# บทที่ 3

# การออกแบบแหล่งกำเนิดแรงดันและกระแสไฟสลับ

## 3.1 หลักการทำงานของแหล่งกำเนิดแรงดันและกระแสไฟสลับ

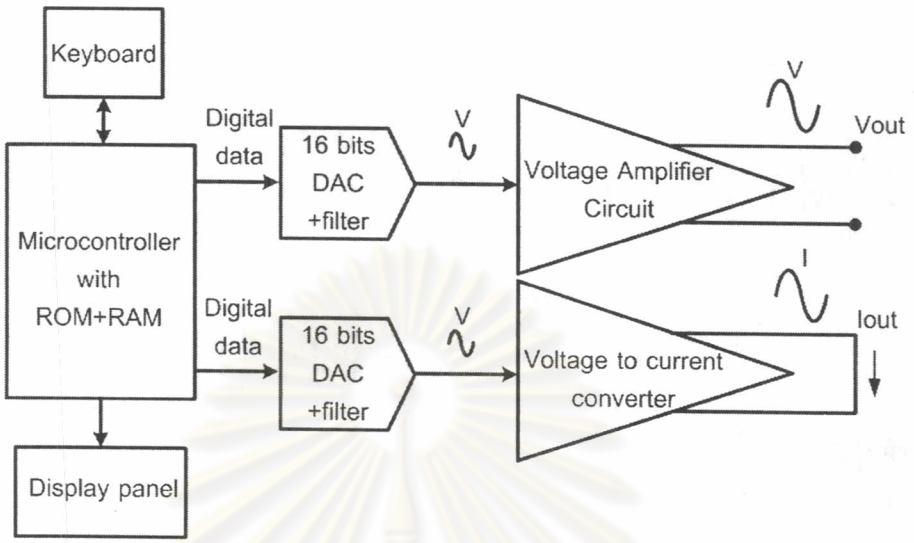
แหล่งกำเนิดแรงดันและกระแสไฟสลับมีหน้าที่สร้างสัญญาณแรงดันและกระแสรูปไซน์สำหรับจ่ายให้ระบบทดสอบความแม่นยำของมิเตอร์วัดพลังงานไฟฟ้า แหล่งกำเนิดสัญญาณดังกล่าวต้องสามารถปรับขนาดแรงดันและกระแสที่จะจ่ายได้ สามารถจ่ายแรงดันและกระแสได้สูงพอ ปรับความต่างเฟสระหว่างสัญญาณแรงดันและกระแสที่จ่ายได้ ปรับความถี่ของสัญญาณทั้งคู่ได้ รวมไปถึงค่าความเพี้ยนเชิงฮาร์มอนิกของสัญญาณที่จ่ายต้องอยู่ในช่วงค่าที่ยอมรับได้ด้วย

แหล่งกำเนิดแรงดันและกระแสไฟสลับที่นำเสนอในวิทยานิพนธ์นี้ตั้งเป้าไว้ให้สามารถจ่ายแรงดันไฟสลับได้ในช่วง 180 – 250 $V_{rms}$ สามารถจ่ายกระแสได้ 0 - 50 $A_{rms}$ โดยมีความผิดพลาดในการตั้งค่าแรงดันและกระแส ± 0.1 % สามารถปรับความถี่ได้ในช่วง 45 - 55 Hz ผิดพลาดไม่เกิน 0.1 % และยังสามารถปรับความต่างเฟสระหว่างแรงดันและกระแสได้ตั้งแต่ 0 - 360 องศา ผิดพลาดไม่เกิน 0.5 องศา โดยสัญญาณที่จ่ายต้องมีความเพี้ยนเชิงฮาร์มอนิกรวม (Total Harmonic Distortion ;THD) ไม่เกิน 2 %

## 3.2 โครงสร้างภายในของแหล่งกำเนิดแรงดันและกระแสไฟสลับ

แหล่งกำเนิดสัญญาณแรงดันและกระแสจะสร้างสัญญาณออกแยกเป็น 2 ช่อง คือ ช่องสัญญาณแรงดันและช่องสัญญาณกระแส ดังแสดงในรูปที่ 3.1 โดยจากรูป โครงสร้างภายในแบ่งออกเป็น 2 ส่วนหลัก ๆ ด้วยกันคือ ส่วนสร้างสัญญาณต้นแบบ และส่วนขยายสัญญาณ ส่วนสร้างสัญญาณต้นแบบจะทำหน้าที่สองอย่าง อย่างแรกคือสร้างสัญญาณต้นแบบเป็นสัญญาณรูปไซน์ขนาดเล็กที่มีความเพี้ยนต่ำซึ่งสามารถปรับขนาด, ความถี่ และความต่างเฟสระหว่างสัญญาณทั้งสองได้โดยอาศัยตัวแปลงดิจิทัลเป็นแอนะล็อกขนาด 16 บิต (16-bit digital to analog converter;DAC) และวงจรกรองความถี่ผ่านต่ำมาช่วยในการสร้างสัญญาณ อีกหน้าที่คือติดต่อสื่อสารกับผู้ใช้ผ่านทางส่วนรับค่า (Keyboard) และส่วนแสดงผล (Display panel) การทำงานทั้งหมดของส่วนนี้ควบคุมโดยไมโครคอนโทรลเลอร์ ส่วนขยายสัญญาณประกอบด้วยวงจรขยายแรงดัน (Voltage amplifier circuit) สำหรับช่องสัญญาณแรงดันและวงจรแปลงแรงดันเป็นกระแส (Voltage to current converter) สำหรับช่องสัญญาณกระแส ส่วนประกอบหลักภายในคืออออปแอมป์กำลังสูงและหม้อแปลงกำลัง ส่วนนี้ทำหน้าที่ขยายสัญญาณต้นแบบให้มีขนาดใหญ่

เพียงพอสำหรับระบบทดสอบความแม่นยำโดยจะต้องไม่ทำให้เกิดความเพี้ยนที่สัญญาณออกมากเกินไป ซึ่งในส่วนนี้จะอาศัยหลักของการป้อนกลับเข้ามาช่วยควบคุมความเพี้ยนของสัญญาณออกที่ได้

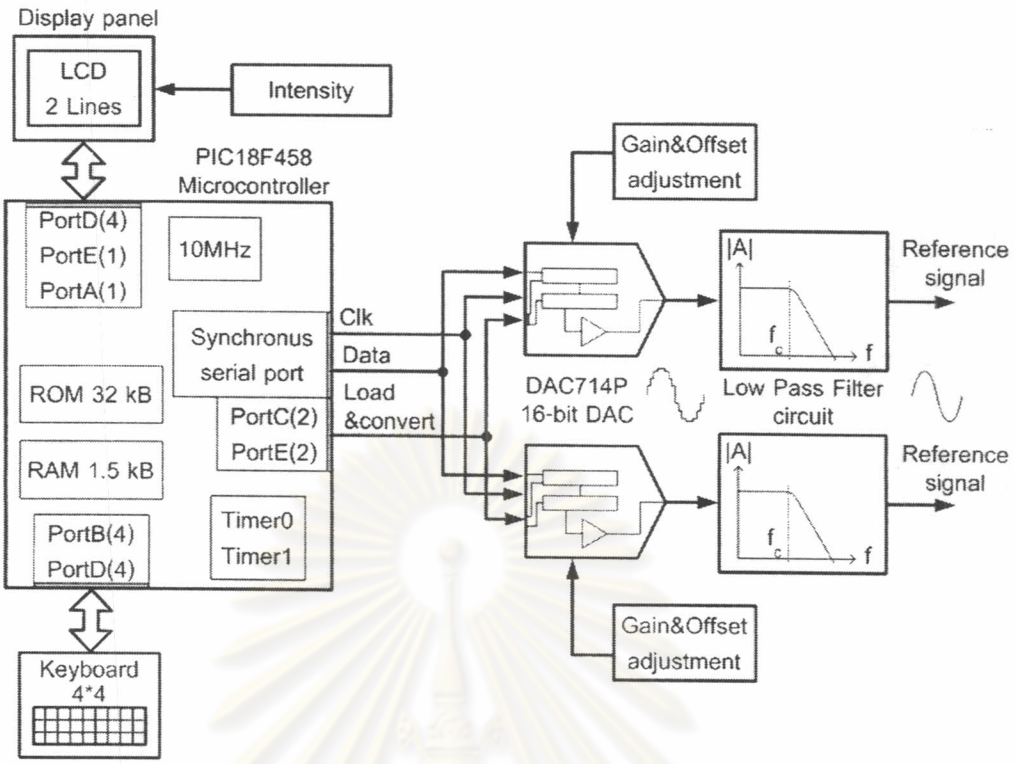



*รูปที่ 3.1 โครงสร้างของแหล่งกำเนิดสัญญาณแรงดันและกระแส*

สำหรับค่าที่ออกแบบไว้ สัญญาณต้นแบบที่ได้จากส่วนสร้างสัญญาณต้นแบบ (สัญญาณที่ออกมาจากวงจรกรองความถี่ผ่านต่ำ) จะมีลักษณะเป็นรูปไซน์ที่มีค่ายอด (Peak value) เป็น $\pm 5$ V อัตราขยายแรงดันของวงจรขยายแรงดันมีค่าประมาณ 80 เท่า ทำให้ได้สัญญาณออกแรงดันมีค่ายอดได้ประมาณ 280 $V_{rms}$ และอัตราขยายของวงจรแปลงสัญญาณแรงดันเป็นกระแสมีค่าประมาณ 15 $A_{rms}/V_{rms}$ ทำให้ได้ขนาดสัญญาณกระแสด้านออกสูงสุดได้ประมาณ 50 $A_{rms}$

## 3.3 ส่วนสร้างสัญญาณต้นแบบ

ส่วนสร้างสัญญาณต้นแบบนั้นมีโครงสร้างโดยละเอียดดังรูปที่ 3.2 ส่วนนี้ประกอบไปด้วย 5 ส่วนด้วยกัน คือ ไมโครคอนโทรลเลอร์ ตัวแปลงดิจิทัลเป็นแอนะล็อกขนาด 16 บิต (16-bit digital to analog converter ;DAC) วงจรกรองความถี่ผ่านต่ำ (Low pass filter) ปุ่มกดรับค่า (Keyboard) และ จอแสดงผล (Display panel) ซึ่งมีรายละเอียดดังต่อไปนี้




*รูปที่ 3.2 โครงสร้างของส่วนสร้างสัญญาณต้นแบบ*

3.3.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ทำหน้าที่ควบคุมการทำงานของปุ่มกดรับค่า จอแสดงผล รวมไปถึงควบคุมการสร้างสัญญาณต้นแบบโดยส่งสัญญาณดิจิทัลไปสั่งงานตัวแปลงดิจิทัลเป็นแอนะล็อก สำหรับในงานวิจัยนี้เลือกใช้ไมโครคอนโทรลเลอร์ของบริษัทไมโครชิพ เบอร์ 18F458 ซึ่งมีลักษณะสมบัติที่สำคัญดังนี้

- มี FLASH ROM (Read only memory) ภายในขนาด 32 กิโลไบต์
- มี RAM (Random access memory) ภายในขนาด 1.5 กิโลไบต์
- ทำงานที่ความเร็ว 10 MHz
- มีตัวจับเวลา (Timer) ภายใน 4 ตัว เป็นขนาด 16 บิต 3 ตัว และขนาด 8 บิต 1 ตัว
- มีช่องทางเข้าออก (IO Port) จำนวน 34 ช่อง นั่นคือ ช่อง A 7 ช่อง, ช่อง B 8 ช่อง, ช่อง C 8 ช่อง, ช่อง D 8 ช่อง และ ช่อง E จำนวน 3 ช่อง
- มีตัวแปลงดิจิทัลเป็นแอนะล็อก (Digital to analog converter;ADC) ภายใน 1 ตัว สามารถเลือกช่องสัญญาณเข้าได้ 8 ช่องทาง
- มีวงจรรับส่งสัญญาณอนุกรมแบบ USART (Universal synchronous asynchronous receiver and transmitter) เพื่อใช้สื่อสารกับคอมพิวเตอร์หรืออุปกรณ์สื่อสารแบบอนุกรมอื่น ๆ

- มีวงจรสื่อสารอนุกรมแบบ SPI (Serial peripheral interface) ภายในซึ่งในที่นี้ใช้สำหรับสื่อสารสั่งงานตัวแปลงดิจิทัลเป็นแอนะล็อก

3.3.2 ตัวแปลงดิจิทัลเป็นแอนะล็อก

วิทยานิพนธ์นี้เลือกใช้ตัวแปลงดิจิทัลเป็นแอนะล็อกขนาด 16 บิต ของบริษัทเท็กซัส อินสตรูเม็นท์ (Texas Instruments) เบอร์ DAC714P ซึ่งมีลักษณะสมบัติดังนี้

- ติดต่อสื่อสารแบบอนุกรม อัตราข้อมูลสูงสุดเป็น 10 MBPS ความถี่สัญญาณนาฬิกาสูงสุดเป็น 10 MHz

- โครงสร้างภายในแบบบัฟเฟอร์สองชั้น (Double buffer)

- สามารถเลือกขนาดแรงดันด้านออกสูงสุดได้ 3 ขนาดคือ $\pm$ 10 V, $\pm$ 5 V, 0 – 10 V

- ความผิดพลาดของภาวะเชิงเส้น (Linearity error) มีค่าเป็น $\pm$ 4 LSB และ ความผิดพลาดของภาวะเชิงเส้นของผลต่าง (Differential linearity error) มีค่าเป็น $\pm$ 4 LSB

- รูปแบบของข้อมูลที่ใช้ติดต่อสื่อสารเป็นแบบ binary two's complement

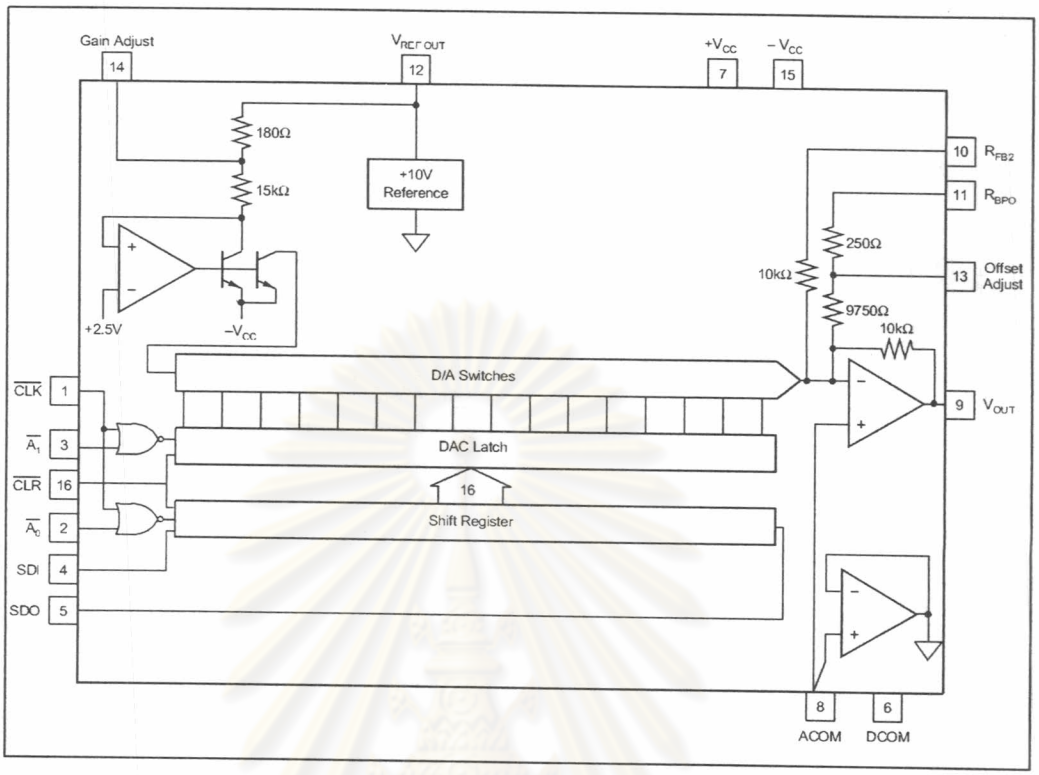
- มีวงจรสร้างสัญญาณแรงดันอ้างอิงของการแปลงค่าอยู่ภายใน

โครงสร้างภายในของตัวแปลงดิจิทัลเป็นแอนะล็อกมีลักษณะดังรูปที่ 3.3 [6] โดยประกอบไปด้วย 4 ส่วนหลัก ๆ ด้วยกันดังนี้

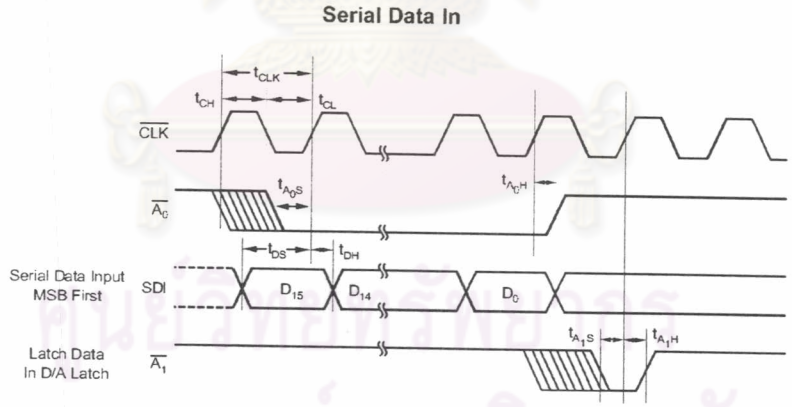
1. วงจรสร้างสัญญาณแรงดันอ้างอิง
2. วงจรสำหรับรับค่าสัญญาณดิจิทัลขาเข้า
3. วงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อก
4. วงจรปรับแก้อัตราขยายและออฟเซตของการแปลงค่า (Gain and offset adjustment circuit)

วงจรสร้างสัญญาณแรงดันอ้างอิงทำหน้าที่สร้างสัญญาณแรงดันไฟตรงที่มีความแม่นยำและมีค่าคงที่ขนาด 10 V เพื่อให้นำไปใช้เป็นสัญญาณแรงดันอ้างอิงของการแปลงค่าสัญญาณจากดิจิทัลเป็นแอนะล็อก โครงสร้างแบบบัฟเฟอร์สองชั้นประกอบไปด้วยเรจิสเตอร์หลักสองตัวคือ เรจิสเตอร์แบบเลื่อน (Shift register) และ เรจิสเตอร์แลตช์ (DAC latch register) โดยเรจิสเตอร์แบบเลื่อนเป็นเรจิสเตอร์ขนาด 16 บิต มีหน้าที่เลื่อนค่าข้อมูลเข้ามาทางขาสัญญาณเข้าของอุปกรณ์ตามจังหวะสัญญาณนาฬิกา โดยสัญญาณที่ควบคุมการเลื่อนค่าคือสัญญาณจากขา $A_0$ ส่วนข้อมูลจะไหลเข้าทางขา SDI (Serial Data In) เรจิสเตอร์แลตช์เป็นเรจิสเตอร์ขนาด 16 บิต ที่มีหน้าที่ย้ายข้อมูลขนาด 16 บิตที่ได้รับจากเรจิสเตอร์แบบเลื่อนเข้าไปในวงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อก หรืออีกนัยหนึ่งก็คือการสั่งให้อุปกรณ์แปลงค่าสัญญาณนั่นเอง โดยสัญญาณที่ควบคุมการย้ายข้อมูลคือสัญญาณจากขา $A_1$ การทำงานและการรับค่าสัญญาณของอุปกรณ์ทั้งหมด

จะเป็นแบบเข้าจังหวะกับสัญญาณนาฬิกา (Synchronous) ดังแสดงในแผนภาพเวลาของการเลื่อนข้อมูลเข้าอุปกรณ์และการสั่งแปลงค่าดังรูปที่ 3.4



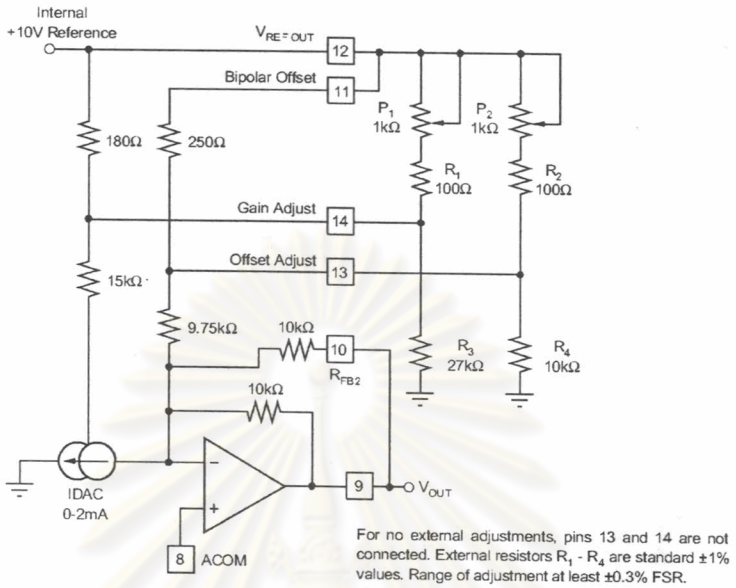

*รูปที่ 3.3 โครงสร้างส่วนประกอบภายในของตัวแปลงดิจิทัลเป็นแอนะล็อก DAC714P*




รูปที่ 3.4 แผนภาพเวลาของการเลื่อนค่าและการสั่งแปลงค่าของ DAC714P

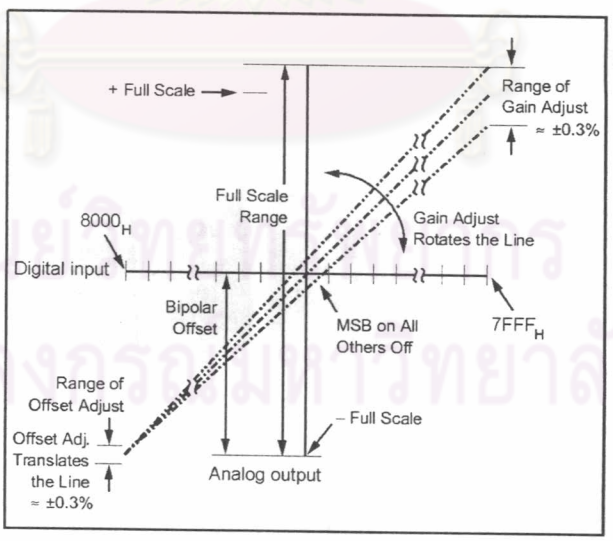
จากรูปจะพบว่าข้อมูลจะถูกเลื่อนเข้าตัวแปลงดิจิทัลเป็นแอนะล็อกตามสัญญาณนาฬิกา (เมื่อสัญญาณนาฬิกาเปลี่ยนสถานะจาก 0 -->1) เมื่อสัญญาณที่ขา $A_0$ เป็นลอจิก 0 และข้อมูลจะถูกแปลงค่าเมื่อสัญญาณ $A_1$ เป็นลอจิก 0 ตามจังหวะสัญญาณนาฬิกาเช่นกัน โดยสำหรับในวิทยานิพนธ์นี้เลือกใช้ความถี่สัญญาณนาฬิกาเป็น 10 MHz ซึ่งเป็นค่าสูงสุดที่อุปกรณ์จะรับได้ ทั้งนี้ก็เพื่อให้การแปลงค่าทำโดยใช้เวลาน้อยที่สุด

ก่อนใช้งาน DAC714P จะต้องมีการปรับแก้อัตราขยายและออฟเซตของการแปลงค่าเพื่อให้สัญญาณออกที่ได้มีสมบัติที่ถูกต้องและมีความแม่นยำตามที่ระบุโดยผู้ผลิต โดยจะต้องต่อวงจรภายนอกเพิ่ม ดังแสดงในรูปที่ 3.5




*รูปที่ 3.5 วงจรปรับค่าอัตราขยายและออฟเซตของ DAC714P*

ขั้นตอนการปรับแก้จะแบ่งเป็นสองขั้นคือปรับแก้ออฟเซตก่อนแล้วค่อยปรับแก้อัตราขยาย รูปที่ 3.6 แสดงกราฟความสัมพันธ์ระหว่างสัญญาณดิจิทัลด้านเข้าและสัญญาณแอนะล็อกด้านออก โดยลักษณะสมบัติของตัวแปลงดิจิทัลเป็นแอนะล็อกนี้จะเป็นเส้นตรง

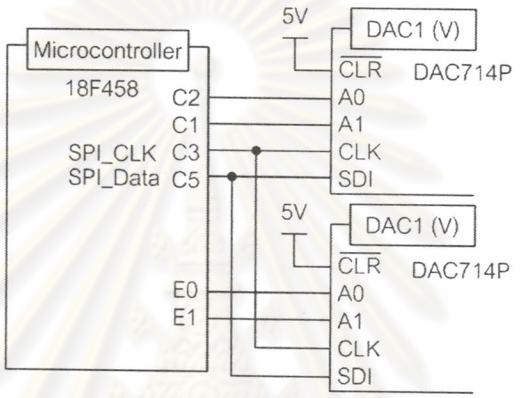



*รูปที่ 3.6 การปรับออฟเซตและอัตราขยายเพื่อปรับลักษณะสมบัติของ DAC714P*

ตัวแปลงนี้ถูกออกแบบไว้ให้ปรับแก้อัตราขยายและออฟเซตได้ การปรับแก้ออฟเซตหมายถึงการปรับเลื่อนกราฟขึ้นหรือลงในแนวตั้ง และการปรับแก้อัตราขยายหมายถึงการปรับความชันของกราฟ เนื่องจากในวิทยานิพนธ์นี้เลือกใช้ช่วงค่าของสัญญาณออกเป็น –5 V ถึง 5 V ดังนั้นการ

ปรับออฟเซตในกรณีนี้จะทำโดยใส่ค่าสัญญาณเข้าเป็น 8000h (ค่าลบที่ต่ำที่สุด) และปรับ $P_2$ ให้ได้สัญญาณออกที่ –5 V จากนั้นจะทำการปรับแก้อัตราขยายต่อโดยใส่สัญญาณเข้าค่ามากที่สุดซึ่งก็คือ 7FFFh แล้วปรับ $P_1$ ให้ได้สัญญาณออกมีขนาดเป็น 5 V นั่นเอง

ไมโครคอนโทรลเลอร์จะสื่อสารกับ DAC714P โดยใช้วงจรสื่อสารอนุกรม SPI ที่มีภายในซึ่งเป็นการสื่อสารแบบเข้าจังหวะโดยในที่นี้จะใช้การส่งแบบทางเดียวคือจากไมโครคอนโทรลเลอร์ไปยังตัวแปลงดิจิทัลเป็นแอนะล็อก เนื่องจากต้องสร้างสัญญาณต้นแบบ 2 ช่องสัญญาณ ดังนั้นไมโครคอนโทรลเลอร์จะต้องสั่งงานตัวแปลงดิจิทัลเป็นแอนะล็อกสองตัว การต่อวงจรเพื่อส่งสัญญาณจากไมโครคอนโทรลเลอร์ไปยังตัวแปลงดิจิทัลเป็นแอนะล็อกเป็นดังรูปที่ 3.7




*รูปที่ 3.7 การต่อสายสื่อสารระหว่างไมโครคอนโทรลเลอร์และตัวแปลงดิจิทัลเป็นแอนะล็อก*

จากรูป เนื่องจากตัวแปลงดิจิทัลเป็นแอนะล็อกมีโครงสร้างแบบบัฟเฟอร์สองชั้นทำให้ไมโครคอนโทรลเลอร์สามารถสั่งแปลงค่าในตัวแปลงดิจิทัลเป็นแอนะล็อกทั้งสองได้อย่างอิสระต่อกันโดยสั่งผ่านทางขาสั่งแปลงค่าของตัวแปลงแต่ละตัว ( ขา $A_1$) ส่วนการเลื่อนค่าเข้าไปในตัวแปลงนั้นจะทำไม่พร้อมกันโดยส่งสัญญาณเลือกว่าจะส่งค่าให้ตัวแปลงตัวไหนผ่านทางขา $A_0$ ซึ่งจากการทดลองส่งข้อมูลที่ความถี่สัญญาณนาฬิกา 10 MHz ว่าการส่งค่าให้ตัวแปลงดิจิทัลเป็นแอนะล็อก 1 ไบต์ จะใช้เวลา 0.8 μs

การออกแบบลายวงจรของตัวแปลงดิจิทัลเป็นแอนะล็อกต้องคำนึงถึงผลของสัญญาณรบกวนเป็นอย่างมาก สำหรับในการวิจัยนี้ได้ออกแบบลายวงจรโดยคำนึงถึงข้อกำหนดหลัก ๆ 3 ประการดังนี้

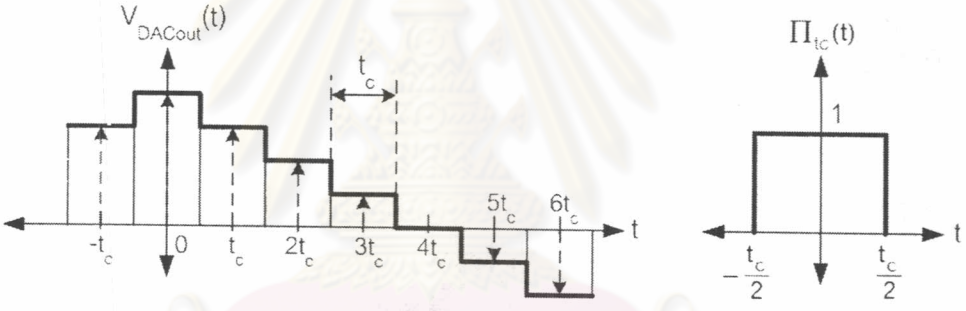
1. ใช้กลวิธีการสร้างระนาบกราวด์ (Ground plane)
2. แยกลายวงจรส่วนที่เป็นสัญญาณแอนะล็อกและสัญญาณดิจิทัลออกจากกันโดยส่วนพื้นที่ใต้อุปกรณ์ให้ใช้ระนาบกราวด์ของสัญญาณแอนะล็อกคลุม และแยกระนาบกราวด์ของสัญญาณแอนะล็อกและของสัญญาณดิจิทัลออกจากกันด้วย แล้วเชื่อมเข้าด้วยกันโดยมี Ferrite bead กั้น

3. ต่อตัวเก็บประจุคร่อมทั้งสัญญาณไฟเลี้ยงบวกและลบเพื่อลดสัญญาณรบกวนที่เกิดในสายไฟเลี้ยง

3.3.3 <u>วงจรกรองความถี่ผ่านต่ำ</u>

วงจรกรองความถี่ผ่านต่ำทำหน้าที่กรองส่วนประกอบความถี่สูงที่เกิดจากอัตราการแปลงสัญญาณของตัวแปลงดิจิทัลเป็นแอนะล็อกออกไป เนื้อหาในส่วนนี้ ขั้นแรกจะวิเคราะห์ส่วนประกอบความถี่ของสัญญาณออกจากตัวแปลงดิจิทัลเป็นแอนะล็อก จากนั้นจะเลือกค่าความถี่หักมุมและอันดับของวงจรกรองที่จะใช้ และสุดท้ายจึงออกแบบเป็นวงจร

วิทยานิพนธ์นี้ต้องการสร้างสัญญาณต้นแบบเป็นสัญญาณรูปไซน์ที่มีขนาดค่ายอดสูงสุดเป็น 5 V การสร้างสัญญาณทำโดยใช้ตัวแปลงดิจิทัลเป็นแอนะล็อก สัญญาณออกจากตัวแปลงดิจิทัลเป็นแอนะล็อกที่ได้จะมีลักษณะเป็นสัญญาณขั้นรูปไซน์ที่มีความกว้างของแต่ละขั้นเท่ากับคาบเวลาของการแปลงค่า ($t_c$) ของตัวแปลงดิจิทัลเป็นแอนะล็อก ดังแสดงเป็นกราฟเส้นทึบในรูปที่ 3.8 ด้านซ้ายมือ




*รูปที่ 3.8 ลักษณะสัญญาณขั้นรูปไซน์ที่ได้จากตัวแปลงดิจิทัลเป็นแอนะล็อก*

สัญญาณทางเวลาที่แสดงในรูปที่ 3.8 ด้านซ้ายนั้นสามารถเขียนเป็นสมการได้ดังสมการที่ 3.1

$$V_{DACout}(t) = \Pi_{tc}(t) \otimes [x(t) \cdot \delta_T(t)] \quad \text{.....สมการที่ 3.1}$$

โดยที่ $\Pi_{tc}(t)$ เป็นสัญญาณรูปสี่เหลี่ยมที่มีความกว้าง $t_c$ ลักษณะดังรูปที่ 3.8 ด้านขวามือ และ $\delta_T(t)$ เป็นฟังก์ชันขบวนพัลส์ (Pulse train function) ที่มีระยะห่างระหว่างพัลส์เป็น $t_c$ ซึ่งมีสมการเป็น

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nt_c) \quad \text{.....สมการที่ 3.2}$$

เครื่องหมาย $\otimes$ ในสมการที่ 3.1 แทนการคอนโวลูชัน (Convolution) x(t) เป็นสัญญาณต้นแบบที่ต้องการสร้างรูปไซน์ที่มีค่ายอดเป็น V และความถี่เป็น $f_s$

$$x(t) = V \cdot \sin(2\pi f_s t) \quad \text{.....สมการที่ 3.3}$$

ส่วนสัญญาณลูกศรที่เป็นเส้นประของสัญญาณด้านซ้ายมือในรูปที่ 3.8 จะแทนสัญญาณ $[x(t) \bullet \delta_T(t)]$ นั่นเอง สัญญาณต่าง ๆ ที่กล่าวมาข้างต้นมีผลการแปลงฟูริเยร์ (Fourier transform) เป็น

$$\prod_{tc}(t) \xrightarrow{\text{fourier transform}} \frac{\sin(\pi f t_c)}{\pi f} \quad \text{.....สมการที่ 3.4}$$

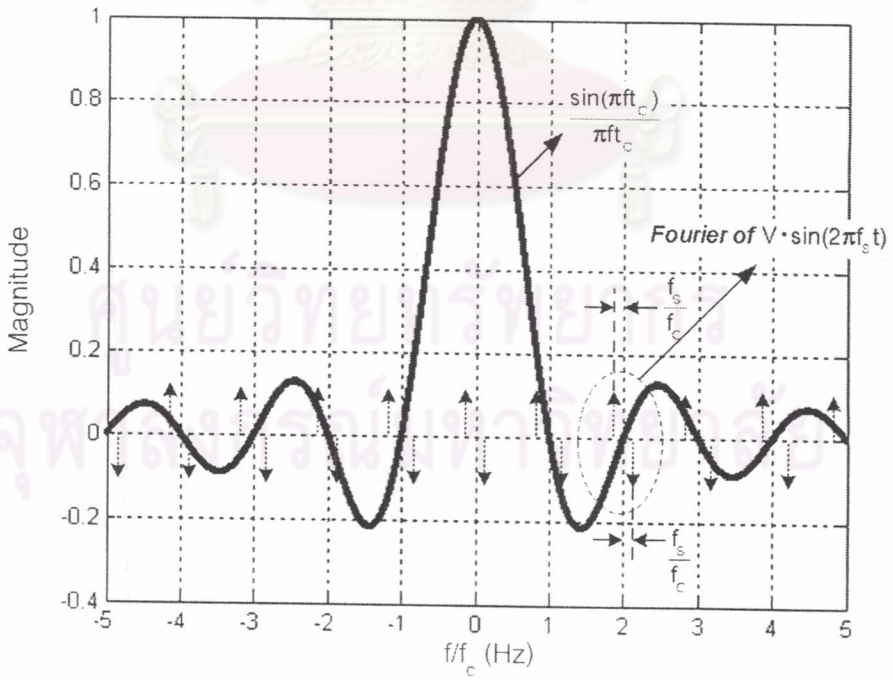
$$\delta_T(t) \xrightarrow{\text{fourier transform}} f_c \cdot \sum_{n=-\infty}^{\infty} \delta(f - nf_c) \qquad ;f_c = \frac{1}{t_c} \quad \text{.....สมการที่ 3.5}$$

$$\sin(2\pi f_s t) \xrightarrow{\text{fourier transform}} \frac{j}{2f_s} \cdot [\frac{1}{2}\delta(\frac{f}{2f_s} + \frac{1}{2}) - \frac{1}{2}\delta(\frac{f}{2f_s} - \frac{1}{2})] \quad \text{.....สมการที่ 3.6}$$

จากสมการที่กล่าวข้างต้น เราสามารถหาผลการแปลงฟูริเยร์ของสมการที่ 3.1 ได้เป็น

$$V_{DACout}(f) = \frac{\sin(\pi f t_c)}{\pi f t_c} \cdot \sum_{n=-\infty}^{\infty} \left( \frac{Vj}{2f_s} \cdot [\frac{1}{2}\delta(\frac{f - nf_c}{2f_s} + \frac{1}{2}) - \frac{1}{2}\delta(\frac{f - nf_c}{2f_s} - \frac{1}{2})] \right) \text{...สมการที่ 3.7}$$

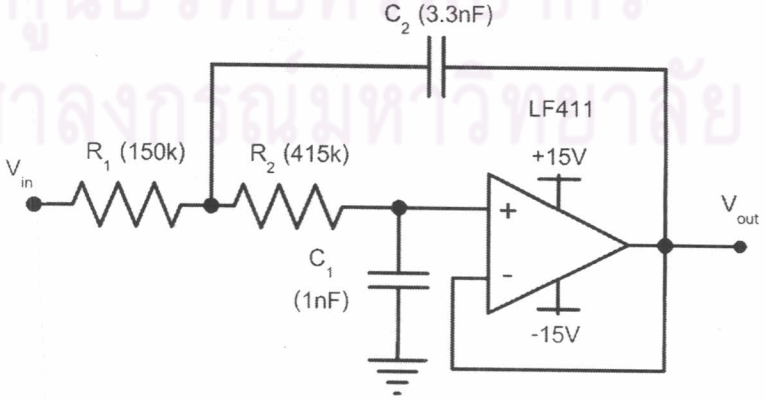
สังเกตว่าสมการในพจน์อนุกรมด้านขวามือก็คือการเลื่อนผลการแปลงฟูริเยร์ของสัญญาณไซน์ไปอยู่ที่ความถี่ $nf_c$ ส่วนพจน์ $\sin(\pi f t_c)/(\pi f t_c)$ ด้านซ้ายมือคือการลดทอนสัญญาณที่ความถี่ต่าง ๆ ซึ่งสามารถแสดงเป็นกราฟของสองส่วนนี้ได้ดังรูปที่ 3.9




*รูปที่ 3.9 กราฟแสดงส่วนประกอบสัญญาณออกจากตัวแปลงดิจิทัลเป็นแอนะล็อกในโดเมนความถี่*

จากกราฟจะเห็นว่าส่วนประกอบความถี่ของสัญญาณออกจะอยู่ที่ความถี่ $\pm f_s$ และความถี่ $nf_c \pm f_s$ เมื่อ n มีค่าตั้งแต่ลบอนันต์ถึงอนันต์ ขนาดของสัญญาณไซน์ดังกล่าวจะถูกลดทอนมากขึ้นที่ความถี่มากขึ้นโดยพจน์ $\sin(\pi ft_c)/(\pi ft_c)$ ในสมการที่ 3.7 สำหรับระบบที่จะออกแบบนั้นค่าความถี่ของการแปลงค่า $f_c$ จะมีค่าอยู่ในช่วง 9-11 kHz เพื่อสร้างสัญญาณออกความถี่ 45-55 Hz (รายละเอียดของขั้นตอนวิธีการสร้างสัญญาณอยู่ในหัวข้อ 3.4)ในการออกแบบ จึงเลือกความถี่ค่ากลางมาใช้คือ ความถี่สัญญาณออกที่ต้องการ $f_s$ เป็น 50 Hz ซึ่งเป็นสัญญาณที่ถูกใช้ในการทดสอบความแม่นยำมากที่สุด ซึ่งต้องใช้ความถี่ของการแปลงค่าเป็น 10 kHz ดังนั้นจากข้อมูลที่กล่าวมาข้างต้นทำให้ทราบว่าสัญญาณออกที่ได้จากตัวแปลงดิจิทัลเป็นแอนะล็อกจะประกอบด้วยความถี่ที่ต้องการ 50 Hz และความถี่ที่ถูกเลื่อนไปอยู่บนความถี่ของการแปลงค่าคือที่ 9950 Hz และ 10050 Hz ซึ่งเมื่อแทนค่าพจน์ $\sin(\pi ft_c)/(\pi ft_c)$ จะพบว่าที่จุดนี้ขนาดถูกลดทอนไป 45.977 dB จากค่าขนาดที่ความถี่ที่ต้องการ และที่ความถี่สูงกว่านี้ก็จะถูกลดทอนมากกว่านี้ ในวิทยานิพนธ์นี้เลือกให้ส่วนประกอบความถี่สูงที่ไม่ต้องการลดทอนลงไปมากกว่า 100 dB ดังนั้นต้องออกแบบวงจรกรองให้ลดทอนสัญญาณที่ความถี่ 9950 Hz ขึ้นไปออกอีก 54 dB (ถูกลดทอนโดยพจน์ $\sin(\pi ft_c)/(\pi ft_c)$ไปแล้ว 45.977 dB)

วงจรกรองที่เลือกใช้เป็นวงจรกรองความถี่ผ่านต่ำอันดับสองชนิดบัตเทอร์เวิร์ทที่มีอัตราขยายเป็นหนึ่ง ($2^{nd}$ order unity-gain Butterworth low pass filter) [7] เนื่องจากให้ผลตอบในช่วงความถี่ผ่านที่เรียบ โดยเลือกความถี่หักมุมให้ต่ำที่สุดเพื่อให้การลดทอนที่ความถี่สูงเกิดมากขึ้นและต้องไม่ทำให้ขนาดสัญญาณออกที่ความถี่ที่ต้องการสร้างลดทอนไป ค่าความถี่ที่เลือกคือ 350 Hz ซึ่งจะทำให้วงจรลดทอนสัญญาณที่ความถี่ 9950 Hz ลงไป 58.15 dB ทำให้สัญญาณที่ความถี่ 9950 Hz มีขนาดต่ำกว่าขนาดสัญญาณออกที่ความถี่ 50 Hz เป็น 104.147 dB โครงสร้างของวงจรเป็นดังรูปที่ 3.10




*รูปที่ 3.10 โครงสร้างของวงจรกรองความถี่ผ่านต่ำ*

จากวงจร เราสามารถหาฟังก์ชันถ่ายโอนของวงจรนี้ได้เป็นดังนี้

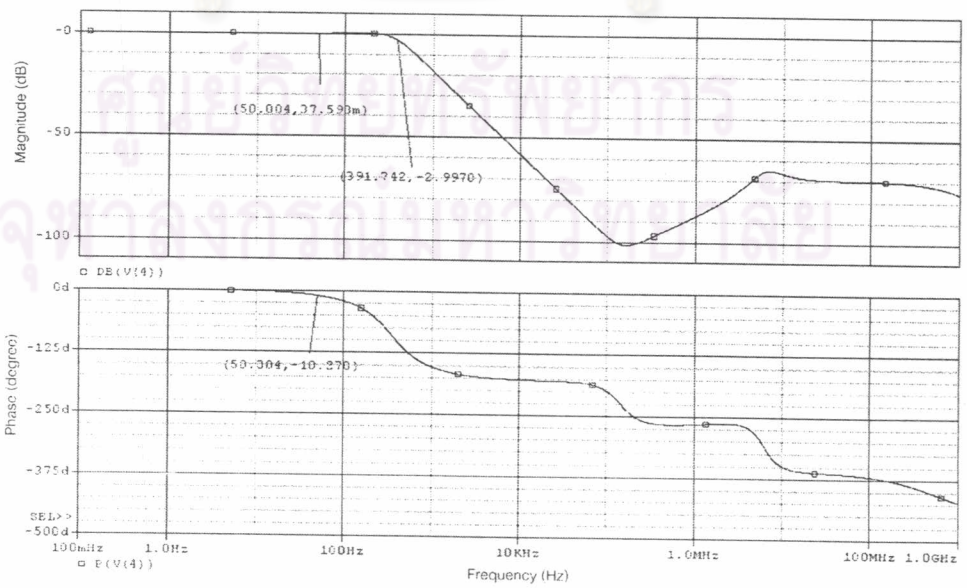
$$\frac{V_{out}}{V_{in}}(s) = \frac{1}{1 + C_1(R_1 + R_2)S + R_1R_2C_1C_2S^2} \quad \text{.....สมการที่ 3.8}$$

ฟังก์ชันถ่ายโอนนี้ใช้สำหรับการเลือกค่าอุปกรณ์ที่จะใช้ โดยจะเลือกค่าให้สัมประสิทธิ์ของสมการลักษณะสมบัติ (Characteristic equation) สอดคล้องกับสัมประสิทธิ์ของวงจรกรองชนิดบัตเทอร์เวิร์ทอันดับสองที่มีความถี่หักมุมที่ 350 Hz จะได้ค่าดังรูปที่ 3.10 ซึ่งเมื่อแทนในสมการที่ 3.8 จะได้ฟังก์ชันถ่ายโอนของวงจรที่ออกแบบเป็น

$$\frac{V_{out}}{V_{in}}(s) = \frac{1}{1 + 5.65*10^{-4}\cdot S + 2.05*10^{-7}\cdot S^2} \quad \text{.....สมการที่ 3.9}$$

จากสมการ เราสามารถหาความถี่หักมุมของฟังก์ชันถ่ายโอนนี้ได้เป็น 351.15 Hz วงจรมีค่าอัตราการหน่วง (Damping ratio) เป็น 0.623 จากนั้นแทนค่า s=jω และหาขนาดของอัตราขยายที่ 50 Hz ได้เป็น 1.0043 เท่า และคำนวณความต่างเฟสที่ 50 Hz ได้เป็น 10.27 องศา โดยสาเหตุที่ค่าอัตราขยายที่ 50 Hz มีค่ามากกว่า 1 เล็กน้อยนั้นเนื่องมาจากการที่ผลตอบเชิงความถี่ของขนาดของวงจรมียอด (Peak) ของสัญญาณ ซึ่งเราจะสามารถสังเกตได้จากค่าอัตราการหน่วง โดยจาก [8] กล่าวว่าผลตอบขนาดของระบบอันดับสองจะเรียบก็ต่อเมื่อค่าอัตราการหน่วงมากกว่า 0.707 แต่ในที่นี้เนื่องจากข้อจำกัดในการเลือกค่าอุปกรณ์ทำให้ระบบมีสมบัติที่คลาดเคลื่อนไปเล็กน้อย ซึ่งขนาดของค่ายอดสูงสุดของกรณีนี้จะน้อยมากเนื่องจากค่าอัตราการหน่วงใกล้ 0.707 นั่นเอง

จากการทดลองจำลองการทำงานของวงจรด้วยโปรแกรม Orcad Pspice จะพบว่าผลตอบสนองเชิงความถี่ของวงจร (Bode plot) เป็นดังรูปที่ 3.11

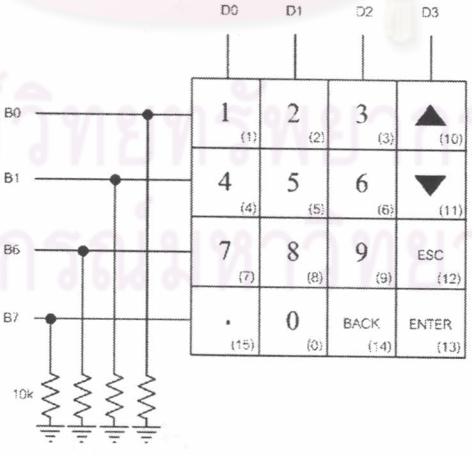



*รูปที่ 3.11 ผลตอบสนองเชิงความถี่ของวงจรกรองความถี่ผ่านต่ำ*

จากรูป กราฟบนแสดงผลตอบสนองเชิงความถี่ของขนาดซึ่งมีแกนตั้งเป็นขนาดของสัญญาณออกเทียบกับสัญญาณเข้าหน่วยเป็นเดซิเบล และกราฟล่างแสดงผลตอบสนองเชิงความถี่ของเฟสของสัญญาณออกเทียบกับสัญญาณเข้าหน่วยเป็นองศา ตัวเลขในวงเล็บที่เห็นในกราฟเป็นการแสดงค่าในกราฟที่ถูกเส้นตรงชี้ เลขตัวแรกคือความถี่หน่วยเป็น Hz (ค่าแกนนอน) และเลขตัวหลังคือขนาดหรือเฟสในกราฟนั้น ๆ (ค่าแกนตั้ง) ซึ่งจากค่าที่แสดงจะเห็นว่าวงจรมีความถี่ 3 dB ที่ค่า 391.74 Hz ซึ่งไม่เท่ากับที่ออกแบบไว้ แต่ยังยอมรับได้เนื่องจากต่างจากค่าที่ออกแบบไปไม่มากนัก (ที่ความหักมุมถี่ค่านี้จะทำให้สัญญาณที่ 9950 Hz ลดทอนไป 58.1075 dB จากที่ 50 Hz และสัญญาณที่ 9945 Hz ลดทอนไป 58.1070 dB จากที่ 55 Hz) และอัตราขยายที่ 50 Hz (ความถี่มูลฐานของสัญญาณออกที่ต้องการ) เป็น 1.0043 เท่า เฟสของสัญญาณออกที่ความถี่ 50 Hz นั้น จะตามสัญญาณเข้าอยู่ 10.27 องศา ส่วนลักษณะผลตอบที่เห็นที่ความถี่มากกว่า 100 kHz นั้นจะเป็นผลมาจากลักษณะสมบัติภายในของออปแอมป์ที่ใช้ และจากค่าที่คำนวณและจำลองการทำงานได้ จะเห็นว่าสัญญาณออกที่ต้องการจะมีเฟสตามหลังสัญญาณเข้าอยู่ค่าหนึ่งซึ่งอาจเป็นผลให้ความต่างเฟสของสัญญาณที่สร้างอาจคลาดเคลื่อนไปจากค่าที่ต้องการได้ แต่อาศัยหลักการที่ว่าต้องมีวงจรกรองทั้งสองช่องสัญญาณดังนั้นถ้าใช้วงจรกรองที่มีลักษณะเหมือนกันทั้งสองช่องสัญญาณ สัญญาณออกที่ได้ก็ควรจะถูกเลื่อนไปเท่า ๆ กันทำให้เกิดการหักล้างความคลาดเคลื่อนในส่วนนี้ไปได้

3.3.4 <u>ปุ่มกดรับค่า</u>

ปุ่มกดรับค่าเป็นอุปกรณ์สำหรับผู้ใช้กดค่าเพื่อสื่อสารกับแหล่งกำเนิดสัญญาณ โดยในวิทยานิพนธ์นี้ใช้ปุ่มกดรับค่าจำนวน 16 ปุ่ม บนแป้นพิมพ์มี 4 แถว 4 คอลัมน์ ดังรูปที่ 3.12



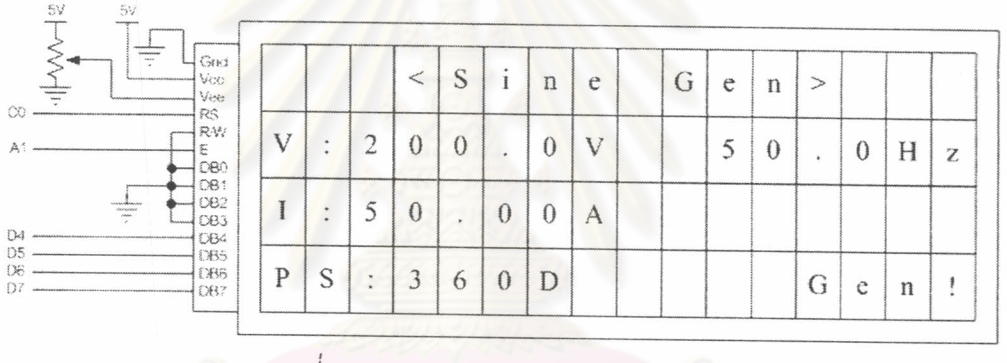

*รูปที่ 3.12 โครงสร้างของปุ่มกดรับค่า*

สัญลักษณ์บนปุ่มจะประกอบไปด้วยตัวเลข สัญลักษณ์จุดและปุ่มพิเศษอื่น ๆ อีก 5 ปุ่ม การทำงานของแป้นพิมพ์นี้คือ เมื่อกดปุ่ม สายไฟของแถวและคอลัมน์ของปุ่มที่ถูกกดนั้น ๆ จะต่อถึงกัน

เช่น เมื่อกดปุ่มเลข 5 สายไฟ B1 จะต่อกับ D1 เป็นต้น สายไฟเหล่านี้จะต่อไปถึงไมโครคอนโทรลเลอร์โดยตัวอักษรที่ระบุอยู่ปลายสายในรูปหมายถึงช่องทางเข้าออก (Port) ที่สายนั้นถูกต่อเข้าไป นอกจากนี้จะสังเกตว่าทุก ๆ แถวจะมีการต่อตัวต้านทานลงดินด้วยเพื่อกำหนดสถานะตอนไม่กดให้เป็นสถานะ Low นั่นเอง สำหรับไมโครคอนโทรลเลอร์จะอ่านค่าปุ่มที่ถูกกดโดยการป้อนค่าสถานะ High ให้กับคอลัมน์ทีละคอลัมน์แล้วตรวจสอบสถานะที่แถวว่ามีแถวใดเป็น High ตามหรือไม่ ถ้ามีแสดงว่ามีการกดปุ่มและเก็บค่าปุ่มต่าง ๆ เป็นรหัสเพื่อนำไปประมวลผลต่อ โดยค่าที่เก็บของแต่ละปุ่มคือค่าในวงเล็บที่มุมล่างขวาของแต่ละปุ่มตามรูป

3.3.5 <u>จอแสดงผล</u>

จอแสดงผลที่ใช้เป็นจอแบบผลึกเหลว (Liquid crystral display ;LCD) ขนาด 16 ตัวอักษร 4 บรรทัด โดยภายในตัวอุปกรณ์มีวงจรขับ (Driver circuit) สำหรับรับค่าคำสั่งซึ่งจะส่งแบบ 4 บิต (DB4-DB7) ร่วมกับขาสั่งงานพิเศษอีก 2 ขา (RS,E) โดยวงจรจะเป็นดังรูปที่ 3.13



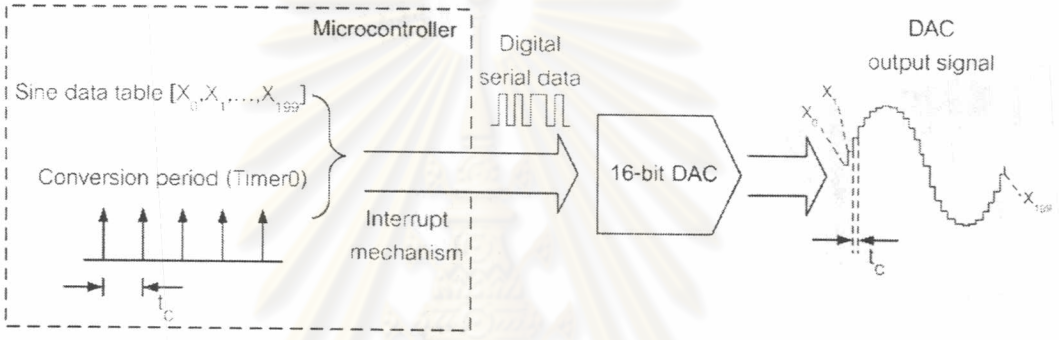

*รูปที่ 3.13 จอแสดงผลและการต่อใช้งาน*

จากรูป $V_{ee}$ คือ แรงดันสำหรับปรับความเข้มของตัวอักษรที่แสดง และตัวอักษรด้านซ้ายสุดคือช่องทางเข้าออกของไมโครคอนโทรลเลอร์ที่ใช้ควบคุมการทำงานของอุปกรณ์นี้ ค่าบนจอจะแสดงให้ผู้ใช้ทราบว่าขณะนี้ค่าที่ตั้งไว้เป็นเท่าไร โดยจะบอกค่าแรงดัน (V) ค่ากระแส (I) และค่าความต่างเฟสระหว่างสัญญาณที่ต้องการ (PS) ส่วนค่าทางด้านขวาบรรทัดเดียวกับค่าแรงดันจะระบุความถี่ของสัญญาณที่จะสร้าง ซึ่งผู้ใช้สามารถใส่ค่าต่าง ๆ เหล่านี้ได้ทางปุ่มกดรับค่า รายละเอียดจะบรรยายในหัวข้อ 3.5

## 3.4 ขั้นตอนวิธีการสร้างสัญญาณต้นแบบ

ขั้นตอนวิธีที่ใช้ในการสร้างสัญญาณต้นแบบในวิทยานิพนธ์นี้ คือ ขั้นตอนวิธีการค้นตาราง (ดังได้กล่าวคร่าว ๆ ไว้ในหัวข้อ 2.4.2) โดยค่าข้อมูลที่จะส่งถูกเก็บลงในตารางขนาด 200 ค่าซึ่งบรรจุอยู่ใน ROM ของไมโครคอนโทรลเลอร์ ข้อมูลที่เก็บเป็นขนาดของสัญญาณไซน์ 1 ลูกที่ถูกแบ่งทางเวลาออกเป็น 200 ค่า หลักการสร้างสัญญาณคือ ไมโครคอนโทรลเลอร์จะอาศัยข้อมูลใน

ตารางใน ROM ที่มีอยู่มาสร้างตารางใหม่ที่มีขนาดของสัญญาณไซน์ตามต้องการใน RAM (ขนาดตารางเป็น 200 ค่าเท่าเดิม) แผนภาพการทำงานคร่าว ๆ ด้วยขั้นตอนวิธีนี้จะเป็นไปดังแสดงในรูปที่ 3.14 ข้อมูลจะถูกทยอยส่งไปแปลงค่าโดยตัวแปลงดิจิทัลเป็นแอนะล็อกด้วยอัตราการแปลงค่าหนึ่งเพื่อให้เกิดความถี่ที่ต้องการ อัตราการแปลงค่าดังกล่าวถูกกำหนดโดยช่วงเวลาการแปลงค่า (Conversion period;$t_c$) โดยจะใช้หลักการของการขัดจังหวะ (Interrupt) ของตัวจับเวลาหมายเลข 0 (Timer0) ซึ่งมีขนาด 16 บิต ภายในไมโครคอนโทรลเลอร์ ตัวจับเวลาจะนับเวลาเท่ากับ $t_c$ จากนั้นจะส่งสัญญาณให้เกิดการขัดจังหวะเพื่อสั่งแปลงค่า และตัวจับเวลาจะเริ่มนับเวลา $t_c$ ใหม่สำหรับการแปลงค่าถัดไป โดยเมื่อครบ 200 ค่าแล้วจะวนไปเริ่มส่งที่ค่าแรกในตารางออกไปใหม่และเรียงไปเรื่อย ๆ




*รูปที่ 3.14 กลวิธีการสร้างสัญญาณต้นแบบ*

3.4.1 <u>กลวิธีการปรับขนาดของสัญญาณ</u>

เมื่อผู้ใช้ตั้งค่าขนาดสัญญาณต้นแบบให้กับแหล่งกำเนิดสัญญาณแล้ว ไมโครคอนโทรลเลอร์จะนำค่าขนาดสัญญาณออกที่ต้องการมาคำนวณตารางสัญญาณต้นแบบโดยใช้ข้อมูลจากจากตารางสัญญาณต้นแบบตัวอย่างใน ROM ซึ่งมีค่ายอดของสัญญาณ (Peak value) เป็น 32767 ซึ่งก็คือค่าเต็มพิกัดของการแปลงค่าขนาด 16 บิต ( ตัดไป 1 บิตสำหรับระบุเครื่องหมายของสัญญาณ) ดังแสดงในสมการที่ 3.10

$$table[i] = 32767 \cdot Sin(i \cdot \frac{2\pi}{200}) \quad ; i = 0,...,199 \quad \text{.....สมการที่ 3.10}$$

โดยจากสมการ i คือ ดรรชนีการชี้ค่าในตาราง และ table[i] หมายถึง ค่าในตารางตัวที่ i ส่วนไมโครคอนโทรลเลอร์จะคำนวณค่าข้อมูลสำหรับสร้างตารางใหม่ตามสมการที่ 3.11

$$new_table[i] = table[i] \cdot V_{in} \cdot \frac{1}{k} \quad ; i = 0,...,199 \; ; k = constant \quad \text{.....สมการที่ 3.11}$$

จากสมการ new_table[i] คือ ค่าในตารางใหม่ที่สร้างได้ตัวที่ i $V_{in}$ คือ ขนาดของสัญญาณเข้าที่ผู้ใช้ป้อนค่าเข้ามาซึ่งสำหรับช่องสัญญาณแรงดันจะมีค่าอยู่ในช่วง 180 – 250 $V_{rms}$ และ

สำหรับช่องสัญญาณกระแสจะเป็น 0-50 $A_{rms}$ ส่วนค่า k นั้นเป็นค่าคงที่ที่นำมาคูณเพื่อปรับขนาดของสัญญาณที่จะสร้างให้เหมาะสม ซึ่งตามทฤษฎีแล้วค่า k จะเท่ากับขนาดของสัญญาณไซน์สูงสุดที่ตัวแปลงดิจิทัลเป็นแอนะล็อกสร้างได้ (มีค่าเป็น 3.535 $V_{rms}$) คูณกับขนาดอัตราขยายของวงจรขยายสัญญาณของช่องสัญญาณนั้น ๆ (อัตราขยายของวงจรขยายแรงดันแรงดันมีค่าประมาณ 80 เท่า ส่วนอัตราขยายของวงจรแปลงแรงดันเป็นกระแสมีค่าประมาณ 15 $A_{rms}/V_{rms}$)

การสร้างตารางสัญญาณต้นแบบใหม่ใน RAM จะทำสำหรับสัญญาณต้นแบบของช่องสัญญาณแรงดัน 1 ตาราง และสัญญาณต้นแบบของช่องสัญญาณกระแส 1 ตาราง ใช้หลักการคำนวณเหมือนกันดังกล่าวข้างต้น เพียงแต่ค่า k ของแต่ละตารางจะแตกต่างกันออกไป โดยการเลือกใช้ข้อมูลขนาด 16 บิตนั้นจะทำให้วงจรที่ออกแบบสามารถสร้างแรงดันด้านออกได้ที่ค่าความละเอียดในการปรับค่าต่ำสุดเป็น 8.63 $mV_{rms}$ และของสัญญาณกระแสเป็น 1.62 $mA_{rms}$

3.4.2 <u>กลวิธีการปรับความถี่ของสัญญาณ</u>

เนื่องจากคาบของสัญญาณไซน์ต้นแบบที่สร้างได้จะมีค่าเป็น 200 คูณกับช่วงเวลาการแปลงค่า ($t_c$) ดังนั้นความถี่ของสัญญาณที่ได้ซึ่งมีค่าเท่ากับส่วนกลับของคาบ จึงถูกกำหนดโดยการปรับช่วงเวลาการแปลงค่าเช่นกัน โดยในวิทยานิพนธ์นี้ใช้ตัวจับเวลาในการนับช่วงเวลาการแปลงค่า ดังนั้นการปรับความถี่ของสัญญาณต้นแบบจึงทำโดยเลือกค่าที่จะให้ตัวจับเวลานับให้เหมาะสม อัตราการนับของตัวจับเวลามีค่าเท่ากับเวลาต่อรอบคำสั่งเครื่อง (Timer per instruction cycle) ของไมโครคอนโทรลเลอร์ซึ่งมีค่าเป็น 0.1 μs ต่อการนับ 1 ครั้ง ทำให้ค่าที่จะใส่ให้ตัวจับเวลาดังกล่าวนับ (Timer value) สามารถคำนวณได้จากสมการดังนี้

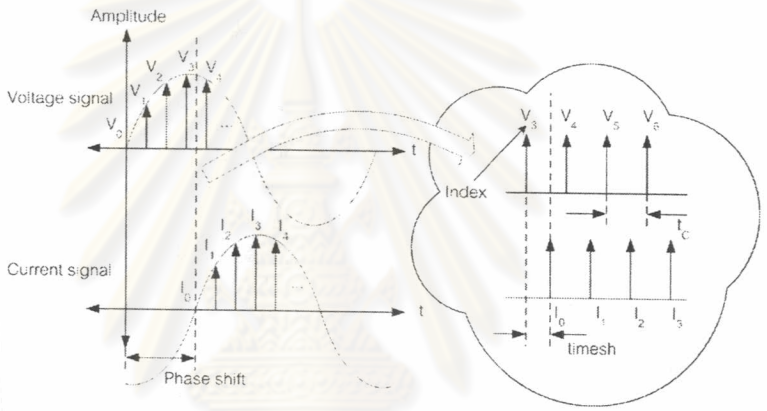
$$\text{Timer value} = \frac{1}{200} \cdot \frac{1}{f} \cdot \frac{1}{t_{IC}} \quad \text{.....สมการที่ 3.12}$$

จากสมการ f คือ ความถี่ที่ต้องการสร้าง และ $t_{IC}$ คือ เวลาต่อรอบคำสั่งเครื่องของไมโครคอนโทรลเลอร์ โดยสำหรับเวลาต่อรอบคำสั่งเครื่องเป็น 0.1 μs นั้น ค่าความละเอียดต่ำสุดในการปรับค่าความถี่จะสามารถคำนวณได้โดยปรับค่าที่จะใส่ในตัวจับเวลาไป 1 ค่า โดยจะคิดในกรณีที่สร้างความถี่ที่มีคาบต่ำสุดนั่นคือ 55 Hz โดยจากสมการที่ 3.12 จะพบว่าที่ค่า Timer value เป็น 909 จะสามารถคำนวณความถี่ f ได้เป็น 55.005 Hz และที่ค่า Timer value เป็น 910 จะคำนวณค่าถี่ f ได้ 54.945 Hz ซึ่งแสดงว่าสามารถปรับค่าความถี่ได้ต่ำสุดประมาณ 0.05 Hz นั่นเอง

3.4.3 <u>กลวิธีการปรับความต่างเฟสของสัญญาณ</u>

การปรับความต่างเฟสของสัญญาณทำได้โดยสร้างการเหลื่อมเวลาระหว่างสัญญาณรูปไซน์ของแรงดันและกระแสต้นแบบที่สร้างขึ้น แต่เนื่องจากการสร้างสัญญาณจะทำโดยส่งค่าข้อมูล

ของสัญญาณทั้งสองออกไปตามอัตราการแปลงค่าที่ตั้งไว้ ดังนั้นการปรับความต่างเฟสก็จะอาศัยการเลือกค่าข้อมูลในตารางที่จะส่งออกไปให้เหมาะสม ร่วมกับการเลื่อนเวลาของการแปลงค่าสำหรับสัญญาณแรงดันและกระแสให้ต่างกันทำให้ได้สัญญาณออกที่มีการเหลื่อมเวลากันตามต้องการ ซึ่งการเลือกค่าของข้อมูลที่จะถูกส่งไปแปลงค่านั้นหมายถึงการเลือกดรรชนีที่ชี้ข้อมูลในตารางให้ถูกต้องนั่นเอง หลักการดังกล่าวสามารถนำเสนอในรูปแบบของแผนภาพได้ดังรูปที่ 3.15 โดยจากรูป Phase shift คือเวลาที่เหลื่อมระหว่างสองสัญญาณที่สร้าง index คือ ดรรชนีชี้ค่าในตารางข้อมูลของสัญญาณแรงดันที่ต้องเลือกค่าเริ่มต้นให้เหมาะสมเพื่อให้ได้ค่าเวลาเหลื่อมที่ต้องการ ส่วนค่าดรรชนีของตารางกระแสจะให้เริ่มต้นที่ค่าศูนย์ timesh คือ เวลาย่อยที่ต้องการให้เหลื่อมเพิ่มจากการปรับ index สำหรับการสร้าง timesh นั้นจะใช้การขัดจังหวะของตัวจับเวลาหมายเลข 1 (Timer1) ซึ่งมีขนาด 16 บิต




*รูปที่ 3.15 กลวิธีการปรับความต่างเฟสระหว่างสัญญาณต้นแบบ*

เมื่อผู้ใช้ป้อนค่าความต่างเฟสที่ต้องการแล้ว ไมโครคอนโทรลเลอร์จะนำค่าที่ได้ไปคำนวณเพื่อหาค่า index เริ่มต้น ดังนี้

$$\text{Index} = \text{floor}\left(\frac{\text{Phase}}{360} \cdot 200\right) \quad \text{.....สมการที่ 3.13}$$

โดยที่ floor(.) หมายถึง การปัดเศษทิ้ง เอาแต่จำนวนเต็ม และ Phase คือ ค่าความต่างเฟสที่ผู้ใช้ตั้งค่าซึ่งอยู่ในช่วง 0-360 องศา ส่วนค่าเวลาเหลื่อมย่อยสร้างได้จากการหาค่าเพื่อใส่ให้ตัวจับเวลาหมายเลข 1 นับดังนี้

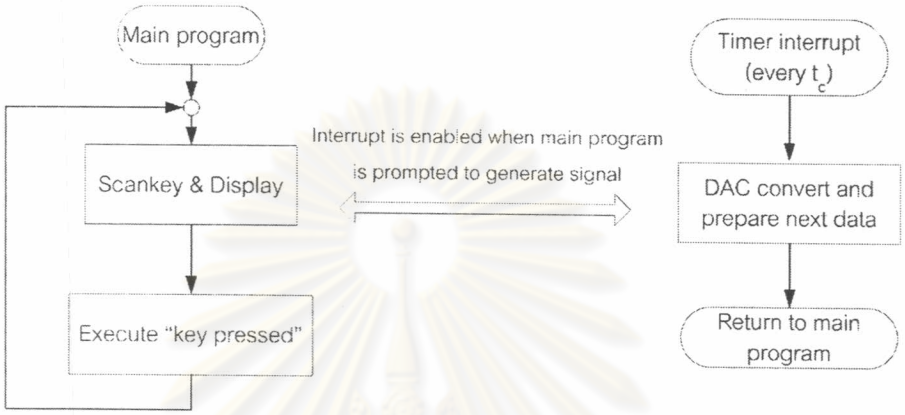
$$\text{timesh} = \text{frac}\left(\frac{\text{Phase}}{360} \cdot 200\right) * \frac{t_C}{t_{IC}} \quad \text{.....สมการที่ 3.14}$$

เมื่อ frac(.) คือ การเอาค่าแต่ส่วนทศนิยม $t_C$ คือ ช่วงเวลาการแปลงค่าที่ต้องการ (ตามรูปที่ 3.15) และ $t_{IC}$ คือ เวลาหนึ่งรอบคำสั่งเครื่องของไมโครคอนโทรลเลอร์ ดังนั้นเวลาเหลื่อมทั้งหมดระหว่างสัญญาณต้นแบบแรงดันและกระแสที่สร้างขึ้น ($T_{\text{Time shift}}$) ก็คือ

$$T_{Time\ shift} = (Index \cdot t_C) + (timesh \cdot t_{IC}) \quad \text{.....สมการที่ 3.15}$$

## 3.5 โปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

หน้าที่ของไมโครคอนโทรลเลอร์คือ ควบคุมการสร้างสัญญาณต้นแบบ และติดต่อสื่อสารกับผู้ใช้ผ่านทางปุ่มกดรับค่าและจอแสดงผล สำหรับขั้นตอนการทำงานโดยคร่าว ๆ เป็นดังแผนผังของโปรแกรมในรูปที่ 3.16

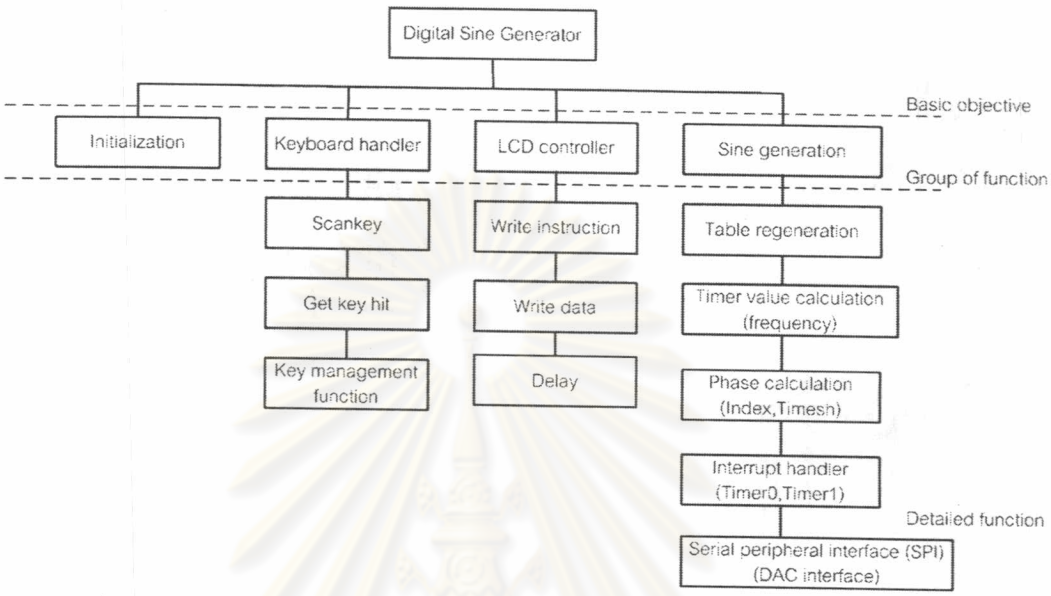



*รูปที่ 3.16 แผนผังการทำงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์*

จากรูป การทำงานจะแบ่งเป็นสองส่วนหลัก ๆ คือ โปรแกรมหลัก และโปรแกรมขัดจังหวะ โปรแกรมหลักจะทำหน้าที่อ่านค่าจากปุ่มกดรับค่าเป็นระยะ ๆ เมื่อมีการกดปุ่ม โปรแกรมจะทำการตอบสนองโดยทำงานตามที่ได้ถูกสั่ง โดยอาจจะเป็นเพียงการแสดงผลบนหน้าจอ ไปจนถึงเมื่อมีการกดปุ่มสั่งให้สร้างสัญญาณ โปรแกรมหลักก็จะทำการคำนวณค่าที่เกี่ยวข้องและเริ่มกระบวนการสร้างสัญญาณต้นแบบโดยการสั่งงานตัวแปลงดิจิทัลเป็นแอนะล็อก สำหรับในขั้นตอนนี้ จะอาศัยกลไกการขัดจังหวะของตัวจับเวลาเพื่อสร้างฐานเวลาสำหรับการแปลงค่า ($t_c$) โดยโปรแกรมหลักจะทำการตั้งค่าตัวบ่งชี้ (Flag) เพื่ออนุญาตให้มีการขัดจังหวะ จากนั้นก็จะเป็นหน้าที่ของโปรแกรมขัดจังหวะในการดำเนินการสั่งการตัวแปลงดิจิทัลเป็นแอนะล็อก การขัดจังหวะจะเกิดทุก ๆ ครั้งที่ตัวจับเวลานับครบช่วงการแปลงค่า ในขณะนั้นโปรแกรมหลักก็จะทำการอ่านค่าจากปุ่มกดไปเรื่อย ๆ เพื่อตรวจสอบการกดปุ่มเพื่อยกเลิกการสร้างสัญญาณ (โปรแกรมจะไม่ประมวลผลค่าปุ่มอื่น ๆ ทั้งสิ้นในขณะที่มีการสร้างสัญญาณ) ซึ่งเมื่อมีการกดยกเลิก โปรแกรมหลักจะลดค่าตัวบ่งชี้เพื่อตัดการทำงานของส่วนขัดจังหวะเพื่อหยุดสร้างสัญญาณ แล้วกลับมารอตรวจสอบการกดปุ่มต่อไป

3.5.1 <u>ส่วนประกอบของโปรแกรม</u>

สำหรับในรายละเอียดของส่วนโปรแกรมนั้นจะขอกล่าวถึงส่วนประกอบของโปรแกรมก่อน เพื่อให้เห็นภาพการทำงานโดยคร่าว โดยจะแสดงเป็นแผนผังการแจกแจงโปรแกรมย่อยเป็นหมวดหมู่ดังรูปที่ 3.17




*รูปที่ 3.17 แผนผังแจกแจงโปรแกรมย่อยของโปรแกรมควบคุมไมโครคอนโทรลเลอร์*

จากรูปจะเห็นว่าโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ประกอบไปด้วยส่วนหลัก ๆ 4 ส่วนด้วยกันคือ

1. ส่วนตั้งค่าเริ่มต้นของตัวแปร (Initialization)

โปรแกรมส่วนนี้จะทำหน้าที่ตั้งค่าเริ่มต้นสำหรับตัวแปรต่าง ๆ ที่จำเป็นต้องใช้ แล้วเก็บลงใน RAM ตั้งค่าทิศทางของช่องทางเข้าออกของไมโครคอนโทรลเลอร์ ตั้งค่าวงจรสื่อสารอนุกรม เพื่อสั่งงานตัวแปลงดิจิทัลเป็นแอนะล็อก ส่งคำสั่งให้จอแสดงผลเริ่มทำงานและแสดงผลหน้าจอเริ่มต้น รวมไปถึงการตั้งค่าการขัดจังหวะที่จะใช้ในโปรแกรม

2. ส่วนควบคุมการรับค่าจากปุ่มกดรับค่า (Keyboard handler)

โปรแกรมส่วนนี้ประกอบไปด้วยโปรแกรมย่อย Scankey ทำหน้าที่ตรวจสอบว่ามีการกดปุ่มหรือไม่ ถ้ามีก็จะตรวจสอบหาปุ่มที่ถูกกดและส่งเป็นรหัสของปุ่มที่ถูกกดนั้น ๆ กลับไปให้โปรแกรมหลักประมวลผลต่อไป โปรแกรมย่อย Get key hit มีหน้าที่ตรวจสอบหาว่าปุ่มใดที่ถูกกด ส่วนโปรแกรมย่อย Key management function ทำหน้าที่แจกแจงการทำงานของไมโครคอนโทรลเลอร์เพื่อตอบสนองปุ่มที่ถูกกด

3. ส่วนควบคุมการทำงานของจอแสดงผล

ส่วนนี้ทำหน้าที่ควบคุมสั่งงานจอแสดงผลแบบผลึกเหลว เนื่องจากการสั่งงานจอแสดงผลชนิดนี้จะมีรูปแบบคำสั่งเฉพาะ โดยโปรแกรมย่อยของส่วนนี้ประกอบไปด้วยโปรแกรมย่อย Write instruction สำหรับส่งค่าคำสั่งเพื่อให้จอแสดงผลปฏิบัติตาม เช่น การล้างหน้าจอ การเลื่อนตัวชี้ตำแหน่งตัวอักษรไปที่ต่าง ๆ บนจอ เป็นต้น โปรแกรมย่อย Write data สำหรับส่งค่าตัวอักษรที่ต้องการให้แสดงบนหน้าจอแสดงผลนั้น ๆ โปรแกรมย่อย Delay ทำหน้าที่ประวิงเวลาเพื่อรอให้จอแสดงผลประมวลคำสั่งก่อนจะส่งคำสั่งถัดไป

4. ส่วนควบคุมการสร้างสัญญาณต้นแบบรูปไซน์

ส่วนนี้ประกอบไปด้วย โปรแกรมย่อยที่ทำหน้าที่เกี่ยวกับการสร้างสัญญาณต้นแบบทั้งหมด โปรแกรมย่อย Table regeneration ทำหน้าที่สร้างตารางข้อมูลสัญญาณไซน์ที่จะส่งไปแปลงค่าขึ้นใหม่ใน RAM โดยใช้ข้อมูลจากขนาดสัญญาณออกที่ผู้ใช้ต้องการ ร่วมกับตารางสัญญาณไซน์ต้นแบบที่มีเก็บไว้ใน ROM โปรแกรมย่อย Timer value calculation ทำหน้าที่คำนวณค่าเวลาที่จะให้ตัวจับเวลานับเพื่อสร้างฐานเวลาการแปลงค่าให้การสร้างสัญญาณต้นแบบ หรืออีกนัยหนึ่งก็คือเป็นการกำหนดความถี่ของสัญญาณออกนั่นเอง โปรแกรมย่อย Phase calculation ทำหน้าที่คำนวณดรรชนีชี้ค่าในตารางสัญญาณต้นแบบของแรงดันและคำนวณหาเวลาที่จะต้องเหลื่อมเพิ่มจากการเหลื่อมดรรชนีด้วยเพื่อให้สัญญาณแรงดันต้นแบบของสัญญาณแรงดันและกระแสมีความต่างเฟสตามต้องการ โปรแกรมย่อย Interrupt handler ทำหน้าที่ควบคุมตั้งค่าการขัดจังหวะเพื่อสร้างฐานเวลาการแปลงค่า และส่วนสุดท้ายโปรแกรมย่อย SPI ทำหน้าที่สั่งงานวงจรสื่อสารอนุกรมภายในให้ส่งค่าไปสั่งงานตัวแปลงดิจิทัลเป็นแอนะล็อกได้ถูกต้อง

### 3.5.2 แผนผังการทำงานของโปรแกรม

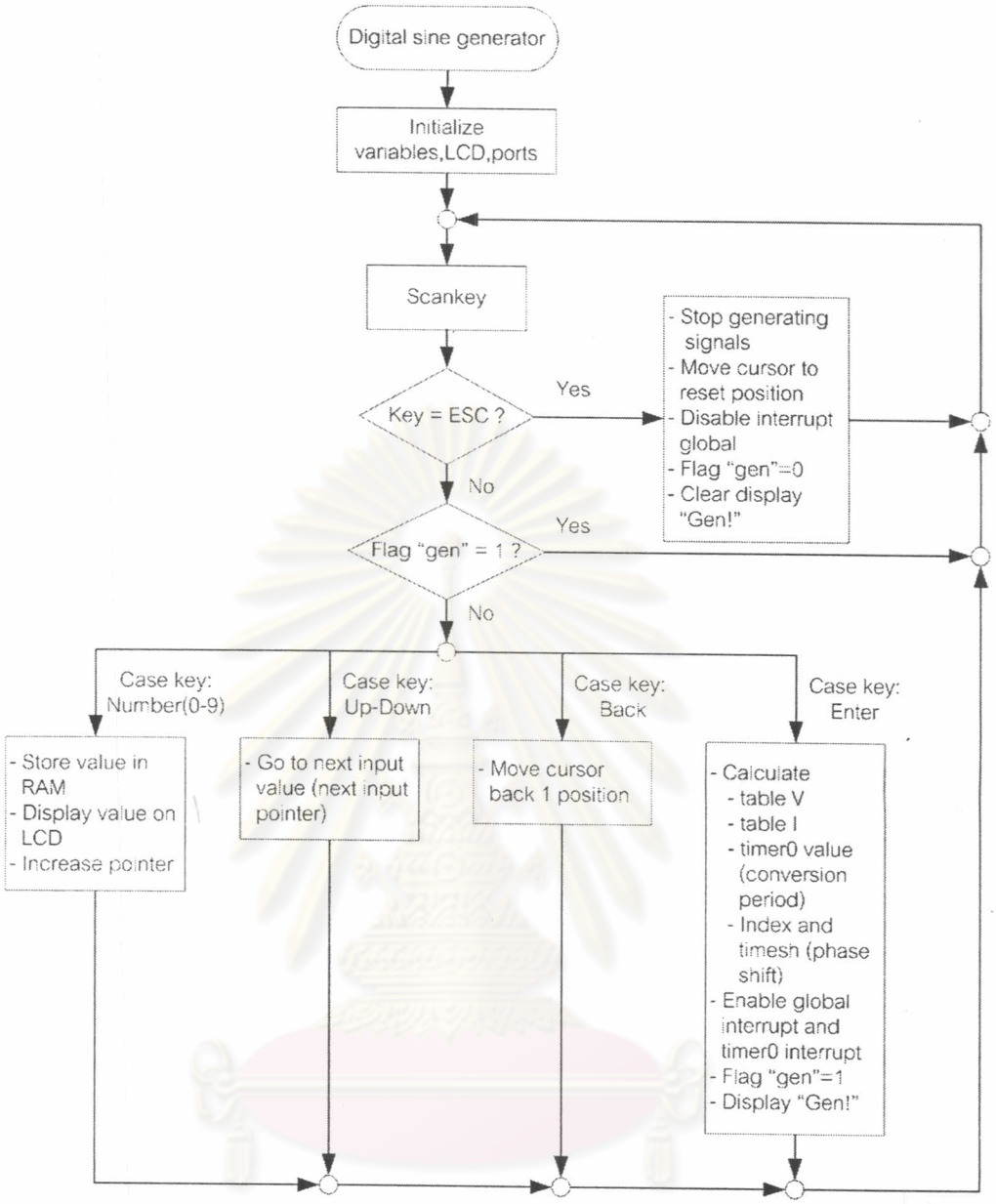
โปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์มีรายละเอียดดังผังการทำงานในรูปที่ 3.18 และ 3.19 โดยในรูปที่ 3.18 เป็นการทำงานในส่วนของโปรแกรมหลัก และในรูปที่ 3.19 เป็นส่วนโปรแกรมขัดจังหวะ โดยจะใช้ตัวจับเวลาหมายเลข 0 เพื่อนับเวลาสร้างฐานเวลาการแปลงค่า ส่วนตัวจับเวลาหมายเลข 1 จะถูกใช้เพื่อเหลื่อมเวลาระหว่างการแปลงค่าเพื่อสร้างความต่างเฟสของสัญญาณทั้งสอง และเพื่อให้เกิดความเข้าใจอย่างถ่องแท้ จะขออธิบายการทำงานจริง ๆ ว่าเป็นอย่างไรและปุ่มต่าง ๆ หมายถึงอะไรไว้ที่นี้ก่อน รวมไปถึงตัวแปรบางตัวที่ใช้ในโปรแกรมและแสดงอยู่ในแผนผังการทำงานด้วยดังนี้

เมื่อผู้ใช้เริ่มต้นเปิดเครื่อง เครื่องจะแสดงหน้าจอเริ่มต้นซึ่งมีลักษณะคล้ายกับรูปที่ 3.13 เพียงแต่ค่าเริ่มต้นที่ให้มาจะเป็นศูนย์หมด และจะมีตัวชี้ตำแหน่งตัวอักษรกระพริบอยู่ในตำแหน่งเริ่มต้น (หลักสูงสุดของพารามิเตอร์ขนาดแรงดัน) ผู้ใช้สามารถกดปุ่มต่าง ๆ เพื่อใส่ค่าที่ต้องการโดยจากรูปปุ่มกดรับค่า (รูปที่ 3.12) ผู้ใช้สามารถกดปุ่มตัวเลขต่าง ๆ เพื่อตั้งค่าสัญญาณที่ต้องการ

ให้สร้างได้ และสามารถเลือกตั้งค่าแรงดัน, ค่ากระแส, ความถี่ของสัญญาณที่สร้าง รวมไปถึงความต่างเฟสระหว่างสัญญาณแรงดันและกระแสที่ต้องการได้โดยใช้ปุ่มขึ้นและลง เช่น เมื่อตัวชี้อยู่ที่ตำแหน่งของการใส่ค่าสัญญาณแรงดันแล้วกดลง (กรุณาดูรูปจอแสดงผลในรูปที่ 3.13 ประกอบ) โปรแกรมจะเปลี่ยนให้ไปใส่ค่าความถี่โดยเลื่อนตัวชี้บนหน้าจอไปที่หลักสูงสุดของความถี่และรอรับค่า เมื่อกดลงอีกครั้งก็จะเลื่อนตัวชี้ไปที่หลักสูงสุดของการตั้งค่ากระแส และเมื่อกดขึ้นก็จะไปรอรับค่าความถี่ที่หลักสูงสุดอีกครั้ง เป็นต้น ในกรณีที่ใส่ค่าผิด ผู้ใช้สามารถกดปุ่ม Back เพื่อย้อนกลับไปแก้ไขค่าที่ใส่ไว้ในหลักที่แล้วได้ และเมื่อตั้งค่าต่าง ๆ เสร็จ จะต้องกดปุ่ม Enter เพื่อสั่งให้วงจรเริ่มสร้างสัญญาณตามที่ได้ตั้งค่าและแสดงผลบนหน้าจอ โดยจะมีตัวบอกสถานะว่าไมโครคอนโทรลเลอร์นั้นกำลังสร้างสัญญาณอยู่ คือตัวอักษรคำว่า “Gen!” ปรากฏอยู่ที่มุมล่างขวาของหน้าจอ (จะขึ้นเมื่อมีการสร้างสัญญาณอยู่เท่านั้น) ในตอนนี้ผู้ใช้ไม่สามารถกดปุ่มรับค่าอื่นใดได้นอกจากปุ่ม ESC การกดปุ่มอื่น ๆ นั้นไมโครคอนโทรลเลอร์จะไม่กระทำการใด ๆ แต่เมื่อกดปุ่ม ESC วงจรจะหยุดสร้างสัญญาณและข้อความ “Gen!” ก็จะหายไปด้วย ซึ่งตอนนี้ผู้ใช้ก็สามารถตั้งค่าต่าง ๆ ที่ต้องการบนหน้าจอได้ใหม่อีกครั้ง

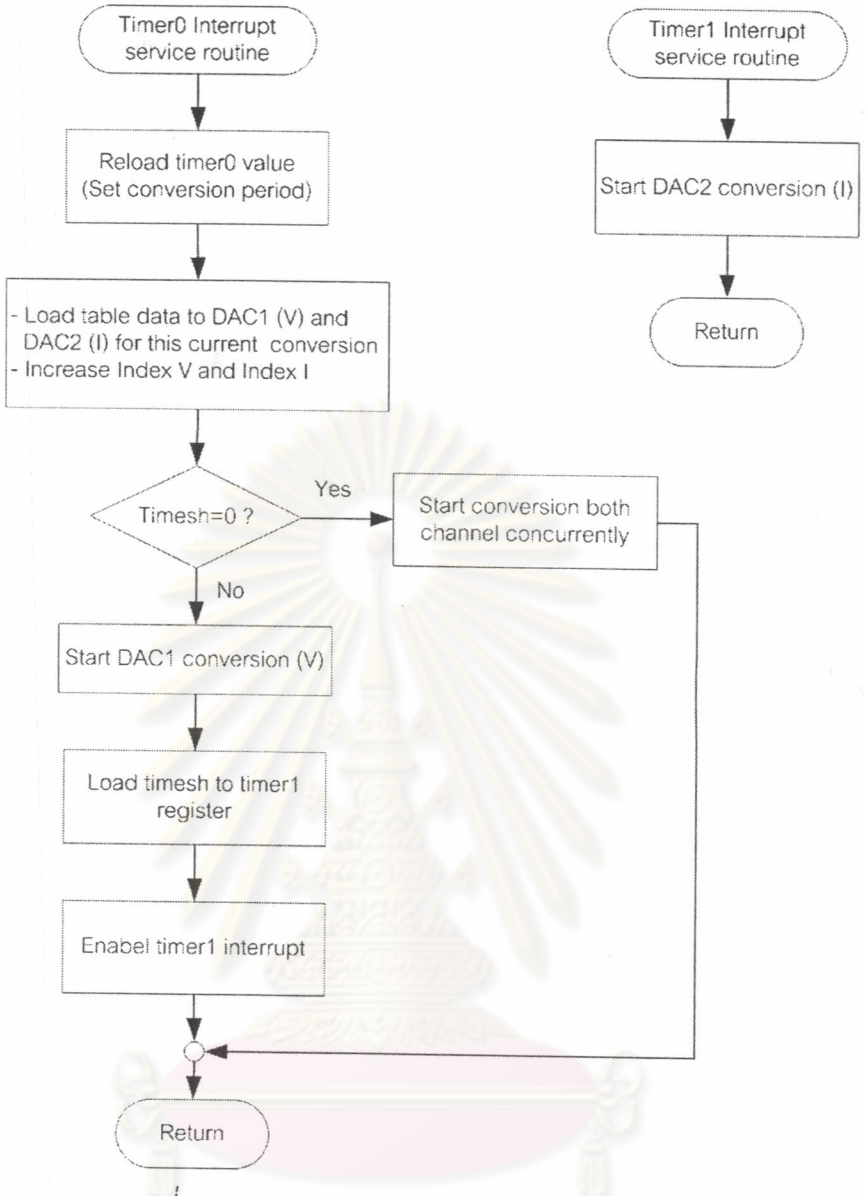
จากรูปที่ 3.18 ตัวบ่งชี้ gen นั้นหมายถึงตัวแปรที่ทำหน้าที่บอกสถานะของวงจรว่ากำลังดำเนินการสร้างสัญญาณต้นแบบอยู่หรือไม่ ตัวบ่งชี้ gen มีค่าเป็น 1 หมายถึงการสร้างสัญญาณกำลังดำเนินอยู่ โปรแกรมจะตรวจสอบตัวบ่งชี้นี้ก่อนทุกครั้งโดยถ้าตัวบ่งชี้ gen เป็น 1 โปรแกรมจะไม่ทำงานตามการกดปุ่มใด ๆ ยกเว้นปุ่ม ESC การที่ทำดังนี้ก็เพื่อไม่ให้ผู้ใช้ทำการตั้งค่าอื่นใดระหว่างวงจรกำลังสร้างสัญญาณอยู่นั่นเอง ส่วนตัวแปร Timesh ในรูปที่ 3.19 คือค่าที่ใส่ให้ตัวจับเวลานับเพื่อสร้างการเหลื่อมเวลาเพิ่มจากการเหลื่อมค่าดรรชนีชี้ค่าในตาราง ส่วนตัวแปลงดิจิทัลเป็นแอนะล็อกจะถูกตั้งชื่อเพื่อให้กระชับโดยของช่องสัญญาณแรงดันเป็น DAC1 และของช่องสัญญาณกระแสเป็น DAC2 ตามลำดับ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย




*รูปที่ 3.18 แผนผังการทำงานของโปรแกรมหลักสำหรับควบคุมไมโครคอนโทรลเลอร์*

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย




*รูปที่ 3.19 แผนผังการทำงานของโปรแกรมขัดจังหวะ*

จากรูปที่ 3.18 การทำงานของแหล่งกำเนิดสัญญาณจะเริ่มจากการตั้งค่าเริ่มต้นให้กับโปรแกรมและแสดงผลหน้าจอเริ่มต้นเพื่อรอผู้ใช้ใส่ค่าที่ต้องการ จากนั้นโปรแกรมจะเริ่มวนตรวจสอบการกดปุ่ม โดยเมื่อพบว่ามีการกดปุ่มเกิดขึ้น จะตรวจสอบหาปุ่มที่ถูกกดและดำเนินการตอบสนองการกดปุ่มนั้น ๆ โดยมีรายละเอียดดังนี้

- การกดปุ่ม Esc เป็นการสั่งหยุดสร้างสัญญาณต้นแบบ โปรแกรมจะลดค่าตัวบ่งชี้สำหรับการขัดจังหวะทำให้หยุดการสร้างสัญญาณ แล้วจะเลื่อนตัวชี้ตำแหน่งตัวอักษรบนหน้าจอให้ไปอยู่ที่ตำแหน่งแรกสุดของหน้าจอ (หลักสูงสุดสำหรับตั้งค่าขนาดสัญญาณแรงดัน) แล้วจึงลดค่าตัวบ่งชี้ gen ให้เป็น 0 เพื่อให้โปรแกรมทำงานตามการกดปุ่มทุกปุ่มได้ตามปกติ

- การกดปุ่มตัวเลขหรือจุดทศนิยม เป็นการใส่ค่าข้อมูลตัวที่ถูกชี้โดยตัวชี้ตำแหน่งตัวอักษร โปรแกรมจะตอบสนองโดยการเก็บค่าลงใน RAM แล้วแสดงผลข้อมูลนั้นบนหน้าจอในหลักที่ถูกชี้ และเลื่อนตัวชี้ไปที่หลักถัดไปของพารามิเตอร์เดิมหรือหลักบนสุดของพารามิเตอร์ใหม่ในกรณีที่ข้อมูลที่ใส่ก่อนหน้านี้เป็นข้อมูลตัวสุดท้ายของพารามิเตอร์นั้น ๆ

- การกดปุ่มขึ้นหรือปุ่มลง เป็นการเปลี่ยนตำแหน่งการใส่ค่าไปเป็นพารามิเตอร์ตัวถัดไป โปรแกรมจะเลื่อนตัวชี้ตำแหน่งตัวอักษรไปที่ตำแหน่งสูงสุดของหลักถัดไป โดยเมื่อกดลง ลำดับการเปลี่ยนพารามิเตอร์จะเป็นดังนี้ ค่าขนาดแรงดัน ค่าความถี่ ค่าขนาดของกระแส และค่าความต่างเฟสระหว่างสัญญาณ ส่วนเมื่อกดขึ้นลำดับจะย้อนกลับกับค่าดังกล่าว

- การกดปุ่ม back เป็นการย้อนกลับไปใส่ค่าในตำแหน่งที่แล้ว โปรแกรมจะเลื่อนตัวชี้ตำแหน่งตัวอักษรไปที่ตำแหน่งก่อนหน้า โดยถ้าตำแหน่งตัวชี้ปัจจุบันอยู่ที่ตำแหน่งหลักสูงสุดของพารามิเตอร์แล้ว โปรแกรมจะเลื่อนตัวชี้ไปที่หลักต่ำสุดของพารามิเตอร์ที่อยู่ก่อนหน้าโดยอัตโนมัติ

- การกดปุ่ม Enter เป็นการสั่งให้สร้างสัญญาณต้นแบบ โปรแกรมจะนำค่าที่ผู้ใช้ตั้งค่าไว้มาคำนวณเป็นตารางขนาดของสัญญาณแรงดันและกระแส, ค่าที่จะใส่ให้ตัวจับเวลาหมายเลข 0 เพื่อสร้างฐานเวลาของการแปลงค่า และค่าดรรชนีสำหรับตารางแรงดันและค่าที่จะใส่ให้ตัวจับเวลาหมายเลข 1 เพื่อสร้างความต่างเฟสระหว่างสัญญาณ จากนั้นจะตั้งค่าตัวบ่งชี้เพื่ออนุญาตให้มีการขัดจังหวะเกิดขึ้นได้ แล้วจึงตั้งค่าตัวบ่งชี้ gen ให้เป็น 1 เพื่อระบุว่าโปรแกรมกำลังสร้างสัญญาณอยู่ รวมทั้งแสดงคำว่า “Gen!” บนจอแสดงผลด้วย

เมื่อตัวจับเวลาหมายเลข 0 นับครบช่วงเวลาการแปลงค่าที่ตั้งไว้ก็จะส่งสัญญาณให้โปรแกรมขัดจังหวะของตัวมันทำงาน (รูปที่ 3.19) การทำงานจะเริ่มที่ตั้งค่าให้ตัวจับเวลาหมายเลข 0 ใหม่และสั่งให้นับเพื่อการสร้างสัญญาณในช่วงเวลาถัดไป จากนั้นจะเริ่มขั้นตอนการสั่งงานตัวแปลงดิจิทัลเป็นแอนะล็อกให้สร้างสัญญาณ ซึ่งทำโดยการตรวจสอบค่าตัวแปร Timesh ถ้าเป็นศูนย์ซึ่งหมายถึงไม่ต้องมีการเหลื่อมเวลาของดรรชนีชี้ค่าข้อมูลของทั้งสองสัญญาณ โปรแกรมก็จะสั่งให้ช่องสัญญาณทั้งสองแปลงค่าพร้อมกัน แต่ถ้า Timesh มีค่ามากกว่าศูนย์ โปรแกรมก็จะสั่งแปลงค่าในส่วนของสัญญาณแรงดันก่อน จากนั้นก็ตั้งค่าเวลาให้กับตัวจับเวลาหมายเลข 1 และอนุญาตให้มีการขัดจังหวะของตัวจับเวลาหมายเลข 1 ได้แล้วออกจากโปรแกรมขัดจังหวะไป และเมื่อตัวจับเวลาหมายเลข 1 นับครบค่าเวลาเหลื่อมก็จะส่งสัญญาณให้โปรแกรมขัดจังหวะของตนเองทำงานโดยสั่งให้ตัวแปลงดิจิทัลเป็นแอนะล็อกของช่องสัญญาณกระแสแปลงค่าแล้วออกจากโปรแกรมขัดจังหวะไปทำงานในโปรแกรมหลักอีกครั้ง

## 3.6 การปรับเทียบสัญญาณออกให้มีความแม่นยำ

เมื่อต่อวงจรและเขียนโปรแกรมตามที่บรรยายไปในหัวข้อที่ผ่านมาพบว่าสัญญาณออกยังไม่แม่นยำตามต้องการเสียทีเดียว ทั้งนี้ก็เนื่องมาจากปัจจัยต่าง ๆ ไม่ว่าจะเป็นลักษณะสมบัติของอุปกรณ์ที่ใช้ การที่อุปกรณ์เหล่านั้นมีความคลาดเคลื่อน ลักษณะการต่อวงจร หรือแม้แต่ลักษณะการเขียนโปรแกรมก็ล้วนแล้วแต่มีผลทำให้สัญญาณออกเกิดความคลาดเคลื่อนทั้งสิ้น ดังนั้นจึงต้องมีการปรับเทียบสัญญาณออกในด้านต่าง ๆ เพื่อให้สัญญาณออกมีความแม่นยำตามที่ต้องการ

### 3.6.1 การปรับเทียบขนาดของสัญญาณ

การปรับเทียบขนาดทำเพื่อให้ขนาดของสัญญาณออกมีความแม่นยำตรงตามค่าที่ตั้งไว้ โดยจะทำเมื่อต่อส่วนสร้างสัญญาณต้นแบบเข้ากับวงจรขยายสัญญาณแล้ว การปรับเทียบจะทำในโปรแกรมควบคุมการสร้างสัญญาณต้นแบบ โดยปรับค่าคงที่ k ในสมการที่ 3.11 ให้เหมาะสม ซึ่งต้องปรับทั้งของช่องสัญญาณแรงดันและช่องสัญญาณกระแส ผลปรากฏว่าค่า k ที่เหมาะสมสำหรับช่องสัญญาณแรงดันมีค่าเป็น 261.7935 และค่า k ที่เหมาะสมกับช่องสัญญาณกระแสคือ 53.1635 ซึ่งค่าที่เลือกมานี้ทำให้การตั้งค่าสัญญาณออกมีความผิดพลาดไม่เกิน 0.1 % เทียบกับค่าเต็มพิกัด (ซึ่งมีค่าเป็น 0.25 $V_{rms}$ สำหรับช่องสัญญาณแรงดันและ 0.05 $A_{rms}$ สำหรับช่องสัญญาณกระแส) ดูรายละเอียดได้จากผลการทดลองในบทที่ 5

### 3.6.2 การปรับเทียบความถี่ของสัญญาณ

การปรับเทียบความถี่ทำโดยปรับค่าที่จะใส่ลงในตัวจับเวลาหมายเลข 0 เพื่อสร้างฐานเวลาการแปลงค่าให้เหมาะสม ซึ่งเมื่อต่อวงจรและทดลองวัดค่าความถี่ของสัญญาณต้นแบบพบว่าความถี่ของสัญญาณออกที่ได้มีความคลาดเคลื่อนไปจากค่าที่ตั้งไว้เกินค่าที่ยอมรับได้ ซึ่งเมื่อลองปรับค่าที่จะใส่ให้ตัวจับเวลาเพื่อให้ได้ความถี่ตรงตามต้องการและมีความแม่นยำอยู่ภายใน ± 0.1 % ของค่าที่ตั้งตามที่เสนอไว้ในขอบเขตจะได้ผลเป็นดังตารางที่ 3.1 คอลัมน์แรกแสดงความถี่ที่ต้องการสร้าง คอลัมน์ที่สองแสดงค่าที่ต้องใส่ให้ตัวจับเวลานับเพื่อให้สร้างความถี่ได้ใกล้เคียงค่าที่ต้องการมากที่สุด คอลัมน์ที่สามแสดงความถี่ที่วัดได้เมื่อใส่ค่าในคอลัมน์ที่สองให้ตัวจับเวลานับ คอลัมน์ที่สี่แสดงค่าที่ต้องใส่ให้ตัวจับเวลานับที่ไมโครคอนโทรลเลอร์คำนวณได้จากสมการที่ 3.12 และคอลัมน์ที่ 5 แสดงผลต่างระหว่างค่าในคอลัมน์ที่สองและคอลัมน์ที่สี่ จากข้อมูลในตารางดังกล่าวจะเห็นว่าค่าที่คำนวณตามสมการที่ 3.12 นั้นไม่สามารถนำมาใช้ได้ ซึ่งน่าจะเป็นผลมาจากการที่ผลึกออสซิลเลเตอร์ (Crystal oscillator) ที่ใช้เพื่อสร้างความถี่ในการนับมีความคลาดเคลื่อนไปจากค่าที่ระบุ ร่วมกับความคลาดเคลื่อนของเวลาภายในไมโครคอนโทรเลอร์

*ตารางที่ 3.1 ตารางแสดงการวัดค่าเพื่อปรับเทียบความถี่ของสัญญาณ*

| ความถี่ที่ตั้ง(Hz) | ค่า Timer ที่ให้ความถี่พอดี | ความถี่ที่วัดได้(Hz) | ค่า Timer ที่คำนวณได้ | ผลต่าง(ค่าที่พอดี-ค่าที่คำนวณ) |
|---|---|---|---|---|
| 45.0 | 1097 | 45.00 | 1111 | -14 |
| 45.5 | 1085 | 45.48 | 1099 | -14 |
| 46.0 | 1072 | 46.01 | 1087 | -15 |
| 46.5 | 1060 | 46.51 | 1075 | -15 |
| 47.0 | 1049 | 46.98 | 1064 | -15 |
| 47.5 | 1037 | 47.50 | 1053 | -16 |
| 48.0 | 1026 | 47.99 | 1042 | -16 |
| 48.5 | 1015 | 48.49 | 1031 | -16 |
| 49.0 | 1004 | 49.00 | 1020 | -16 |
| 49.5 | 993 | 49.52 | 1010 | -17 |
| 50.0 | 983 | 50.00 | 1000 | -17 |
| 50.5 | 973 | 50.49 | 990 | -17 |
| 51.0 | 963 | 51.00 | 980 | -17 |
| 51.5 | 953 | 51.51 | 971 | -18 |
| 52.0 | 943 | 52.03 | 962 | -19 |
| 52.5 | 934 | 52.51 | 952 | -18 |
| 53.0 | 925 | 53.00 | 943 | -18 |
| 53.5 | 916 | 53.49 | 935 | -19 |
| 54.0 | 907 | 54.00 | 926 | -19 |
| 54.5 | 898 | 54.52 | 917 | -19 |
| 55.0 | 890 | 54.99 | 909 | -19 |

ดังนั้นเราจะทำการปรับเทียบโดยการอาศัยการค้นตารางแทนการคำนวณค่าที่จะต้องใส่ลงในตัวจับเวลาโดยตรง ข้อมูลในตารางก็คือค่าในคอลัมน์ที่สอง เริ่มจากค่าแรกคือค่าที่ใส่ให้ตัวจับเวลานับที่ทำให้ได้ความถี่ 45 Hz ไปจนถึงค่าสุดท้ายคือค่าที่ใส่ให้ตัวจับเวลานับที่ทำให้ได้ความถี่ 55 Hz รวมทั้งหมด 21 ค่า แต่เนื่องจากความละเอียดในการปรับที่ต้องการเป็น 0.1 Hz แต่ข้อมูลที่มีในตารางละเอียดแค่ 0.5 Hz เท่านั้น ดังนั้นจึงต้องมีการคำนวณค่าที่ขาดหายไปเพิ่มซึ่งจะอาศัยกลวิธีการประมาณค่าในช่วงแบบเชิงเส้น (Linear interpolation) เข้ามาช่วย โดยการคำนวณจะเป็นดังนี้

$$\text{Indexf} = \text{floor}\left(\frac{f_{in} - 45}{0.5}\right) \quad \text{.....สมการที่ 3.16}$$

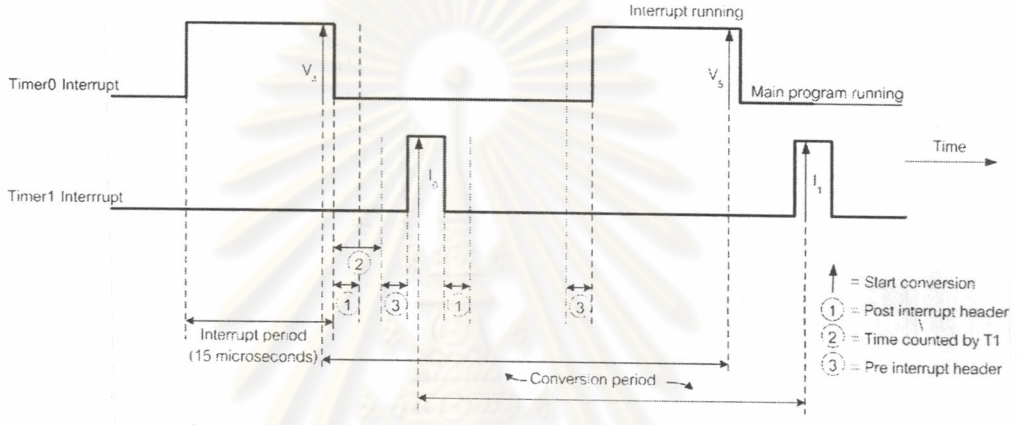
$$\text{T0value} = \text{ft}[\text{indexf}] + [f_{in} - (45 + 0.5 \cdot \text{indexf})]\left[\frac{\text{ft}[\text{indexf}+1] - \text{ft}[\text{indexf}]}{0.5}\right]$$

.....สมการที่ 3.17

จากสมการ Indexf คือ ดรรชนีชี้ค่าในตาราง $f_{in}$ คือ ความถี่ที่ต้องการสร้าง T0value คือ ค่าที่คำนวณได้เพื่อใส่ให้ตัวจับเวลาหมายเลข 0 และ ft[indexf] หมายถึงค่าในตารางตัวที่ชี้โดย indexf ซึ่งเมื่อใช้วิธีนี้จะทำให้ได้สัญญาณออกที่มีความแม่นยำของความถี่อยู่ในช่วง ±0.1 % เทียบกับค่าที่ตั้ง

### 3.6.3 การปรับเทียบความต่างเฟสของสัญญาณออก

การปรับความต่างเฟสของสัญญาณจะทำดังที่ได้กล่าวไว้ในหัวข้อ 3.4.3 ซึ่งผลที่ได้จะสามารถเขียนเป็นแผนภาพของการขัดจังหวะในกรณีที่มีการเหลื่อมเฟสได้ดังนี้

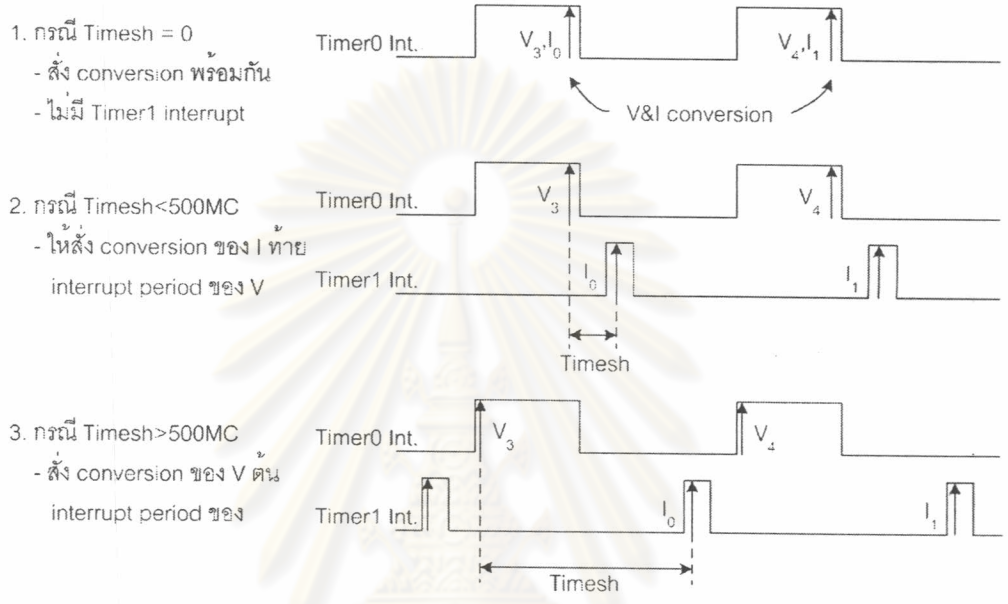



*รูปที่ 3.20 แผนผังเวลาของการขัดจังหวะเพื่อสร้างความต่างเฟส*

จากภาพแสดงช่วงเวลาการขัดจังหวะของตัวจับเวลาหมายเลข 0 และหมายเลข 1 ซึ่งจากภาพนั้นจะเห็นว่าในการขัดจังหวะแต่ละครั้ง โปรแกรมจะต้องมีการตั้งค่าและเก็บรักษาเรจิสเตอร์บางตัวทำให้เสียเวลาไปส่วนหนึ่งทั้งก่อนและหลังการขัดจังหวะ โดยเวลาที่เสียไปก่อนการขัดจังหวะจะเรียกว่า Pre interrupt header หรือเวลาหมายเลข 3 ที่แสดงในรูป ส่วนเวลาที่เสียไปหลังจากการขัดจังหวะจะเรียกว่า Post interrupt header หรือเวลาหมายเลข 1 ตามรูป ดังนั้นเวลาที่ตัวจับเวลาหมายเลข 1 นับเพื่อสร้างการเหลื่อมเวลาจะเหลือแค่ช่วงเวลาหมายเลข 2 เท่านั้น ทำให้เกิดความคลาดเคลื่อนในการปรับเฟสในโปรแกรมขึ้นเนื่องจากจะต้องรวมเวลาสูญเสียเพิ่มไปจากค่าเวลาที่จะต้องนับด้วย โดยจากการวัดค่าพบว่า Pre interrupt header มีค่าเป็น 3 μs ส่วน Post interrupt header มีค่าเป็น 4.9 μs ดังนั้นการแก้ไขทำโดยลบค่าเวลาสูญเสียก่อนการเข้าโปรแกรมขัดจังหวะออกจากค่า Timesh ที่คำนวณได้จากสมการที่ 3.14 เสียก่อนแล้วจึงนำค่าที่ได้ไปใส่ให้ตัวจับเวลา นอกจากนี้การมีช่วงเวลาดังกล่าวเพิ่มเข้ามาก็ยังมีผลทำให้เกิดข้อจำกัดทางด้านเวลาของการเหลื่อมด้วยนั่นคือทำให้ไม่สามารถเหลื่อมเวลาได้น้อยกว่าค่า ๆ หนึ่ง ซึ่งก็คือช่วงเวลาที่ต้องออกจากโปรแกรมขัดจังหวะของตัวจับเวลาหมายเลข 0 ไปเข้าโปรแกรมขัดจังหวะของตัวจับเวลาหมายเลข 1 (ช่วงเวลาหมายเลข 1 บวกกับช่วงเวลาหมายเลข 3) ซึ่งมีค่าเท่ากับ 7.9 μs

ดังนั้นถ้ามีการเหลื่อมเวลาที่ทำให้การขัดจังหวะทั้งสองต้องเกิดห่างกันน้อยกว่า 7.9 μs ก็จะไม่สามารถทำได้ แต่จากการทดลองคำนวณค่าเวลาที่จะต้องเหลื่อมทั้งหมดพบว่าค่าเวลาที่จะต้องการให้เหลื่อมต่ำสุดมีค่าเป็น 9.8 μs ก็ทำให้การสร้างยังเป็นไปได้ แต่ต้องกำหนดตำแหน่งเวลาที่จะสั่งให้แปลงค่าให้ดีด้วย

เพื่อแก้ปัญหาที่กล่าวมาทั้งหมด เราจำเป็นต้องปรับขั้นตอนวิธีการเหลื่อมเวลาในส่วนของโปรแกรมควบคุมใหม่ โดยจะแบ่งออกเป็น 3 กรณีดังแสดงในรูปที่ 3.21




*รูปที่ 3.21 การสร้างการเหลื่อมเวลาแบบใหม่เพื่อกำจัดปัญหาด้านเวลา*

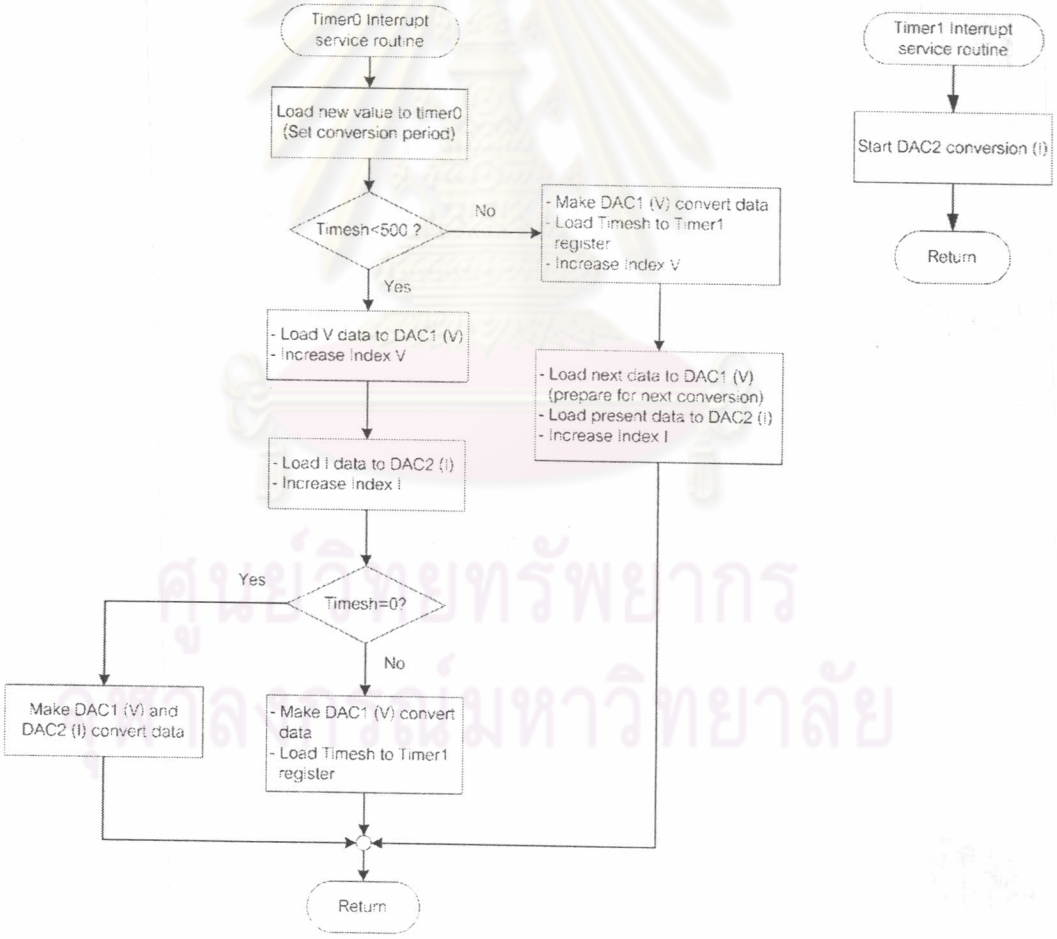
จากรูป MC หมายถึง รอบคำสั่งเครื่อง (Machine cycle) ของไมโครคอนโทรลเลอร์ซึ่งเป็นหน่วยของค่าที่จะต้องใส่ให้ตัวจับเวลาหรือถ้าเทียบเป็นเวลาก็คือ 0.1 μs หลักการของวิธีนี้ก็คือเลือกตำแหน่งการสั่งแปลงค่าให้เหมาะสมโดยทั้ง 3 กรณีมีรายละเอียดดังนี้

1. กรณีที่ Timesh = 0 หรือไม่ต้องมีการเหลื่อมเวลา กรณีนี้จะไม่ต้องมีการขัดจังหวะของตัวจับเวลาหมายเลข 1 ไมโครคอนโทรลเลอร์จะสั่งให้ตัวแปลงดิจิทัลเป็นแอนะล็อกแปลงค่าของทั้งสองช่องสัญญาณพร้อมกัน
2. กรณีที่ Timesh < 500 MC สำหรับในกรณีนี้จะเลือกให้ไมโครคอนโทรลเลอร์สั่งแปลงค่าของช่องสัญญาณแรงดันท้ายช่วงเวลาของการขัดจังหวะเพื่อให้ช่วงห่างระหว่างท้ายการขัดจังหวะของตัวจับเวลาหมายเลข 0 ใกล้กับต้นการขัดจังหวะหมายเลข 1 ได้มากที่สุด ซึ่งจะมีผลทำให้สามารถรองรับค่า Timesh น้อย ๆ ได้โดยไม่เกิดการซ้อนทับกันของเวลาการขัดจังหวะทั้งสอง
3. กรณีที่ Timesh > 500 MC ในกรณีนี้ จะให้มีการสั่งแปลงค่าตั้งแต่ต้นของช่วงเวลาการขัดจังหวะ เพื่อให้ช่วงห่างระหว่างท้ายการขัดจังหวะของตัวจับเวลาหมายเลข 1

ใกล้กับต้นการขัดจังหวะหมายเลข 0 ได้มากที่สุด ซึ่งจะมีผลทำให้สามารถรองรับค่า Timesh มาก ๆ ได้โดยไม่เกิดการทับกันของเวลาการขัดจังหวะทั้งสอง

สำหรับในตำแหน่งของการแปลงค่าในส่วนขัดจังหวะของตัวจับเวลาหมายเลข 1 นั้นจะไม่จำเป็นต้องมีการเลือก เนื่องจากในส่วนนี้ไม่มีส่วนของคำสั่งอื่นอยู่เลยนอกจากคำสั่งแปลงค่าเท่านั้น

การปรับเปลี่ยนโปรแกรมควบคุมนี้ทำให้แผนผังการทำงานในส่วนของโปรแกรมขัดจังหวะเปลี่ยนไปเป็นดังรูปที่ 3.22 โดยสิ่งที่ต้องคำนึงเพิ่มเติมก็คือจังหวะในการเลื่อนข้อมูลเข้าไปที่ตัวแปลงดิจิทัลเป็นแอนะล็อกซึ่งจะไม่เหมือนกันในแต่ละกรณีนั่นคือ สำหรับกรณีที่ 1 และ 2 ด้านบนการเลื่อนค่าจะทำก่อนการแปลงค่าในการขัดจังหวะครั้งเดียวกันโดยจะทำการเลื่อนค่าก่อนแล้วจึงสั่งแปลงค่าในภายหลัง แต่สำหรับในกรณีที่ 3 จะมีการสั่งแปลงค่าที่ต้นคาบการขัดจังหวะ ทำให้ต้องมีการเลื่อนค่าข้อมูลเข้าไปในตัวแปลงดิจิทัลเป็นแอนะล็อกเตรียมไว้ล่วงหน้าสำหรับช่องสัญญาณแรงดัน (DAC1)




*รูปที่ 3.22 แผนผังการทำงานของโปรแกรมขัดจังหวะที่ปรับขั้นตอนวิธีใหม่*