	ใบเนื้อหา	หน้าที่ 1
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟlop	

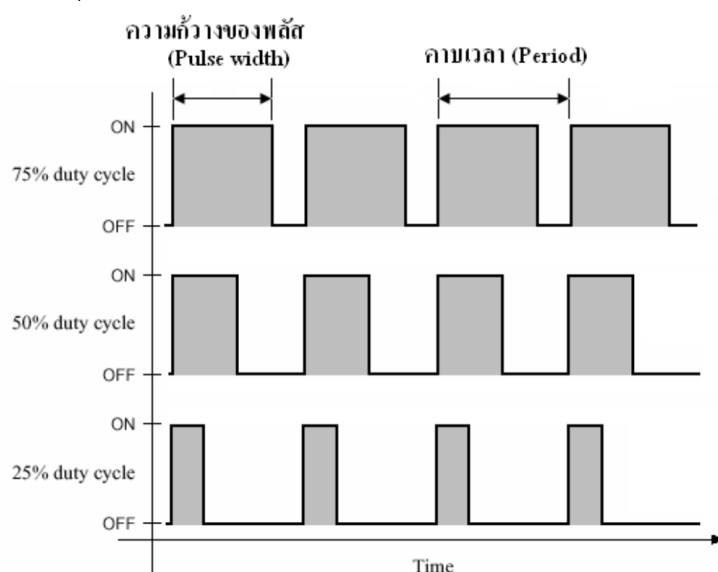
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

หน่วยที่ 3 สัญญาณนาฬิกา และวงจรฟลิปฟlop

การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

1. การสร้างสัญญาณนาฬิกาด้วยโมดูล PWM

PWM หรือ Pulse Width Modulation คือสัญญาณพัลส์ที่มีค่าความถี่คงที่แต่ความกว้างของพัลส์เปลี่ยนแปลงได้ตามความต้องการของผู้ใช้งาน โดยสัญญาณ PWM จะประกอบด้วยองค์ประกอบ 3 ส่วนได้แก่ ขนาดความกว้างของสัญญาณพัลส์ซึ่งเป็นช่วงเวลาของการปรากฏสัญญาณลอจิก '1' (W : Width), คาบเวลาของสัญญาณซึ่งเป็นช่วงเวลาของการปรากฏสัญญาณลอจิก '1' และลอจิก '0' (T : Time Period) และดิวตีไซเคิลมีค่าเท่ากับอัตราส่วนของความกว้างของสัญญาณพัลส์กับคาบเวลาของสัญญาณ ($D : \text{Duty cycle} = W/T$) โดยสัญญาณ PWM จะถูกประยุกต์ใช้งานสำหรับการสร้างสัญญาณนาฬิกาที่ความถี่ต่าง ๆ และสามารถกำหนดค่าของดิวตีไซเคิลได้ หรือประยุกต์ใช้งานสำหรับการควบคุมความเร็วมอเตอร์ เป็นต้น




รูปที่ 1.1 ตัวอย่างสัญญาณ PWM

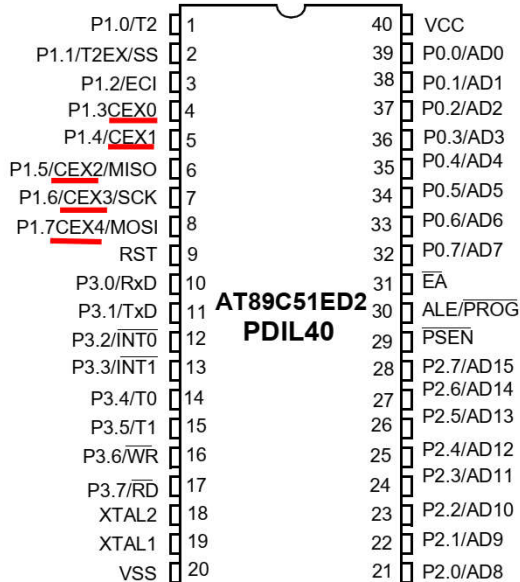
1.1 ลักษณะการทำงานของโมดูล PWM

ลักษณะการทำงานของโมดูล PWM หรือ Pulse Width Modulation ในตัวไมโครคอนโทรลเลอร์ คือโมดูลที่ใช้สำหรับสร้างสัญญาณพัลส์ที่มีค่าความถี่คงที่ แต่ความกว้างของพัลส์สามารถเปลี่ยนแปลงได้ตามความต้องการของผู้ใช้งาน โดยโมดูล PWM ของไมโครคอนโทรลเลอร์ส่วนใหญ่จะเป็นฟังก์ชันย่อยของโมดูลพิเศษในไมโครคอนโทรลเลอร์เบอร์นั้น ๆ ซึ่งเป็นฟังก์ชันย่อยของโมดูลพิเศษเกี่ยวกับการนับ เพื่อให้โมดูลเกี่ยวกับการนับเป็นตัวสร้างฐานเวลาในการกำหนดความถี่ของสัญญาณ PWM และใช้รีจิสเตอร์พิเศษอีกตัวหนึ่งเพื่อทำการเปรียบเทียบค่าของการนับในการกำหนดค่าดิวตีไซเคิลของสัญญาณ PWM

	ใบเนื้อหา		หน้าที่ 2																	
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3																	
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป																			
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์																				
<p>ดังนั้นการกำหนดลักษณะการทำงานของโมดูล PWM ในไมโครคอนโทรลเลอร์แต่ละตระกูลสามารถแบ่งได้ดังนี้</p> <p>1.1.1 การกำหนดลักษณะการทำงานของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2</p> <p>ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2 จะมีโมดูล PWM ที่เป็นฟังก์ชันย่อยของโมดูลพิเศษ PCA (Programmable Counter Array) จำนวน 5 วงจร โดยโมดูล PCA มีฟังก์ชันการทำงานย่อย 3 อย่างได้แก่การ Capture สัญญาณ ,การสร้างสัญญาณ PWM และใช้เป็น Programmable Watch Dog Timer ซึ่งโมดูล PCA จะเชื่อมต่อกับพอร์ต P1 ตั้งแต่ขา P1.2 (ECI) ถึง P1.7 (CEX4) ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ดังรูปที่ 1.2 ซึ่งขาที่สามารถสร้างสัญญาณ PWM ได้คือขา P1.3 – P1.7 (CEX0 – CEX4) โดยสัญญาณ PWM ที่สร้างขึ้นจะมีความถี่เดียวกัน แต่ดิวตี้ไซเคิลแตกต่างกัน และมีรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดลักษณะการทำงานของโมดูล PWM ดังนี้</p> <p>CMOD คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่กำหนดแหล่งกำเนิดสัญญาณนาฬิกาของโมดูล PCA โดยส่วนที่เกี่ยวข้องกับการทำงานของ PWM คือบิต 2 และบิต 1 (CPS1 และ CPS0) ดังนี้</p> <p>ตารางที่ 1.1 การกำหนดค่าบิต CPS1 และ CPS0 เพื่อเลือกแหล่งจ่ายสัญญาณนาฬิกาให้แก่ PCA</p> <table><tr><th>CPS1</th><th>CPS0</th><th>แหล่งจ่ายสัญญาณนาฬิกาของโมดูล PCA</th></tr><tr><td rowspan="2">0</td><td rowspan="2">0</td><td>Internal Clock (F_{CLK})/6 (6 clock Mode)</td></tr><tr><td>Internal Clock (F_{CLK})/12 (12 clock Mode)</td></tr><tr><td rowspan="2">0</td><td rowspan="2">1</td><td>Internal Clock (F_{CLK})/2 (6 clock Mode)</td></tr><tr><td>Internal Clock (F_{CLK})/4 (12 clock Mode)</td></tr><tr><td>1</td><td>0</td><td>Timer 0 Overflow</td></tr><tr><td>1</td><td>1</td><td>External clock at ECI/P1.2 pin</td></tr></table> <p>CCON คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ควบคุมการทำงานของโมดูล PCA โดยส่วนที่เกี่ยวข้องกับการทำงานของ PWM คือบิต 6 บิต CR (PCA Counter Run Control bit) ทำหน้าที่เปิดปิดการทำงานของโมดูล PCA</p> <p>CCAPMn คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่กำหนดลักษณะการทำงานของโมดูล PCA ดังรูปที่ 1.3 โดยไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2 สามารถกำหนดให้โมดูล PCA สร้างสัญญาณ PWM ในรูปแบบ 8-bit PWM คือใน 1 คาบเวลาที่มีความละเอียดของการนับที่ 256 ค่า</p> <p>CCAPnH และ CCAPnL คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าดิวตี้ไซเคิลโดย CCAPnH จะเป็นรีจิสเตอร์สำหรับเก็บค่าคงที่ของการเปรียบเทียบสำหรับการสร้างคาบเวลาของดิวตี้ไซเคิล ส่วนรีจิสเตอร์ CCAPnL จะเป็นรีจิสเตอร์สำหรับการนับ โดยมีลักษณะการทำงานดังรูปที่ 1.4</p>				CPS1	CPS0	แหล่งจ่ายสัญญาณนาฬิกาของโมดูล PCA	0	0	Internal Clock (F_{CLK})/6 (6 clock Mode)	Internal Clock (F_{CLK})/12 (12 clock Mode)	0	1	Internal Clock (F_{CLK})/2 (6 clock Mode)	Internal Clock (F_{CLK})/4 (12 clock Mode)	1	0	Timer 0 Overflow	1	1	External clock at ECI/P1.2 pin
CPS1	CPS0	แหล่งจ่ายสัญญาณนาฬิกาของโมดูล PCA																		
0	0	Internal Clock (F_{CLK})/6 (6 clock Mode)																		
		Internal Clock (F_{CLK})/12 (12 clock Mode)																		
0	1	Internal Clock (F_{CLK})/2 (6 clock Mode)																		
		Internal Clock (F_{CLK})/4 (12 clock Mode)																		
1	0	Timer 0 Overflow																		
1	1	External clock at ECI/P1.2 pin																		

	ใบเนื้อหา		หน้าที่ 3
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		


ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์



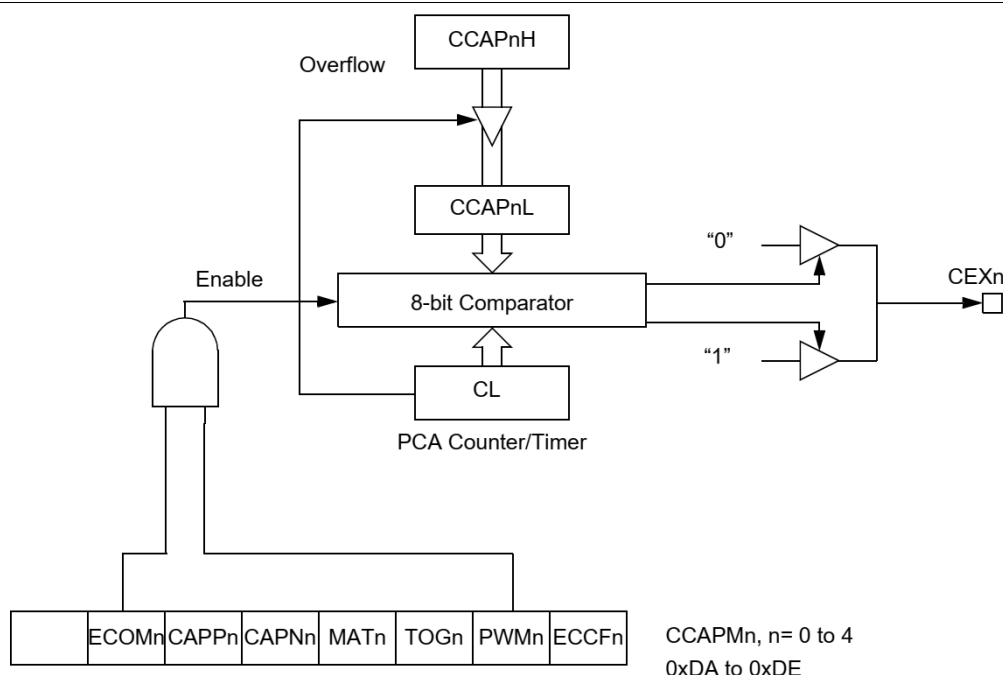
รูปที่ 1.2 แสดงตำแหน่งขาสัญญาณที่สามารถสร้างรูปคลื่น PWM ของไมโครคอนโทรลเลอร์ AT89C51ED2

7	6	5	4	3	2	1	0
-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMm	ECCFn	Module Function
0	0	0	0	0	0	0	No Operation
X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	0	1	0	0	0	X	16-bit capture by a negative trigger on CEXn
X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
1	0	0	1	0	0	X	16-bit Software Timer/Compare mode.
1	0	0	1	1	0	X	16-bit High Speed Output
1	0	0	0	0	1	0	8-bit PWM
1	0	0	1	X	0	X	Watchdog Timer (module 4 only)

รูปที่ 1.3 รูปตารางการกำหนดค่ารีจิสเตอร์ CCAPMn เพื่อโมดูล PCA ให้ทำงานในลักษณะ 8-bit PWM

	ใบเนื้อหา		หน้าที่ 4
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์



รูปที่ 1.4 โครงสร้างการทำงานของขา CEXn ในการสร้างสัญญาณ PWM


1.1.2 การกำหนดลักษณะการทำงานของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล PIC16F เบอร์ PIC16F877 หรือ PIC16F887

ไมโครคอนโทรลเลอร์ PIC16F ขนาด 40 ขาได้เตรียมโมดูล CCP (Capture/Compare/PWM) สำหรับใช้ในการตรวจจับสัญญาณ, เปรียบเทียบข้อมูล และสร้างสัญญาณ PWM (Pulse-width modulation) มาพร้อมใช้งาน โดยจะขอกว่าเฉพาะการใช้งานโมดูล PWM เพียงอย่างเดียว ซึ่งใน PIC16F877 หรือ PIC16F887 จะมีโมดูล PWM ให้ใช้งาน 2 ชุด ได้แก่ โมดูล CCP1 และโมดูล CCP2

โมดูล CCP (Capture/Compare/PWM) สามารถใช้กำเนิดสัญญาณ PWM ที่มีความละเอียดสูงสุดขนาด 10 บิต โดยจะทำงานร่วมกันกับ TIMER2 เมื่อกำหนดค่าเริ่มต้นในการทำงานให้กับโมดูลแล้ว โมดูลจะทำงานต่อไปเองโดยอิสระ และการทำงานของโมดูล PWM ของไมโครคอนโทรลเลอร์ตระกูล PIC16F เบอร์ PIC16F877 หรือ PIC16F887 จะมีรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดลักษณะการทำงานของโมดูล PWM ดังนี้

PR2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าในสูตรเพื่อคำนวณหาคาบเวลาของสัญญาณ PWM ที่สร้างขึ้น โดย $PWM\ Period = (PR2 + 1) \times 4 \times T_{OSC} \times (TMR2\ Prescale\ Value)$

T2CON คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าการนับของ Timer 2 เพื่อสร้างฐานเวลาของความถี่ PWM ที่สร้างขึ้นใช้เฉพาะบิต TMR2ON, T2CKPS1 และ T2CKPS0 โดยบิต TMR2ON คือบิตที่ทำหน้าที่ในการเปิดปิดการทำงานของโมดูล Timer 2 ส่วนบิต T2CKPS1 และ T2CKPS0 ทำหน้าที่ในการกำหนดค่า Prescaler ของโมดูล Timer 2 ดังตารางที่ 1.2

	ใบเนื้อหา	หน้าที่ 5
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟlop	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

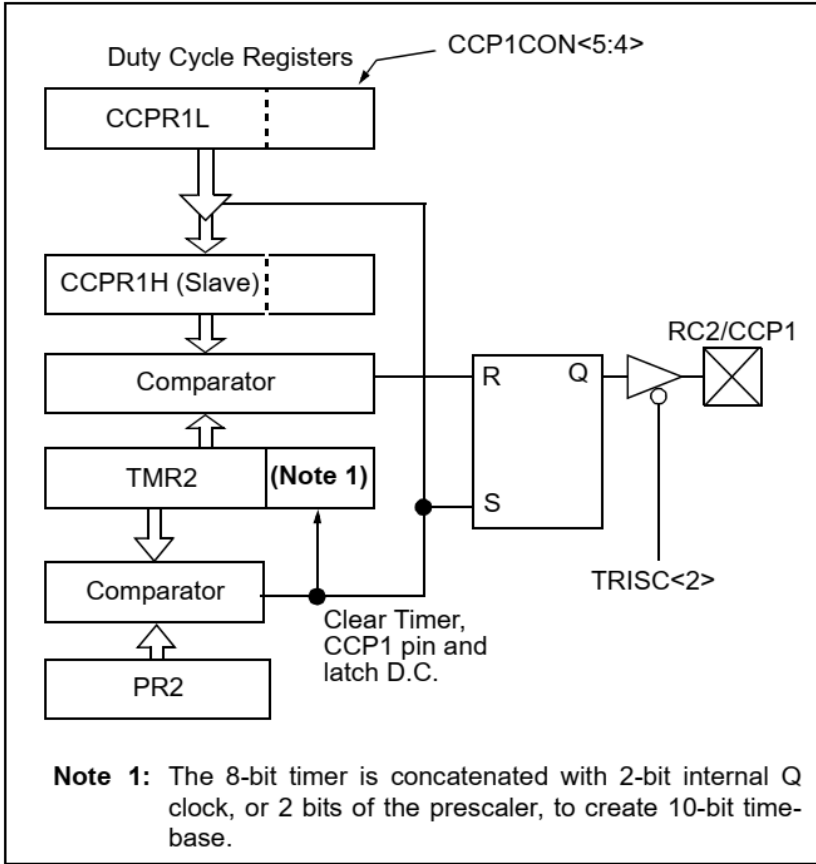
ตารางที่ 1.2 การกำหนดค่าบิต T2CKPS1 และ T2CKPS0 เพื่อกำหนดค่า Prescaler ของโมดูล Timer 2

T2CKPS1	T2CKPS0	ค่า Prescaler ของโมดูล Timer 2
0	0	ค่า Prescaler = 1
0	1	ค่า Prescaler = 4
1	x	ค่า Prescaler = 16

CCPRxL คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าดิวิตไซเกิล 8 บิตบน โดยค่าคาบเวลาของดิวิตไซเกิลหาได้จาก


PWM duty cycle = (CCPRxL:CCPxCON<5:4>) x T_{OSC} x (TMR2 Prescale value)

CCPxCON คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าดิวิตไซเกิล 2 บิตล่าง และกำหนดโหมดการทำงานของโมดูล CCP ซึ่งจะใช้บิต 3 ถึงบิต 0 ของรีจิสเตอร์ CCPxCON

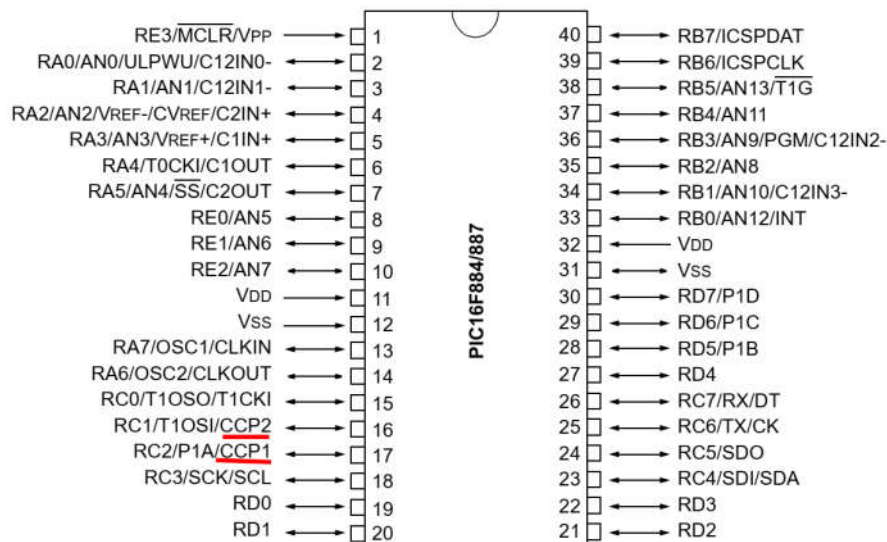


Note 1: The 8-bit timer is concatenated with 2-bit internal Q clock, or 2 bits of the prescaler, to create 10-bit time-base.

รูปที่ 1.5 แสดงโครงสร้างการทำงานของโมดูล CCP ในโหมด PWM

	ใบเนื้อหา		หน้าที่ 6
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์



รูปที่ 1.6 แสดงตำแหน่งขาสัญญาณที่สามารถสร้างรูปคลื่น PWM ของไมโครคอนโทรลเลอร์ PIC16F887

1.1.3 การกำหนดลักษณะการทำงานของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32

ไมโครคอนโทรลเลอร์ ATMEGA32 ขนาด 40 ขาได้เตรียมโมดูล PWM สำหรับสร้างสัญญาณ PWM มาพร้อมใช้งานจำนวน 4 ชุด ได้แก่ โมดูล PWM ที่มีความละเอียดขนาด 8 บิต จาก Timer 0 จำนวน 1 ชุด ออกที่ขาสัญญาณ OC0 , โมดูล PWM ที่มีความละเอียดขนาด 16 บิต จาก Timer 1 จำนวน 2 ชุดออกที่ขาสัญญาณ OC1A และ OC1B และโมดูล PWM ที่มีความละเอียดขนาด 8 บิต จาก Timer 2 จำนวน 1 ชุดออกที่ขาสัญญาณ OC2 โดยโมดูลแต่ละชุดของไมโครคอนโทรลเลอร์ ATMEGA32 มีรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดลักษณะการทำงานดังนี้

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของโมดูล PWM เนื่องจาก Timer 0 ซึ่งจะมีโหมดการทำงานที่เกี่ยวข้องจำนวน 2 โหมดได้แก่ PWM, Phase Correct และ Fast PWM ซึ่งมีรีจิสเตอร์ที่เกี่ยวข้องได้แก่

TCCR0 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดโหมดการทำงานของโมดูล Timer 0 ให้ทำงานในลักษณะต่าง ๆ และกำหนดค่า Prescale ของ Timer 0 โดยรีจิสเตอร์ TCCR0 มีบิตต่างดังรูปที่ 1.7

7	6	5	4	3	2	1	0	
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

รูปที่ 1.7 รูปตำแหน่งบิตต่าง ๆ ของรีจิสเตอร์ TCCR0

ใบเนื้อหา
ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004
ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป

หน้าที่ 7
หน่วยที่ 3

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

รูปที่ 1.8 รูปตารางกำหนดโหมดของ Timer 0 ด้วยรีจิสเตอร์ TCCR0 โดยบิต WGM01 และ WGM00

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)


รูปที่ 1.9 รูปตารางกำหนดการทำงานของขา OC0 ด้วยรีจิสเตอร์ TCCR0 โดยบิต COM01 และ COM00 ในโหมด Fast PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match when up-counting. Set OC0 on compare match when downcounting.
1	1	Set OC0 on compare match when up-counting. Clear OC0 on compare match when downcounting.

รูปที่ 1.10 รูปตารางกำหนดการทำงานของขา OC0 ด้วยรีจิสเตอร์ TCCR0 โดยบิต COM01 และ COM00 ในโหมด Phase Correct PWM Mode

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}/(No\ prescaling)$
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

รูปที่ 1.11 รูปตารางกำหนดแหล่งกำเนิดสัญญาณนาฬิกาของ Timer 0 ด้วยรีจิสเตอร์ TCCR0 โดยบิต CS02 ,CS01 และ CS00

	ใบเนื้อหา		หน้าที่ 8
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

TCNT0 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับคาบเวลาของโมดูล Timer 0 เพื่อสร้างคาบเวลาของสัญญาณ PWM

OCR0 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับเพื่อใช้ในการเปรียบเทียบค่ากับ TCNT0 เพื่อกำหนดค่าตัวชี้ไขเกิลของสัญญาณ PWM

ในโหมด PWM, Phase Correct ของ Timer 0 สามารถคำนวณหาค่าความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnPCPWM} = \frac{f_{clk_I/O}}{N \cdot 510}$$

ในโหมด Fast PWM ของ Timer 0 สามารถคำนวณหาค่าความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnPWM} = \frac{f_{clk_I/O}}{N \cdot 256}$$

เมื่อ N คือค่าของ Prescaler ในรูปที่ 1.11

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของโมดูล PWM เนื่องจาก Timer 1 ซึ่งจะมีโหมดการทำงานที่เกี่ยวข้องจำนวน 3 โหมดได้แก่ Fast PWM , Phase Correct PWM Mode และ Phase and Frequency Correct PWM Mode ซึ่งมีรีจิสเตอร์ที่เกี่ยวข้องได้แก่

TCCR1A คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดโหมดการทำงานของโมดูล Timer 1 ให้ทำงานในลักษณะต่าง ๆ ร่วมกับรีจิสเตอร์ TCCR1B มีบิตต่างดังรูปที่ 1.12


TCCR1B คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดโหมดการทำงานของโมดูล Timer 1 ให้ทำงานในลักษณะต่าง ๆ ร่วมกับรีจิสเตอร์ TCCR1A และกำหนดค่า Prescale ของ Timer 1 โดยรีจิสเตอร์ TCCR1B มีบิตต่างดังรูปที่ 1.13

7	6	5	4	3	2	1	0	
COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
R/W	R/W	R/W	R/W	W	W	R/W	R/W	
0	0	0	0	0	0	0	0	

รูปที่ 1.12 รูปตำแหน่งบิตต่าง ๆ ของรีจิสเตอร์ TCCR1A

7	6	5	4	3	2	1	0	
ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

รูปที่ 1.13 รูปตำแหน่งบิตต่าง ๆ ของรีจิสเตอร์ TCCR1B

	ใบเนื้อหา		หน้าที่ 9
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		


ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM13:0 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on compare match, set OC1A/OC1B at BOTTOM, (non-inverting mode)
1	1	Set OC1A/OC1B on compare match, clear OC1A/OC1B at BOTTOM, (inverting mode)

รูปที่ 1.14 รูปตารางกำหนดการทำงานของขา OC1A, OC1B ด้วยรีจิสเตอร์ TCCR1A โดยบิต COM1A1/COM1B1 และ COM1A0/COM1B0 ในโหมด Fast PWM Mode

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 9 or 14: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM13:0 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on compare match when up-counting. Set OC1A/OC1B on compare match when downcounting.
1	1	Set OC1A/OC1B on compare match when up-counting. Clear OC1A/OC1B on compare match when downcounting.

รูปที่ 1.15 รูปตารางกำหนดการทำงานของขา OC1A, OC1B ด้วยรีจิสเตอร์ TCCR1A โดยบิต COM1A1/COM1B1 และ COM1A0/COM1B0 ในโหมด Phase Correct PWM Mode และ Phase and Frequency Correct PWM Mode

	ใบเนื้อหา		หน้าที่ 10
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		


ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

รูปที่ 1.16 รูปตารางกำหนดโหมดของ Timer 1 ด้วยรีจิสเตอร์ TCCR1A และ TCCR1B โดยบิต WGM13 ,WGM12 ,WGM11 และ WGM10

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{IO} /1 (No prescaling)
0	1	0	clk _{IO} /8 (From prescaler)
0	1	1	clk _{IO} /64 (From prescaler)
1	0	0	clk _{IO} /256 (From prescaler)
1	0	1	clk _{IO} /1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

รูปที่ 1.17 รูปตารางกำหนดแหล่งกำเนิดสัญญาณนาฬิกาของ Timer 1 ด้วยรีจิสเตอร์ TCCR1B โดยบิต CS12 ,CS11 และ CS10

	ใบเนื้อหา		หน้าที่ 11
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

TCNT1H และ TCNT1L คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับคาบเวลาของโมดูล Timer 1 เพื่อสร้างคาบเวลาของสัญญาณ PWM เมื่อรวมกันจะได้ความละเอียดของการนับ 16 บิต

OCR1AH และ OCR1AL คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับเพื่อใช้ในการเปรียบเทียบกับ TCNT1 เพื่อกำหนดค่าดีวตี้ไซเคิลของสัญญาณ PWM เมื่อรวมกันจะได้ความละเอียดของการนับ 16 บิต

OCR1BH และ OCR1BL คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับเพื่อใช้ในการเปรียบเทียบกับ TCNT1 เพื่อกำหนดค่าดีวตี้ไซเคิลของสัญญาณ PWM เมื่อรวมกันจะได้ความละเอียดของการนับ 16 บิต

ในโหมด Fast PWM Mode ของ Timer 1 สามารถคำนวณหาความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnxPWM} = \frac{f_{clk_I/O}}{N \cdot (1 + TOP)}$$

ในโหมด Phase Correct PWM Mode ของ Timer 1 สามารถคำนวณหาความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnxPCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP}$$

ในโหมด Phase and Frequency Correct PWM Mode ของ Timer 1 สามารถคำนวณหาความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnxPFCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP}$$

เมื่อ N คือค่าของ Prescaler ในรูปที่ 1.17

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของโมดูล PWM เนื่องจาก Timer 2 ซึ่งจะมีโหมดการทำงานที่เกี่ยวข้องจำนวน 2 โหมดได้แก่ PWM, Phase Correct และ Fast PWM ซึ่งมีรีจิสเตอร์ที่เกี่ยวข้องได้แก่

TCCR2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดโหมดการทำงานของโมดูล Timer 2 ให้ทำงานในลักษณะต่าง ๆ และกำหนดค่า Prescale ของ Timer 2 โดยรีจิสเตอร์ TCCR2 มีบิตต่างดังรูปที่ 1.18

7	6	5	4	3	2	1	0	
FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

รูปที่ 1.18 รูปตำแหน่งบิตต่าง ๆ ของรีจิสเตอร์ TCCR2

ใบเนื้อหา

หน้าที่ 12

ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004

หน่วยที่ 3

ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟlop

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

Mode	WGM21 (CTC2)	WGM20 (PWM2)	Timer/Counter Mode of Operation	TOP	Update of OCR2	TOV2 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR2	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

รูปที่ 1.19 รูปตารางกำหนดโหมดของ Timer 2 ด้วยรีจิสเตอร์ TCCR2 โดยบิต WGM21 และ WGM20

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on compare match, set OC2 at BOTTOM, (non-inverting mode)
1	1	Set OC2 on compare match, clear OC2 at BOTTOM, (inverting mode)


รูปที่ 1.20 รูปตารางกำหนดการทำงานของขา OC2 ด้วยรีจิสเตอร์ TCCR2 โดยบิต COM21 และ COM20 ในโหมด Fast PWM Mode

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on compare match when up-counting. Set OC2 on compare match when downcounting.
1	1	Set OC2 on compare match when up-counting. Clear OC2 on compare match when downcounting.

รูปที่ 1.21 รูปตารางกำหนดการทำงานของขา OC2 ด้วยรีจิสเตอร์ TCCR2 โดยบิต COM21 และ COM20 ในโหมด Phase Correct PWM Mode

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{T2S} /(No prescaling)
0	1	0	clk _{T2S} /8 (From prescaler)
0	1	1	clk _{T2S} /32 (From prescaler)
1	0	0	clk _{T2S} /64 (From prescaler)
1	0	1	clk _{T2S} /128 (From prescaler)
1	1	0	clk _{T2S} /256 (From prescaler)
1	1	1	clk _{T2S} /1024 (From prescaler)

รูปที่ 1.22 รูปตารางกำหนดแหล่งกำเนิดสัญญาณนาฬิกาของ Timer 2 ด้วยรีจิสเตอร์ TCCR2 โดยบิต CS22 ,CS21 และ CS20

	ใบเนื้อหา		หน้าที่ 13
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟlop		

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

TCNT2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับคาบเวลาของโมดูล Timer 2 เพื่อสร้างคาบเวลาของสัญญาณ PWM

OCR2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับเพื่อใช้ในการเปรียบเทียบค่ากับ TCNT2 เพื่อกำหนดค่าดิวตี้ไซเคิลของสัญญาณ PWM

ในโหมด PWM, Phase Correct ของ Timer 2 สามารถคำนวณหาความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnPCPWM} = \frac{f_{clk_I/O}}{N \cdot 510}$$

ในโหมด Fast PWM ของ Timer 2 สามารถคำนวณหาความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnPWM} = \frac{f_{clk_I/O}}{N \cdot 256}$$

เมื่อ N คือค่าของ Prescaler ในรูปที่ 1.22

(XCK/T0)	PB0	1	40	PA0 (ADC0)
(T1)	PB1	2	39	PA1 (ADC1)
(INT2/AIN0)	PB2	3	38	PA2 (ADC2)
(OC0/AIN1)	PB3	4	37	PA3 (ADC3)
(SS)	PB4	5	36	PA4 (ADC4)
(MOSI)	PB5	6	35	PA5 (ADC5)
(MISO)	PB6	7	34	PA6 (ADC6)
(SCK)	PB7	8	33	PA7 (ADC7)
RESET		9	32	AREF
VCC		10	31	GND
GND		11	30	AVCC
XTAL2		12	29	PC7 (TOSC2)
XTAL1		13	28	PC6 (TOSC1)
(RXD)	PD0	14	27	PC5 (TDI)
(TXD)	PD1	15	26	PC4 (TDO)
(INT0)	PD2	16	25	PC3 (TMS)
(INT1)	PD3	17	24	PC2 (TCK)
(OC1B)	PD4	18	23	PC1 (SDA)
(OC1A)	PD5	19	22	PC0 (SCL)
(ICP1)	PD6	20	21	PD7 (OC2)

รูปที่ 1.23 แสดงตำแหน่งขาสัญญาณที่สามารถสร้างรูปคลื่น PWM ของไมโครคอนโทรลเลอร์ ATMEGA32

	ใบเนื้อหา	หน้าที่ 14
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

1.2 การกำหนดค่าความถี่และดิวตี้ไซเคิลของโมดูล PWM

การกำหนดค่าความถี่และดิวตี้ไซเคิลของโมดูล PWM ในไมโครคอนโทรลเลอร์แต่ละตระกูลนั้นจะมีลักษณะที่คล้ายคลึงกันดังเนื้อหาในหัวข้อ 1.1 กล่าวคือในการกำหนดค่าความถี่ของสัญญาณ PWM จะใช้โมดูลหลักที่เกี่ยวข้องกับการนับสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ในการสร้างค่าคาบเวลาของความถี่ PWM คือโมดูล Timer/Counter ส่วนค่าดิวตี้ไซเคิลของสัญญาณ PWM จะใช้รีจิสเตอร์พิเศษอีกตัวหนึ่งเพื่อกำหนดค่าเปรียบเทียบของการนับเพื่อให้เกิดช่วงเวลาของดิวตี้ไซเคิล ซึ่งไมโครคอนโทรลเลอร์แต่ละตระกูลสามารถสรุปการกำหนดค่าความถี่และดิวตี้ไซเคิลได้ดังนี้


1.2.1 การกำหนดค่าความถี่และดิวตี้ไซเคิลของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2


1.2.1.1 การกำหนดค่าความถี่ของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2 สามารถทำได้ดังนี้

- กำหนดค่าความถี่ของสัญญาณ PWM ที่ต้องการจากค่าคาบเวลาของความถี่จากสมการ
$$T = 256 * (\text{ค่าเวลาของการนับจากการเลือกบิต CPS1 และ CPS0 ของรีจิสเตอร์ CMOD})$$
- ค่าเวลาของการนับ คือ การกำหนดค่าบิต CPS1 และ CPS0 ในรีจิสเตอร์ CMOD เพื่อเลือกแหล่งจ่ายสัญญาณนาฬิกาให้แก่โมดูล PCA เพื่อทำการนับ 0 – 255 (คือการนับ 256 ค่า) เช่นถ้าต้องการเลือกแหล่งจ่ายสัญญาณนาฬิกาให้แก่โมดูล PCA จาก Timer 0 เกิดการ Overflow จะได้ค่าคาบเวลาของสัญญาณ PWM มีค่าเท่ากับ
$$T = 256 * (\text{Timer 0 Overflow})$$
โดย $F = 1/T$

1.2.1.2 การกำหนดค่าดิวตี้ไซเคิลของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2 สามารถทำได้ดังนี้

- กำหนดค่าคาบเวลาของดิวตี้ไซเคิลที่ต้องการแล้วนำค่าที่ได้เข้าไปแทนในสมการดังนี้
$$\text{Duty} = 255 - (255 * (\text{duty cycle}/100))$$
เมื่อ Duty มีค่าอยู่ระหว่าง 0 – 255 เนื่องจากเป็น PWM ที่มีความละเอียด 8 บิตduty cycle มีค่าอยู่ระหว่าง 0 – 100%
- นำค่าที่คำนวณได้นำไปให้ค่ากับรีจิสเตอร์ CCAPnH โดย n คือขาของไมโครคอนโทรลเลอร์ที่สามารถสร้างสัญญาณ PWM ได้มีค่า 0 – 4 (CEX0 – CEX4)

	ใบเนื้อหา	หน้าที่ 15
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์		
<p>1.2.2 การกำหนดค่าความถี่และดิวตี้ไซเคิลของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877 หรือ PIC16F887</p> <p>1.2.2.1 การกำหนดค่าความถี่ของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877 หรือ PIC16F887 สามารถทำได้ดังนี้</p> <p>1. กำหนดค่าความถี่ของสัญญาณ PWM ที่ต้องการจากค่าคาบเวลาของความถี่จากสมการ</p> $\text{PWM Period} = (\text{PR2} + 1) \times 4 \times T_{\text{OSC}} \times (\text{TMR2 Prescale Value})$ <p>โดย T_{osc} คือค่าของ 1/Xtal (เมื่อ Xtal คืออุปกรณ์ที่ต่อกับขา OSC1 และ OSC2 ของไมโครคอนโทรลเลอร์)</p> <p>TMR2 Prescale Value คือค่าจากตารางที่ 1.2 ที่เราต้องการ</p> <p>และ T = PWM Period</p> $F = 1/T$ <p>2. ให้ค่ารีจิสเตอร์ PR2 ลงในสมการในข้อที่ 1 และทำการคำนวณหาค่าคาบเวลาของสัญญาณ PWM</p> <p>1.2.2.2 การกำหนดค่าดิวตี้ไซเคิลของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877 หรือ PIC16F887</p> <p>1. กำหนดค่าของดิวตี้ไซเคิลที่ต้องการจากสมการดังนี้</p> $\text{duty cycle} = 100 - \text{PWM duty cycle (มีค่าอยู่ระหว่าง 0 – 100\%)}$ <p>เมื่อ PWM duty cycle = (CCPRxL:CCPxCON<5:4>) × T_{OSC} × (TMR2 Prescale)</p> <p>T_{osc} คือค่าของ 1/Xtal (เมื่อ Xtal คืออุปกรณ์ที่ต่อกับขา OSC1 และ OSC2 ของไมโครคอนโทรลเลอร์)</p> <p>(CCPRxL:CCPxCON<5:4>) คือค่าข้อมูลที่จะใช้ในการคำนวณหาของช่วงเวลา Duty Cycle มีความละเอียดขนาด 10 บิต</p> <p>2. ค่ารีจิสเตอร์ CCPRxL ที่ใช้คำนวณหาของดิวตี้ไซเคิลจะต้องมีค่าไม่เกินค่าของรีจิสเตอร์ PR2</p> <p>1.2.3 การกำหนดค่าความถี่และดิวตี้ไซเคิลของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32</p> <p>1.2.3.1 การกำหนดค่าความถี่ของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32 ที่หาสัญญาณ OCR0 ,OCR1A ,OCR1B และ OCR2 มีหลักการที่เหมือนกันสามารถทำได้ดังนี้</p> <p>1. เลือกโหมดการทำงานของการทำงานสร้างสัญญาณ PWM ,ลักษณะการทำงานของหาสัญญาณที่สามารถสร้างสัญญาณ PWM และความละเอียดของการนับสัญญาณนาฬิกาเพื่อสร้างคาบเวลาของสัญญาณ PWM ด้วยรีจิสเตอร์ TCCR_x โดย x สามารถแทนได้ด้วยตัวเลข 0 – 2</p>		

	ใบเนื้อหา	หน้าที่ 16
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

2. คำนวณหาค่าความถี่ของสัญญาณนาฬิกาจากสมการที่กำหนดในแต่ละโหมดของการสร้างสัญญาณ PWM เช่นการสร้างสัญญาณ PWM ในโหมด Fast PWM Mode จะใช้สมการในการหาความถี่คือ

$$f_{OCnPWM} = \frac{f_{clk_I/O}}{N \cdot 256}$$

โดย $f_{clk_I/O}$ คือค่าของอุปกรณ์ XTAL ที่ต่ออยู่กับขา XTAL1 และ XTAL2
N คือค่าของ Prescaler ที่เรากำหนดตามตาราง
 $T = 1/F$

1.2.3.2 การกำหนดค่าดิวตี้ไซเคิลของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32

การกำหนดค่าของดิวตี้ไซเคิลสามารถทำได้ด้วยการกำหนดค่าลงในรีจิสเตอร์ OCRx ซึ่งการให้ค่าแก่รีจิสเตอร์ OCRx จะมีความหมายไม่เหมือนกัน ขึ้นอยู่กับการเลือกโหมด และหน้าที่การทำงานของขาสัญญาณ OCRx ในการสร้างสัญญาณ PWM เช่นถ้าเราเลือกโหมดการทำงานแบบ Fast PWM Mode กำหนดให้ บิต COMx1 เป็นลอจิก ‘1’ และ บิต COMx0 เป็นลอจิก ‘0’ ขาสัญญาณ OCRx จะเป็นสัญญาณลอจิก ‘1’ เมื่อค่าข้อมูลของ OCRx มีค่ามากกว่าค่าข้อมูลของ TCNT0 ดังนั้นค่าของ OCRx เพื่อสร้างช่วงเวลาของดิวตี้ไซเคิลหาได้จาก

$$OCRx = 255 * (duty / 100)$$

เมื่อ duty มีค่า 0 -100 % โดยสมการนี้ใช้ได้กับขา OC0 และ OC2


และถ้ากำหนดให้บิต COMx1 เป็นลอจิก ‘1’ และ บิต COMx0 เป็นลอจิก ‘1’ ขาสัญญาณ OCRx จะเป็นสัญญาณลอจิก ‘1’ เมื่อค่าข้อมูลของ OCRx มีค่าน้อยกว่าหรือเท่ากับค่าข้อมูลของ TCNT0 ดังนั้นค่าของ OCRx เพื่อสร้างช่วงเวลาของดิวตี้ไซเคิลหาได้จาก


$$OCRx = 255 - (255 * (duty / 100))$$

1.3 ตัวอย่างการสร้างสัญญาณ PWM ด้วยไมโครคอนโทรลเลอร์

1.3.1 ตัวอย่างการสร้างนาฬิกาด้วยโมดูล PWM ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51RD2 หรือ AT89C51ED2 ที่ความถี่ 3.9kHz Duty Cycle 50% โดยใช้ขาสัญญาณ CEX1

```
void pwmOnCEX1(){
    CMODE = 0x00; //CPS1 = 0 ,CPS0 = 0 รีจิสเตอร์ PCA นับค่าทุก ๆ 1uS จาก 1/(12MHz/12)
                //T = 256 * 1uS = 256uS และ F = 1/256uS = 3,906.25Hz
    CCAPM1 = 0x42; // ให้สร้างสัญญาณ PWM ด้วยโมดูล PCA ที่ขา CEX1
    CCON = 0x40;  // ให้โมดูล PCA เริ่มทำการนับเพื่อสร้างคาบเวลาให้แก่สัญญาณ PWM
    CCAP1H = 255 - ((255*50)/100) ; //ให้ขา CEX1 สร้างสัญญาณ PWM ที่มีค่า Duty Cycle
                ; //50%
}
```

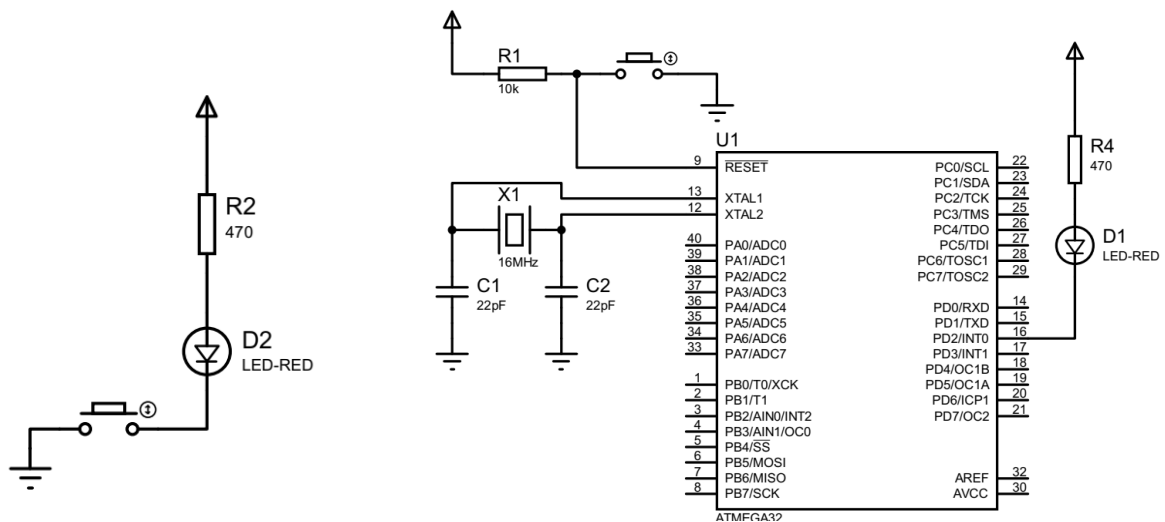
	ใบเนื้อหา		หน้าที่ 17
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟลอป		
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์			
<p>1.3.2 ตัวอย่างการสร้างนาฬิกาด้วยโมดูล PWM ของไมโครคอนโทรลเลอร์ตระกูล PIC16F เบอร์ PIC16F877 หรือ PIC16F887 ที่ความถี่ 4kHz Duty Cycle 50% โดยใช้สัญญาณ CCP1</p> <pre>void pwmOnCCP1(){ CCP1CON = 0x0f; //กำหนดให้โมดูล CCP ทำงานในรูปแบบ PWM Mode PR2 = 77; // กำหนดคาบเวลาของสัญญาณ PWM = 0.25ms // PR2 = (0.25ms/(4x(1/20MHz)x16)) - 1 CCPR1L = 39; // กำหนดค่า CCPR1L ให้สร้าง Duty Cycle = 50% T2CON = 0x07; //ให้โมดูล Timer2 เริ่มทำงานเพื่อป้อนสัญญาณการนับให้แก่โมดูล CCP //และใช้ค่า Prescaler ของการนับเท่ากับ 16 }</pre> <p>1.3.3 ตัวอย่างการสร้างนาฬิกาด้วยโมดูล PWM ของไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32 ที่ความถี่ 3.9kHz Duty Cycle 50% โดยใช้สัญญาณ OCR0</p> <pre>void pwmOnOCR0(){ TCCR0 = 0x62; //กำหนดให้โมดูล Timer0 ทำงานในรูปแบบ PWM Phase Correct Mode // Timer0 นับ 0 – 255 ค่า ใช้ Prescaler 1/8 และขา OCR0 จะเป็นลอจิก // ‘1’ เมื่อค่าของ OCR0 น้อยกว่าค่า Timer0 เป็นผลให้ความถี่ที่ขา OCR0 // มีค่าเท่ากับ 16Mz/(8*510) = 3,921.57Hz OCR0 = (255 * 50) / 100; // กำหนดค่าคาบเวลาพัลส์ให้เท่ากับ 50% }</pre>			
<h2>2. วงจรแสดงผลด้วยอุปกรณ์ LED</h2> <p>ในการใช้งานอุปกรณ์ LED เพื่อใช้เป็นอุปกรณ์ในการแสดงผลสามารถนำมาต่อใช้งานในวงจรดิจิทัลและไมโครคอนโทรลเลอร์จำนวน 2 รูปแบบ โดยการต่อใช้งานจะต้องคำนึงถึงแรงดันตกคร่อมอุปกรณ์ LED กระแสไฟฟ้าที่ไหลผ่านอุปกรณ์ LED และกระแสที่ไหลเข้าสู่อุปกรณ์หรือไหลออกจากอุปกรณ์ที่ทำการเชื่อมต่อกับ LED ซึ่งมีลักษณะการต่อใช้งานดังนี้</p> <h3>2.1 การต่อวงจร LED ในรูปแบบ Active High</h3> <p>การต่อวงจร LED ในรูปแบบ Active High คือการต่อวงจรควบคุมการแสดงผลของอุปกรณ์ LED ด้วยแรงดัน Vcc หรือสัญญาณลอจิก ‘1’ เพื่อให้กระแสไหลจากขั้วบวกของแหล่งจ่ายไฟ หรือไหลจากขาเอาต์พุตที่ทำหน้าที่เสมือนเป็นขั้วบวกของแหล่งจ่ายไฟ ผ่านตัวต้านทานและผ่านอุปกรณ์ LED แล้วครบวงจรที่จุด GND ของแหล่งจ่าย โดยส่วนใหญ่อุปกรณ์ LED สามารถทนกระแสได้ไม่เกิน 20mA ดังนั้นจึงควรต่อตัวต้านทานอนุกรมกับอุปกรณ์ LED เพื่อจำกัดกระแสไฟฟ้าไม่ให้ไหลผ่านตัวอุปกรณ์ LED เกินค่าพิกัด ซึ่งลักษณะการต่อใช้งานสามารถแสดงได้ดังรูปที่ 1.24</p>			

	ใบเนื้อหา	หน้าที่ 19
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

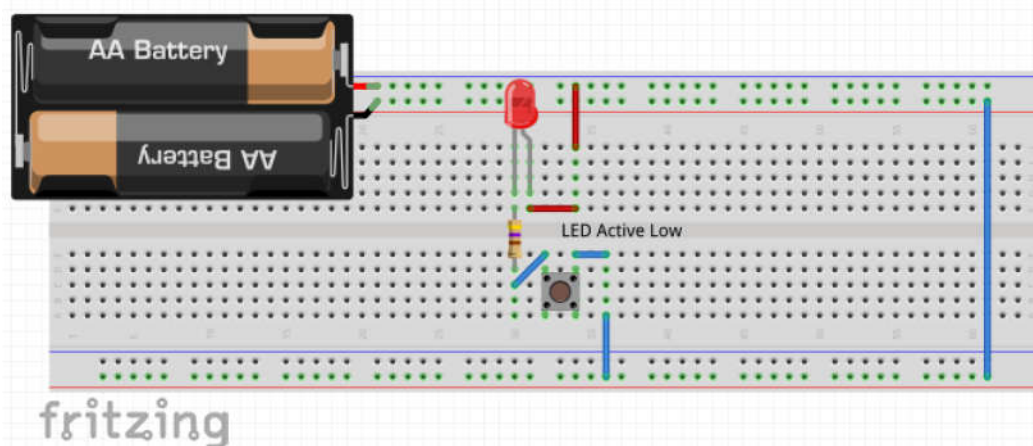
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

2.2 การต่อวงจร LED ในรูปแบบ Active Low


การต่อวงจร LED ในรูปแบบ Active Low คือการต่อวงจรควบคุมการแสดงผลของอุปกรณ์ LED ด้วยแรงดัน GND หรือสัญญาณลอจิก '0' เพื่อให้กระแสไหลจากขั้วบวกของแหล่งจ่ายไฟไปยังขาเอาต์พุตของอุปกรณ์ที่ทำหน้าที่เสมือนเป็นขั้ว GND ของแหล่งจ่ายไฟ ผ่านตัวต้านทานและผ่านอุปกรณ์ LED แล้วครบวงจรดังรูปที่ 1.25 ซึ่งการต่อในลักษณะนี้ขาเอาต์พุตของอุปกรณ์ดิจิทัลหรือไมโครคอนโทรลเลอร์จะทำหน้าที่ในการรับกระแสจากอุปกรณ์ LED ที่เรียกว่าการ Sink



(ก) วงจรทดสอบการทำงานของ LED แบบ Active Low (ข) การต่อใช้งาน LED กับไมโครคอนโทรลเลอร์แบบ Active Low



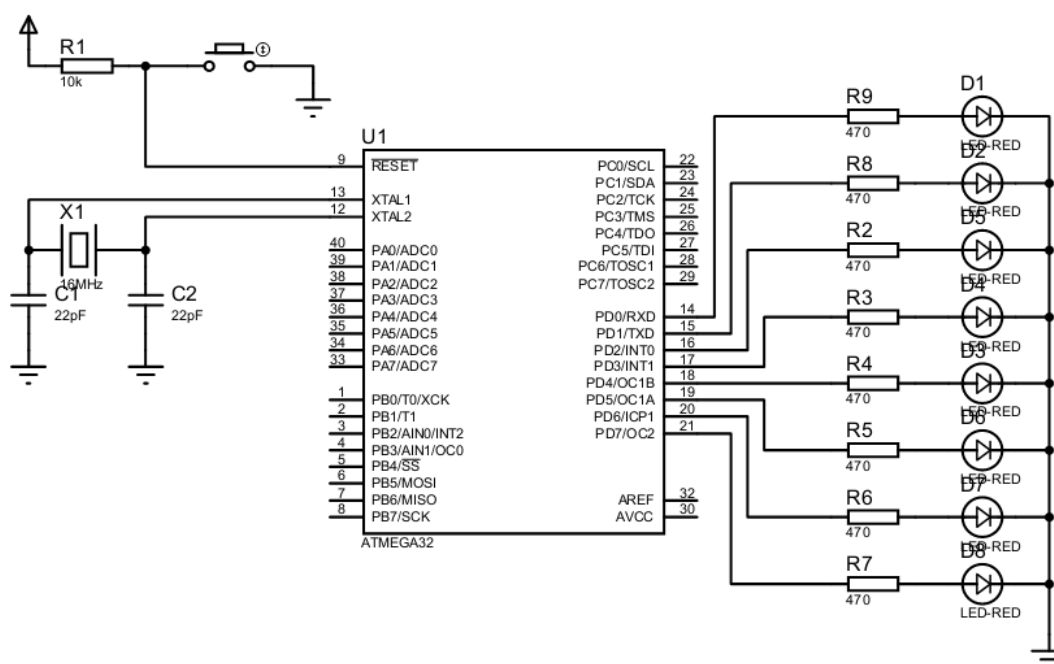
(ค) แสดงการต่อวงจร LED แบบ Active Low แบบเสมือนจริงด้วยโปรแกรม fritzing รูปที่ 1.25 ลักษณะการต่อวงจร LED แบบ Active Low

	ใบเนื้อหา		หน้าที่ 20
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

2.3 ตัวอย่างการต่อวงจรใช้งาน LED ในรูปแบบไฟวิ่ง 8 บิต

รูปที่ 1.26 เป็นตัวอย่างการต่อวงจรใช้งาน LED แบบ Active High ในรูปแบบไฟวิ่ง 8 บิต โดยใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32



รูปที่ 1.26 การต่อวงจร LED แบบ Active High ในรูปแบบไฟวิ่ง 8 บิต ด้วย ATMEGA32


3. การแสดงผลบน 7-Segment ด้วยไมโครคอนโทรลเลอร์

อุปกรณ์ 7-Segment ก็คือการนำเอาอุปกรณ์ LED มาต่อร่วมกันทั้งหมด 7-8 ตัวเพื่อใช้สำหรับการแสดงผลตัวเลข หรือตัวอักษร ซึ่งขาต่อใช้งานจะแบ่งออกเป็น 2 กลุ่ม คือกลุ่มขารับสัญญาณข้อมูลการแสดงผล หรืออาจเรียกว่ากลุ่มขาสัญญาณดาต้า และกลุ่มขาสัญญาณควบคุมการปิดเปิดการแสดงผล หรือเรียกว่าขาคอมมอน โดย 7-Segment มี 2 ชนิดใหญ่ได้แก่ ชนิดคอมมอนคาโทด และชนิดคอมมอนแอนโนด ซึ่งการต่อวงจรแสดงผลด้วยอุปกรณ์ 7-Segment ร่วมกับไมโครคอนโทรลเลอร์สามารถทำได้ดังนี้

3.1 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์จำนวน 1 หลัก

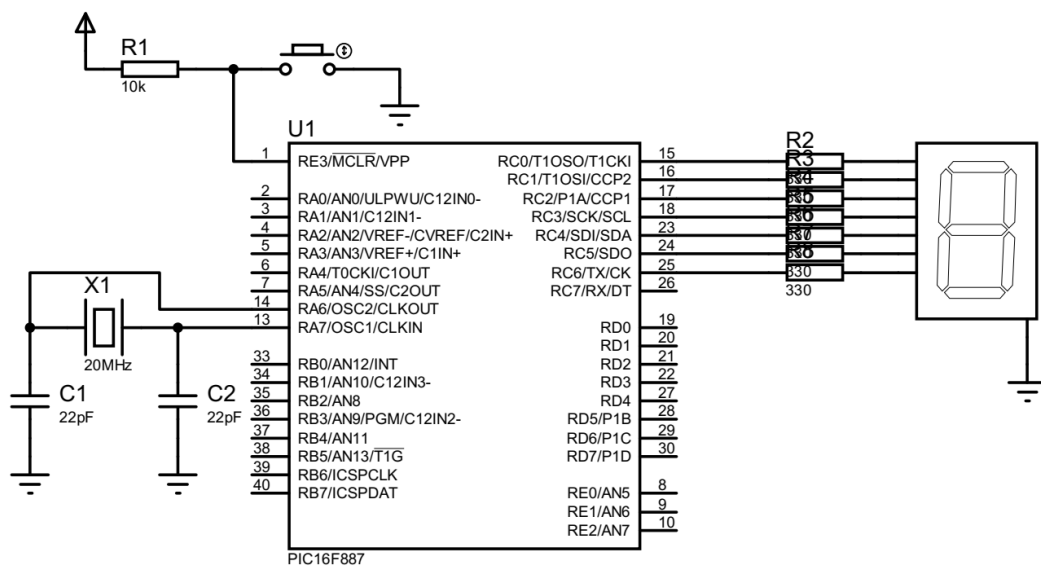
การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์จำนวน 1 หลักสามารถทำได้โดยการต่อวงจรแบบ latch data ซึ่งสามารถทำได้ 2 วิธีคือ

1. ให้ทำการต่อขาสัญญาณดาต้าของ 7-Segment เข้ากับขาของไมโครคอนโทรลเลอร์โดยตรง แล้วทำการต่อขาคอมมอนของ 7-Segment เข้ากับ Vcc หรือ GND ขึ้นอยู่กับชนิดของ 7-Segment ดังรูปที่ 1.27

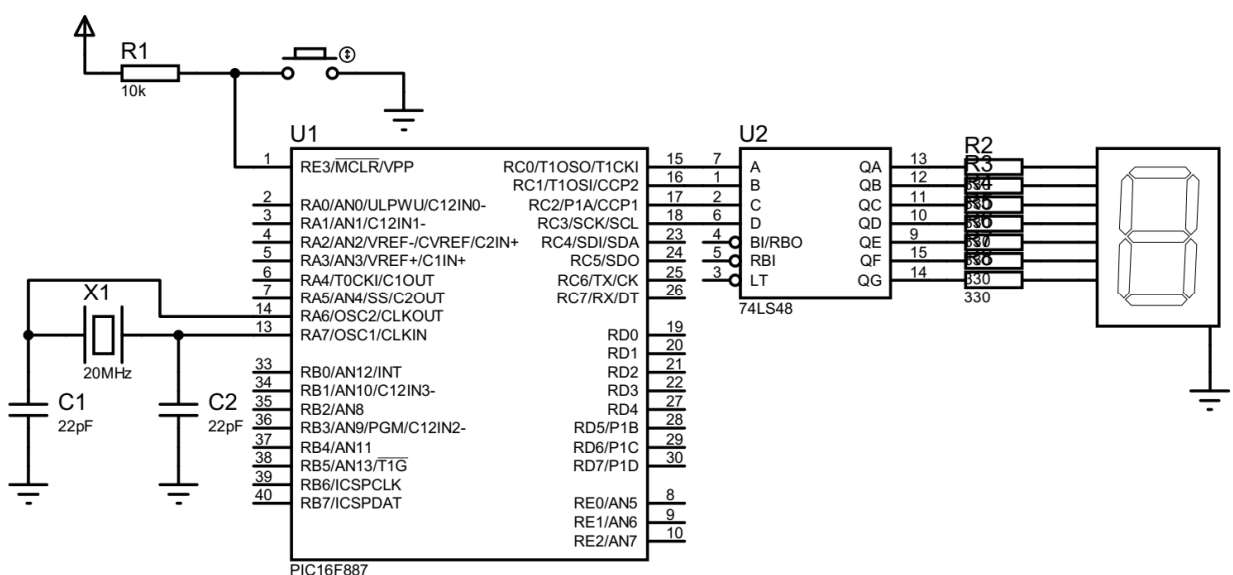
	ใบเนื้อหา	หน้าที่ 21
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรพลิกฟลอป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์


2. ให้ทำการต่อขาสัญญาณของไมโครคอนโทรลเลอร์เข้ากับไอซีถอดรหัสสัญญาณ 7-Segment เบอร์ 74LS47 หรือ 74LS48 ขึ้นอยู่กับชนิดของ 7-Segment ทำการต่อขาสัญญาณคาต้าของ 7-Segment เข้ากับขาของไอซีถอดรหัส แล้วทำการต่อขาคอมมอนของ 7-Segment เข้ากับ Vcc หรือ GND ขึ้นอยู่กับชนิดของ 7-Segment ดังรูปที่ 1.28



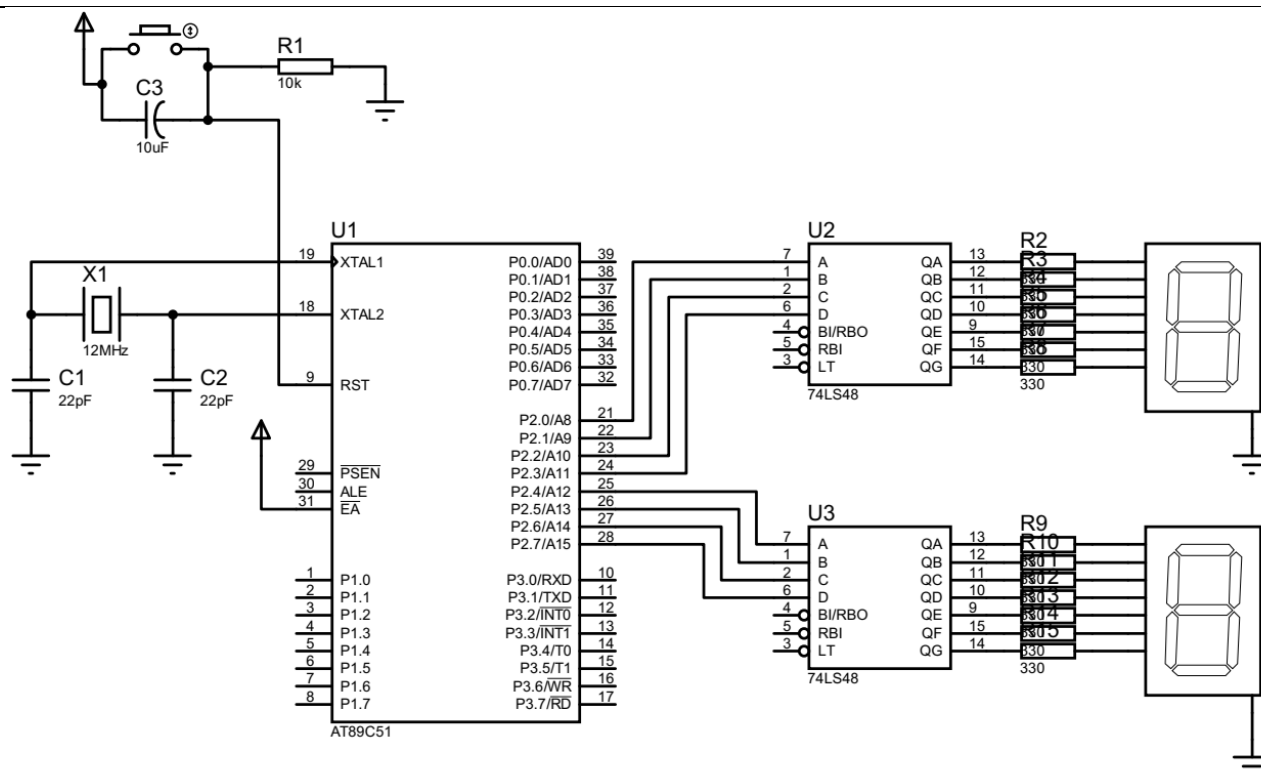
รูปที่ 1.27 ตัวอย่างการใช้ PIC16F887 ต่อใช้งานอุปกรณ์ 7-Segment แสดงผล 1 หลัก โดยไม่มีไอซีถอดรหัส



รูปที่ 1.28 ตัวอย่างการใช้ PIC16F887 ต่อใช้งานอุปกรณ์ 7-Segment แสดงผล 1 หลัก โดยใช้ไอซีถอดรหัส

	ใบเนื้อหา	หน้าที่ 23
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์



รูปที่ 1.30 ตัวอย่างการใช้ AT89C51 ต่อใช้งานอุปกรณ์ 7-Segment แสดงผล 2 หลัก โดยใช้ไอซีถอดรหัส

3.2.2 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์จำนวนมากกว่า 1 หลักด้วยวิธีการมัลติเพล็กซ์สัญญาณ ซึ่งวิธีการนี้จะทำให้ประหยัดพอร์ตของไมโครคอนโทรลเลอร์ในการส่งข้อมูลไปแสดงผลที่อุปกรณ์ 7-Segment และยังสามารถแบ่งออกได้เป็น 2 วิธีการ คือ

3.2.2.1 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์จำนวนมากกว่า 1 หลักด้วยวิธีการมัลติเพล็กซ์สัญญาณ โดยใช้ขาพอร์ตของไมโครคอนโทรลเลอร์ 1 พอร์ตทำหน้าที่ในการส่งรหัสตัวเลข หรือตัวอักษร เพื่อแสดงบนอุปกรณ์ 7-Segment ส่วนอีกพอร์ตหนึ่งทำหน้าที่ในการควบคุมตำแหน่งของหลักที่จะแสดงผลของ 7-Segment ผ่านไอซีถอดรหัสสัญญาณ ดังตัวอย่างในรูปที่ 1.31

3.2.2.2 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์จำนวนมากกว่า 1 หลักด้วยวิธีการมัลติเพล็กซ์สัญญาณ โดยใช้ขาพอร์ตของไมโครคอนโทรลเลอร์ 1 พอร์ตเท่านั้น โดยขาของไมโครคอนโทรลเลอร์ จำนวน 4 บิตทำหน้าที่ในการส่งข้อมูลไปยังไอซีถอดเพื่อแปลงเป็นตัวเลข หรือตัวอักษรเพื่อแสดงบนอุปกรณ์ 7-Segment ส่วนอีก 4 บิตถัดมาทำหน้าที่ในการควบคุมตำแหน่งของหลักที่จะแสดงผลของ 7-Segment ผ่านไอซีถอดรหัสสัญญาณ ดังตัวอย่างในรูปที่ 1.32



ใบเนื้อหา

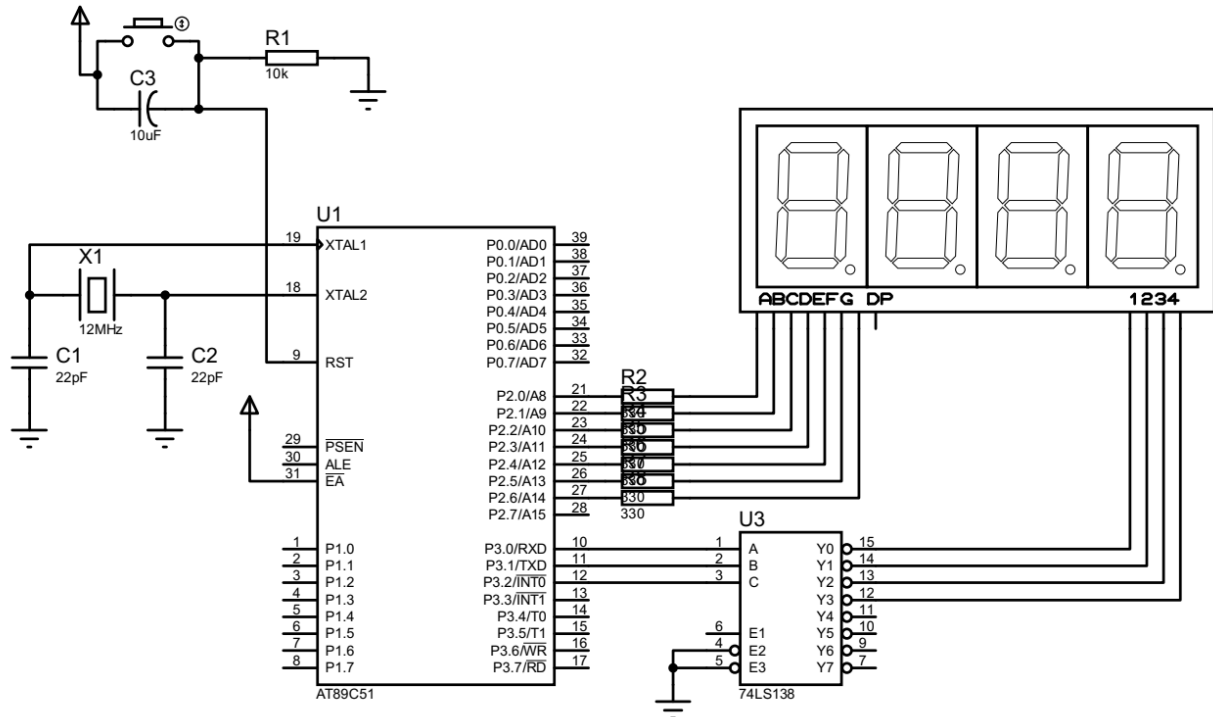
หน้าที่ 24

ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004

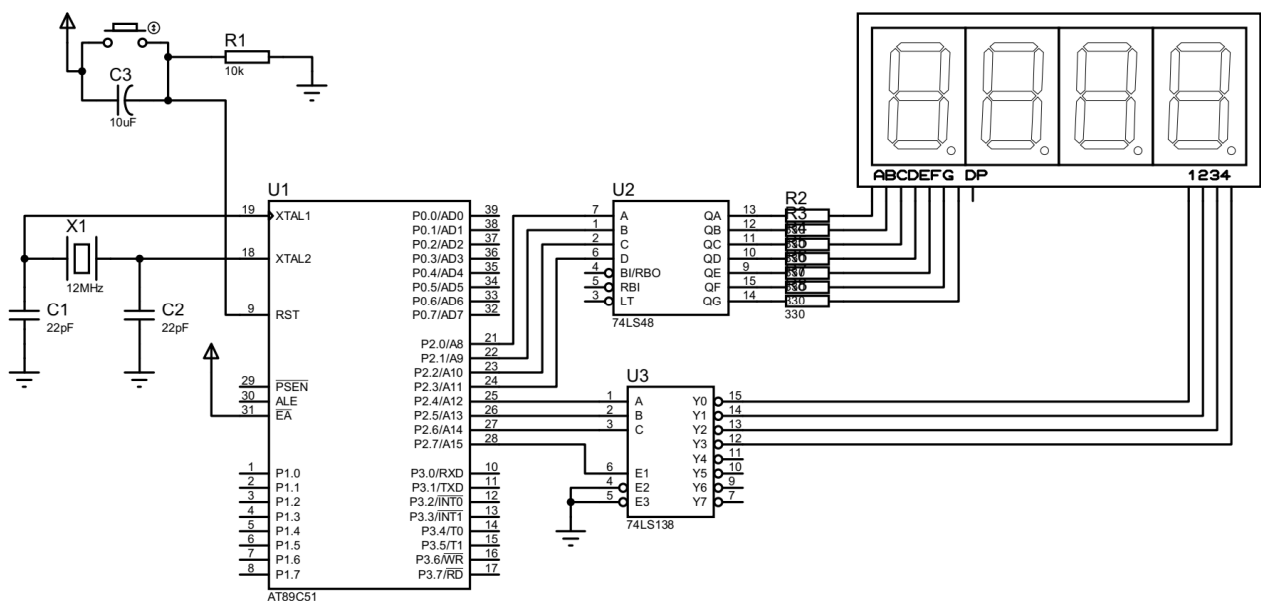
หน่วยที่ 3

ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์



รูปที่ 1.31 ตัวอย่างการใช้ AT89C51 ต่อใช้งานอุปกรณ์ 7-Segment แสดงผล 4 หลัก รูปแบบที่ 1



รูปที่ 1.32 ตัวอย่างการใช้ AT89C51 ต่อใช้งานอุปกรณ์ 7-Segment แสดงผล 4 หลัก รูปแบบที่ 2

	ใบเนื้อหา	หน้าที่ 25
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

4. การแสดงผลบน Character LCD ด้วยไมโครคอนโทรลเลอร์

4.1 คุณสมบัติของจอ LCD

LCD ย่อมาจากคำว่า Liquid Crystal Display โดยจอแอลซีดีจะประกอบด้วยแผ่นแก้ว 2 แผ่นประกบกัน เว้นช่องว่างตรงกลางไว้ประมาณ 6-10 ไมโครเมตร ด้านในจะเคลือบด้วยตัวนำไฟฟ้าแบบใสไว้ในระหว่างตัวนำไฟฟ้าแบบใส และจะมีโมเลกุลผลึกรวมตัวกันในทิศทางที่แสงส่องผ่าน หลักการทำงานของ LCD คือ ด้านหลังจอแอลซีดีจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight ต่ออยู่ เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึกคริสตัล ก็จะทำให้ผลึกคริสตัลโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอแอลซีดี ส่วนอื่นที่โดนผลึกปิดกั้นไว้ จะมีสีที่แตกต่างตามสีของผลึกคริสตัล เช่น สีเขียว หรือ สีฟ้า ทำให้เมื่อมองไปที่จอก็จะพบกับตัวหนังสือสีขาว และพื้นหลังที่มีสีเดียวกันกับผลึกคริสตัล ซึ่งจอ LCD จะแบ่งเป็น 2 แบบใหญ่ ๆ ตามลักษณะการแสดงผลดังนี้

1. Character LCD เป็นจอแสดงผลที่สามารถแสดงผลเป็นตัวอักษรหรือตัวเลข และสัญลักษณ์พิเศษต่าง ๆ ในภาษาอังกฤษ และภาษาญี่ปุ่น ตามช่องแบบตายตัว เช่น จอ LCD ขนาด 16x2 หมายถึงใน 1 บรรทัด สามารถแสดงผลได้สูงสุด 16 ตัวอักษร และมีทั้งหมด 2 บรรทัด เป็นต้น

2. Graphic LCD เป็นจอแสดงผลที่สามารถกำหนดได้ว่าจะให้จุดบนหน้าจอ ณ ตำแหน่งใด ๆ สามารถกันแสง หรือปล่อยแสงออกมา ทำให้จอแสดงผลสามารถสร้างรูปภาพหรือตัวอักษรขึ้นมาบนหน้าจอได้ ส่วนขนาดของ Graphic LCD จะทำการระบุขนาดได้โดยระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอ Graphic LCD ที่มีขนาดจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด เป็นต้น

4.2 รูปแบบการเชื่อมต่อ Character LCD กับไมโครคอนโทรลเลอร์

Character LCD มีหลายขนาดให้เลือกใช้งานได้แก่ ขนาด 8x1 บรรทัด, 8x2 บรรทัด, 16x1 บรรทัด, 16x2 บรรทัด, 20x2 บรรทัด และ 20x4 บรรทัด เป็นต้น และมี 2 ชนิด คือ แบบมี Back light และ แบบไม่มี Back light ซึ่ง Character LCD จะมีขาให้ต่อใช้งานทั้งหมด 16 ขาดังนี้

ขาที่ 1 คือขา VSS ใช้สำหรับต่อกับขา GND ของแหล่งจ่ายไฟ

ขาที่ 2 คือขา VDD ใช้สำหรับต่อกับขาไฟบวก 5V ของแหล่งจ่ายไฟ

ขาที่ 3 คือขา VO เป็นขาที่ใช้สำหรับปรับความคมชัดของตัวอักษรที่แสดงบนจอ LCD

ขาที่ 4 คือขา RS เป็นขาอินพุตที่ใช้สำหรับกำหนดว่าข้อมูลที่เข้ามาทางขา data ของ LCD เป็นข้อมูลที่จะใช้แสดงผล หรือข้อมูลชุดคำสั่ง ดังนี้

ถ้า RS = '0' แสดงว่าข้อมูลที่ขา data คือข้อมูลชุดคำสั่ง

ถ้า RS = '1' แสดงว่าข้อมูลที่ขา data คือชุดข้อมูลที่จะใช้สำหรับการแสดงผลบนจอ LCD

ขาที่ 5 คือขา R/W เป็นขาอินพุตที่ใช้กำหนดว่าการเชื่อมต่อระหว่าง LCD กับไมโครคอนโทรลเลอร์จะเป็นลักษณะของการอ่านข้อมูล หรือการเขียนข้อมูลให้กับ LCD ดังนี้

ถ้า R/W = '0' แสดงว่าไมโครคอนโทรลเลอร์ต้องการเขียนข้อมูลให้กับ LCD


ถ้า R/W = '1' แสดงว่าไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจาก LCD

	ใบเนื้อหา	หน้าที่ 26
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์
<p>ขาที่ 6 คือขา E เป็นขาอินพุตที่ใช้สำหรับกำหนดให้ LCD เริ่มทำงานตามคำสั่ง</p> <p>ขาที่ 7 – 14 คือขา D0-D7 เป็นขาอินพุตที่ทำหน้าที่ในการรับข้อมูลที่เป็นชุดคำสั่ง และข้อมูลที่ต้องการแสดงผลออกจอ LCD</p> <p>ขาที่ 15 คือขา A เป็นขา Anode ของ LED Black light</p> <p>ขาที่ 16 คือขา K เป็นขา Cathode ของ LED Black light</p> <p>ส่วนการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับจอ Character LCD สามารถทำได้ 2 วิธีคือ</p> <p>1. การเชื่อมต่อในรูปแบบการใช้ขาสัญญาณ data จำนวน 4 ขา ได้แก่ขา D4 – D7</p> <p>1.1 ใช้ขาสัญญาณควบคุมจำนวน 2 ขา คือขา RS และ EN ส่วนขา R/W ต่อลงกราวด์ ซึ่งการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผลได้อย่างเดียว และใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 6 ขา</p> <p>1.2 ใช้ขาสัญญาณควบคุมจำนวน 3 ขา คือขา RS , EN และขา R/W โดยการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผล และอ่านข้อมูลการแสดงผลได้ ซึ่งใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 7 ขา</p> <p>2. การเชื่อมต่อในรูปแบบการใช้ขาสัญญาณ data จำนวน 8 ขา ได้แก่ขา D0 – D7</p> <p>2.1 ใช้ขาสัญญาณควบคุมจำนวน 2 ขา คือขา RS และ EN ส่วนขา R/W ต่อลงกราวด์ ซึ่งการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผลได้อย่างเดียว และใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 10 ขา (2 พอร์ต)</p> <p>2.2 ใช้ขาสัญญาณควบคุมจำนวน 3 ขา คือขา RS , EN และขา R/W โดยการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผล และอ่านข้อมูลการแสดงผลได้ ซึ่งใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 11 ขา (2 พอร์ต)</p> <p>การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับจอ Character LCD ส่วนใหญ่จะใช้วิธีที่ 1.1 เนื่องจากประหยัดขาสัญญาณในการควบคุม</p>



รูปที่ 1.33 รูปตัวอย่างจอแสดงผล Character LCD ขนาด 16x2

	ใบเนื้อหา		หน้าที่ 27
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรพลิกฟลอป		


ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

ตารางที่ 1.3 ตารางแสดงความสัมพันธ์ในการทำงานของขาสัญญาณ RS,R/W และ E ของ Character LCD

RS	R/W	E	การทำงาน
0	0		เขียนคำสั่ง
0	1		อ่านสถานะของโมดูล LCD
1	0		เขียนข้อมูล
1	1		อ่านข้อมูล

Instruction	Instruction Code										Description Instruction Code	Execution time (fsoc=270kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM, and set DDRAM address to "00H" from AC.	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	X	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and make shift of entire display enable.	39μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	X	X	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	39μs
Function Set	0	0	0	0	1	DL	N	F	X	X	Set interface data length (DL : 4-bit/8-bit), numbers of display line (N : 1-line/2-line), display font type(F : 5 X 8 dots/ 5 X 11 dots)	39μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39μs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43μs

รูปที่ 1.34 รูปตารางชุดคำสั่งของ Character LCD

	ใบเนื้อหา	หน้าที่ 28
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

บรรทัดที่ 1

00H	01H	02H	03H	04H	05H	06H	07H	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
40H	41H	42H	43H	44H	45H	46H	47H	48H	49H	4AH	4BH	4CH	4DH	4EH	4FH

บรรทัดที่ 2

รูปที่ 1.35 แสดงแอดเดรสของการแสดงผลบนจอ Character LCD ขนาด 16x2

ชุดคำสั่งพื้นฐานที่ใช้ในการควบคุมการทำงานของ Character LCD ได้แก่ชุดคำสั่ง


1. คำสั่ง 0x01 คือคำสั่งที่ใช้ในการเคลียร์หน้าจอการแสดงผล
2. คำสั่ง 0x06 คือคำสั่งที่ใช้ในการกำหนดว่าเมื่ออ่านหรือเขียนข้อมูลให้ LCD จะทำให้ DDRAM ของ LCD เพิ่มขึ้น 1 ตำแหน่ง และเคอร์เซอร์จะถูกเลื่อนไปทางขวามือ
3. คำสั่ง 0x0C คือคำสั่งที่ใช้ในการกำหนดให้ LCD ทำการแสดงผลได้โดยไม่ให้แสดงเคอร์เซอร์
4. คำสั่ง 0x28 คือคำสั่งที่ใช้ในการกำหนดให้ LCD ทำการติดต่อกับ CPU แบบ 4 บิต และขนาดตัวอักษรเท่ากับ 5*7 Dot
5. คำสั่ง 0x38 คือคำสั่งที่ใช้ในการกำหนดให้ LCD ทำการติดต่อกับ CPU แบบ 8 บิต และขนาดตัวอักษรเท่ากับ 5*7 Dot

4.3 การเขียนโปรแกรมแสดงผลบน Character LCD ขนาด 16x2 ด้วยไมโครคอนโทรลเลอร์

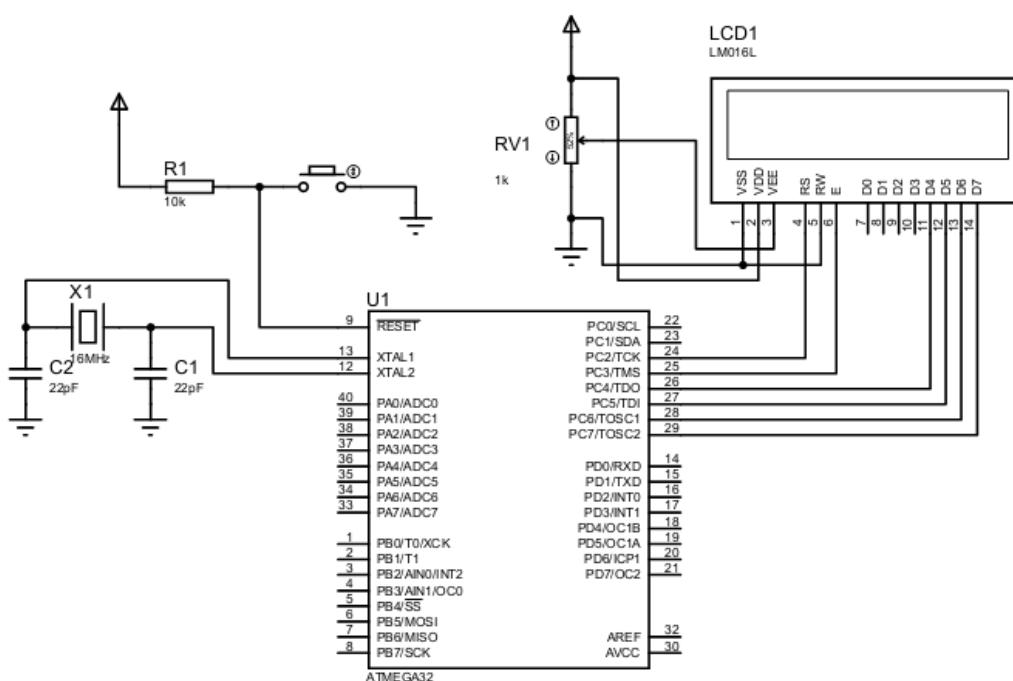
สำหรับการเขียนโปรแกรมควบคุมอุปกรณ์ Character LCD จะเป็นการเขียนโปรแกรมควบคุม Character LCD ที่ทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ด้วยขา Data จำนวน 4 ขา และใช้ขาควบคุมจำนวน 2 ขา คือขา RS และขา E ซึ่งการเขียนโปรแกรมควบคุมอุปกรณ์ Character LCD จะมีฟังก์ชันที่จำเป็นจำนวน 4 ฟังก์ชัน คือ

1. ฟังก์ชัน void lcd_busy(int time) คือฟังก์ชันสำหรับการสร้างสัญญาณ Enable ให้แก่ขา E ของ Character LCD เพื่อให้ Character LCD ทำการประมวลผลข้อมูลตามสัญญาณที่ขา RS ,R/W และ D0 – D7
2. ฟังก์ชัน void lcd_command(unsigned char cmd) คือฟังก์ชันที่มีหน้าที่สำหรับเขียนข้อมูลชุดคำสั่งให้แก่อุปกรณ์ Character LCD
3. ฟังก์ชัน void lcd_putc(unsigned char dat) คือฟังก์ชันที่มีหน้าที่สำหรับเขียนข้อมูลที่ต้องการแสดงผลให้แก่อุปกรณ์ Character LCD
4. ฟังก์ชัน void lcd_init() คือฟังก์ชันที่มีหน้าที่กำหนดค่าเริ่มต้นให้แก่อุปกรณ์ Character LCD

ในเนื้อหาหน่วยนี้จะเป็นการเขียนโปรแกรมเชื่อมต่ออุปกรณ์ Character LCD ขนาด 16x2 ด้วยไมโครคอนโทรลเลอร์ AVR ผ่านพอร์ต PC ดังรูปที่ 1.36 และมีตัวอย่างการเขียนโปรแกรดังนี้


	ใบเนื้อหา		หน้าที่ 29
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟlop		


ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์





รูปที่ 1.36 แสดงวงจรการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ ATMEGA32 กับ Character LCD ตัวอย่างโปรแกรมควบคุมการแสดงผลด้วย Character LCD ขนาด 16x2

```
#include<avr/io.h> //เรียกใช้งาน Library เกี่ยวกับชื่อรีจิสเตอร์ของไมโครคอนโทรลเลอร์ ATMEGA32
#define F_CPU 16000000UL //ไมโครคอนโทรลเลอร์ ATMEGA32 ใช้ XTAL ค่า 16MHZ
#include<util/delay.h> //เรียกใช้งาน Library เกี่ยวกับฟังก์ชันหน่วงเวลา
#define lcd_ddr DDRC //กำหนดข้อความ lcd_ddr แทนรีจิสเตอร์ DDRC
#define lcd_port PORTC //กำหนดข้อความ lcd_port แทนรีจิสเตอร์ PORTC
# define rs_lcd 2 //กำหนดข้อความ rs_lcd แทนตัวเลข 2
# define en_lcd 3 //กำหนดข้อความ en_lcd แทนตัวเลข 3
const unsigned char ascii[] = "0123456789ABCDEF"; //ประกาศตัวแปรอาร์เรย์ชื่อ ascii แบบอ่านได้
//อย่างเดียวเท่านั้น
void lcd_busy(int time){ //สร้างฟังก์ชันชื่อ lcd_busy เพื่อสร้างสัญญาณ Enable ให้แก่ LCD
    lcd_port &= ~(1<<en_lcd); //ให้ขา Enable ของ LCD มีค่าเป็นลอจิก '0'
    for(;time>0;time--) _delay_us(500); //วนรอบเรียกใช้งานฟังก์ชัน _delay_us(500) จนกว่าค่า
    //ตัวแปร time จะมีค่าเป็น 0
    lcd_port |= (1<<en_lcd); //ให้ขา Enable ของ LCD มีค่าเป็นลอจิก '1'
}
```

	ใบเนื้อหา		หน้าที่ 30
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป		
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์			
<pre>void lcd_command(unsigned char cmd){ //สร้างฟังก์ชันชื่อ lcd_command เพื่อส่งคำสั่งให้ LCD lcd_port &= ~(1<<rs_lcd); //ให้ขา RS ของ LCD มีค่าเป็นลอจิก '0' lcd_port = (lcd_port & 0x0f) (cmd & 0xf0); //ส่งค่าคำสั่ง 4 บิตบนให้แก่ LCD lcd_busy(2); //เรียกใช้งานฟังก์ชัน lcd_busy และให้ค่าฟังก์ชันเท่ากับ 2 lcd_port = (lcd_port & 0x0f) (cmd << 4); //ส่งค่าคำสั่ง 4 บิตล่างให้แก่ LCD lcd_busy(2); //เรียกใช้งานฟังก์ชัน lcd_busy และให้ค่าฟังก์ชันเท่ากับ 2 lcd_port = (1<<rs_lcd); //ให้ขา RS ของ LCD มีค่าเป็นลอจิก '1' } void lcd_putc(unsigned char dat){ //สร้างฟังก์ชันชื่อ lcd_putc เพื่อส่งข้อมูลการแสดงผลให้ LCD lcd_port = (1<<rs_lcd); //ให้ขา RS ของ LCD มีค่าเป็นลอจิก '1' lcd_port = (lcd_port & 0x0f) (dat & 0xf0); //ส่งค่าข้อมูลการแสดงผล 4 บิตบนให้แก่ LCD lcd_busy(2); //เรียกใช้งานฟังก์ชัน lcd_busy และให้ค่าฟังก์ชันเท่ากับ 2 lcd_port = (lcd_port & 0x0f) (dat << 4); //ส่งค่าข้อมูลการแสดงผล 4 บิตล่างให้แก่ LCD lcd_busy(2); //เรียกใช้งานฟังก์ชัน lcd_busy และให้ค่าฟังก์ชันเท่ากับ 2 lcd_port = (1<<rs_lcd); //ให้ขา RS ของ LCD มีค่าเป็นลอจิก '1' } void lcd_puts(char *str){ //สร้างฟังก์ชันชื่อ lcd_puts เพื่อส่งชุดข้อความให้แก่ LCD while(*str != '\0') lcd_putc(*str++); //วนรอบเรียกใช้งาน lcd_putc จนกว่าส่งข้อความครบทุกตัว } void lcd_init(){ //สร้างฟังก์ชันชื่อ lcd_init เพื่อกำหนดค่าเริ่มต้นการทำงานของ LCD lcd_ddr = 0b11111100; //กำหนดให้ขาของไมโครคอนโทรลเลอร์ที่ต่อกับ LCD เป็นขาเอาต์พุต lcd_port = (1<<en_lcd); //ให้ขา Enable ของ LCD มีค่าเป็นลอจิก '1' lcd_port = (1<<rs_lcd); //ให้ขา RS ของ LCD มีค่าเป็นลอจิก '1' lcd_command(0x33); //ให้ LCD ติดต่อกับ MCU แบบ 8 บิต และทำการหน่วงเวลา lcd_command(0x32); //ให้ LCD ติดต่อกับ MCU แบบ 8 บิต และทำการหน่วงเวลา lcd_command(0x28); //ให้ LCD ติดต่อกับ MCU แบบ 4 บิต และขนาดตัวอักษรเท่ากับ 5*7 Dot lcd_command(0x0c); //ให้ LCD ทำการแสดงผลได้โดยไม่ให้แสดงเคอร์เซอร์ lcd_command(0x06); //เมื่อมีการแสดงผลทำให้ DDRAM + 1 ตำแหน่ง และเคอร์เซอร์จะถูกเลื่อนไปทางขวามือ lcd_command(0x01); //เคลียร์หน้าจอการแสดงผล และเคอร์เซอร์กลับไปยังตำแหน่ง home _delay_ms(500); //หน่วงเวลา 500 มิลลิวินาที }</pre>			

	ใบเนื้อหา	หน้าที่ 31
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์		
<pre>int main(){//สร้างฟังก์ชัน main ของโปรแกรม init_lcd(); //เรียกใช้งานฟังก์ชัน init_lcd เพื่อกำหนดค่าเริ่มต้นของ LCD lcd_command(0x80); //เรียกใช้งานฟังก์ชัน lcd_command เพื่อกำหนดตำแหน่งการแสดงผล //บรรทัดแรกตำแหน่งแรก lcd_puts("Test LCD16*2"); //เรียกใช้งานฟังก์ชัน lcd_puts เพื่อแสดงข้อความ "Test LCD16*2" //บนบรรทัดแรก lcd_command(0xc0); //เรียกใช้งานฟังก์ชัน lcd_command เพื่อกำหนดตำแหน่งการแสดงผล //บรรทัดที่สองตำแหน่งแรก lcd_puts("By ATMEGA32"); //เรียกใช้งานฟังก์ชัน lcd_puts เพื่อแสดงข้อความ "By ATMEGA32" //บนบรรทัดที่สอง while(1){ //วนรอบแบบไม่รู้จบเนื่องจากเงื่อนไขของคำสั่ง while เป็นจริงตลอดเวลา } return 0; }</pre>		

	แบบฝึกหัด	หน้าที่ 1
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	
ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์		
<p><u>คำสั่ง</u> จงตอบคำถามต่อไปนี้ให้ถูกต้อง</p> <ol style="list-style-type: none"> จงอธิบายวิธีการสร้างสัญญาณนาฬิกาด้วยไมโครคอนโทรลเลอร์ AT89C51RD2 จงอธิบายวิธีการสร้างสัญญาณนาฬิกาด้วยไมโครคอนโทรลเลอร์ PIC16F887 จงอธิบายวิธีการสร้างสัญญาณนาฬิกาด้วยไมโครคอนโทรลเลอร์ ATMEGA32 จงอธิบายวิธีการต่อ LED เข้ากับไมโครคอนโทรลเลอร์ในรูปแบบ Active Low จงอธิบายวิธีการต่อ LED เข้ากับไมโครคอนโทรลเลอร์ในรูปแบบ Active High จงอธิบายวิธีการต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์ในรูปแบบ latch จงอธิบายวิธีการต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์ในรูปแบบมัลติเพล็กซ์สัญญาณ 		

	แบบฝึกหัด	หน้าที่ 2
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 3
	ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป	

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

8. การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ Character LCD มีกี่รูปแบบ อะไรบ้าง

.....

9. จอแสดงผล Character LCD ขนาด 20x4 สามารถแสดงผลได้สูงสุดกี่ตัวอักษร จำนวนกี่บรรทัด และบรรทัดละกี่ตัวอักษร

.....

10. คำสั่ง 0x28 ในการควบคุมจอ Character LCD มีความหมายอย่างไร

.....

.....

.....

.....

.....

.....

.....

.....