

## Qplus 패키지 자동 생성 도구 개발

우덕균<sup>0</sup>, 임채덕, 김홍남  
 한국전자통신연구원 임베디드S/W기술센터  
 (dkwu<sup>0</sup>, cdlim, hnkim)@etri.re.kr

### Development of Tool to Generate Qplus Package Automatically

Duk-Kyun Woo<sup>0</sup>, Chaedeok Lim, Heung-Nam Kim  
 Embedded Software Technology Center, Electronics and Telecommunications Research Institute

#### 요 약

한국전자통신연구원(ETRI)에서 개발한 임베디드 리눅스인 Qplus[1]의 개발환경은 타겟 설정 도구인 타겟 빌더[2]와 통합개발환경인 Esto[3]를 포함한다. 타겟 빌더는 타겟에 설치될 응용, 라이브러리 등에 대해서 패키지 형태로 관리하고 있다. 개발자가 Esto를 사용하여 개발된 응용 프로그램을 타겟 빌더의 패키지로 추가하기 위해서는 패키지 명세 구문을 익혀서 수작업으로 패키지를 작성해야 한다. 이와 같은 과정은 개발자에게 프로그램 개발 이외의 오버헤드를 제공하기 때문에 전체적인 개발 시간이 길어질 수 있다. 본 연구에서는 이와 같은 문제를 해결하기 위하여 타겟 빌더의 패키지를 자동으로 생성하는 도구를 개발하였다. 본 도구는 Eclipse[4] 기반으로 개발되었으며, 개발중인 Eclipse 기반의 Esto와 타겟 빌더와 연동되어 개발자에게 사용하기 편리한 통합 Qplus 개발환경을 제공하여, Qplus 기반의 임베디드 시스템 개발 시간을 단축시킬 수 있을 것으로 기대한다.

#### 1. 서론

최근의 임베디드 리눅스 개발환경은 임베디드 리눅스 타겟 시스템을 설정하고 설치하는 타겟 설정 도구와 설치된 타겟 시스템에서 수행되는 소프트웨어 개발을 도와주는 통합개발환경을 기본으로 포함하고 있다. Montavista의 임베디드 리눅스 개발환경인 DevRocket은 타겟 설정 도구와 소스 편집-빌드-실행-디버깅을 지원하는 통합개발환경을 포함하고 있으며[5], Timesys의 임베디드 리눅스 개발환경인 TimeStorm도 타겟 설정 도구와 통합개발환경을 포함하고 있다[6]. 한국전자통신연구원(ETRI)에서 개발한 임베디드 리눅스인 Qplus 개발환경에서는 타겟 설정 도구인 타겟 빌더와 통합개발환경인 Esto를 포함한다[3].

이와 같은 타겟 설정 도구는 타겟 시스템에 설치될 라이브러리, 응용들을 패키지 형태로 관리하고 있다. DevRocket과 TimeStorm의 타겟 설정 도구의 패키지는 바이너리 기반으로 구성되며 타겟 빌더의 패키지는 소스 기반으로 구성된다. 또한 타겟 빌더의 패키지는 DevRocket과 TimeStorm의 타겟 설정 도구의 패키지와는 다르게, 개발자가 타겟 빌더에 새로운 패키지를 추가할 수 있는 기능을 제공한다.

타겟 빌더에 새로운 패키지를 추가하는 기능은 다른 타겟 설정 도구와 비교하여 타겟 빌더의 확장성을 높이며, 소스 기반의 패키지 형태는 타겟 빌더의 새로운 타겟 플랫폼과 크로스 컴파일 물체에 대한 이식성을 높이는 장점을 갖는다[2].

그러나 타겟 빌더에 새로운 패키지를 추가하는 과정은 많은 시간과 노력을 필요로 한다. 개발자가 패키지 작성에 필요한 패키지 명세 파일에 대한 구문을 익혀야 하는 추가적인 오버헤드를 제공하기 때문이다. 이러한 문제를 해결하기 위하여 본 연구에서는 Esto에서 개발이 완료된 프로그램을 보다 쉽게

타겟 빌더의 패키지로 생성하는 패키지 자동 생성 도구를 개발하였다.

#### 2. Qplus 개발 환경

Qplus는 한국전자통신연구원에서 개발한 임베디드 리눅스로, 기존 리눅스에서 저전력 관리, 실시간 스케줄링 등의 기능이 확장되었다[1]. 이와 같은 Qplus가 탑재된 임베디드 시스템의 소프트웨어를 개발할 수 있는 환경을 Qplus 개발환경이라 하며, 본 개발 환경은 타겟 빌더와 Esto로 구성된다.

타겟 빌더는 Qplus 타겟 시스템에 설치될 커널, 라이브러리, 응용 등을 GUI 기반으로 쉽게 설정하고, 빌드하고, 설치할 수 있게 해주는 도구이다[2].

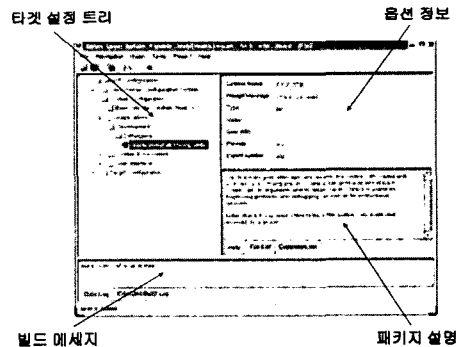


그림 1 타겟 빌더 GUI

그림 1은 타겟 빌더 GUI 창을 나타내며, 좌측의 트리 형

태는 커널, 라이브러리, 응용 등에 대한 설정 트리이며, 하단 은 빌드 메시지가 출력되는 뷰이다. 그리고 우측 상단은 선택 된 패키지에 대한 옵션 정보들이 나타나는 뷰이며, 우측 하단 은 패키지에 대한 정보가 나타난다. 개발자는 이와 같은 GUI 환경에서 타겟 시스템에 포함할 커널, 라이브러리 응용 등을 설정하고, 빌드하여 커널 이미지와 루트 파일 시스템 이미지를 생성한다. 그리고 마지막으로 개발자는 이와 같은 커널 이미지와 루트 파일 시스템 이미지를 타겟에 적재하여, Qplus가 부팅되도록 한다.

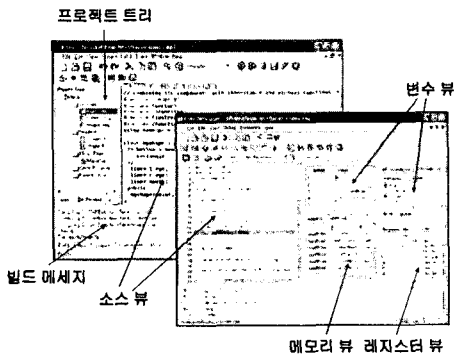


그림 2 Esto GUI

Esto는 Qplus 타겟 시스템에서 수행되는 프로그램 개발을 위한 GUI 기반의 통합개발환경이다[3]. Esto는 소스 편집 및 브라우징, 크로스 컴파일, 원격 실행, 원격 디버깅, 원격 모니터링 등의 기능을 제공하여, 개발자는 편리하게 호스트 시스템에서 타겟에서 수행되는 프로그램을 개발할 수 있다.

그림 2는 Esto GUI를 나타낸다. 좌측의 창은 소스 편집-빌드-실행이 수행되는 창이며, 우측의 창은 디버깅이 수행되는 창이다. 작성된 소스 코드에 대한 빌드는 크로스 컴파일러를 사용한 크로스 컴파일과 링크를 포함하며, 실행은 빌드된 바이너리 코드를 타겟으로 다운로드하여 실행하는 과정을 말한다. 디버깅은 타겟에 다운로드된 코드의 실행을 정지시켜가며 타겟의 변수, 메모리, 레지스터 정보를 확인하여 프로그램의 실행 오류를 발견하는 과정을 말한다.

개발자는 타겟 빌더를 사용하여 소프트웨어 개발에 필요한 기본 타겟 시스템을 구축하고 나서, Esto를 사용하여 타겟에서 수행되는 라이브러리, 응용 등을 개발한다. 개발자는 개발된 프로그램들을 패키지 형태로 타겟 빌더에 포함하고, 타겟 빌더를 사용하여 최종적인 타겟 시스템을 완성한다. 그러나 이와 같은 과정에서 Esto에서 개발된 프로그램을 타겟 빌더가 처리하는 패키지 형태로 변환하는 작업은 자동화되기 보다는 개발자가 패키지의 영세 구문을 익혀서 일일이 수작업으로 패키지를 만들어야 하기 때문에, 최종 타겟 시스템 개발 시간은 길어질 수 있다. 따라서 본 연구에서는 이와 같은 문제를 해결하기 위하여 Esto에서 개발된 프로그램을 자동으로 타겟 빌더의 패키지로 변환하는 다음과 같은 도구를 개발하였다.

### 3. Qplus 패키지 자동 생성 도구

#### 3.1 Qplus 패키지

타겟 빌더가 포함하는 라이브러리, 응용들에 대한 패키지를 본 논문에서는 Qplus 패키지로 부른다. 즉 Qplus 패키지는 Qplus 타겟 시스템에 설치되는 라이브러리, 응용들에 대한 패키지를 말하며, 소스 RPM과 QPD(Qplus Package Descriptor)로 구성된다. 소스 RPM은 RedHat에서 제공하는 소스 기반의 패키지를 말하고[7], QPD는 본 Qplus 패키지에 대한 정보를 표시하는 영세 파일을 말한다[2]. 사용자는 타겟 빌더 UI를 통하여 Qplus 패키지를 선택적으로 타겟 루트 파일 시스템에 포함시킬 수 있으며, Qplus 패키지들 간의 의존 정보도 확인할 수 있다.

사용자는 새로운 응용 프로그램을 개발하고, 본 응용 프로그램을 Qplus 패키지 형태로 만들기 위해서는 소스 RPM과 QPD 파일을 작성해야 한다. 소스 RPM은 RPM의 SPEC 파일을 작성하고, 소스와 SPEC 파일을 바탕으로 소스를 빌드하여 만들지며, QPD 파일은 SPEC 파일을 바탕으로 QPD 파일을 생성한다. 또한 기존 Qplus 패키지의 소스를 수정하는 경우에는, 패키지에 포함된 소스 RPM으로부터 소스를 빼내는 작업을 해야 한다. 이와 같은 작업들은 모두 사용자가 커맨드 라인에서 RPM 명령과 편집기를 통하여 작업을 해야 한다. 또한 RPM의 SPEC 파일에 대한 구문을 이해하고 습득해야 하는 부담을 제공한다.

#### 3.2 자동 생성 도구 구조

본 논문에서 소개하는 Qplus 패키지 자동 생성 도구는 앞에서 소개한 Qplus 패키지의 생성 또는 패키지로부터의 소스 추출 등을 GUI 기반으로 사용자가 편리하게 수행해 주는 도구이다. 본 도구의 구조는

그림 3과 같다.

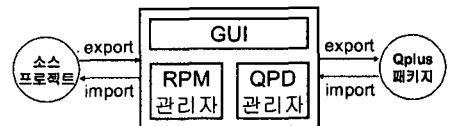


그림 3 Qplus 패키지 자동 생성 도구 구조도

본 도구는 크게 세 모듈로 구성되어, 사용자로부터 패키지 이름, 버전 등의 정보를 입력 받은 GUI 모듈, 소스 RPM의 생성, 소스 RPM으로부터 소스 추출 등의 기능을 담당하는 RPM 관리자 모듈, QPD 파일을 생성하는 QPD 관리자 모듈로 구성된다. 본 도구는 Qplus 패키지를 생성하기 위하여 소스 프로젝트를 입력 받는다. 소스 프로젝트는 프로그램의 소스와 빌드 정보 등을 포함하는 프로그램 개발의 기본 단위이며, Esto를 포함한 대부분의 GUI 기반의 통합개발환경은 이와 같은 소스 프로젝트 기반의 개발환경을 지원한다.

### 3.3 자동 생성 도구 기능

Qplus 패키지 자동 생성 도구의 기능은 패키지의 익스포트(export) 기능과 임포트(import) 기능을 갖는다. 패키지의 익스포트는 소스 프로젝트로부터 Qplus 패키지를 생성하는 과정을 말하며, 임포트는 Qplus 패키지로부터 소스 프로젝트를 만드는 과정을 말한다. Qplus 패키지 익스포트 과정은 다음과 같은 순서로 진행된다.

1. 소스 프로젝트 정보와 패키지 정보 입력
2. 소스 프로젝트로부터 tarball 파일과 SPEC 파일 생성
3. tarball 파일과 SPEC 파일로부터 소스 RPM 파일 생성
4. SPEC 파일로부터 QPD 파일 생성

첫번째 단계는 GUI 모듈이 처리하는 단계로 사용자로부터 소스 프로젝트 이름과 생성될 패키지 이름, 버전 등의 패키지 정보를 입력 받는 단계를 말하며, 두번째 단계는 소스 프로젝트로부터 소스 RPM 생성에 필요한 tarball 파일 생성과 소스 RPM 파일 생성에 필요한 SPEC 파일 생성 단계이다. tarball 파일은 tar파일로 묶이고 압축된 파일을 의미한다. spec 파일은 패키지 이름, 패키지 버전, 릴리즈 번호 등의 정보로부터 자동 생성된다. 세번째 단계는 두번째 단계에서 생성된 tarball 파일과 SPEC 파일로부터 RPM 명령을 사용하여 소스 RPM을 생성하는 단계이다. 그리고 마지막 네번째 단계는 SPEC 파일로부터 QPD 파일을 자동으로 생성하는 단계이다.

Qplus 패키지 임포트 과정은 다음과 같다.

1. 패키지 파일 이름과 프로젝트 이름 입력
2. 패키지 파일의 소스 RPM 파일로부터 tarball 파일 추출
3. tarball 파일로부터 소스 파일과 빌드 정보 프로젝트에 추가

사용자는 자동 생성 도구 GUI로부터 패키지 파일 이름과 프로젝트 이름을 사용자로부터 입력 받고, 패키지 파일 이름과 프로젝트 이름은 생성 도구의 RPM 관리자에게 전달된다. RPM 관리자는 패키지 파일의 소스 RPM 파일을 RPM 명령으로 풀어 tarball 소스 정보를 추출하고, 다시 tarball 파일을 풀어 소스와 빌드 정보를 추출한다. 소스와 빌드 정보는 소스 프로젝트에 추가되어 사용자는 바로 소스를 수정하여, 빌드, 실행할 수 있다.

### 3.4 구현

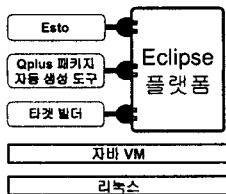


그림 4 Qplus 패키지 자동 생성 도구 구현

Qplus 패키지 자동 생성 도구의 구현은

그림 4와 같이 Eclipse 플랫폼 기반의 플러그인 형태로 개발되었다. 플러그인 개발은 Eclipse SDK 3.0 에서 개발하였으며, 리눅스는 RedHat 9.0을 사용하였다. 기존 Qt[8] 기반의 Esto와 wxWindows[9] 기반의 타겟 빌더는 현재 Eclipse 플랫폼 기반으로 개발 중이다.

Esto, 자동 생성 도구, 타겟 빌더가 같은 Eclipse 플랫폼에서 동작하므로, 사용자는 같은 Eclipse 기반의 GUI 환경에서 보다 편리하게 Esto에서 개발된 프로그램을 패키지로 변환하여 타겟 빌더에게 제공할 수 있으며, 타겟 빌더의 패키지를 보다 쉽게 Esto의 소스 프로그램으로 가져올 수 있다.

### 4. 결론

한국전자통신연구원에서 개발한 Qplus 타겟 시스템의 개발 환경으로 타겟 빌더와 Esto가 널리 사용되고 있다. 타겟 빌더는 타겟 시스템을 설정, 빌드, 설치하는 도구이며, 타겟 시스템에 설치되는 라이브러리, 응용 등은 Qplus 패키지라 불리는 소스 RPM 기반의 패키지 형태로 제공된다. 개발자가 프로그램 개발 도구인 Esto를 사용하여 개발한 프로그램을 타겟 빌더에 포함시키거나 또는 기존 패키지가 포함하는 소스를 수정 또는 브라우징하기 위하여 Esto로 가져오는 과정은 많은 시간을 필요로 하였다. 결과적으로 최종적인 타겟 시스템을 구축하는 과정에서, 응용 개발하는 시간에 더하여 패키징하는 데 많은 시간이 할애되었다.

이와 같은 문제를 해결하기 위하여 본 연구에서는 Esto에서 개발한 응용 프로그램을 Qplus 패키지로 가급적 쉽고 빠르게 임포트, 익스포트하는 Qplus 패키지 자동 생성 도구를 개발하였다.

본 자동화 도구는 Qplus 개발환경의 타겟 빌더와 Esto를 보다 밀접하게 연동시켜줌으로써, 개발자에게 보다 편리한 Qplus 개발환경을 제공하여 Qplus 기반의 임베디드 시스템 개발 시간을 단축시킬 수 있을 것으로 기대한다.

### 참고문헌

- [1] Qplus, <http://qplus.etri.re.kr>.
- [2] W.C. Kang, etc. "Target Builder: An Embedded Linux Development Toolkit", The 6th IASTED Conference on Software Engineering and Applications, 2002.
- [3] 임채덕, "Esto : Qplus 통합개발환경", 한국정보과학회 컴퓨터시스템연구회 주관 2002년 동계 워크샵 발표집, pp. 116-143, 2002년 1월.
- [4] Eclipse, <http://www.eclipse.org>.
- [5] MontaVista DevRocket,  
<http://www.mvista.com/pro/devrocket.html>.
- [6] TimeStorm,  
[http://www.timesys.com/index.cfm?bdy=tools\\_bdy.cfm](http://www.timesys.com/index.cfm?bdy=tools_bdy.cfm).
- [7] E.C. Bailey, "Maximum RPM : Taking the RedHat Package Manager to the Limit", Sams publishing, 1997.
- [8] Trolltech, "Qt 3.0 Whitepaper", Available in <http://www.trolltech.com/products/qt/whitepaper/whitepaper.html>.
- [9] wxWindows, <http://www.wxwindows.org/>.