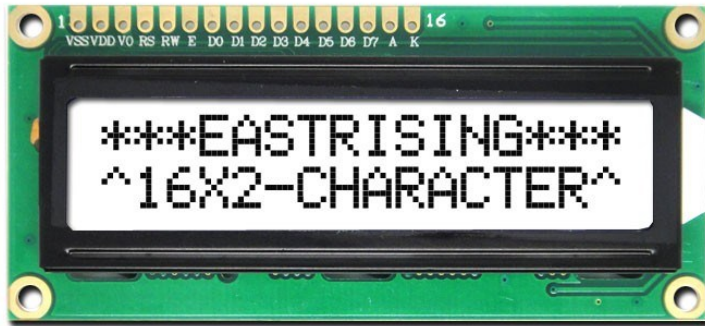


# LCD Module Interface (HD44780)

# LCD Modules

---

Character LCD Module

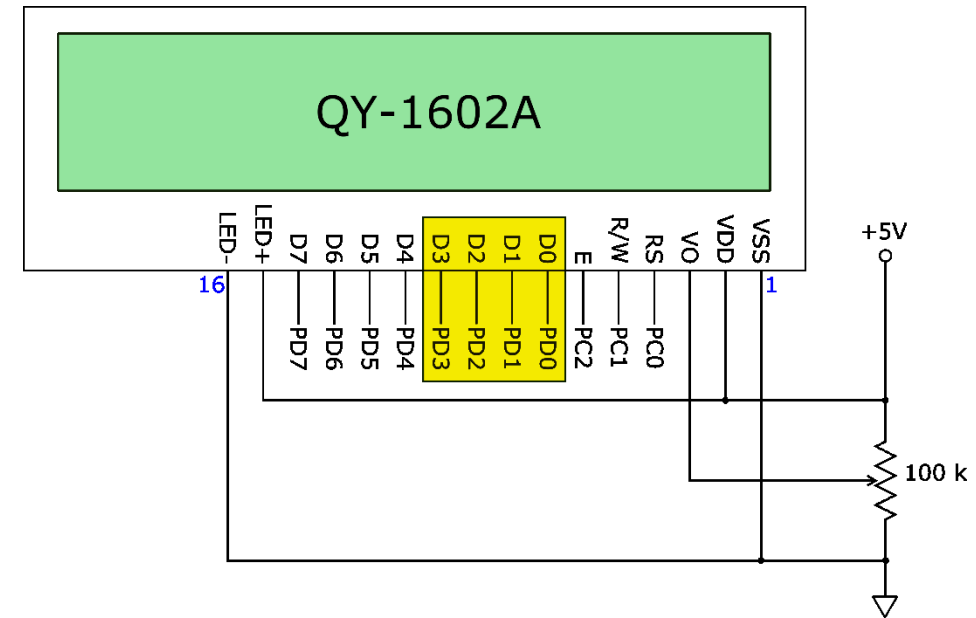
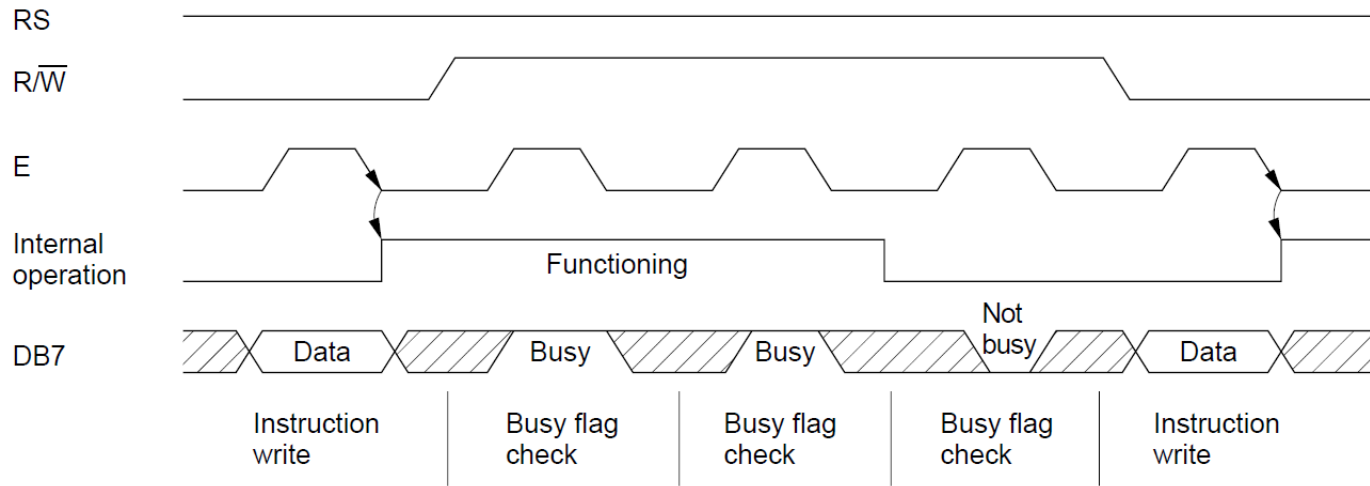


Graphic LCD Module



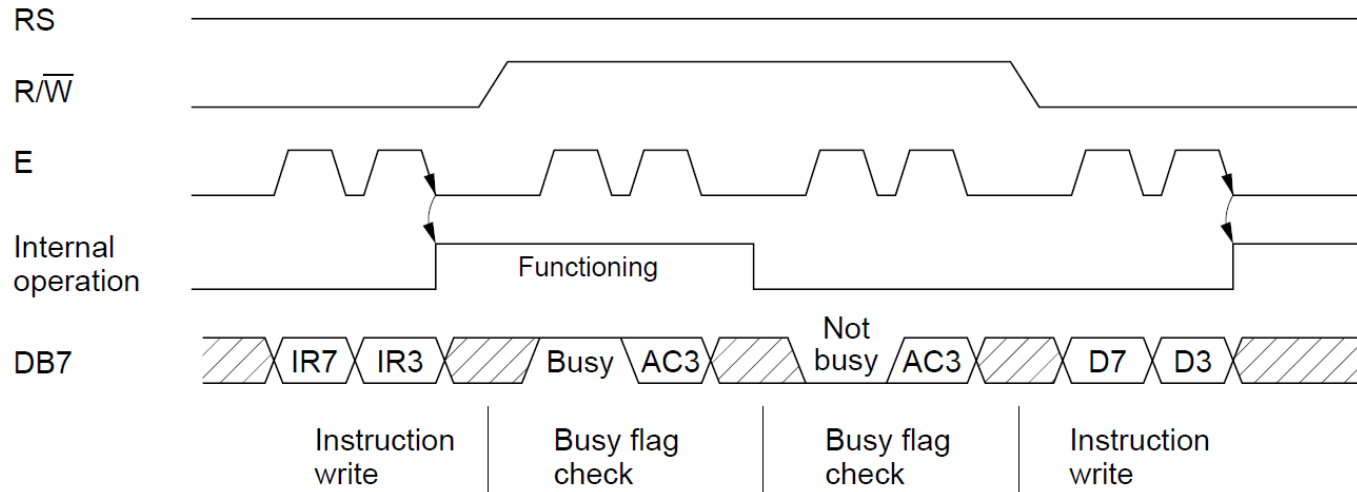
# HD44780-based LCD Module Interface

## 8-bit Interface

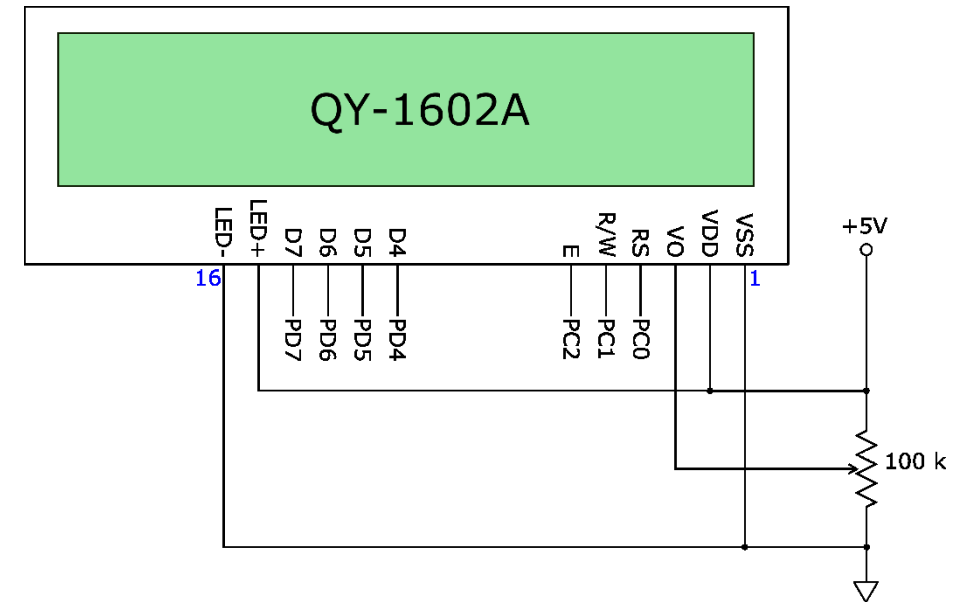


# HD44780-based LCD Module Interface

## 4-bit Interface



Note: IR7 , IR3 are the 7th and 3rd bits of the instruction.  
AC3 is the 3rd bit of the address counter.



# HD44780-based LCD Module Signals

---

- D[7:0] (입출력)

HD44780을 제어하기 위한 명령어 혹은 액정 화면에 표시할 데이터를 전송하기 위한 버스로서 8 비트를 모두 사용하거나 혹은 D[7:4]의 4 비트만을 사용할 수도 있다.

- RS 신호 (입력)

외부에서 HD44780으로 공급해 주는 신호로서,

이 신호가 논리값 '0'일 때에는 현재 데이터 버스에 전송되는 내용이 명령어임을 나타내고,

이 신호가 논리값 '1'일 때에는 데이터 버스에 전송되는 내용이 화면에 표시될 데이터임을 알려주는 신호이다.

# HD44780-based LCD Module Signals

- $R/\overline{W}$  신호 (입력)

- ✓ 외부에서 HD44780으로 공급해 주는 신호
- ✓ 이 신호의 논리값 '0'일 때에는 현재 데이터 버스에 전송되는 내용을 HD44780에 써 넣으라는 명령임을 알림.
- ✓ 이 신호가 논리값 '1'일 때에는 HD44780에서 상태 혹은 데이터를 읽어 내는 명령임을 나타내는 신호.

- E 신호 (입력)

외부에서 HD44780으로 공급해 주는 신호로서, RS 신호와  $R/\overline{W}$  신호의 동기를 맞춰주기 위한 클럭 신호이다.

# HD44780-based LCD Module Signals

---

- VO 신호 (입력)

이 신호의 전압을 변화시키면 액정 표시창의 흑백대비 비율이 달라진다.

- LED+ 및 LED-

이 액정 표시 장치의 배경 조명을 위한 LED 전원을 공급해 주는 단자임.

- VDD 및 VSS

HD44780 전원을 공급해 주는 단자임.

# HD44780-based LCD Module Commands

Instruction	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	설명
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.
Return home	0	0	0	0	0	0	0	0	1	-	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	-	-	Moves cursor and shifts display without changing DDRAM contents.
Function set	0	0	0	0	1	DL	N	F	-	-	Sets interface data length (DL), number of display lines (N), and character font (F).
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.
Write data to CG or DDRAM	1	0	Data to be written							Writes data into DDRAM or CGRAM.	
Read data from CG or DDRAM	1	1	Read data							Reads data from DDRAM or CGRAM.	

<p>I/D = 1: Increment                      I/D = 0: Decrement                      S = 1: Accompanies display shift                      S/C = 1: Display shift                      S/C = 0: Cursor move                      R/L = 1: Shift to the right                      R/L = 0: Shift to the left                      DL = 1: 8 bits, DL = 0: 4 bits                      N = 1: 2 lines, N = 0: 1 line                      F = 1: 5 × 10 dots, F = 0: 5 × 8 dots                      BF = 1: Internally operating                      BF = 0: Instructions acceptable</p>	<p>DDRAM: Display data RAM                      CGRAM: Character generator RAM                      ACG: CGRAM address                      ADD: DDRAM address (corresponds to cursor address)                      AC: Address counter used for both DD and CGRAM addresses</p>
---	--







# HD44780-based LCD Module Commands

## 3) Entry Mode Set

### (1) I/D

Increments ( $I/D = 1$ ) or decrements ( $I/D = 0$ ) the DDRAM address by 1 when a character code is written into or read from DDRAM. The cursor or blinking moves to the right when incremented by 1 and to the left when decremented by 1. The same applies to writing and reading of CGRAM.

### (2) S

Shifts the entire display either to the right ( $I/D = 0$ ) or to the left ( $I/D = 1$ ) when S is 1. The display does not shift if S is 0. If S is 1, it will seem as if the cursor does not move but the display does. The display does not shift when reading from DDRAM. Also, writing into or reading out from CGRAM does not shift the display.

RS	R/ $\bar{W}$	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1	I/D	S

# HD44780-based LCD Module Commands

## 4) Display On/Off Control

### (1) D

The display is on when D is 1 and off when D is 0. When off, the display data remains in DDRAM, but can be displayed instantly by setting D to 1.

### (2) C

The cursor is displayed when C is 1 and not displayed when C is 0. Even if the cursor disappears, the function of I/D or other specifications will not change during display data write. The cursor is displayed using 5 dots in the 8th line for 5x8 dot character font selection and in the 11th line for the 5x10 dot character font selection.

### (3) B

The character indicated by the cursor blinks when B is 1. The blinking is displayed as switching between all blank dots and displayed characters at a speed of 409.6 ms intervals when fcp or fOSC is 250 kHz. The cursor and blinking can be set to display simultaneously.

RS	R/ $\bar{W}$	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	D	C	B

# HD44780-based LCD Module Commands

## 5) Cursor or Display Shift

Cursor or display shift shifts the cursor position or display to the right or left without writing or reading display data. This function is used to correct or search the display. In a 2-line display, the cursor moves to the second line when it passes the 40th digit of the first line. Note that the first and second line displays will shift at the same time. When the displayed data is shifted repeatedly each line moves only horizontally. The second line display does not shift into the first line position. The address counter (AC) contents will not change if the only action performed is a display shift.

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	S/C	R/L	X	X

S/C	R/L	
0	0	Shifts the cursor position to the left. (AC is decremented by one.)
0	1	Shifts the cursor position to the right. (AC is incremented by one.)
1	0	Shifts the entire display to the left. The cursor follows the display shift.
1	1	Shifts the entire display to the right. The cursor follows the display shift.

# HD44780-based LCD Module Commands

## 6) Function Set

### (1) DL

Sets the interface data width. Data is sent or received in 8-bit width (DB7 to DB0) when DL is 1, and in 4-bit width (DB7 to DB4) when DL is 0. When 4-bit width is selected, data must be sent or received twice.

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	DL	N	F	X	X

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5x8 dots	1/8	
0	1	1	5x10 dots	1/11	
1	X	2	5x8 dots	1/16	*주)

# HD44780-based LCD Module Commands

## 6) Function Set

### (2) N

Sets the number of display lines.

### (3) F

Sets the character font.

Note: Perform the function at the head of the program before executing any instructions (except for the read busy flag and address instruction). From this point, the function set instruction cannot be executed unless the interface data length is changed.

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	DL	N	F	X	X

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5x8 dots	1/8	
0	1	1	5x10 dots	1/11	
1	X	2	5x8 dots	1/16	

# HD44780-based LCD Module Commands

## 7) Set CGRAM Address

Set CGRAM address sets the CGRAM address binary AAAAAA into the address counter. Data is then written to or read from the MPU for CGRAM.

RS	$\overline{R/W}$	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	A	A	A	A	A	A



# HD44780-based LCD Module Commands

## 8) Set DDRAM Address

- ✓ Set DDRAM address sets the DDRAM address binary AAAAAAA into the address counter.
- ✓ Data is then written to or read from the MPU for DDRAM.
- ✓ However, when N is 0 (1-line display), AAAAAAA can be 00H to 4FH.
- ✓ When N is 1 (2-line display), AAAAAAA can be 00H to 27H for the first line, and 40H to 67H for the second line.

RS	R/ $\overline{W}$	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	A	A	A	A	A	A	A

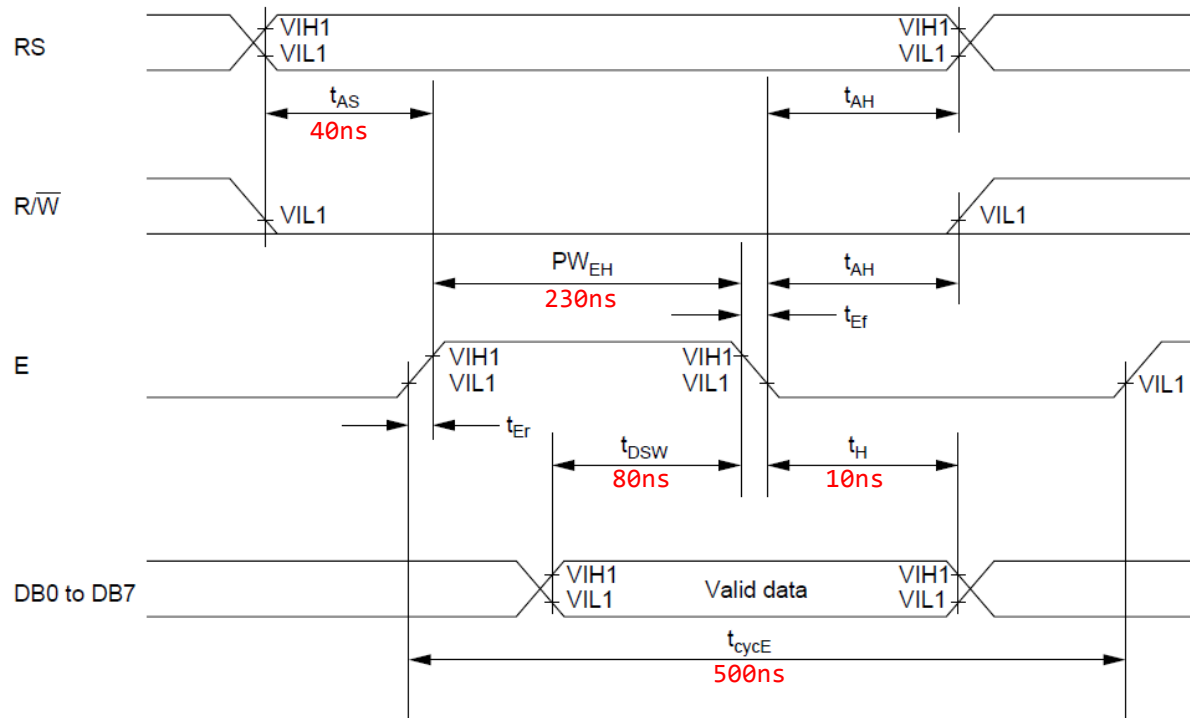
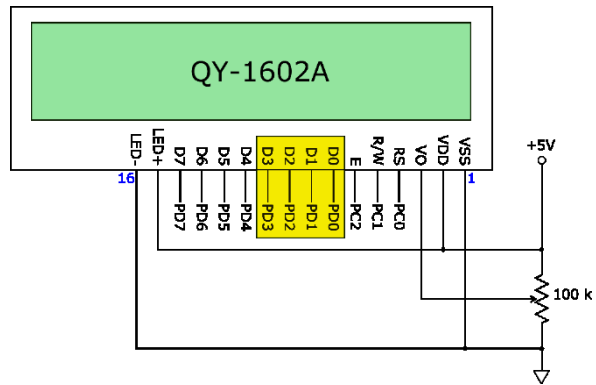
# HD44780-based LCD Module Commands

## 9) Read Busy Flag and Address

- ✓ Read busy flag and address reads the busy flag (BF) indicating that the system is now internally operating on a previously received instruction. **If BF is 1, the internal operation is in progress.**
- ✓ The next instruction will not be accepted until BF is reset to 0.
- ✓ Check the BF status before the next write operation.
- ✓ At the same time, the value of the address counter in binary AAAAAAA is read out.
- ✓ This address counter is used by both CG and DDRAM addresses, and its value is determined by the previous instruction.
- ✓ The address contents are the same as for instructions set CGRAM address and set DDRAM address.

RS	R/ $\overline{W}$	D7	D6	D5	D4	D3	D2	D1	D0
0	1	BF	A	A	A	A	A	A	A

# HD44780-based LCD Module Interface (8-bit)

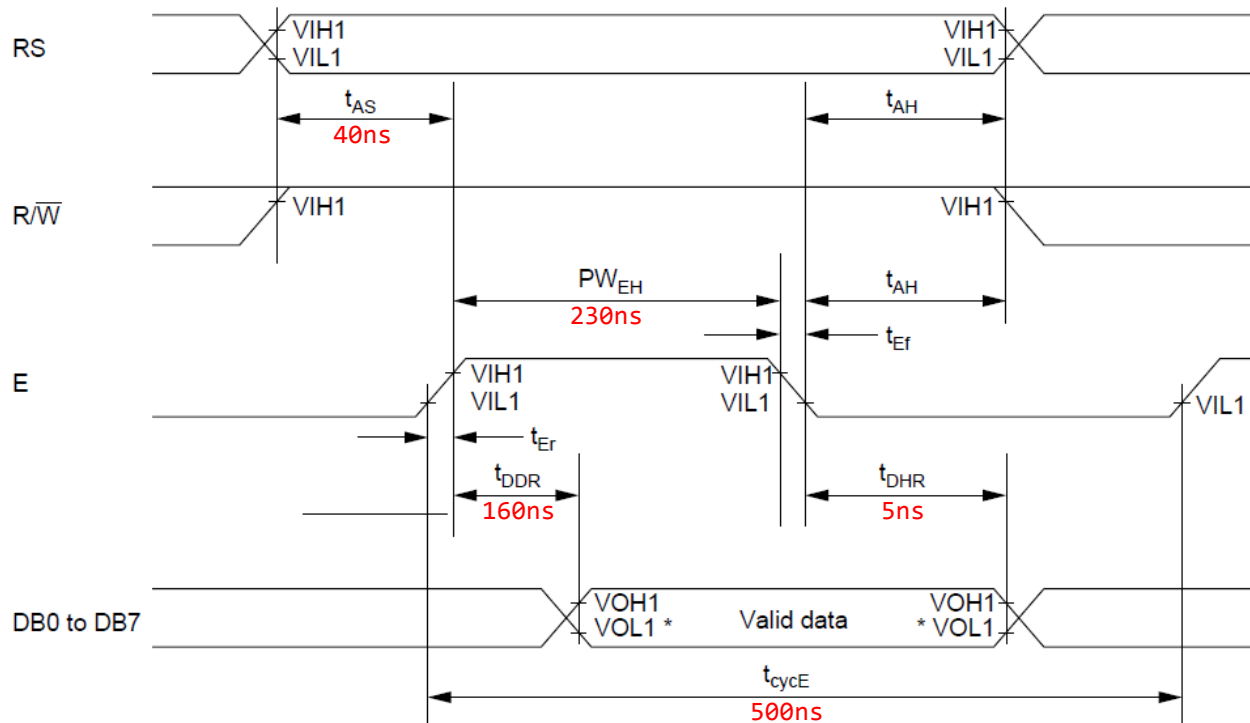
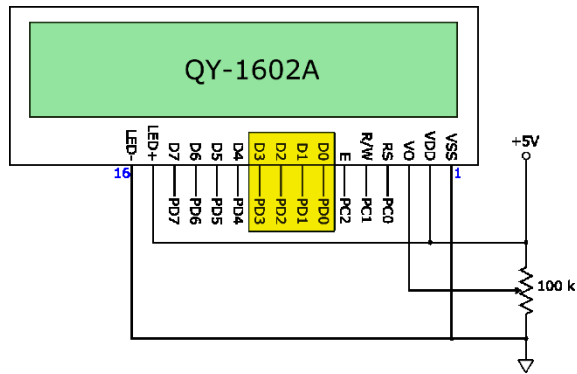


```
void init_port(void)
{
    DDRC |= 0b111;    // Set PC[2:0] to OUTPUT mode
    DDRD = 0xFF;     // Set PD[7:0] to OUTPUT mode
}
```

```
void write_command(uint8_t cmd)
{
    PORTC &= ~(1 << 0);    // Set RS to LOW
    PORTC &= ~(1 << 1);    // Set RW to LOW
    DDRD = 0xFF;           // Set PORTD to OUTPUT mode
    PORTC |= (1 << 2);     // Set E to HIGH
    PORTD = cmd;
    _delay_us(1);          // t_DSW = 80 ns max
                           // PW_EH=230ns
    PORTC &= ~(1 << 2);    // Set E to LOW
}
```

```
void write_data(uint8_t data)
{
    PORTC |= (1 << 0);     // Set RS to HIGH
    PORTC &= ~(1 << 1);    // Set RW to LOW
    DDRD = 0xFF;           // Set PORTD to OUTPUT mode
    PORTC |= (1 << 2);     // Set E to HIGH
    PORTD = data;
    _delay_us(1);          // t_DSW = 80 ns max
    PORTC &= ~(1 << 2);    // Set E to LOW
}
```

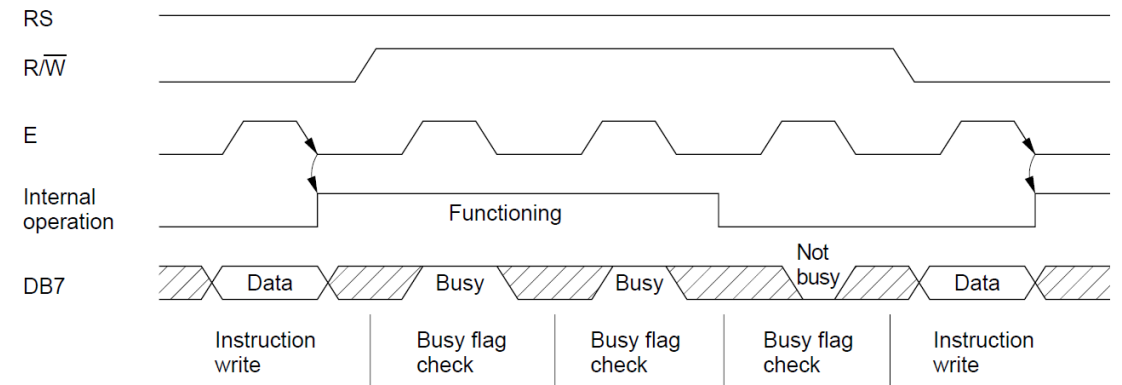
# HD44780-based LCD Module Interface (8-bit)



```

uint8_t check_busy(void)
{
    uint8_t status;

    PORTC &= ~(1 << 0);           // Set RS to LOW
    PORTC |= (1 << 1);           // Set RW to HIGH
    DDRD = 0x00;                 // Set PORTD to INPUT mode
    PORTC |= (1 << 2);           // Set E to HIGH
    _delay_us(1);                // t_DDR = 160 ns max
                                // PW_EH=230ns
    status = PIND;               // Read input status
    PORTC &= ~(1 << 2);         // Set E to LOW
    return status & 0x80;
}
    
```



# HD44780-based LCD Module Interface (8-bit)

```
int main(void)
{
    unsigned int count = 0;
    char lcd_buf[16];

    init_port();
    init_LCD();

    put_string("LCD1602(HD44780)");

    set_cursor(0, 1);
    put_string("COUNTER=");

    while (1)
    {
        set_cursor(8, 1);
        sprintf(lcd_buf, "%u", count);
        put_string(lcd_buf);
        count++;
        _delay_ms(1000);
    }
}
```

```
void set_cursor(uint8_t x, uint8_t y)
{
    uint8_t addr;

    addr = 0x80 + (y << 6) + x;
    write_command_with_busy_check(addr);
}
```

```
void write_command_with_busy_check(uint8_t cmd)
{
    while(check_busy() != 0);

    write_command(cmd);
}
```

```
void put_string(char *str)
{
    while(*str != '\0')
        write_data(*str++);
}
```

```
void init_LCD(void)
{
    write_command_with_busy_check(0x38); // function set: 8-bit interface, 2 display lines, 5x7 font
    write_command_with_busy_check(0x06); // entry mode set: increment automatically, no display shift
    write_command_with_busy_check(0x0C); // display control: turn display on, cursor off, no blinking
    write_command_with_busy_check(0x01); // clear display, set cursor position to zero
}
```

End