บทที่ 3

การประยุกต์ใช้ไมโครคอนโทรลเลอร์เป็นโครงข่ายประสาทเพื่อลดสัญญาณรบกวน

ในบทนี้เป็นการกล่าวถึงฮาร์ดแวร์ของระบบ ซึ่งโดยหลัก ๆก็คือไมโครคอนโทรลเลอร์ และวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก ซึ่งการทำงานของโครงข่ายประสาท ADALINE นั้นจะต้องมี อินพุต เอาต์พุต และส่วนประมวลผล ดังนั้นการนำไมโครคอนโทรเลอร์มาประยุกต์ใช้งานเป็นโครงข่ายประสาท ADALINE ก็จะต้องมี ส่วนนำข้อมูลเข้า (Input) ส่วนประมวลผล (Processing) และส่วนนำข้อมูลออก (Output) ด้วยเช่นกัน และนอกจากนี้ การนำสัญญาณไฟฟ้าของกล้ามเนื้อลายซึ่งปกติจะมีขนาดแรงดันต่ำมากประมาณ 50 ไมโครโวลท์ถึง 100 มิลลิโวลท์ ป้อนเป็นสัญญาณอินพุตให้แก่โครงข่าย ADALINE จะต้องทำการขยายสัญญาณก่อนส่งให้กับอินพุตของไมโครคอนโทรลเลอร์ ซึ่งในงานวิจัยนี้ได้มีการประยุกต์ใช้ไมโครคอนโทรลเลอร์ 2 ตระกูล คือ AVR ATmega32 และ dsPIC30F2010 เนื่องจากการทดสอบในช่วงแรกจำเป็นต้องใช้ AVR ATmega32 จำนวน 2 ตัว เพื่อช่วยกันทำงานเนื่องจากหากใช้ตัวเดียว จะไม่สามารถรองรับอัตราสุ่มค่าสัญญาณ (Sampling Rate) 1000 ข้อมูลต่อวินาทีได้ จึงจำเป็นต้องใช้ AVR ATmega32 จำนวน 2 ตัว ซึ่งก็สามารถกำจัดสัญญาณรบกวนได้ดี แต่พบว่าหากใช้ไมโครคอนโทรลเลอร์ ATmega32 ซึ่งมีขนาด 40 ขาจำนวน 2 ตัว จะทำให้ใช้พื้นที่ในการวางชิปไอซีมากขึ้น ไม่เหมาะสมในการนำไปใช้งาน จึงได้เปลี่ยนมาใช้ไมโครคอนโทรลเลอร์ dsPIC30F2010 เป็นชิปประมวลผลสัญญาณ ซึ่งมีความเร็วในการทำงานสูงกว่า และสามารถรองรับอัตราการสุ่มค่าสัญญาณ 1000 ข้อมูลต่อวินาทีได้ และมีขนาดทางกายภาพเล็กกว่าจึงมีความเหมาะสมที่จะนำไปประยุกต์ใช้งาน โดยลักษณะของฮาร์ดแวร์ที่เกี่ยวข้องกับการทดสอบจะได้กล่าวถึงดังต่อไปนี้

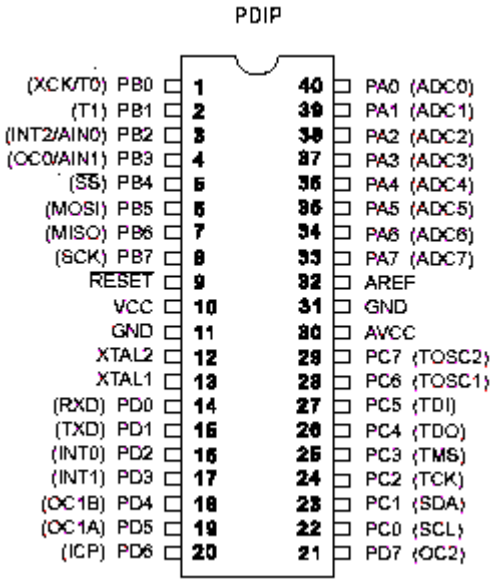
3.1 เครื่องขยายสัญญาณ

เครื่องขยายสัญญาณที่นำมาใช้ขยายสัญญาณไฟฟ้าของกล้ามเนื้อลาย มีอัตราการขยายสัญญาณ 46 dB ผลตอบสนองความถี่ย่าน 10 – 500 เฮิรตซ์ สามารถกำจัดแรงดันออฟเซตได้ −150 ถึง 150 มิลลิโวลท์ ซึ่งได้จากการใช้ชิปวงจรขยายอินสตรูเมนต์ เบอร์ INA114 ของบริษัท เบอร์บราว์ คอร์เปอร์เรชั่น จำกัด (สาวิตร์ สุวรรณรัตน์.,2546)

3.2 ไมโครคอนโทรลเลอร์ AVR ATmega32

ไมโครคอนโทรลเลอร์ตระกูลแรกที่นำมาประยุกต์ใช้เป็นโครงข่ายประสาท - ADALINE คือไมโครคอนโทรลเลอร์เบอร์ ATmega32 ของบริษัท ATMEL ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต มีสถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer) ซึ่งมีความเร็วในการประมวลผล 1 ล้านคำสั่งต่อ 1 วินาที ต่อ 1 MHz (1 MIPS/MHz) ซึ่งบอร์ดที่นำมาทดลองใช้คริสตอล 16 MHz ดังนั้นจึงทำให้ไมโครคอนโทรลเลอร์เบอร์นี้มีความเร็วในการประมวลผล 16 ล้านคำสั่ง ต่อ 1 วินาที (16 MIPS) และคุณสมบัติอื่นๆ มีดังต่อไปนี้

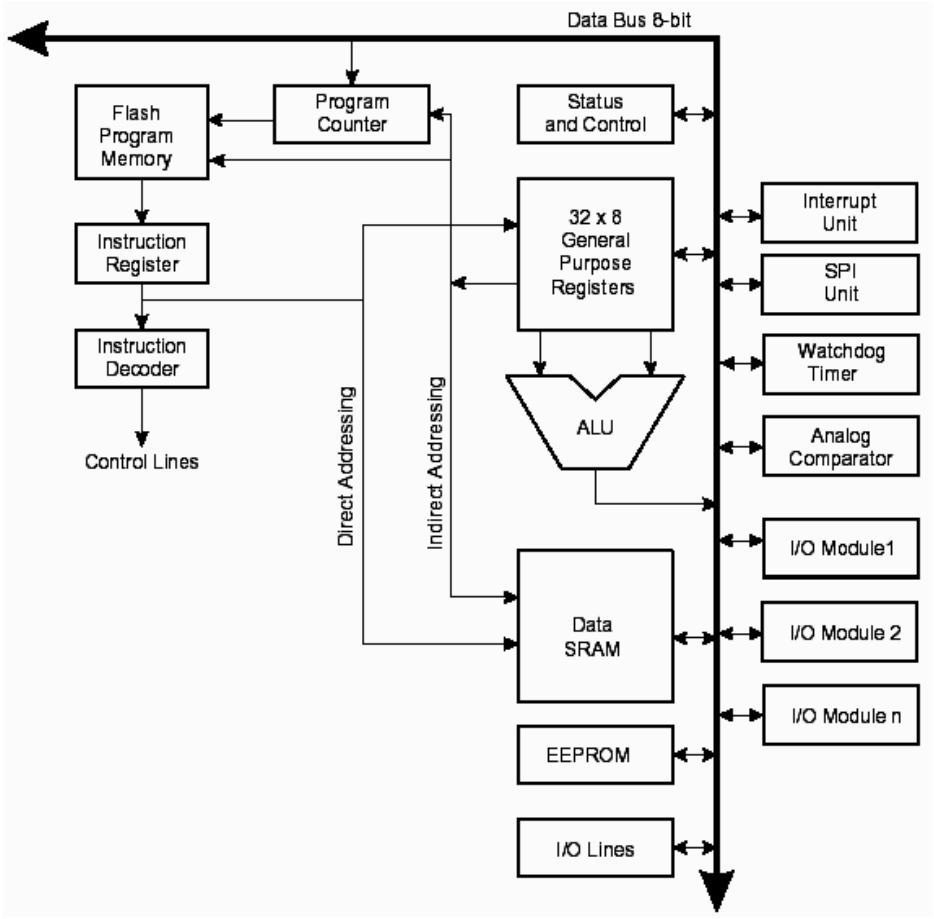
- พอร์ตอินพุต เอาท์พุตแบบดิจิตอล 4 พอร์ต พอร์ตละ 8 บิต
- หน่วยความจำแบบแฟลช (Flash) เพื่อเก็บโปรแกรม ขนาด 32 กิโลไบต์
- หน่วยความจำชนิดแรม (SRAM) ขนาด 2 กิโลไบต์
- หน่วยความจำแบบอีอีพรอม (EEPROM) ขนาด 1 กิโลไบต์
- ระบบการเปลี่ยนสัญญาณอะนาลอกเป็นดิจิตอล (Analog to Digital Converter) ขนาด 10 บิตจำนวน 8 ช่อง
- คำสั่งในการควบคุมการทำงานของไมโครคอนโทรลเลอร์จำนวน 131 คำสั่ง
- ความเร็วในการทำงาน 1 คำสั่งต่อ 1 สัญญาณนาฬิกา
- วงจรนับ/จับเวลา ขนาด 8 บิต 2 ตัว
- วงจรนับ/จับเวลา ขนาด 16 บิต 1 ตัว
- ระบบการสื่อสารข้อมูลดิจิตอลแบบอะซิงโครนัส(UART) 1 ช่อง
- ระบบการสื่อสารข้อมูลดิจิตอลแบบซิงโครนัส(SPI) 1 ช่อง
- ระบบตรวจจับการทำงานผิดพลาดของไมโครคอนโทรลเลอร์(Watchdog Timer)
- ระบบการกำเนิดสัญญาณแบบ PWM 1 ช่อง
- ระบบการอินเตอร์รัพต์จากภายนอก (External Interrupt)
- แรงดันใช้งาน 4.5 – 5.5 โวลท์



ภาพประกอบ 3-1 แสดงลักษณะของ AVR ATmega32 และตำแหน่งขา

3.2.1 สถาปัตยกรรม

โครงสร้างภายในประกอบด้วยรีจิสเตอร์ใช้งานทั่วไปขนาด 8 บิตจำนวน 32 ตัว ที่สามารถเข้าถึงข้อมูลได้ภายใน 1 สัญญาณนาฬิกา (1 Clock) โดยรีจิสเตอร์ R26 – R31 เป็นรีจิสเตอร์ขนาด 8 บิตจำนวน 6 ตัวที่สามารถจับคู่เพื่อใช้เป็นรีจิสเตอร์ขนาด 16 บิต 3 ตัว โดยใช้ชื่อว่า รีจิสเตอร์ X,Y และ Z



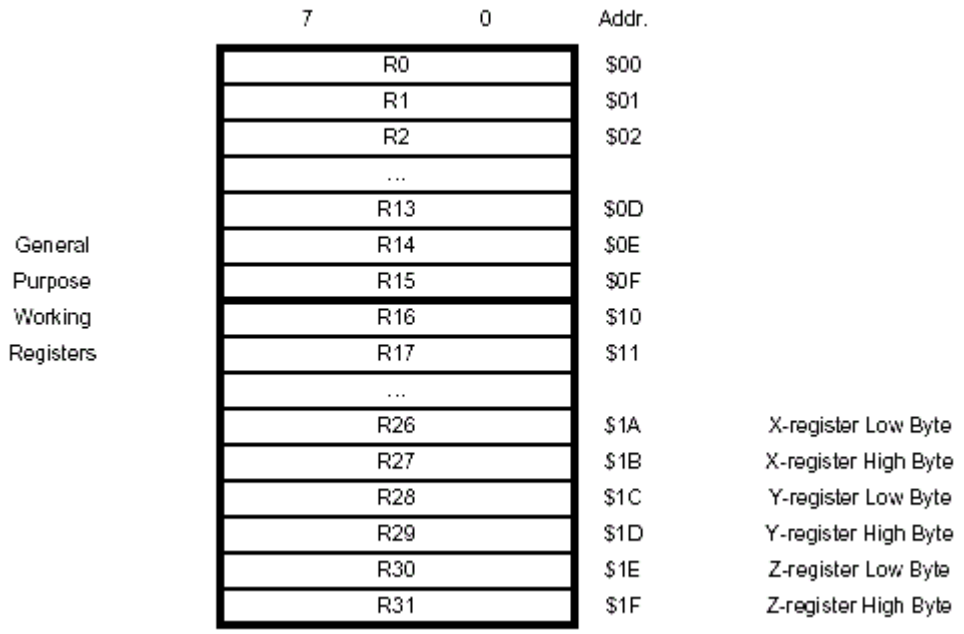
ภาพประกอบ 3-2 สถาปัตยกรรมแบบ RISC ของ AVR ATmega32

3.2.2 หน่วยประมวลผลทางคณิตศาสตร์และลอจิก

ระบบการประมวลผลที่มีประสิทธิภาพของ AVR คือ ALU (Arithmetic Logic Unit) ซึ่งสามารถสื่อสารข้อมูลโดยตรงกับรีจิสเตอร์ใช้งานทั่วไปทั้ง 32 ตัว โดย ALU ได้จัดแบ่งข้อมูลไว้ 3 ส่วนคือ ส่วนจัดการทางคณิตศาสตร์ (Arithmetic Function) ส่วนของการกระทำทางลอจิก (Logic Function) และส่วนของการกระทำกับบิต (Bit Function) และมีส่วนสนับสนุนอื่นที่มีประโยชน์เกี่ยวกับการคูณตัวเลขแบบคิดเครื่องหมายหรือไม่คิดเครื่องหมายบวกหรือลบและการคำนวณแบบเศษส่วน

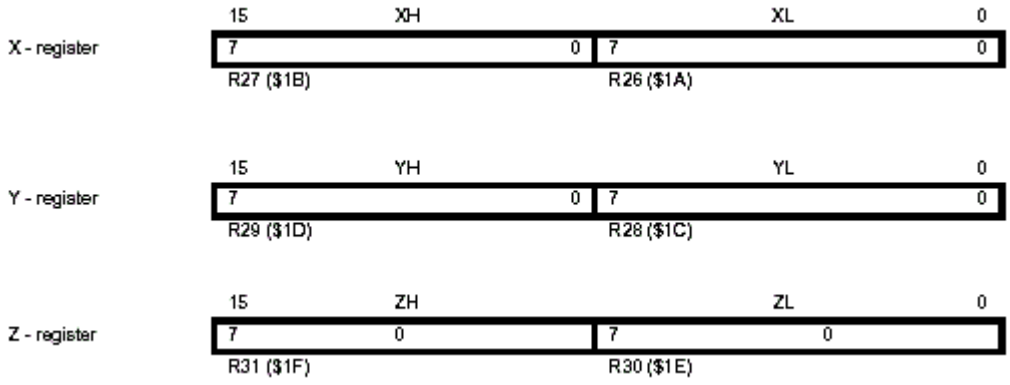
3.2.3 รีจิสเตอร์ใช้งานทั่วไป

รีจิสเตอร์ใช้งานทั่วไปมีขนาด 8 บิตจำนวน 32 ตัว คือ R0 – R31 ดังแสดงโครงสร้างตามภาพประกอบ 3-3 โดยรีจิสเตอร์ทั้งหมดสามารถใช้ชุดคำสั่ง เพื่อเข้าถึงได้โดยตรงและใช้เวลาในการเข้าถึงเพียง 1 สัญญาณนาฬิกา ซึ่งอาจจะเป็นคำสั่งที่กระทำระหว่างค่าคงที่กับรีจิสเตอร์ หรือ รีจิสเตอร์กับรีจิสเตอร์



ภาพประกอบ 3-3 แสดงโครงสร้างของรีจิสเตอร์ใช้งานทั่วไป

นอกจากนี้ รีจิสเตอร์ R26 – R31 ยังเป็นรีจิสเตอร์ขนาด 8 บิตจำนวน 6 ตัวที่สามารถจับคู่เพื่อใช้เป็น รีจิสเตอร์ขนาด 16 บิต 3 ตัว โดยใช้ชื่อว่ารีจิสเตอร์ X,Y และ Z เพื่อเป็นตัวชี้หน่วยความจำ หรือเพื่อการเก็บข้อมูลที่มากขึ้น



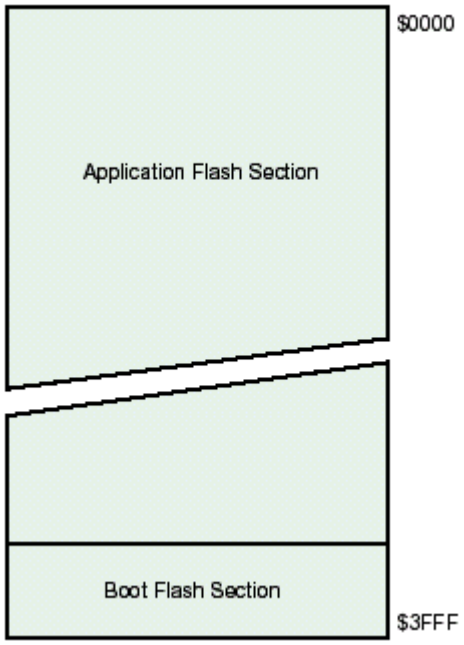
ภาพประกอบ 3-4 แสดงรีจิสเตอร์ X, Y และ Z

3.2.4 หน่วยความจำ

หน่วยความจำของ AVR มี 2 ส่วนหลัก ๆ คือ หน่วยความจำโปรแกรม(Program Memory) และหน่วยความจำข้อมูล (Data Memory)

หน่วยความจำโปรแกรม (Program Memory)

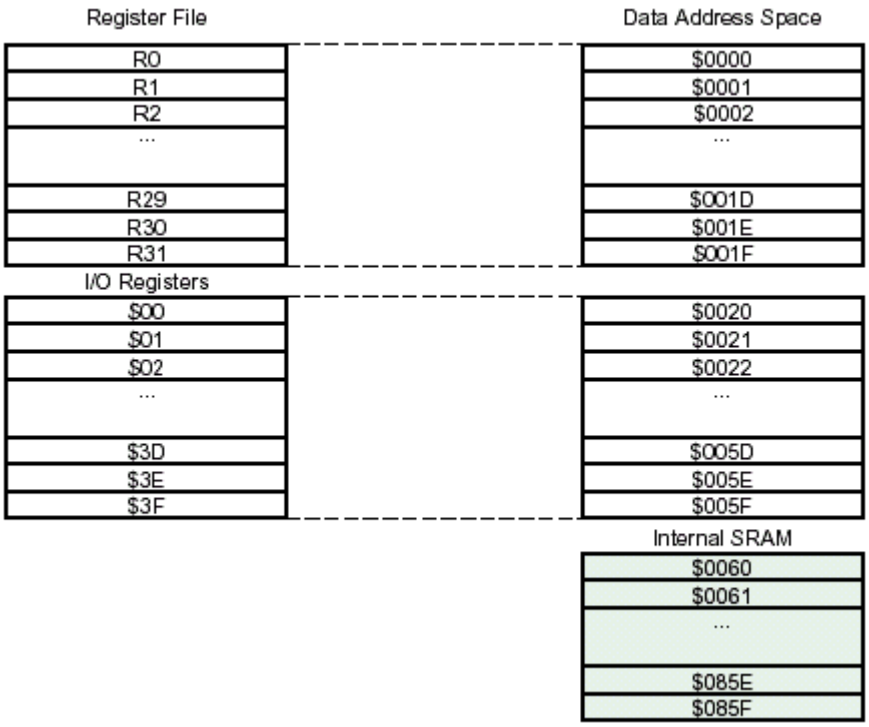
เป็นหน่วยความจำสำหรับเก็บโปรแกรม ซึ่งมีลักษณะการทำงานแบบแฟลช (Flash Memory) ขนาด 32 กิโลไบต์ และเพื่อความปลอดภัยของโปรแกรมได้ทำการแบ่งหน่วยความจำแบบแฟลช เป็น 2 ส่วนคือ ส่วนโปรแกรมบูต (Boot Program Section) และ ส่วนโปรแกรมประยุกต์ (Application Program Section) ดังแสดงในภาพประกอบ 3-5 ซึ่งหน่วยความจำโปรแกรมแบบแฟลชสามารถลบหรือเขียนโปรแกรมได้ถึง 10,000 ครั้ง



ภาพประกอบ 3-5 โครงสร้างหน่วยความจำโปรแกรม

หน่วยความจำข้อมูล (Data Memory)

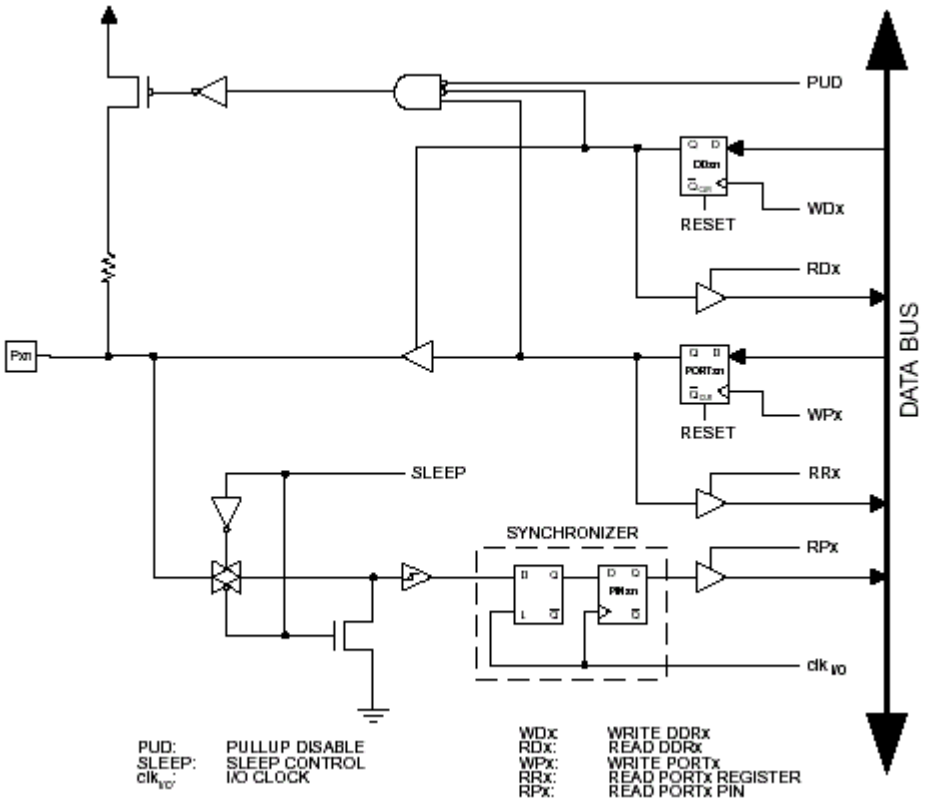
เป็นหน่วยความจำ ที่ใช้เก็บข้อมูลในระหว่างที่ไมโครคอนโทรลเลอร์ประมวลผล ซึ่งมีอยู่ทั้งหมด 2114 ไบต์ โดยแบ่งเป็นหน่วยความจำของรีจิสเตอร์ 32 ไบต์ หน่วยความจำรีจิสเตอร์อินพุตเอาท์พุต (I/O Register) 64 ไบต์ และหน่วยความจำข้อมูลทั่วไปแบบ SRAM 2048 ไบต์ ดังแสดงในภาพประกอบ 3-6



ภาพประกอบ 3-6 แสดงหน่วยความจำข้อมูล

3.2.5 พอร์ตอินพุตและพอร์ตเอาท์พุต

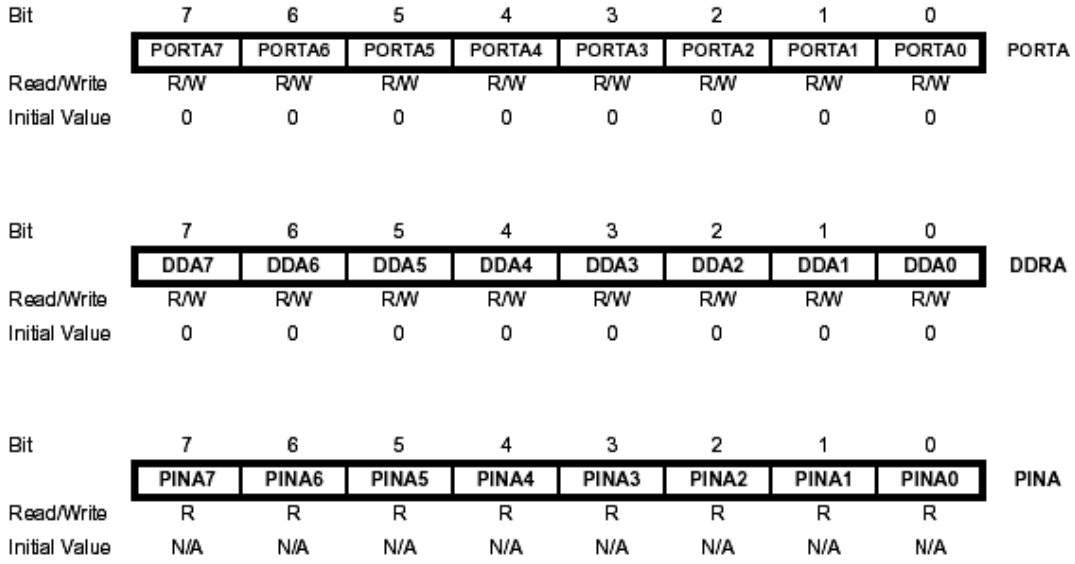
ไมโครคอนโทรลเลอร์ AVR ATmega32 มีพอร์ตอินพุตเอาท์พุตจำนวน 4 พอร์ตด้วยกัน คือ พอร์ต A พอร์ต B พอร์ต C และพอร์ต D แต่ละพอร์ตสามารถใช้งาน เป็นพอร์ตอินพุตหรือพอร์ตเอาท์พุตแบบดิจิตอลได้ โดยการกำหนดค่าในรีจิสเตอร์ควบคุมพอร์ตของแต่ละพอร์ต แต่ละขาของแต่ละพอร์ตสามารถเลือกหรือไม่เลือกใช้ความต้านทาน PULL UP ได้ ซึ่งโครงสร้างการทำงานของแต่ละขาของแต่ละพอร์ตทำงานแบบดิจิตอล มีลักษณะดังภาพประกอบ 3-7



ภาพประกอบ 3-7 แสดงโครงสร้างของพอร์ต ต่างๆ

พอร์ต A

พอร์ต A เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยมีหน่วยควบคุมการทำงานต่างๆ ของพอร์ต คือ รีจิสเตอร์พอร์ต A (DATA REGISTER) อยู่ที่ตำแหน่ง $1B($3B) , รีจิเตอร์ DDRA (DATA DIRECTION REGISTER) อยู่ที่ตำแหน่ง $1A($3A) และ PIN A (PORT A INPUT PINS) อยู่ที่ตำแหน่ง $19($39) โดย PIN A จะสามารถอ่านได้อย่างเดียวไม่สามารถเขียนข้อมูลลงไปได้ ในขณะที่พอร์ต A และ DDRA สามารถอ่านและเขียนได้ โดยแต่ละขาสัญญาณของพอร์ต A สามารถกำหนดให้ความต้านทานที่ PULL UP ทำงานในสภาวะ ACTIVE หรือไม่ ACTIVE ก็ได้ แต่ละขาสัญญาณของพอร์ต A จะสามารถรับกระแส (SINK CURRENT) ได้ 20 mA



ภาพประกอบ 3-8 แสดงส่วนประกอบของรีจิสเตอร์ควบคุมพอร์ต A

พอร์ต A นอกจากจะใช้เป็นพอร์ตอินพุตหรือเอาท์พุตแล้วยังสามารถนำไปใช้เป็นขารับสัญญาณของระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอล (Analog To Digital Converter) ดังแสดงในตารางประกอบ 3-1

ตารางที่ 3-1 แสดงชื่อและหน้าที่แต่ละบิตของพอร์ต A

พอร์ต A บิต	หน้าที่
PA7	ADC7 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 7)
PA6	ADC6 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 6)
PA5	ADC5 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 5)
PA4	ADC4 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 4)
PA3	ADC3 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 3)
PA2	ADC2 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 2)
PA1	ADC1 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 1)
PA0	ADC0 (อะนาลอกเป็นดิจิตอลช่องสัญญาณที่ 0)

พอร์ต B

เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยมีจิสเตอร์ควบคุมพอร์ตอยู่ 3 ตัว คือ รีจิสเตอร์พอร์ต B อยู่ที่ตำแหน่ง $18($38) , รีจิเตอร์ DDRB อยู่ที่ตำแหน่ง $17($37) และ PIN B อยู่ที่ตำแหน่ง $16($36) โดย PIN B จะสามารถอ่านได้อย่างเดียวไม่สามารถเขียนข้อมูลลงไปได้ ในขณะที่พอร์ต B และ DDRB สามารถอ่านและเขียนได้ โดยแต่ละขาของพอร์ต B สามารถแยกกำหนดให้มีความต้านทาน PULL UP ได้ตามต้องการ ซึ่งในแต่ละขาสามารถรับกระแส (SINK CURRENT) ได้ 20 mA

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

ภาพประกอบ 3-9 แสดงส่วนประกอบของรีจิสเตอร์ควบคุมพอร์ต B

พอร์ต C

เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยมีรีจิสเตอร์ควบคุมการทำงานของพอร์ต 3 ตัว คือ รีจิสเตอร์พอร์ต C อยู่ที่ตำแหน่ง $15($35), รีจิเตอร์ DDRC อยู่ที่ตำแหน่ง $14 ($34) และ PIN C อยู่ที่ตำแหน่ง $13($33) โดย PIN C จะสามารถอ่านได้อย่างเดียวไม่สามารถเขียนข้อมูลลงไปได้ ในขณะที่พอร์ต C และ DDRC สามารถอ่านและเขียนได้ โดยแต่ละขาของพอร์ต C สามารถแยกกำหนดความต้านทาน PULL UP ได้ตามต้องการ ซึ่งแต่ละขาสามารถรับกระแส (SINK CURRENT) ได้ 20 mA

Bit	7	6	5	4	3	2	1	0	
	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

ภาพประกอบ 3-10 แสดงส่วนประกอบของรีจิสเตอร์ควบคุมพอร์ต C

พอร์ต D

เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยมีรีจิสเตอร์ควบคุมการทำงานของพอร์ต 3 ตัว คือ รีจิสเตอร์พอร์ต D อยู่ที่ตำแหน่ง $12($32), รีจิเตอร์ DDRD อยู่ที่ตำแหน่ง $11($31) และ PIN D อยู่ที่ตำแหน่ง $10($30) โดย PIN D จะสามารถอ่านได้อย่างเดียวไม่สามารถเขียนข้อมูลลงไปได้ ในขณะที่พอร์ต D และ DDRD สามารถอ่านและเขียนได้ โดยแต่ละขาของพอร์ต D สามารถแยกกำหนดความต้านทาน PULL UP ได้ตามต้องการ ซึ่งแต่ละขาสามารถรับกระแส (SINK CURRENT) ได้ 20 mA

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

ภาพประกอบ 3-11 แสดงส่วนประกอบของรีจิสเตอร์ควบคุมพอร์ต D

การควบคุมพอร์ตแต่ละพอร์ต

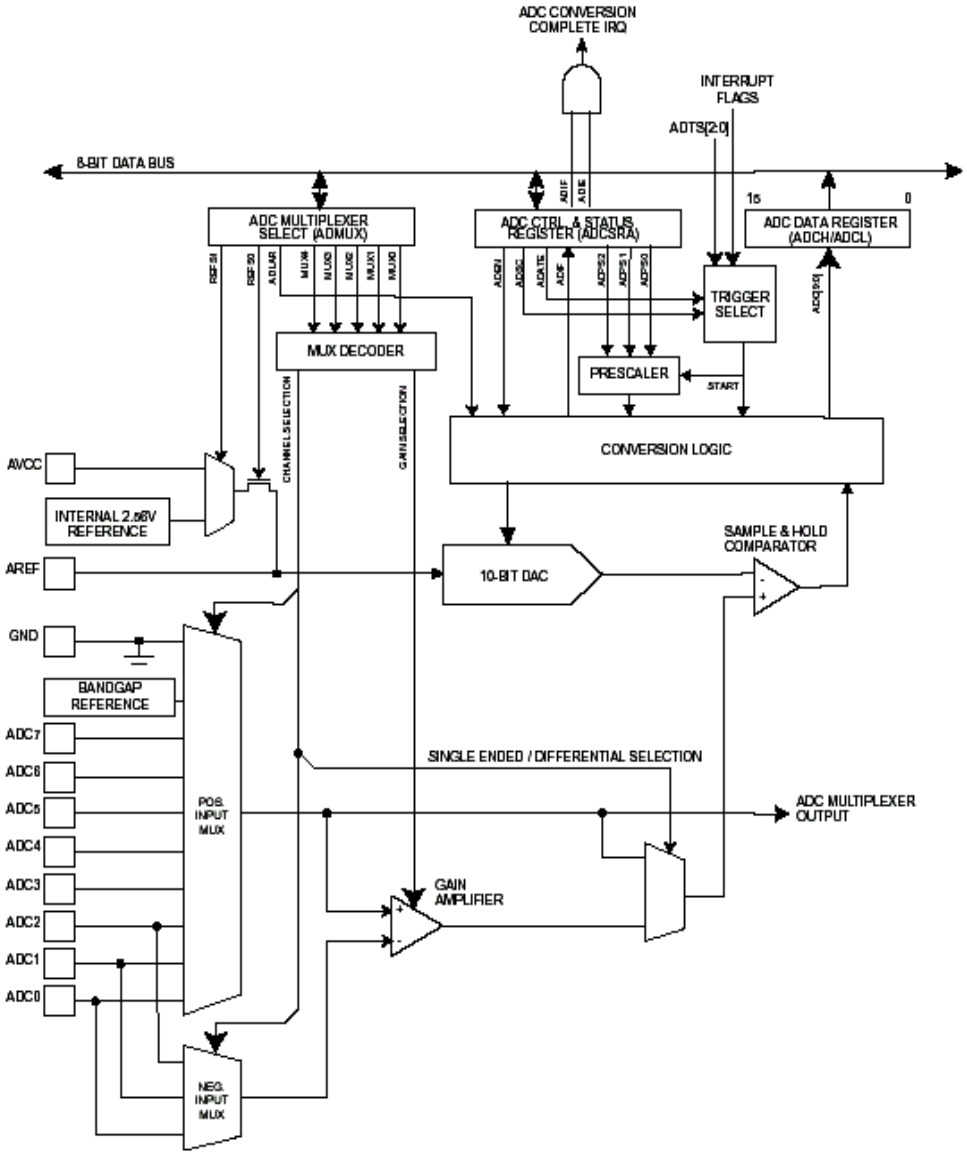
บิต DDXn ในรีจิสเตอร์ DDRX จะเป็นบิตที่ใช้ในการกำหนดทิศทางของแต่ละพอร์ต ถ้าบิต DDXn ถูกกำหนดให้เป็น 1 จะทำให้ขาของพอร์ตนั้นเป็นเอาท์พุต แต่หากเมื่อ DDXn ถูกกำหนดเป็น 0 จะทำให้ขานั้นถูกกำหนดให้เป็นอินพุต ซึ่งสภาวะของขาต่าง ๆ แสดงดังตารางที่ 3-2

ตารางที่ 3-2 แสดงคุณสมบัติของการกำหนดบิต DDXn ในรีจิสเตอร์ DDRX

DDXn	PORTXn	อินพุต/เอาท์พุต	PULL UP	หมายเหตุ
0	0	อินพุต	NO	3 สถานะ (Tri-State)
0	1	อินพุต	YES	จะจ่ายกระแส ถ้ามี PULL Low ภายนอก
1	0	เอาท์พุต	NO	ให้เอาท์พุตมีสถานะเป็นลอจิก 0
1	1	เอาท์พุต	NO	ให้เอาท์พุตมีสถานะเป็นลอจิก 1

3.2.6 การแปลงสัญญาณอะนาลอกเป็นสัญญาณดิจิตอล

การนำสัญญาณอะนาลอกเข้าสู่ไมโครคอนโทรลเลอร์ต้องทำการแปลงเป็นสัญญาณดิจิตอลเสียก่อนจึงจะทำให้ไมโครคอนโทรลเลอร์สามารถนำสัญญาณอะนาลอกที่ได้ไปประมวลผลต่อได้ ซึ่งไมโครคอนโทรลเลอร์ ATmega32 มีระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด 10 บิตจำนวน 8 ช่องอยู่ภายในตัวแล้ว โดยแต่ละช่องจะรับสัญญาณเข้ามาทางพอร์ต A ซึ่งในระบบจะมีวงจร SAMPLE AND HOLD เพื่อช่วยให้สัญญาณอะนาลอกที่รับเข้ามาแปลงเป็นสัญญาณดิจิตอลมีระดับสัญญาณคงที่ ซึ่งค่าที่น้อยที่สุดของการแปลงอะนาลอกเป็นดิจิตอลคือ 0 โวลท์ หรือกราวด์ และค่าสูงสุดคือค่าของแรงดันอ้างอิง (AREF) และการเลือกช่องสัญญาณ (ADC Multiplexer Select) สามารถกำหนดได้ด้วยการกำหนดค่าในรีจิสเตอร์ ADMUX และข้อมูลที่ได้จากการแปลงสัญญาณอะนาลอกเป็นดิจิตอลจะถูกเก็บไว้ในรีจิสเตอร์ ADCH และ ADCL ดังแสดงลักษณะการทำงานในภาพประกอบ 3-12

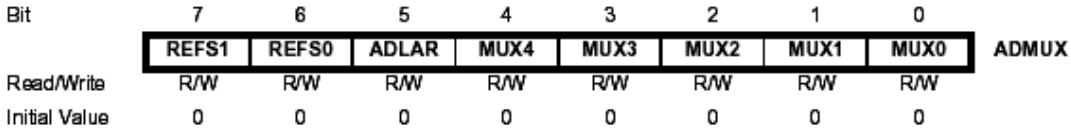


ภาพประกอบ 3-12 แสดงบล็อกไดอะแกรมของระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอล

การควบคุมให้ระบบการแปลงสัญญาณอะนาลอกเป็นดิจิตตอลทำงานอย่างถูกต้องตามที่ต้องการนั้นต้องกำหนดค่าที่ริจิสเตอร์ (Register) ต่าง ๆที่เกี่ยวข้องดังต่อไปนี้

รีจิสเตอร์ ADMUX

เป็นรีจิสเตอร์เลือกช่องสัญญาณอะนาลอกโดยแต่ละบิตที่ใช้ควบคุมมีความหมายดังต่อไปนี้



ภาพประกอบ 3-13 แสดงส่วนประกอบของรีจิสเตอร์ ADMUX

- บิต 7 และ 6 REFS1และ REFS0 เป็นบิตเลือกแรงดันอ้างอิงของระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอล โดยในขณะทำการแปลงสัญญาณอะนาลอกเป็นดิจิตอลจะไม่สามารถเปลี่ยนค่าในบิตเหล่านี้ได้จนกว่ากระบวนการแปลงสัญญาณจะแล้วเสร็จ

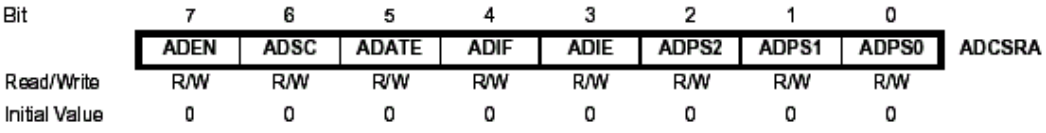
ตารางที่ 3-3 แสดงลักษณะการเลือกแรงดันอ้างอิงของบิต REFS ในรีจิสเตอร์ ADMUX

REFS1	REFS2	การเลือกแรงดันอ้างอิง
0	0	ปิดการใช้แรงดันอ้างอิงภายใน
0	1	จ่ายแรงดัน AVCC ที่ขา AREF
1	0	สงวน
1	1	แรงดันอ้างอิงภายใน 2.56 โวลท์ด้วยคาปาซิเตอร์ที่ขา AREF

- บิต 5 ADLAR เป็นบิตควบคุมการอ่านข้อมูลจากรีจิสเตอร์ ADCH และ ADCL

- บิต 4 ถึง 0 MUX4 ถึง MUX0 เป็นบิตควบคุมการเลือกช่องสัญญาณและอัตราการขยาย

รีจิสเตอร์ ADCSRA



ภาพประกอบ 3-14 แสดงส่วนประกอบรีจิสเตอร์ ADCSRA

- บิต 7 ADEN เมื่อต้องการให้ระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอลทำงานต้องกำหนดให้บิตนี้เป็น 1 และเมื่อไม่ต้องการให้ระบบทำงาน กำหนดบิตนี้ให้เป็น 0

- บิต 6 ADSC เมื่อต้องการให้ระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอลเริ่มต้นทำงานกำหนดให้บิตนี้เป็น 1 แต่ต้องกำหนดหลังจากที่กำหนดให้บิต ADEN เป็น 1 แล้ว

- **บิต 5 ADATE** เมื่อกำหนดบิตนี้เป็น 1 ทำให้ระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอลทำงานโดยอัตโนมัติโดยเริ่มต้นทำงานที่ขอบขาขึ้นของสัญญาณ

- **บิต 4 ADIF** บิตนี้จะถูกเซ็ต (Set) ให้เป็น 1 โดยอัตโนมัติ เมื่อการแปลงอะนาลอกเป็นดิจิตอลเสร็จสมบูรณ์

- **บิต 3 ADIE** หากบิตนี้ถูกกำหนดให้เป็น 1 จะทำให้เกิดการอินเตอร์รัพต์

- **บิต 2 ถึง 0 ADPS2 ถึง ADPS0** เป็นบิตที่ใช้กำหนดตัวหารเพื่อกำหนดอัตราสุ่ม (Sampling Rate) ของระบบแปลงสัญญาณอะนาลอกเป็นดิจิตอล ซึ่งแสดงดังตารางที่ 3-4

ตารางที่ 3-4 แสดงบิตที่ใช้กำหนดตัวหารเพื่อกำหนดอัตราสุ่ม

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

รีจิสเตอร์ ADCL , ADCH

เมื่อการแปลงสัญญาณอะนาลอกเป็นดิจิตอลเสร็จสมบูรณ์ ข้อมูลที่ได้จะถูกเก็บในรีจิสเตอร์ ADCL และ ADCH โดยรีจิสเตอร์ ADCH เก็บค่า 2 บิตบนที่ได้จากการแปลงอะนาลอกเป็นดิจิตอลและรีจิสเตอร์ ADCL เก็บค่า 8 บิตล่างที่ได้จากการแปลงสัญญาณอะนาลอกเป็นดิจิตอล ดังแสดงในภาพประกอบ 3-15

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ภาพประกอบ 3-15 แสดงส่วนประกอบรีจิสเตอร์ ADCH และ ADCL

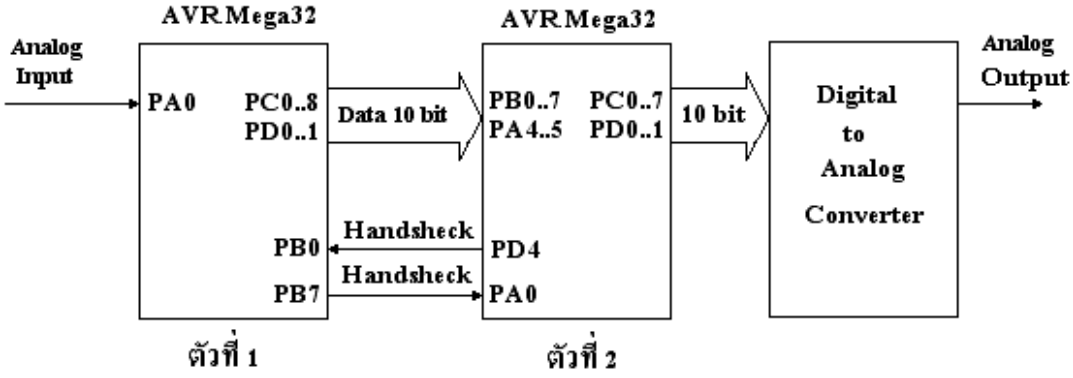
3.3 การนำไมโครคอนโทรลเลอร์ AVR ATmega32 ประยุกต์เป็นโครงข่ายประสาท ADALINE

จากผลการวิจัย (รักกฤตว์ ดวงสร้อยทอง.,2544) ได้สรุปว่าการทำงานของโครงข่าย ADALINE ขนาด 8 แท็ป (Tab) ระยะการหน่วงเวลา (Delay) เท่ากับ 10 และอัตราการเรียนรู้ของโครงข่ายประสาท ADALINE เท่ากับ 0.005 สามารถกำจัดสัญญาณรบกวนที่มีลักษณะเป็นคาบได้ดีที่สุด โดยใช้อัตราสุ่มค่าสัญญาณ (Sampling Rate) 1000 ข้อมูลต่อวินาที (1000 Sampling / Sec) จึงได้ออกแบบให้ไมโครคอนโทรลเลอร์มีการทำงาน 2 ส่วนหลักๆ คือ

1) การรับข้อมูลอินพุตและเลื่อนข้อมูล เพื่อสร้างเป็นสัญญาณอ้างอิงให้กับโครงข่าย ADALINE

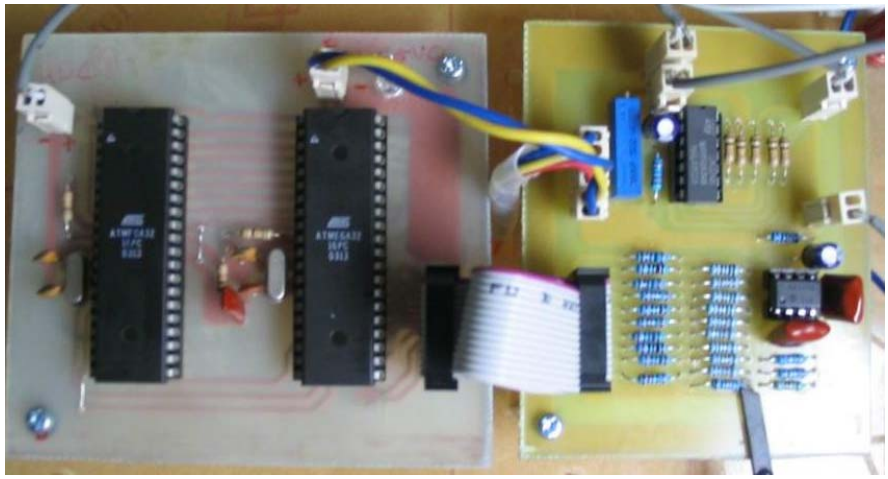
2) การทำงานเป็นโครงข่าย ADLINE และส่งข้อมูลเอาท์พุตของโครงข่าย ADALINE ให้กับวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก

ดังนั้นจึงจำเป็นต้องใช้ ไมโครคอนโทรลเลอร์ AVR ATmega32 จำนวน 2 ตัว ทำงานร่วมกัน เพื่อให้ระบบสามารถทำงานด้วยอัตราสุ่มค่าสัญญาณ 1000 ข้อมูลต่อวินาที โดยให้ไมโครคอนโทรลเลอร์ AVR ATmega32 ตัวแรกทำหน้าที่รับสัญญาณอินพุตโดยแปลงสัญญาณอะนาลอกเป็นดิจิตอล และทำการเลื่อนข้อมูล เพื่อสร้างสัญญาณอ้างอิงให้กับโครงข่าย ADALINE ส่วนไมโครคอนโทรลเลอร์ AVR ATmega32 ตัวที่สองจะรับสัญญาณอินพุตและสัญญาณอ้างอิงมาจากไมโครคอนโทรลเลอร์ตัวแรก มาทำการคำนวณด้วยโครงข่าย ADALINE และนำเอาท์พุตที่ได้ส่งออกไปยังวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก ซึ่งมีลักษณะการทำงานดังภาพประกอบ 3-16 และหลังจากได้ข้อสรุปของวงจรต่างๆที่เกี่ยวข้องแล้ว ได้ทำการสร้างลายวงจรลงบนแผ่นปริ้นและทำการบัดกรี AVR ATmega32 และอุปกรณ์ต่างๆ ลงแผ่นปริ้นเพื่อนำไปใช้ในการทดสอบการทำงานของโครงข่าย ADALINE ต่อไปดังแสดงในภาพประกอบ 3-17 (รายละเอียดของวงจรให้ดูในภาคผนวก ผก.2 และ ผก.3)



ภาพประกอบ 3-16 แสดงบล็อกไดอะแกรมการประยุกต์ใช้ไมโครคอนโทรลเลอร์

AVR ATmega32 เป็นโครงข่าย ADALINE

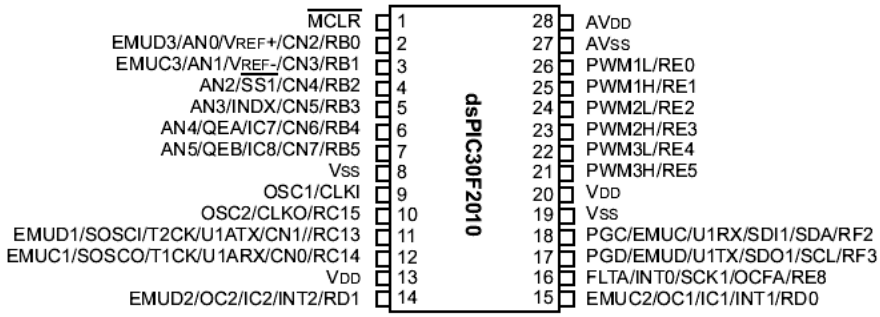


ภาพประกอบ 3-17 แสดงภาพวงจรไมโครคอนโทรลเลอร์ AVR ATmega32 ที่ใช้เป็นเป็นโครงข่าย ADALINE

3.4 ไมโครคอนโทรลเลอร์ dsPIC30F2010

dsPIC คือชื่อของไมโครคอนโทรลเลอร์ขนาด 16 บิต จากบริษัทไมโครชิป ซึ่งได้ออกแบบเป็น Digital Signal Controller (DSC) ซึ่งหมายความว่า dsPIC เป็นไมโครคอนโทรลเลอร์ที่ได้รับการออกแบบมาเป็นพิเศษเพื่องานประมวลผลสัญญาณดิจิตอลสำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูงโดยเบอร์ที่นำมาใช้เป็นโครงข่าย ADALINE คือ dsPIC30F2010

28-Pin SDIP and SOIC



ภาพประกอบ 3-18 แสดงลักษณะของ dsPIC302010 และตำแหน่งขา

3.4.1 คุณสมบัติของไมโครคอลโทรลเลอร์

- มี 84 คำสั่งมาตรฐานสามารถรองรับรูปแบบการอ้างแอดเดรสได้อย่างอิสระ
- ชุดคำสั่งมีขนาด 24 บิตสามารถประมวลผลข้อมูลได้ 16 บิต
- มีหน่วยความจำโปรแกรมแบบแฟลช ความจุ 12 กิโลไบต์
- สามารถโปรแกรมหน่วยความจำโปรแกรมได้ด้วยตัวเอง
- มีหน่วยความจำข้อมูล อีอีพรอม 1 กิโลไบต์
- มีหน่วยความจำแรม 512 ไบต์
- ความเร็วในการทำงานสูงถึง 30 ล้านคำสั่งต่อวินาที
- ความถี่สัญญาณนาฬิกาภายนอกตั้งแต่ ย่านไฟตรง ถึง 40 MHz
- ความถี่สัญญาณนาฬิกาในกรณีใช้งานร่วมกับวงจรเฟสล็อกลูปภายในตั้งแต่ 4 MHzถึง 10 MHz ซึ่งเลือกได้ 3 ระดับ คือ 4, 8 หรือ 16 เท่า
- รองรับแหล่งกำเหนิดสัญญาณอินเตอร์รัพต์สูงสุด 62 แหล่ง รวมทั้งสัญญาณอินเตอรัพต์จากภายนอก 3 แหล่ง
- สามารถกำหนดระดับความสำคัญในการตอบสนองอินเตอร์รัพต์ ได้ 8 ระดับ
- มีอินเตอร์รัพต์เวกเตอร์ 48 แหล่ง
- มีวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ความละเอียด 10 บิต 6 ช่อง โดยมีอัตราการสุ่มและแปลงสัญญาณ 500 กิโลแซมเปิลต่อวินาที
- มีวงจรตรวจจับแรงดันไฟเลี้ยงต่ำกว่ากำหนดแบบโปรแกรมได้
- มีเพาเวอร์ ออนรีเซ็ตและเพาเวอร์ อัปไทเมอร์
- ย่านไฟเลี้ยง 2.5 ถึง 5.5 โวลท์ กระแสไฟฟ้า 2.6 ถึง 44 mA ที่ไฟเลี้ยง +5V ขึ้นอยู่กับความเร็วในการทำงาน

คุณสมบัติด้านการประมวลผลสัญญาณดิจิตอล

- มีแอกคิวมูเลเตอร์ขนาด 40 บิต จำนวน 2 ตัว รองรับการประมวลทางคณิตศาสตร์ได้เป็นอย่างดี
- มีหน่วยประมวลผลด้านการคูณและหารเลข 17 บิต ในรูปของฮาร์ดแวร์ จึงทำให้สามารถทำการคูณและหารเลขได้อย่างรวดเร็ว
- ทำการคูณเลขขนาด 16 บิตได้ภายใน 1 สัญญาณนาฬิกา
- มีตัวเลื่อนข้อมูลบาร์เรล 40 สเตจ ช่วยให้ประมวลผลข้อมูลที่จำนวนบิตมากทำได้รวดเร็ว

3.4.2 สถาปัตยกรรมของ dsPIC30F2010

หน่วยประมวลกลาง

หน่วยประมวลผลของ dsPIC30F2010 ใช้คำสั่งที่มีความยาว 1 เวิร์ด ขนาด 24 บิต โดยมีโปรแกรมเคาน์เตอร์ขนาด 23 บิต เพื่อแจ้งแอดเดรสของหน่วยความจำโปรแกรมที่เข้าไปประมวลผล dsPIC30F2010 มีความจุของหน่วยความจำโปรแกรม 12 กิโลไบต์ เมื่อคำสั่งมีความยาว 24 บิต จึงบรรจุคำสั่งได้จริง 4 กิโลเวิร์ด รีจิสเตอร์หลักที่ใช้ในการทำงานคือ รีจิสเตอร์ W (Working Register) สำหรับใน dsPIC30F2010 จะแตกต่างจากไมโครคอนโทรลเลอร์ PIC อย่างมาก โดย รีจิสเตอร์ W ได้รับการจัดโครงสร้างเป็น อะเรย์ขนาด 16 บิต จึงทำให้สามารถรองรับทั้งข้อมูล, ค่าแอดเดรส หรือค่าของรีจิสเตอร์ใด ๆ ที่ต้องการนำมาประมวลผล โดยใน dsPIC30F2010 มีรีจิสเตอร์ W ให้ใช้งานถึง 16 ตัว ส่วนใหญ่ใช้ในการประมวลผลหลัก ส่วนอีกตัวหนึ่งคือ รีจิสเตอร์ W15 จะใช้ทำงานร่วมกับตัวชี้สแต็กในการทำงานของโปรแกรมย่อยและบริการอินเตอร์รัพต์ ด้านการตอบสนองอินเตอร์รัพต์นั้น dsPIC30F2010 มีการจัดสรรพื้นที่เก็บค่าอินเตอร์รัปต์เวกเตอร์ไว้มากถึง 54 ตำแหน่ง และยังสามารถกำหนดระดับความสำคัญได้อีก 8 ระดับด้วย

หน่วยความจำ

dsPIC30F2010 มีหน่วยความจำโปรแกรม 4 กิโลเวิร์ด แอดเดรสอยู่ในช่วง 0x000100 ถึง 0x001FFE สามารถโปรแกรมหรือเขียนข้อมูลลงไปได้ 2 วิธีคือ

1) โดยใช้การโปรแกรมในวงจรแบบอนุกรมหรือ ICSP ผ่านทางขา PGD และ PGC (ขาที่ 17 และ 18) แล้วป้อนสัญญาณพัลส์แรงดันสูงสำหรับโปรแกรมผ่านเข้ามาทางขา MCLR

2) โปรแกรมตัวเองในขณะทำงานหรือ RTSP (Run Time Self Programming) ส่วนหน่วยความจำข้อมูลแรมนั้น ได้จัดสรรเป็น 2 ส่วนคือ หน่วยความจำข้อมูลแรม X และ Y แต่ละส่วนมีขนาด 16 บิต ความจุ 256 ไบต์ รวมเป็น 512 ไบต์ โดยแต่ละส่วนจะมีตัวกำหนดแอดเดรสแยกออกจากกัน เรียกว่า AGU (Address Generation Unit) ส่วนหน่วยความจำข้อมูลอีอีพรอม มีความจุ 1 กิโลไบต์ จัดสรรไว้ที่แอดเดรส 0x7FFC00 ถึง 0x7FFFFE

ส่วนประมวลผลสัญญาณดิจิตอล (DSP Engine)

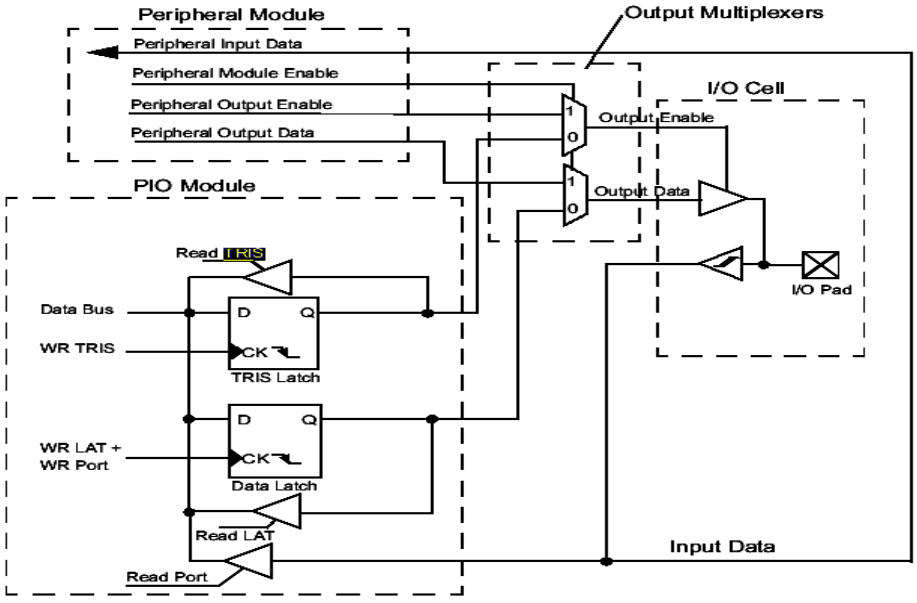
เนื่องจาก dsPIC ได้รับการออกแบบมาให้ทำงานในด้านการประมวลผลสัญญาณดิจิตอลเป็นหลัก ดังนั้นจึงต้องมีการเพิ่มความสามารถในหน่วยคำนวณทางคณิตศาสตร์และลอจิกอย่างมาก โดยในส่วนประมวลผลสัญญาณดิจิตอลมีหน่วยจัดสรรการคูณเลขขนาด 17 x 17 บิต ความเร็วสูง, หน่วยประมวลผลทางคณิตศาสตร์และลอจิกหรือ ALU ขนาด 40 บิต , แอกคิวมูเลเตอร์ขนาด 40 บิต อีก 2 ตัว ตัวเลื่อนข้อมูล 2 ทิศทางแบบบาเรล (Barrel Shifter) ขนาด 40 บิต จึงทำให้สามารถจัดการข้อมูลขนาด 16 บิตได้เสร็จสิ้นภายใน 1 สัญญาณนาฬิกา

3.4.3 พอร์ตอินพุตเอาท์พุตของไมโครคอนโทรลเลอร์ dsPIC30F2010

ด้วยคุณสมบัติของไมโครคอนโทรลเลอร์สมัยใหม่ ขาต่อใช้งานหรือพอร์ตอินพุตเอาท์พุตนั้นมักมีความสามารถพิเศษอื่นร่วมอยู่ด้วย โดยไม่ได้เป็นพียงพอร์ตอินพุตเอาท์พุตเพียงอย่างเดียว ดังแสดงในรูป 3-18 แสดงไดอะแกรมการทำงานพื้นฐานของขาพอร์ตอินพุตเอาท์พุตใน dsPIC30F2010 ซึ่งแบ่งการทำงานเป็น 2 ส่วนคือ โมดูลพอร์ต (Port Module) และเซลล์อินพุตเอาต์พุต (I/O Cell) โดยโมดูลพอร์ตจะใช้ในการกำหนดทิศทางของสัญญาณและเป็นที่พักข้อมูลชั่วคราวก่อนรับเข้าหรือส่งออกส่วนเซลล์อินพุตเป็นส่วนที่เชื่อมต่อกับขาพอร์ตจริงและบรรจุวงจรบัฟเฟอร์เพื่อจัดการสัญญาณที่เข้าออกขาพอร์ตให้มีเสถียรภาพ ทั้งนี้ dsPIC30F2010 มีพอร์ตให้ใช้งานมากถึง 5 พอร์ต รวม 20 ขา โดยแต่ละขาสามารถกำหนดให้เป็นอินพุตหรือเอาท์พุตได้ดังนี้

ตารางที่ 3-5 แสดงชื่อและจำนวนขาของแต่ละพอร์ตที่สามารถใช้งานได้

ชื่อพอร์ต	จำนวนขาของแต่ละพอร์ตที่ใช้งานได้	ขาของแต่ละพอร์ตที่ใช้งานได้
พอร์ต B	6 ขา	RB0 – RB5
พอร์ต C	3 ขา	RC13 – RC15
พอร์ต D	2 ขา	RD0 – RD1
พอร์ต E	7 ขา	RE0- RE5 และ RE8
พอร์ต F	2 ขา	RF2 และ RF3



ภาพประกอบ 3-19 แสดงโครงสร้างของขาพอร์ตอินพุตเอาท์พุตของไมโครคอนโทรลเลอร์ dsPIC30F2010

รีจิสเตอร์ควบคุมขาพอร์ตอินพุตเอาท์พุต

ทุก ๆขาพอร์ตอินพุตของ dsPIC30F2010 มีรีจิสเตอร์ในการควบคุมการทำงานโดยตรง 3 ตัว คือ

รีจิสเตอร์ TRISx

เป็นรีจิสเตอร์กำหนดทิศทางข้อมูลของขาแต่ละพอร์ตให้เป็นอินพุตหรือเอาท์พุตดิจิตอล โดยหากกำหนดเป็น 0 หมายถึงกำหนดให้เป็นเอาท์พุต แต่หากกำหนดเป็น 1 หมายถึงกำหนดให้เป็นอินพุต หากรีเซ็ต ขาพอร์ตดิจิตอลทั้งหมดจะถูกกำหนดเป็นอินพุตก่อน หลังจากนั้นจะเปลี่ยนแปลงตามการกำหนดที่รีจิสเตอร์ TRIS

รีจิสเตอร์ PORTx

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่ปรากฏขึ้นที่ขาพอร์ตดิจิตอล โดยหากเป็นขาพอร์ตอินพุต ข้อมูลที่ขาพอร์ตจะปรากฏที่รีจิสเตอร์ PORT สามารถอ่านได้โดยตรง ถ้ากำหนดเป็นขาพอร์ตเอาท์พุตข้อมูลที่รีจิสเตอร์ PORT จะเป็นข้อมูลที่ส่งออกไปยังอุปกรณ์ภายนอกที่ต่อกับขาพอร์ตจริง โดยสามารถเลือกให้ผ่านรีจิสเตอร์ LAT ก่อนก็ได้ เพื่อช่วยลดข้อผิดพลาดในกรณีที่ใช้ขาพอร์ตเดียวกันเป็นทั้งอินพุตและเอาท์พุต

รีจิสเตอร์ LATx

เป็นรีจิสเตอร์ที่ถูกใช้เก็บข้อมูลที่ต้องการส่งออกไปยังขาพอร์ต เมื่อทำงานเป็นเอาท์พุตทั้งนี้เพื่อช่วยในกรณีที่ขาพอร์ตเป็นทั้งอินพุตและเอาท์พุตในเวลาที่ใกล้เคียงกัน โดยต้องการรักษาข้อมูลของเอาท์พุตเดิมไว้ แม้ว่าจะมีการเปลี่ยนแปลงที่ขาพอร์ตเกิดขึ้นเมื่อเป็นอินพุต ถ้าหากต้องการสลับลอจิกที่เอาท์พุต (Toggle) จะสามารถทำได้โดยนำข้อมูลในรีจิสเตอร์ LAT นี้มากลับค่าลอจิกได้ทันที

3.4.4 ไทเมอร์/เคาน์เตอร์

ไทเมอร์/เคาน์เตอร์หลักใน dsPIC30F2010 มี 3 ตัวคือ ไทเมอร์ 1 ถึงไทเมอร์ 3 โดยแต่ละตัวมีขนาด 16 บิต สำหรับไทเมอร์ 2 และ 3 สามารถทำงานร่วมกันเป็นไทเมอร์ขนาด 32 บิต

คุณสมบัติของไทเมอร์ 1 (T1)

- เป็นรีจิสเตอร์ตัวนับความละเอียด 16 บิต
- ทำงานได้ทั้งแบบซิงโครนัสและอะซิงโครนัสเคาน์เตอร์ทำงานร่วมกับขาอินพุตประจำตัวของไทเมอร์ได้
- มีปรีสเกลเลอร์สำหรับหารความถี่การนับ
- สามารถกำหนดการอินเตอร์รัพต์จากการนับ หรือจากการตรวจพบสัญญาณของขอบขาลงที่ขาอินพุตของไทเมอร์

รีจิสเตอร์ TxCON Type A

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
TON	—	TSIDL	—	—	—	—	—
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—

ภาพประกอบ 3-20 แสดงส่วนประกอบรีจิสเตอร์ TxCON

- **บิต 15 TON**(Timer On Control bit) บิตควบคุมการเปิดให้ไทเมอร์ทำงาน

“0” = หยุดการทำงานของไทเมอร์

“1” = เลือกให้ไทเมอร์เริ่มทำงาน

- **บิต 14** ไม่ใช้งานกำหนดเป็น 0

- **บิต 13 TSIDL** (Stop in IDLE Mode bit) บิตกำหนดให้ไทเมอร์หยุดทำงานในโหมดไอเดิล

“0” = ยังคงให้ไทเมอร์ทำงานต่อไปแม้ว่า ซีพียูจะเข้าสู่โหมดไอเดิลก็ตาม

“1” = หยุดการทำงานของไทเมอร์ทันทีที่ซีพียูเข้าโหมดไอเดิล

- **บิต 12 ถึง 7** ไม่ใช้งานกำหนดเป็น 0

- **บิต 6 TGATE** (Timer Gate Time Accumualtion Eable bit) บิตเอ็นเอเบิลการเปิดเกตเพื่อรับสัญญาณจากภายนอก

“0” = ดิสเอเบิลการทำงานแบบนี้

“1” = เอ็นเอเบิลการทำงานแบบนี้

เมื่อเลือกบิตนี้เป็น 1 บิต TCS (บิต 1 ของรีจิสเตอร์ TxCON) ต้องกำหนดเป็น 0 อ่านค่าบิตนี้เป็น 0 ถ้า บิต TCS เป็น 1

- **บิต 5 และ 4 TCKPS1 และ TCKPS0** (Timer Input Clock Prescale Select bit) บิตเลือกอัตราปรีสเกลเลอร์สัญญาณนาฬิกาของไทเมอร์

“00” = เลือกอัตรา 1:1

“01” = เลือกอัตรา 1 : 8

“10” = เลือกอัตรา 1 : 64

“11” = เลือกอัตรา 1 : 256

- **บิต 3** ไม่ใช้งาน กำหนดเป็น 0

- **บิต 2 TSYNC** (Timer External Clock Input Synchronization Select bit) บิตเลือกการซิงโครไนซ์ของสัญญาณนาฬิกาจากภายนอกกับการทำงานของไทเมอร์

- **บิต 1 TCS** (Timer clock Source Select bit) บิตเลือกแหล่งกำเนิดสัญญาณนาฬิกาของไทเมอร์

“0” = เลือกใช้สัญญาณนาฬิกาภายในความถี่ FOSC/4

“1” = เลือกใช้สัญญาณนาฬิกาภายนอกผ่านทางขา TxCKI

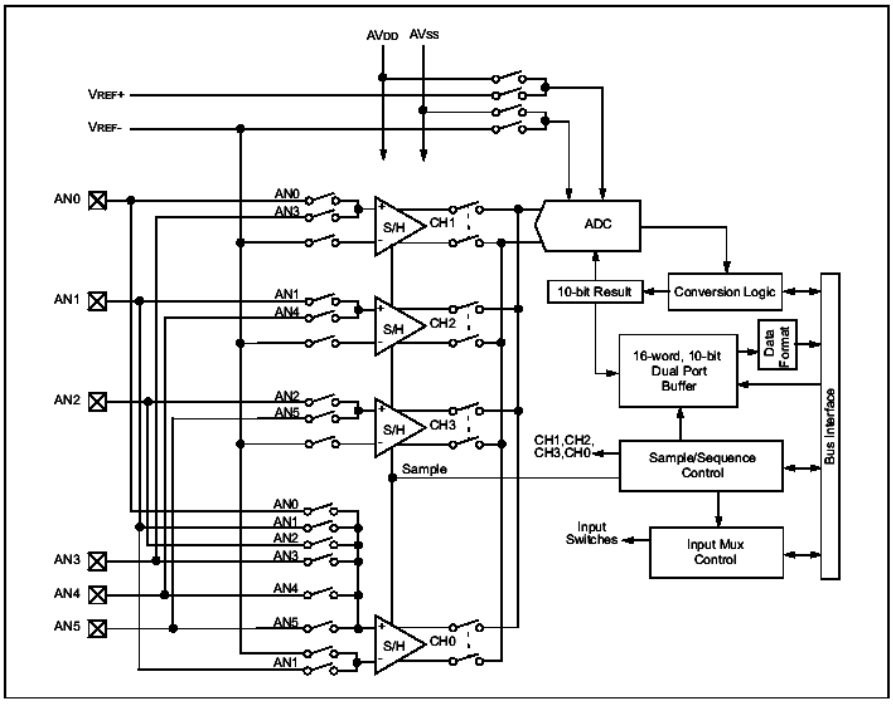
- **บิต 0** ไม่ใช้งานกำหนดเป็น 0

3.4.5 โมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล

ไมโครคอนโทรลเลอร์ dsPIC30F2010 ได้บรรจุโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอลที่มีความละเอียด 10 บิตจำนวน 6 ช่องซึ่งที่ความละเอียดของการแปลงข้อมูล 10 บิต ทำให้ได้ข้อมูลดิจิตอลมีค่าตั้งแต่ 0 ถึง 1023

คุณสมบัติของโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล

- เป็นโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอลที่มีความละเอียด 10 บิต จำนวน 6 ช่อง
- ใช้วิธีการแปลงสัญญาณแบบประมาณค่าหรือ ซักเซสซีฟ แอ็ปพร็อกซิเมชั่น (Successive Approximation)
- มีอัตราเร็วในการสุ่มสัญญาณสูงสุด 500 กิโลแซมเปิลต่อวินาที (ksps) หรือ 500,000 จุดต่อวินาที
- สามารถกำหนดระดับแรงดันอ้างอิงได้ทั้งจากภายในผ่านทางขา AVDD กับ AVSS และภายนอกผ่านทางขา VREF+ กับ VREF-



ภาพประกอบ 3-21 แสดงไดอะแกรมการทำงานของโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล

การทำงานของโมดูล ADC ดังแสดงในภาพประกอบ 3-21 เป็นไดอะแกรมการทำงานของโมดูล ADC ซึ่งมีขาพอร์ตอินพุตอะนาลอกทั้งสิ้น 6 ขาคือ AN0 – AN5 โดยมี 2 ขาที่สามารถใช้รับแรงดันอ้างอิงเพื่อขยายย่านของแรงดันอินพุต ภายในโมดูลมีวงจรสุ่มและเก็บค่าสัญญาณ (Sample and Hold :S/H) จำนวน 4 ชุด โดยทำงานร่วมกับส่วนควบคุมการมัลติเพล็กซ์สัญญาณอินพุต ทำให้สามารถจัดสรรวงจร S/H ให้สามารถรองรับกับสัญญาณอินพุตอะนาลอกทั้ง 6 ช่อง ได้ด้วยความเร็วสูงสุด สัญญาณที่ผ่านจากวงจร S/H จะถูกป้อนเข้าสู่วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลแบบซักเซสซีฟแอ็ปพร็อกซิเมชั่น ขนาด 10 บิต ข้อมูลที่ได้จากการแปลงจะถูกพักไว้ในหน่วยความจำแรม จากนั้นจะได้รับการจัดรูปแบบตามที่ผู้พัฒนาโปรแกรมกำหนด

รีจิสเตอร์หลักที่ใช้ในโมดูล ADC

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

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
ADON	—	ADSIDL	—	—	—	FORM 1	FORM 0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SSRC 2	SSRC 1	SSRC0	—	SIMSAM	ASAM	SAMP	DONE

ภาพประกอบ 3-22 แสดงส่วนประกอบของรีจิสเตอร์ ADCON1

- **บิต 15 ADON** (A/D Operating Mode bit) บิตเลือกให้โมดูล ADC ทำงาน

  “1” = เลือกให้โมดูล ADC ทำงาน

  “0” = ปิดการทำงานของดูล ADC

- **บิต 14** ไม่ใช้งานกำหนดเป็น 0

- **บิต 13 ADSIDL** (Stop in IDLE Mode bit) บิตกำหนดให้โมดูล ADC หยุดทำงานในโหมดไอเดิล

  “0” = ยังคงให้โมดูล ADC ทำงานต่อไปแม้ว่า ซีพียูจะเข้าสู่โหมดไอเดิลก็ตาม

  “1” = หยุดการทำงานของโมดูล ADC ทันทีที่ซีพียูเข้าสู่โหมดไอเดิล

- **บิต 12 ถึง 10** ไม่ใช้งาน กำหนดเป็น 0

- **บิต 9 และ 8 FORM1 และ FORM0** (Data Output Format bits) บิตเลือกรูปแบบข้อมูลเอาท์พุต

  “00” = ข้อมูลเป็นเลขจำนวนเต็มแบบไม่คิดเครื่องหมาย

  “01” = ข้อมูลเป็นเลขจำนวนเต็มแบบคิดเครื่องหมาย

  “10” = ข้อมูลเป็นเลขทศนิยมแบบไม่คิดเครื่องหมาย

“11” = ข้อมูลเป็นเลขทศนิยมแบบคิดเครื่องหมาย

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

“000” = เลือกให้กระตุ้นเมื่อเกิดการเคลียร์บิต SAMP

“001” = เลือกให้กระตุ้นเมื่อเกิดการเปลี่ยนแปลงสัญญาณที่ขา INT0

“010” = เลือกให้กระตุ้นเมื่อเกิดการเปรียบเทียบข้อมูลในไทเมอร์ 3 เสร็จสิ้น

“011” = เลือกให้กระตุ้นเมื่อเสร็จสิ้นการสุ่มสัญญาณจากโมดูล PWM ควบคุมมอเตอร์

“100” = สำรองไว้

“101” = สำรองไว้

“110” = สำรองไว้

“111” = เลือกให้กระตุ้นเมื่อตัวนับค่าภายในเสร็จสิ้นการสุ่มสัญญาณ และเริ่มต้นการแปลงสัญญาณ เป็นการกำหนดให้แปลงสัญญาณโดยอัตโนมัติ

- **บิต 4** ไม่ใช้งานกำหนดเป็น 0

- **บิต 3 SIMSAM** (Simultaneous Sample Select bit) บิตเลือกการสุ่มสัญญาณแบบทันทีทันใด จะใช้ก็ต่อเมื่อบิต CHPS = “01” , “10”, หรือ “11”

“0” = สุ่มสัญญาณเรียงลำดับตามหมายเลขของช่องสัญญาณ

“1” = เลือกสุ่มสัญญาณจากวงจร S/H ช่อง CH0, CH1, CH2, CH3 ทันทีทันใด

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

“0” = เริ่มสุ่มสัญญาณเมื่อบิต SAMP ถูกเซ็ตเป็น 1

“1” = เริ่มสุ่มสัญญาณทันทีที่การแปลงสัญญาณครั้งล่าสุดเสร็จสิ้นลง ทำให้บิต SAMP เซ็ตอัตโนมัติ

- **บิต 1 SAMP** (A/D Sample Enable bit) บิตเอ็นเอเบิลการสุ่มสัญญาณของโมดูล ADC

“0” = เลือกให้พักการทำงานของวงจร S/H หรือดิสเอเบิลการสุ่มสัญญาณ

“1” = เลือกให้วงจร S/H อย่างน้อยหนึ่งวงจรทำการสุ่มสัญญาณหรือ เอ็นเอ – เบิลให้เกิดการสุ่มสัญญาณ

- **บิต 0 DONE** (A/D Conversion Status bit) บิตแสดงสถานะการแปลงสัญญาณอะนาลอกเป็นดิจิตอล

“0” = การแปลงสัญญาณยังไม่เสร็จสิ้น

“1” = การแปลงสัญญาณเสร็จสิ้น

บิตนี้สามารถเคลียร์ได้ด้วยกระบวนการทางซอฟต์แวร์ หรือเมื่อเริ่มต้นการแปลงสัญญาณในรอบใหม่ นอกจากนั้นการเคลียร์บิตนี้จะไม่กระทบต่อการทำงานของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลแต่อย่างใด

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

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
VCFG2	VCFG1	VCFG0	—	—	CSCNA	CHPS1	CHPS0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS

ภาพประกอบ 3-23 แสดงส่วนประกอบของรีจิสเตอร์ ADCON2

- **บิต 15 ถึง 13 VCFG2 ถึง VCFG0** (Voltage Reference Configuration bits) บิตกำหนดแรงดันอ้างอิงสำหรับใช้ในโมดูล ADC

ค่าข้อมูล	แรงดันอ้างอิงด้านสูง V_{REFH}	แรงดันอ้างอิงด้านต่ำ V_{REFL}
“000”	จากขา AVDD (ปกติมีค่า +5 V)	จากขา AVSS (ปกติต่อกราวด์)
“001”	จากภายนอกผ่านขา VREF+	จากขา AVSS (ปกติต่อกราวด์)
“010”	จากขา AVDD (ปกติมีค่า +5 V)	จากภายนอกผ่านขา VREF-
“011”	จากภายนอกผ่านขา VREF+	จากภายนอกผ่านขา VREF-
“1xx”	จากขา AVDD (ปกติมีค่า +5 V)	จากขา AVSS (ปกติต่อกราวด์)

- **บิต 12 และ 11** ไม่ใช้งานกำหนดเป็น 0
- **บิต 10 CSCNA** (Scan Input Selections for CH0+ S/H Input for MUX A Input Multiplexer Setting bit) บิตเลือกการแสดงช่องสัญญาณของวงจร S/H ผ่านมัลติเพล็กเซอร์ A
  - “0” = ไม่มีการสแกนอินพุต
  - “1” = กำหนดให้สแกนอินพุต
- **บิต 9 และ 8 CHPS1 และ CHPS0** (Select Channels Utilized bits) บิตเลือกกลุ่มของช่องสัญญาณผ่านทางวงจร S/H
  - “00” = ต้องการแปลงวงจร S/H ช่อง 0
  - “01” = ต้องการแปลงวงจร S/H ช่อง CH0 และ CH1
  - “1x” = ต้องการแปลงวงจร S/H ช่อง CH0, CH1, CH2 และ CH3
- **บิต 7 BUFS** (Buffer Fill Status bit) บิตแสดงสถานะบัฟเฟอร์ จะมีการแสดงผลเกิดขึ้นเมื่อบิต BUFM ถูกเซตเป็น 1 (รีจิสเตอร์ ADRES แบ่งเป็น 2 ส่วน ส่วนละ 8 เวิร์ด)
  - “0” = แจ้งว่าขณะนี้โมดูล ADC กำลังเขียนข้อมูลลงในบัฟเฟอร์ที่ตำแหน่ง 0x0 – 0x7

“1” = แจ้งว่าขณะนี้โมดูล ADC กำลังเขียนข้อมูลลงในบัฟเฟอร์ที่ตำแหน่ง 0x8 – 0xF

- **บิต 6** ไม่ใช้งาน กำหนดเป็น 0

- **บิต 5 ถึง 2 SMPI3 ถึง SMPI0** (Sample/Convert Sequences Per Interrupt Seclection bits) บิตเลือกการเกิดอินเตอร์รัฟต์ในกระบวนการสุ่มและแปลงสัญญาณในโมดูล ADC

- **บิต 1 BUFM** (Buffer Mode Selection bit) บิตเลือกโหมดของบัฟเฟอร์

“0” = กำหนดให้บัฟเฟอร์มีความจุ 16 เวิร์ด มีชื่อเป็น ADCBUF0 ถึง ADCBUF15

“1” = แบ่งบัฟเฟอร์เป็น 2 ส่วน ส่วนละ 8 เวิร์ด คือรีจิสเตอร์ ADCBUF8 ถึง 15 และรีจิสเตอร์ ADCBUF7 ถึง ADCBUF0

- **บิต 0 ALTS** (Alternate Input Sample Mode Select bit) บิตเลือกโหมดการทำงานของอินพุตมัลติเพล็กเซอร์

“1” = เลือกใช้อินพุต A สำหรับการสุ่มสัญญาณครั้งแรก จากนั้นสลับกันระหว่างอินพุต B

“0” = เลือกใช้อินพุต A ตลอดการทำงาน

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

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
—	—	—	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
ADRC	—	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0

ภาพประกอบ 3-24 แสดงส่วนประกอบของรีจิสเตอร์ ADCON3

- **บิต 15 ถึง 13** ไม่ใช้งานกำหนดเป็น 0

- **บิต 12 ถึง 8 SAMC4** (Auto Sample Time bits) บิตเลือกค่าเวลาในการสุ่มสัญญาณอัตโนมัติ

“00000” = $0T_{AD}$

“00000” = $1T_{AD}$

“00001” = $2T_{AD}$

.....

“11111” = $31T_{AD}$

- **บิต 7 ADRC** (A/D Conversion Clock Source bit) บิตเลือกแหล่งกำเนิดสัญญาณ

“0” = ใช้จากสัญญาณนาฬิกาหลักของระบบ

“1” = ใช้จากวงจร RC ภายในโมดูล ADC

- **บิต 6** ไม่ใช้งานกำหนดเป็น 0

- **บิต 5 ถึง 0 ADCS5 ถึง ADCSS0** (A/D Conversion Clock Select bits) บิตเลือกค่าเวลาที่ใช้ในการแปลงสัญญาณ

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

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
CH123NB1	CH123NB0	CH123SB	CH0NB	CH0SB 3	CH0SB 2	CH0SB 1	CH0SB 0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CH123NA1	CH123NA0	CH123SA	CH0NA	CH0SA 3	CH0SA 2	CH0SA 1	CH0SA 0

ภาพประกอบ 3-25 แสดงส่วนประกอบของรีจิสเตอร์ ADCHHS

- **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” = อินพุตลบของCH1 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 13 CH123SB** (Channel 1,2,3 Positive Input Select for MUX B Multiplexer Setting bit) บิตเลือกช่องอินพุตของวงจร 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

- **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 A 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" = อินพุตบวกของ CH1 คือ AN3 , อินพุตบวกของ CH2 คือ AN4, อินพุตบวกของ CH3 คือ 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)

"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

.....

“1110” = อินพุตบวกของ CH0 ต่อกับขา AN14

“1111” = อินพุตบวกของ CH0 ต่อกับขา AN15

รีจิสเตอร์กำหนดค่าทางฮาร์ดแวร์ของพอร์ตอะนาลอกอินพุต ADPCFG (A/D Port Configuration Register)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
—	—	—	—	—	—	—	—
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0

ภาพประกอบ 3-26 แสดงส่วนประกอบของรีจิสเตอร์ ADPCFG

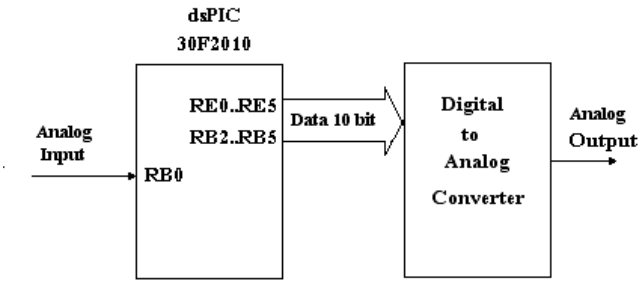
รีจิสเตอร์ตัวนี้สามารถรองรับการกำหนดค่าของพอร์ตอินพุตอะนาลอกได้ถึง 16 ช่อง แต่ในเบอร์ ds30F2010 สามารถกำหนดได้เพียง 6 ช่อง โดยการกำหนดช่องนี้สามารถทำได้อย่างอิสระ โดยหากบิตใดกำหนดเป็น

“0” = กำหนดให้อินพุตอะนาลอกทำงานในโหมดอะนาลอก

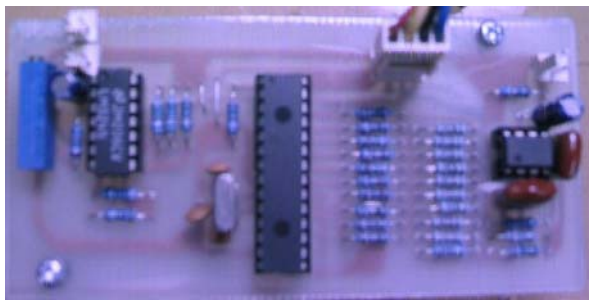
“1” = กำหนดให้อินพุตอะนาลอกทำงานในโหมดดิจิตอล

3.5 การนำไมโครคอนโทรลเลอร์ dsPIC30F2010 ประยุกต์เป็นโครงข่ายประสาท ADALINE

dsPIC30F2010 เป็นไมโครคอนโทรลเลอร์ที่ความสามารถและความเร็วด้านการประมวลผลด้านดิจิตอลสูง ซึ่งสามารถประมวลผลได้ 30 ล้านคำสั่งต่อวินาที จึงทำให้สามารถแปลงและรับค่าสัญญาณอะนาลอกเป็นดิจิตอล ประมวลผลตามอัลกอลิทึมของโครงข่าย ADALINE และให้เอาท์พุตได้ภายใน 1 มิลลิวินาทีได้ ทำให้สามารถกำหนดอัตราสุ่มสัญญาณ 1000 ข้อมูลต่อวินาทีได้ ดังนั้น จึงสามารถนำฟังก์ชั่นการเลื่อนข้อมูลและฟังก์ชันโครงข่าย ADALINE มารวมอยู่ในไมโครคอนโทรลเลอร์เพียงตัวเดียว ทำหน้าที่เป็น Adaptive Filter เพื่อกรองหรือกำจัดสัญญาณรบกวนออกจากสัญญาณไฟฟ้าของกล้ามเนื้อลายได้ โดยกำหนดให้ขา RB0 รับสัญญาณอินพุตที่เป็น อะนาลอก และให้ ขา RE0 – RE5, RB2 – RB5 เป็นเอาท์พุตของโครงข่าย ซึ่งเป็นค่าดิจิตอลขนาด 10 บิต เพื่อส่งต่อไปยังวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกขนาด 10 บิตต่อไปดังแสดงในภาพประกอบ 3-27 (รายละเอียดของวงจรให้ดูในภาคผนวก ผก.4)



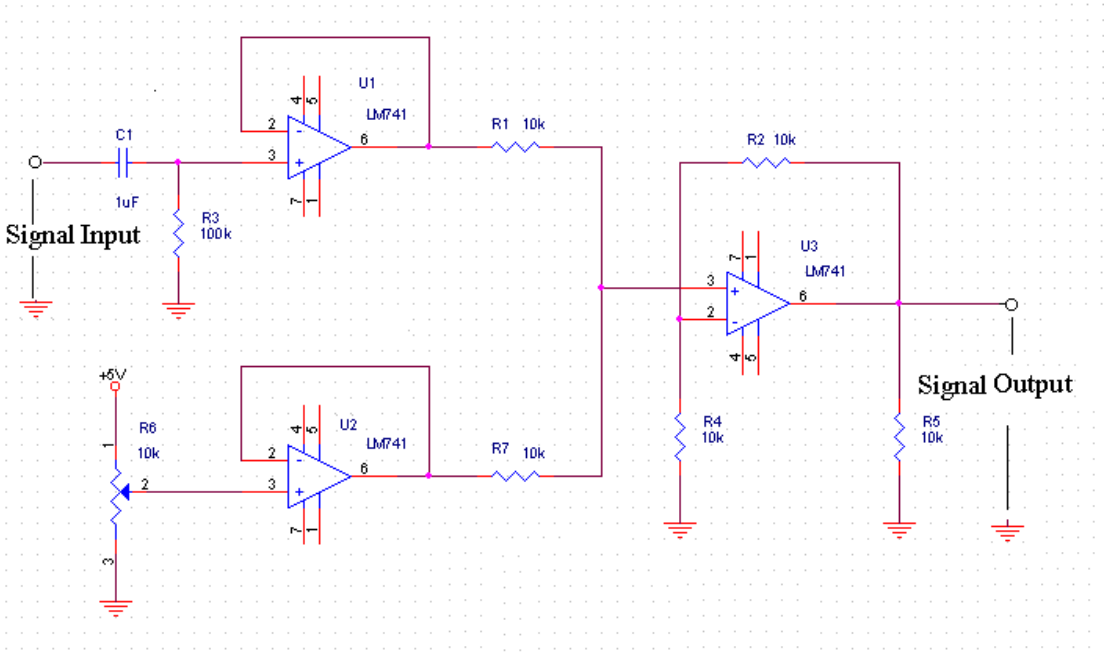
ภาพประกอบ 3-27 แสดงบล็อกไดอะแกรมการประยุกต์ใช้ไมโครคอนโทรลเลอร์ dsPIC30F2010 เป็นโครงข่าย ADALINE



ภาพประกอบ 3-28 แสดงวงจรไมโครคอนโทรลเลอร์ dsPIC30F2010 ที่ใช้เป็นโครงข่าย ADALINE

3.6 วงจรยกระดับแรงดันและผสมสัญญาณ (DC Offsetting and Signal)

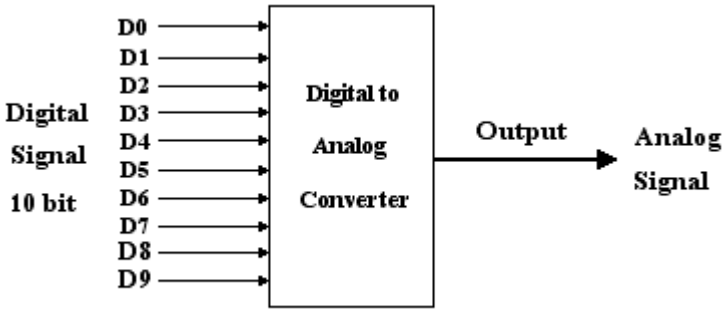
เนื่องจากวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลของไมโครคอนโทรลเลอร์ทั้งของ AVR ATmega32 และ dsPIC30F2010 ไม่สามารถแปลงสัญญาณระดับแรงดันลบได้ ซึ่งทำให้ค่าของสัญญาณที่เป็นค่าลบจะถูกแปลงเป็นค่าศูนย์ ทำให้ค่าที่ป้อนให้กับอินพุตของโครงข่าย ADALINE ไม่ถูกต้อง จึงจำเป็นต้องทำการยกระดับแรงดันไฟฟ้าหรือยกระดับสัญญาณขึ้นไปยังค่าบวก เพื่อให้วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลสามารถรับค่าของสัญญาณได้ทุก ๆค่าของสัญญาณ โดยทำการยกระดับแรงดันขึ้นไป 2.5 โวลท์ ด้วยการใช้ Opamp จำนวน 3 ตัวต่อเป็นวงจรยกระดับแรงดันของสัญญาณ ดังแสดงในภาพประกอบ 3-29 ซึ่งสองตัวแรก จะทำหน้าที่เป็นวงจรตามแรงดัน (Voltage Follow) เพื่อใช้เป็นบัฟเฟอร์ ก่อนที่จะนำสัญญาณไฟฟ้าของกล้ามเนื้อลายและแรงดันไฟฟ้ากระแสตรง 2.5 โวลท์มารวมกันที่ไอซีตัวที่ 3 โดยทำหน้าที่รวมสัญญาณอินพุตทั้งสองและทำหน้าที่ขยายแบบนอนอินเวอร์ติ้ง (Non Iverting Amplifier) ซึ่งมีเกณฑ์ขยายเพียง 1 เท่า สัญญาณไฟฟ้าของกล้ามเนื้อลายจะถูกยกระดับแรงดันขึ้นไป 2.5 โวลท์ตามค่าแรงดันไฟฟ้ากระแสตรง ซึ่งทำให้ระดับสัญญาณที่เคยเป็นค่าลบเปลี่ยนเป็นระดับแรงดันค่าบวก แล้วเอาท์พุตที่ได้จะส่งต่อไปยังวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด 10 บิตของไมโครคอนโทรลเลอร์ต่อไป



ภาพประกอบ 3-29 แสดงวงจรยกระดับแรงดันและผสมสัญญาณไฟฟ้า

3.7 วงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก (Ditital to Analog Converter)

เอาท์พุตของไมโครคอนโทรลเลอร์ ส่วนใหญ่เป็นเอาท์พุตแบบดิจิตอล ซึ่งแต่ละขาจะมีลอจิกเป็น 0 และ 1 หรือระดับแรงดัน 0 และ 5 โวลท์ ไม่สามารถให้เอาท์พุตเป็นสัญญาณอะนาลอกได้ ดังนั้นจึงต้องมีวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก โดยวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกจะรับสัญญาณอินพุตดิจิตอลขนาด 10 บิตมาจากเอาท์พุตของไมโครคอนโทรลเลอร์ ซึ่งเป็นเอาท์พุตของโครงข่าย ADALINE มาทำการแปลงเป็นสัญญาณอะนาลอกดังแสดงในภาพประกอบ 3-30

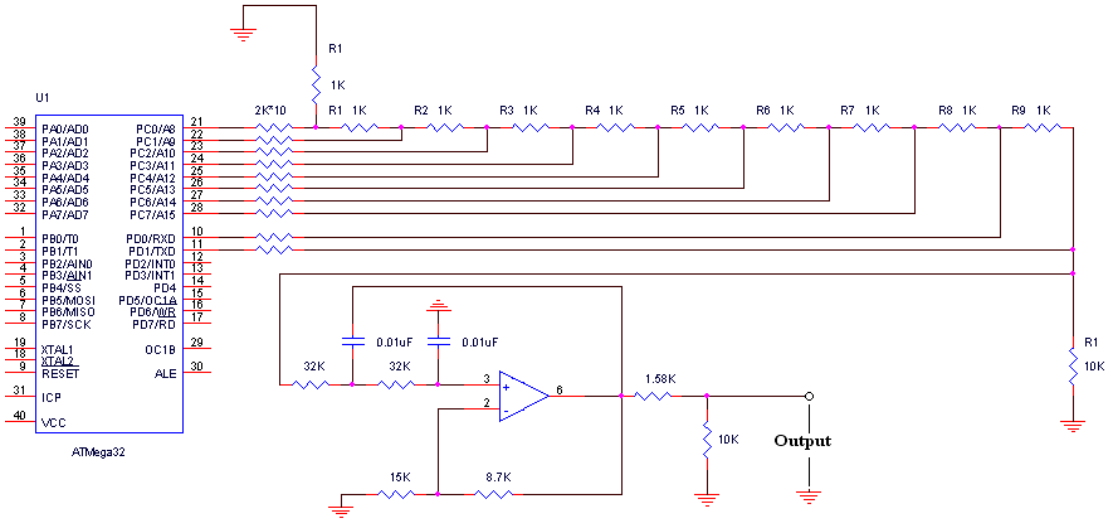


ภาพประกอบ 3-30 แสดงบล็อกไดอะแกรมการแปลงสัญญาณดิจิตอลเป็นอะนาลอก ขนาด 10 บิต

วงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกได้ใช้วงจร R-2R แลดเดอร์ จำนวนอินพุต 10 บิตดังภาพประกอบ 3-31 ซึ่งจะมีความสามารถแยกระดับสัญญาณ (Resolution) ได้ถึง 1024 ระดับ แต่ละระดับจะมีแรงดันแตกต่างกัน 0.0049 โวลท์ (ประมาณ 4.9 mV) เนื่องจากเลือกใช้แรงดันอ้างอิง (Voltage Reference : V_{REF}) 5 โวลท์ โดยสมการคำนวณค่าแรงดันอะนาลอกเอาท์พุตดังสมการ 3-1

$$V_{Out} = \frac{V_{REF}}{1024} \times B \qquad (3\text{-}1)$$

เมื่อ V_{Out} คือค่าแรงดันที่ได้จากเอาท์พุต, V_{REF} คือแรงดันอ้างอิงของวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก และ B คือ ค่าเลขฐาน 10 ของดิจิตอลอินพุตขนาด 10 บิต ซึ่งค่าแรงดันเอาท์พุตต่ำสุดของวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก คือ 0 โวลท์ หรือเป็นกราวด์ และค่าแรงดันเอาท์พุตสูงสุดของวงจรคือ 4.995 โวลท์



ภาพประกอบ 3-31 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกชนิด R2R แลดเดอร์และวงจรกรองสัญญาณความถี่ต่ำ

จากภาพประกอบ 3-31 เอาท์พุตจากวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกชนิด R2R แลดเดอร์ ได้ถูกป้อนเข้าสู่วงจรกรองสัญญาณความถี่ต่ำ (Low Pass Filter) แบบบัตเตอร์เวิร์ท ซึ่งมีอัตราลดทอนสัญญาณ 40 dB ต่อดีเคด ที่ความถี่คัตออฟ 500 Hz เพื่อป้องกันความถี่สูงออกไปยังเอาท์พุต และทำหน้าที่เป็นบัฟเฟอร์เอาท์พุตของวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกด้วย

การทดสอบวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก

เป็นการทดสอบการทำงานของวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก โดยทำการเพิ่มค่าด้านอินพุตของวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก ขึ้นครั้งละ 10 แล้วทำการวัดและบันทึกระดับแรงดันเอาท์พุตที่ได้ แล้วทำการเปรียบเทียบกับค่าที่ได้จากการคำนวณ และคำนวณหาอัตราค่าความแตกต่างดังแสดงในตารางที่ 3-6 ซึ่งส่วนใหญ่อัตราความแตกต่างไม่เกิน 0.5 เปอร์เซ็นต์ แล้วนำค่าทั้งสองมาพล็อตกราฟเปรียบเทียบกัน แสดงดังภาพประกอบ 3-32 ซึ่งสมการคำนวณอัตราความแตกต่างของสัญญาณที่วัดได้กับสัญญาณที่คำนวณได้ดังแสดงในสมการ 3-2

$$\text{อัตราความแตกต่าง} = \frac{V_{\text{คำนวณ}} - V_{\text{วัดได้}}}{V_{\text{คำนวณ}}} \times 100\% \qquad (3\text{-}2)$$

ตารางที่ 3-6 แสดงผลการบันทึกจากการทดสอบวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก

ระดับค่าฐานสิบ	ผลที่ได้จากเอาท์พุต (โวลท์)	ผลที่ได้จากการคำนวณ (โวลท์)	เปอร์เซ็นต์ความแตกต่าง
0	0	0	-
10	0.049	0.0488	-0.35
20	0.097	0.0977	0.67
30	0.146	0.1465	0.33
40	0.1957	0.1953	-0.2
50	0.2441	0.2441	0.02
60	0.2934	0.2930	-0.15
70	0.3411	0.3418	0.20
80	0.391	0.3906	-0.10
90	0.439	0.4395	0.10
100	0.488	0.4883	0.06
110	0.537	0.5371	0.02
120	0.586	0.5859	-0.01
130	0.633	0.6348	0.28
140	0.682	0.6836	0.23

ตารางที่ 3-6 (ต่อ)

ระดับค่าฐานสิบ	ผลที่ได้จากเอาท์พุต (โวลท์)	ผลที่ได้จากการคำนวณ (โวลท์)	เปอร์เซ็นต์ความแตกต่าง
150	0.731	0.7324	0.19
160	0.78	0.7813	0.16
170	0.829	0.8301	0.13
180	0.878	0.8789	0.10
190	0.926	0.9277	0.19
200	0.975	0.9766	0.16
210	1.024	1.0254	0.14
220	1.074	1.0742	0.02
230	1.121	1.1230	0.18
240	1.171	1.1719	0.07
250	1.219	1.2207	0.14
260	1.266	1.2695	0.28
270	1.314	1.3184	0.33
280	1.364	1.3672	0.23
290	1.412	1.4160	0.28
300	1.462	1.4648	0.19
310	1.51	1.5137	0.24
320	1.559	1.5625	0.22
330	1.607	1.6113	0.27
340	1.657	1.6602	0.19
350	1.705	1.7090	0.23
360	1.755	1.7578	0.16
370	1.803	1.8066	0.20
380	1.852	1.8555	0.19
390	1.899	1.9043	0.28
400	1.949	1.9531	0.21

ตารางที่ 3-6 (ต่อ)

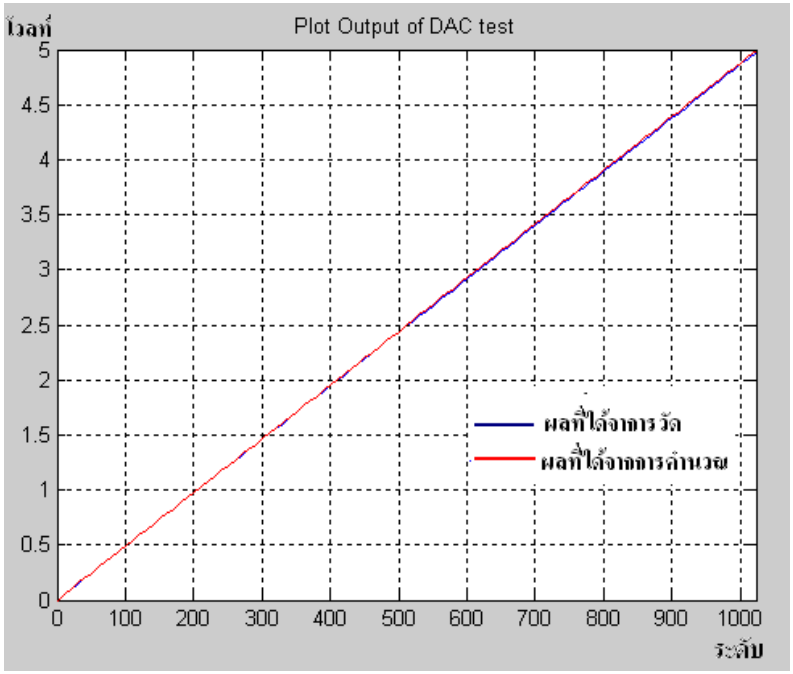
ระดับค่าฐานสิบ	ผลที่ได้จากเอาท์พุต (โวลท์)	ผลที่ได้จากการคำนวณ (โวลท์)	เปอร์เซ็นต์ความแตกต่าง
410	1.997	2.0020	0.25
420	2.046	2.0508	0.23
430	2.095	2.0996	0.22
440	2.145	2.1484	0.16
450	2.192	2.1973	0.24
460	2.242	2.2461	0.18
470	2.29	2.2949	0.21
480	2.34	2.3438	0.16
490	2.388	2.392	0.19
500	2.438	2.4414	0.14
510	2.46	2.4902	0.17
520	2.527	2.5391	0.48
530	2.575	2.5879	0.50
540	2.625	2.6367	0.44
550	2.673	2.6855	0.47
560	2.723	2.7344	0.42
570	2.771	2.7832	0.44
580	2.82	2.8320	0.42
590	2.868	2.8809	0.45
600	2.918	2.9297	0.40
610	2.966	2.9785	0.42
620	3.016	3.0273	0.37
630	3.064	3.0762	0.40
640	3.112	3.1250	0.42
650	3.16	3.1738	0.44
660	3.21	3.2227	0.39

ตารางที่ 3-6 (ต่อ)

ระดับค่าฐานสิบ	ผลที่ได้จากเอาท์พุต (โวลท์)	ผลที่ได้จากการคำนวณ (โวลท์)	เปอร์เซ็นต์ความแตกต่าง
670	3.258	3.2715	0.41
680	3.308	3.3203	0.37
690	3.356	3.3691	0.39
700	3.405	3.4180	0.38
710	3.453	3.4668	0.40
720	3.503	3.5156	0.36
730	3.551	3.5645	0.38
740	3.600	3.6133	0.37
750	3.649	3.6621	0.36
760	3.699	3.7109	0.32
770	3.740	3.7598	0.53
780	3.790	3.8086	0.49
790	3.840	3.8574	0.45
800	3.890	3.9063	0.42
810	3.94	3.9551	0.38
820	3.990	4.0039	0.35
830	4.040	4.0527	0.31
840	4.090	4.1016	0.28
850	4.130	4.1504	0.49
860	4.180	4.1992	0.46
870	4.230	4.2480	0.42
880	4.280	4.2969	0.39
890	4.330	4.3457	0.36
900	4.380	4.3945	0.33
910	4.420	4.4434	0.53
920	4.470	4.4922	0.49

ตารางที่ 3-6 (ต่อ)

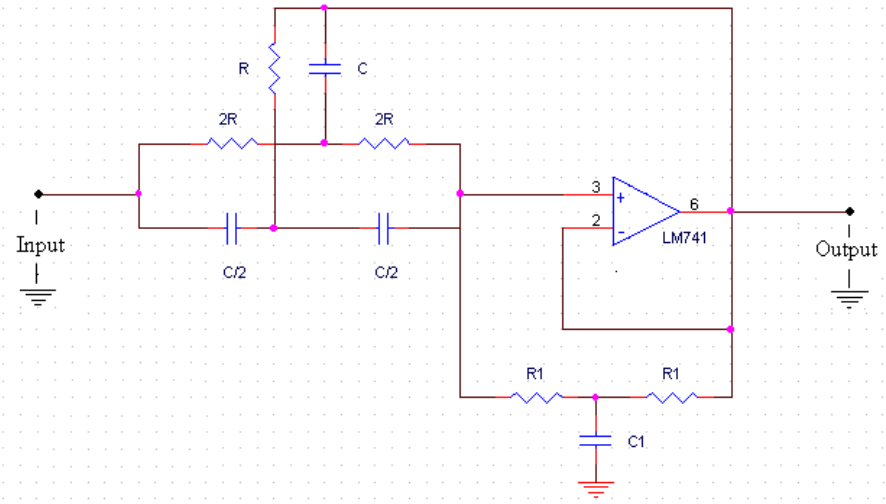
ระดับค่าฐานสิบ	ผลที่ได้จากเอาท์พุต (โวลท์)	ผลที่ได้จากการคำนวณ (โวลท์)	เปอร์เซ็นต์ความแตกต่าง
930	4.520	4.5410	0.46
940	4.570	4.5898	0.43
950	4.620	4.6387	0.40
960	4.670	4.6875	0.37
970	4.720	4.7363	0.34
980	4.770	4.7852	0.32
990	4.820	4.8340	0.29
1000	4.870	4.8340	0.26
1010	4.910	4.9316	0.44
1023	4.980	4.9951	0.30



ภาพประกอบ 3-32 แสดงกราฟเปรียบเทียบระหว่างสัญญาณที่วัดได้จากวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกกับค่าที่คำนวณ

3.8 วงจรกรองความถี่ออกเฉพาะช่วง (Band Reject Filter)

วงจรกรองความถี่ออกเฉพาะช่วงนี้ สร้างขึ้นเพื่อใช้เปรียบเทียบความสามารถในการกำจัดสัญญาณรบกวนที่ความถี่ 50 เฮิรตซ์กับโครงข่าย ADALINE ที่ประยุกต์ด้วยไมโครคอนโทรลเลอร์ ซึ่งวงจรที่นำมาใช้เรียกว่าวงจรนอตช์ฟิลเตอร์ (Notch Filter) มีลักษณะวงจรดังแสดงในภาพประกอบ 3-33 โดยคุณสมบัติของวงจรนอตช์ฟิลเตอร์จะลดทอนสัญญาณรบกวน ที่ความถี่ 50 เฮิรตซ์ ซึ่งเป็นความถี่รบกวนหลักของสัญญาณไฟฟ้าของกล้ามเนื้อลายให้มีขนาดลดลง โดยมีขั้นตอนการออกแบบดังนี้



ภาพประกอบ 3-33 แสดงวงจรนอตช์ฟิลเตอร์ (Notch Filter Circuit)

1) กำหนดความถี่ตรงกลางซึ่งเรียกว่าความถี่เรโซแนนต์ (Wr) ที่ 50 เฮิรตซ์
2) กำหนดค่าควอลิตี้แฟกเตอร์ (Q) เท่ากับ 1
3) เลือกค่าความจุคาปาซิเตอร์ C เท่ากับ 0.1 uF
4) คำนวณหาค่าความต้านทาน R ดังสมการ

$$R = \frac{1}{2\pi \times f \times C} \qquad (3\text{-}3)$$

แทนค่า

$$R = \frac{1}{2\pi \times 50 \times 0.1 \times 10^{-6}} \quad = 31.85 \ \text{K}\Omega \qquad (3\text{-}4)$$

เลือกใช้ค่าความต้านทาน 32 KΩ

5) คำนวณหาค่าความต้านทาน R_1 ดังสมการ

$$R_1 = \frac{4Q}{2\pi \times f \times C} \qquad (3\text{-}5)$$

แทนค่า

$$R_1 = \frac{4}{2\pi \times 50 \times 0.1 \times 10^{-6}} \quad = \quad 127.32 \ \text{K}\Omega \qquad (3\text{-}6)$$

เลือกใช้ค่าความต้านทาน 128 KΩ

6) คำนวณหาค่าความจุคาปาซิเตอร์ C_1 ดังสมการ

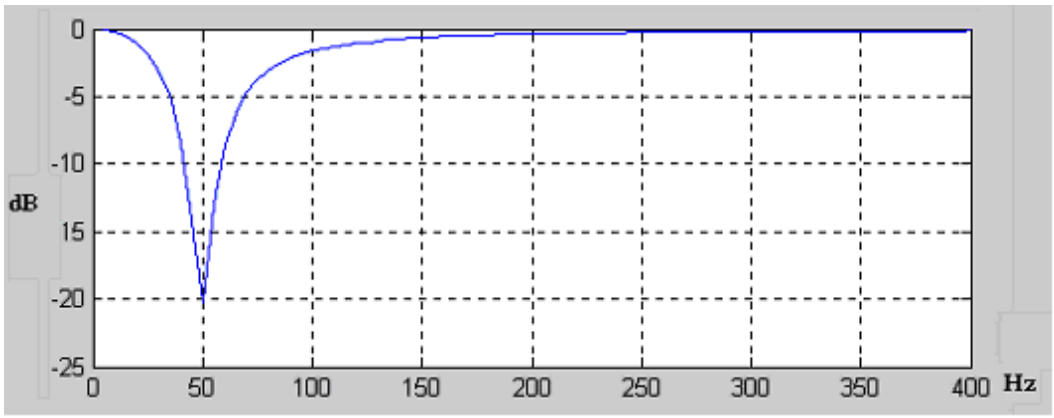
$$C_1 = \frac{C}{2Q} \qquad (3\text{-}7)$$

แทนค่า

$$C_1 = \frac{0.1}{2 \times 1} \quad = 0.05 \ \mu\text{F} \qquad (3\text{-}8)$$

เลือกใช้ค่าคาปาซิเตอร์ 47 nF

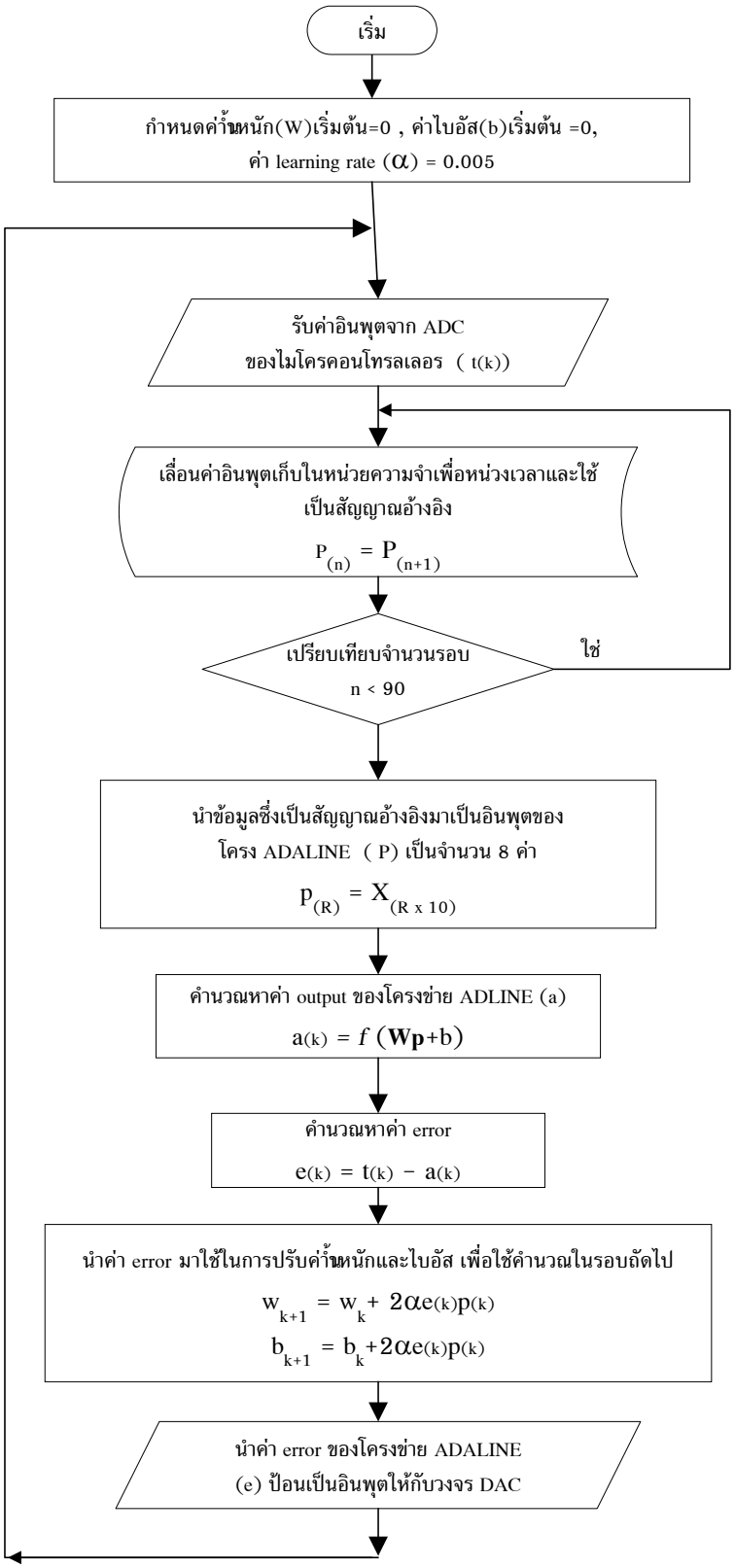
การทดสอบความสามารถในการลดทอนสัญญาณ 50 เฮิรตซ์ของวงจรกรองความถี่นอตช์ฟิลเตอร์นั้น ใช้วิธีป้อนสัญญาณไซน์ตั้งแต่ความถี่ 5 เฮิรตซ์จนถึง 500 เฮิรตซ์ ระดับแรงดัน 1 โวลท์ทางด้านอินพุตของวงจร แล้วใช้โปรแกรม Labview บันทึกเพาเวอร์สเปคตรัมของความถี่ต่าง ๆทั้งของสัญญาณด้านอินพุตและเอาท์พุทของวงจร ซึ่งจะเห็นว่าลักษณะการลดทอนสัญญาณจะลดลงมาคล้ายรูประฆังหงาย โดยเฉพาะที่ความถี่ 50 เฮิร์ตซ์ สามารถลดทอนสัญญาณได้ประมาณ 20 dB ดังแสดงในภาพประกอบ 3-34



ภาพประกอบ 3-34 แสดงลักษณะการลดทอนสัญญาณของวงจรกรองความถี่นอตช์ฟิลเตอร์

3.9 การพัฒนาโปรแกรมเพื่อประยุกต์ใช้โครงข่ายประสาทเป็น Adaptive Filter

การพัฒนาโปรแกรมเพื่อประยุกต์ใช้โครงข่ายประสาทเป็น Adaptive Filter ได้ทำการเขียนโปรแกรมด้วยภาษาซี เพื่อควบคุมให้ไมโครคอนโทรลเลอร์ทำหน้าที่เป็นโครงข่าย ADALINE สำหรับไมโครคอนโทรลเลอร์ AVR ATmega32 ได้ใช้ Code VisionAVR C Compiler เวอร์ชั่น 1.23.3 ของบริษัท ATMEL ซึ่งเป็นคอมไพเลอร์แปลงคำสั่งในไฟล์ที่เขียนด้วยภาษาซีให้เป็นไฟล์ภาษาเครื่องซึ่งเป็นเลขฐานสิบหก (HEX File) เพื่อใช้สำหรับเขียน (Burn) ลงสู่ชิปของไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ทำหน้าที่ลดทอนสัญญาณรบกวน ส่วนไมโครคอนโทรลเลอร์ dsPIC30F2010 ได้ใช้ C30 Compiler ร่วมกับโปรแกรม MPLAP เวอร์ชั่น 7.0 ซึ่งเป็นของบริษัทไมโครชิปเทคโนโลยีทำหน้าที่แปลงคำสั่งในไฟล์ที่เขียนด้วยภาษาซีให้เป็นไฟล์ภาษาเครื่อง โดยโปรแกรมคอมไพเลอร์ของไมโครคอนโทรลเลอร์ทั้งสองตระกูลที่ได้กล่าวถึงสามารถดาวน์โหลดได้จากเวปไซต์ของบริษัทผู้ผลิตดังกล่าว โดยในงานวิจัยนี้การเขียนโปรแกรมได้กำหนดตัวแปรหรือการคำนวณแบบค่าจริง (Floating Point) ซึ่งทำให้ค่าในการคำนวณมีความละเอียดมากกว่าการกำหนดตัวแปรแบบจำนวนเต็ม (Integer หรือ Fixing Point) แต่จะทำให้ความเร็วในการคำนวณตามอัลกอลิทึมของโครงข่าย ADALINE ในแต่ละรอบช้ากว่า อย่างไรก็ตามได้กำหนดให้ความเร็วของการคำนวณเป็น 1000 รอบการคำนวณต่อวินาที โฟลชาร์ทการเขียนโปรแกรมเพื่อควบคุมให้ไมโครคอนโทรลเลอร์ทำหน้าที่เป็นโครงข่าย ADALINE เพื่อลดทอนสัญญาณรบกวนมีลักษณะดังภาพประกอบที่ 3-35



ภาพประกอบ 3-35 โฟลชาร์ทการทำงานของโครงข่าย ADALINE ที่ประยุกต์ใช้เป็น Adaptive Filter ชนิดที่ไม่ใช้สัญญาณอ้างอิงจากภายนอก