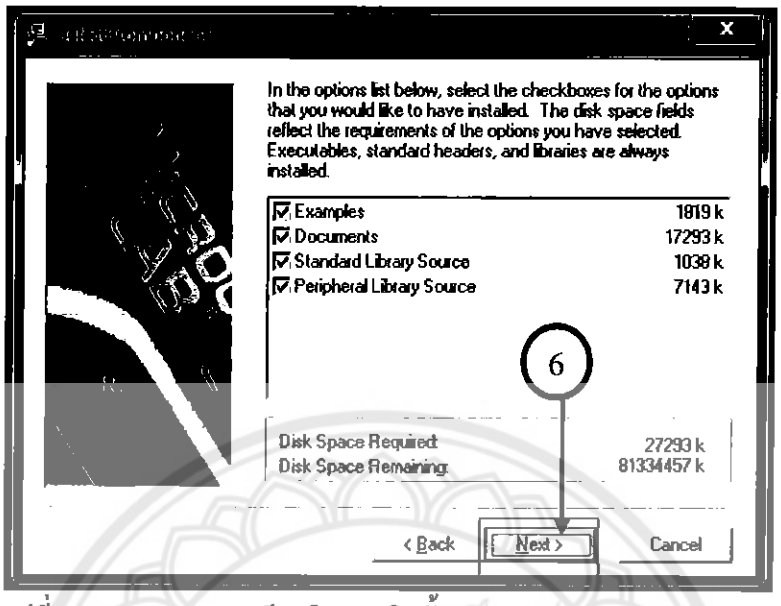
รูปที่ 1.24 แสดงรายละเอียด "License Agreement"

5. กำหนดที่อยู่แฟ้ม (folder) โปรแกรมที่จะทำการติดตั้งโดยการเลือก "Browse..." เสร็จแล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.25



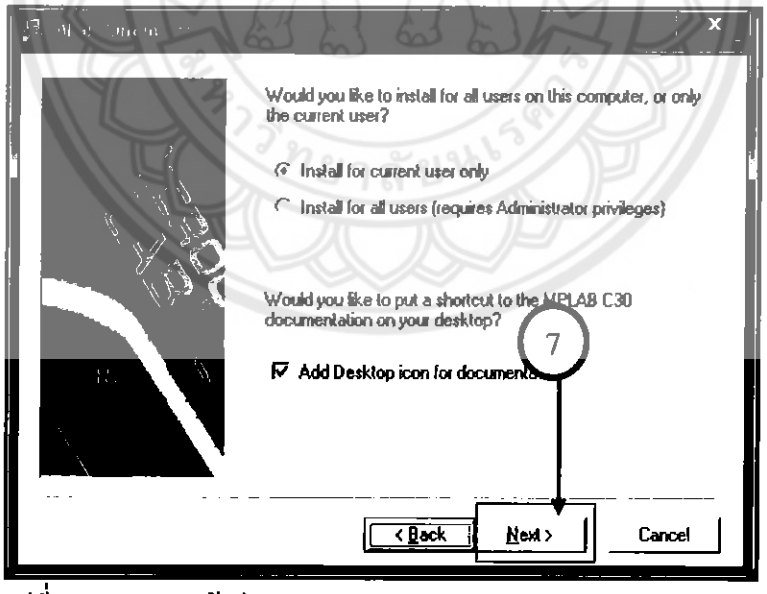
รูปที่ 1.25 กำหนดที่อยู่ของแฟ้มของโปรแกรมที่จะทำการติดตั้ง

6. แสดงรายละเอียดในการติดตั้ง แล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.26



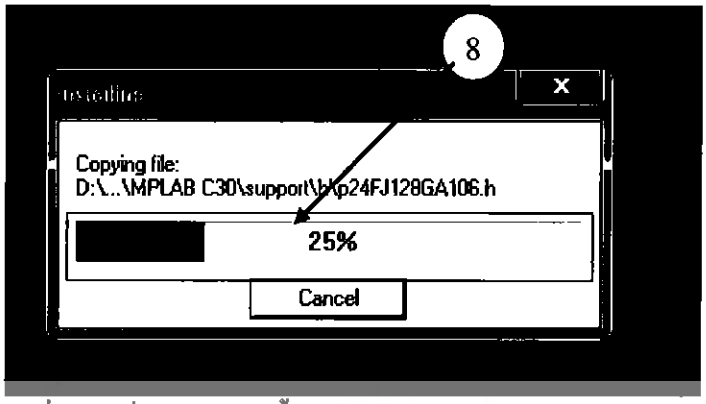
รูปที่ 1.26 แสดงรายละเอียดในการติดตั้ง

7. คลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.27



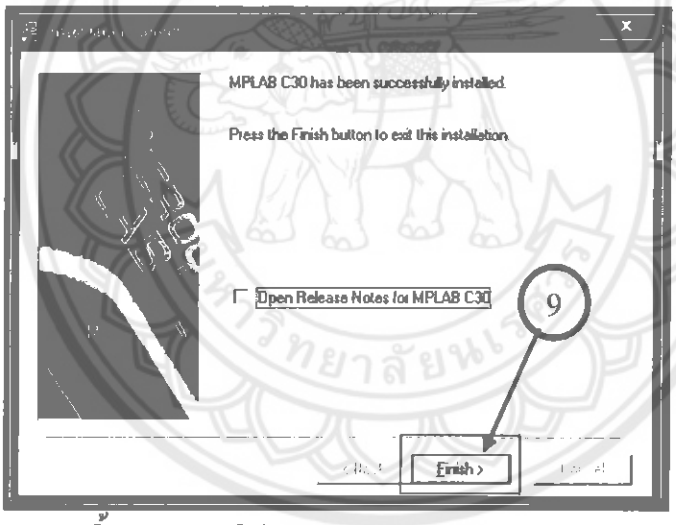
รูปที่ 1.27 แสดงหน้าต่าง "All or Current User"

8. เริ่มต้นการติดตั้งโปรแกรม ดังรูปที่ 1.28



รูปที่ 1.28 เริ่มต้นการติดตั้งโปรแกรม

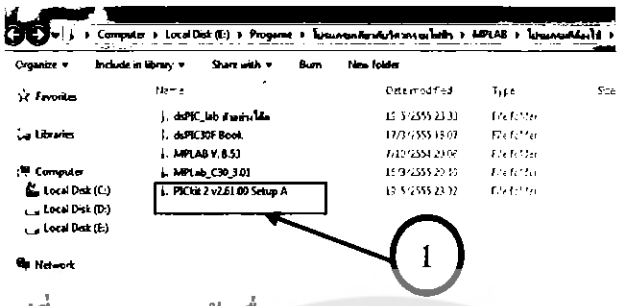
9. คลิกปุ่มคำสั่ง "Finish" เพื่อสิ้นสุดการลงโปรแกรม "MPlab_C30_3.01" ดังรูปที่ 1.29



รูปที่ 1.29 แสดงสิ้นสุดการลงโปรแกรม "MPlab_C30_3.01"

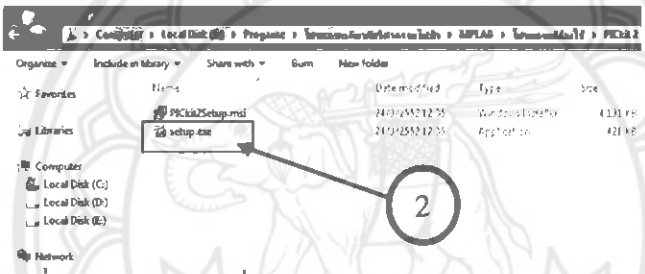
1.3 วิธีการติดตั้งโปรแกรม PICkit 2 v2.61

1. เริ่มต้นโดยการไปที่หน้าต่าง ที่มีโปรแกรมพร้อมทุกๆโปรแกรมที่เตรียมไว้ให้ ดังรูปที่ 1.30 จากนั้นดับเบิลคลิกที่แฟ้ม (folder) ชื่อ "PICkit 2 v2.61.00 Setup A"



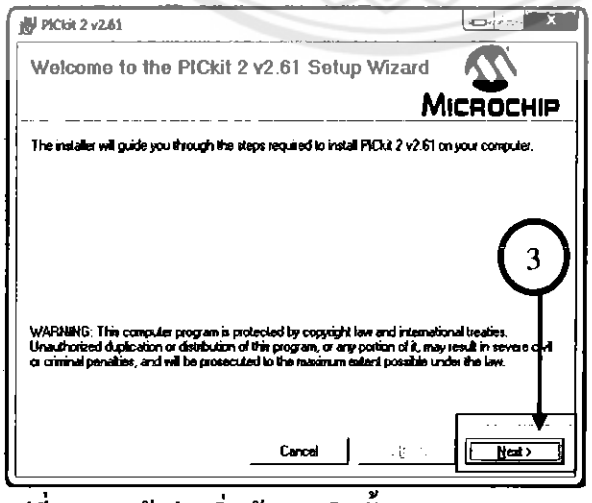
รูปที่ 1.30 แสดงแฟ้มชื่อ "PICkit 2 v2.61.00 Setup A"

2. ดับเบิลคลิกที่แฟ้มชื่อ "setup.exe" ดังรูปที่ 1.31



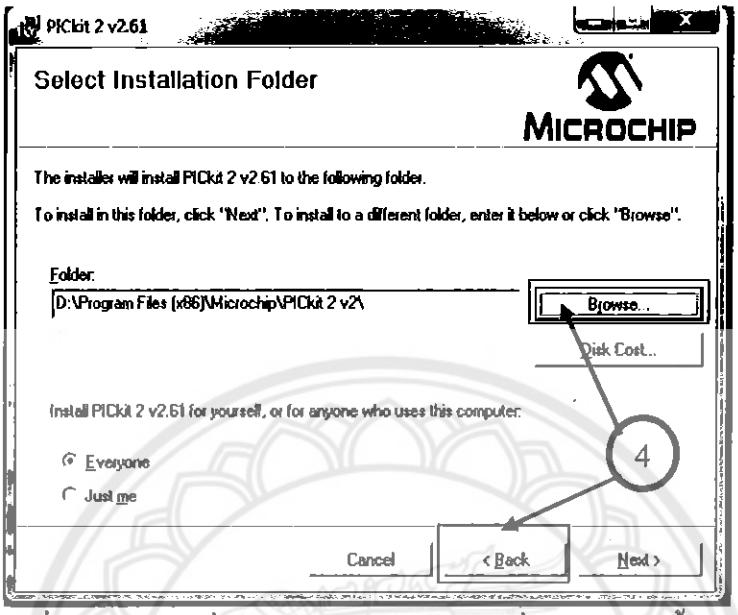
รูปที่ 1.31 แสดงแฟ้มชื่อ "setup.exe"

3. หลังจากดับเบิลคลิกที่ "setup.exe" แล้วจะได้หน้าต่างเริ่มต้นการติดตั้งขึ้นมา จากนั้นก็คลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.32



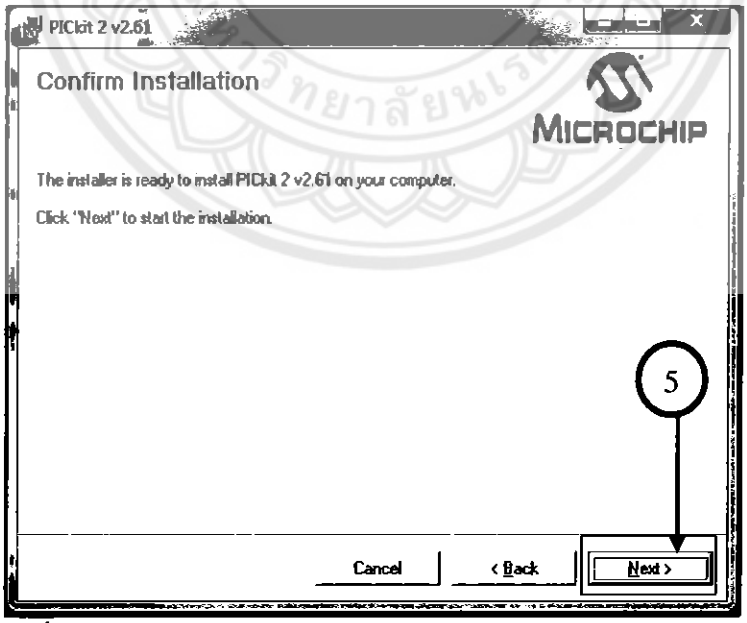
รูปที่ 1.32 หน้าต่างเริ่มต้นการติดตั้ง

4. กำหนดที่อยู่แฟ้ม (folder) ของโปรแกรมที่จะทำการติดตั้งโดยการเลือก “Browse…” เสร็จแล้วคลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.33



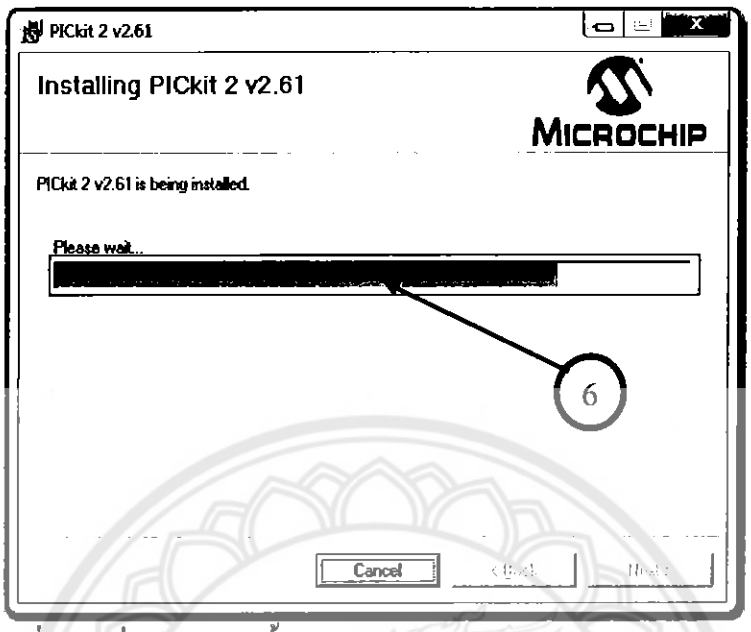
รูปที่ 1.33 กำหนดที่อยู่ของแฟ้มของโปรแกรมที่จะทำการติดตั้ง

5. คลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.34



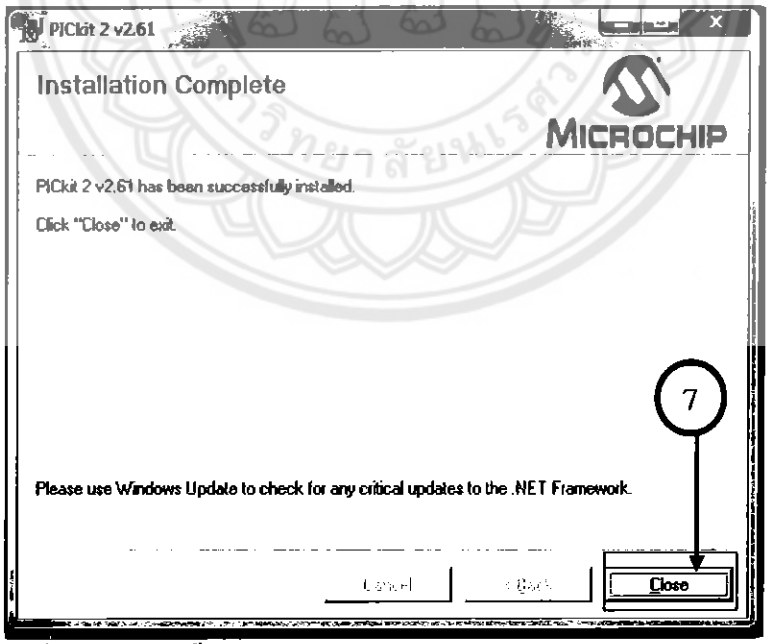
รูปที่ 1.34 แสดงการยืนยันในการลงโปรแกรม

6. เริ่มต้นการติดตั้งโปรแกรม ดังรูปที่ 1.35



รูปที่ 1.35 เริ่มต้นการติดตั้งโปรแกรม

7. คลิกปุ่มคำสั่ง "Close" เพื่อสิ้นสุดการลงโปรแกรม "PICkit 2 v2.61" ดังรูปที่ 1.36



รูปที่ 1.36 แสดงสิ้นสุดการลงโปรแกรม "PICkit 2 v2.61"

ภาคผนวก ข

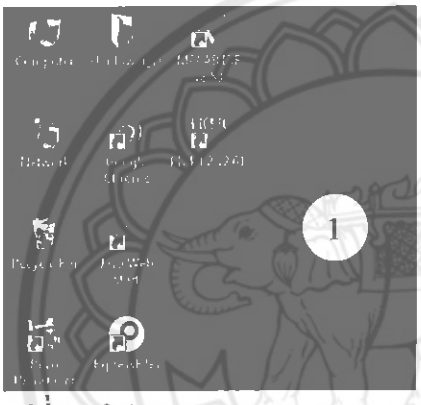
ขั้นตอนการสร้างโปรเจ็กต์ใน MPLAB C Compiler และการสร้างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่

ขั้นตอนการสร้างโปรเจ็กต์ใน MPLAB C Compiler และการสร้างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่

จากหัวข้อที่แล้วจะเป็นวิธีการลงโปรแกรมที่ใช้ในการทดลองประกอบไปด้วยสามโปรแกรมด้วยกัน และในหัวข้อนี้จะกล่าวถึงขั้นตอนการสร้างโปรเจ็กต์ใน MPLAB C Compiler และการสร้างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่เพื่อทำการทดลอง โดยจะยกตัวอย่างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 50 ข้อมูล

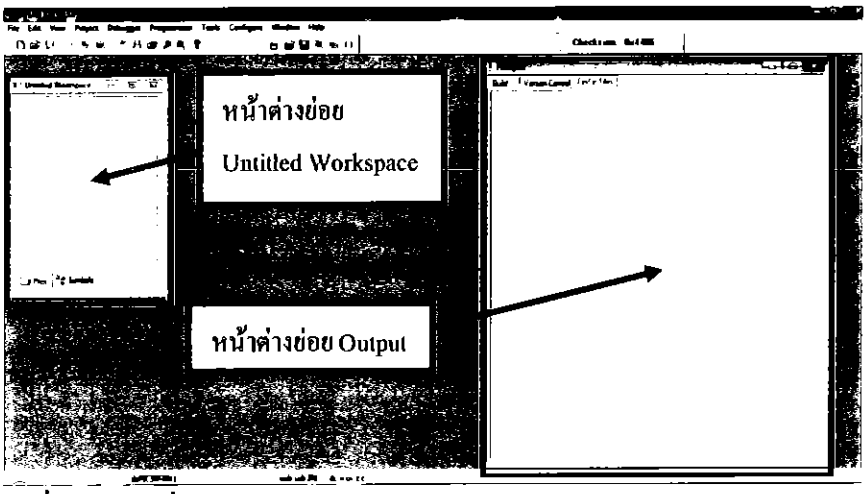
1.1 ขั้นตอนการสร้างโปรเจ็กต์ใน MPLAB C Compiler

1. ดับเบิลคลิกเรียกใช้งานโปรแกรม "MPLAB IDE v8.53" ดังรูปที่ 1.1



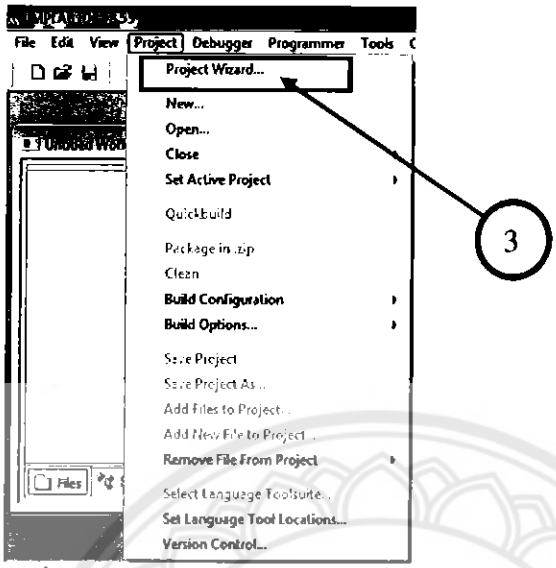
รูปที่ 1.1 โปรแกรม "MPLAB IDE v8.53"

2. หลังจากดับเบิลคลิกเรียกใช้งานโปรแกรม "MPLAB IDE v8.53" แล้วจะได้รับหน้าต่างของโปรแกรม "MPLAB IDE" ดังรูปที่ 2.2ซึ่งประกอบด้วยสองหน้าต่างย่อยคือ Untitled Workspace กับหน้าต่าง Output ดังแสดงไว้ในรูปที่ 1.2



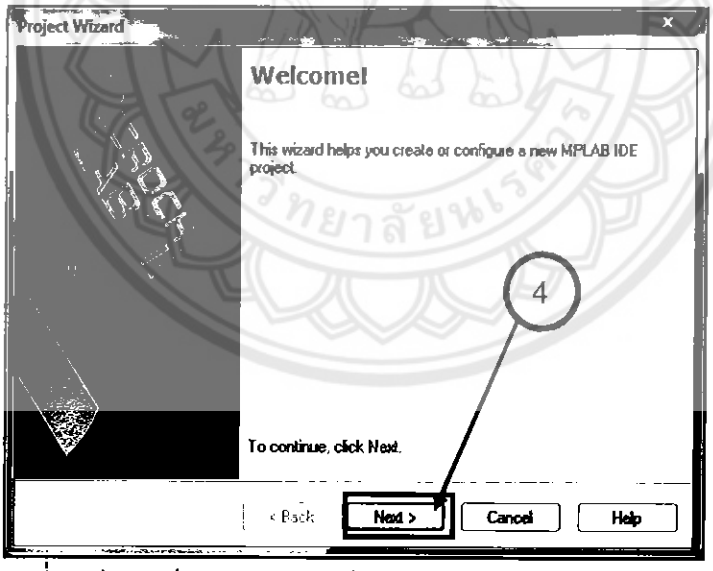
รูปที่ 1.2 หน้าเมื่อ Run โปรแกรม MPLAB IDE

3. ไปที่แถบเมนูแล้วคลิก “Project” เลือก “Project Wizard…” แสดงในรูปที่ 1.3



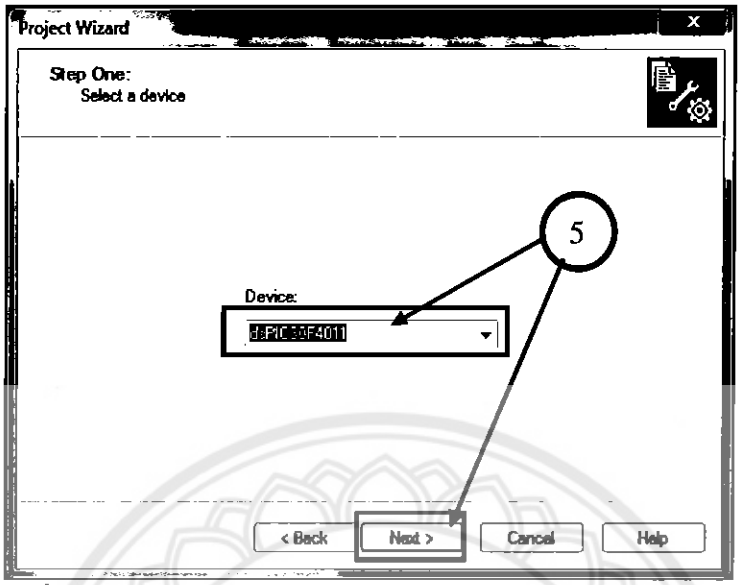
รูปที่ 1.3 แสดงแถบเมนู “Project” เลือก “Project Wizard…”

4. ไดอะล็อกแสดงการใช้งาน “Project Wizard” คลิกปุ่มคำสั่ง “Next” ดังรูปที่ 1.4



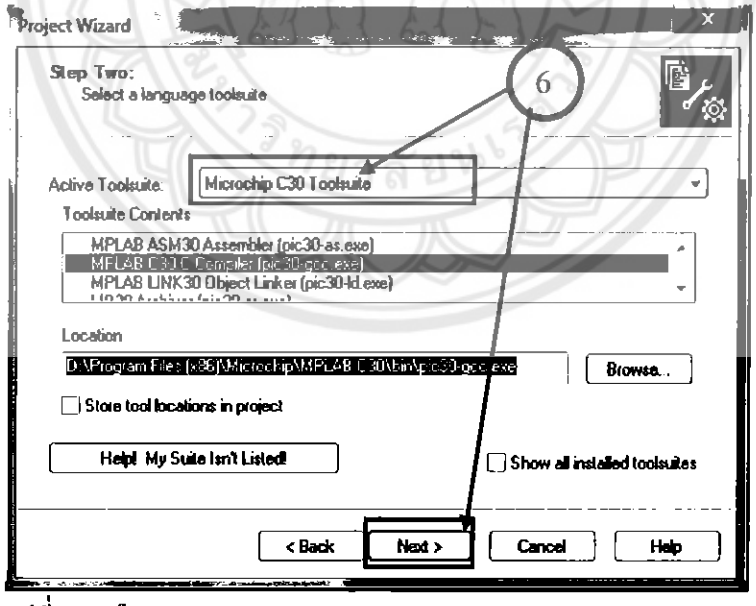
รูปที่ 1.4 ไดอะล็อกแสดงการใช้งาน “Project Wizard”

5. เลือกที่ "Device" ชื่อ "dsPIC30F4011" แล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.5



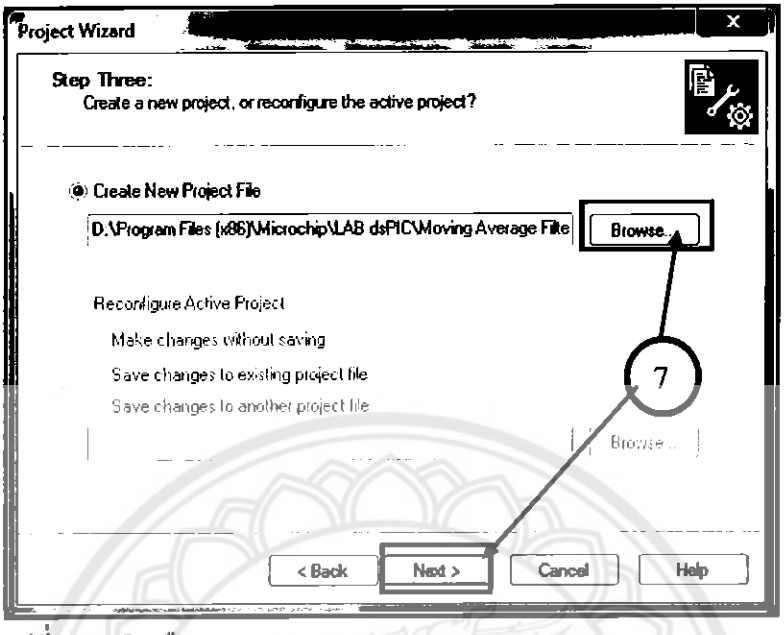
รูปที่ 1.5 เลือก "dsPIC30F4011"

6. เลือก "Microchip C30 Toolsuite" แล้วคลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.6



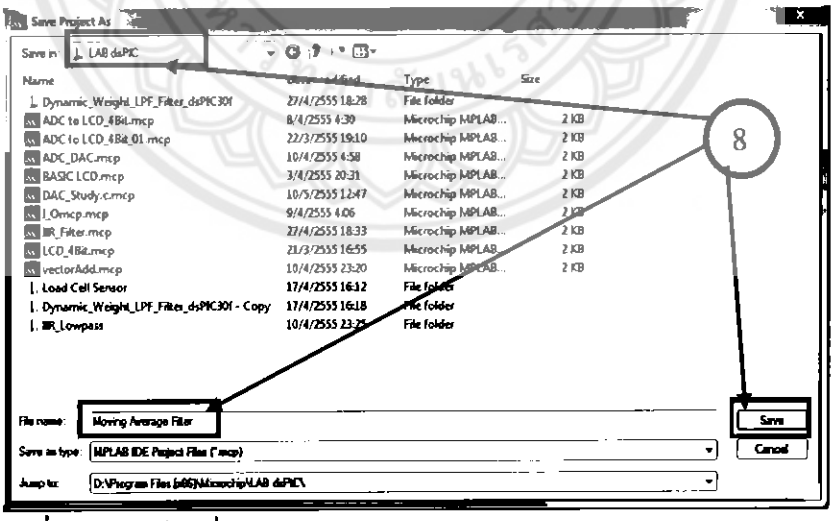
รูปที่ 1.6 เลือก "Microchip C30 Toolsuite"

7. คลิกเลือก “Browse…” เพื่อกำหนดที่อยู่ของแฟ้มที่ต้องการสร้าง ดังรูปที่ 1.7



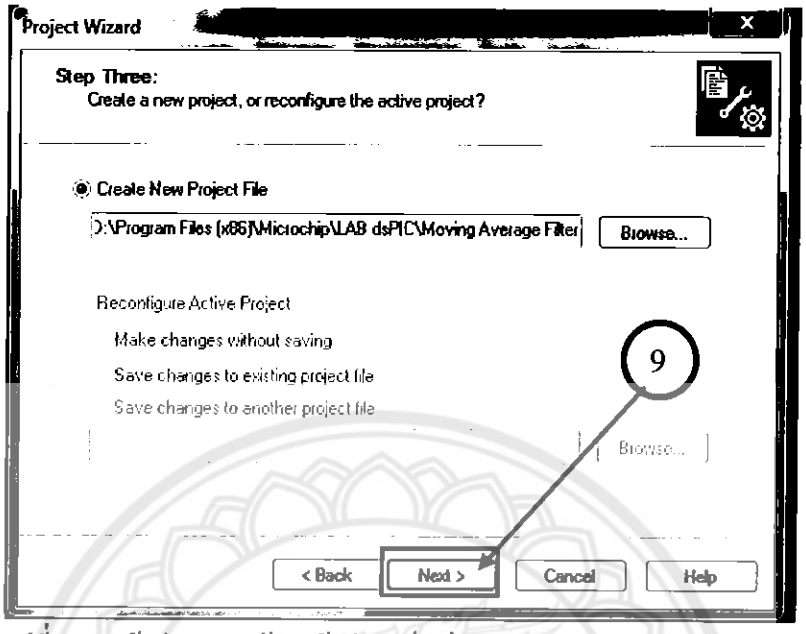
รูปที่ 1.7 คลิกเลือก “Browse…”

8. สร้างแฟ้มโปรเจ็กต์ (folder) ชื่อ “LAB dsPIC” จากนั้นดับเบิลคลิกเข้าไปแล้วทำการสร้างชื่อโปรเจ็กต์ (file name) “Moving Average Filter” แล้วคลิก “Save” ดังรูปที่ 1.8



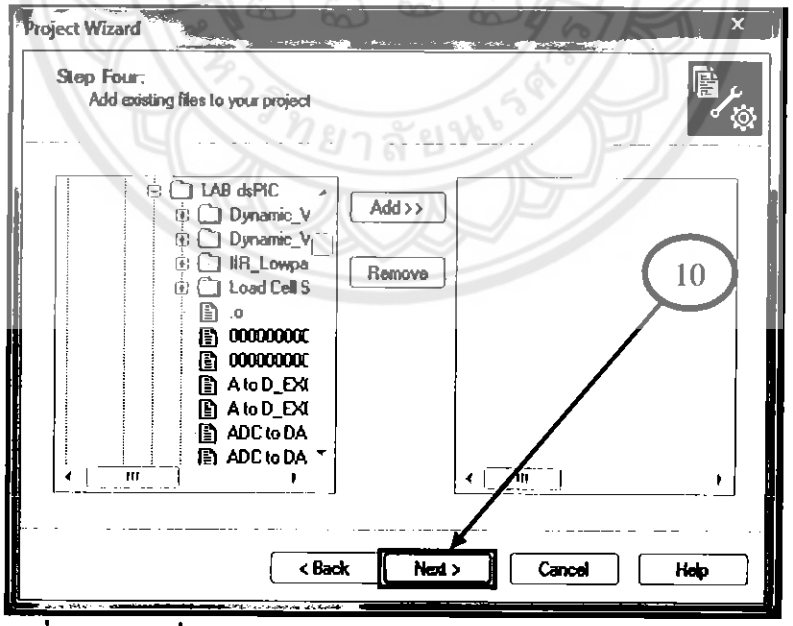
รูปที่ 1.8 โปรเจ็กต์ชื่อ “Moving Average Filter”

9. เมื่อสร้างแฟ้มโปรเจ็กต์เสร็จแล้วทำการคลิก "Next" ดังรูปที่ 1.9



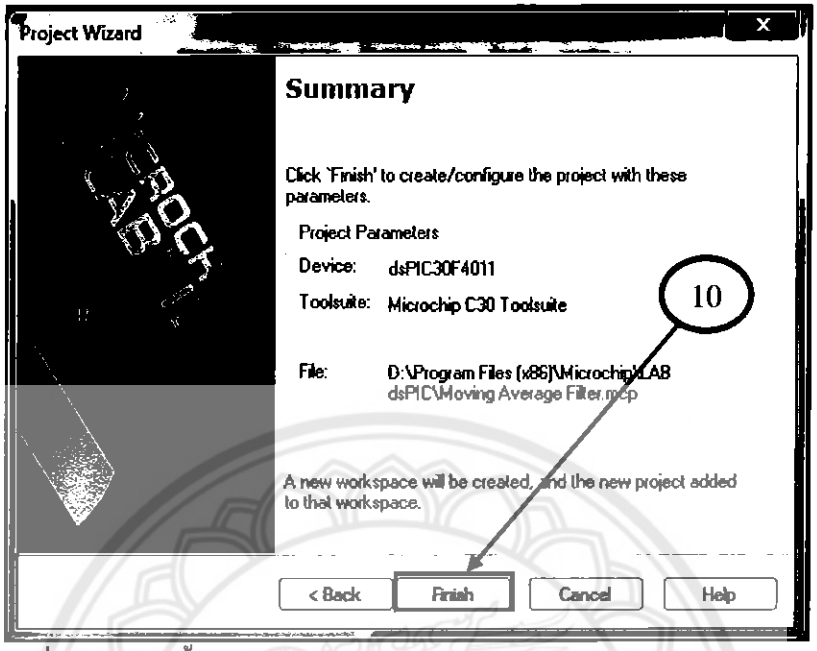
รูปที่ 1.9 หน้าต่างการสร้างแฟ้มโปรเจ็กต์

10. คลิกปุ่มคำสั่ง "Next" ดังรูปที่ 1.10



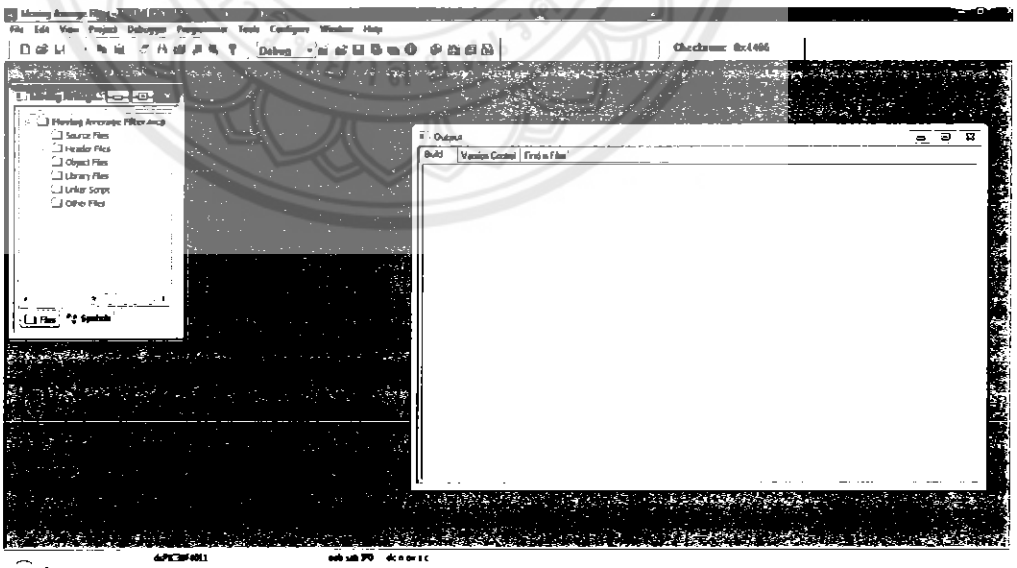
รูปที่ 1.10 การเพิ่มไฟล์เข้าไปในโปรเจ็กต์

11. คลิกปุ่มคำสั่ง "Finish" เพื่อเสร็จสิ้นการสร้างไฟล์โปรเจ็กต์ ดังรูปที่ 1.11



รูปที่ 1.11 เสร็จสิ้นการสร้างไฟล์โปรเจ็กต์

12. หลังจากที่ได้ทำการสร้างไฟล์โปรเจ็กต์เสร็จแล้วก็จะได้หน้าต่างที่พร้อมที่จะทำการสร้างโปรแกรมเพื่อทำการทดลอง ดังรูปที่ 1.12

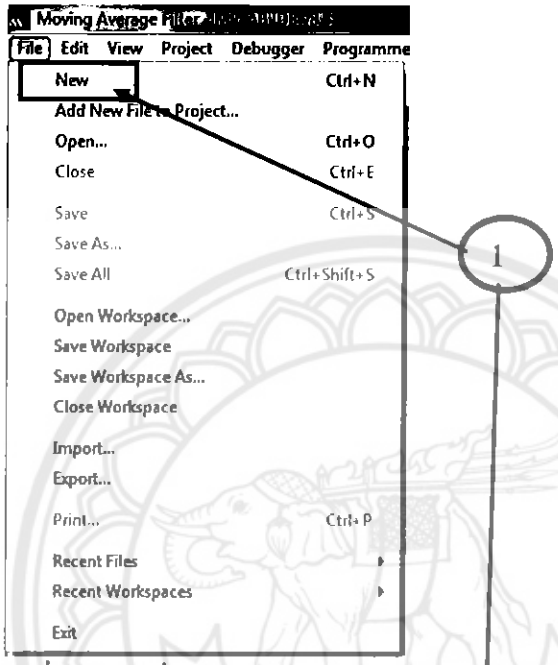


รูปที่ 1.12 หน้าต่างที่พร้อมที่จะทำการสร้างโปรแกรมเพื่อทำการทดลอง

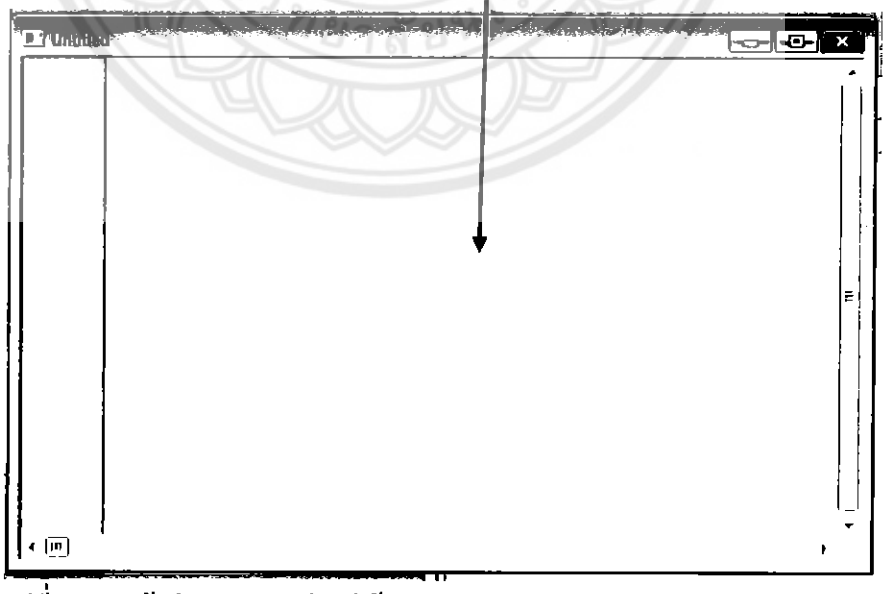
1.2 การสร้างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่

โดยจะยกตัวอย่างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 50 ข้อมูล

1. เริ่มต้นโดยการไปที่ "File" เลือก "New" ดังรูปที่ 1.13 เพื่อจะเปิดหน้าต่างสำหรับเขียนโค้ดโปรแกรม ดังรูปที่ 1.14

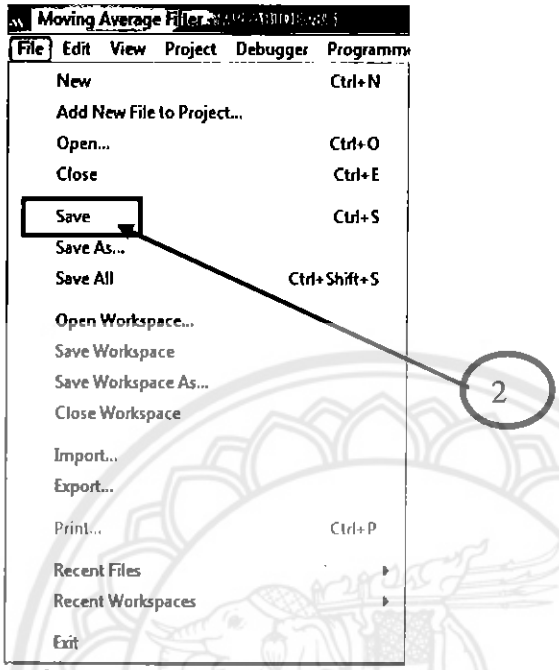


รูปที่ 1.13 ไปที่ "File" เลือก "New"



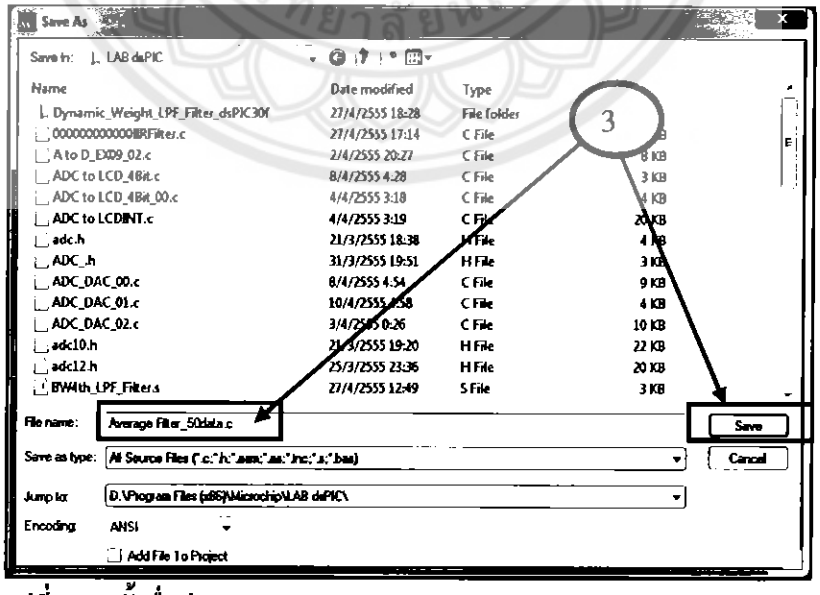
รูปที่ 1.14 หน้าต่างสำหรับเขียนโค้ดโปรแกรม

2. จากนั้นทำการบันทึกหน้าต่างสำหรับเขียนโค้ดโปรแกรมโดยไปที่ "File" เลือก "Save" ดังรูปที่ 1.15



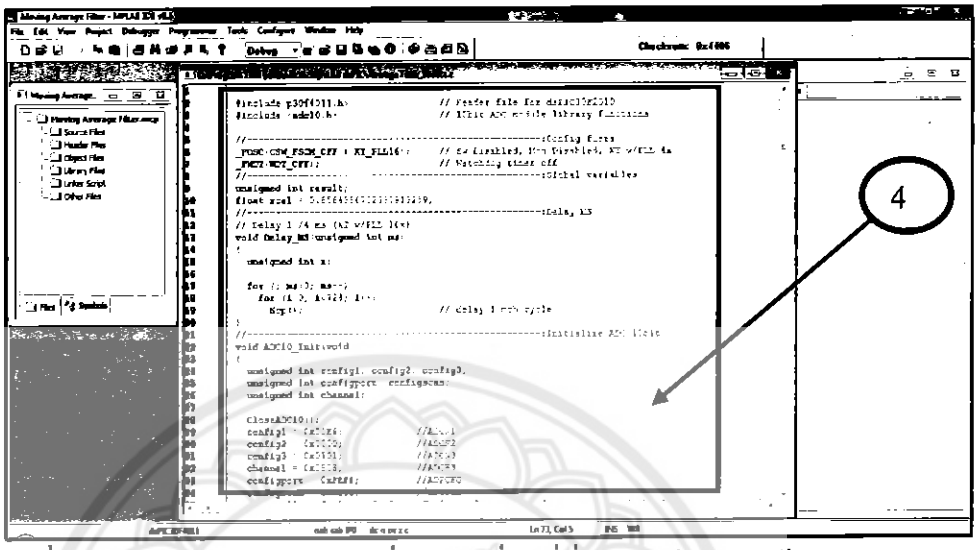
รูปที่ 1.15 ทำการ "Save" หน้าต่างสำหรับเขียนโค้ดโปรแกรม

3. ทำการตั้งชื่อว่า "Average Filter_50data.c" เสร็จแล้วคลิก "Save" ดังรูปที่ 1.16



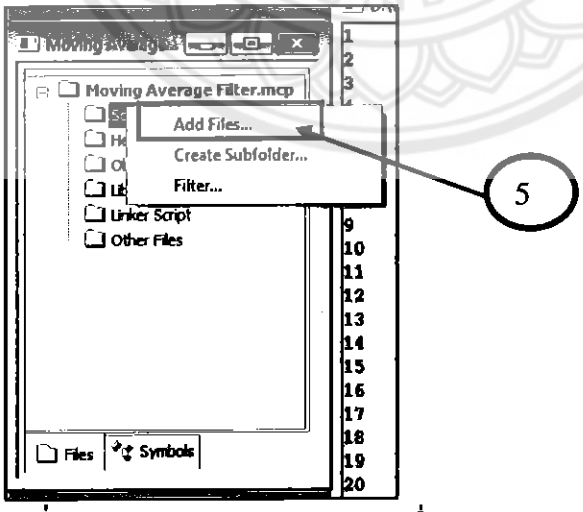
รูปที่ 1.16 ตั้งชื่อว่า "Average Filter_50data.c"

4. เมื่อทำการบันทึกเสร็จแล้ว จากนั้นทำการเขียนการกรองเฉลี่ยแบบเคลื่อนที่ ที่จำนวนข้อมูล 50 ข้อมูล ดังรูปที่ 1.17

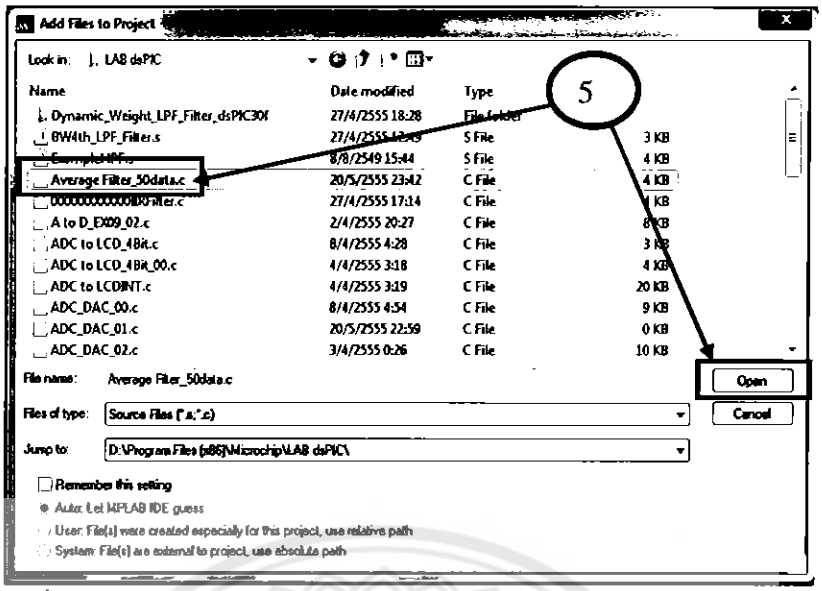


รูปที่ 1.17 ทำการเขียนการกรองเฉลี่ยแบบเคลื่อนที่ที่จำนวนข้อมูล 50 ข้อมูล

5. ต่อไปเป็นการแอดไฟล์เอามาเพื่อทำการคอมไฟล์ ทดสอบความถูกต้องของโปรแกรมโดยไปที่หน้าต่างโปรเจ็กต์ คลิกขวาที่ "Source Files" เพื่อเลือก "Add Files..." เลือก Sound Code โปรแกรมชื่อ "Average Filter_50data.c" แล้วคลิก "Open" เข้ามาในโปรเจ็กต์ไฟล์ ดังรูปที่ 1.18 และ 1.19

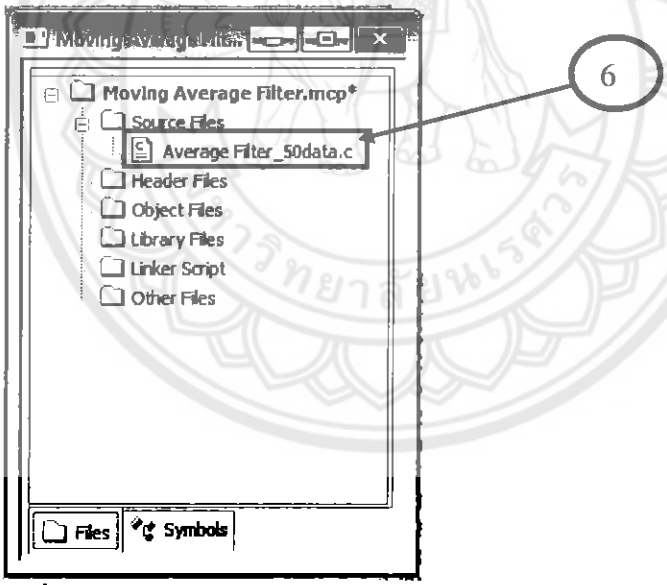


รูปที่ 1.18 คลิกเลือก "Source Files" เพื่อ "Add Files..."



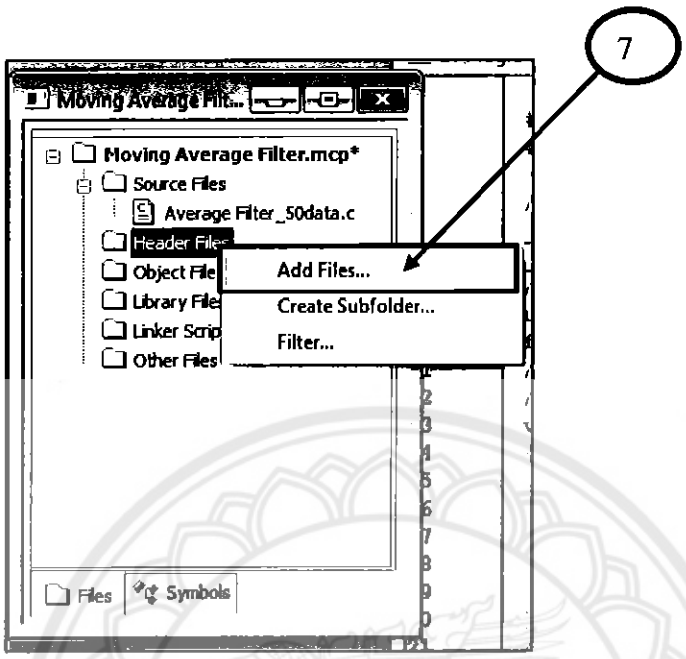
รูปที่ 1.19 โค้ดโปรแกรมชื่อ "Average Filter_50data.c" เข้ามาในโปรเจ็กต์ไฟล์

6. เมื่อเพิ่มไฟล์ "Average Filter_50data.c" เข้ามาในโปรเจ็กต์ไฟล์แล้วจะได้ ดังรูปที่ 1.20



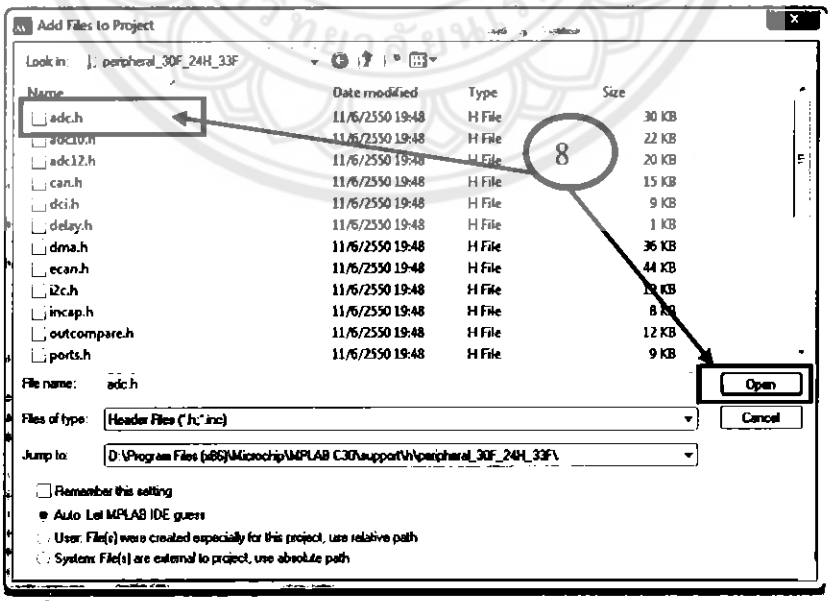
รูปที่ 1.20 ไฟล์ "Average Filter_50data.c" เข้ามาในโปรเจ็กต์ไฟล์

7. ลำดับต่อไปคลิกเลือก “Header Files” เพื่อ “Add Files…”ชื่อ “adc10.h” เข้ามาในโปรเจ็กต์ไฟล์ ดังรูปที่ 1.21



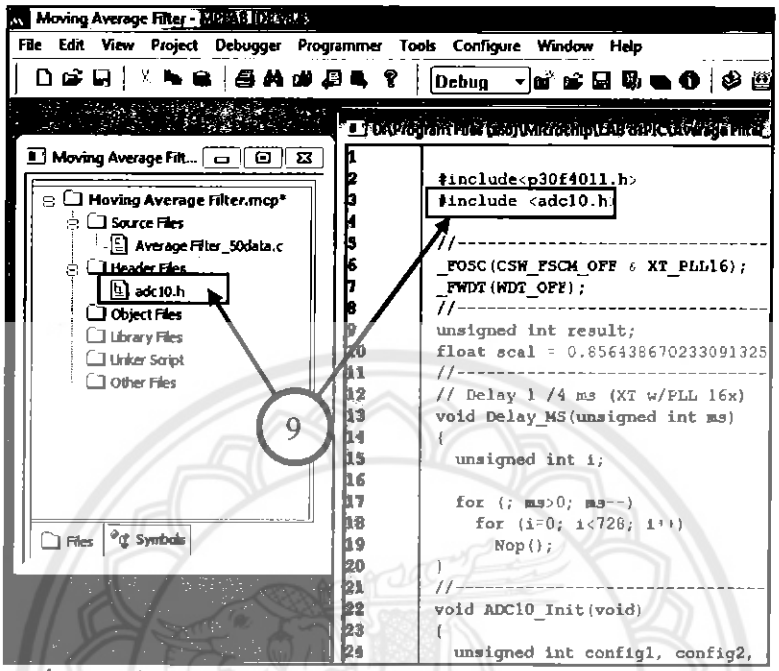
รูปที่ 1.21 “Add Files…”ชื่อ “adc10.h” เข้ามาในโปรเจ็กต์ไฟล์

8. ไปที่ “\Microchip\MPLAB 30\support\h\peripheral_30F_24H_33F” แล้วเลือก“adc10.h” แล้วคลิก “Open” ดังรูปที่ 1.22



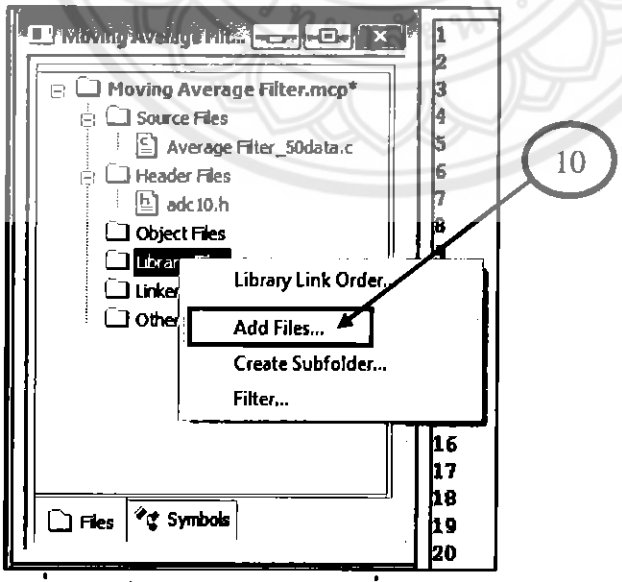
รูปที่ 1.22 เลือก “adc10.h”

9. เมื่อได้เพิ่มไฟล์ "adc10.h" เข้ามาในโปรเจ็กต์ไฟล์แล้ว เหตุผลเพราะเมื่อได้เขียนโปรแกรม include ไฟล์ "adc10.h" แล้วจำเป็นต้องทำการ Add Files ใน Header Files ดังรูปที่ 1.23



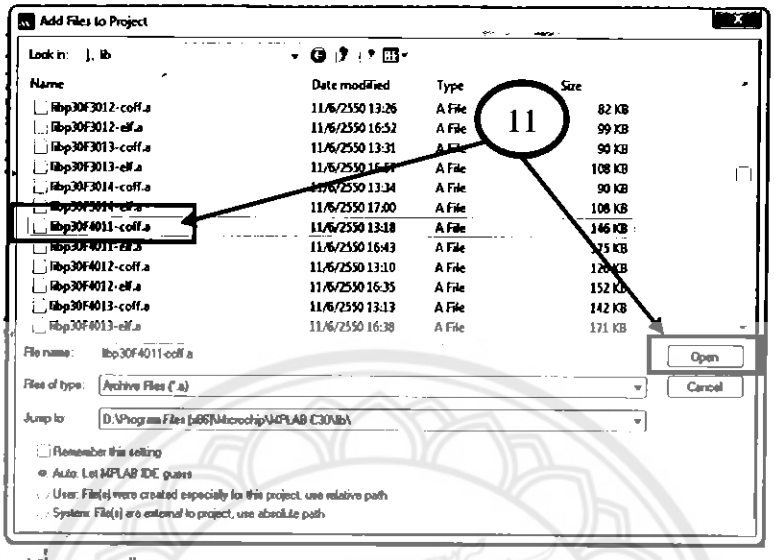
รูปที่ 1.23 เพิ่มไฟล์ "adc10.h" เข้ามาในโปรเจ็กต์ไฟล์

10. ลำดับต่อไปคลิกเลือก "Library Files" เพื่อ "Add Files..." ดังรูปที่ 1.24



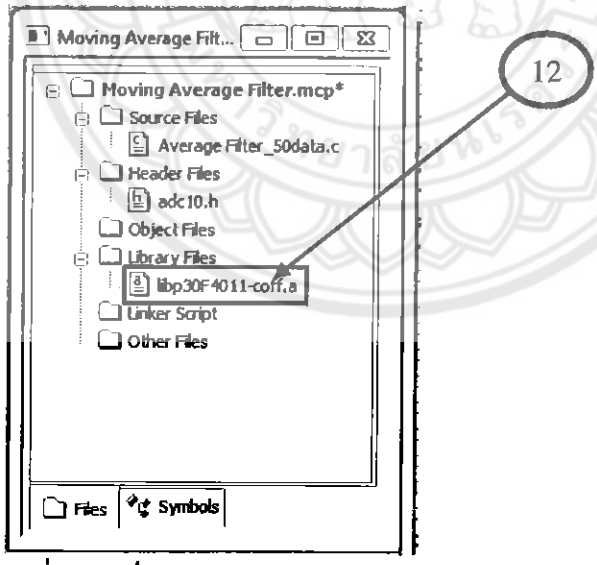
รูปที่ 1.24 เลือก "Library Files" เพื่อ "Add Files..."

11. ไปที่ "...\Microchip\MPLAB C30\lib" แล้วเลือก "libp30F4011-coff.a" แล้วคลิก "Open" ดังรูปที่ 1.25



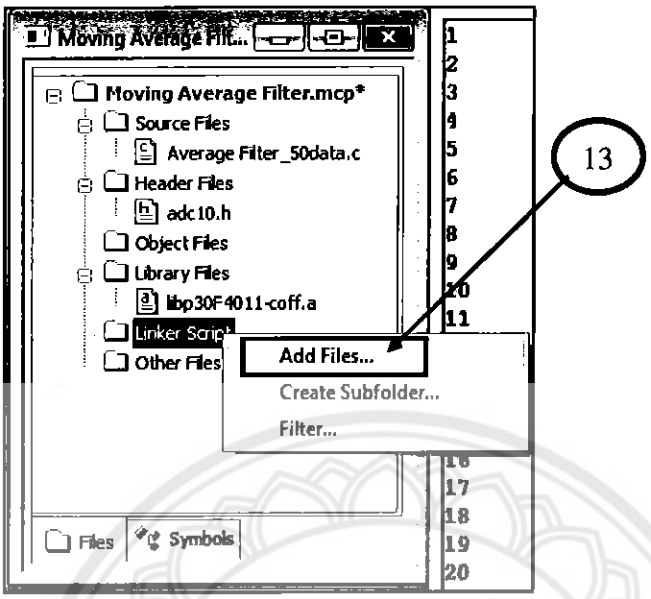
รูปที่ 1.25 เลือก "libp30F4011-coff.a"

12. เมื่อเพิ่มไฟล์ "libp30F4011-coff.a" เข้ามาในโปรเจ็กต์ไฟล์แล้วจะได้ **ดังรูปที่ 1.26**



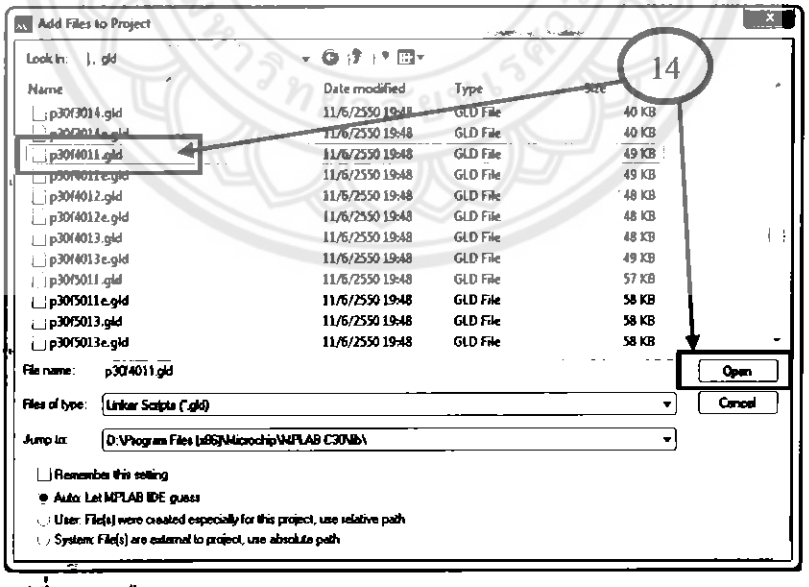
รูปที่ 1.26 เพิ่มไฟล์ "libp30F4011-coff.a" เข้ามาในโปรเจ็กต์ไฟล์

13. ลำดับต่อไปคลิกเลือก "Linker Script" เพื่อ "Add Files..." ดังรูปที่ 1.27



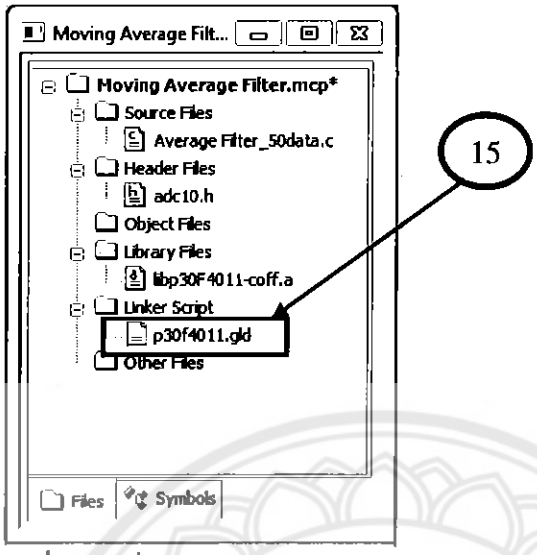
รูปที่ 1.27 คลิกเลือก "Linker Script" เพื่อ "Add Files..."

14. ไปที่ "\Microchip\MPLAB C30\support\gld" แล้วทำการเลือก "p30f4011.gld" แล้วก็คลิก "Open" ดังรูปที่ 1.28



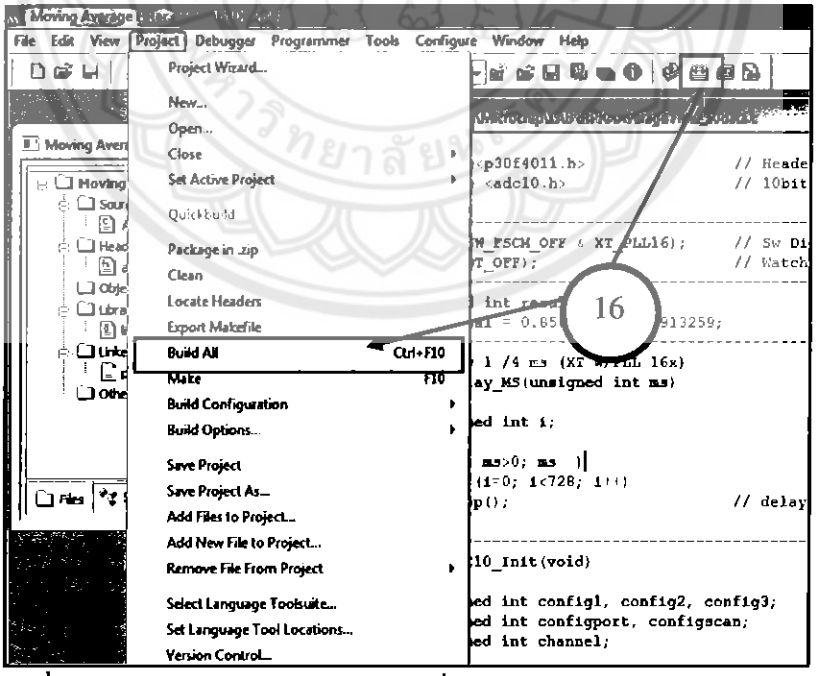
รูปที่ 1.28 เลือก "p30f4011.gld"

15. เมื่อเพิ่มไฟล์ “p30f4011.gld” เข้ามาในโปรเจ็กต์ไฟล์แล้วจะได้ ดังรูปที่ 1.29



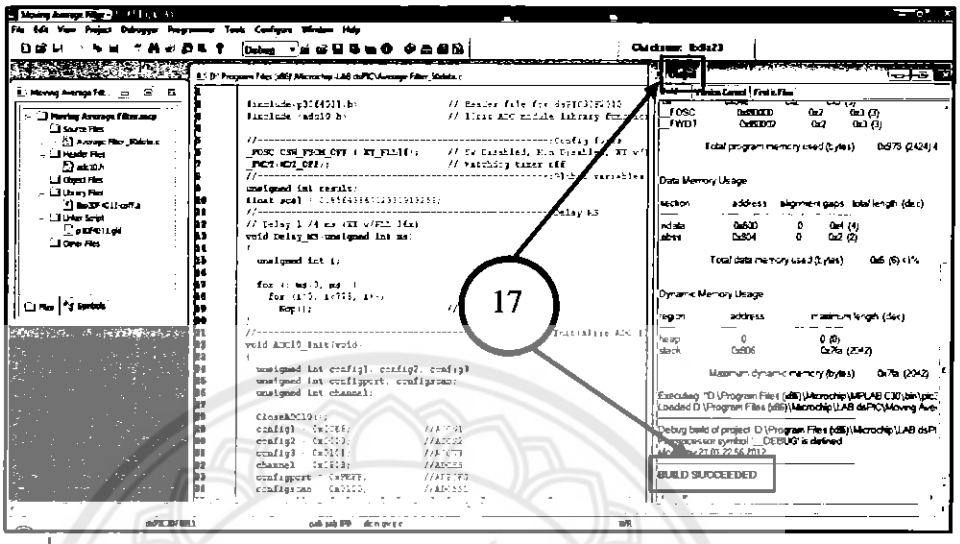
รูปที่ 1.29 เพิ่มไฟล์ “p30f4011.gld” เข้ามาในโปรเจ็กต์ไฟล์

16. ต่อไปจะเป็นการคอมไพล์โค้ดโปรแกรม เพื่อทดสอบความถูกต้อง เริ่มจากไปที่แถบเมนู “Project” จากนั้นเลือก “Build All” หรือไปที่ไอคอน ดังรูปที่ 1.30



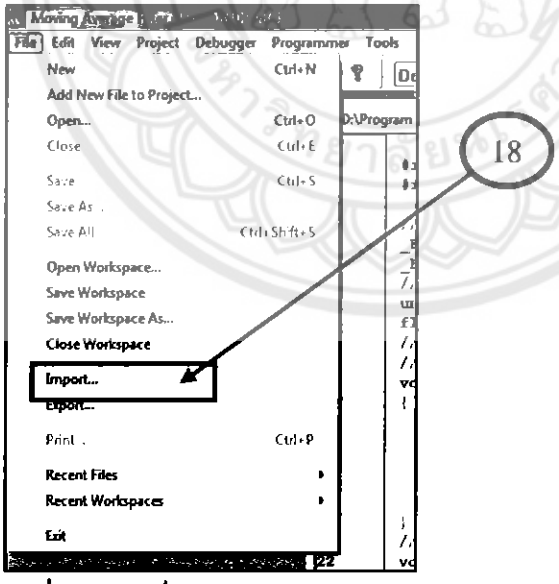
รูปที่ 1.30 การคอมไพล์โค้ดโปรแกรมเพื่อทดสอบความถูกต้อง

17. เมื่อคอมไพล์โค้ดโปรแกรม แล้วถูกต้องจะมีข้อความขึ้นมาคือ “BUILD SUCCEEDED” ที่หน้าต่าง “Output” ดังรูปที่ 1.31



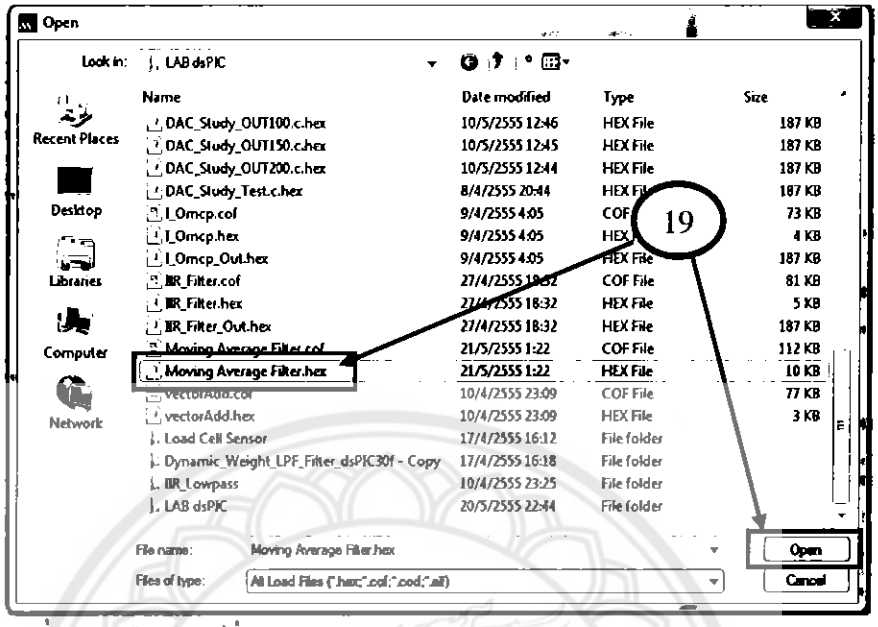
รูปที่ 1.31 ลักษณะการคอมไพล์โปรแกรมแล้วถูกต้อง

18. จากนั้นไปที่ “File” แล้วคลิกเลือก “Import...” ดังรูปที่ 1.32



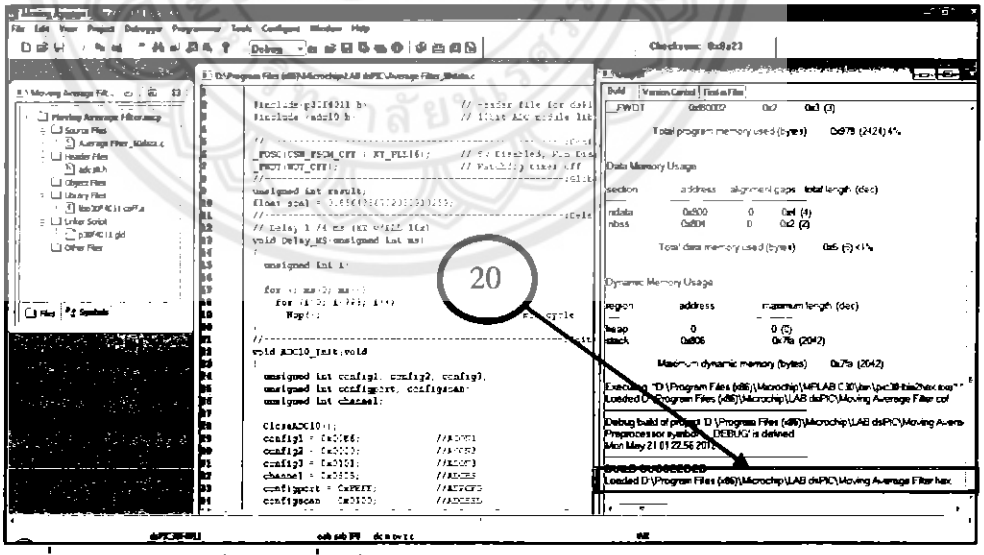
รูปที่ 1.32 ไปที่ “File” แล้วเลือก “Import”

19. จากนั้นเลือกไฟล์ชื่อ “Moving Average Filter.hex” เสร็จแล้วคลิก “Open” ดังรูปที่ 1.33



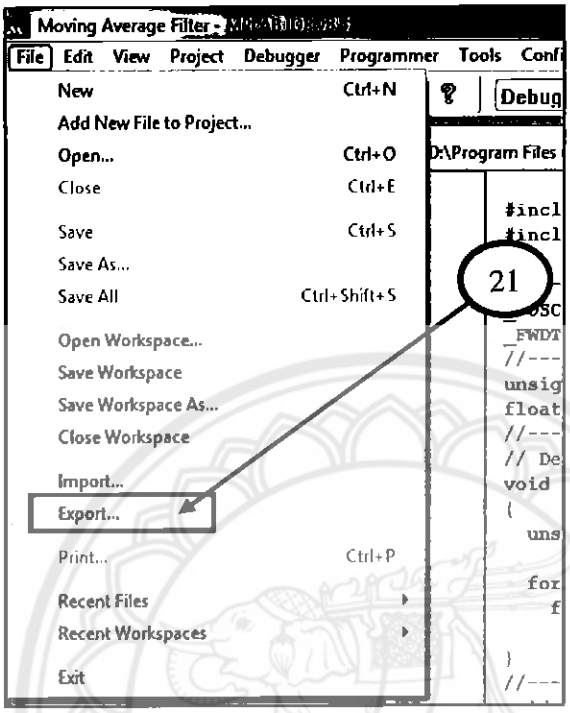
รูปที่ 1.33 เลือกไฟล์ชื่อ “Moving Average Filter.hex”

20. จากข้อที่ 19 เมื่อได้คลิก “Open” เสร็จแล้ว ก็จะมีลักษณะข้อความคือ “…\Microchip\LAB dsPIC\Moving Average Filter.hex.” ที่หน้าต่าง Output ดังรูปที่ 1.34



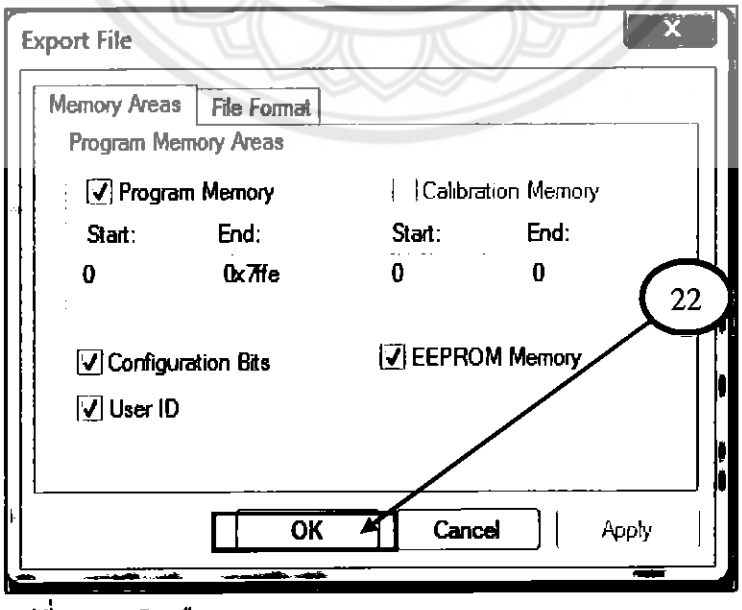
รูปที่ 1.34 ลักษณะข้อความที่หน้าต่าง “Output”

21. จากนั้นไปที่ "File" แล้วคลิกเลือก "Export..." ดังรูปที่ 1.35 เพื่อสร้าง file ต้นแบบสำหรับ Burn โปรแกรม



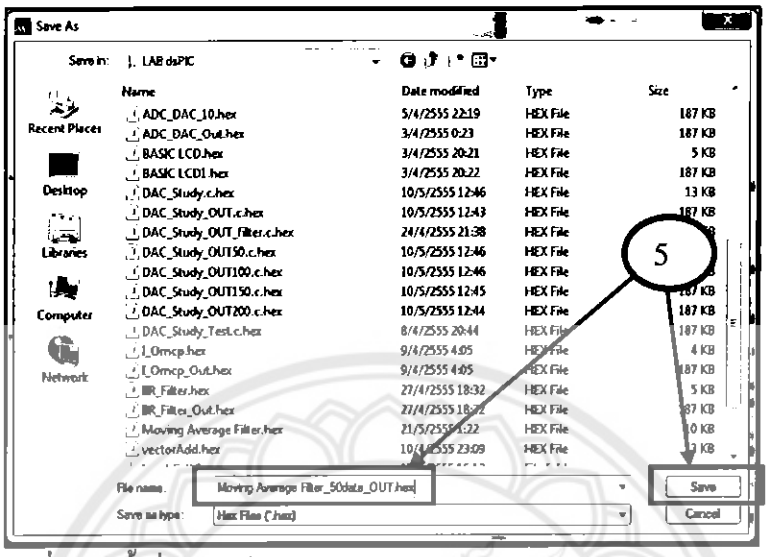
รูปที่ 1.35 คลิกเลือก "Export..."

22. คลิกเลือก "OK" ดังรูปที่ 1.36



รูปที่ 1.36 คลิกเลือก "OK"

23. ทำการตั้งชื่อใหม่เป็น "Moving Average Filter_50data_OUT.hex" และคลิก "Save" ดังรูปที่ 1.37 แล้วก็เป็นอันเสร็จขั้นตอนการสร้างโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่



รูปที่ 1.37 ตั้งชื่อใหม่เป็น "Moving Average Filter_50data_OUT.hex"

ดังนั้นก็จะได้ไฟล์ที่ใช้ในการทดลองของโครงงานในครั้งนี้มีชื่อว่า "Moving Average Filter_50data_OUT.hex" สำหรับการกรองที่ใช้จำนวนข้อมูล 100 150 และ 200 ข้อมูล ที่ไม่ได้ยกตัวอย่าง สามารถทำตามวิธีข้างต้นได้และบันทึกเป็นชื่อ

Moving Average Filter_100data_OUT.hex

Moving Average Filter_150data_OUT.hex

Moving Average Filter_200data_OUT.hex

ตามลำดับ

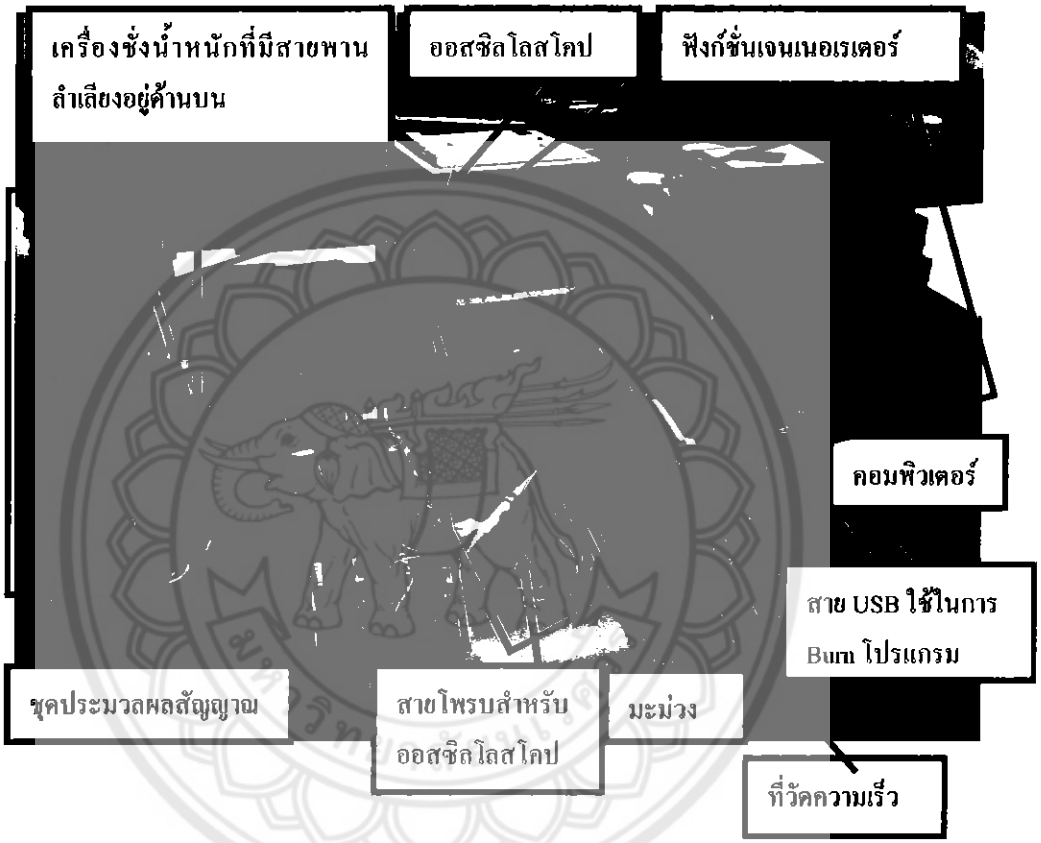
ภาคผนวก ค

วิธีการทดลองโครงงานการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง

วิธีการทดลองโครงงานการประยุกต์ใช้วงจรกรองดิจิตอลสำหรับประมาณค่าน้ำหนักของมะม่วงบนสายพานลำเลียง

ในหัวข้อนี้จะกล่าวถึงขั้นตอนการทำการทดลองโดยจะยกตัวอย่างการทดลองการกรองเฉลี่ยแบบเคลื่อนที่ที่ข้อมูล 50 ข้อมูล โดย

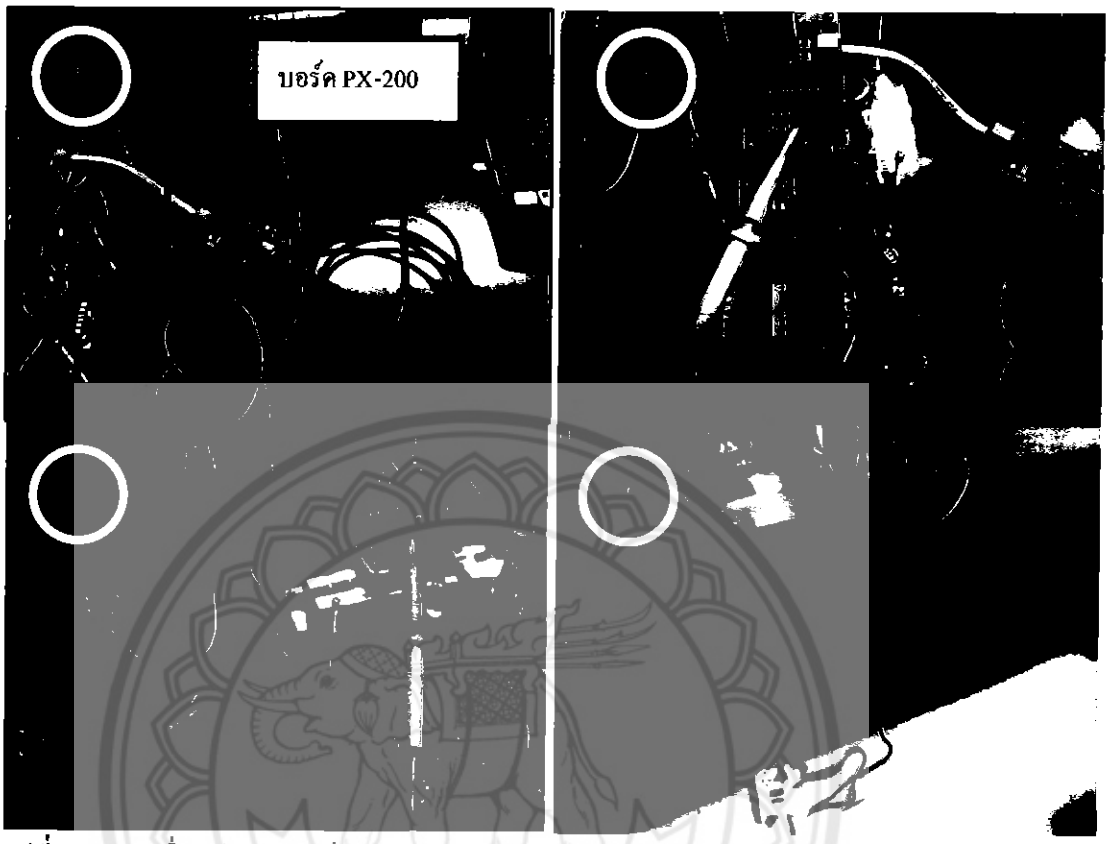
1. อุปกรณ์ที่ใช้ในการทดลอง ดังรูปที่ 1.1



รูปที่ 1.1 แสดงอุปกรณ์การทดลอง

- เครื่องชั่งน้ำหนักที่มีสายพานลำเลียงอยู่ด้านบน
- ที่วัดความเร็ว
- สายเชื่อมต่อในการ Burn โปรแกรม
- ออสซิลโลสโคป
- มะม่วง
- ชุดประมวลผลสัญญาณ
- สายโพรบสำหรับออสซิลโลสโคป
- คอมพิวเตอร์

2. เชื่อมต่ออุปกรณ์ทดลอง ดังรูปที่ 1.2



รูปที่ 1.2 การเชื่อมต่ออุปกรณ์ทดลอง

จากรูปที่ 1.2 แสดงการเชื่อมต่ออุปกรณ์ทดลองมีรายละเอียดดังนี้

" 1 " ทำการเชื่อมต่อสาย USB เพื่อใช้ในการ Burn โปรแกรมลงวงจรไมโครคอนโทรเลอร์ โดยใช้บอร์ด PX-200 เป็นตัวเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรไมโครคอนโทรเลอร์

" 2 " ทำการเชื่อมต่อสายโพรบระหว่างออสซิลโลสโคปและวงจรไมโครคอนโทรเลอร์ที่อยู่ในส่วนของอินพุตและเอาท์พุต จำนวนสองสาย

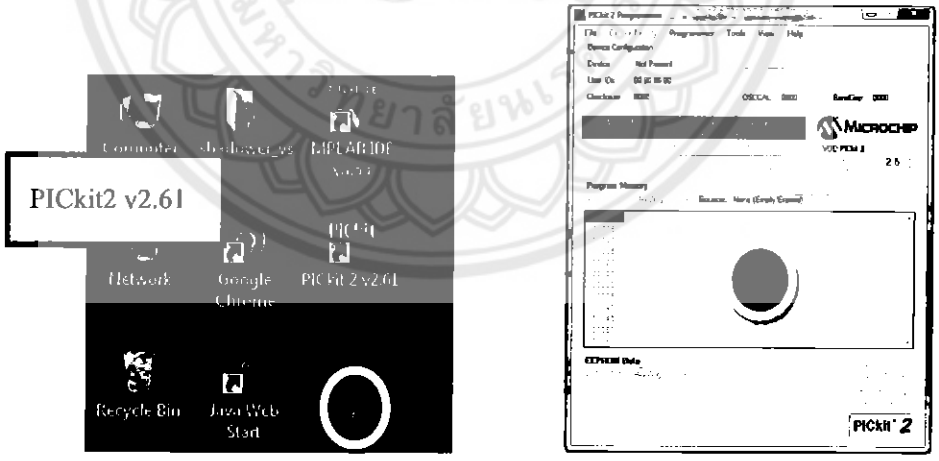
" 3 " ทำการเชื่อมต่อสายของโหลดเซลล์และมอเตอร์เข้ากับชุดประมวลผลสัญญาณ

" 4 " ทำการเชื่อมต่อสายไฟเพื่อป้อนไฟ 220 โวลต์ ให้กับชุดประมวลผมสัญญาณ

2. เมื่อเชื่อมต่ออุปกรณ์เสร็จแล้วจากนั้นทำการเปิดไฟที่ชุดประมวลผลสัญญาณที่สวิตช์สีแดงและเปิดเครื่องออสซิลโลสโคป เพื่อเตรียมพร้อมในการทดลองดังรูปที่ 1.3

รูปที่ 1.3 เปิดไฟชุดประมวลผลสัญญาณและออสซิลโลสโคป

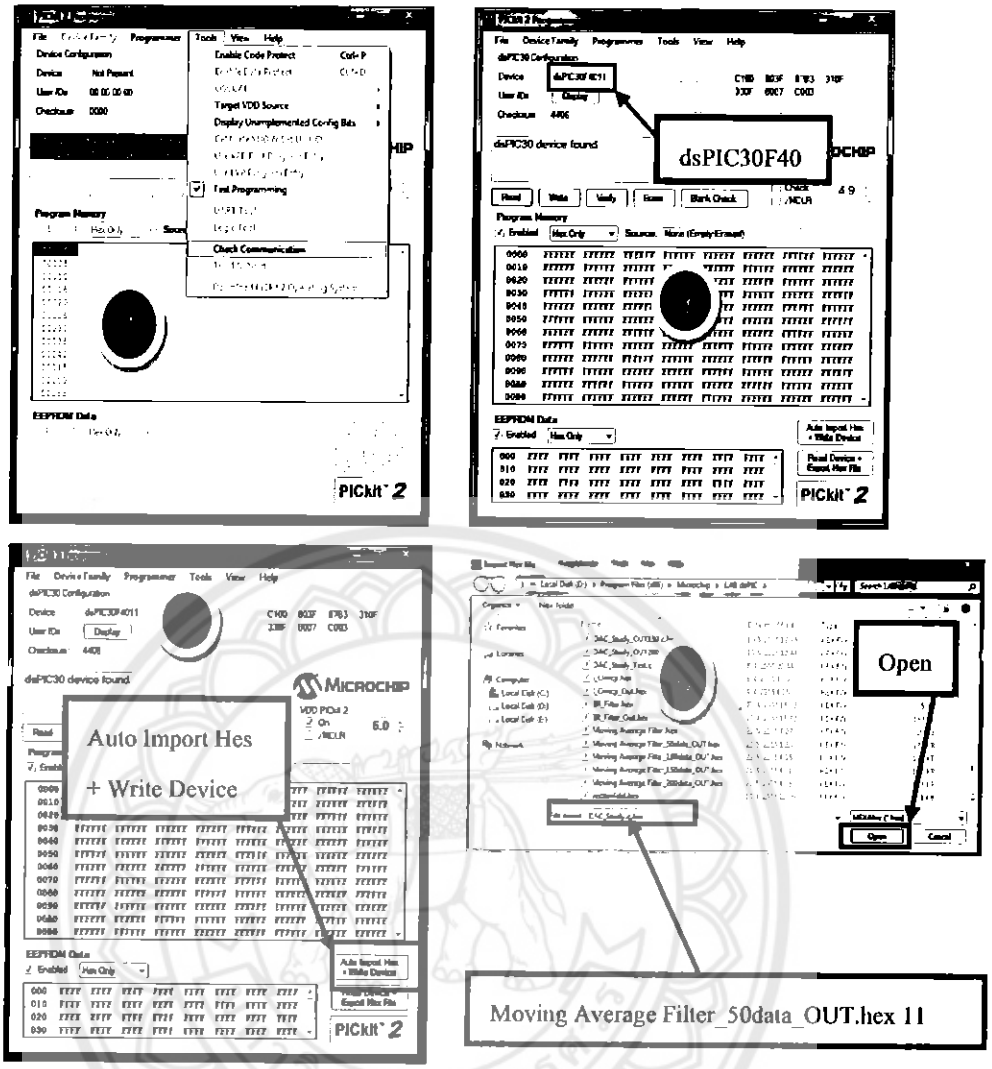
3. เมื่อทำการเปิดไฟที่ชุดประมวลผลสัญญาณและเครื่องออสซิลโลสโคป เพื่อเตรียมพร้อมในการทดลองแล้วต่อไปจะทำการ Burn โค้ดโปรแกรมการกรองเฉลี่ยแบบเคลื่อนที่โดยมีขั้นตอน ดังรูปที่ 1.4 ถึง 1.6



รูปที่ 1.4 ขั้นตอนการเริ่มทำการ Burn โปรแกรม

" 1 " คลิกเลือกที่ไอคอน "PICkit2 v2.61"

" 2 " เมื่อคลิกเลือกที่ไอคอน "PICkit2 v2.61" แล้วจะได้หน้าเริ่มต้นของโปรแกรม PICkit2



รูปที่ 1.5 ขั้นตอนการเริ่มทำการ Burn โปรแกรม (ต่อ)

" 3 " ไปที่ "Tools" แล้วคลิกเลือก "Check Communication" เพื่อทำการค้นหาไมโครคอนโทรเลอร์ dsPIC30F4011 ผ่านบอร์ด PX-200

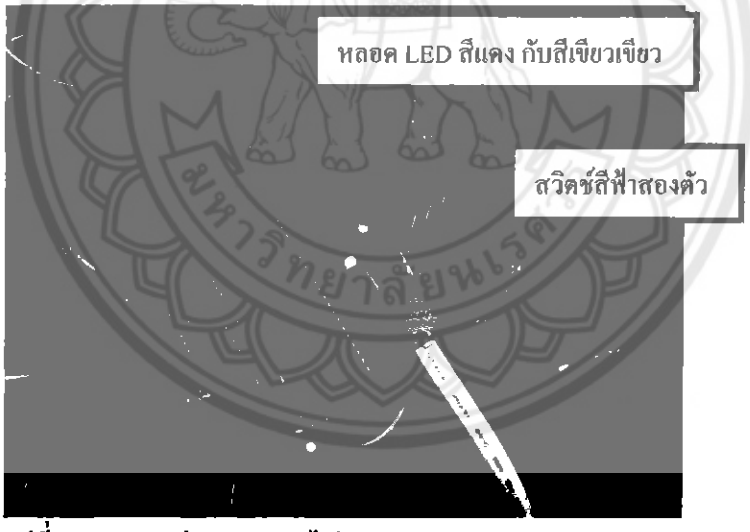
" 4 " เมื่อสามารถติดต่อได้แล้ว ก็จะมีเบอร์ของไมโครคอนโทรเลอร์ขึ้นที่หน้าต่างเริ่มต้นโปรแกรม ในที่นี้ใช้ dsPIC30F4011

" 5 " จากนั้นคลิกที่ "Auto Import Hes + Write Device"

" 6 " แล้วเลือกไฟล์ชื่อ "Moving Average Filter_50data_OUT.hex" เสร็จแล้วคลิกที่ปุ่ม "Open"

รูปที่ 1.6 ขั้นตอนการเริ่มทำการ Burn โปรแกรม (ต่อ)

เมื่อทำการ Burn โปรแกรมเสร็จ ก็จะแสดงแท็บสีเขียวขึ้นมาดังรูปที่ 1.6 แต่การทำการโปรแกรมทุกครั้งต้องกดสวิตช์สีฟ้าสองตัวพร้อมกันเพื่อให้หลอดไฟ LED ติด จากสีเขียวเป็นสีแดง ก่อนทุกครั้งที่มีการทำการ Burn โปรแกรมเข้าไป ดังรูปที่ 1.7



รูปที่ 1.7 แสดงสีของหลอดไฟ LED

สีแดง คือ Burn โปรแกรมลงไมโครคอนโทรเลอร์ได้

สีเขียว คือ ไมโครคอนโทรเลอร์เริ่มทำงาน พร้อมที่จะทำการทดลอง

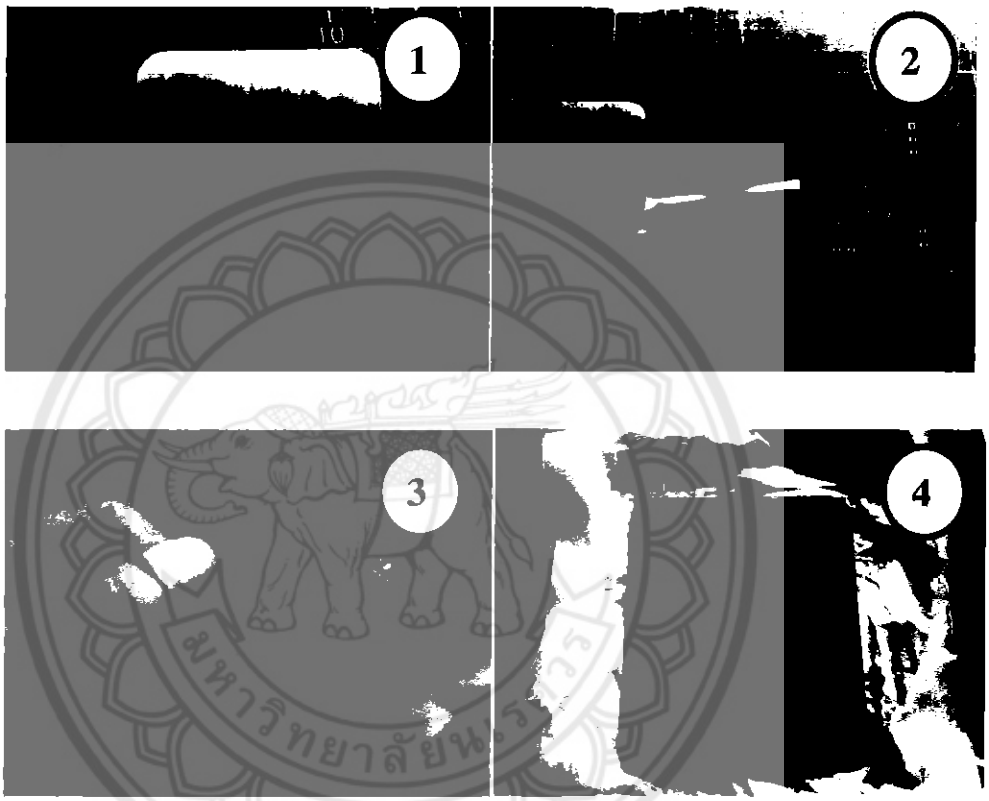
จากนั้นก็ทำการทดลองและบันทึกผลตามที่ได้ออกแบบไว้และเปลี่ยนโปรแกรมจาก 50 ข้อมูลเป็น 100 ข้อมูล 150 ข้อมูล และ200 ข้อมูล ตามลำดับ

ภาคผนวก ง

วิธีการทำ PCB โดยใช้ Dry film

วิธีการทำ PCB โดยใช้ Dry film

ในหัวข้อนี้จะกล่าวถึงวิธีการทำ (Printed Circuit Board: PCB) โดยวิธีการใช้ Dry film เป็นวิธีที่รวดเร็วและสามารถทำ PCB ที่มีเส้นขนาดเล็กๆได้ และ Dry film ที่ใช้นี้ไว้ต่อแสง UV ดังนั้นขณะทำจึงต้องปิดไฟในห้องให้หมดเหลือไว้แค่ไฟแสงสีเหลืองที่ติดกับเครื่องฉายแสงก็พอ

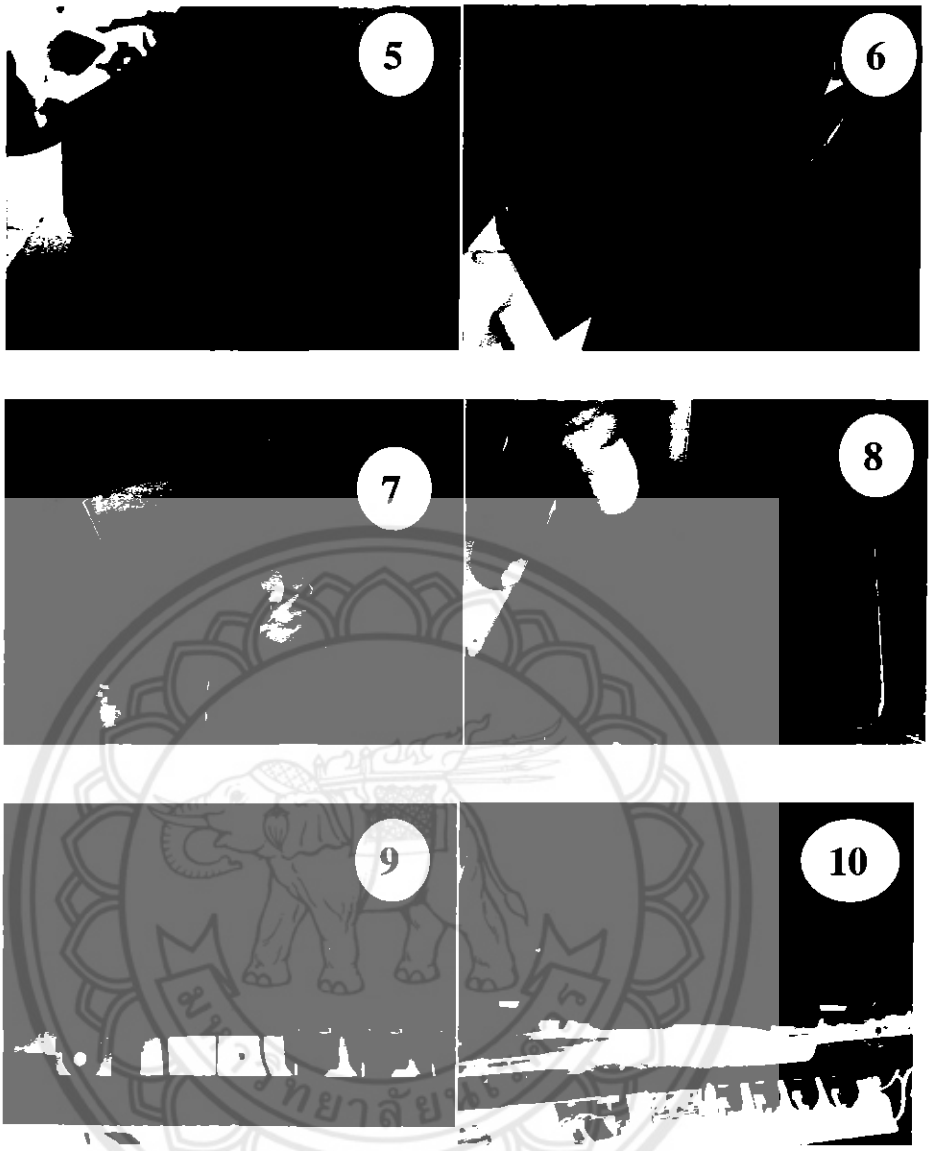


" 1 " ตัดแผ่นปริ้น (แผ่นทองแดง) ตามขนาดของวงจรที่ได้ออกแบบไว้แล้วนำมาขัดให้สะอาดโดยใช้น้ำยาล้างจานหรือน้ำยาทำความสะอาดอื่นๆ (ขณะขัดควรขัดไปทางเดียวกัน)

" 2 " ทำการตัดแผ่น Dry film ให้มีขนาดใหญ่กว่าแผ่นวงจรสีดำที่เตรียมมาเล็กน้อยเพื่อความคลาดเคลื่อน

" 3 " นำสก็อตเทปสองอันมาติดประกบกันโดยมีแผ่น Dry film อยู่ตรงกลางจากนั้นทำการดึงแผ่นสก็อตเทปทั้งสองด้านออกเพื่อจะเอาพลาสติกใสที่เคลือบบนแผ่น Dry film ออก

" 4 " เมื่อได้ทำการดึงพลาสติกใส ที่เคลือบบนแผ่น Dry film ออกแล้ว จากนั้นนำด้านที่ถูกดึงออกมาประกบบนแผ่นปริ้น (ต้องทำให้แผ่นปริ้นเปียกน้ำก่อนทุกครั้ง) จากนั้นก็ใช้มือหรือกระดาษทิชชูเพื่อไล่น้ำออก (ต้องไม่ให้มีฟองน้ำอย่างเด็ดขาด)



“ 5 ” ผับกระดาษสีขาวเป็นช่องเพื่อใส่แผ่นปริ้นเข้าไป

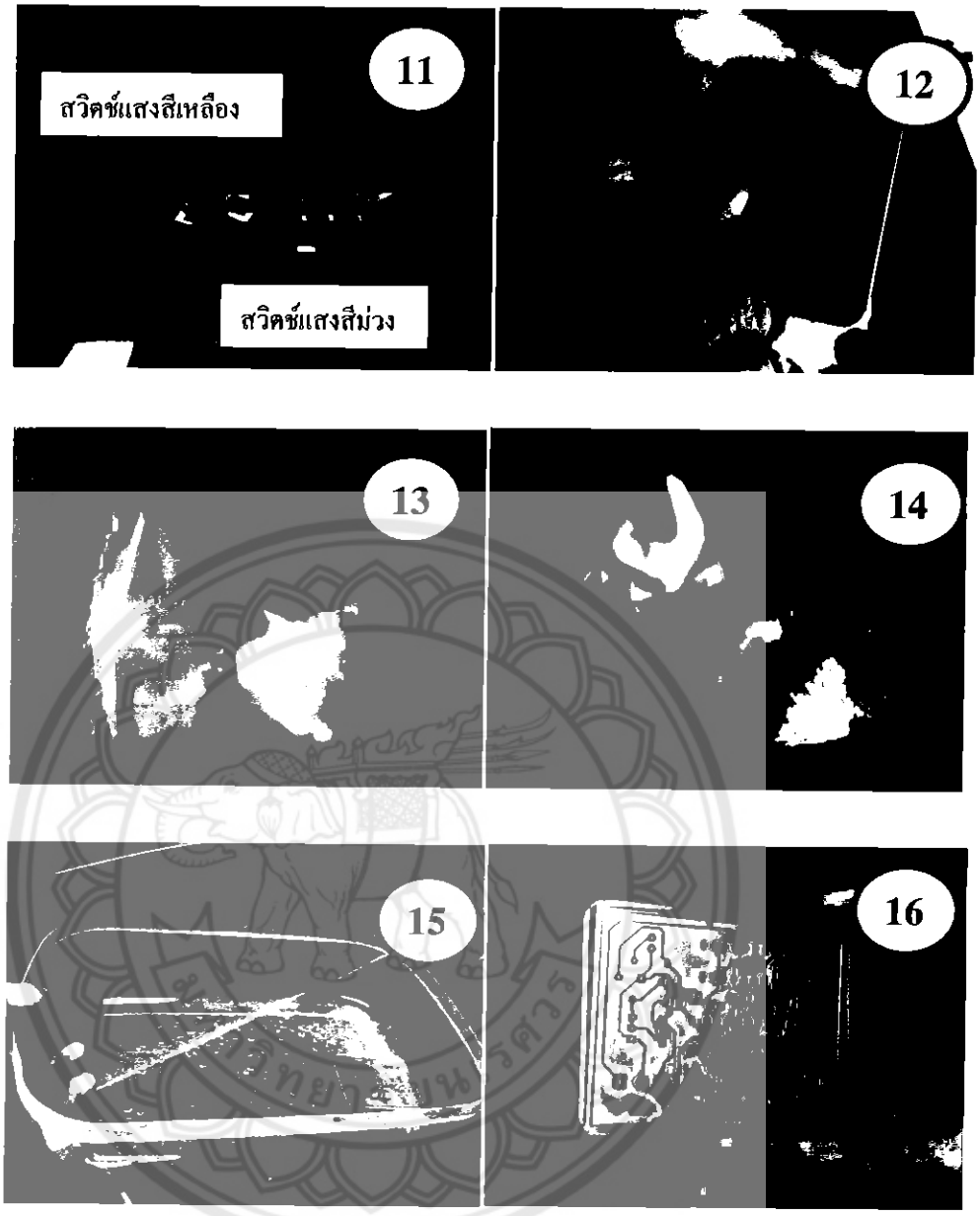
“ 6 ” ทำการรีด 2-3 ที ให้ร้อนพอประมาณ

“ 7 ” จากนั้นทำการนำแผ่นปริ้นออกมาเช็ดด้วยกระดาษทิชชูเพื่อเช็ดไอน้ำออกจากแผ่นปริ้น (ทำวนขั้นตอนที่ 5 ถึง 7 ประมาณ 5 ครั้ง)

“ 8 ” นำแผ่นลายวงจรที่ได้เตรียมไว้ มาประกอบบนแผ่นปริ้นโดยหันด้านที่มีสี นำเข้มติดกับแผ่นปริ้น ดังรูปที่ 8

“ 9 ” นำแผ่นปริ้นมาวางบนเครื่องฉายแสง โดยนำทางด้านที่มีแผ่นลายวงจรติดกับกระจกของเครื่องฉายแสง ดังรูปที่ 9

“ 10 ” จากนั้นนำแผ่นเหล็กหรือแผ่นอลูมิเนียมมากดทับแผ่นปริ้นที่วางอยู่เพื่อไม่ให้แผ่นปริ้นเคลื่อนและไม่ให้แผ่นปริ้นโค้งงอ ดังรูปที่ 10



" 11 " ทำการปิดไฟแสงสีเหลือง จากนั้นทำการเปิดไฟแสงสีม่วง จากสวิตช์สีดำด้านข้างของเครื่องฉายแสง จากนั้นรอจนไฟดับเอง

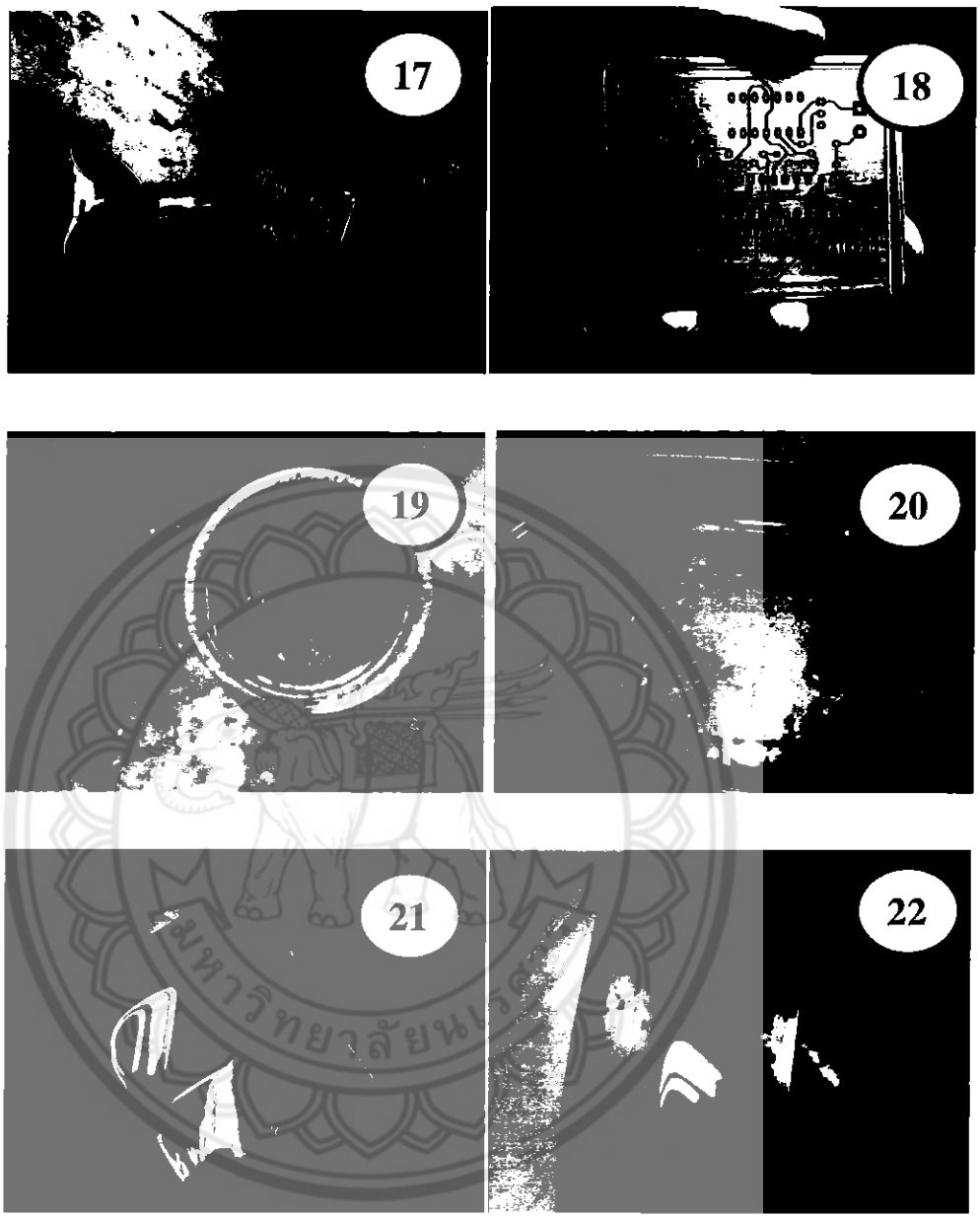
" 12 " เมื่อฉายแสงเสดแล้ว จากนั้นทำการดึงพลาสติกอีกด้านออก โดยใช้คัดเตอร์หรืออื่นๆ ที่ดีกว่า ดังรูปที่ 12

" 13 " เตรียมสารโซเดียมคาร์บอเนต (เมื่อซื้อแผ่น Dry film จะมีโซเดียมคาร์บอเนตที่ต้องใช้งานแถมมาให้)

" 14 " นำโซเดียมคาร์บอเนตละลายกับน้ำ

" 15 " นำแผ่นปริ้นมาใส่ในน้ำที่มีโซเดียมคาร์บอเนต เพื่อทำการขึ้นลายวงจร

" 16 " เมื่อได้ตามรูปแล้วก็สามารถนำไปล้างน้ำได้เลยโดยไม่ต้องรอ Dry film ละลายหมด



“ 17 ” นำแผ่นปริ้นไปล้างน้ำเพื่อล้างแผ่น Dry film ออก

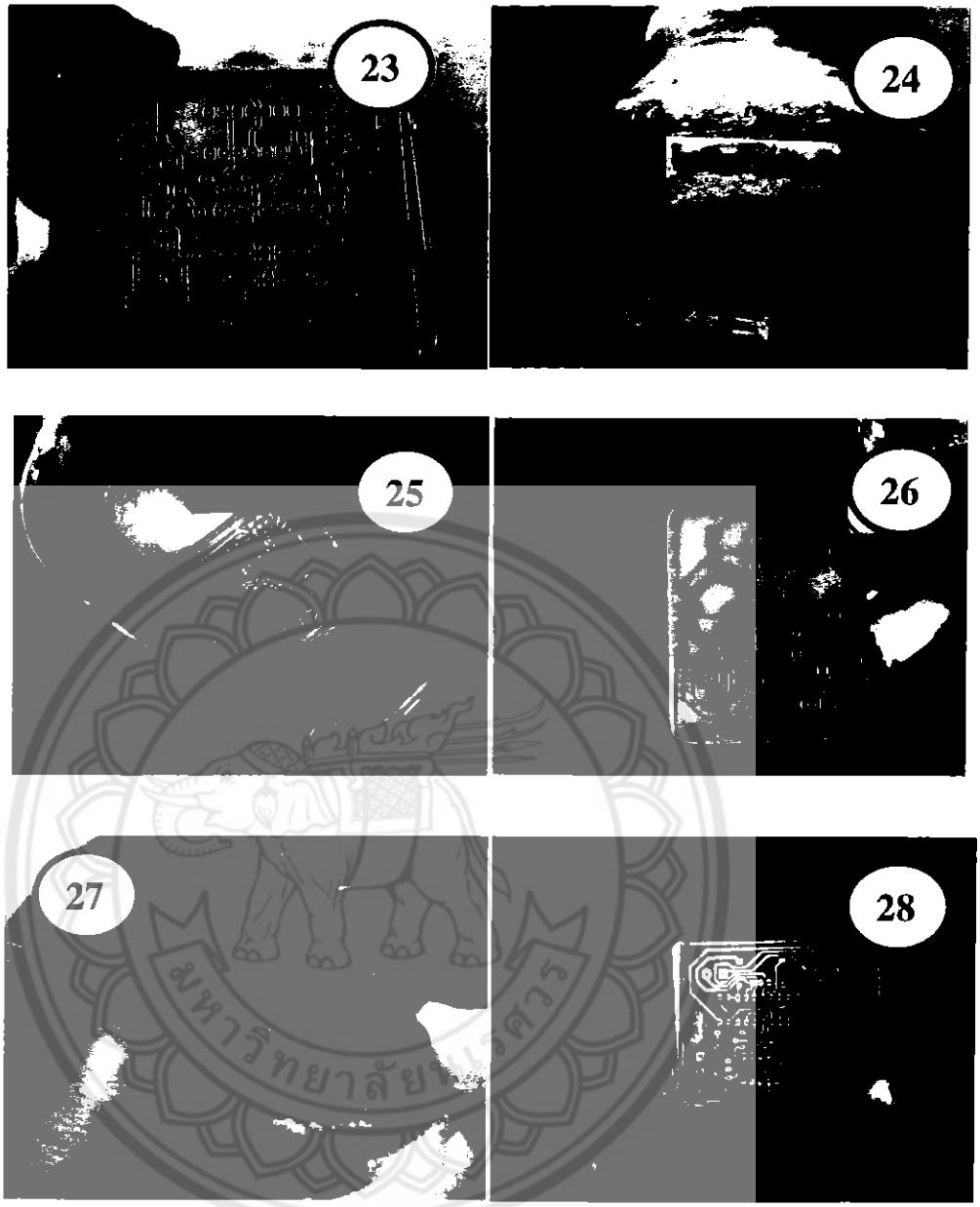
“ 18 ” เมื่อล้างเสร็จแล้วก็จะได้ลายวงจรตามรูปที่ 18

“ 19 ” เตรียมกรดกัดปริ้น

“ 20 ” ใส่กรดกัดปริ้นพอประมาณและใส่น้ำให้ท่วมแผ่นปริ้นเล็กน้อย

“ 21 ” ละลายกรดกัดปริ้นในน้ำให้ละลายจนหมด

“ 22 ” ทำการละลายเนื้อทองบนแผ่นปริ้นโดยใช้กรดกัดปริ้นที่ละลายเตรียมไว้



“ 23 ” เมื่อละลายทองแดงบนแผ่นปริ้นแล้วจะได้ลายวงจรตามรูปที่ 23

“ 24 ” เตรียมสารโซเดียมไฮดรอกไซด์

“ 25 ” นำโซเดียมไฮดรอกไซด์ มาละบายบนแผ่นปริ้นตามรูปที่ 25 ทิ้งไว้ประมาณ 10-20 นาที

“ 26 ” จากนั้นก็จะได้แผ่นปริ้นที่มีแผ่น Dry film ติดอยู่บางๆที่ไม่สามารถละลายออกเองได้

“ 27 ” ทำการนำแผ่นปริ้นมาขัดหรือขูด เพื่อเอาแผ่น Dry film ออกให้หมดซึ่งโดยการใช้ช้อนหรืออื่นๆ ที่ดีกว่าตามรูปที่ 27

“ 28 ” เสร็จแล้วก็จะได้แผ่นปริ้น PCB ตามความต้องการ ดังรูปที่ 28

จากรูปที่ 28 สังเกตว่าจะมีบ้างจุดที่ยังเหลือเนื้อทองแดงอยู่เนื่องจากการทำในขั้นตอนที่ 155 ถึง 17 ไม่ดีพอ