

15 การอินเตอร์เฟสกับ Dot Matrix LCD Module

15.1 บทนำ

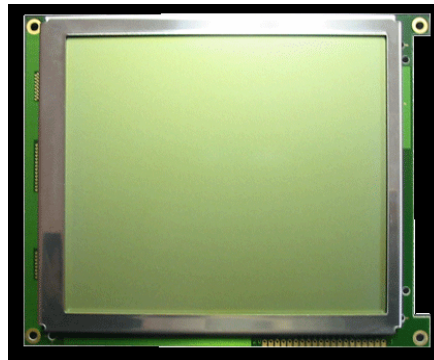
โมดูลแอลซีดี (LCD Module) เป็นอุปกรณ์แสดงผลที่สามารถแสดงได้ทั้งข้อความ ตัวเลข และรูปภาพ เหมาะสำหรับงานแสดงผลที่ต้องการให้สิ้นเปลืองพลังงานต่ำ โดยทั่วไป โมดูลแอลซีดีสามารถแบ่งออกได้เป็น 3 ประเภทได้แก่

- **Character LCD Module** แบบนี้สามารถแสดงผลได้ทั้งตัวอักษรและตัวเลข โดยจัดแบ่งเป็นแถวๆ เช่น แบบ 1 แถว แถวละ 16 ตัวอักษร หรือแบบ 4 แถว แถวละ 16 ตัวอักษรเป็นต้น ตามรูปที่ 15.1 เป็นแบบ 1 แถว 16 ตัวอักษร



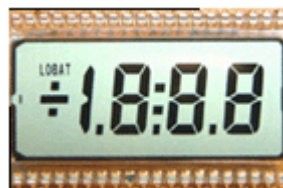
รูปที่ 15.1 โมดูลแอลซีดีแบบอักขระ

- **Graphic LCD Module** แบบนี้สามารถแสดงผลเป็นรูปภาพได้ และสามารถทำเป็นรูปภาพของอักษรแบบต่างๆได้



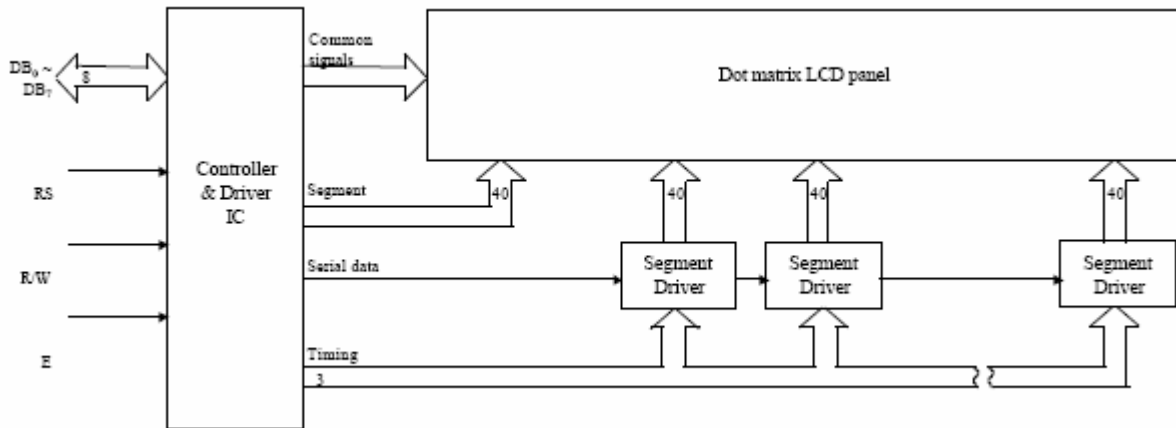
รูปที่ 15.2 โมดูลแอลซีดีแบบกราฟ

- **Segment Display LCD Module** เป็นแบบที่มีรูปแบบการแสดงผลจำกัด ไม่กี่แบบ



รูปที่ 15.3 โมดูลแอลซีดีแบบเซกเมนต์ แสดงผลเป็นตัวเลข

15.2 ส่วนประกอบของโมดูลแอลซีดี



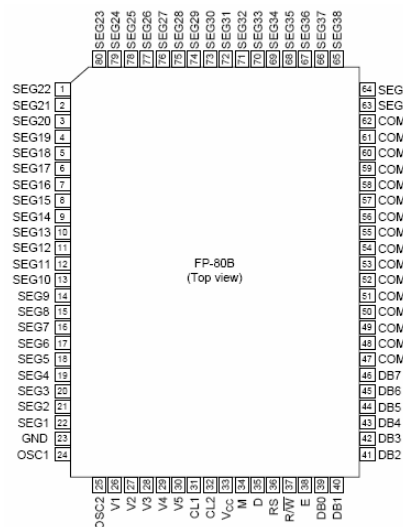
รูปที่ 15.4 บล็อกไดอะแกรมของโมดูลแอลซีดี

ส่วนประกอบของโมดูลแอลซีดีประกอบด้วยกัน 3 ส่วนคือ

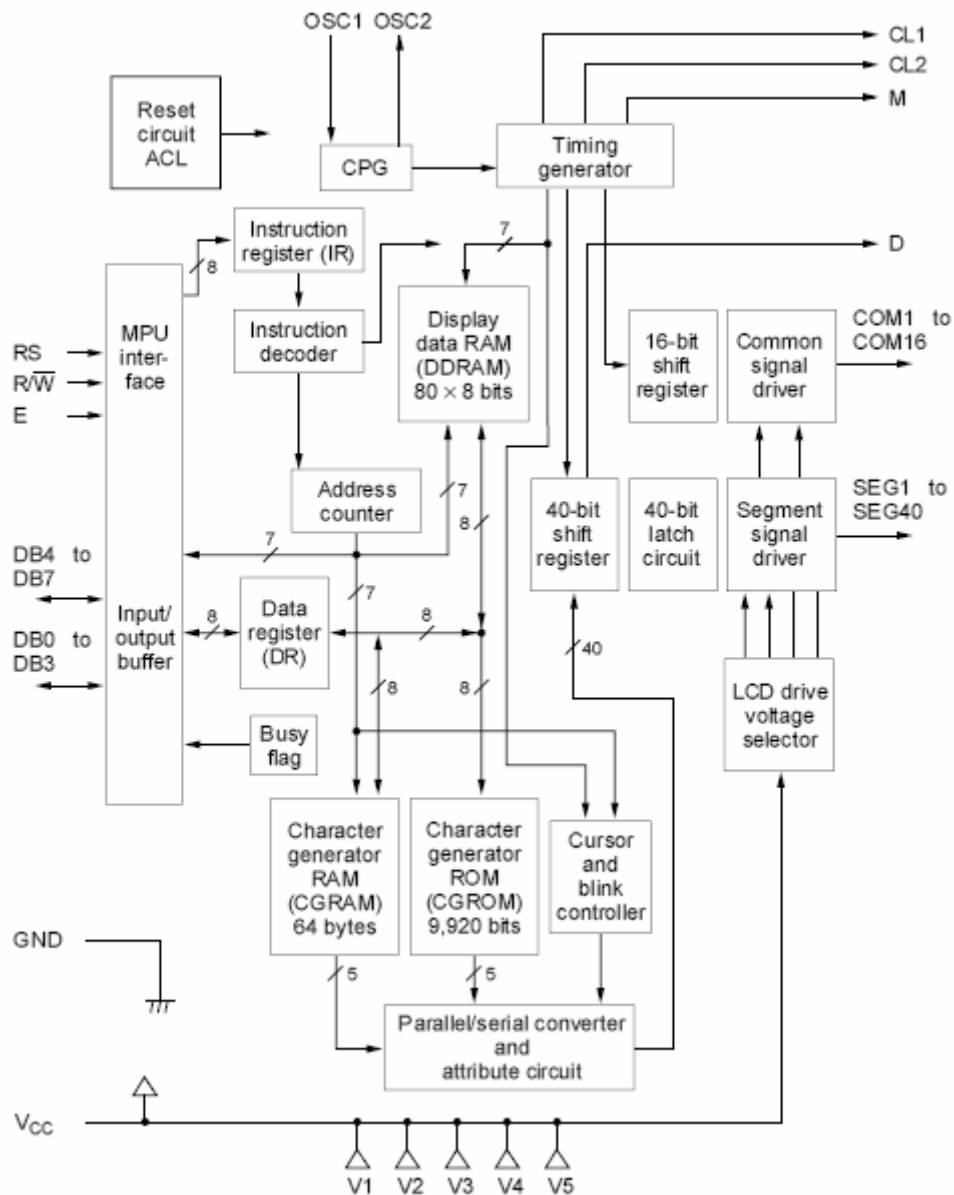
- Dot Matrix LCD เป็นตัวแสดงผล ทำงานในลักษณะของการปิดหรือเปิด ตัวเองกับแสง
- Driver เป็นตัวขับ LCD รับสัญญาณมาจากส่วนควบคุม เบอร์ที่นิยมใช้ได้แก่ HD44100H และ MSM5259
- Controller เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอก แล้วควบคุมการทำงานของ LCD เบอร์ที่นิยมใช้ สำหรับแบบ Character ได้แก่ HD4478 ส่วนแบบ Graphic ได้แก่ HD61830

15.2.1 HD44780

HD 44780 เป็นส่วนควบคุมของแอลซีดีโมดูลที่นิยมใช้กันมาก มีรูปแบบการจัดขาและบล็อกไดอะแกรมตามรูปที่ 15.5 และ 15.6 ตามลำดับ



รูปที่ 15.5 รูปแบบและการจัดตำแหน่งขาของ HD44780



รูปที่ 15.6 บล็อกไดอะแกรมของ HD44780

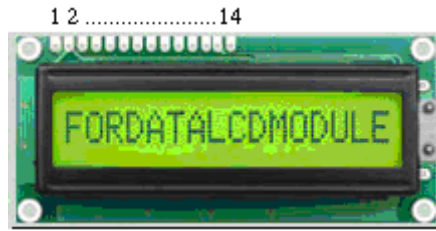
คุณสมบัติพอสั่งของ HD44780

- จำนวนจุดต่อตัวอักษรทำได้ทั้งแบบ 5 x 8 และ 5 x 10
- ใช้แหล่งจ่ายกำลังงานต่ำ 2.7 - 5.5 โวลท์
- ให้กำลังขับแอลซีดีได้ย่านกว้าง 3.0 - 11 โวลท์
- ใช้กับระบบบัสของซีพียูได้ความถี่ถึง 2 MHz (ที่ VCC = 5V)
- สามารถอินเตอร์เฟสกับซีพียูแบบ 4- บิตหรือ 8 บิตก็ได้
- หน่วยความแรมสำหรับการแสดงผล (DDRAM = Display Data RAM) มีขนาดสูงสุด 80 X 8-bit (80 ตัวอักษร)

- คาแรคเตอร์ เจนเนอเรเตอร์รอม (CGROM = Character Generator ROM) มีขนาด 9,920 บิต ซึ่งใช้สำหรับเป็น ตัวอักษร 240 ตัว
 - ตัวอักษรขนาด 5 X 8 จุด ได้ 208 ตัว
 - ตัวอักษรขนาด 5 X 10 จุด ได้ 32 ตัว
- มีแรมภายในสำหรับคาแรคเตอร์ เจนเนอเรเตอร์ (CGRAM = Character Generator RAM) ขนาด 64 x 8-bit
 - 8 character fonts (5 X 8 จุด)
 - 4 character fonts (5 X 10 จุด)
- ตัวขับภาคแสดงผลมีขนาด 16-common X 40-segment
- มีคำสั่งควบคุมการทำงานหลายอย่าง
 - Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- มีการรีเซ็ตวงจร โดยอัตโนมัติเมื่อเริ่มจ่ายไฟเลี้ยง
- มีวงจรออสซิลเลเตอร์ภายในที่ใช้เพียงตัวต้านทานภายนอก
- กำลังงานสูญเสียต่ำ

15.2.2 ขาของโมดูลแอลซีดี

ขาที่	สัญญาณ	อินพุต/เอาทพุท	ต่อกับ	รายละเอียด
1	Vss		แหล่งจ่าย	0 V Gnd
2	Vcc		แหล่งจ่าย	+ 5V
3	Vee		แหล่งจ่าย	ใช้ปรับความสว่างของ LCD ถ้าต่อลงดินจะสว่างที่สุด
4	RS	อินพุท	ซีพียู	สัญญาณ Register Select ใช้เลือกกรีจิสเตอร์ควบคุมหรือหน่วยความจำแสดงผล
5	R/W	อินพุท	ซีพียู	สัญญาณควบคุมการอ่าน/เขียน "0" เขียนหรือส่งข้อมูลให้แก่โมดูล "1" อ่านข้อมูลจากโมดูล
6	E	อินพุท	ซีพียู	Enable - สัญญาณสั่งให้เริ่มต้นการทำงาน สำหรับการอ่าน/เขียนข้อมูล การรับส่งข้อมูลจะเกิดเมื่อเป็น '1' และขอบขาลง
7 ~ 0	DB0 ~ DB3	อินพุท/เอาทพุท	ซีพียู	เป็นบัสแบบสองทิศทางใช้สำหรับส่งถ่ายข้อมูลระหว่างซีพียูกับโมดูล
11 ~ 14	DB4 ~ DB7	อินพุท/เอาทพุท	ซีพียู	เป็นบัสแบบสองทิศทางใช้สำหรับส่งถ่ายข้อมูลระหว่างซีพียูกับโมดูล



รูปที่ 15.7 แสดงตำแหน่งขาของโมดูลแอลซีดี

สัญญาณ RS หรือ Register Select

ใช้กำหนดว่าต้องการติดต่อกับรีจิสเตอร์ควบคุมหรือหน่วยความจำแสดงผล

ถ้าเป็น "0" แสดงว่าต้องการติดต่อกับรีจิสเตอร์ควบคุม

ถ้าเป็น "1" แสดงว่าต้องการติดต่อกับรีจิสเตอร์แสดงผล

สัญญาณ R/W

เป็นสัญญาณควบคุมการอ่านหรือเขียน

ถ้าเป็น "0" แสดงว่าต้องการเขียนหรือส่งข้อมูลให้แก่โมดูล

ถ้าเป็น "1" แสดงว่าต้องการอ่านข้อมูลจากโมดูล

สัญญาณ RS และ R/W ทำงานร่วมกันโดยมีความหมายดังนี้

RS	R/W	การทำงาน
0	0	ส่งคำสั่งไปควบคุม โมดูล
0	1	อ่านแฟล็ก Busy flag หรือค่าตำแหน่ง
1	0	ส่งตัวอักขระออกแสดงผล
1	1	อ่านอักขระที่ตำแหน่ง

สัญญาณ DB0 – DB3

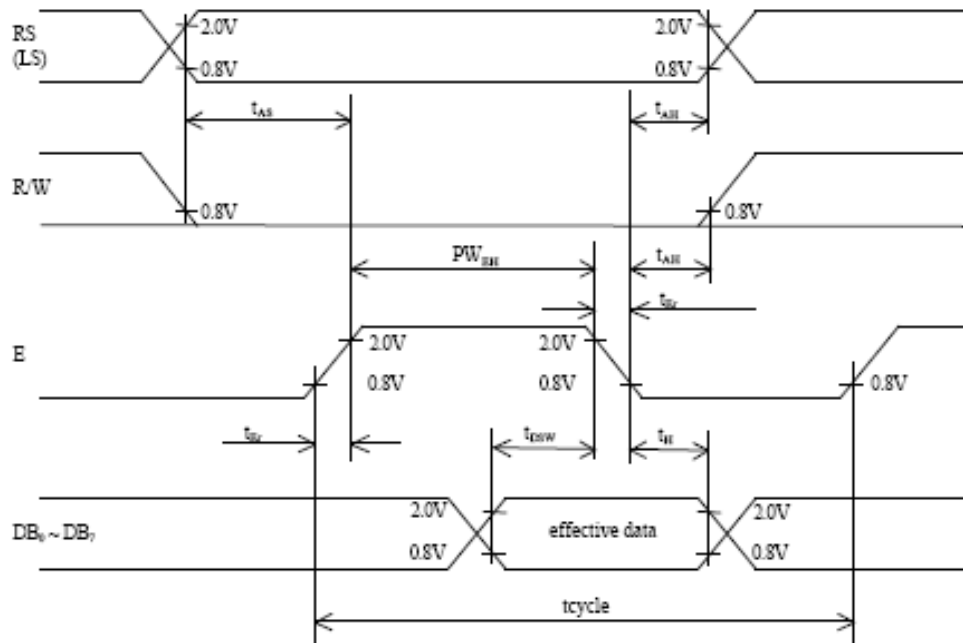
เป็นบัสแบบสองทิศทางใช้สำหรับส่งถ่ายข้อมูลระหว่างชิพกับโมดูล ในกรณีที่การทำงานเป็นแบบ 4 บิต บัสนี้ไม่ได้ใช้และควรต่อลงดินด้วย แต่ถ้าเป็นการทำงานแบบ 8 บิต บัสนี้จะเป็น 4 บิตต่ำ ใช้เพื่อการส่งถ่ายข้อมูล

สัญญาณ DB4 – DB7

เป็นบัสแบบสองทิศทางใช้สำหรับส่งถ่ายข้อมูลระหว่างชิพกับโมดูล ในกรณีที่การทำงานเป็นแบบ 4 บิต จะใช้บัสนี้ส่งถ่ายข้อมูล แต่ถ้าเป็นการทำงานแบบ 8 บิต บัสนี้จะเป็น 4 บิตสูง นอกจากนี้ DB7 ยังใช้เป็นบิตแสดงสถานะ Busy ด้วย

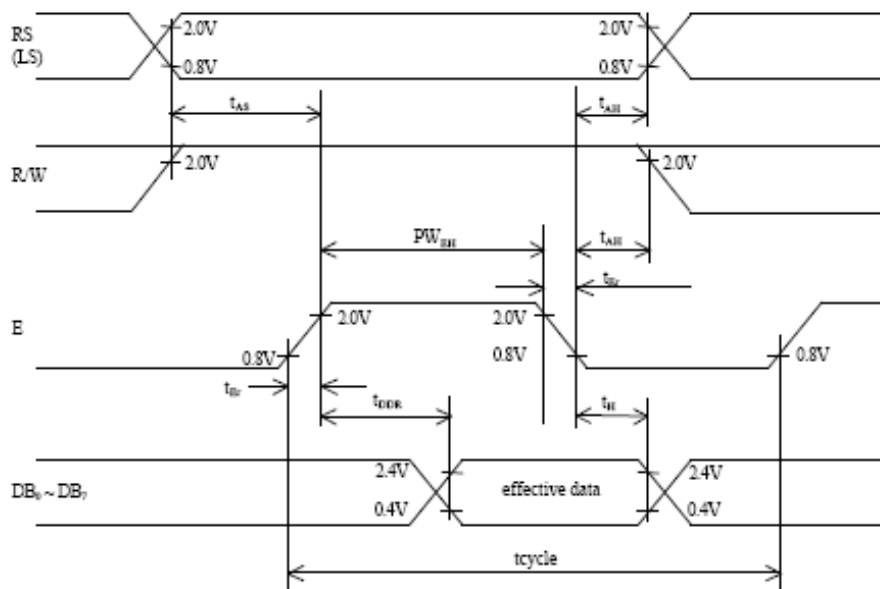
สัญญาณต่างๆเหล่านี้ ซีพียูจะใช้สำหรับติดต่อกับ โมดูล โดยมีรูปแบบการเขียนและอ่านดังนี้

การเขียนข้อมูลให้โมดูล



รูปที่ 15.8 ไคอะแกรมเวลาขณะเขียนข้อมูลให้กับ โมดูลแอลซีดี

การอ่านข้อมูลจากโมดูล



รูปที่ 15.9 ไคอะแกรมเวลาขณะอ่านข้อมูลจากโมดูลแอลซีดี

15.2.3 คำสั่งควบคุม

คำสั่งควบคุมการทำงานของ โมดูลแอลซีดีก็จะเป็นการสั่งควบคุมของตัวควบคุมนั่นเองในที่นี้ก็หมายถึงคำสั่งของ HD44780 ก่อนที่จะกล่าวถึงคำสั่ง ขอให้เข้าใจความหมายของคำต่อไปนี้ก่อน

ความหมาย

1. DDRAM (Display Data Ram) คือหน่วยความจำภายในตัวโมดูลแอลซีดีที่เป็นบัฟเฟอร์ของข้อมูล ถ้าเขียนรหัส ASCII ใดๆ ลงในหน่วยความจำนี้ก็จะปรากฏเป็นตัวอักษรที่จอแสดงผลทันที
- 2.CGRAM (Character Generator Ram) เป็นหน่วยความจำภายในตัวโมดูล ใช้สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สร้างขึ้นเอง (สร้างได้ 8 ตัว) โดยอ้างตำแหน่งได้ 64 ไบต์ (8 ตัวอักษรคูณด้วย 8 แถว)
3. การเขียนข้อมูลให้โมดูลแต่ละครั้งต้องตรวจสอบความพร้อมของโมดูล โดยตรวจสอบได้จาก Busy Flag หรือระยะเวลาการทำงานของโมดูล ซึ่งดูได้จากตารางคำสั่ง ดังนั้นเมื่อเขียนข้อมูลหนึ่งๆ ไปเราต้องหน่วงเวลารอให้โมดูลพร้อม จึงจะเขียนชุดใหม่ต่อไป
4. การเขียนข้อมูลให้โมดูลสามารถทำได้ทั้งแบบ 4 บิตและแบบ 8 บิตขึ้นอยู่กับ การเชื่อมต่อกับโมดูล ถ้าเป็นการเชื่อมแบบ 4 บิต การเขียนข้อมูลหรืออ่านข้อมูลต้องทำ 2 ครั้ง โดยครั้งแรกต้องเป็นบิต 4 ถึงบิต 7 และครั้งที่ 2 จะเป็นบิตที่ 0 ถึง 3
5. ระยะเวลาที่ให้ไว้ในตารางใช้ $F_{osc} = 250kHz$.

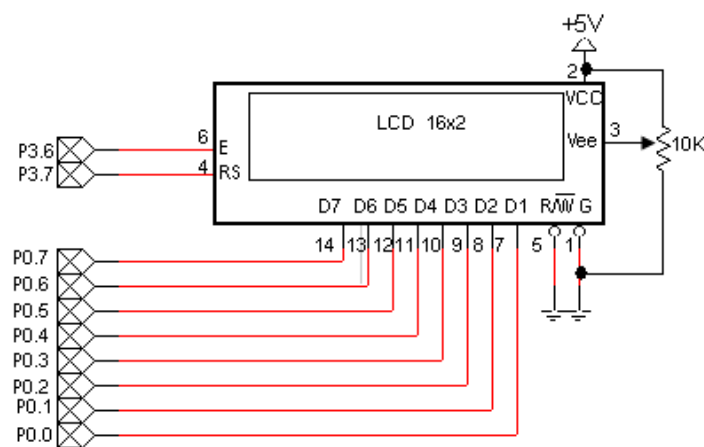
ตารางที่ 15.1 ความหมายบิตต่างๆที่ใช้ในตารางที่ 15.2

Bit name	Setting / Status	
I/D	0 = Decrement cursor position	1 = Increment cursor position
S	0 = No display shift	1 = Display shift
D	0 = Display off	1 = Display on
C	0 = Cursor off	1 = Cursor on
B	0 = Cursor blink off	1 = Cursor blink on
S/C	0 = Move cursor	1 = Shift display
R/L	0 = Shift left	1 = Shift right
DL	0 = 4-bit interface	1 = 8-bit interface
N	0 = 1/8 or 1/11 Duty (1 line)	1 = 1/16 Duty (2 lines)
F	0 = 5x7 dots	1 = 5x10 dots
BF	0 = Can accept instruction	1 = Internal operation in progress

ตารางที่ 15.2 ชุดคำสั่งของ HD44780

Instruction	Code										Description	Execution time**
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears display and returns cursor to the home position (address 0).	1.64mS
Cursor home	0	0	0	0	0	0	0	0	1	*	Returns cursor to home position (address 0). Also returns display being shifted to the original position. DDRAM contents remains unchanged.	1.64mS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write.	40uS
Display On/Off control	0	0	0	0	0	0	1	D	C	B	Sets On/Off of all display (D), cursor On/Off (C) and blink of cursor position character (B).	40uS
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	*	*	Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM contents remains unchanged.	40uS
Function set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL), number of display line (N) and character font(F).	40uS
Set CGRAM address	0	0	0	1	CGRAM address					Sets the CGRAM address. CGRAM data is sent and received after this setting.	40uS	
Set DDRAM address	0	0	1	DDRAM address					Sets the DDRAM address. DDRAM data is sent and received after this setting.	40uS		
Read busy-flag and address counter	0	1	BF	CGRAM / DDRAM address					Reads Busy-flag (BF) indicating internal operation is being performed and reads CGRAM or DDRAM address counter contents (depending on previous instruction).	0uS		
Write to CGRAM or DDRAM	1	0	write data					Writes data to CGRAM or DDRAM.	40uS			
Read from CGRAM or DDRAM	1	1	read data					Reads data from CGRAM or DDRAM.	40uS			

ตัวอย่างการเชื่อมต่อกับ MCS-51



รูปที่ 15.10 ตัวอย่างการเชื่อมต่อโมดูลแอลซีดีกับไมโครคอนโทรลเลอร์ตระกูล MCS-51

ตัวอย่างโปรแกรม

```
sbit e = P3^6;
sbit rs = P3^7;

/*-----
d300us: 300 microsec delay time
-----*/

void dms(unsigned int t)
{
    unsigned int i;
    for ( i = 0; i < t; i++ )
    {
        d300us();          // 324 uS delay time
    }
    return;
}

void lcd_command(unsigned char com)
{
    rs = 0;
    e = 1;
    P0 = com;
    dms(1);
    e = 0;
    dms(1);
    return;
}

void lcd_init(void)
{
    dms(10);
    lcd_command(0x38);
    lcd_command(0x38);
    lcd_command(0x38);
    lcd_command(0x0c);
    lcd_command(0x01);
    return;
}

void lcd_char(unsigned char text)
{
    unsigned char i;

    rs = 1;
    e = 1;
    P0 = text;
    dms(10);
    e = 0;
    dms(10);
    return;
}

void lcd_print(unsigned char text[16])
{
    unsigned char i;
    for (i = 0; i < 16; i++)
    {
        lcd_char(text[i]);
    }
    return;
}

/*-----
Main Program
```

```

-----*/
void main( void )
{
    char const hello[ ] = {"hello Thammasat "};

    unsigned char i;

    int_cpu();
    lcd_init();
    lcd_command(0x80);

    for (i = 0; i < 16; i++)
    {
        lcd_char(hello[i]); // แสดงการพิมพ์โดยใช้ lcd_char
    }
    lcd_command(0x02);
    lcd_command(0xc0);

    lcd_print("MicrocontrolleR"); // แสดงการพิมพ์โดยใช้ lcd_print

    while(1)
}

```

