

사용 설명서

(SONIX SN8F27E65 Test Board)

Version 1.0



- 회사명 : DIWELL electronics Co., LTD
- 전 화 : 070-8235-0820
- F A X : 031-429-0821
- E-Mail : expoeb2@diwell.com

목 차

1. 제품 소개
2. SN8F27E65 Starter Kit 구성
3. 각부 명칭 및 상세 설명
 - 3-1. 사용을 위한 설치방법
 - 3-2. Tact Switch
 - 3-3. LED/ 가변 저항/ ADC
4. Tutorial
 - 4-1. 프로그램 소스 작성 및 컴파일
 - 4-2. ISP 를 통한 다운로드
 - 4-3. ISP pin description
5. 보드 회로도

1. 제품 소개

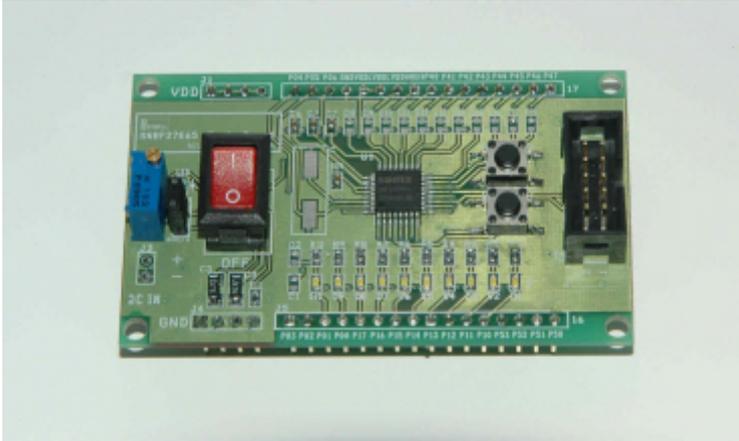
- 1-1. ㈜디웰전자의 SN8F267E65 Starter Kit 는 유사 RISC 아키텍처 방식의 8 비트 마이크로 컨트롤러(MCU)인 SONIX 社의 MCU 를 탑재하고 있습니다.
- 1-2. 본 제품은 처음 MCU 를 다루는 엔지니어 또는 학생의 프로젝트 진행 중 8 비트 MCU 구현이 필요한 경우에 쉽게 사용할 수 있도록 고안된 교육 및 개발용 Starter Kit 입니다.
- 1-3. SN8F27E65 Starter Kit 는 Sonix 社의 최신 반도체 기술이 적용된 Flash Rom type 의 **SN8F27E65** 를 탑재하고 있으며, 본 제품과 같이 구매하신 ISP 를 사용하여 프로그램을 손쉽게 디버깅할 수 있습니다. SN8F27E65 MCU 를 사용하는데 필요한 소프트웨어는 C Studio 로 <http://www.diwell.com> → Support → C Studio 에서 무료로 다운로드 하여 사용할 수 있습니다. 사용자는 C Studio 컴파일러를 사용하고 ISP 를 USB 케이블로 연결하여 프로그램 소스 컴파일, 다운로드, 디버깅 등의 모든 과정을 쉽고 간편하게 수행할 수 있습니다.
- 1-4. SN8F27E65 Starter Kit 는 모든 MCU 포트가 별도의 핀 헤더로 확장돼 있어 SMD type 의 MCU 사용을 위한 변환기판 구매 및 별도의 Artwork 작업을 하실 필요가 없습니다. 특히 12 채널 10 비트의 ADC 가 사용 가능하며, 사용자가 손쉽게 ADC test 및 눈으로 쉽게 확인을 할 수 있도록 Starter Kit 내에 가변저항과 Chip LED 가 장착되어 있습니다.
- 1-5. 보드 외형



< Fig 1.1. SN8F27E65 Starter Kit >

2. SN8F27E65 Starter Kit 구성

2-1. SN8F27E65 Starter Kit 는 Fig 2.1. 과 같은 SN8F27E65 보드, 프로그램 다운로드를 위한 ISP / USB cable 로 구성되어 있습니다.



< Fig 2.1. SN8F27E65 Starter Kit >



< ISP / USB cable >

NOTE . 본 SN8F27E65 는 **DC 5V** 전원을 사용합니다.

2-2. MCU 제원

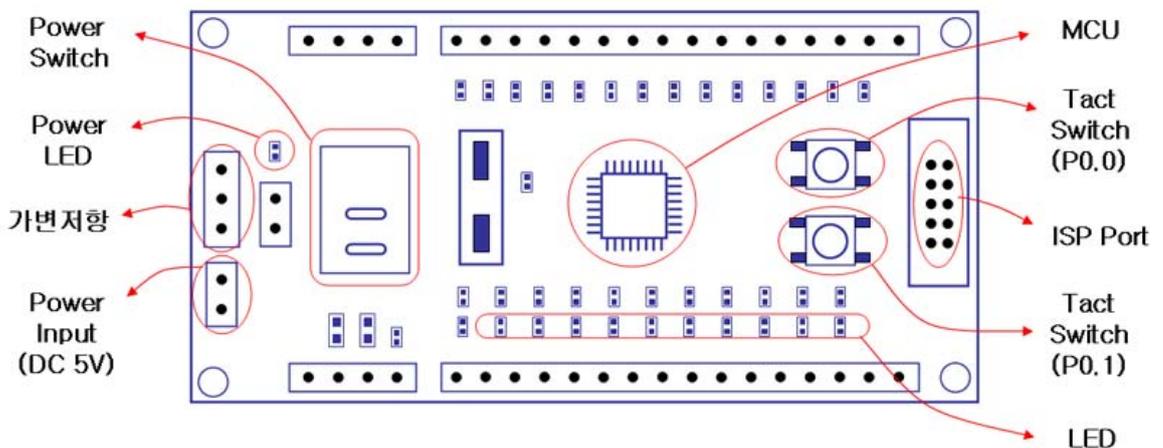
본 Starter Kit 는 Sonix 社의 MCU 제품군중 **SN8F27E65** 가 탑재 되어 있습니다. Table 2.2. 에 더욱 자세한 특징들을 표시하였습니다.

Table 2.2. Sonix SN8F27E65 Feature

Feature	Description
Powerful instructions	1 Word 의 Instruction length(대부분 1 clock cycle 으로 실행)
I/O configuration	27 개의 범용 I/O Port
	Bi-directional : P0, P1, P4, P5
	Wakeup : P0, P1 level change
	Pull-up resisters : P0, P1, P4, P5
	External interrupt : P0.0, P0.1
memory configuration	6K x 16 bits 내부 Flash(Including EEROM emulation)
	512 x 8 bits 내부 RAM
Stack buffer	8 Levels

4 개의 8 비트 타이머	T0 : Basic Timer	
	TC0 : Timer / Counter / PWM0	
	TC1 : Timer / Counter / PWM1	
	TC2 : Timer / Counter / PWM2	
16 비트 타이머	T1 (Capture)	
3 개의 PWM 채널	PWM 0~2(Buzzer and IR carrier signals)	
ADC	12-channel 10-bit SAR ADC	
Serial Interface	SPI(Master/slave), UART, MSP	
4 개의 system clocks	External high clock	RC type up to 10Mhz
		Crystal type up to 16Mhz
	Internal high clock :RC type 16Mhz	
Internal low clock : RC type 16khz		
4 개의 동작 모드	Normal mode : Both high and low clock active	
	Slow mode : Low clock only	
	Sleep mode : Both high and low clock stop	
	Green mode : Periodical wakeup by timer	
동작 전압	1.8V ~ 5.5V	
Package	DIP32/ LQFP32/ QFN32	

3. 각부 명칭 및 상세 설명



3-1. 사용을 위한 설치

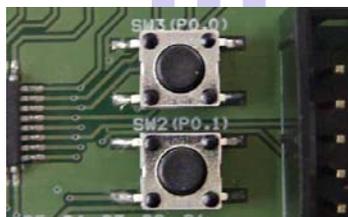


Fig 3.1. PC에서 ISP 인터페이스를 통하여 MCU로 직접 다운로드/ 디버깅

ISP는 USB HID Class Driver와 Plug & Play 지원으로 따로 사용자가 드라이버를 다운받아 설치하실 필요가 없습니다.

3-2. Tact Switch

사용자의 Button 입력을 위하여 2개의 Tact Switch를 장착하였습니다.



→ SW3 (P0.0)

→ SW2 (P0.1)

각 스위치는 MCU의 일반 I/O port(P0.0, P0.1)에 연결되어 있습니다.

SN8F27E65는 MCU 내부 풀업 설정이 가능하므로 추가로 외부 풀업 저항을 연결하지 않았습니다. 오직 I/O port가 input 모드 일 경우에만 풀업 레지스터 설정을 지원합니다.

예제)

```
POM = 0x00; // Port0의 입력 설정
           // AVR의 DDR 포트 입출력 설정과 같은 과정(0:입력, 1:출력)
POUR = 0x03; // P0.0, P0.1의 내부 풀업 설정 (0:disable, 1:enable pull-up)
```

위와 같이 코딩 하게 되면 스위치가 눌러지지 않았을 경우에는 logic high 값을 가지게 되며, 스위치가 눌러지게 되면 logic low 를 가지게 됩니다.

사용자는 프로그램 상에서 2 가지 방법으로 포트 상태를 체크할 수 있습니다.

예제 1)

```

if(FP00 == 1){user code} //P0.0 상태가 1 일 경우 = 스위치가 눌러지지 않은 경우
if(FP00 == 0){user code} //P0.0 상태가 0 일 경우 = 스위치가 눌러진 경우
if(FP01 == 1){user code} //P0.1 상태가 1 일 경우 = 스위치가 눌러지지 않은 경우
if(FP01 == 0){user code} //P0.1 상태가 0 일 경우 = 스위치가 눌러진 경우
    
```

예제 2)

```

임의 변수 TEMP = FP00; //P0.0 상태가 1 인 경우 TEMP 변수에 1 저장
                TEMP = FP01; //P0.1 상태가 0 인 경우 TEMP 변수에 0 저장
    
```

위와 같이 코딩을 하게 되면 포트 상태에 따라 원하는 동작의 구현이 가능합니다.
AVR 의 TEMP = PINA; TEMP = PIND... 의 방법과 같습니다.

3 - 3 . LED / 가변저항 / ADC

본 SN8F27E65 Starter Kit 에는 10 개의 LED 가 있습니다.

일반적으로 많이 쓰는 1 Byte 의 HEX 값을 P1.0~1.7(8) 포트와 연결 되 있는 LED 로 쉽게 확인이 가능합니다. 본 Starter Kit 출하 시 기본적으로 LED 가 Shift 하는 프로그램이 다운로드 되어 있습니다. 관련 소스 자료는 <http://www.diwell.com> → Support → technical support 에서 다운로드 가능합니다.

본 Starter Kit 의 MCU 는 10 비트의 ADC 가 존재합니다. 바로 이 10bit 의 AD 값을 표현하기 위해 2 개의 LED(P0.0~0.1)와 P4.0 에 1uF 캐패시터와 10K 의 가변저항을 추가하였습니다.(Fig 3.3. 참고) 저항 값을 변화 시키면서 P4.0 의 AD 값(10bit)을 10 개의 LED 로 간단히 테스트가 가능합니다.

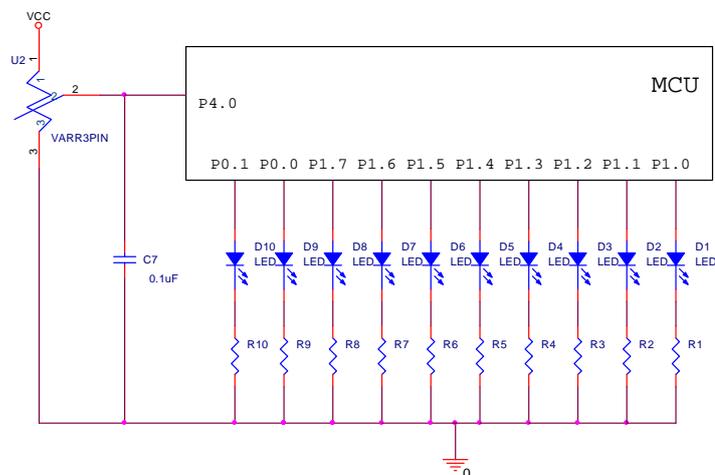


Fig 3.3. connection of 10 LED array and variable resistor

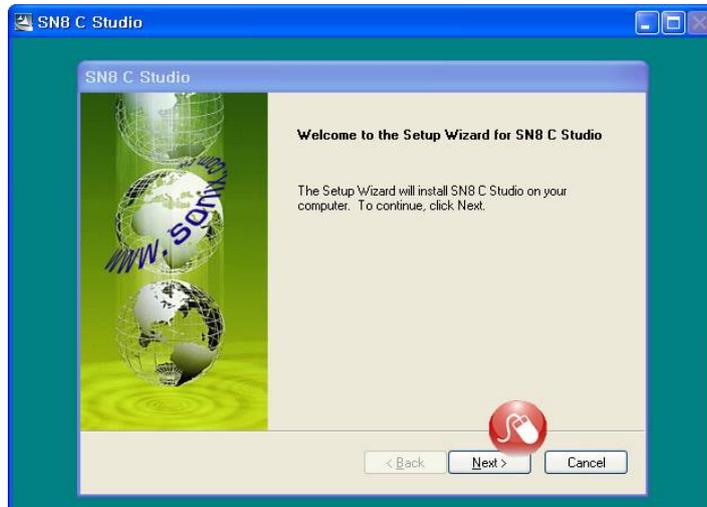


Fig 4.3. Setup C studio

4 - 1 - 2 . C Studio 실행



Fig 4.4. Click SN8 C Studio

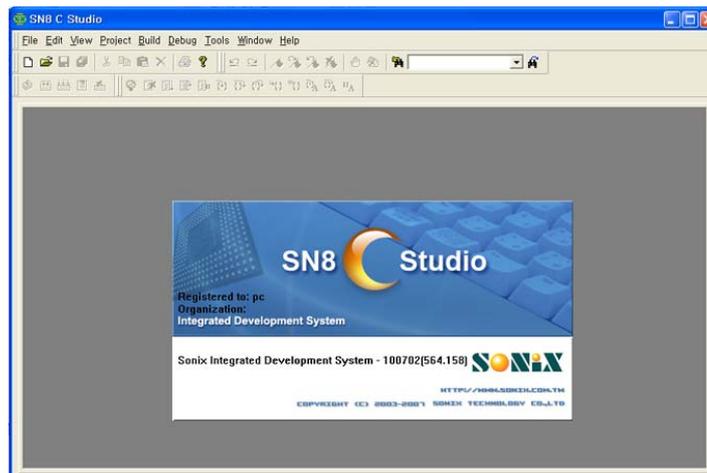


Fig 4.5. C Studio 실행 모습

4-1-3. 새 프로젝트/ 워크스페이스 생성

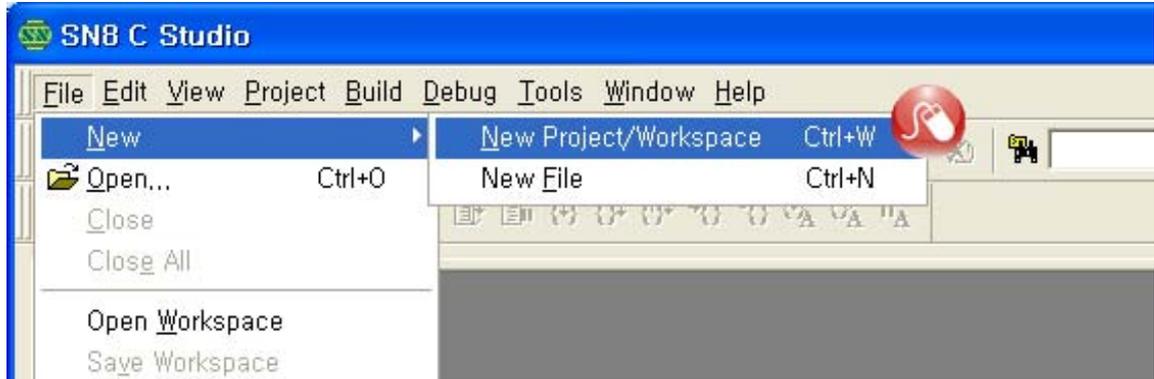


Fig 4.6. File → New → New Project/ Workspace

Workspace 가 저장될 폴더를 미리 생성 시킨 후 만든 폴더의 Location 을 지정한다.

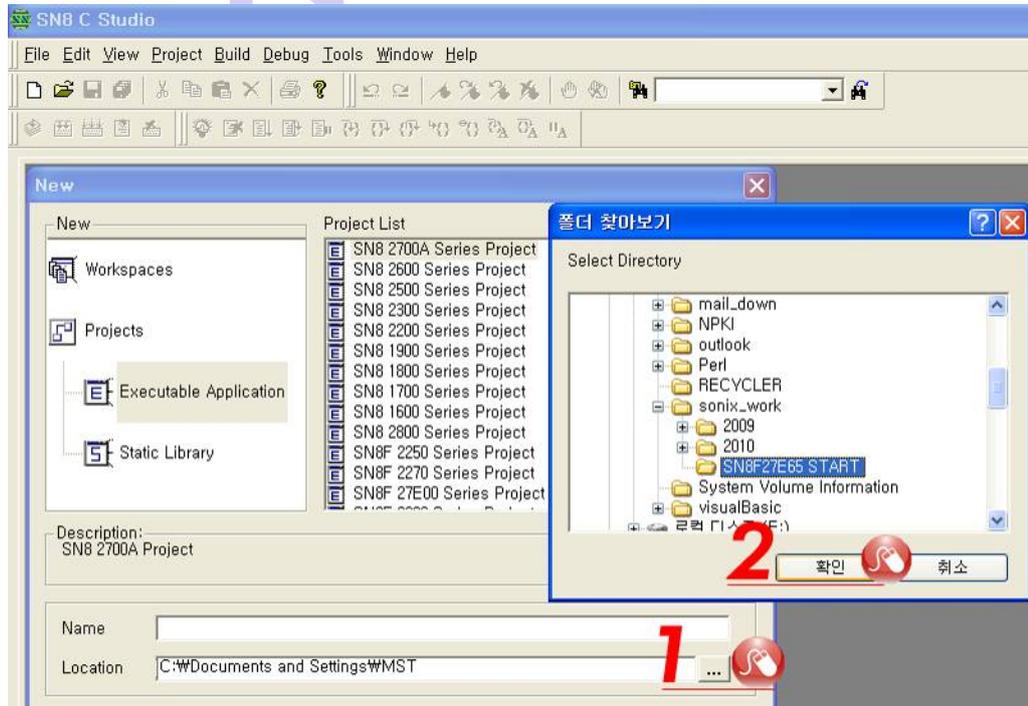


Fig 4.7. 프로젝트가 저장될 Location 지정

Name 작성 후 SN8F 27E00 Series Project 를 선택합니다.

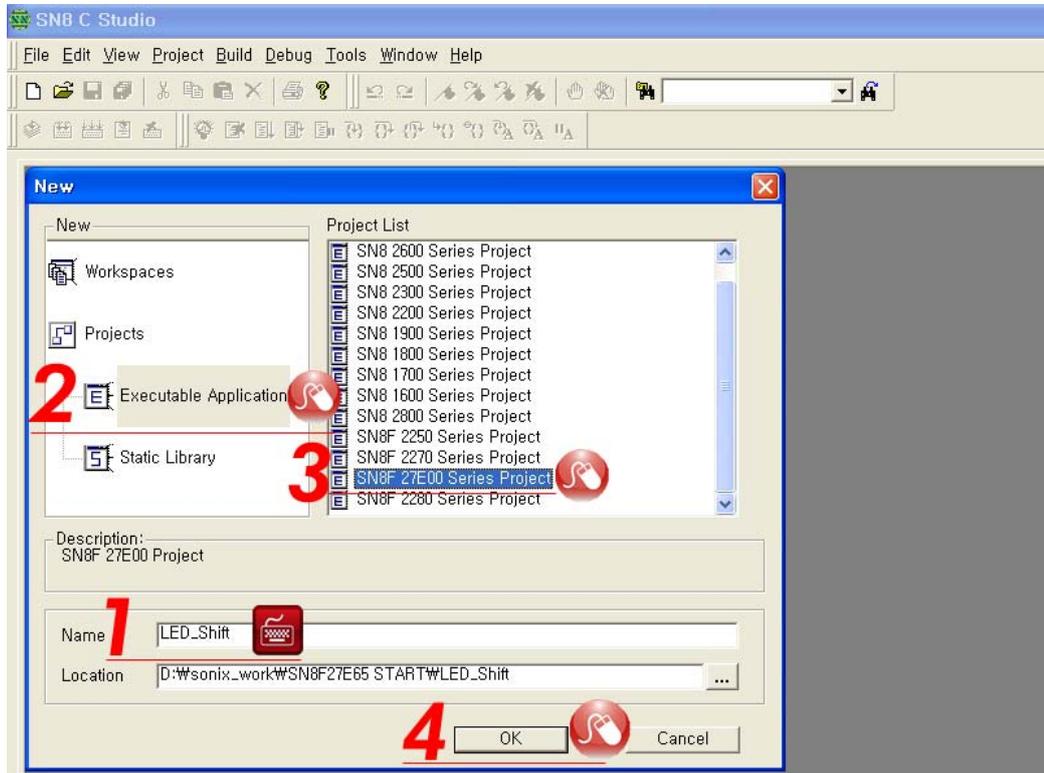


Fig 4.8. Project Description

Chip → SN8F27E65 선택 후 Code Option 으로 넘어갑니다.

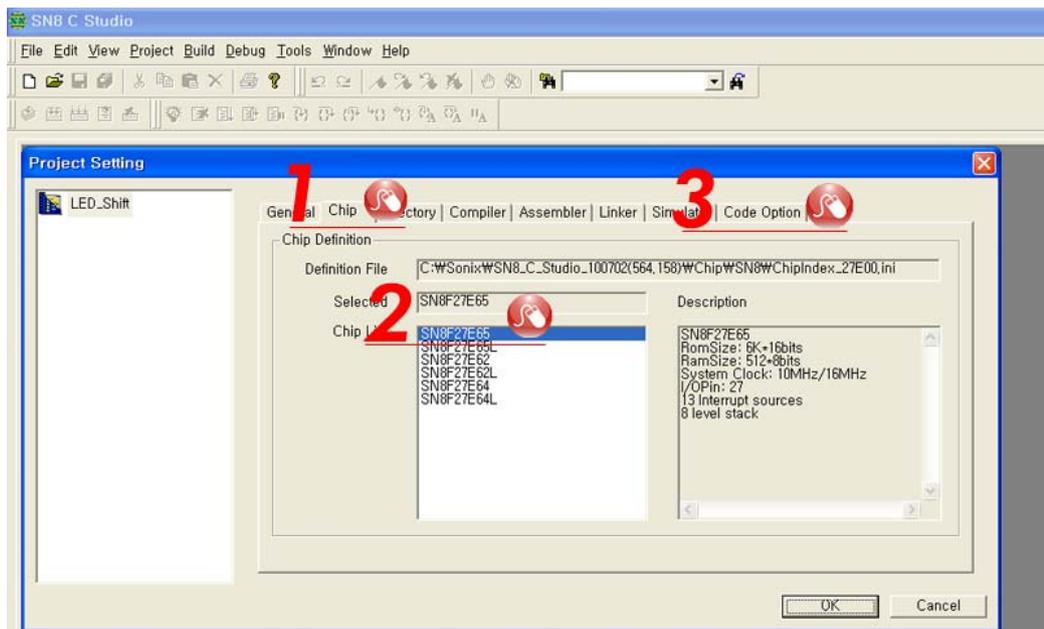


Fig 4.9. Chip Definition

코드옵션 Watch_Dog = Disable , High_Clk : IHRC_16M 로 설정하십시오
 크리스탈을 보드에 별도로 추가하실 경우 High_Clk 설정을 바꿔 주시면 됩니다.
 코드옵션에 대한 부가설명은 다음 페이지의 **Table 4.1.** 을 참고하십시오.

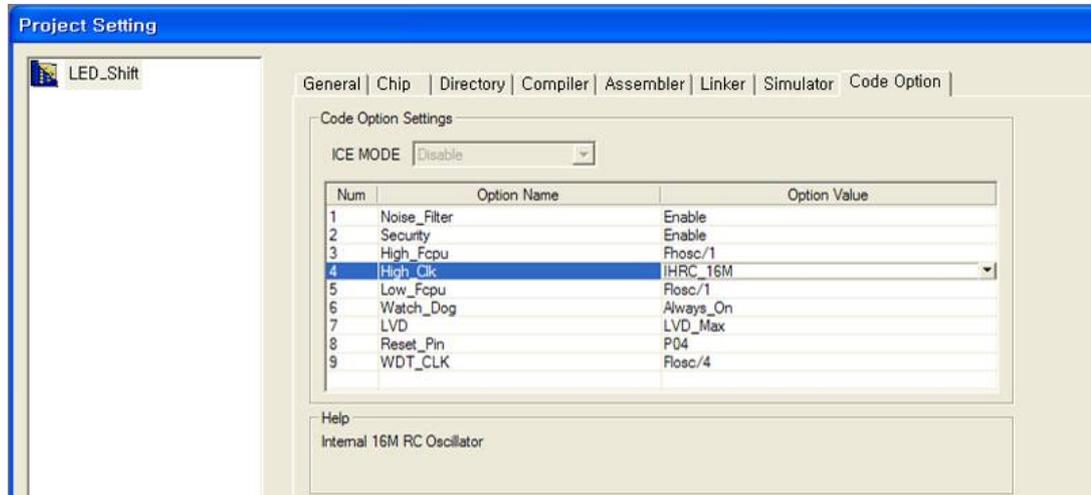


Fig 4.10. Code Option Settings

Table. 4.1 Code Option Table

Code Option	Content	Function Description
High_Clk	IHRC_16M	High speed internal 16MHz RC. XIN/XOUT pins are bi-direction GPIO mode.
	IHRC_RTC	High speed internal 16MHz RC. XIN/XOUT pins are connected to external 32768Hz crystal.
	RC	Low cost RC for external high clock oscillator. XIN pin is connected to RC oscillator. XOUT pin is bi-direction GPIO mode.
	32K X'tal	Low frequency, power saving crystal (e.g. 32.768KHz) for external high clock oscillator.
	12M X'tal	High speed crystal /resonator (e.g. 12MHz) for external high clock oscillator.
	4M X'tal	Standard crystal /resonator (e.g. 4M) for external high clock oscillator.
High_Fcpu	Fhosc/1	Normal mode instruction cycle is 1 high speed oscillator clocks.
	Fhosc/2	Normal mode instruction cycle is 2 high speed oscillator clocks.
	Fhosc/4	Normal mode instruction cycle is 4 high speed oscillator clocks.
	Fhosc/8	Normal mode instruction cycle is 8 high speed oscillator clocks.
	Fhosc/16	Normal mode instruction cycle is 16 high speed oscillator clocks.
	Fhosc/32	Normal mode instruction cycle is 32 high speed oscillator clocks.
	Fhosc/64	Normal mode instruction cycle is 64 high speed oscillator clocks.
	Fhosc/128	Normal mode instruction cycle is 128 high speed oscillator clocks.

Low_Fcpu	Fosc/1	Slow mode instruction cycle is 1 low speed oscillator clocks.
	Fosc/2	Slow mode instruction cycle is 2 low speed oscillator clocks.
	Fosc/4	Slow mode instruction cycle is 4 low speed oscillator clocks.
	Fosc/8	Slow mode instruction cycle is 8 low speed oscillator clocks.
Noise_Filter	Enable	Enable Noise Filter.
	Disable	Disable Noise Filter.
WDT_CLK	Fosc/4	Watchdog timer clock source Fosc/4.
	Fosc/8	Watchdog timer clock source Fosc/8.
	Fosc/16	Watchdog timer clock source Fosc/16.
	Fosc/32	Watchdog timer clock source Fosc/32.
Watch_Dog	Always_On	Watchdog timer is always on enable even in power down and green mode.
	Enable	Enable watchdog timer. Watchdog timer stops in power down mode and green mode.
	Disable	Disable Watchdog function.
Reset_Pin	Reset	Enable External reset pin.
	P04	Enable P0.4.
Security	Enable	Enable ROM code Security function.
	Disable	Disable ROM code Security function.
LVD	LVD_L	LVD will reset chip if VDD is below 1.8V
	LVD_M	LVD will reset chip if VDD is below 1.8V Enable LVD24 bit of PFLAG register for 2.4V low voltage indicator.
	LVD_H	LVD will reset chip if VDD is below 2.4V Enable LVD33 bit of PFLAG register for 3.3V low voltage indicator.
	LVD_MAX	LVD will reset chip if VDD is below 3.3V

워크스페이스 설정이 끝났으므로 이제 C 파일을 만들면 됩니다.

4-1-4. C 파일 생성

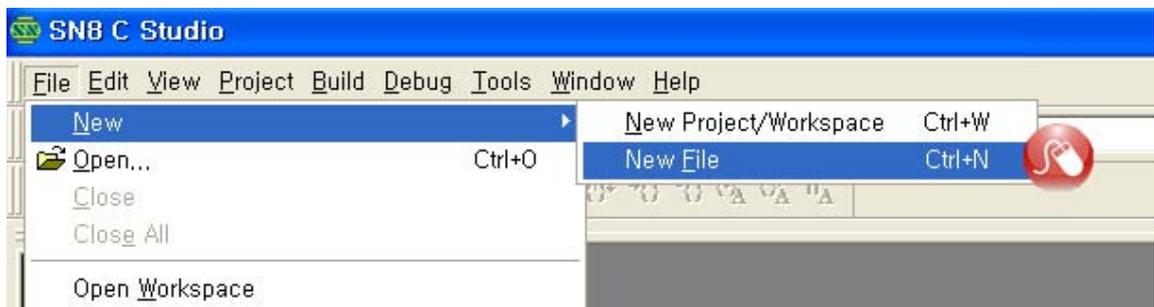


Fig 4.11. New File

File List 항목의 C Source File 클릭
 소스의 파일명을 적은 후 OK 버튼 클릭

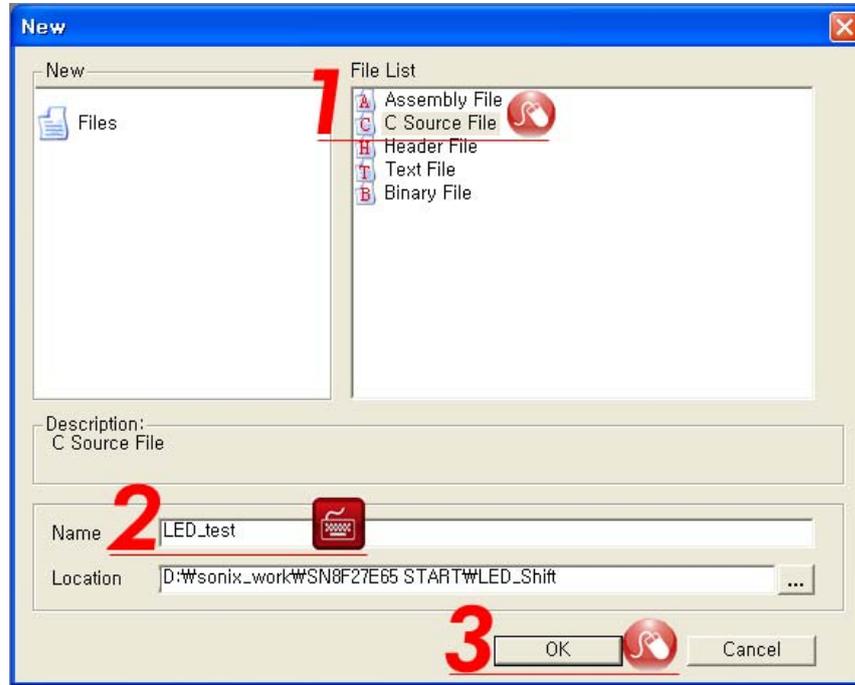


Fig 4.12. C File 생성



Fig 4.13. 파일 생성 완료

4-1-5. 간단한 예제 실습 : LED shift 프로그램 컴파일

<http://www.diwell.com> → Support → Technical Support 항목에서 예제 파일을 다운받아
 컴파일을 해보겠습니다. 파일을 다운받은 후 원하는 폴더에 압축해제 후 C Studio 를 통해
 워크스페이스를 여십시오. Open 이 아닌 Open Workspace 를 클릭하세요.

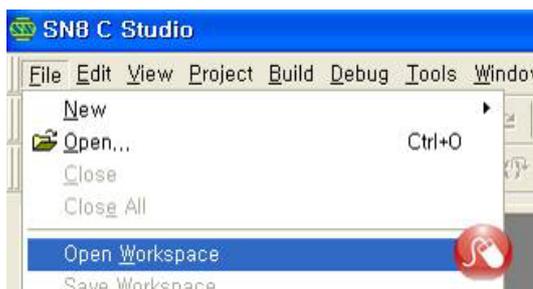


Fig 4.14. Open Workspace

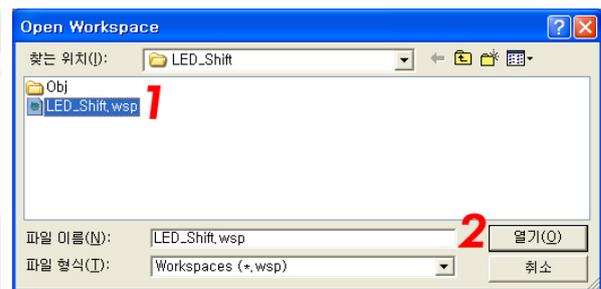


Fig 4.15. Open wsp File

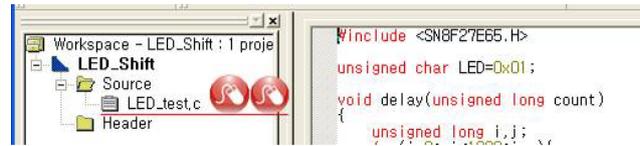


Fig 4.16. C File 더블클릭

메뉴바의 **Build → Build Project** 선택 또는 키보드 **F7** 을 누르면 컴파일이 시작됩니다.
 컴파일이 완료되면 Fig 4.17. 의 하단부에 코드 사이즈와 램 사용 여부를 알 수 있습니다.

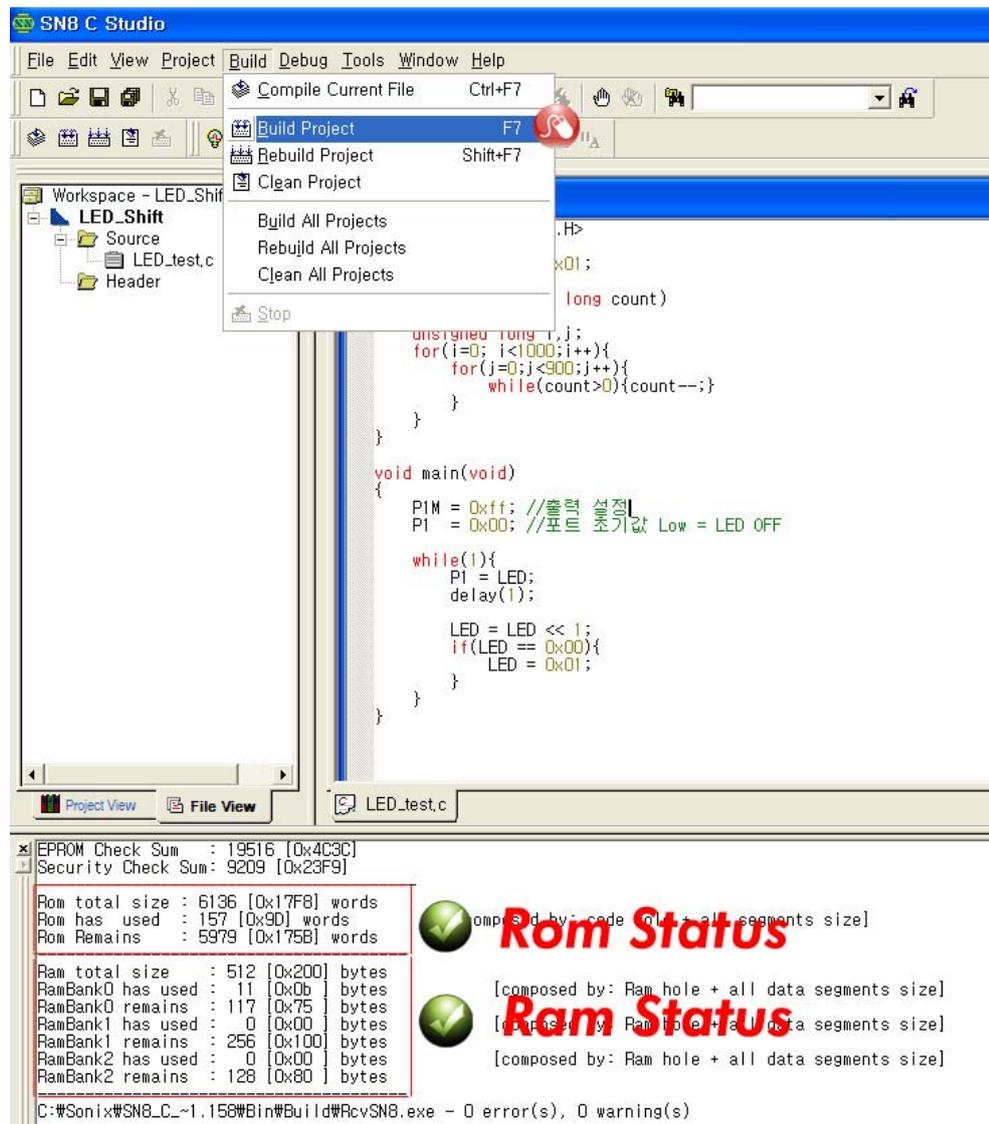


Fig 4.17. 컴파일

4-2. ISP 를 통한 다운로드

Fig 3.1. 의 그림과 같이 본 Starter Kit 와 ISP 가 정상 연결이 되면 Starter Kit 의 D1, D2 백색 LED 와 ISP 의 Run Stop 의 주황색 LED 가 들어오게 됩니다. D1, D2 와 연결된 P1.0 과 P1.1 은 ISP 다운로드를 위한 공유 포트입니다. 따라서 디버깅 과정에 있어서 항상 LED 가 들어오게 되지만, 실제 ISP 연결 해제 후 MCU 구동 시에는 영향이 없으므로 P1.0, P1.1 포트를 Debugging mode 에서 사용시 이를 염두에 두십시오.

4-2-1. Running & Debugging

메뉴바 → **Debug** → **Begin Debug** 또는 키보드 **F5** 를 누르시면 ISP 를 통해 다운로드/ 디버깅 윈도우로 바뀝니다.



Fig 4.18. ISP Download & Run

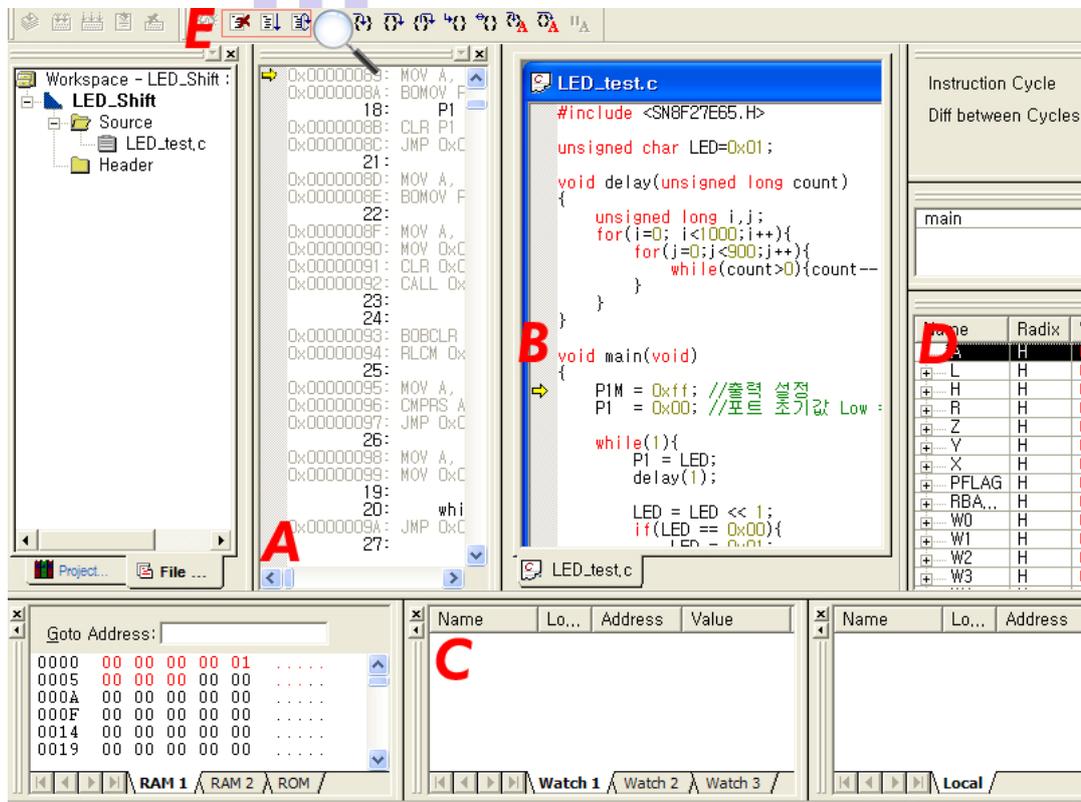


Fig 4.19. Debugging Mode

- A : 소스 코드의 동작 상태를 어셈블리어로 확인 가능합니다.
- B : C 언어로 코딩 했던 소스 내용이 보여집니다. 현재 화면에서는 편집이 불가능합니다. 편집을 위해선 **Shift+F5** 를 누르시면 다시 편집 모드로 넘어가게 됩니다.
- Break point** 설정이 가능합니다. break 지점을 더블클릭 하시거나 우클릭하여 설정합니다.
- C : 전역변수의 값을 확인 가능합니다. 값을 확인하고자 하는 변수를 블록지정 한 후 우클릭 하여 Watch 창에 추가합니다. 지역변수는 확인이 불가능 하니 변수 값을 확인 하시려면 전역변수로 변환 후 확인 하시길 바랍니다.



Fig 4.20. Add variable

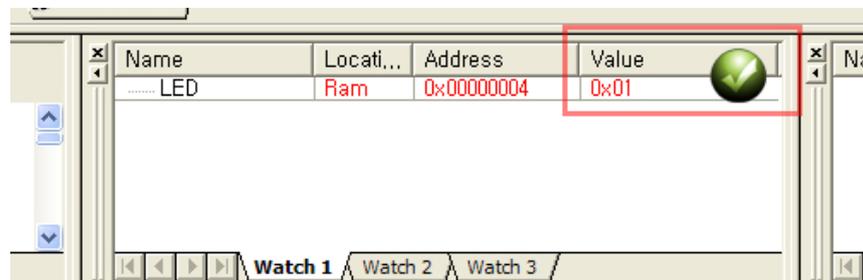


Fig 4.21. Watch Window

- D : Port 의 상태 및 모든 레지스터의 현재 값을 확인할 수 있습니다.
- E :  Exit Debug(Shift+ F5)  Run(F5)  Restart(Ctrl+ F5)

4-2-2. Running 시작/ 일시 정지 & Debugging

Fig 4.19. 화면에서 **F5** 를 누르면 Run 창이 뜨면서 ISP 의 Run Stop LED 가 녹색으로 바뀝니다. Starter Kit 의 LED 가 Shift 하기 시작하면 성공입니다. 만약 중지시키고 싶거나 현재의 상태(Resister/ 변수 값)를 확인하고자 하면 키보드 **F5** 를 누르거나 “**Stop Run**”을 클릭하시면 됩니다.

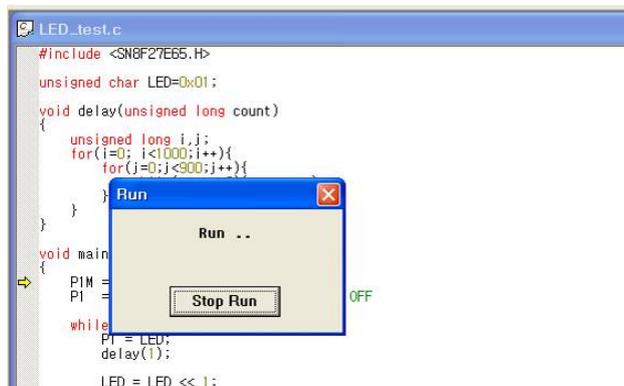


Fig 4.22. Running

4 - 3 . ISP pin description



Fig 4.23. ISP

- A : USB Port
- B : Power / Link LED(●)
- C : 디버그 모드 LED(● / ●)
- D : 10pin 케이블 커넥터

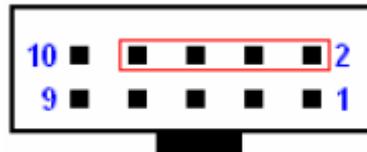


Fig 4.24. Pin assignment

Pin 지정	Pin 정의
1,3,5,7,9,10	Reserved
2	VDD
4	EICK
6	EIDA
8	GND

Table 4.2. ISP LED status information

LED 이름	정 의	LED
Power	USB 연결이 정상일 때	●
Run/ Stop	1. ISP 와 Starter Kit 와의 연결이 정상 2. debug 환경을 빠져 나왔을 때 3. 컴파일된 SN8 파일이 다운로드 완료 됐을 때	●
	1. debug 모드 진입할 때 2. SN8 파일 다운로드시	●

5. 보드 회로도

