

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### รหัสวิชา 30127-2004 (2-3-3) ดิจิทัลและไมโครคอนโทรลเลอร์

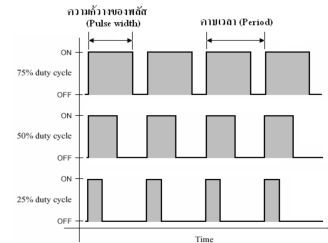
Digital And Microcontroller

1

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 1. การสร้างสัญญาณนาฬิกาด้วยโมดูล PWM

PWM หรือ Pulse Width Modulation คือสัญญาณพัลส์ที่มีค่าความถี่คงที่ แต่ความกว้างของพัลส์เปลี่ยนแปลงได้ตามความต้องการของผู้ใช้งาน



รูปที่ 1.1 ตัวอย่างสัญญาณ PWM

Digital And Microcontroller

4

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

1. การสร้างสัญญาณนาฬิกาด้วยโมดูล PWM
  - 1.1 ลักษณะการทำงานของโมดูล PWM
  - 1.2 การกำหนดค่าความถี่และดีวตี้ไซเคิลของโมดูล PWM
  - 1.3 ตัวอย่างการสร้างสัญญาณ PWM ด้วยไมโครคอนโทรลเลอร์
2. วงจรแสดงผลด้วยอุปกรณ์ LED
  - 2.1 การต่อวงจร LED ในรูปแบบ Active High
  - 2.2 การต่อวงจร LED ในรูปแบบ Active Low
  - 2.3 ตัวอย่างการต่อวงจรใช้งาน LED ในรูปแบบไฟวิ่ง 8 บิต

Digital And Microcontroller

2

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 1.1 ลักษณะการทำงานของโมดูล PWM

ลักษณะการทำงานของโมดูล PWM หรือ Pulse Width Modulation ในตัวไมโครคอนโทรลเลอร์ คือโมดูลที่ใช้สำหรับสร้างสัญญาณพัลส์ที่มีค่าความถี่คงที่ แต่ความกว้างของพัลส์สามารถเปลี่ยนแปลงได้ตามความต้องการของผู้ใช้งาน โดยโมดูล PWM ของไมโครคอนโทรลเลอร์ส่วนใหญ่จะเป็นฟังก์ชันย่อยของโมดูลพิเศษในไมโครคอนโทรลเลอร์เบอร์นั้น ๆ ซึ่งเป็นฟังก์ชันย่อยของโมดูลพิเศษเกี่ยวกับการนับ เพื่อให้โมดูลเกี่ยวกับการนับเป็นตัวสร้างฐานเวลาในการกำหนดค่าความถี่ของสัญญาณ PWM และใช้รีจิสเตอร์พิเศษอีกตัวหนึ่งเพื่อทำการเปรียบเทียบค่าของการนับในการกำหนดค่าดีวตี้ไซเคิลของสัญญาณ PWM

Digital And Microcontroller

5

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

3. การแสดงผลบน 7-Segment ด้วยไมโครคอนโทรลเลอร์
  - 3.1 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์จำนวน 1 หลัก
  - 3.2 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์มากกว่า 1 หลัก
4. การแสดงผลบน Character LCD ด้วยไมโครคอนโทรลเลอร์
  - 4.1 คุณสมบัติของจอ LCD
  - 4.2 รูปแบบการเชื่อมต่อ Character LCD กับไมโครคอนโทรลเลอร์
  - 4.3 การเขียนโปรแกรมแสดงผลบน Character LCD ขนาด 16x2 ด้วยไมโครคอนโทรลเลอร์

Digital And Microcontroller

3

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 1.1.1 การกำหนดลักษณะการทำงานของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51ED2 หรือ AT89C51RD2 มีโมดูล PWM ที่เป็นฟังก์ชันย่อยของโมดูลพิเศษ PCA (Programmable Counter Array) จำนวน 5 วงจร โดยโมดูล PCA ซึ่งโมดูล PCA ที่สามารถสร้างสัญญาณ PWM ได้คือ ขา P1.3 – P1.7 (CEX0 – CEX4) โดยสัญญาณ PWM ที่สร้างขึ้นจะมีความถี่เดียวกัน แต่ดีวตี้ไซเคิลแตกต่างกัน โดยมีรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดลักษณะการทำงานของโมดูล PWM ได้แก่

CMOD คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่กำหนดแหล่งกำเนิดสัญญาณนาฬิกาของโมดูล PCA โดยส่วนที่เกี่ยวข้องกับการทำงานของ PWM คือบิต 2 และบิต 1 (CPS1 และ CPS0)

Digital And Microcontroller

6

การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

ตารางที่ 1.1 การกำหนดค่าบิต CPS1 และ CPS0 เพื่อเลือกแหล่งจ่ายสัญญาณนาฬิกาให้แก่ PCA

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

CCON คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ควบคุมการทำงานของโมดูล PCA โดยส่วนที่เกี่ยวข้องกับการทำงานของ PWM คือบิต 6 บิต CR (PCA Counter Run Control bit) ทำหน้าที่ปิดเปิดการทำงานของโมดูล PCA

CCAPMn คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่กำหนดลักษณะการทำงานของโมดูล PCA สามารถกำหนดให้โมดูล PCA สร้างสัญญาณ PWM ในรูปแบบ 8-bit PWM คือใน 1 คาบเวลามีความละเอียดของการนับที่ 256 ค่า

Digital And Microcontroller

7

การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

1.1.2 การกำหนดลักษณะการทำงานของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล PIC16F เบอร์ PIC16F877 หรือ PIC16F887

ไมโครคอนโทรลเลอร์ PIC16F ขนาด 40 เบอร์ PIC16F877 หรือ PIC16F887 จะมีโมดูล PWM ให้ใช้งาน 2 ชุด ได้แก่ โมดูล CCP1 และโมดูล CCP2 โดยสามารถใช้กำเนิดสัญญาณ PWM ที่มีความละเอียดสูงสุดขนาด 10 บิต ซึ่งจะทำงานร่วมกับ TIMER2 และมีรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดลักษณะการทำงานของโมดูล PWM ได้แก่

PR2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าในสูตรเพื่อคำนวณหา คาบเวลาของสัญญาณ PWM ที่สร้างขึ้น โดย

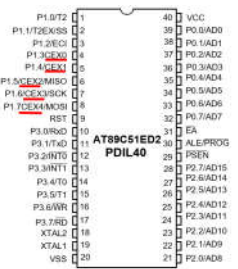
$$PWM \text{ Period} = (PR2 + 1) \times 4 \times T_{OSC} \times (TMR2 \text{ Prescale Value})$$

Digital And Microcontroller

10

การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

CCAPnH และ CCAPnL คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่า ตัวชี้เขตโดย CCAPnH จะเป็นรีจิสเตอร์สำหรับเก็บค่าคงที่ของการเปรียบเทียบสำหรับการสร้างคาบเวลาของตัวชี้เขต ส่วนรีจิสเตอร์ CCAPnL จะเป็นรีจิสเตอร์สำหรับการนับ



รูปที่ 1.2 แสดงตำแหน่งขาสัญญาณที่สามารถสร้างรูปคลื่น PWM ของไมโครคอนโทรลเลอร์ AT89C51ED2

Digital And Microcontroller

8

การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

T2CON คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าการนับของ Timer 2 เพื่อสร้างคาบเวลาของคลื่น PWM ที่สร้างขึ้นใช้เฉพาะบิต TMR2ON, T2CKPS1 และ T2CKPS0 โดยบิต TMR2ON คือบิตที่ทำหน้าที่ในการเปิดเปิดการทำงานของโมดูล Timer 2 ส่วนบิต T2CKPS1 และ T2CKPS0 ทำหน้าที่ในการกำหนดค่า 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

Digital And Microcontroller

11

การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

7	6	5	4	3	2	1	0
-	ECOMn	CAPPn	CAPnN	MATn	TOGn	PWMn	ECCFn
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

Digital And Microcontroller

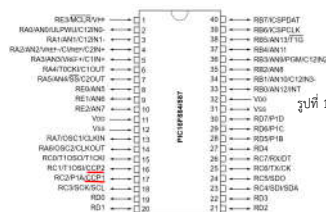
9

การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

CCPRxL คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดค่าตัวชี้เขต 8 บิตบน

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

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



รูปที่ 1.6 แสดงตำแหน่งขาสัญญาณที่สามารถสร้างรูปคลื่น PWM ของไมโครคอนโทรลเลอร์ PIC16F887

Digital And Microcontroller

12

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

#### 1.1.3 การกำหนดลักษณะการทำงานของโมดูล PWM ในไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32

ไมโครคอนโทรลเลอร์ ATMEGA32 ขนาด 40 ขาได้เตรียมโมดูล PWM สำหรับสร้างสัญญาณ PWM มาพร้อมใช้งานจำนวน 4 ชุด โดยมีความละเอียดขนาด 8 บิต จาก Timer 0 จำนวน 1 ชุดออกที่ขาสัญญาณ 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

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของโมดูล PWM เนื่องจาก Timer 0 มีรีจิสเตอร์ที่เกี่ยวข้องได้แก่

TCCR0 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดโหมดการทำงานของโมดูล Timer 0 ให้ทำงานในลักษณะต่าง ๆ และกำหนดค่า Prescale ของ Timer 0

Digital And Microcontroller

13

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

TCNT0 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับคาบเวลาของโมดูล Timer 0 เพื่อสร้างคาบเวลาของสัญญาณ PWM

OCR0 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับเพื่อใช้ในการเปรียบเทียบกับ TCNT0 เพื่อกำหนดค่าดีวี่ไซเคิลของสัญญาณ PWM

ในโหมด PWM, Phase Correct สามารถคำนวณหาความสัมพันธ์ของสัญญาณ PWM จากสมการ

$$f_{OCnPCPWM} = \frac{f_{clk\_IO}}{N \cdot 510}$$

ในโหมด Fast PWM สามารถคำนวณหาความสัมพันธ์ของสัญญาณ PWM จากสมการ

$$f_{OCnPWM} = \frac{f_{clk\_IO}}{N \cdot 256}$$

เมื่อ N คือค่าของ Prescaler ในรูปที่ 1.11

Digital And Microcontroller

16

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

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	

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

Digital And Microcontroller

14

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของโมดูล PWM เนื่องจาก Timer 1 มีรีจิสเตอร์ที่เกี่ยวข้องได้แก่

TCCR1A คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดโหมดการทำงานของโมดูล Timer 1 ให้ทำงานในลักษณะต่าง ๆ ร่วมกับรีจิสเตอร์ TCCR1B

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

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	

7	6	5	4	3	2	1	0	
ICN11	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	

Digital And Microcontroller

17

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

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 <sub>in</sub> (No prescaling)
0	1	0	clk <sub>in</sub> /8 (From prescaler)
0	1	1	clk <sub>in</sub> /64 (From prescaler)
1	0	0	clk <sub>in</sub> /256 (From prescaler)
1	0	1	clk <sub>in</sub> /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

Digital And Microcontroller

15

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 15. Toggle OC1A or 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 Control PWM Mode

Digital And Microcontroller

18

### 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	0x0FFF	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	0x0FFF	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

Digital And Microcontroller

19

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของโมดูล PWM เนื่องจาก Timer 2 มีรีจิสเตอร์ที่เกี่ยวข้องได้แก่

TCCR2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการกำหนดโหมดการทำงานของโมดูล Timer 2 ให้ทำงานในลักษณะต่าง ๆ และกำหนดค่า Prescale ของ Timer 2

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	

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

Digital And Microcontroller

22

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk <sub>IO</sub> /1 (No prescaling)
0	1	0	clk <sub>IO</sub> /8 (From prescaler)
0	1	1	clk <sub>IO</sub> /64 (From prescaler)
1	0	0	clk <sub>IO</sub> /256 (From prescaler)
1	0	1	clk <sub>IO</sub> /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

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

OCR1AH , OCR1AL และ OCR1BH , OCR1BL คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับเพื่อใช้ในการเปรียบเทียบค่ากับ TCNT1 เพื่อกำหนดค่าดีวี่ไซเคิลของสัญญาณ PWM เมื่อรวมกันจะได้ความละเอียดของการนับ 16 บิต

Digital And Microcontroller

20

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

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

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

Digital And Microcontroller

23

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

ในโหมด Fast PWM Mode ของ Timer 1 สามารถคำนวณหาความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnPWM} = \frac{f_{clk\_IO}}{N \cdot (1 + TOP)}$$

ในโหมด Phase Correct PWM Mode ของ Timer 1 สามารถคำนวณหาความถี่ของสัญญาณ PWM จากสมการ

$$f_{OCnPCPWM} = \frac{f_{clk\_IO}}{2 \cdot N \cdot TOP}$$

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

$$f_{OCnPFPCPWM} = \frac{f_{clk\_IO}}{2 \cdot N \cdot TOP}$$

เมื่อ N คือค่าของ Prescaler ในรูปที่ 1.17

Digital And Microcontroller

21

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk <sub>IO</sub> /1 (No prescaling)
0	1	0	clk <sub>IO</sub> /8 (From prescaler)
0	1	1	clk <sub>IO</sub> /32 (From prescaler)
1	0	0	clk <sub>IO</sub> /64 (From prescaler)
1	0	1	clk <sub>IO</sub> /128 (From prescaler)
1	1	0	clk <sub>IO</sub> /256 (From prescaler)
1	1	1	clk <sub>IO</sub> /1024 (From prescaler)

รูปที่ 1.22 รูปตารางกำหนดแหล่งกำเนิดสัญญาณนาฬิกาของ Timer 2 ด้วย

รีจิสเตอร์ TCCR2 โดยบิต CS22 ,CS21 และ CS20

TCNT2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับคาบเวลาของโมดูล Timer 2 เพื่อสร้างคาบเวลาของสัญญาณ PWM

OCR2 คือรีจิสเตอร์ขนาด 8 บิตที่ทำหน้าที่ในการนับเพื่อใช้ในการเปรียบเทียบค่ากับ TCNT2 เพื่อกำหนดค่าดีวี่ไซเคิลของสัญญาณ PWM

Digital And Microcontroller

24



## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 1.2.3.2 การกำหนดค่าตัวชี้แจงของโมดูล PWM ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32

การกำหนดค่าตัวชี้แจงสามารถทำได้ด้วยการกำหนดค่าในรีจิสเตอร์ OCRx ซึ่งการให้ค่าเรจิสเตอร์ OCRx จะมีความหมายไม่เหมือนกัน ขึ้นอยู่กับโหมด และหน้าที่การทำงานของซายสัญญาณ OCRx ในการสร้างสัญญาณ PWM เช่นถ้าเราเลือกโหมดการทำงานแบบ Fast PWM Mode กำหนดให้บิต COMx1 เป็นลอจิก '1' และ บิต COMx0 เป็นลอจิก '0' ซายสัญญาณ OCRx จะเป็นสัญญาณลอจิก '1' เมื่อค่าข้อมูลของ OCRx มีค่ามากกว่าค่าข้อมูลของ TCNT0 ดังนั้นค่าของ OCRx เพื่อสร้างช่วงเวลาของตัวชี้แจงหาได้จาก

$$OCRx = 255 * (\text{duty} / 100)$$

เมื่อ duty มีค่า 0 -100 % โดยสมการนี้ใช้ได้กับขา OC0 และ OC2

และถ้ากำหนดให้บิต COMx1 เป็นลอจิก '1' และ บิต COMx0 เป็นลอจิก '1' ซายสัญญาณ OCRx จะเป็นสัญญาณลอจิก '1' เมื่อค่าข้อมูลของ OCRx มีค่าน้อยกว่าหรือเท่ากับค่าข้อมูลของ TCNT0 ดังนั้นค่าของ OCRx เพื่อสร้างช่วงเวลาของตัวชี้แจงหาได้จาก

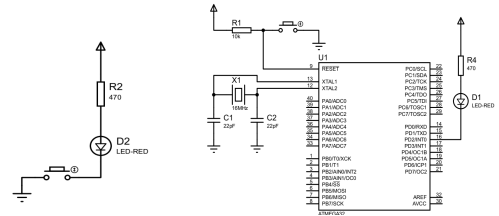
$$OCRx = 255 - (255 * (\text{duty} / 100))$$

Digital And Microcontroller

31

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 2.2 การต่อวงจร LED ในรูปแบบ Active Low



Digital And Microcontroller

34

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

ตัวอย่าง การสร้างนาฬิกาด้วยโมดูล PWM ของไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA32 ที่ความถี่ 3.9kHz Duty Cycle 50% โดยใช้ซายสัญญาณ OCR0

```
void pwmOnOCR0() {
```

```
    TCCR0 |= 0x62; //กำหนดให้โมดูล Timer0 ทำงานในรูปแบบ PWM Phase Correct Mode
```

```
    // Timer0 นับ 0 - 255 ค่า ใช้ Prescaler 1/8 และซาย OCR0 จะเป็นลอจิก
```

```
    // '1' เมื่อค่าของ OCR0 น้อยกว่าค่า Timer0 เป็นผลให้ความถี่ที่ซาย OCR0
```

```
    // มีค่าเท่ากับ 16Mz/(8*510) = 3,921.57Hz
```

```
    OCR0 = (255 * 50) / 100; // กำหนดค่าความเร็วตัวชี้แจงเท่ากับ 50%
```

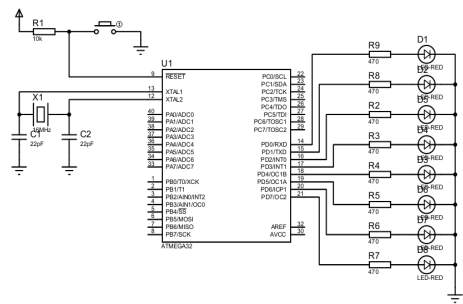
```
}
```

Digital And Microcontroller

32

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 2.3 ตัวอย่างการต่อวงจรใช้งาน LED ในรูปแบบไฟวิ่ง 8 บิต



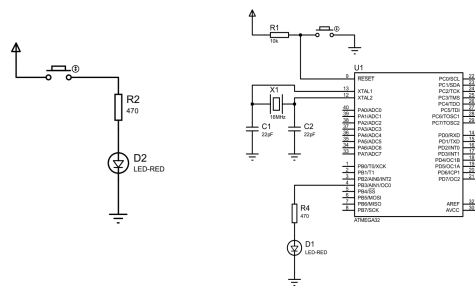
Digital And Microcontroller

35

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 2. วงจรแสดงผลด้วยอุปกรณ์ LED

#### 2.1 การต่อวงจร LED ในรูปแบบ Active High



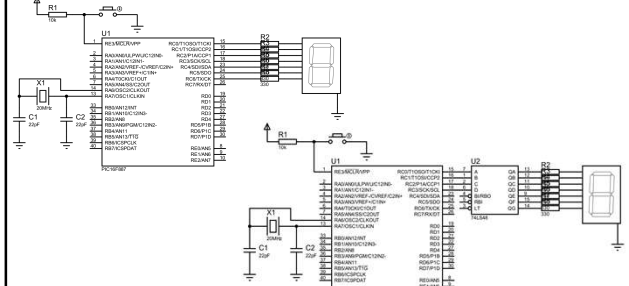
Digital And Microcontroller

33

## การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

### 3. การแสดงผลบน 7-Segment ด้วยไมโครคอนโทรลเลอร์

#### 3.1 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์จำนวน 1 หลัก

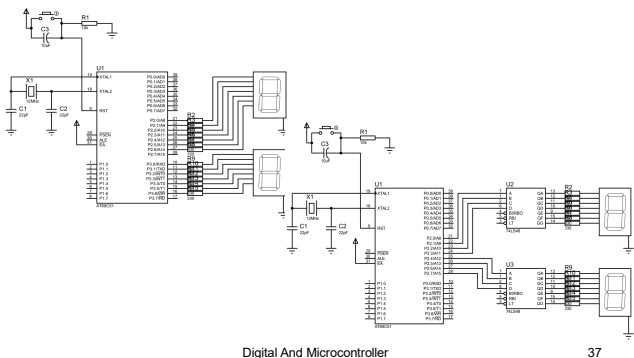


Digital And Microcontroller

36

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

#### 3.2 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์มากกว่า 1 หลัก



Digital And Microcontroller

37

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

#### 4. การแสดงผลบน Character LCD ด้วยไมโครคอนโทรลเลอร์

##### 4.1 คุณสมบัติของจอ LCD

LCD ย่อมาจากคำว่า Liquid Crystal Display โดยจอแอลซีดีจะประกอบด้วยแผ่นแก้ว 2 แผ่นประกบกันเว้นช่องว่างตรงกลางไว้ประมาณ 6-10 ไมโครเมตร ด้านในจะเคลือบด้วยตัวนำไฟฟ้าแบบใสไว้ระหว่างตัวนำไฟฟ้าแบบใส และจะมีโมเลกุลผลึกรวมตัวกันในทิศทางที่แสงส่องผ่าน โดยอุปกรณ์การแสดงผลจอ LCD จะแบ่งเป็น 2 แบบใหญ่ ๆ ตามลักษณะการแสดงผลดังนี้

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

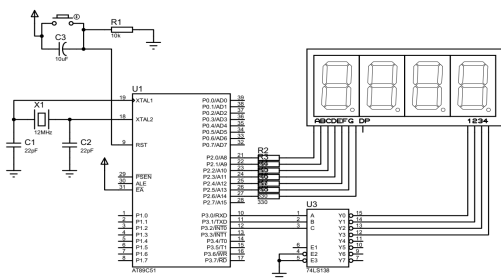
2. **Graphic LCD** เป็นจอแสดงผลที่สามารถกำหนดได้ว่าจะให้จุดบนหน้าจอ ณ ตำแหน่งใด ๆ สามารถกันแสง หรือปล่อยแสงออกมา ทำให้จอแสดงผลสามารถสร้างรูปภาพหรือตัวอักษรขึ้นมาบนหน้าจอได้ ส่วนขนาดของ **Graphic LCD** จะทำการระบุขนาดได้โดยระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอ **Graphic LCD**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

Digital And Microcontroller

40

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

#### 3.2 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์มากกว่า 1 หลัก (ต่อ)

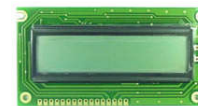


Digital And Microcontroller

38

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

#### 4.2 รูปแบบการเชื่อมต่อ Character LCD กับไมโครคอนโทรลเลอร์



Character LCD จะมีขาให้ต่อใช้งานทั้งหมด 16 ขาดังนี้

ขาที่ 1 คือขา VSS ใช้สำหรับต่อกับขา GND ของแหล่งจ่ายไฟ  
ขาที่ 2 คือขา VDD ใช้สำหรับต่อกับขาไฟบวก 5V ของแหล่งจ่ายไฟ  
ขาที่ 3 คือขา VO เป็นขาที่ใช้สำหรับปรับความคมชัดของตัวอักษร  
ขาที่ 4 คือขา RS เป็นขาอินพุตที่ใช้สำหรับกำหนดว่าข้อมูลที่เข้ามาทางขาคือของ LCD เป็นข้อมูลที่จะใช้แสดงผล หรือข้อมูลชุดคำสั่ง  
ขาที่ 5 คือขา R/W เป็นขาอินพุตที่ใช้กำหนดว่าการเชื่อมต่อระหว่าง LCD กับไมโครคอนโทรลเลอร์จะเป็นลักษณะของการอ่านข้อมูล หรือการเขียนข้อมูลให้กับ LCD

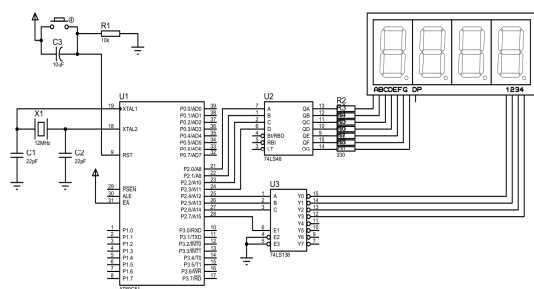
ขาที่ 6 คือขา E เป็นขาอินพุตที่ใช้สำหรับกำหนดให้ LCD เริ่มทำงานตามคำสั่ง  
ขาที่ 7 - 14 คือขา D0-D7 เป็นขาอินพุตที่ทำหน้าที่ในการรับข้อมูลที่เป็นชุดคำสั่ง และข้อมูลที่ต้องการแสดงผลออกจอ LCD  
ขาที่ 15 คือขา A เป็นขา Anode ของ LED Black light  
ขาที่ 16 คือขา K เป็นขา Cathode ของ LED Black light

Digital And Microcontroller

41

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

#### 3.2 การต่อวงจร 7-Segment กับไมโครคอนโทรลเลอร์มากกว่า 1 หลัก (ต่อ)



Digital And Microcontroller

39

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับจอ Character LCD สามารถกระทำได้ 2 วิธีคือ

1. การเชื่อมต่อในรูปแบบการใช้ขาสัญญาณ data จำนวน 4 ขา ได้แก่ขา D4 - D7

1.1 ใช้ขาสัญญาณควบคุมจำนวน 2 ขา คือขา RS และ EN ส่วนขา R/W ต่อลงกราวด์ ซึ่งการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผลได้อย่างเดียว และใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 6 ขา

1.2 ใช้ขาสัญญาณควบคุมจำนวน 3 ขา คือขา RS , EN และขา R/W โดยการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผล และอ่านข้อมูลการแสดงผลได้ ซึ่งใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 7 ขา

2. การเชื่อมต่อในรูปแบบการใช้ขาสัญญาณ data จำนวน 8 ขา ได้แก่ขา D0 - D7

2.1 ใช้ขาสัญญาณควบคุมจำนวน 2 ขา คือขา RS และ EN ส่วนขา R/W ต่อลงกราวด์ ซึ่งการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผลได้อย่างเดียว และใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 10 ขา (2 พอร์ต)

2.2 ใช้ขาสัญญาณควบคุมจำนวน 3 ขา คือขา RS , EN และขา R/W โดยการต่อในลักษณะนี้สามารถสั่งงานให้ Character LCD สามารถแสดงผล และอ่านข้อมูลการแสดงผลได้ ซึ่งใช้ขาสัญญาณจากไมโครคอนโทรลเลอร์มาควบคุม LCD ทั้งหมด 11 ขา (2 พอร์ต)

Digital And Microcontroller

42



### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

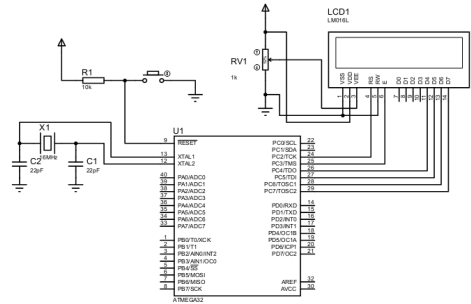
ตารางที่ 1.3 ตารางแสดงความสัมพันธ์ในการทำงานของขาสัญญาณ RS,R/W และ E ของ Character LCD

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

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

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์



รูปที่ 1.36 แสดงวงจรการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ ATMEGA32 กับ Character LCD

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

Instruction	Instruction Code								Description Instruction Code	Execution time (@27MHz)			
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2					
Clear Display	0	0	0	0	0	0	0	0	1	Write "00H" 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	UD	SH	Set cursor moving direction, cursor and blanking of cursor's control bit.	30µs	
Display ON/OFF Control	0	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blanking of cursor's control bit.	30µs
Cursor or Display Shift	0	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.	30µs
Function Set	0	0	0	0	1	DL	N	F	X	X	X	Set interface data length (DL), 4/5-bit, numbers of display line (N), 1/2/4 lines, display font (F), 5/7/9 dots (X), 1/16 dots (X).	30µs
Set DDRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0		Set DDRAM address in address counter.	30µs
Set DDRAM Address	0	0	1	AD5	AD4	AD3	AD2	AD1	AD0			Set DDRAM address in address counter.	30µs
Read Busy Flag and Address	0	1	BF	AD5	AD4	AD3	AD2	AD1	AD0			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	DT	DS	DS	DS	DS	DS	DS	DS		Write data into internal RAM (DDRAM/CDRAM).	45µs
Read Data from RAM	1	1	DT	DS	DS	DS	DS	DS	DS	DS		Read data from internal RAM (DDRAM/CDRAM).	45µs

รูปที่ 1.34 รูปตารางชุดคำสั่งของ Character LCD

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

#### 4.3 การเขียนโปรแกรมแสดงผลบน Character LCD ขนาด 16x2 ด้วยไมโครคอนโทรลเลอร์

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

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

### การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

ชุดคำสั่งพื้นฐานที่ใช้ในการควบคุมการทำงานของ Character LCD ได้แก่ชุดคำสั่ง

1. คำสั่ง 0x01 คือคำสั่งที่ใช้ในการเคลียร์หน้าจอการแสดงผล
2. คำสั่ง 0x06 คือคำสั่งที่ใช้ในการกำหนดว่าเมื่ออ่านหรือเขียนข้อมูลให้ LCD จะทำให้ DDRAM ของ LCD เพิ่มขึ้น 1 ตำแหน่ง และเคอร์เซอร์จะเลื่อนไปทางขวามือ
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