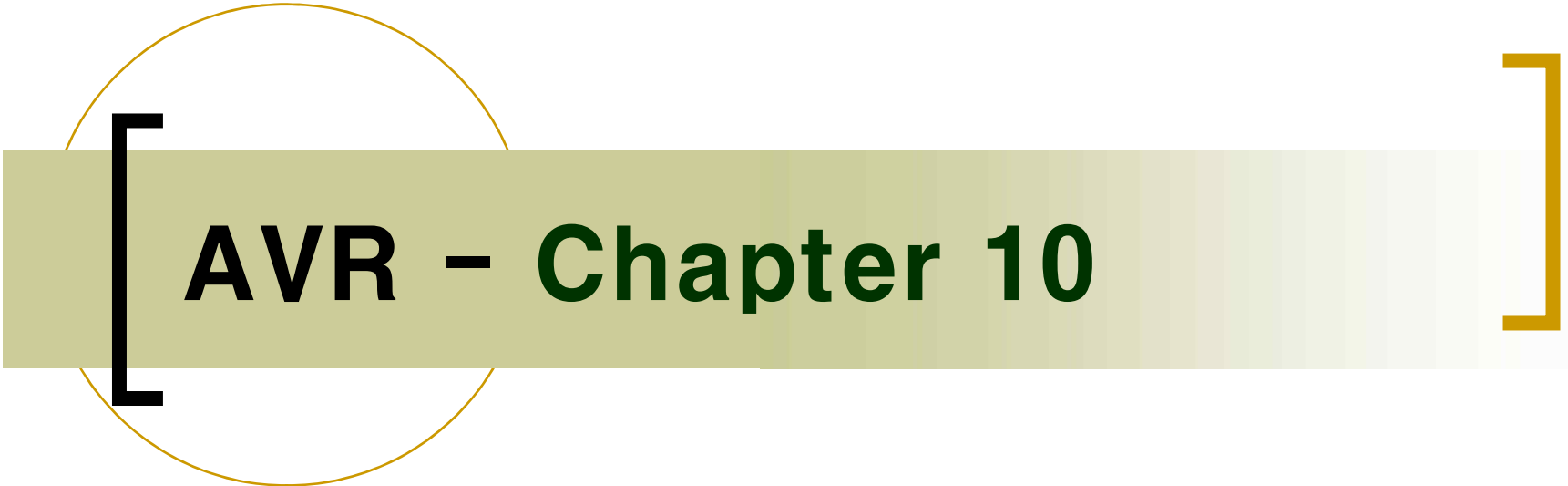




AVR - Chapter 10

류대우

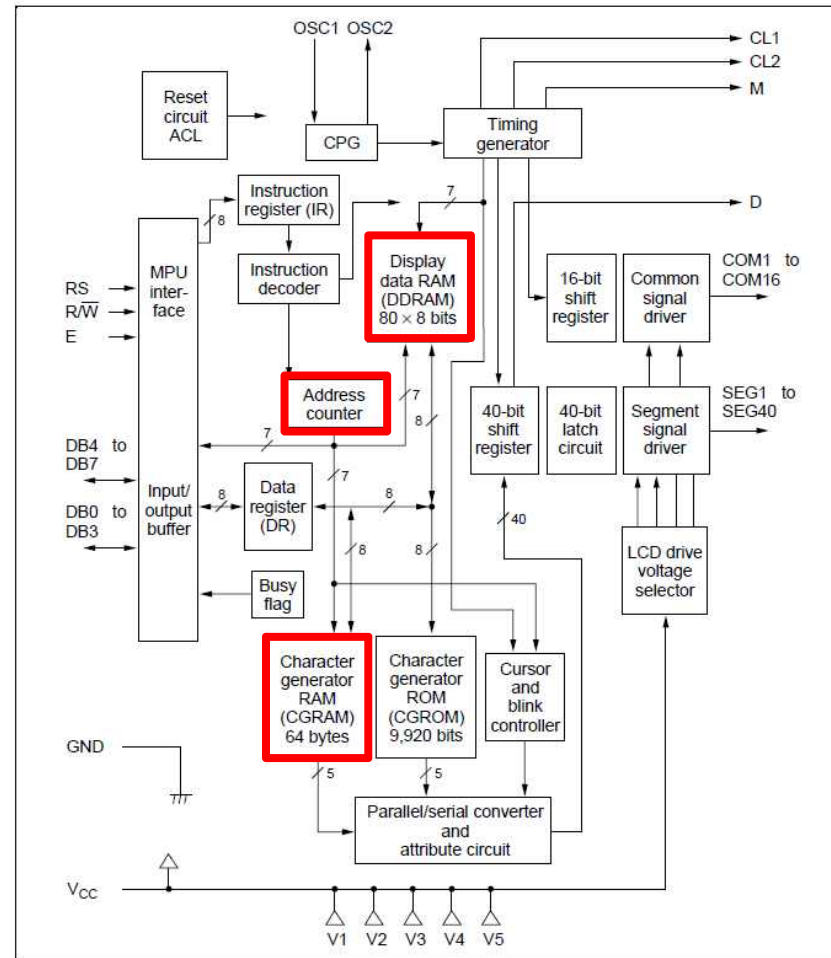
(주)뉴티씨(NEWTC) 대표이사

davidryu@newtc.co.kr

[텍스트 LCD 제어 응용편]

- printf를 활용하여 LCD에 쉽고 빠르게 글을 쓰는 방법을 배운다.
- 시리얼 LCD 제작하기
 - 컴퓨터에서 입력한 것을 시리얼로 보내서 텍스트 LCD에 출력하는 것을 만들어본다.
- CGRAM을 활용하여 한글 출력을 해본다.
 - 시리얼 한글 출력 LCD 제작

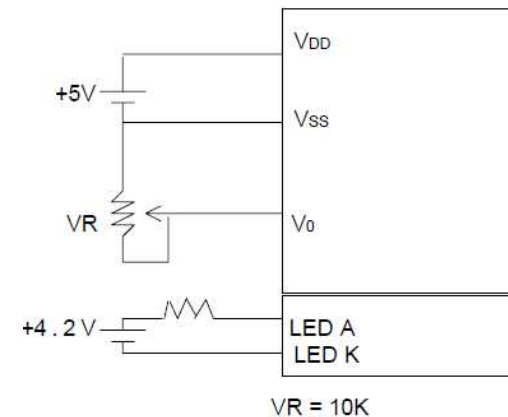
[히다찌 HD44780U 블록도]



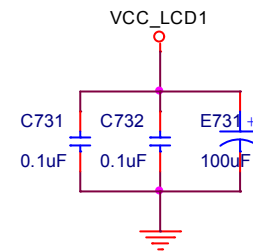
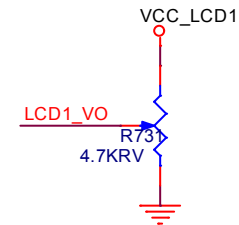
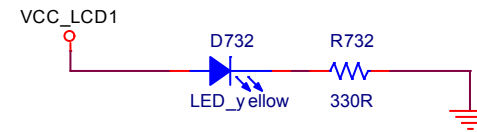
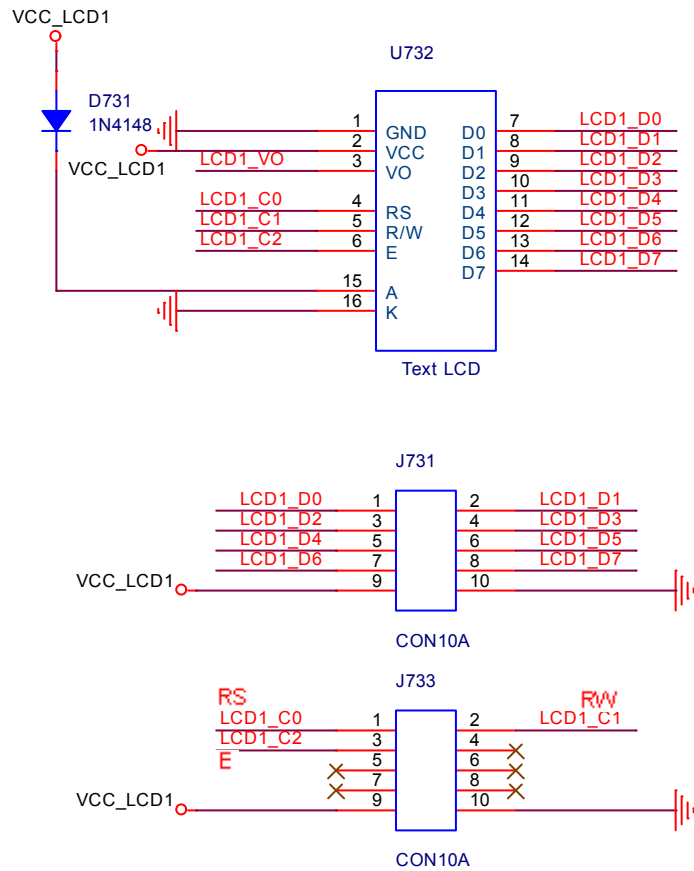
[LCD 핀배치와 전원부]

■ 사용한 텍스트 LCD 자체의 핀배치

No.	Symbol	Function
1	VSS	Ground, 0V
2	VDD	Logic power supply, +5V
3	V0	Voltage for LCD drive
4	RS	Data / Instruction register select
5	R/W	Read / Write
6	E	Enable signal, start data read/write
7	DB0	Data Bus Line
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED A	LED Anode, power supply +
16	LED K	LED Cathode, ground 0V

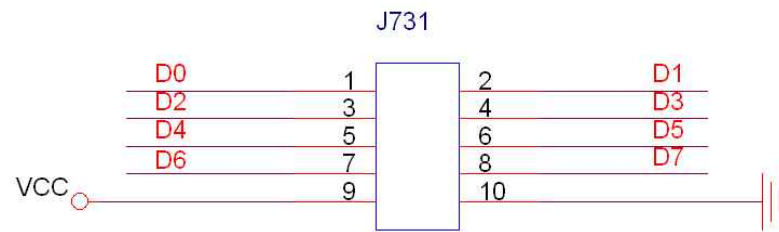


[텍스트 LCD 회로도]



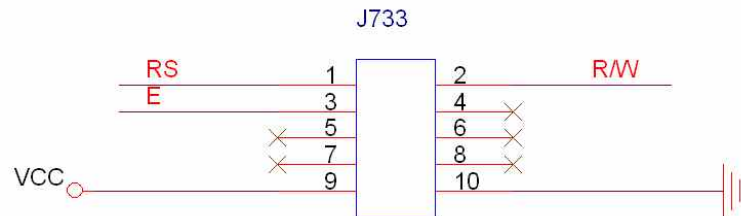
[커넥터 연결]

■ 데이터 신호 핀 번호



1 번	3 번	5 번	7 번	9 번
D0	D2	D4	D6	전원 5V
2 번	4 번	6 번	8 번	10 번
D1	D3	D5	D7	전원 GND

■ 컨트롤 신호 핀 번호



1 번	3 번	5 번	7 번	9 번
RS	E			전원 5V
2 번	4 번	6 번	8 번	10 번
R/W				전원 GND

[LCD 제어용 비트]

- **Control : RS, RW, E 3개 핀을 제어해야 한다.**
- **Data : D7~D0 까지 8개 핀을 제어해야 한다.**
- **RS : 0번 비트**
 - `#define rs(m) m?(CONTROL|=0x01):(CONTROL&=0xFE)`
- **RW : 1번 비트**
 - `#define rw(m) m?(CONTROL|=0x02):(CONTROL&=0xFD)`
- **E : 2번 비트**
 - `#define en(m) m?(CONTROL|=0x04):(CONTROL&=0xFB)`

[Instruction



Instruction	Instruction Code										DESCRIPTION	Executed Time(fosc =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
Cursor At Home	0	0	0	0	0	0	0	0	1	-	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 enable the shift of entire display.	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	-	-	Set cursor moving and display shifts cursor bit, and the direction, without changing of DDRAM data.	39μS
Function Set	0	0	0	0	1	DL	N	F	-	-	Sets interface data length (DL:8-BIT/4-BIT), number of display lines(N:2-line/1-line) and, display font type (F:5x11dots/5x8 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	Reads data from internal RAM (DDRAM / CGRAM).	43μS

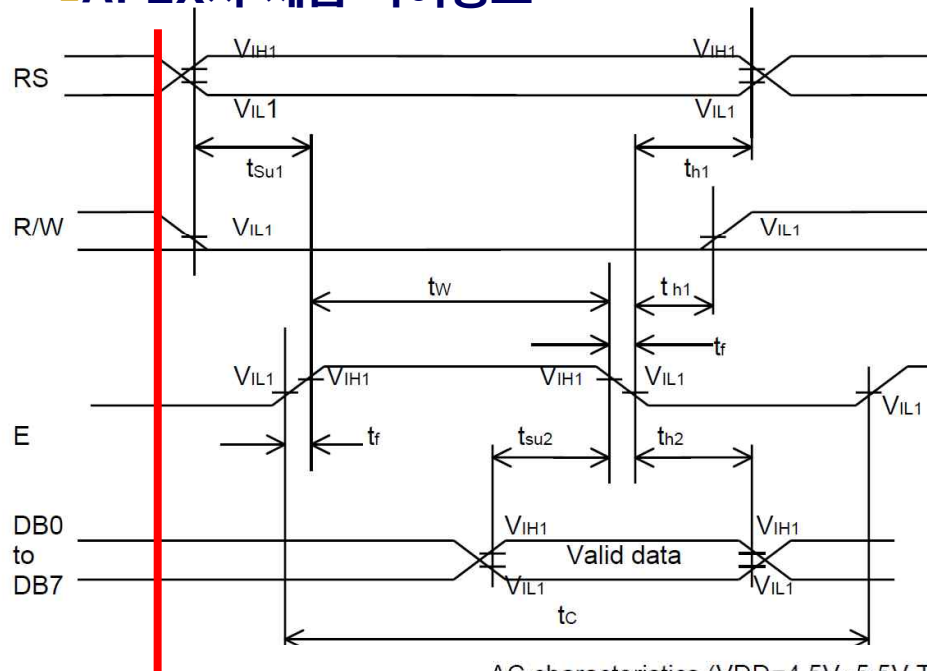
I/D = 1 : Increment 0 : Decrement
 S = 1 : Accompanies Display Shift
 S/C = 1 : Display Shift 0 : Cursor Move
 R/L = 1 : Shift right 0 : Shift left
 DL = 1 : 8 Bits 0 : 4 Bits
 N = 1 : 2 Lines 0 : 1 Line
 F = 1 : 5 x 10 Dots 0 : 5 x 7 Dots
 BF = 1 : Internally Operating
 0 : Can Accept Instruction

DD RAM : Display Data RAM
 CG RAM : Character Generator RAM
 ACG : CG RAM Address
 ADD : DD RAM Address
 Corresponds to Cursor Address
 AC : Address Counter used for
 Both DD and CG RAM Address
 * No effect (Don't care)

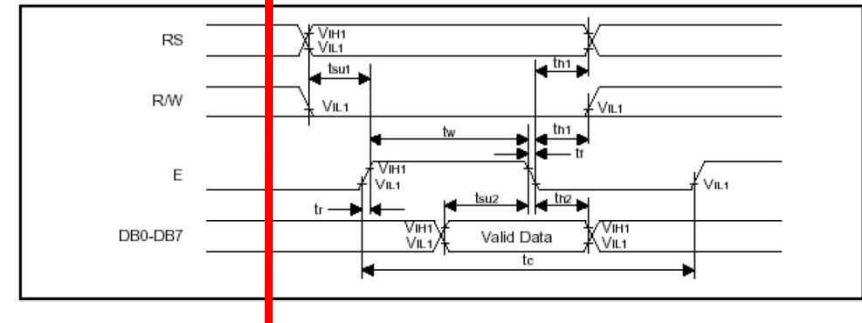
[WRITE 타이밍도

■ 사용 LCD : APEX-AX07001-01

■ APEX사 제품 타이밍도



■ 히다찌사 제품 타이밍도



AC characteristics (VDD=4.5V~5.5V, Ta=-30~85°C)

Item	Symbol	Condition	Min.	Max.	Unit
E cycle time	t_C	VDD = 5.0V	500	--	ns
E pulse width (high level)	t_W		230	--	
E rise/fall time	t_R, t_F		--	20	
R/W and RS Setup time	t_{Su1}		40	--	
R/W and RS Hold time	t_{H1}		10	--	
Data setup time	t_{Su2}		80	--	
Data hold time	t_{H2}		10	--	

[WriteCommand 함수]

- `void WriteCommand(unsigned char command){`
- `rs(0);`
- `// en(0);delay_lcd();`
- `// LCD에 따라서 이렇게 OFF 시켜 줘야 할 때도 있음.`
- `rw(0);delay_lcd();`
- `//en(1);delay_lcd();`
- `// LCD에 따라서 이렇게 OFF 시켜 줘야 할 때도 있음.`
- `DATA = command;delay_lcd();`
- `en(0);delay_lcd();`
- `en(1);delay_lcd();`
- `}`

[WriteData 함수

]

- `void WriteData(unsigned char data){`
- `rs(1);`
- `//en(0);delay_lcd();`
- `// LCD에 따라서 이렇게 OFF 시켜 줘야 할 때도 있음.`
- `rw(0);delay_lcd();`
- `//en(1);delay_lcd();`
- `// LCD에 따라서 이렇게 OFF 시켜 줘야 할 때도 있음.`
- `DATA = data;delay_lcd();`
- `en(0);delay_lcd();`
- `en(1);delay_lcd();`
- `}`

[WriteStr(X, Y, *str)]

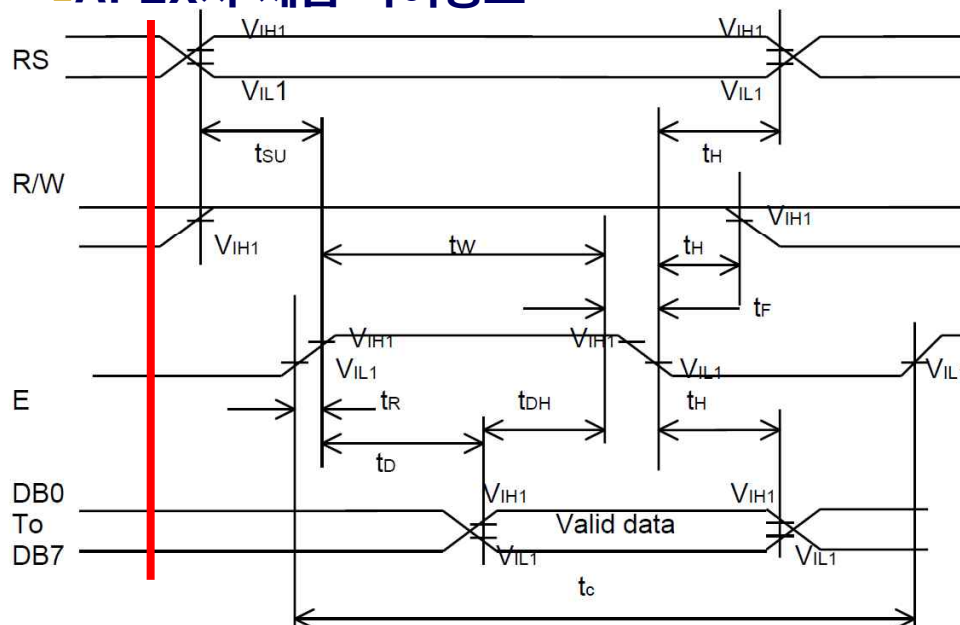
- // printf 함수 사용시 추가할 것.
- int putchar(char c)
- {
- WriteData(c);
- return c;
- }

- void WriteStr(int X, int Y, char *str){
- Move(X,Y);
- printf(str);
- }

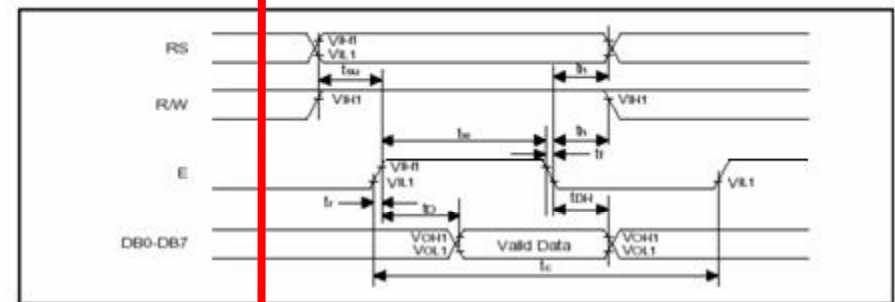
[READ 타이밍도]

■ 사용 LCD : APEX-AX07001-01

■ APEX사 제품 타이밍도



■ 히다찌사 제품 타이밍도



AC characteristics (VDD=4.5V~5.5V, Ta=-30~85°C)

Item	Symbol	Condition	Min.	Max.	Unit
E cycle time	t_c	VDD = 5.0V	500	--	ns
E pulse width (high level)	t_w		230	--	
E rise/fall time	t_R, t_F		--	20	
R/W and RS shetup time	t_{su}		40	--	
R/W and RS hold time	t_H		10	--	
Data output delay time	t_D		--	120	
Data hold time	t_{DH}		5	--	

[ReadData 함수

]

```
■ unsigned char ReadData(void){  
■     unsigned char data;  
■     rs(1);  
■     en(0);delay_lcd();  
■     rw(1);delay_lcd();  
■     en(1);delay_lcd();  
■     data = DATA;delay_lcd();  
■     en(0);delay_lcd();  
■     en(1);delay_lcd();  
■ }
```

[Command 함수들(1/3)]

- `void ClearDisplay(void){`
- `WriteCommand(0x01);`
- `}`
- `void ReturnHome(void){`
- `WriteCommand(0x02);`
- `}`
- `void EntryModeSet(unsigned char I_D, unsigned char SH){`
- `WriteCommand(0x04|(I_D<<1)|SH);`
- `}`
- `void DisplayOnOff(unsigned char D, unsigned char C, unsigned char B){`
- `WriteCommand(0x08|(D<<2)|(C<<1)|B);`
- `}`

[Command 함수들(2/3)]

- `void Shift(unsigned char S_C, unsigned char R_L){`
- `WriteCommand(0x10|(S_C<<3)|(R_L<<2));`
- `}`
- `void FunctionSet(unsigned char DL, unsigned char N, unsigned char F){`
- `WriteCommand(0x20|(DL<<4)|(N<<3)|(F<<2));`
- `}`
- `void CGRAM_Set(unsigned char addr){`
- `WriteCommand(0x40|(addr&0x3f));`
- `}`
- `void DDRAM_Set(unsigned char addr){`
- `WriteCommand(0x80|(addr&0x7f));`
- `}`

[Command 함수들(3/3)]

- `void Move(unsigned char X, unsigned char Y){`
- `unsigned char addr;`
- `if((X>40)||(X<0)){ // X : Column`
- `X=0;`
- `}`
- `if((Y>=2)||(Y<0)){ // Y : Line`
- `Y=0;`
- `}`
- `addr = Y*40+X;`
- `DDRAM_Set(addr);`
- `}`

[초기화 코드]

```
■ void InitTLCD(void){  
■     delay(30);           // 전원 5V 도달 이후 최소 15ms 대기  
■     FunctionSet(1,1,1); // 8비트길이, 2라인 LCD, 5x11 dots 폰트  
■     delay(10);           // 최소 4.1ms 대기  
■     FunctionSet(1,1,1);  
■     delay(1);            // 최소 100us 대기  
■     FunctionSet(1,1,1);  
■     FunctionSet(1,1,1);  
■     DisplayOnOff(0,0,0); // 디스플레이, Cursor, Blink 모두 끄고  
■     DisplayOnOff(1,0,0); // 디스플레이만 켜고  
■     EntryModeSet(0,0);  // Shift 안함.  
■     ClearDisplay();  
■ }
```

[Textlcd.h 의 내용 정리(선언)]

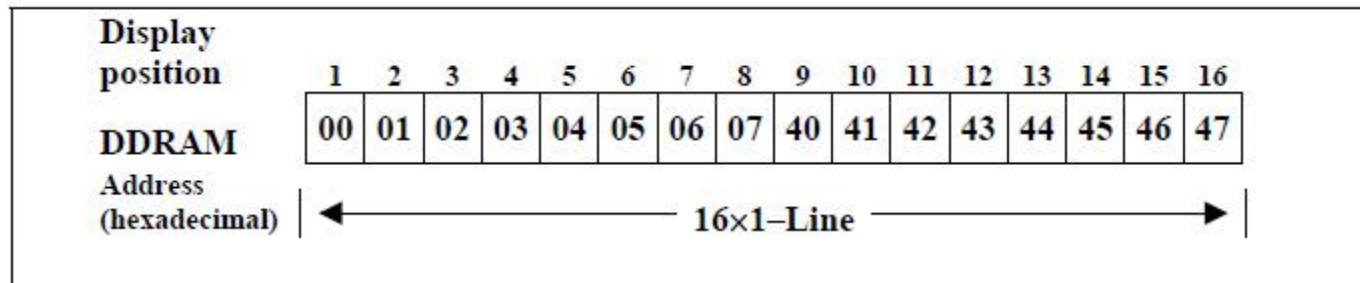
- `#include <stdio.h>`
- `// PORT definition`
- `#define CONTROL PORTA`
- `#define DATA PORTF`

[딜레이 함수]

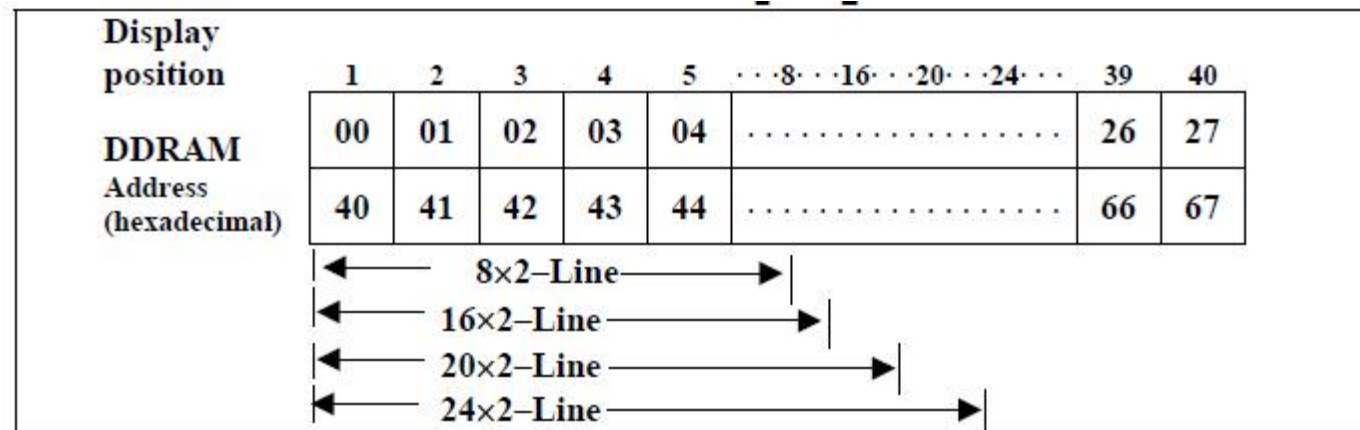
```
■ void delay_lcd(void){  
■     int i=2000;  
■     while(i--){  
■     }  
■ }  
■ void delay(int delaytime){  
■     int i=1000,j;  
■     while(i--){  
■         for(j=0;j<delaytime;j++){  
■         }  
■     }  
■ }
```

[DDRAM의 주소]

■ 1-Line Display



■ 2-Line Display



[한글 사용하기]

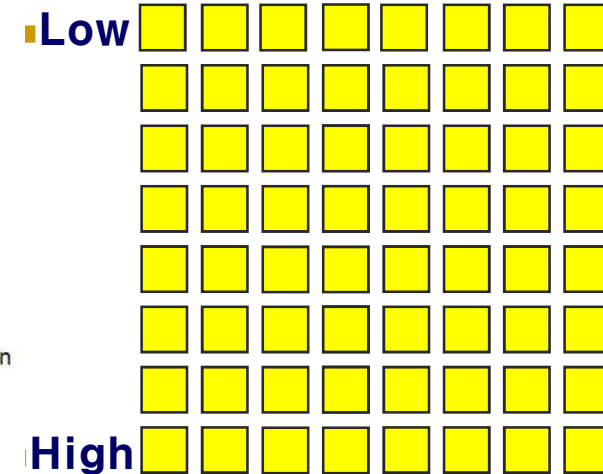
- AVR에서 한글은 조합형으로 만들어진다.
- 이 때, CGRAM을 이용하면 매우 편리하다.
- 최대 사용 가능 개수(64 바이트)
 - 5 x 8 dots : 8 character
 - 5 x 10 dots : 4 character
- 16비트를 각각 5개 비트씩 쪼개서, 초성+중성+종성으로 분류되므로, 다 더하면 한글자가 된다. MSB는 한/영 선택이다.

[CGRAM (5 x 8 dots)]

■ CGRAM addresses, DDRAM, CGRAM Data

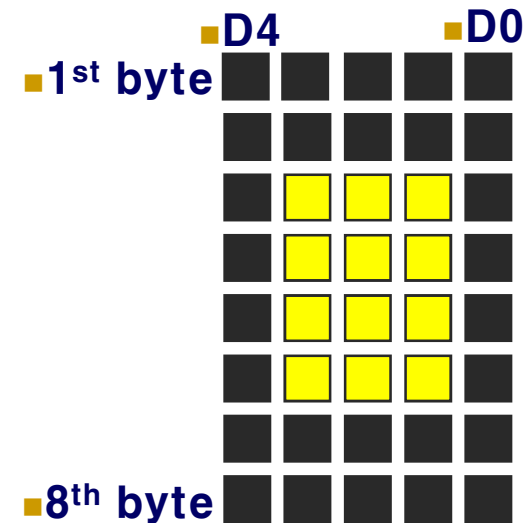
For 5 × 8 dot character patterns

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)
7 6 5 4 3 2 1 0 High Low	5 4 3 2 1 0 High Low	7 6 5 4 3 2 1 0 High Low
0 0 0 0 * 0 0 0	0 0 0	* * * 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 Character pattern (1) Cursor position
0 0 0 0 * 0 0 1	0 0 1	* * * 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 Character pattern (2) Cursor position
0 0 0 0 * 1 1 1	1 1 1	* * * 1 0 0 1 0 1 1 1 0 1 1 1

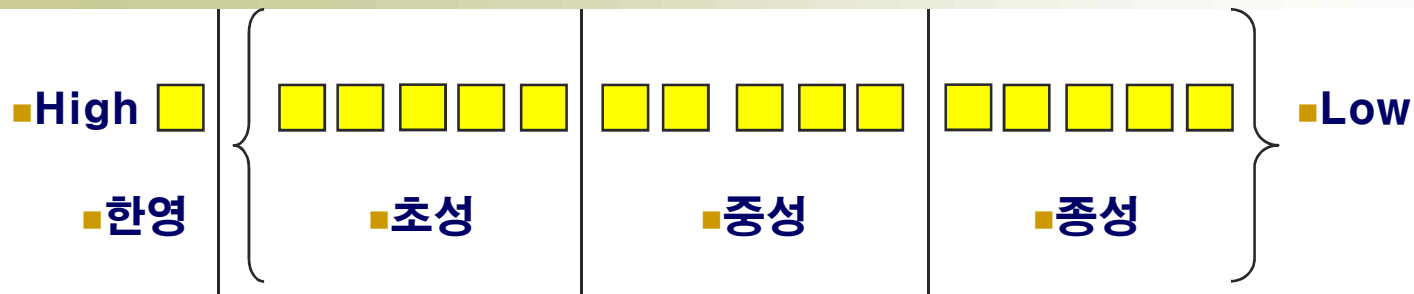


[CGRAM 이용하기 기본]

- **CGRAM_Set(0x00);**
WriteData(0xFF);
WriteData(0xFF);
WriteData(0x11);
WriteData(0x11);
WriteData(0x11);
WriteData(0x11);
WriteData(0xFF);
WriteData(0xFF);
DDRAM_Set(0x80);
WriteData(0x00);



[8x8 한글 조합형 폰트(종성)]



■ 1:한글
■ 0:영어

■ 코드 값으로 폰트에서 위치 찾는다. 찾을 때, 종성부터 비교해서 찾아가고서,
■ 출력하기 전에 비트 OR 한다.

■ Ex) '각' → 0x8862 바이너리 코드이다.

■ 1000 1000 0110 0010

1 00010 00011 00010

■ 한글 초성 중성 종성

[8x8 한글 조합형 폰트(종성)]



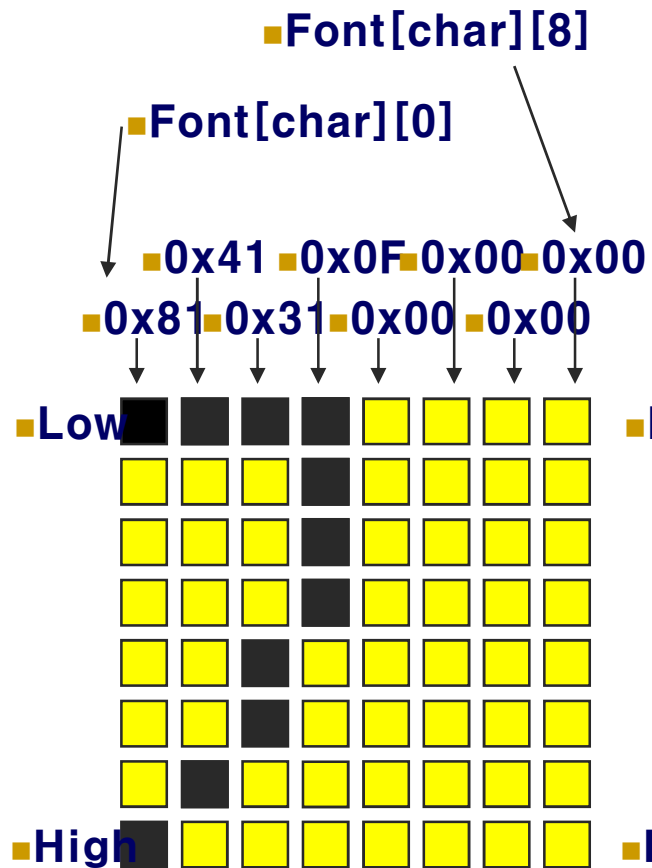
[한글 조합형/영문아스키 코드표]

비트 값		조성	중성	중성
10진	2진			
0	00000	<사용불가>	<사용불가>	<사용불가>
1	00001	<제음>	<사용불가>	<제음>
2	00010	ㄱ	<제음>	ㄱ
3	00011	ㄴ	ㅏ	ㅑ
4	00100	ㄷ	ㅓ	ㅕ
5	00101	ㄹ	ㅗ	ㅛ
6	00110	ㅁ	ㅜ	ㅠ
7	00111	ㅂ	ㅡ	ㅝ
8	01000	ㅅ	<사용불가>	ㅆ
9	01001	ㅇ	<사용불가>	ㅈ
10	01010	ㅊ	ㅊ	ㅊ
11	01011	ㅋ	ㅊ	ㅊ
12	01100	ㅌ	ㅊ	ㅊ
13	01101	ㅍ	ㅊ	ㅊ
14	01110	ㅊ	ㅊ	ㅊ
15	01111	ㅊ	ㅊ	ㅊ
16	10000	ㅊ	<사용불가>	ㅊ
17	10001	ㅊ	<사용불가>	ㅊ
18	10010	ㅊ	ㅊ	<28>
19	10011	ㅊ	ㅊ	ㅊ
20	10100	ㅊ	ㅊ	ㅊ
21	10101	<한자>	ㅊ	ㅊ
22	10110	<한자>	ㅊ	ㅊ
23	10111	<한자>	ㅊ	ㅊ
24	11000	<한자>	<사용불가>	ㅊ
25	11001	<한자>	<사용불가>	ㅊ
26	11010	<한자>	ㅊ	ㅊ
27	11011	<한자>	ㅊ	ㅊ
28	11100	<한자>	ㅊ	ㅊ
29	11101	<한자>	ㅊ	ㅊ
30	11110	<한자>	<.>	<.>
31	11111	<사용불가>	<.>	<사용불가>

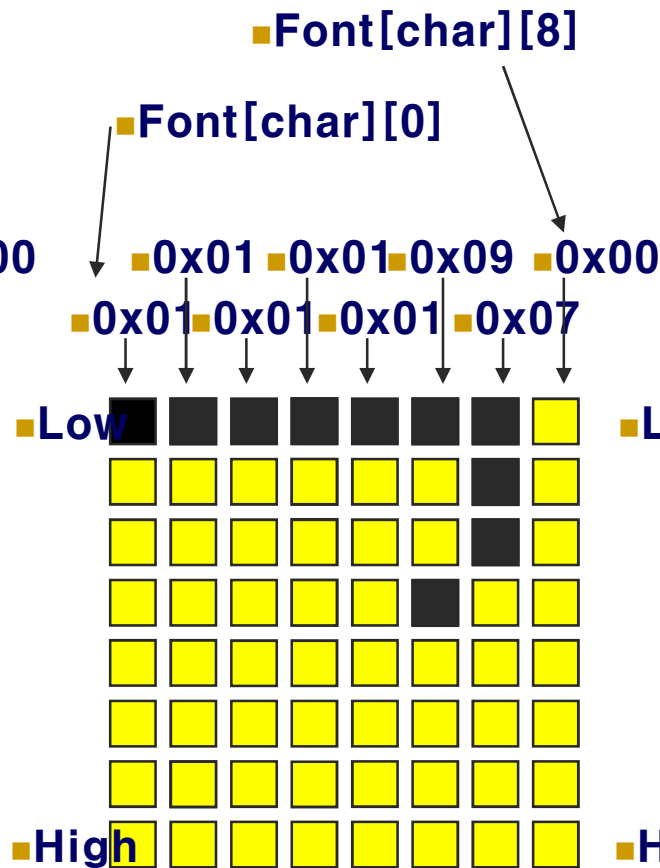
		상위 코드 값									
		16진	0	1	3	3	4	5	6	7	
16진	2진	0000	0001	0010	0011	0100	0101	0110	0111		
하	0	0000		공백	0	@	P		P		
위	1	0001		DC1	!	1	A	Q	a	q	
코	2	0010		DC2	~	2	B	R	b	r	
드	3	0011		DC3	#	3	C	S	c	s	
값	4	0100		DC4	\$	4	D	T	d	t	
	5	0101			%	5	E	U	e	u	
	6	0110			&	6	F	V	f	v	
	7	0111	BEL		'	7	G	W	g	w	
	8	1000	BS	CAN	(	8	H	X	h	x	
	9	1001	HT	EM	)	9	I	Y	i	y	
	A	1010	LF		*	:	J	Z	j	z	
	B	1011	VT	ESC	+	:	K	[	k	{	
	C	1100	FF	FS	.	<	L	\	l		
	D	1101	CR		-	=	M	]	m	}	
	E	1110	SO		.	>	N	^	n	~	
	F	1111	SI		/	?	O	_	o	△	

[8x8 한글 조합형 폰트(초성)]

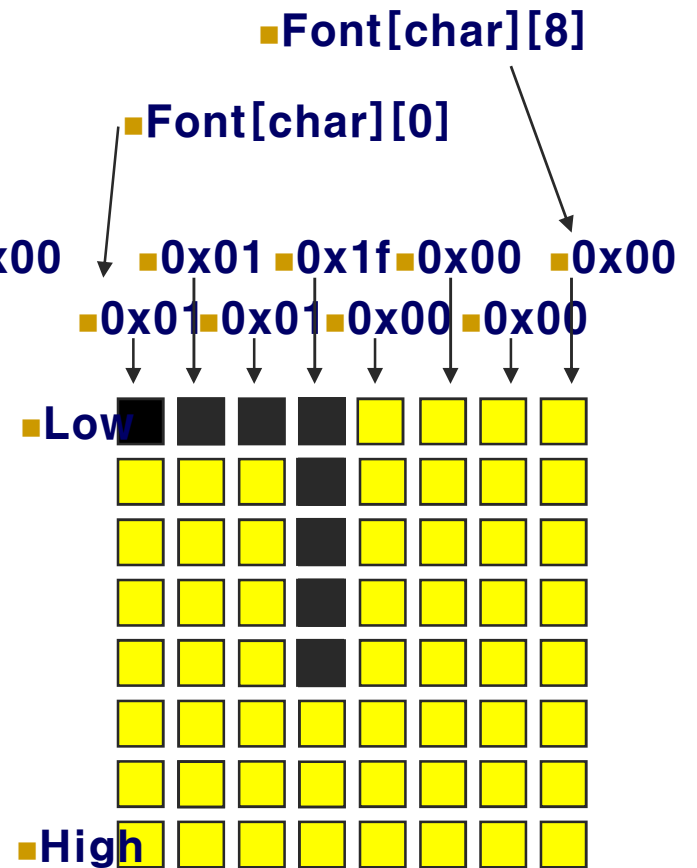
■ 초성 첫번째 폰트



■ 초성 두번째 폰트

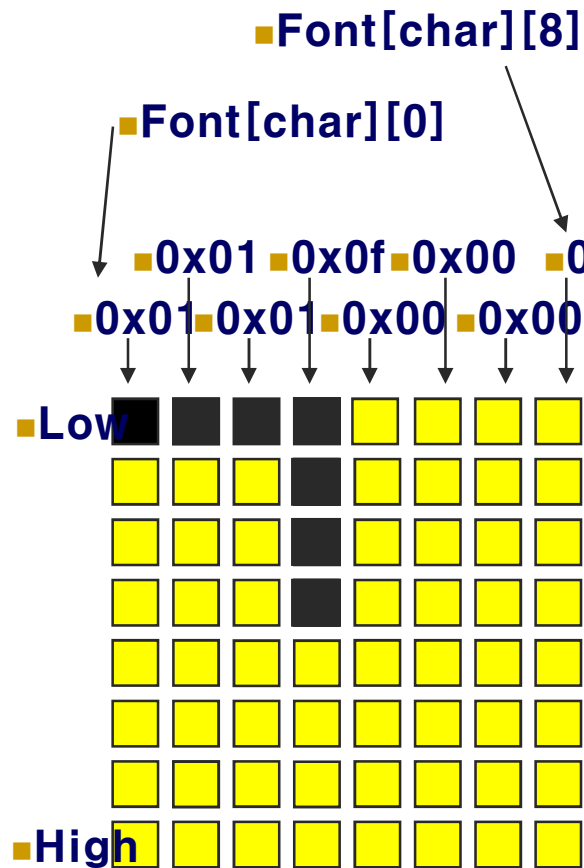


■ 초성 세번째 폰트

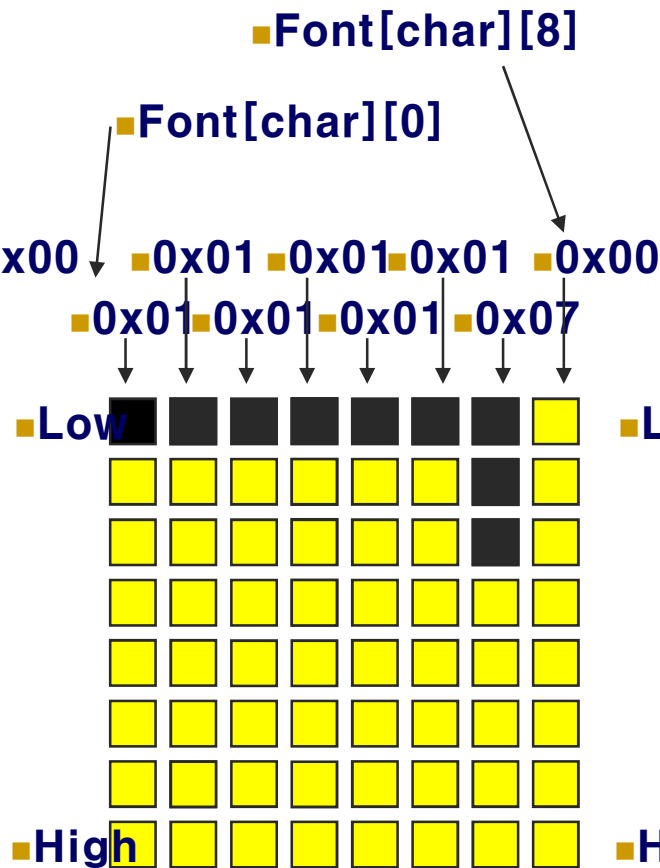


[8x8 한글 조합형 폰트(초성)]

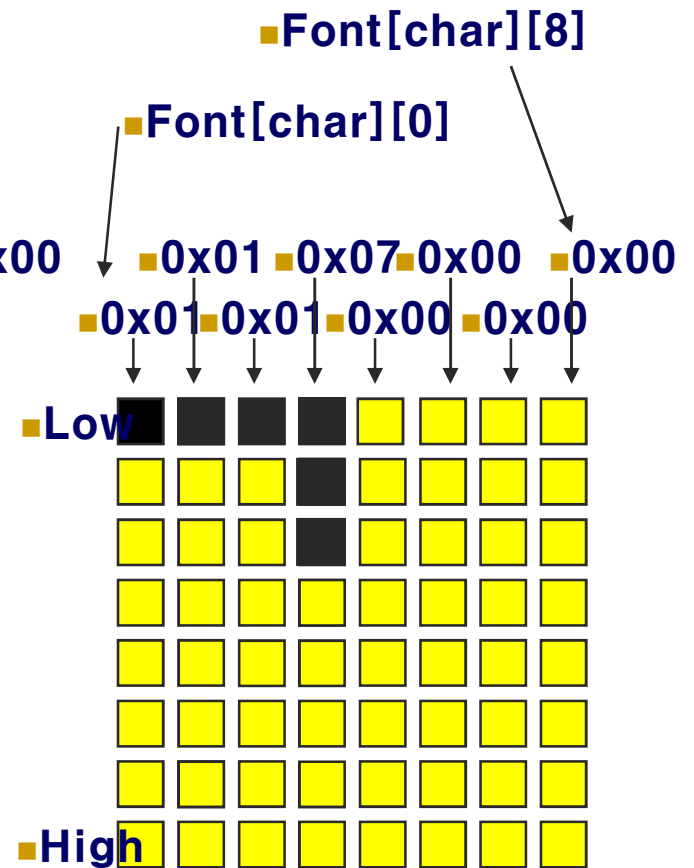
■ 초성 네번째 폰트



■ 초성 다섯번째 폰트



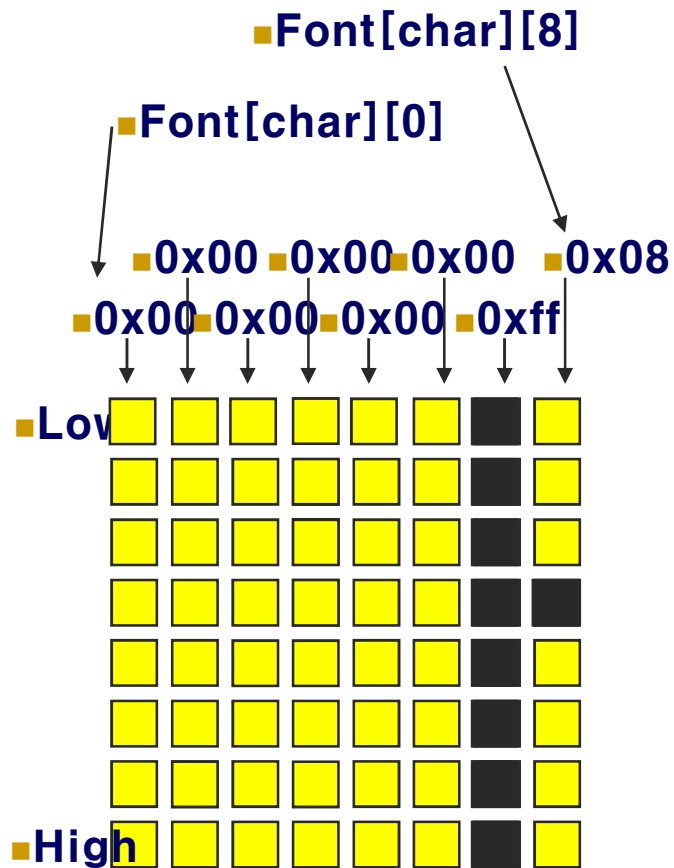
■ 초성 여섯번째 폰트



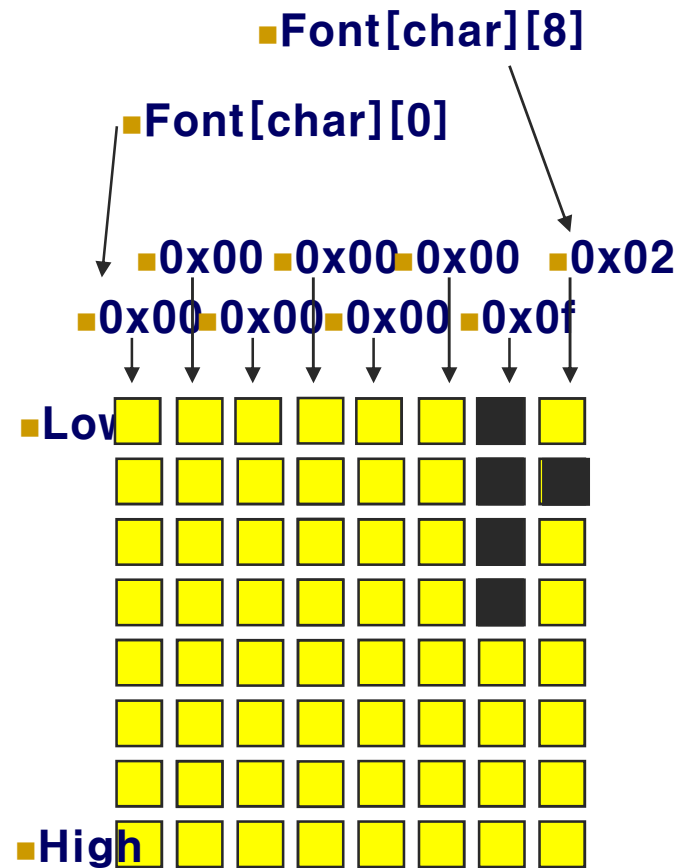
8x8 한글 조합형 폰트(중성)

[1 타입]

■ 중성 첫번째 폰트



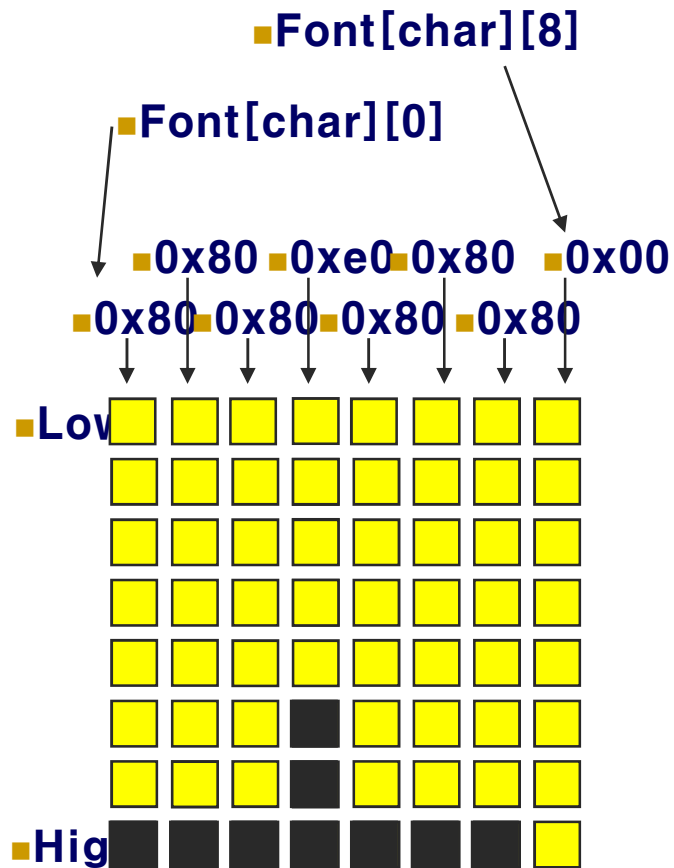
■ 중성 두번째 폰트



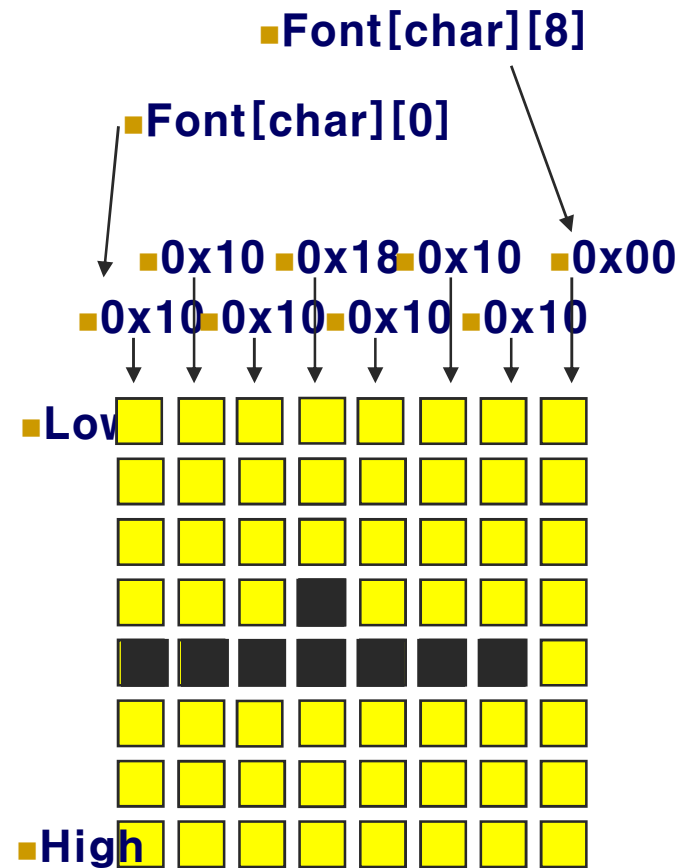
8x8 한글 조합형 폰트(중성)

[┐ 타입

■ 중성 첫번째 폰트



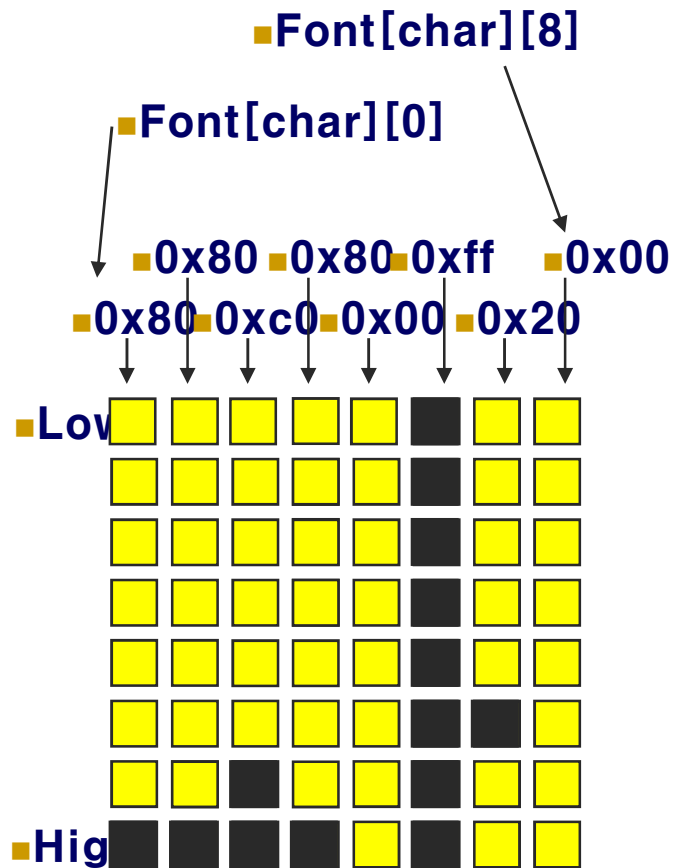
■ 중성 두번째 폰트



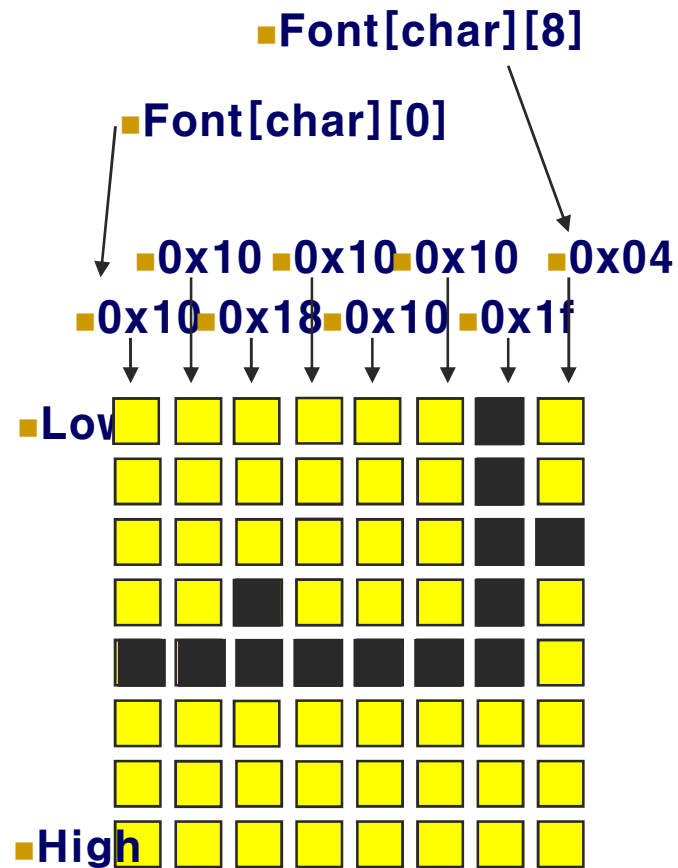
8x8 한글 조합형 폰트(중성)

[나 타입]

■ 중성 첫번째 폰트

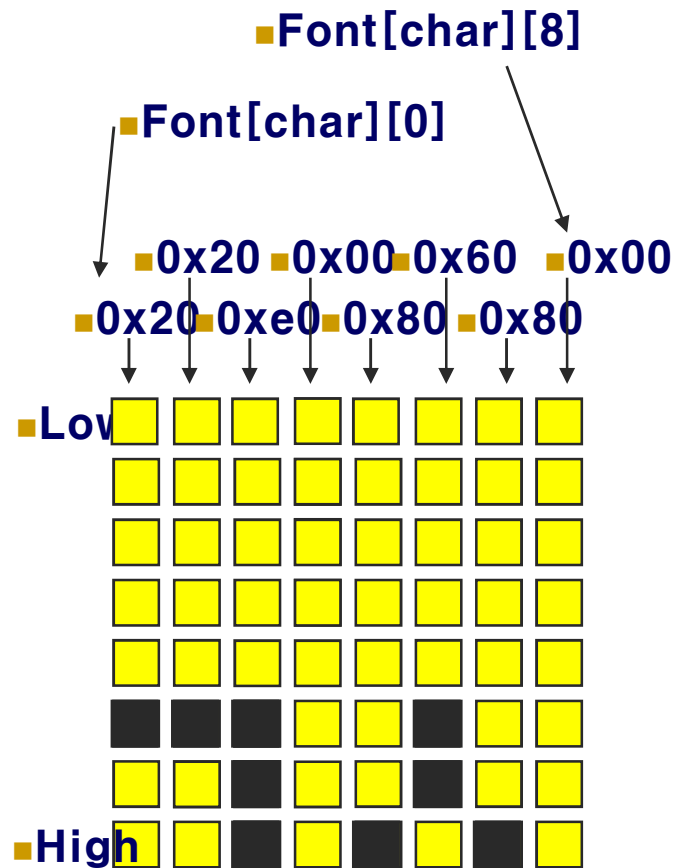


■ 중성 두번째 폰트

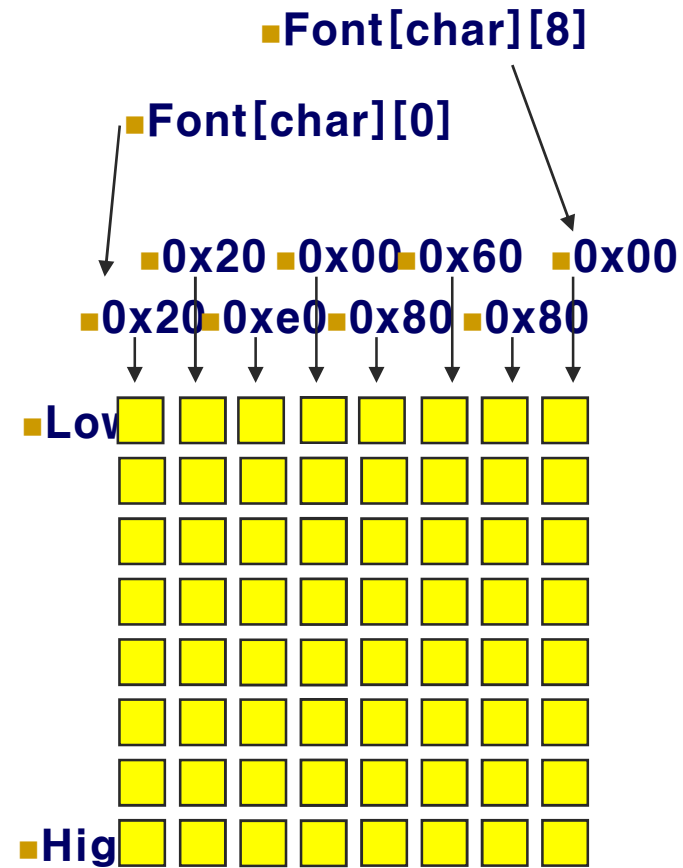


[8x8 한글 조합형 폰트(종성)]

■ 종성 첫번째 폰트



■ 종성 없는 폰트



[DS1307의 메모리 주소]

00H

SECONDS
MINUTES
HOURS
DAY
DATE
MONTH
YEAR
CONTROL
RAM 56 x 8

07H

08H

3FH

BIT7								BIT0	
00H	CH	10 SECONDS			SECONDS				00-59
	0	10 MINUTES			MINUTES				00-59
	0	12 / 24	10 HR A/P	10 HR	HOURS			01-12 00-23	
	0	0	0	0	0	DAY		1-7	
	0	0	10 DATE		DATE			01-28/29 01-30 01-31	
	0	0	0	10 MONTH	MONTH			01-12	
	10 YEAR				YEAR				00-99
07H	OUT	0	0	SQWE	0	0	RS1	RS0	

[I2C 사용 방법

]

- 초기화

- `init_ds1307();`

- 사용

- 쓰기 : `writebyte(0x00,0);` //초 쓰기

- 읽기 : `second = readbyte(0x00);` // 초 읽기

[I2C로 시간(RTC) 관련 사용]

- `writebyte(0x00,0); //초 쓰기`
- `writebyte(0x01,0); //분 쓰기`
- `writebyte(0x02,0); //시 쓰기`
- `second = readbyte(0x00); //초 읽기`
- `min = readbyte(0x01); //분 읽기`
- `hour = readbyte(0x02); //시 읽기`

- `// DS1307에서 각 값들이 튀는 걸 보정하기`
- `second = 10*((second&0x70)>>4)+(second&0x0F);`
- `min = 10*((min&0x70)>>4)+(min&0x0F);`
- `hour = 10*((hour&0x10)>>4)+(hour&0x0F);`

[스탑와치 만들기]

- PG2와 PG3 스위치를 이용하여, 텍스트 LCD로 스톱와치를 구현하세요.
 - 핀의 기능
 - PG2 : START / STOP
 - PG3 : RESET
 - 방법
 - 1/100 초 타이머 구현
 - 1초 타이머 구현
 - 시계를 증가시켜 가면서 세븐 세그먼트로 출력한다.
 - 시간을 맞추는 모드를 추가한다.
 - 알람을 집어 넣어서, LED를 번쩍거린다.
 - 제한조건 : 필수적으로 printf 문을 사용하세요.

[시리얼 LCD 제작하기]

- 컴퓨터의 터미널에서 입력하는 데이터가 KD-128PRO 키트를 거쳐서 텍스트 LCD에 표시되도록 프로그래밍 하시오.
- 방법
 - scanf 함수를 이용하여 UART0에서 입력받은 데이터를 printf 함수를 이용하여 TEXTLCD로 출력한다.
 - 텍스트 LCD 제어 포트 : PORTA
 - 텍스트 LCD 데이터 포트 : PORTF

[한글 출력하기]

- 텍스트 LCD의 CGRAM을 이용하여 한글을 출력하도록 프로그래밍 하시오.
- 방법
 - CGRAM에 해당하는 한글을 기록한 후,
 - 그 CGRAM과 연결된 해당 DDRAM 주소를 이용하여 글을 표시한다.
 - 이 때, 한글 폰트를 이용하여, 매 번 기록한 후, 출력한다.