

# 사용 설명서

## (SONIX SN8F27E00 Series)

Version 1.1



- 회사명 : DIWELL electronics Co., ltd
- 전 화 : 070-8235-0820
- F A X : 031-429-0821
- E-Mail : [expoeb2@diwell.com](mailto:expoeb2@diwell.com)

# 목 차

## 1. 제품 소개

## 2. SN8F27E00 Series 구성

## 3. 설치방법

## 4. Tutorial

### 4-1. 프로그램 소스 작성 및 컴파일

### 4-2. ISP 를 통한 다운로드

### 4-3. ISP pin description

## 5. 보드 회로도

### 5-1. SN8F27E62

### 5-2. SN8F27E64

### 5-3. SN8F27E93

## 1. 제품 소개

1-1. (주)디웰전자의 SN8F267E00 Series Test Board 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 있습니다.

1-3. SN8F27E00 Series TestBoard 는 Sonix 社의 최신 반도체 기술이 적용된 Flash Rom type 의 **SN8F27E00 Series** 를 탑재하고 있으며, 본 제품과 같이 구매하신 ISP 를 사용하여 프로그램을 손쉽게 디버깅할 수 있습니다. MCU 를 사용하는데 필요한 소프트웨어는 C Studio로 <http://www.diwell.com> → Support → C Studio에서 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. **※단, 기존 C studio 사용자가 SN8F27E93 을 사용시 반드시 최신 C studio 로 새로 설치를 하셔야 합니다.**

SN8F27E00 Series TestBoard 는 모든 MCU 포트가 별도의 핀 헤더로 확장돼 있어 MCU 사용을 위한 변환기판 구매 및 별도의 Artwork 작업을 하실 필요가 없습니다.

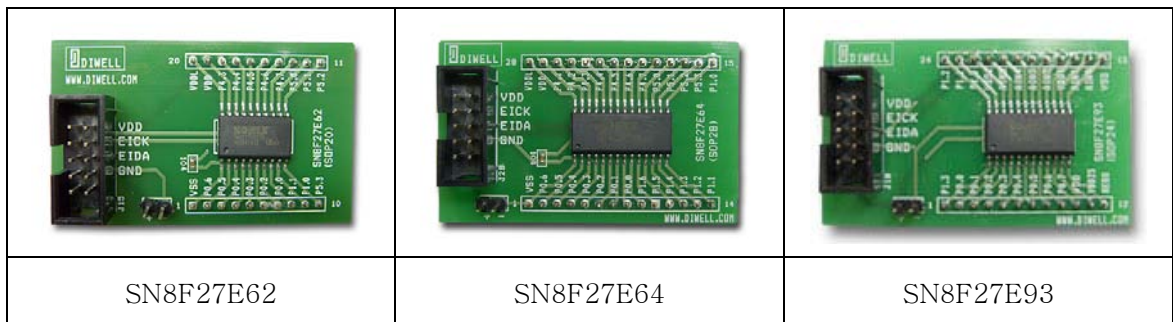
1-4. 홈페이지의 C Code 는 SN8F27E65 칩에 맞게 코딩 돼 있습니다. 코드를 다운 받으신 후 각 소스 별 Header 파일/ 포트 설정만 각 spec 을 참고하여 바꿔 주시면 됩니다.

예) `#include <SN8F27E65.H>` → `#include <SN8F27E62.H>`

`#include <SN8F27E65.H>` → `#include <SN8F27E64.H>`

`#include <SN8F27E65.H>` → `#include <SN8F27E93.H>`

### 1-5. 보드 외형



< Fig 1.1. SN8F27E00 Series Test Board >

## 2. SN8F27E00 Series 구성

2-1. SN8F27E00 Series Test Board 는 Fig 2.1. 과 같은 SN8F27E00 보드, 프로그램 다운로드를 위한 ISP / USB cable 로 구성되어 있습니다.



< Fig 2.1. SN8F27E00 Series Test Board + ISP >

2-2. 다음은 SN8F27E62/ 64/ 93 의 pin map/ Features selection table 입니다.

### 2-2-1. Pin map

VSS	1	U	20	VDDL
XIN/P0.6	2		19	VDD
XOUT/P0.5	3		18	P4.3/AIN3
RST/P0.4	4		17	P4.4/AIN4
P0.3/UTX/T1	5		16	P4.5/AIN5
P0.2/URX/TC2	6		15	P4.6/AIN6
P0.0/INT0/TC0	7		14	P4.7/AIN7
P1.1/EIDA	8		13	P5.0/AIN8
P1.0/EICK	9		12	P5.1/AIN9/PWM0
P5.3/AIN11/PWM2	10		11	P5.2/AIN10/PWM1

Fig 2.2. SN8F27E62(SOP20)

VSS	1	U	28	VDDL
XIN/P0.6	2		27	VDD
XOUT/P0.5	3		26	P4.1/AIN1
RST/P0.4	4		25	P4.2/AIN2
P0.3/UTX/T1	5		24	P4.3/AIN3
P0.2/URX/TC2	6		23	P4.4/AIN4
P0.1/INT1/TC1	7		22	P4.5/AIN5
P0.0/INT0/TC0	8		21	P4.6/AIN6
P1.6/SCK	9		20	P4.7/AIN7
P1.5/SDI	10		19	P5.0/AIN8
P1.4/SDO	11		18	P5.1/AIN9/PWM0
P1.3/SCL	12		17	P5.2/AIN10/PWM1
P1.2/SDA	13		16	P5.3/AIN11/PWM2
P1.1/EIDA	14		15	P1.0/EICK

Fig 2.3. SN8F27E64(SOP28)

P1.3/RST	1	U	24	P1.2
P0.0/INT0/TC0	2		23	P1.1/EIDA
P0.1/INT1	3		22	P1.0/EICK
P0.2/URX/SCL/SDI	4		21	VSS1
P0.3/UTX/SDA/SDO	5		20	AIN5
P0.4/SCK	6		19	AIN4
P0.5/PWM0	7		18	AIN3
P0.6/XIN	8		17	AIN2
P0.7/XOUT	9		16	VDD1
VDD	10		15	P4.1/AIN1
VDD25	11		14	P4.0/AIN0
REGO	12		13	VSS

Fig 2.4. SN8F27E93(SOP24)

2 - 2 - 2 . Features Selection Table

CHIP	ROM	RAM	Timer	I/O	PWM	ADC	SIO	UART	MSP	Ext.INT	Operating Voltage
SN8F27E62	6K*16	512	8-bit*4 16-bit*1	17	3-ch	10bit 9-ch	-	O	-	1	1.8V~5.5V
SN8F27E64	6K*16	512	8-bit*4 16-bit*1	25	3-ch	10bit 11-ch	O	O	O	2	1.8V~5.5V
SN8F27E93	16K*16	1K	8-bit*2 16-bit*1	14	1-ch	12bit 6-ch	O	O	O	2	2.3V~5.5V

### 3. 설치 방법

ISP 는 USB HID Class Driver 와 Plug & Play 지원으로 따로 사용자가 드라이버를 다운받아 설치하실 필요가 없습니다.

전원 연결 순서가 중요하오니 반드시 다음 순서를 지켜주십시오.

1. C studio 실행
2. 프로젝트 열기를 수행하여 원하는 프로젝트 파일을 불러옵니다.  
or 본 사용 설명서의 **4. Tutorial** 을 참고하여 원하는 프로젝트를 생성합니다.
3. ISP 와 컴퓨터 USB 를 연결하여 ISP 의 **Power LED(RED)** 확인  
(제어판의 장치관리자에 휴먼인터페이스(Sonix)로 장치 인식 여부 확인)
4. SN8F27E00 **테스트 보드**에 **전원 공급**(순서중요) Run/Stop LED 확인
5. ISP 의 Run/Stop 의 녹색/ 주황은 바뀔 수 있습니다.
6. C studio 의 메뉴바 -> Build -> Rebuild Project(Shift+F7) 을 한번 수행해 주신 후
7. C studio 의 메뉴바 -> Debug -> Begin Debug(F5) 를 실행하시면 다운로드가 됩니다.



### 4. TUTORIAL

지금까지 **(주)디웰전자** 에서 제공하는 SN8F27E00 Series Test Board 구성과 간단한 사용에 대해서 기술하였지만 처음 하드웨어를 접하는 사용자는 쉽게 이해 할 수 없을 수도 있습니다. 따라서 Tutorial 에서는 간단한 예제들을 통해서 프로그램 컴파일, 다운로드, 디버깅 관련의 과정을 살펴 보도록 하겠습니다.

#### 4-1. 프로그램 소스 작성 및 컴파일

프로그램 작성에 앞서서 개발 환경을 갖추어야 할 것입니다. PC 상에서 설치 해야 할 프로그램에 대해 설명하겠습니다.

##### 4-1-1. 개발 프로그램 다운로드(C Studio)

<http://www.diwell.com> → Support → C studio 항목의 C Studio를 다운로드 후에 압축을 해제 한 후 인스톨 파일을 실행하여 설치합니다.



Fig 4.1. C studio download

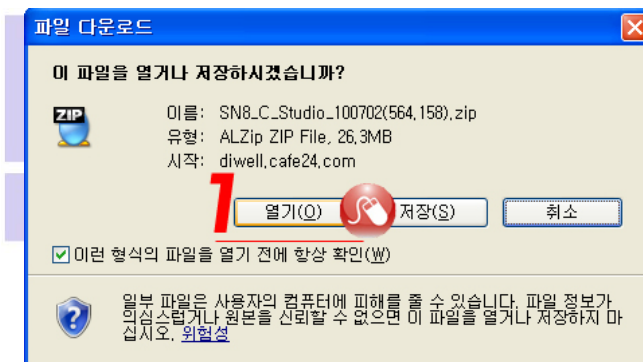


Fig 4.2. download a file



Fig 4.3. Setup C studio

#### 4-1-2. C Studio 실행



Fig 4.4. Click SN8 C Studio

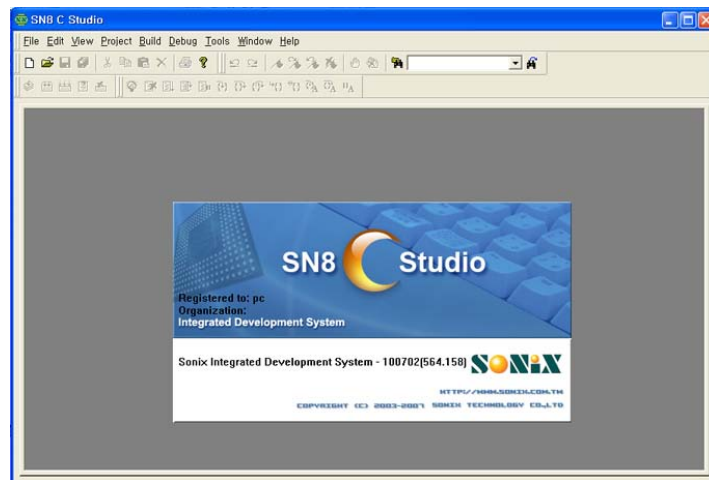


Fig 4.5. C Studio 실행 모습

#### 4-1-3. 새 프로젝트/ 워크스페이스 생성

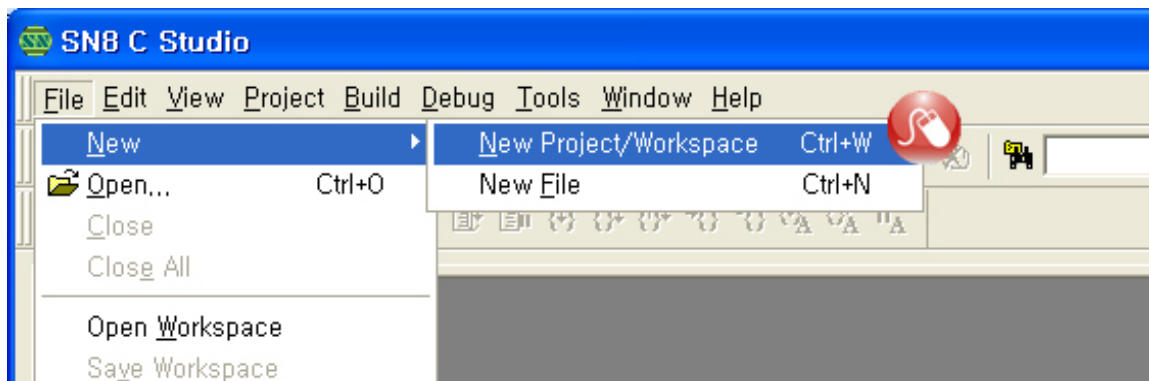


Fig 4.6. File → New → New Project/ Workspace



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

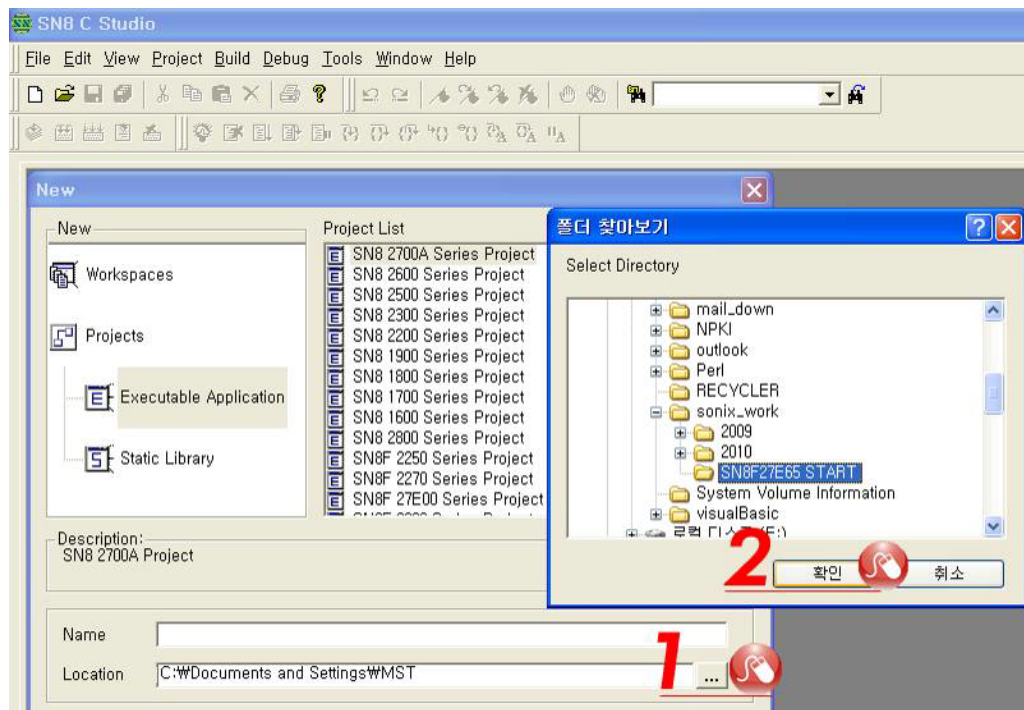


Fig 4.7. 프로젝트가 저장될 Location 지정

Name 작성 후 SN8F 27E00 Series Project 를 선택합니다.

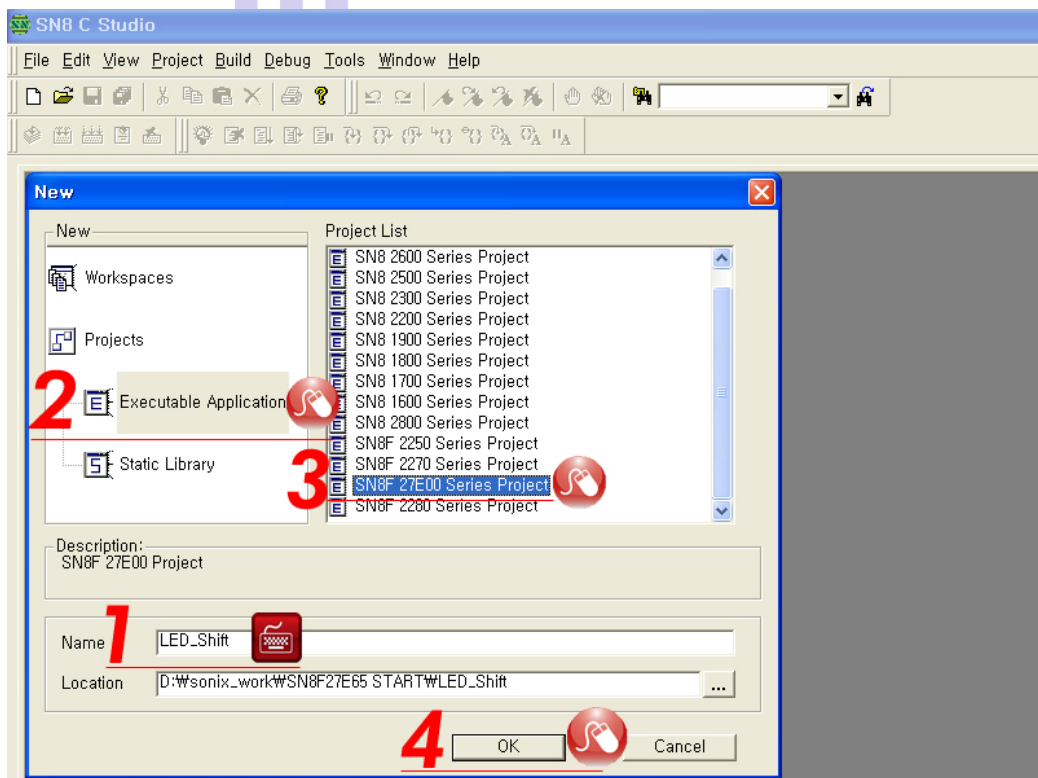


Fig 4.8. Project Description

Chip → MCU 를 선택 후 Code Option 으로 넘어갑니다.

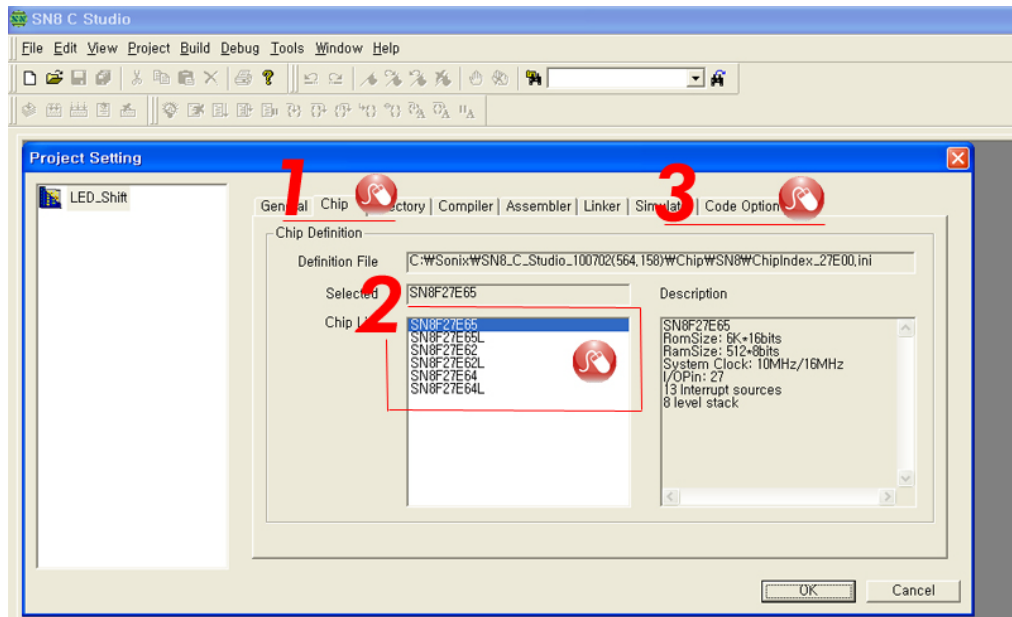


Fig 4.9. Chip Definition

코드옵션 Watch\_Dog = Disable , High\_Clk : IHRC\_16M 로 설정하십시오  
크리스탈을 보드에 별도로 추가하실 경우 High\_Clk 설정을 바꿔 주시면 됩니다.

### ※ SN8F27E62,64 의 LVD 설정법

LVD 항목은 사용할 전압에 따른 설정을 하면 됩니다.

LVD\_L : VDD < 1.8V 일 경우 시스템 리셋(Disable LVD24, LVD33)

LVD\_M : VDD < 1.8V 일 경우 시스템 리셋(Enable LVD24, Disable LVD33)

VDD > 2.4V 일 경우 LVD24 레지스터 값 : 0

VDD < 2.4V 일 경우 LVD24 레지스터 값 : 1

LVD\_H : VDD < 2.4V 일 경우 시스템 리셋(Enable LVD24, LVD33)

VDD > 2.4V 일 경우 LVD24 레지스터 값 : 0

VDD < 2.4V 일 경우 LVD24 레지스터 값 : 1

VDD > 3.3V 일 경우 LVD33 레지스터 값 : 0

VDD < 3.3V 일 경우 LVD33 레지스터 값 : 1

LVD\_MAX : VDD < 3.3V 일 경우 시스템 리셋

LVD (SN8F27E62,64)	LVD Code Option		
	LVD_L	LVD_M	LVD_H
1.8V Reset	가 능	가 능	가 능
2.4V Flag	-	가 능	-
2.4V Reset	-	-	가 능
3.3V Flag	-	-	가 능

### ※ SN8F27E93 의 LVD 설정법

LVD\_L : VDD < 1.8V 일 경우 시스템 리셋(Disable LVD23,LVD30)

LVD\_M : VDD < 1.8V 일 경우 시스템 리셋(Enable LVD23, Disable LVD30)

VDD > 2.3V 일 경우 LVD23 레지스터 값 : 0

VDD < 2.3V 일 경우 LVD23 레지스터 값 : 1

LVD\_H : VDD < 2.3V 일 경우 시스템 리셋(Enable LVD23, LVD30)

VDD > 2.3V 일 경우 LVD23 레지스터 값 : 0

VDD < 2.3V 일 경우 LVD23 레지스터 값 : 1

VDD > 3.0V 일 경우 LVD30 레지스터 값 : 0

VDD < 3.0V 일 경우 LVD30 레지스터 값 : 1

LVD\_MAX : VDD < 3.0V 일 경우 시스템 리셋

LVD (SN8F27E93)	LVD Code Option		
	LVD_L	LVD_M	LVD_H
1.8V Reset	가 능	가 능	가 능
2.3V Flag	-	가 능	-
2.3V Reset	-	-	가 능
3.0V Flag	-	-	가 능

※ 건전지를 이용한 전원을 사용할 시에 단순히 LVD 레지스터의 상태확인만으로 저전압 체크가 가능합니다.(Easy Low battery detector)  
단, 참고용으로 쓰셔야 하며, 정확한 VDD 측정용으로 사용하진 마십시오.

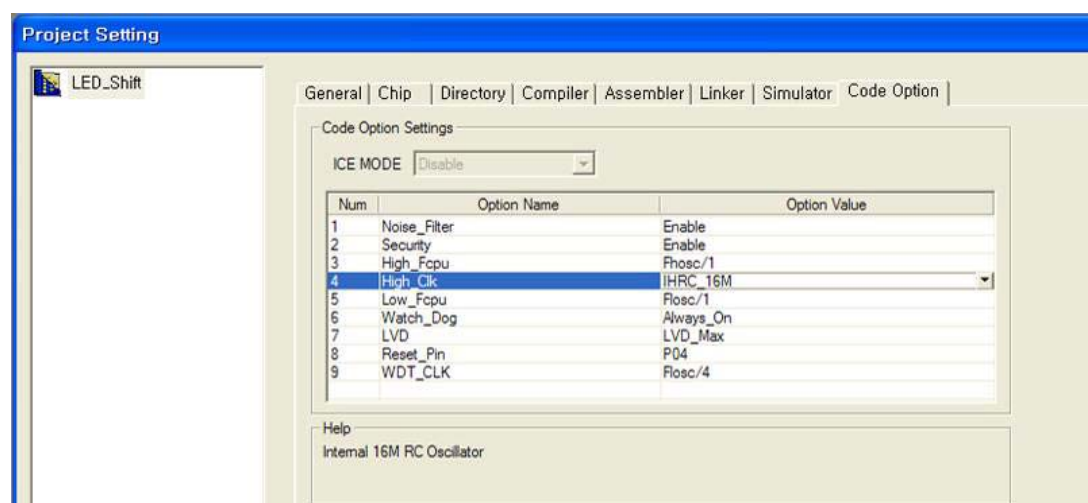


Fig 4.10. Code Option Settings

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	Fhosc/1	Slow mode instruction cycle is 1 low speed oscillator clocks.
	Fhosc/2	Slow mode instruction cycle is 2 low speed oscillator clocks.
	Fhosc/4	Slow mode instruction cycle is 4 low speed oscillator clocks.
	Fhosc/8	Slow mode instruction cycle is 8 low speed oscillator clocks.
Noise_Filter	Enable	Enable Noise Filter.
	Disable	Disable Noise Filter.
WDT_CLK	Fhosc/4	Watchdog timer clock source Fhosc/4.
	Fhosc/8	Watchdog timer clock source Fhosc/8.
	Fhosc/16	Watchdog timer clock source Fhosc/16.
	Fhosc/32	Watchdog timer clock source Fhosc/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.(27E62/64/65)
	P13	Enable P1.3.(27E93)
Security	Enable	Enable ROM code Security function.
	Disable	Disable ROM code Security function.
LVD (27E62/64/65)	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
LVD (27E93)	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 2.3V low voltage indicator.
	LVD_H	LVD will reset chip if VDD is below 2.3V Enable LVD33 bit of PFLAG register for 3.0V low voltage indicator.
	LVD_MAX	LVD will reset chip if VDD is below 3.0V

워크스페이스 설정이 끝났으므로 이제 C 파일을 만들면 됩니다.

#### 4-1-4. C 파일 생성

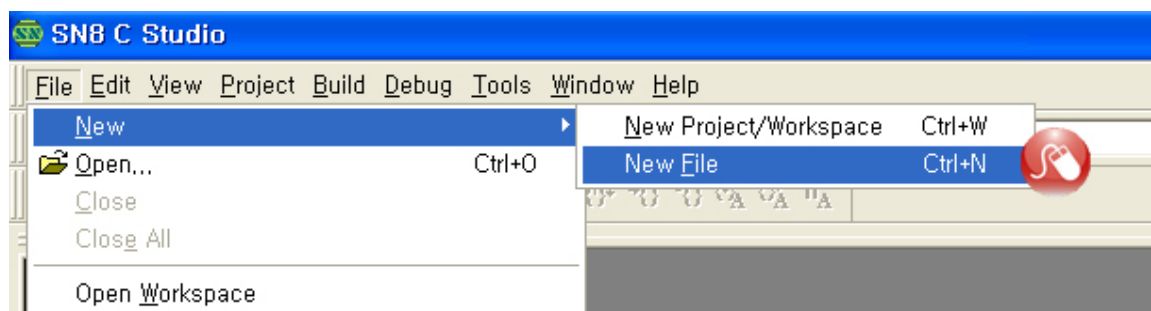


Fig 4.11. New File

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

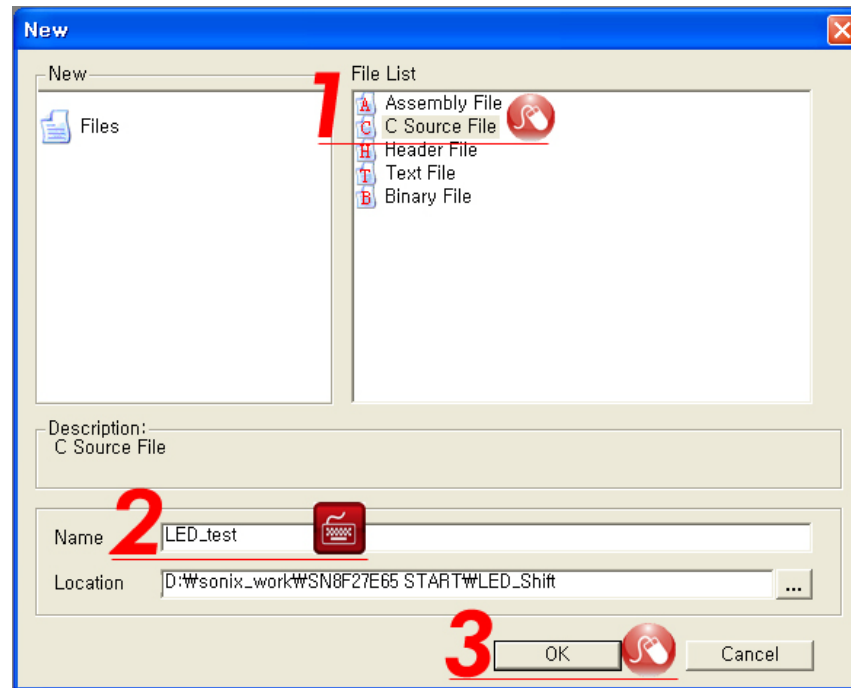


Fig 4.12. C File 생성

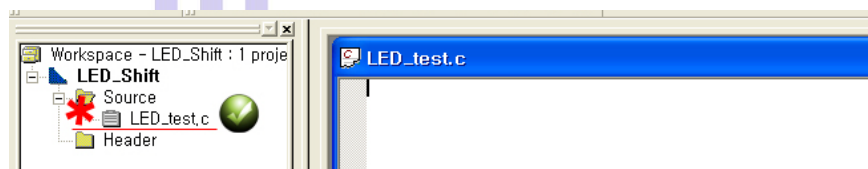


Fig 4.13. 파일 생성 완료

#### 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.

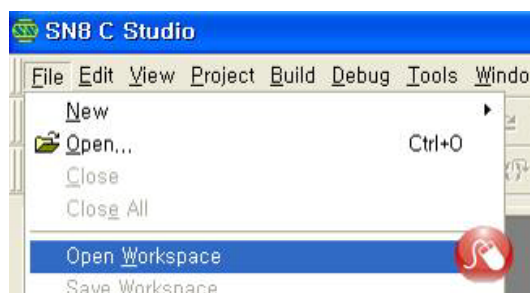


Fig 4.14. Open Workspace

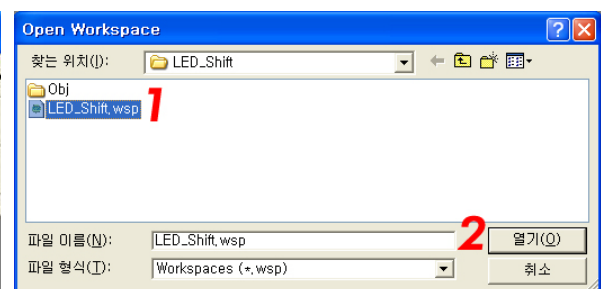


Fig 4.15. Open wsp File

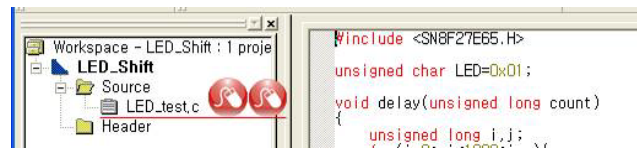


Fig 4.16. C File 더블클릭

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.

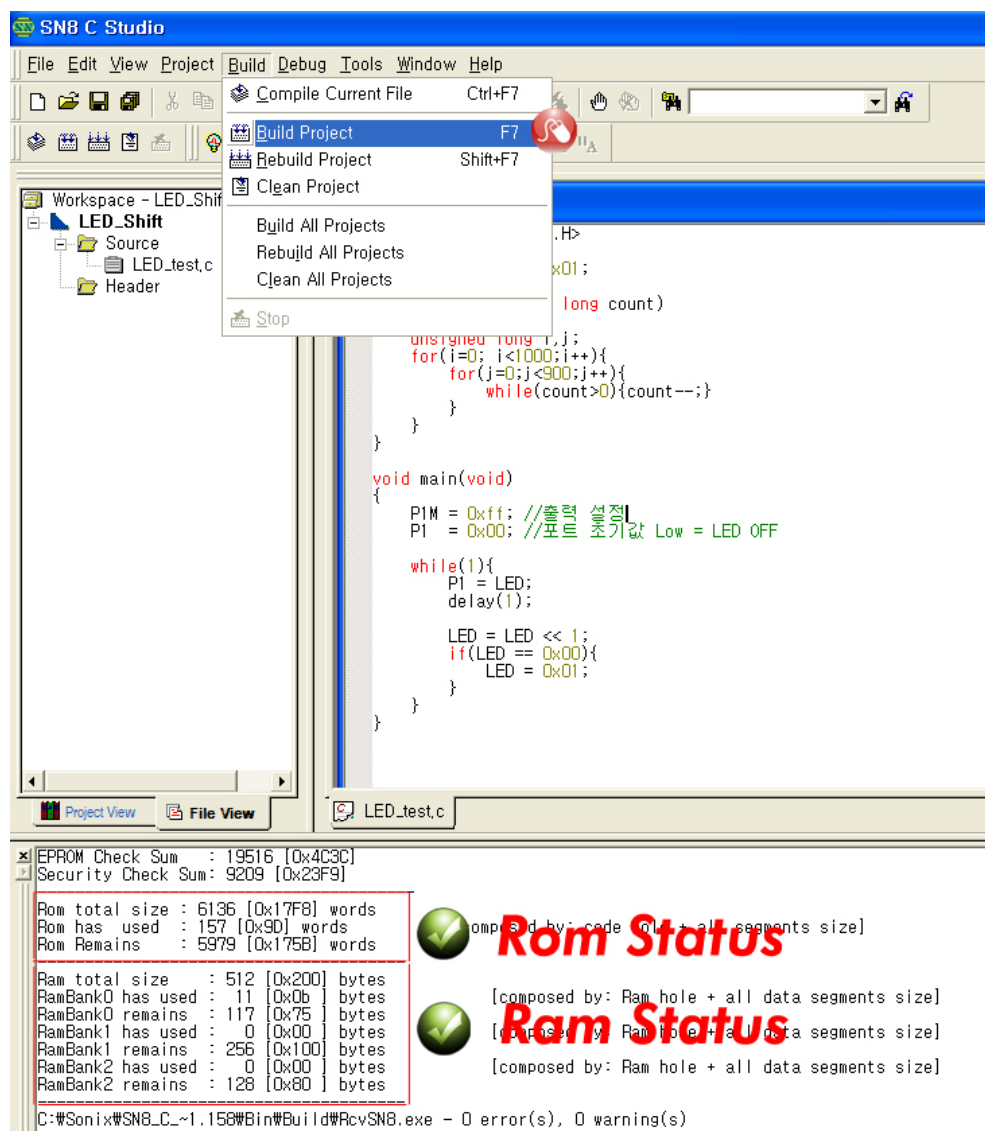


Fig 4.17. 컴파일



## 4-2. ISP 를 통한 다운로드

Fig 3.1. 의 그림과 같이 본 SN8F27E00 Series Test Board 와 ISP 가 정상 연결이 되면 ISP 의 Run/Stop 의 LED 에 주황색(or 녹색)이 켜집니다.

※ Run/Stop LED 색은 바뀔 수 있습니다.

### 4-2-1. Running & Debugging

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

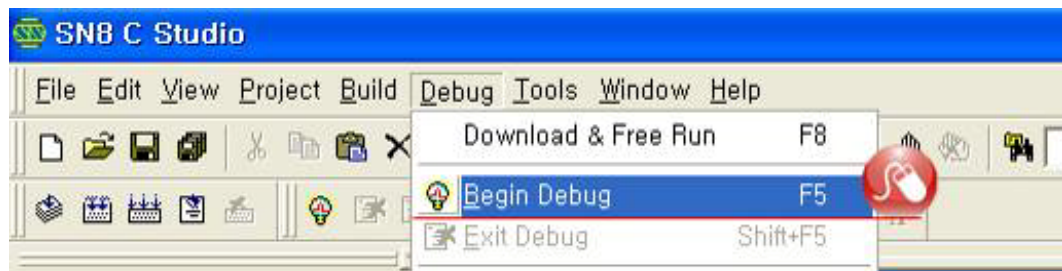


Fig 4.18. ISP Download & Run

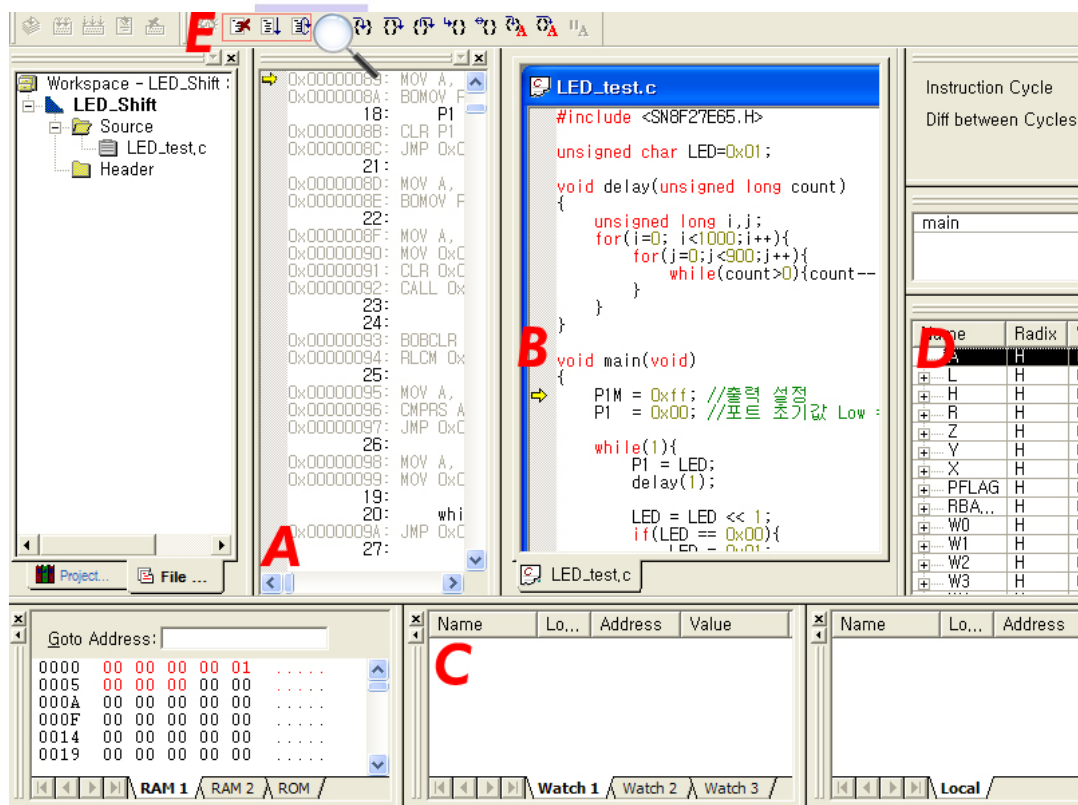


Fig 4.19. Debugging Mode

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.

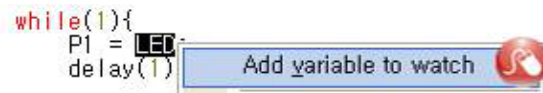


Fig 4.20. Add variable

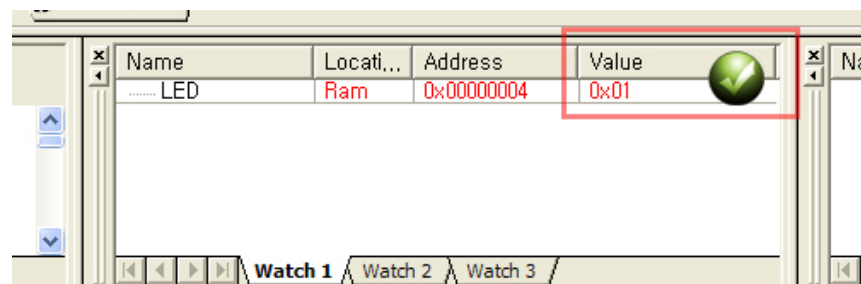


Fig 4.21. Watch Window

**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.

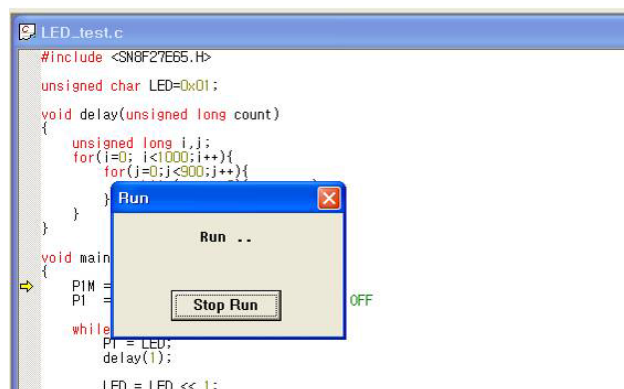


Fig 4.22. Running

#### 4 - 3 . ISP pin description



Fig 4.23. ISP

A : USB Port

B : Power / Link LED( ● )

C : 디버그 모드 LED( ● / ● )

D : 10pin 케이블 커넥터

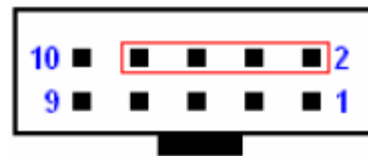


Fig 4.24. Pin assignment

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

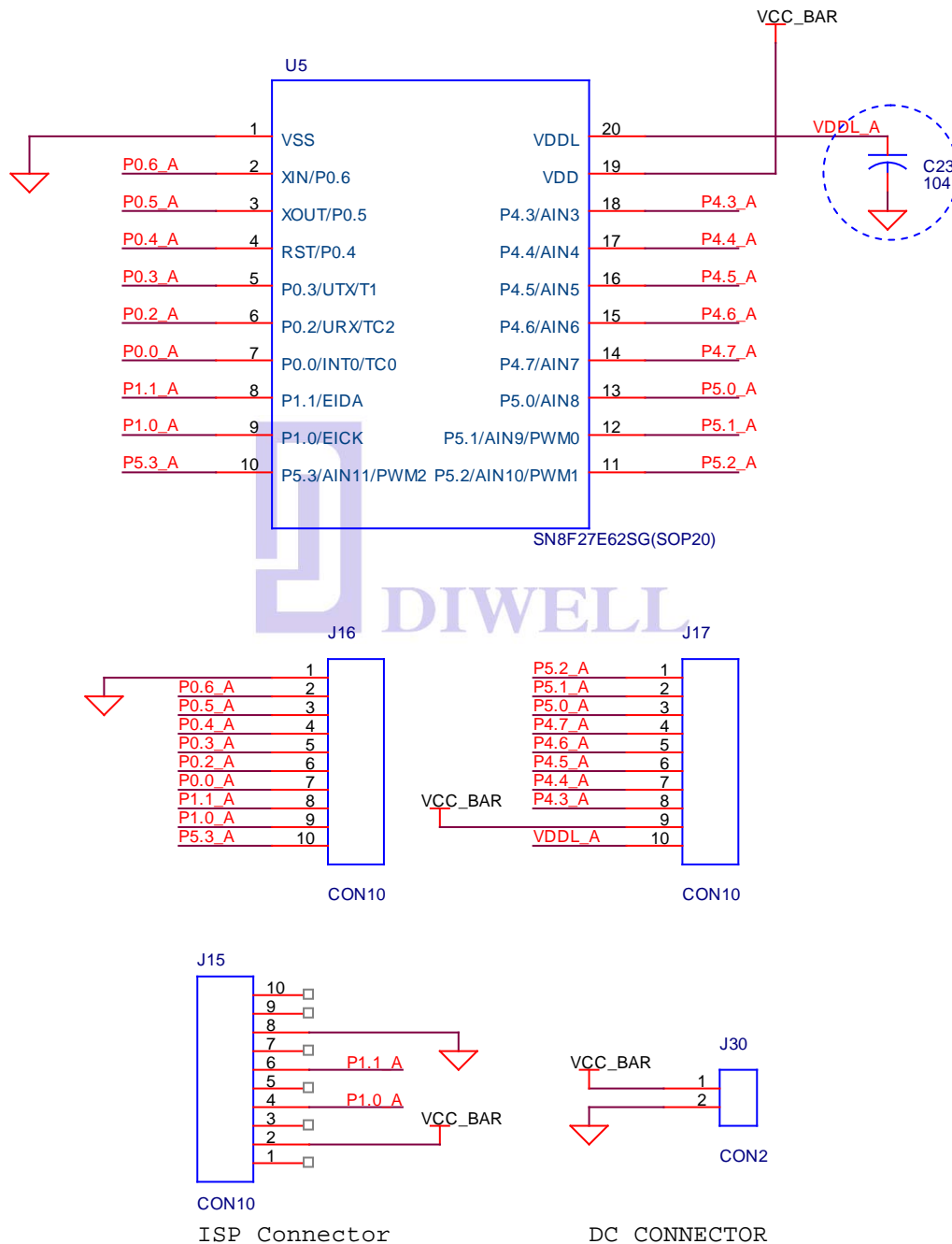
Table 4.2. ISP LED status information

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

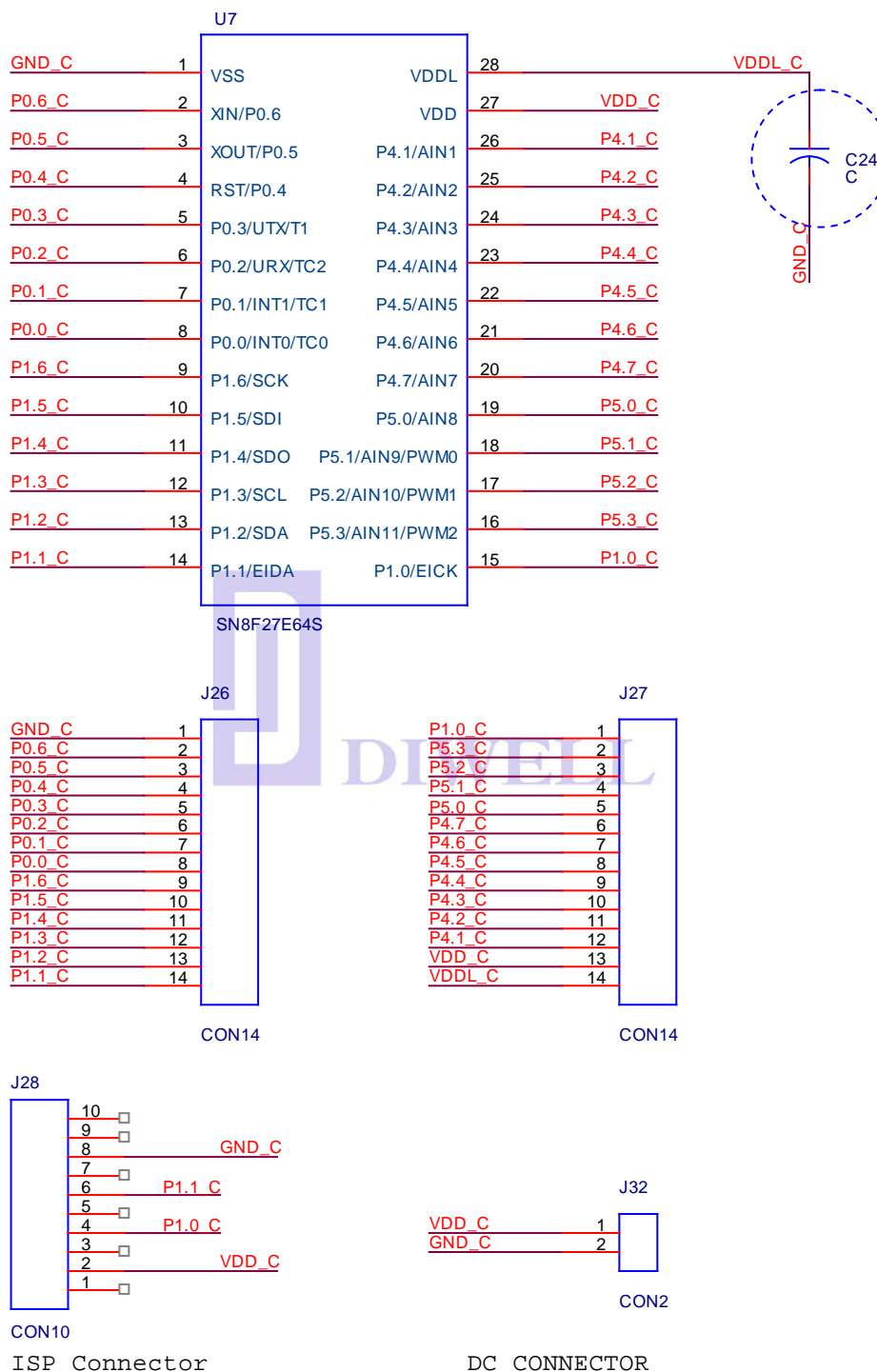
### 5-1. SN8F27E62

※ VDDL(20) 은 반드시 캐패시터(104)를 통하여 그라운드와 연결해주십시오.  
테스트보드에는 연결되어 있습니다.



## 5 - 2 . SN8F27E64

※ VDDL(20) 은 반드시 캐패시터(104)를 통하여 그라운드와 연결해주십시오.  
테스트보드에는 연결되어 있습니다.



### 5 - 3 . SN8F27E93

- ※ VDD 와 VSS 간 캐패시터(104)를 연결해주십시오.
- ※ VDD25 - REGO 는 반드시 연결하고, 캐패시터(104)를 VSS 와 연결하십시오  
테스트보드에는 연결되어 있습니다. 회로도 참고

