

ACCOMPLISH A PROJECT USING ECLIPSE

(for embedded controller compiler)

Author : Dopesoul (Alex Lim)

Department : Robot Research at Inha Univ.

E-mail : nowready@naver.com

1. INTRODUCTION INTO ECLIPSE

이클립스는 IDE(Integrated Development Enviroments) 플랫폼으로써 독립적인 개발환경이 아닌 통일된 개발 환경에서 다른 언어들을 이용할수 있게 하는 프레임워크(Frame Work)이다.

기존의 IDE 는 특정 타겟과 특정 컴파일러를 탑재하여 하나의 언어로만 작업하게 되어있고 Syntax Highlighting, Code parser, Template 등을 제한된 범위에서만 수정 가능하였다.

그래서 개발자들은 주로 프로젝트는 IDE 틀에서 등록해놓고 Source code 의 수정은 자신이 익숙한 에디터(Vi, Emacs, Editplus, Source Insight, Notepad)로 하는 것이 보통이었다. 이렇게 작업하게 되면 두개의 어플리케이션을 번갈아가며 실행하여야 하며, 소스코드의 일관성도 떨어지게 되는 것이 보통이다. 또한 컴파일러나 링커 차원에서 얻을수 있는 정보들 예를들면 바이너리 파일의 링커구조, 컴파일된 오브젝트 파일간의 관계, 빌드옵션들을 보려면 복잡한 절차를 거쳐서 보아야 하기 때문에 IDE 의 본연의 뜻에 미치지 못하게 된다.

디버거도 원하는 형식으로의 Parsing 하기란 사실상 불가능하여 자신이 틀에 익숙해 져야 하는 것이 보통이었다. 하지만 Eclipse 라는 프로젝트의 시작으로 이와 같은 불편함들은 해소되기 시작했고, 현재 3.0.1(2005 년 1월 현재) 버전까지 출시되어 점점더 완벽한 개발 환경을 지원하고 있다.

이클립스의 장점으로는 유연함(원문 : Flexibility)에 있다. 우리는 공개된 GNU 컴파일러를 상용툴보다 뛰어난 Editor 와 Views 를 갖춘 Eclipse 와 연동시킬 수 있다. 이는 Opensource 개발자들에게 반가운 소식이었으며, 비단 Opensource 뿐만이 아닌 상용 제품도 회사차원에서 스폰서를 하는 방식으로 Eclipse 를 차용하고 있다.

빌드시 나오는 메시지를 원하는 방식으로 Parsing 하고 개별 디버거들과 통합이 가능하다. 기존의 GNU 환경에서 가장 부족했던점은 통합개발이다. GNU 툴로 이루어진 Compiler + Debugger + Editor 가 통합된 환경은 찾아보기 힘들다. Eclipse 가 이러한 모자란 부분을 채워줌으로써 Opensource 개발툴들은 더욱 신뢰와 인기를 얻게 되었다.

2. INSTALLATION

이클립스는 다음의 Spec 을 만족해야 원활하게 이용할 수 있다.

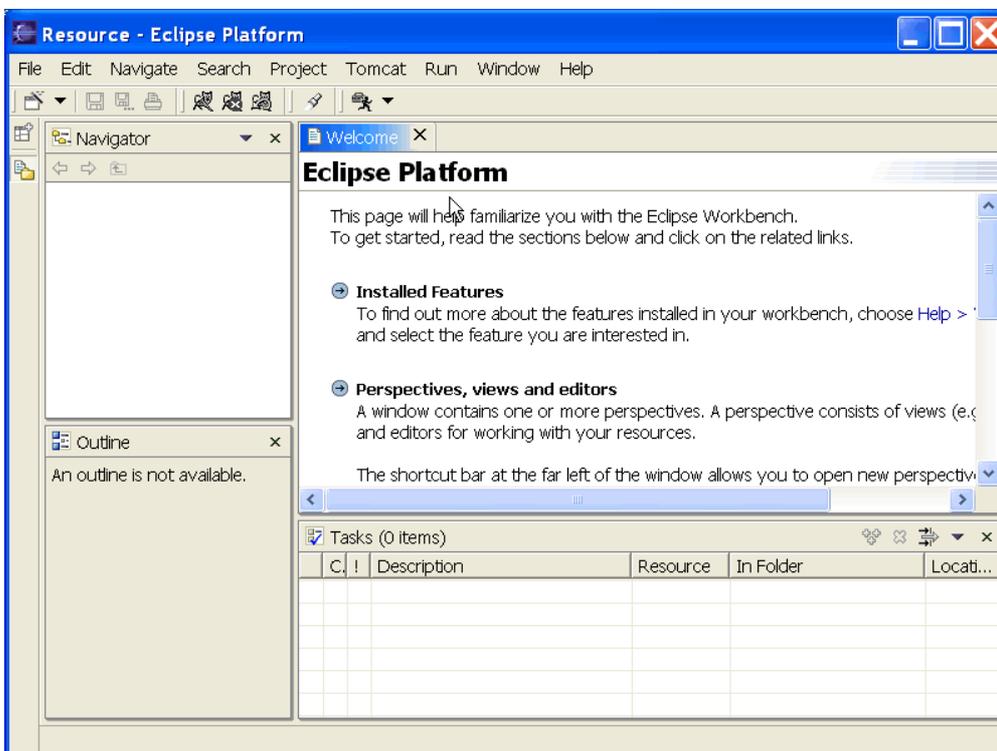
- Windows, Linux, Solaris, QNX or MAC OS/X operating system.
- A Java 2 runtime environment(JRE) or Java 2 Software Development kit(J2SDK)

이클립스는 JAVA 로 제작되어 Java 2 Runtime Environment 를 필요로 한다.

설치절차는 다음과 같다.

1. JRE 혹은 J2SDK 를 인스톨한다.
2. Eclipse 의 원하는 버전의 압축파일을 받은 후, 적절한 곳에 압축을 해제한다.
3. eclipse.exe 를 실행시킨다.

따로 Install 을 요구하지 않는다.



<인스톨후 볼수 있는 화면>

3. C/C++ PROJECT IN ECLIPSE

Eclipse 의 기본 환경은 Java 에 초점이 맞추어져 있다. 그러나 이것은 Eclipse 가 오직 Java 만의 툴이라는 것을 의미하지는 않는다. Eclipse 는 합리적인 구조로, 사용자마다 다를수 있는 부분들을 Plug-in 형태로 제공하고 있다. 누구나 Eclipse Plug-in 의 개발방법을 알고 또한 지원받을 수 있다. 단 지원의 형태는 프로젝트마다 다를 수 있다는 점은 염두에 두어야 한다.

Eclipse 에서 C/C++ 프로젝트는 CDT 라는 Plug-in 으로 수행 가능하다. CDT 는 기존의 프로그래머들이 익숙한 Syntax Highlighting 을 제공하며, XML 을 이용하여 여러가지 환경 설정을 자신에게 최적화 할 수 있다. 그러나 CDT 에서 기본으로 제공하는 기능만 가지고도 기존의 IDE 보다 나은 환경을 가질 수 있다.

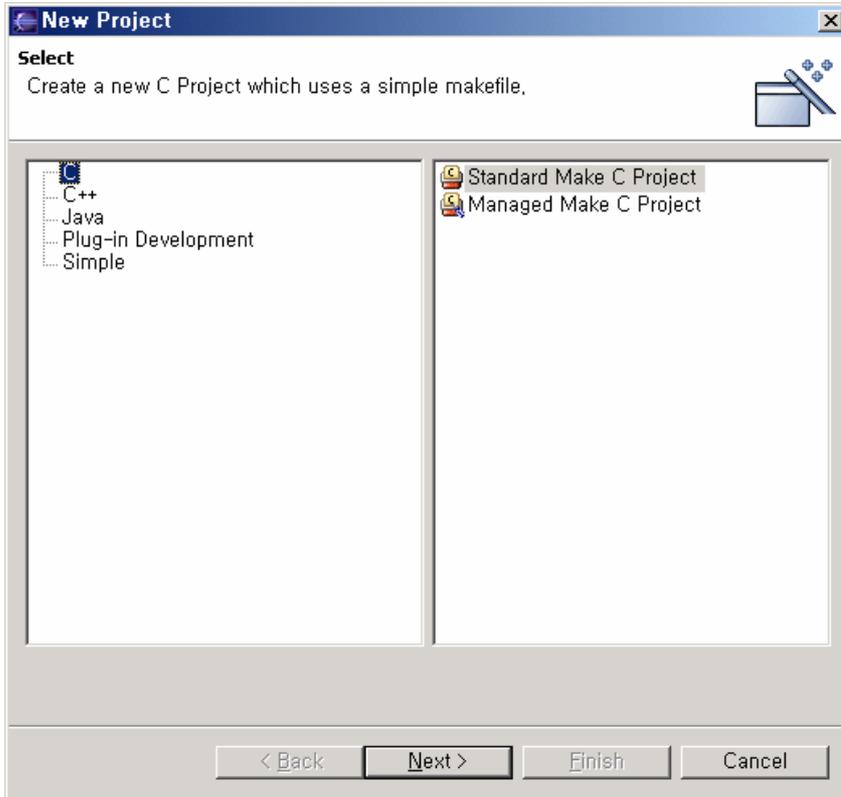
CDT 는 다음의 Spec 을 갖추고 있다.

- C/C++ Editor (basic functionality, syntax highlighting, code completion etc.)
- C/C++ Debugger (APIs & Default implementation, using GDB)
- C/C++ Launcher (APIs & Default implementation, launches and external application)
- Parser
- Search Engine
- Content Assist Provider
- Makefile generator

Eclipse 와 CDT 는 Commercial-quality 이상의 기능들을 가지고 있다. 특히, GNU 툴과의 불임성, 유연성은 현존하는 IDE 와는 비교할 수 없다.

4. CDT INSTALLATION

CDT 인스톨도 별도의 절차는 없다. CDT 압축을 Eclipse 가 있는 곳에 해제하는 것으로 인스톨은 끝난다. CDT 를 설치하면 Eclipse 는 Plug-in 을 감지한다. Plug-in 이 제대로 설치되면 File>New>Project 에서 C/C++ Project 생성화면을 볼 수 있다.



<C/C++ 프로젝트 가 보인다>

5. SIMPLE “HELLO WORLD!” PROJECT IN AVR-GCC

본 문서에서 최종적으로 다루고자 하는 GNU 컴파일러와 Eclipse 의 연동을 이번 장에서 다룰 것이다. GNU Toolchain 을 개별적으로 설치하지 않고 Opensource Project 인 WinAVR(whenever 로 발음한다)을 설치하고 Eclipse 로 빌드하는 것으로 마치겠다. Eclipse 의 자세한 설정법은 기존의 문서와 책이 많으므로 참고하기 바란다. (사실 나는 기본 설정으로도 훌륭하게 잘 쓰고 있다)

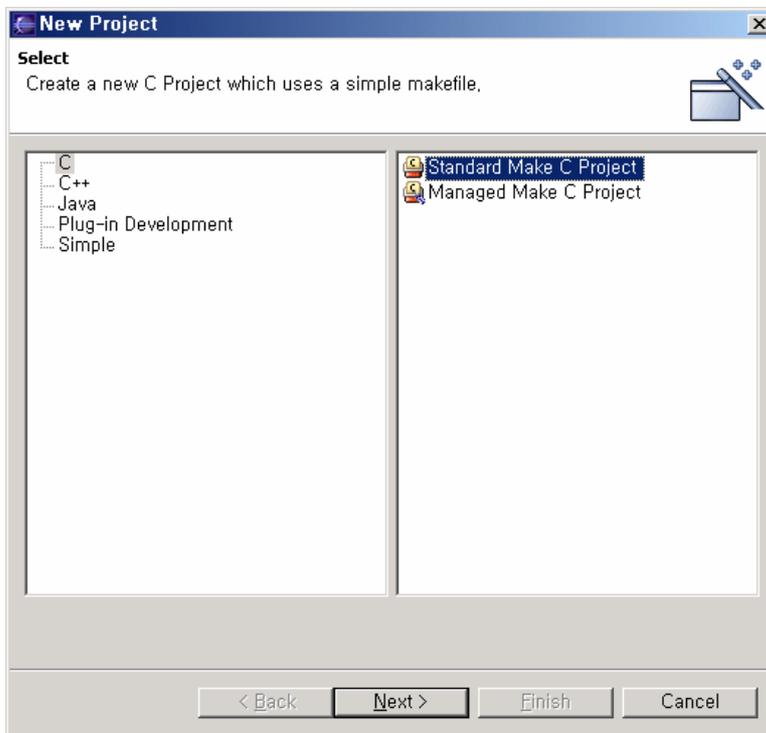
WinAVR 은 Sourforge.net 에서 다운로드 받을 수 있다. WinAVR 은 별도의 설정 없이 인스톨하면 된다. 자동적으로 path 를 잡아주고, toolchain 들을 설치해준다. WinAVR 을 설치한 후 Eclipse 를 띄운다. 만약, MiniGW 등의 x86 용 GCC 컴파일러를 설치하였다면, 같은 명령어의 충돌이 예상되므로, make 를 바꾸거나 프로젝트마다 path 를 달리 함으로써 해결 가능 할 것이다.

- Makefile

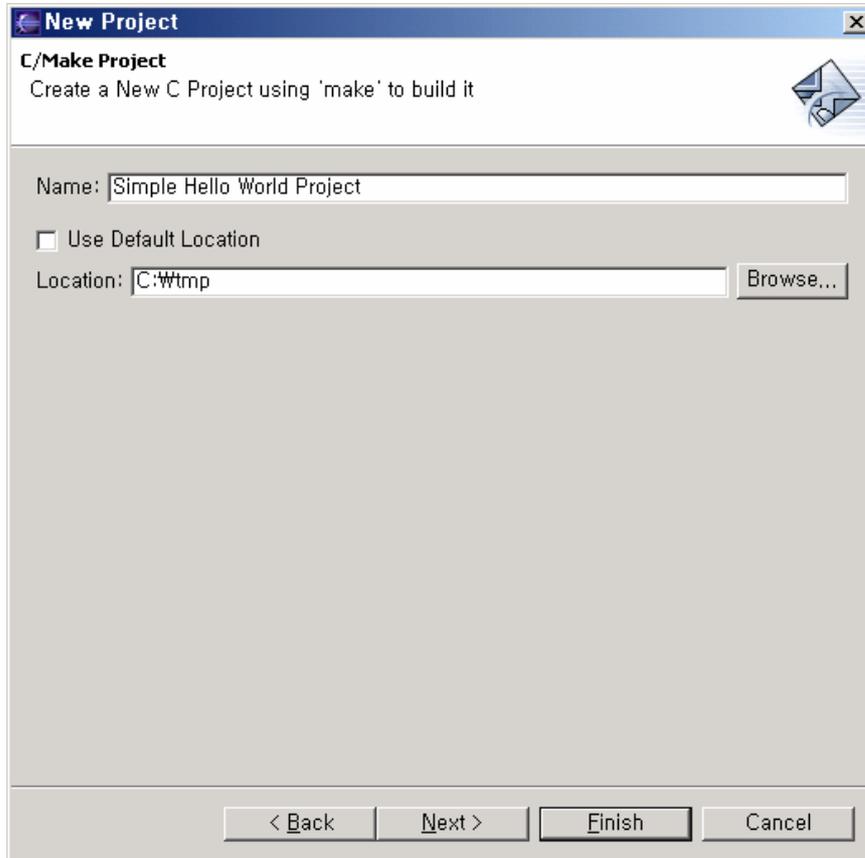
Makefile 은 make 유틸리티가 참조하는 파일로, 빌드 과정을 담당한다. AVR 은 ISP(In System Programmer) 를 지원한다. Writer 로는 Ponyprog 를 이용 할 것이다. Output 은 Binary format 이 아니라 Intel-HEX format 을 이용할 것이다.

Makefile 세팅에 대한 설명은 본 문서의 목적이 아니므로 생략한다.

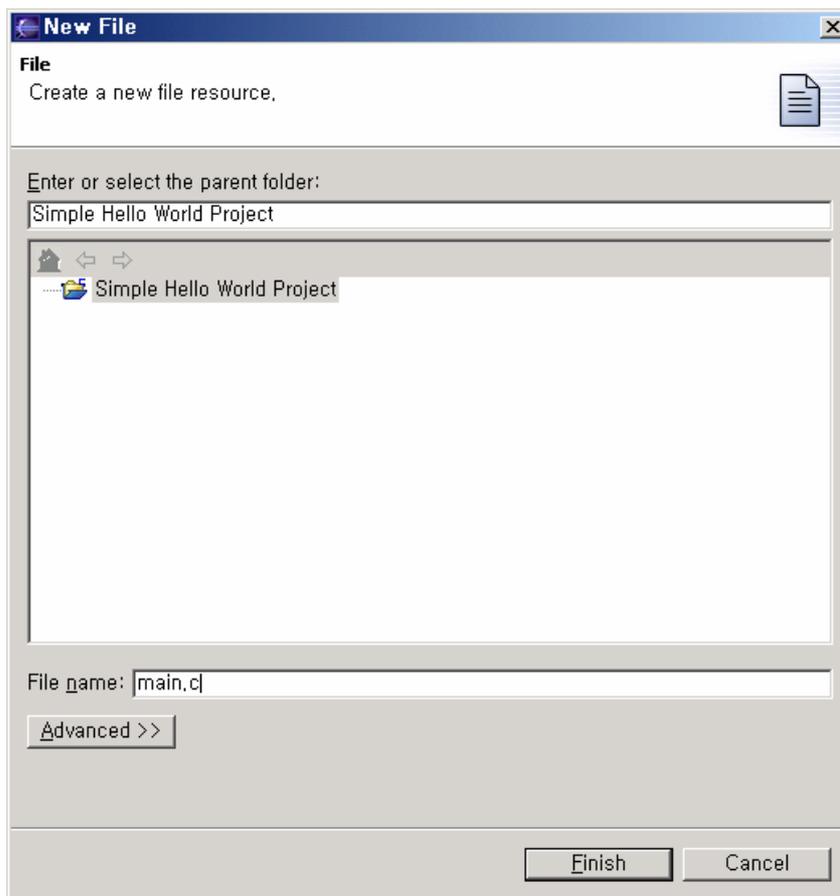
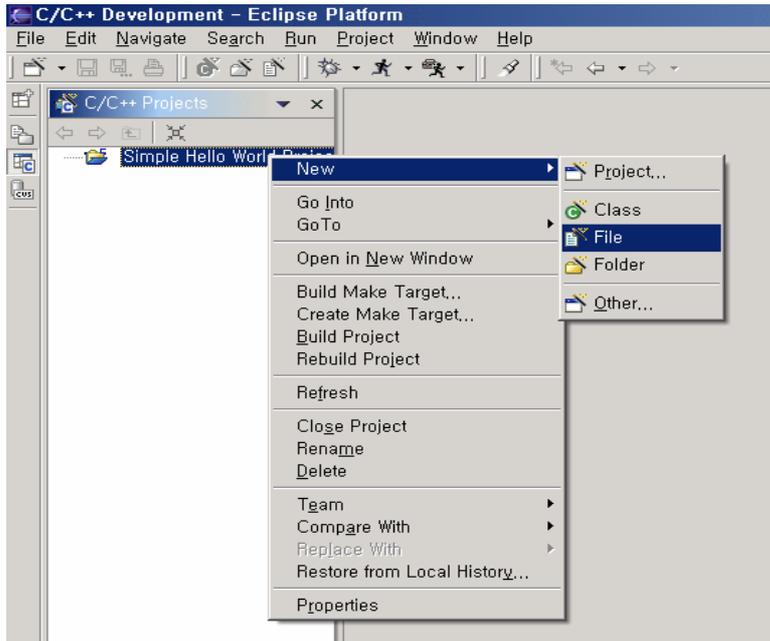
1. Standard Make C Project 를 선택한다. Managed Make C Project 는 CDT 가 Makefile 의 생성을 담당하는 것인데, 아키텍처와 링커마다 상이하므로 본 문서에서는 다루지 않는다.



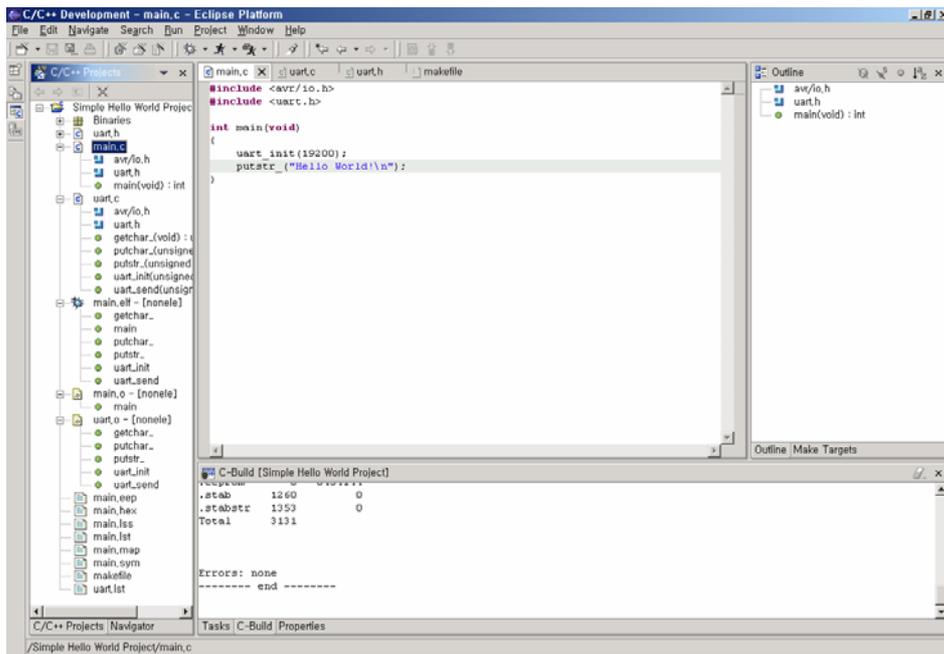
2. Project 의 Root 와 Name 을 지정한다. 지정하는 Location 은 프로젝트의 root 가 된다.



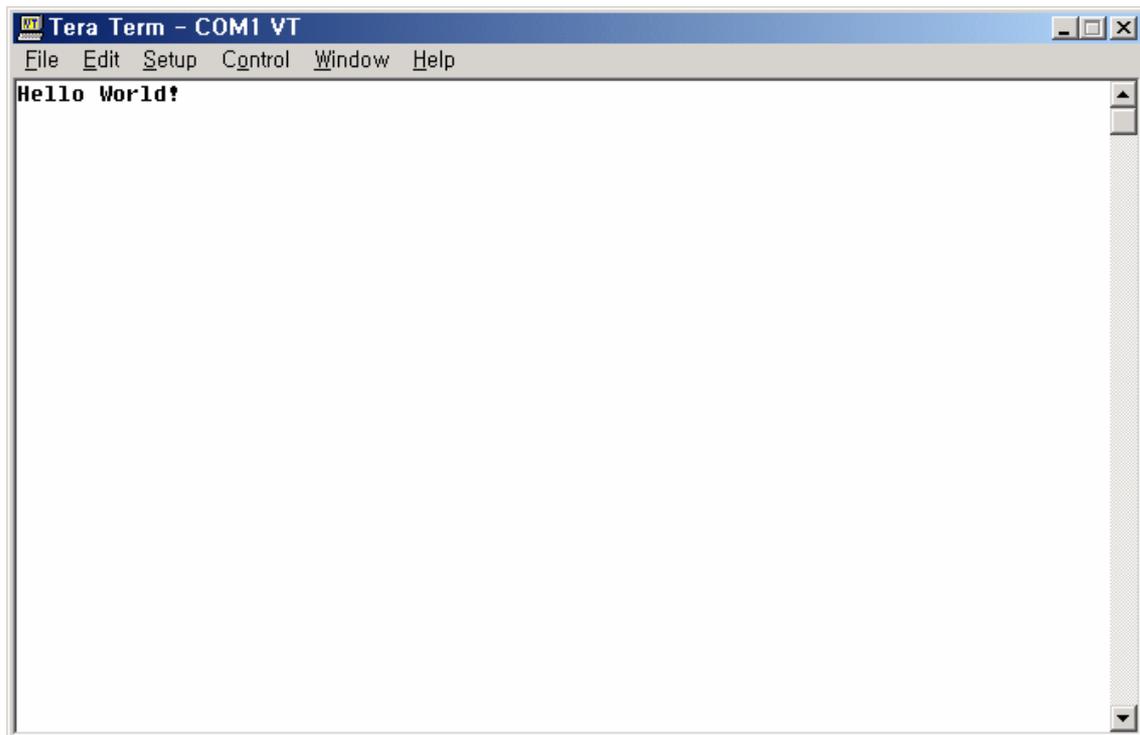
3. 새 파일을 프로젝트에 등록한다.



4. 개발을 완료하고 빌드를 한다.



5. 시리얼 터미널에 Hello World! 가 출력된 모습



6. FUTURE AND ECLIPSE

이클립스는 앞으로 출시될 GNU C 컴파일러와도 무리 없이 연동될 것이다. AVR 이외에도

다양한 아키텍처와 연동하는 문서가 빨리 나와 주었으면 하는 바램이다.

7. To do

다양한 아키텍처의 GDB 를 Eclipse 와 연동하는 것. JTAG 이 아닌 GDB 로 디버깅 하는 것은 Circuit 안에서 행하여 지는 것이 아니기 때문에 큰 의미는 없다. X86 용 GDB 의 경우 CPU 레지스터를 직접 추적 할 수 있으므로 디버거의 가치가 있는 것이다. 단, JTAG 연동이 지원되는 GDB 의 경우 Eclipse 와 불일만한 가치가 높다.