

eclipse

```

int (*readpages)(struct file *filp, struct address_space *mapping,
                 struct list_head *pages, unsigned nr_pages);
int (*prepare_write)(struct file *, struct page *, unsigned, unsigned);
int (*commit_write)(struct file *, struct page *, unsigned, unsigned);
sector_t (*bmap)(struct address_space *, sector_t);
int (*invalidatepage)(struct page *, unsigned long);
int (*releasepage)(struct page *, int);
ssize_t (*direct_IO)(int, struct kiocb *, const struct iovec *iov,
                     loff_t offset, unsigned long nr_segs);

```

Eclipse는 IBM에서 시작하여 오픈소스로 개발이 진행되고 있는 대규모 통합 개발 환경이다. 이 소프트웨어는 본래 JAVA용으로 개발되었지만, 강력한 유연성을 가지고 있기 때문에 플러그 인을 추가하는 것만으로 C/C++, PHP, Python, Ruby와 같은 많은 언어와 환경에 대응할 수 있다. 또한, CVS나 Subversion의 소스 버전 관리기능을 기본적으로 갖추고 있기 때문에 오픈 소스 커뮤니티에서 넓게 이용되고 있다.

김민수 / minsu.kim@gmail.com

Eclipse를 임베디드 소프트웨어 개발에 사용하면 여러가지 작업이 매우 편리해진다. 버전관리는 당연히 사용할 수 있고, 에디터는 문법이 강조되어 컬러링되고, 파일을 저장하면 빌드가 시작되어 에러가 일어난 행에 언더라인이 표시되므로 문제를 쉽게 알 수 있다. 본래 JAVA 언어용 환경은 임베디드시스템과 거리가 있어 사용하기에 조금 무거운 소프트웨어이기도 하다. 하지만, 커스터마이징을 통해 조금 더 최적화를 이룬다면 훌륭한 임베디드 개발 환경이 될 수 있다.

이번 연재에서는 임베디드 개발 환경으로 만들기 위한 Eclipse의 설정을 설명한다. 윈도우 환경에서의 기준이며, 툴 체인에는 GNU-ARM 컴파일러를 통해 gdb를 사용하여 JTAG경로로 디버그 할 수 있다.

필요한 소프트웨어

아래의 소프트웨어가 필요하게 된다.

- JAVA런타임 - <http://java.sun.com>
- Eclipse - <http://www.eclipse.org>
- CDT - <http://www.zylin.com/embeddedcdt.html>

Eclipse Setup



/필/자/소/개/

현재 회사에서 OS 커널과 시스템 프로그래밍에 대한 개발을 맡고 있다. 여러 가지 최신기술에 대한 관심과 새로운 시도를 해보는 것이 취미이다. 구글과 같은 기업 문화를 좋아하며 선진적 개발환경 인프라를 배우려고 노력하고 있다.

임베디드 개발환경으로써 이클립스의 활용

1. Eclipse Setup
2. Embedded CDT
3. Debugging

Cygwin - <http://www.cygwin.com>
 GNUARM - <http://www.gnuarm.com>

JAVA 런타임은 Eclipse를 실행하기 위해 필요하며 Sun의 사이트로부터 다운로드할 수 있다. 여기서는 J2SE 5.0 Update 6을 사용하고 있다. 최신이어도 크게 문제는 없지만 Eclipse가 요구하는 버전을 사용하면 좋을 것이다. J2SE는 JRE라는 개발 툴을 포함한다. JDK는 JAVA를 개발하지 않고 임베디드 개발환경으로써 C/C++ 크로스 컴파일러를 사용하므로 JRE만 가지고도 구동에 문제가 없을 것이다.

Eclipse는 통합 개발 환경의 본체이며 CDT는 이번 개발 환경 구축에 대해 핵심이 되는 소프트웨어이다. 이것은 Eclipse의 플러그-인으로 Eclipse에 C언어 지원 기능을 추가한다. 하지만 현재 표준의 CDT는 임베디드 디버그에 대응하고 있지

않으므로 표준CDT에 패치를 적용한 Zylind CDT를 사용한다. Cygwin은 gnu 툴 체인을 실행하기 위해 필요하며, 최신판도 버전에 따른 문제는 없을 것이다. GNU-ARM은 ARM 프로세서용 gcc, binutil, gdb등을 정리한 컬렉션이다. 여기서는 3.4.3을 사용하고 있지만, 특별한 이유는 없다.

인스톨의 순서

인스톨의 순서는

- JAVA런타임
- Eclipse
- CDT
- Cygwin
- GNUARM

의 순서로 한다. 1~3 과 4~6은 서로 의존하지 않기 때문에 차례를 뒤집어도 괜찮다. 1,4,5,6은 윈도우상의 다른 소프트웨어와 같은 인스톨러를 사용하고 있기 때문에 특히 주의할 점은 없다. Eclipse는 윈도우 환경에서 단순한 Zip 파일로 배포되고 있으며 인스톨 필요 없이 압축을 해제하면 Eclipse폴더를 만들어 그 아래에 필요한 서브 폴더를 전개한다.

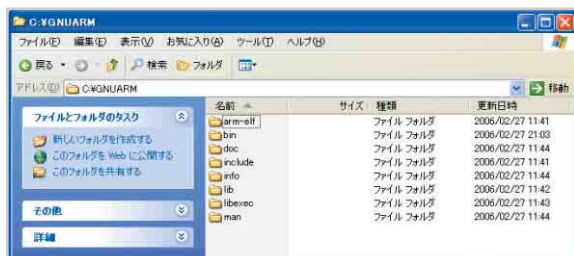
CDT도 zip파일이며 Eclipse 폴더에 해제하면 Plugins 서브 폴더와 Features의 아래에 필요한 파일을 해제한다. 인스톨이 끝나면 Cygwin의 Bash를 가동해 Arm-Elf-gcc.exe가 실행되고 있는지 패스를 확인하면 끝난다.

인스톨

- Eclipse
c:\eclipse에 아래와 같이 해제한다.



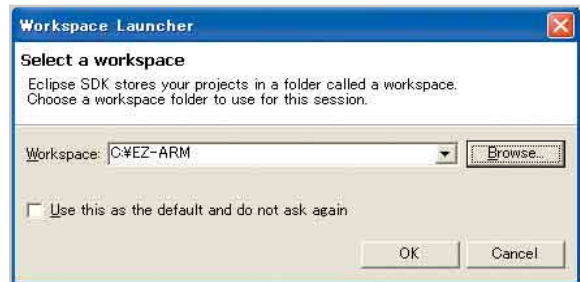
- CDT
CDT도 위와 같이 해제한다.
- GNUARM
c:\GNUARM에 해제한다.



프로젝트 설정

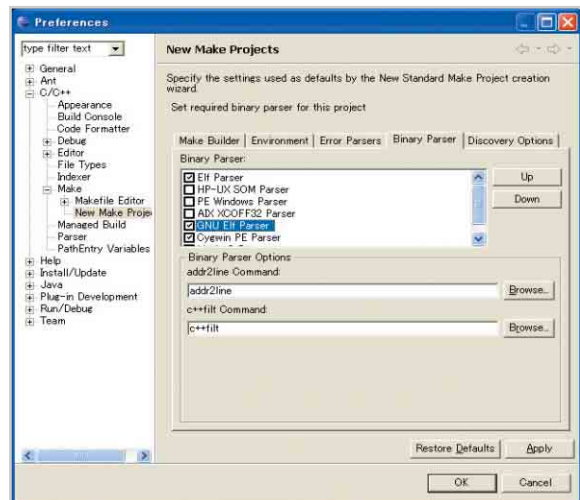
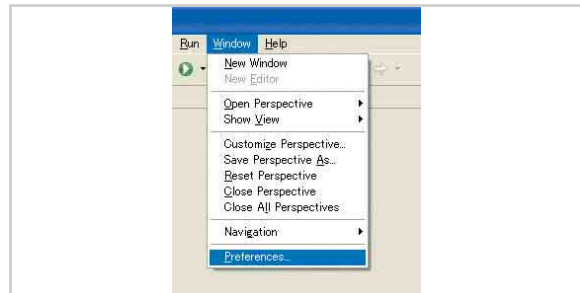
프로그램 개발은 프로젝트 단위로 행해진다.

- Eclipse의 실행



이번 설정에서는 각종 프로젝트 파일을 c:\EZ-ARM 에 두기 때문에, 그 디렉토리를 지정한다.

- 바이너리 파서의 설정

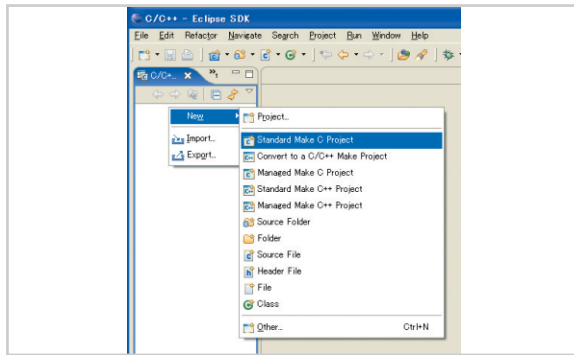


이번에는 타깃보드에서 지원되는 바이너리를 체크한다.

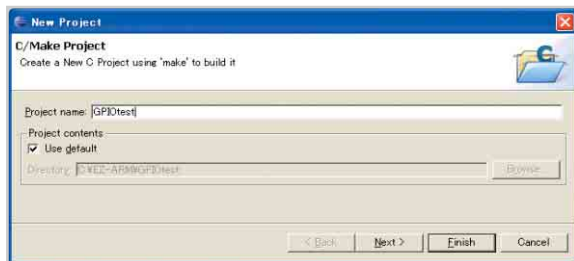
• 신규 프로젝트의 작성

프로그램을 작성하는 경우 아래의 순서대로 실시한다.

1. 좌측의 윈도우로 우클릭 New -> Standard Make C Project

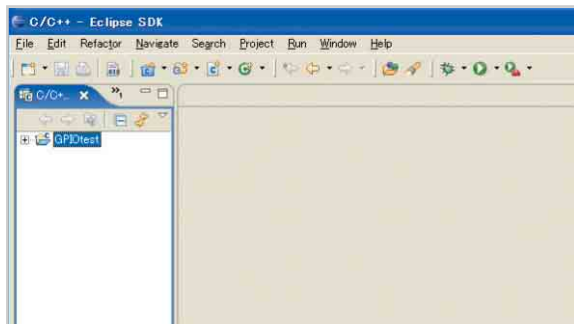


2. 프로젝트명 입력



프로젝트명을 입력한다. 이번에는 GPIOtest를 입력하고, Finish를 눌러 종료한다.

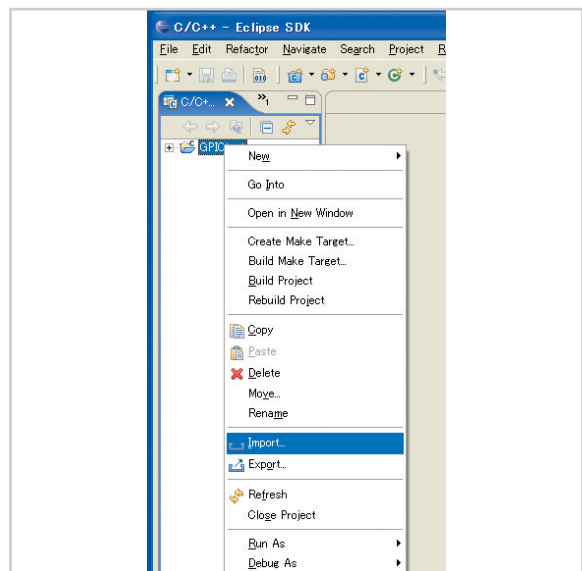
3. 확인



여기까지의 작업으로 그림과 같이 프로젝트명이 설정되었다.

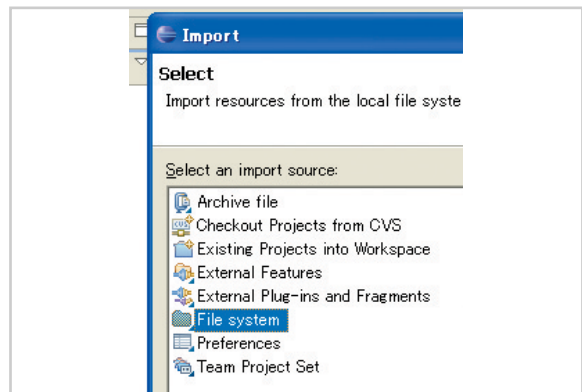
기존 프로젝트로부터의 импорт

• Import



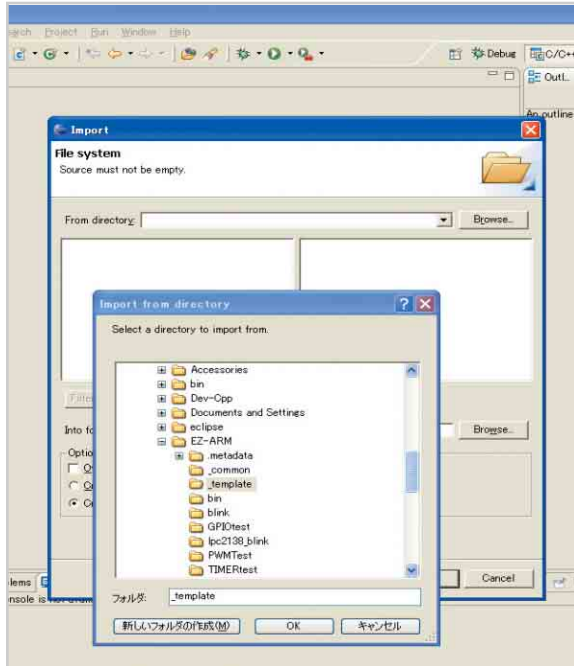
먼저 설정한, C:\EZ-ARM\template로부터 스켈리턴 프로그램을 импорт한다.

• File system 선택

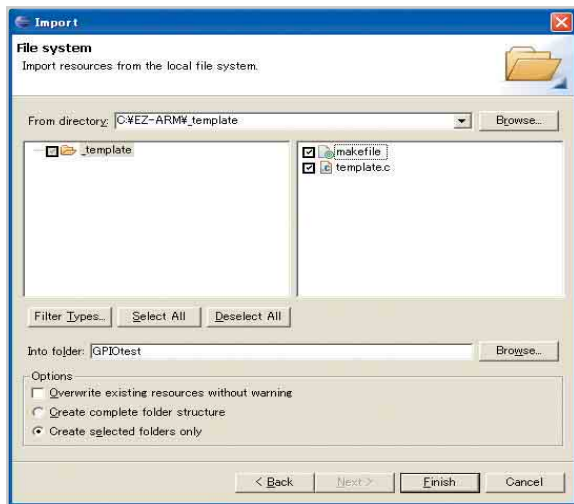


파일 시스템을 선택한다.

- 임포트 디렉토리 선택



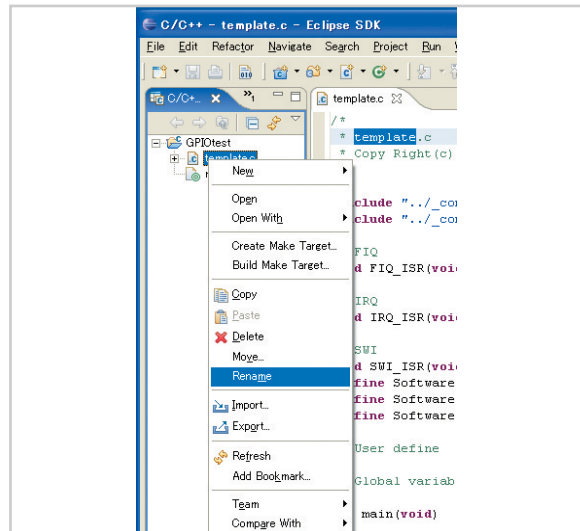
- 임포트 하는 파일의 선택



임포트 하는 파일을 선택한다.

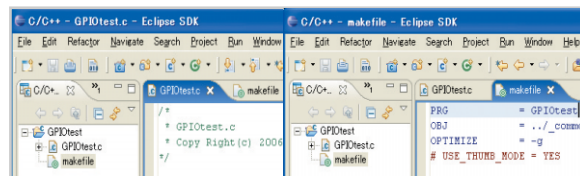
- template.c
- makefile

- Rename



파일명을 template.c -> GPIOtest.c로 변경한다.

- 파일의 수정



파일중의 template를 GPIOtest에 각각 수정한다.

마치며

이번 연재에서는 간단하게 설치와 설정에 관련된 사항만을 알아보았다. 위에서 보듯 PC에서 쓰는 Eclipse 개발환경과 같은 툴이지만 임베디드 개발을 위해서는 추가적으로 설정해야 되는 부분이 있고 차이점도 있다. 다음회부터는 Eclipse CDT 환경의 사용법에 대한 내용을 다루도록 하겠다. ^{RealTime}