

태터데스크 관리자

[도움말](#)
[적용하기](#)
[첫페이지 만들기](#)

태터데스크 메시지

저장하였습니다.

전문가가 되는 그날까지!!

[홈](#) | [위치로그](#) | [태그](#) | [미디어로그](#) | [방명록](#) | [관리자](#) | [글쓰기](#)

[블로그 구독하기](#)

텍스트 LCD에서 사용자 정의문자 지정하기

| [ARM7](#)

2009.06.04 10:52 [신고](#)

★ 텍스트형 LCD모듈에서의 사용자 정의문자

- 지난 글에서 실습해 보았듯이 텍스트형 LCD 모듈에서 글자를 찍을 때 우리는 거의 CG ROM에 저장된 기본

문자 폰트를 이용하여 문자의 코드를 DD RAM에 써주기만 하면 자동으로 CG ROM에서 찾아 화면에 디스플레이

레이 시켜주었다. (CG ROM의 문자코드는 문자의 아스키코드값과 일치)

이번 글에서는 CG ROM에 정의된 문자 대신 CG RAM을 이용하여 사용자 정의 문자를 지정해보겠다.

★ CG RAM

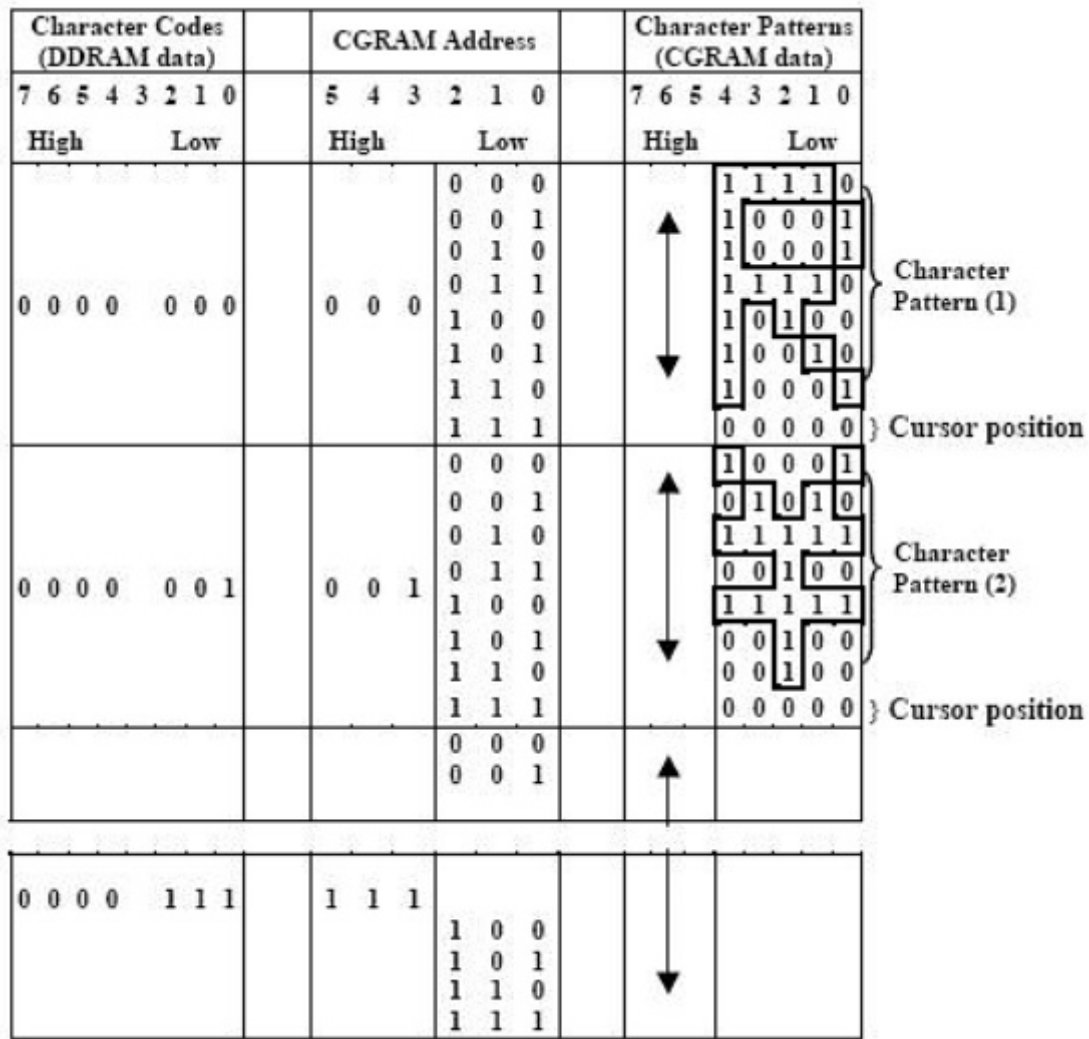
- 사용자 정의 문자를 지정하는데 사용하는 메모리
- 5x7 도트를 사용하는 경우 최대 8문자까지 정의 가능
- 5x10도트를 사용하는 경우 최대 4문자까지 정의 가능
- 사용자 정의 문자는 0x00~0x0F 영역에 할당되어 있는데 1문자가 2개의 코드 영역을 차지하기 때문에

실제 코드 값은 0x00~0x07(또는 0x08~0x0F)로 사용

★ 텍스트형 LCD 모듈에서의 사용자 정의문자 지정방법

Register
Address hex.

	CGRAM Memory Registers	LCD Display															
00	<table><tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>			0	1	0	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								<p>First symbol in CGRAM memory (latin letter c in lowercase)</p> <p>Symbol Address: 0000 0000</p>
		0	1	0	1	0											
01	<table><tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>			0	0	1	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	1	0	0											
02	<table><tr><td></td><td></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>			0	1	1	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	1	1	1	0											
03	<table><tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>			1	0	0	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	0	0	0	0											
04	<table><tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>			1	0	0	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	0	0	0	0											
05	<table><tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>			1	0	0	0	1	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	0	0	0	1											
06	<table><tr><td></td><td></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>			0	1	1	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	1	1	1	0											
07	<table><tr><td></td><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>			0	0	0	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	0	0	0											
08	<table><tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>			0	1	0	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								<p>Second symbol in CGRAM memory (latin letter z in lowercase)</p> <p>Symbol Address: 0000 0001</p>
		0	1	0	1	0											
09	<table><tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>			0	0	1	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	1	0	0											
0A	<table><tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>			1	1	1	1	1	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	1	1	1	1											
0B	<table><tr><td></td><td></td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>			0	0	0	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	0	1	0											
0C	<table><tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>			0	0	1	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	1	0	0											
0D	<table><tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>			0	1	0	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	1	0	0	0											
0E	<table><tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>			1	1	1	1	1	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	1	1	1	1											
0F	<table><tr><td></td><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>			0	0	0	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	0	0	0											
10	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																
11	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																
12	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																
14	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																
38	<table><tr><td></td><td></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>			0	1	1	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								<p>Eight symbol in CGRAM memory (figure)</p> <p>Symbol Address: 0000 0111</p>
		0	1	1	1	0											
39	<table><tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>			1	0	0	0	1	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	0	0	0	1											
3A	<table><tr><td></td><td></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>			0	1	1	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	1	1	1	0											
3B	<table><tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>			0	0	1	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	1	0	0											
3C	<table><tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>			1	1	1	1	1	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	1	1	1	1											
3D	<table><tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>			0	0	1	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	0	1	0	0											
3E	<table><tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>			0	1	0	1	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		0	1	0	1	0											
3F	<table><tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>			1	0	0	0	1	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
		1	0	0	0	1											



- 위 그림에서 보듯이 문자코드는 하위 3비트만 유효하고 이에 코드 값은 0x00~0x07로 지정해야한다.
 - CGRAM Address에서 최상위 2비트는 01이다. 따라서 CGRAM의 첫 시작 주소는 0x40이 된다.
 - 예를 들어 문자코드 0x00의 문자폰트를 써넣으려면 먼저 CG RAM의 어드레스 지정명령 0x40을 명령 번지
에 쓰고 다음에 8개의 CGRAM 데이터를 순서대로 Data번지에 쓴다.(이 때 Entry mode set명령에서 커서
를 증가시키는 방향으로 설정했다면 자동으로 1씩 증가되어 8개의 폰트 데이터를 순서대로 써넣을 수 있
다.)
 - 문자 폰트 데이터는 도트를 검게 표시하고 싶은 부분의 비트를 1로 하고 공백부분은 0으로 한다.
 - 5x8 도트의 경우 마지막 8번째 줄은 커서가 나타나는 위치이므로 실제로는 5x7도트라 생각하고 구성한다.
- ★ 5x7 도트 문자 폰트 데이터 생성

```

○ ○ ● ○ ○ 0x04
○ ● ○ ● ○ 0x0A
● ○ ○ ○ ● 0x11
● ○ ○ ○ ● 0x11
● ○ ○ ○ ● 0x11
○ ● ○ ● ○ 0x0A

```

○ ○ ● ○ ○ 0x04
○ ○ ○ ○ ○ 0x00

★ 사용자 정의 문자 디스플레이 예제

```
/* -----*/
/* 2009 EHA Source - By Sung-bin Lee, 2009.06.04. */
/* -----*/
#include "project.h"

// CLCD 사용자 정의 문자

#define D_BUS 0x000000FF // Data Bus (PA0~PA7)
#define RS 0x00000800 // Register Select(1=Data, 0=Instruction) (PA11)
#define RW 0x00001000 // Read/Write(1=Read, 0=Write) (PA12)
#define EN 0x00002000 // LCD Enable (PA13)
#define BL 0x00004000 // BackLight cathod (1=Off, 0=On) (PA14)

void LCD_Instruction(unsigned char ucData);
void LCD_Data(unsigned char ucData);
void LCD_Init();

int main()
{
    volatile unsigned int iCnt = 0;

    unsigned int pName[] = {
        0x04, 0x0A, 0x11, 0x11, 0x11, 0x0A, 0x04, 0x00, // 'ㅇ' 0x00
        0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, // 'ㅣ' 0x01
        0x00, 0x04, 0x0A, 0x11, 0x11, 0x11, 0x00, 0x00, // 'ㄴ' 0x02
        0x01, 0x01, 0x01, 0x0F, 0x01, 0x01, 0x01, 0x00, // 'ㅏ' 0x03
        0x11, 0x11, 0x11, 0x1F, 0x11, 0x11, 0x1F, 0x00, // 'ㅑ' 0x04
        0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, // 'ㅣ' 0x05
        0x10, 0x10, 0x10, 0x1F, 0x00, 0x00, 0x00, 0x00, // 'ㄷ' 0x06
        0x00, 0x0A, 0x15, 0x11, 0x0A, 0x04, 0x00, 0x00}; // '♡' 0x07

    *PIO_PER = D_BUS | RS | RW | EN | BL; // PIO Enable (PA0~PA7, PA11, PA12, PA13, PA14)
    *PIO_OER = D_BUS | RS | RW | EN | BL; // PIO Output Enable (PA0~PA7, PA11, PA12, PA13, PA14)

    LCD_Init(); // LCD 초기화
    LCD_Instruction(0x06); // Entry Mode Set (커서 오른쪽증가, shift Off)

    LCD_Instruction(0x40);
    for(iCnt=0; iCnt<64; ++iCnt){
        LCD_Data(pName[iCnt]);
    }

    LCD_Instruction(0x80); // DD RAM Address (첫라인 첫위치)
    LCD_Data(0x07); // '♡'

    LCD_Instruction(0x82);
    LCD_Data(0x00); // 'ㅇ'
    LCD_Data(0x01); // 'ㅣ'

    LCD_Instruction(0x85);
    LCD_Data(0x02); // 'ㄴ'
    LCD_Data(0x03); // 'ㅏ'

    LCD_Instruction(0x88);
    LCD_Data(0x04); // 'ㅑ'
    LCD_Data(0x05); // 'ㅣ'

    LCD_Instruction(0x8B);
    LCD_Data(0x07); // '♡'

    LCD_Instruction(0xC0); // DD RAM Address (두번째 라인 첫위치)
    LCD_Data(0x07); // '♡'

    LCD_Instruction(0xC6);
    LCD_Data(0x00); // 'ㅇ'
```

```

LCD_Instruction(0xC9);
LCD_Data(0x06);      // '└'

LCD_Instruction(0xCB);
LCD_Data(0x07);      // '♡'

return 0;
}

void LCD_Init()
{
    LCD_Instruction(0x38); // Function Set (8bit DataLength, 2Line, 5x7Dot)
    LCD_Instruction(0x0c); // Display On/Off control (화면표시On, 커서Off, 커서깜빡임Off)
    LCD_Instruction(0x1);  // Display Clear
}

void LCD_Instruction(unsigned char ucData)
{
    volatile unsigned int iCnt = 0;

    *PIO_CODR = EN;      // LCD 비활성화
    *PIO_CODR = RS;      // RS = 0 명령모드
    *PIO_CODR = RW;      // RW = 0 Write
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 40nSec이상)
    *PIO_SODR = EN;      // LCD 활성화
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 150nSec이상)
    *PIO_CODR = D_BUS;   // DataBus 초기화
    *PIO_SODR = ucData;
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 80nSec이상)
    *PIO_CODR = EN;      // LCD 비활성화
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 10nSec이상)

    return;
}

void LCD_Data(unsigned char ucData)
{
    volatile unsigned int iCnt = 0;

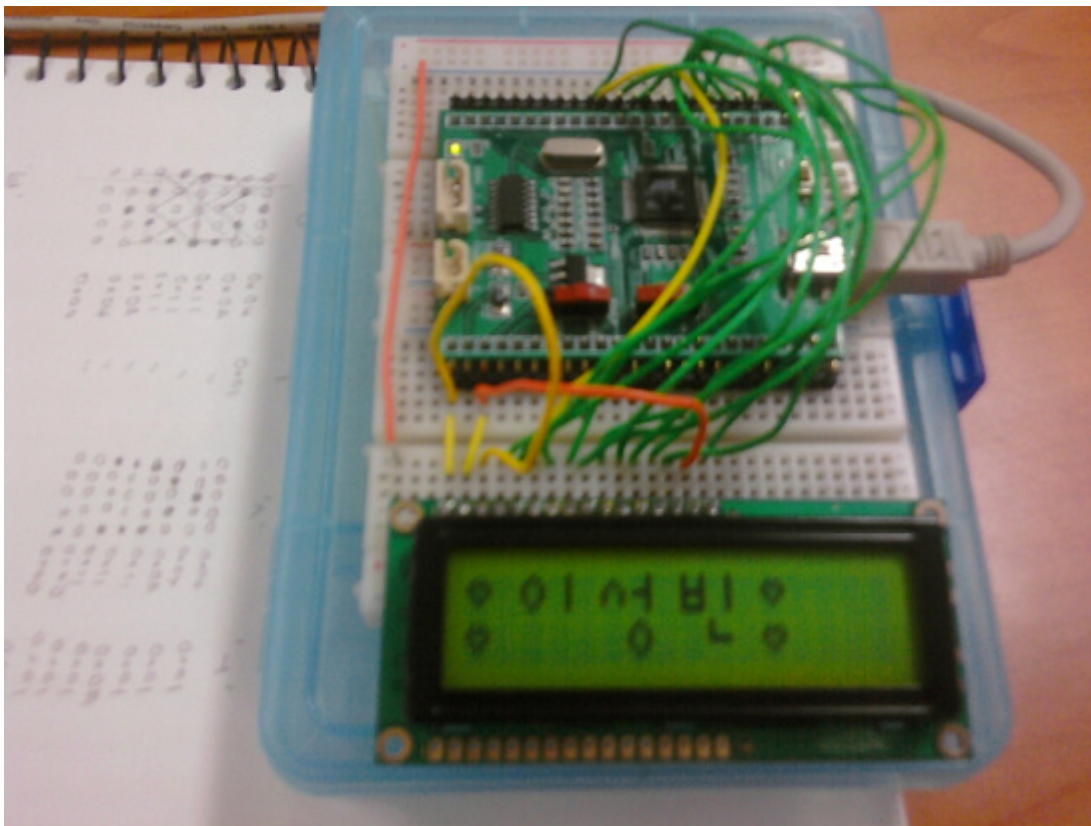
    *PIO_CODR = EN;      // LCD 비활성화
    *PIO_SODR = RS;      // RS = 1 Data모드
    *PIO_CODR = RW;      // RW = 0 Write
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 40nSec이상)
    *PIO_SODR = EN;      // LCD 활성화
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 150nSec이상)
    *PIO_CODR = D_BUS;   // DataBus 초기화
    *PIO_SODR = ucData;
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 80nSec이상)
    *PIO_CODR = EN;      // LCD 비활성화
    for(iCnt=0; iCnt<=10000;++iCnt); // delay (최소 10nSec이상)

    return;
}

```

★ 위 소스를 이용한 실제 구동 사진

- 브레드보드가 협소한 관계로 점프선이 정신없더라도 이해바람 ^.^;



'ARM7' 카테고리의 다른 글

ADC를 이용한 값 변환 출력 (0)	2009.06.08
타이머카운터0(TC0)를 이용한 인터럽트 (1)	2009.06.08
<u>텍스트 LCD에서 사용자 정의문자 지정하기</u> (3)	2009.06.04
텍스트 LCD 제어 (0)	2009.06.03
디버그 유닛(DBGU)의 UART동작 (1)	2009.05.28
PA0를 이용한 LED ON/OFF 제어 (0)	2009.05.28

Posted by 골골이

[트랙백 0](#) : [댓글 3](#)

[<< PREV](#) : [\[1\]](#) : ... [\[68\]](#) : [\[69\]](#) : [\[70\]](#) : [\[71\]](#) : [\[72\]](#) : [\[73\]](#) : [\[74\]](#) : [\[75\]](#) : [\[76\]](#) : ... [\[100\]](#) : [NEXT >>](#)



부족한 자료지만 이용하실땐 댓글이라도 달아주심 힘나요!ㅎ by 골골이

공지사항

카테고리

[전체 보기](#) (100)

[C언어](#) (9)

- C++ (37)
- Win API (8)
- MFC (9)
- ATmega128 (6)
- ARM7 (11)
- 지그비 (1)
- 네트워크 (8)
- 어셈블리어 (4)
- 프로젝트 (0)
- 자격증 정보 (3)
- 유용 (4)

최근에 올라온 글

- [헤더파일에 선언된 변수처..](#)
- [mfcs80d.lib\(dllmodul.obj\)..](#)
- [Run-Time Check Failure #0](#)
- [scanf s함수 \(1\)](#)
- [gcc 컴파일 옵션 \(2\)](#)

최근에 달린 댓글

- [오오~ 좋은 자료 감사합니다^^ 퍼가요~ EHOzIO 2014](#)
- [감사합니다 덕분에 해결했습니다^^~* 많 2014](#)
- [잘보고갑니다 ^ ^!! NATA 2013](#)
- [이해하기 쉽게 잘 설명된 글이네요 ^^ 잘.. tackbro 2013](#)
- [L'ancien conseiller de Bernard Kouchne.. moncler online 2013](#)

최근에 받은 트랙백

- <http://photodune.net>
<http://photodune.net> 2014
- pikchur.com
pikchur.com 2014

Total : 102,226

Today : 65 Yesterday : 102



[티스토리 가입하기!](#)



태그목록

글 보관함

- [2010/10](#) (2)
- [2010/03](#) (1)
- [2009/11](#) (3)
- [2009/10](#) (2)

- [2009/09](#) (4)

달력

		≪ 2016/06 ≫			
일	월	화	수	목	금 토
			1	2	3 4
5	6	7	8	9	10 11
12	13	14	15	16	17 18
19	20	21	22	23	24 25
26	27	28	29	30	

링크

- [까만닭](#)
- [술이형](#)
- [WoWcat](#)

[홈](#) | [위치로그](#) | [태그](#) | [미디어로그](#) | [방명록](#) | [관리자](#) | [글쓰기](#)
 꿀꿀이's Blog is powered by [Daum](#) / Designed by [Tistory](#)