이순신 운영체제

사용자 설명서

목차

1. 본 문서는

이순신 운영체제의 사용 설명서로 운영체제의 모든 정보를 담고 수시로 업데이트가 될 예정입니다.

이 문서에서 다루는 MCU의 코어들은 모두 ARM Cortex-M 계열 제품으로 별도의 설명이 없는 경우 ARM Cortex-M3, M4F, M7을 기준으로 한다.

2. 이순신 운영체제는?

2.1. 운영체제의 시작

2007년 정부 지원으로 MDS 아카데미에서 6개월간 임베디드 교육을 받던 때였습니다. ARM 아키텍처와 네오스라는 MDS에서 직접 만든 운영체제 수업을 듣던 중에 문맥 전환을 하는 방법에 대해 강한 호기심을 갖게 됐습니다.

처음에는 ATmega128(AVR 8bit)을 시작으로 AT32UC3(AVR 32bit), ARM Cortex-M 계열에서 동작하는 선점형 라운드 로빈 스케줄링을 사용하는 문맥 전환 코드를 구현했습니다.

JAVA의 GUI 프로그래밍에 강한 매력을 느끼고 그와 유사한 방식으로 MCU에서 사용 가능한 라이브러리를 만들어보고 싶었습니다. 그 결과 c++을 사용하여 유사한 느낌의 그래픽 라이브러리 구현에 성공하게 되었습니다.

2007년부터 2014년까지 개인적으로 공부해보는 수준에서 다루어 보았지만, 2015년에 처음 코드를 오픈 했습니다.

2.2. 개발 컨셉

모든 ARM Cortex-M 계열의 MCU들을 제조사에 관계 없이 동일한 환경에서 개발이 가능하도록 하는 것입니다. 기본 내부 장치 드라이버를 모두 통일하고 MCU의 내부 뿐만 아니라 외부에 장착되는 부품들에 대한 라이브러리까지도 완벽하게 공유할 수 있도록 하는 것입니다.

최근 MCU 제조사들이 주변 장치에 대한 코드 생성기를 제공 해줌으로써 주변 장치를 이전보다 쉽게 사용하는 것을 고려했다면, 이에 더 나아가 주변 부품들에 대해 단순한 포팅과 초기화만 해주면 쉽게 사용할 수 있도록 하는 것 입니다.

지원하는 모든 MCU는 동일한 그래픽 라이브러리를 지원하기 때문에 MCU가 바뀌더라도 이전에 작성한 GUI 코드를 그대로 활용이 가능하도록 하고 있습니다.

2.3. 적용된 프로젝트

os가 적용된 대표적인 프로젝트는 아래와 같습니다.

> 카드 프린터

열 전사 방식 컬러 카드 프린터의 인쇄 엔진과 프린터 운용을 동시에 처리하는 프로젝트에서 적용 되었습니다. 2019년부터 양산에 들어갔습니다.

> 선박 엔진 모니터

선박의 ECU 엔진으로부터 들어오는 정보를 디스플레이 하는 프로젝트에서 적용 되었습니다. 해당 제품은 2020년부터 양산에 들어갔습니다.

> 가스 엔진 쓰로틀 밸브

스피트 컨트롤러로부터 밸브의 열림 양을 받아, 제어하여 엔진의 RPM을 컨트롤 하는 장치로

현재 테스트 까지 진행되어 있습니다.

> 가스 엔진 믹서

람다 컨트롤러에서 넘어오는 열림 양을 받아, 제어하여 가스와 공기의 혼합 비율을 제어하는 장치로 현재 테스트 까지 진행되어 있습니다.

> AVR(Auto Voltage Regulator) 컨트롤러

비상 발전기용 디젤 엔진의 동체의 필드 전압을 제어하여 요구되는 전압으로 조정하는 프로젝트에서 적용 되었습니다. 해당 제품은 2021년부터 양산에 들어갔습니다.

앞으로도 계속 새로운 프로젝트에 적용될 예정입니다.

2.4. 향후 계획

지속적인 업그레이드를 해 나아가고 이미 작성된 코드에 대해서도 최적화를 해 나갈 계획입니다.

3. 기본 구성

3.1. 지원 MCU

자세한 지원 목록은 아래 주소를 참고하시기 바랍니다.

https://cafe.naver.com/yssoperatingsystem/384

3.2. IDE

현재 사용 중인 IDE는 **Crossworks for ARM**으로 ST를 포함한 많은 MCU 제조사의 ARM Cortex MCU들을 지원하는 개발 환경입니다. 리눅스, 윈도우, 맥을 지원합니다.

현재 Crossworks for ARM 외의 다른 툴은 지원하고 있지 않습니다.

4. Crossworks for ARM

4.1. Crossworks for ARM을 사용하는 이유?

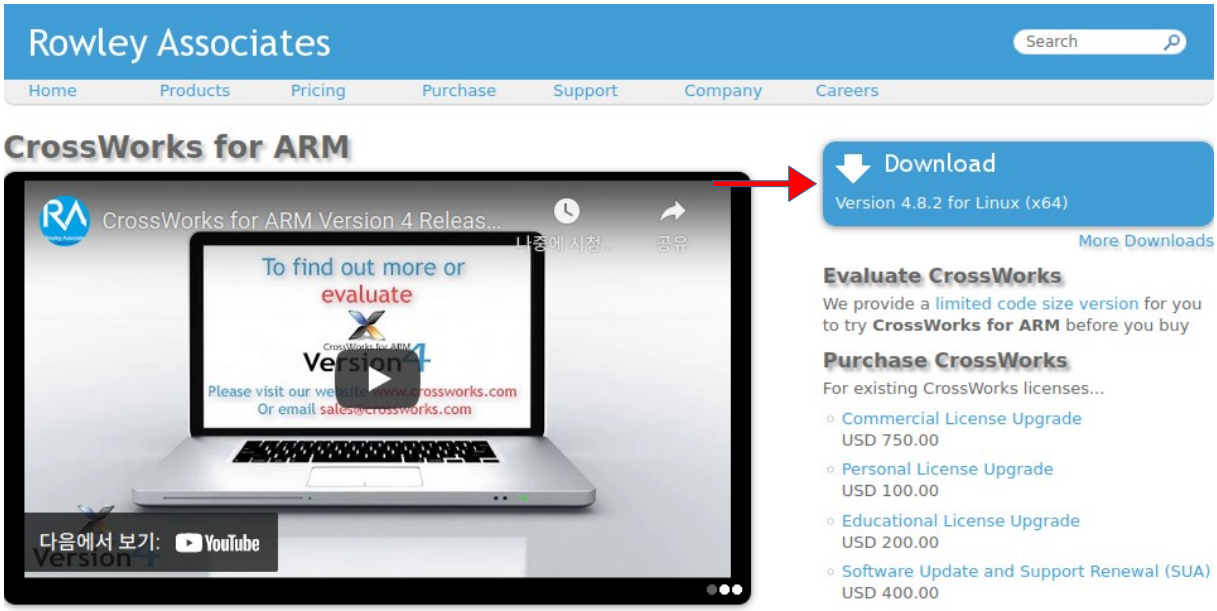
그래픽 라이브러리 개발에 c++을 사용하기 원했습니다. 그러나 IAR, Keil 등의 툴에서 전역으로 선언된 객체의 생성자를 호출 해주는 startup code 부분이 제공이 안됐습니다. crossworks for ARM에서는 해당 부분을 잘 지원 해주고 부가적으로 리눅스와 맥에서도 사용 가능한 장점이 있었습니다.

이순신 운영체제는 ST사의 MCU만 지원 할 것이 아니라 다른 MCU들도 지원할 계획이어서 제조사 전용 IDE가 아닌 범용 IDE가 필요 했고, 가격도 비교적 저렴한 편입니다.

4.2. 다운로드

아래 URL을 클릭하여 Rowley Associates 홈페이지에 방문합니다.

https://www.rowley.co.uk/arm/index.htm



위와 같이 Download 버튼을 클릭하여 설치 파일을 다운로드 받습니다.

웹 페이지에 접속한 PC의 운영체제에 따라 Download 버튼의 다운로드 받을 프로그램의 OS 버전이 결정됩니다. 만약 다른 OS의 버전이나 이전 버전을 원하실 경우, 버튼의 우측 하단에 나타난 More Downloads 글자를 클릭 합니다.

4.3. 설치

Rowley Associates 홈페이지에 나타난 방법을 참고 바랍니다.

4.4. 라이센스

Crossworks for ARM의 라이센스는 네 가지가 있습니다.

- **개발자 공유 라이센스** (Shared Developer)

  모든 개발자가 USB 동글을 이용하여 라이센스를 공유하는 방식입니다.

- **개발자 지정 라이센스** (Named Developer)

  지정된 사람에 대해 본인의 PC에 설치 갯 수에 관계 없이 사용하는 방식입니다.

- **교육기관용 라이센스** (Educational Workstation License)

  대학, 학교등의 교육 기관에서 라이센스당 한 대의 PC에 설치 할 수 있는 방식입니다.

- **개인용 라이센스** (Personal Non-Commercial License)

  개인이 비 상업적으로 사용할 목적으로 본인의 PC에 설치 갯 수에 관계 없이 사용하는 방식입니다. 비 상업적 사용이 불가능한 것 외에는 **개발자 지정 라이센스**와 동일합니다.

라이센스는 웹에서 구입 가능합니다. 개인용 라이센스의 경우 구입 절차 상에 비 상업적 사용에 관한 서약서를 작성해야 합니다.

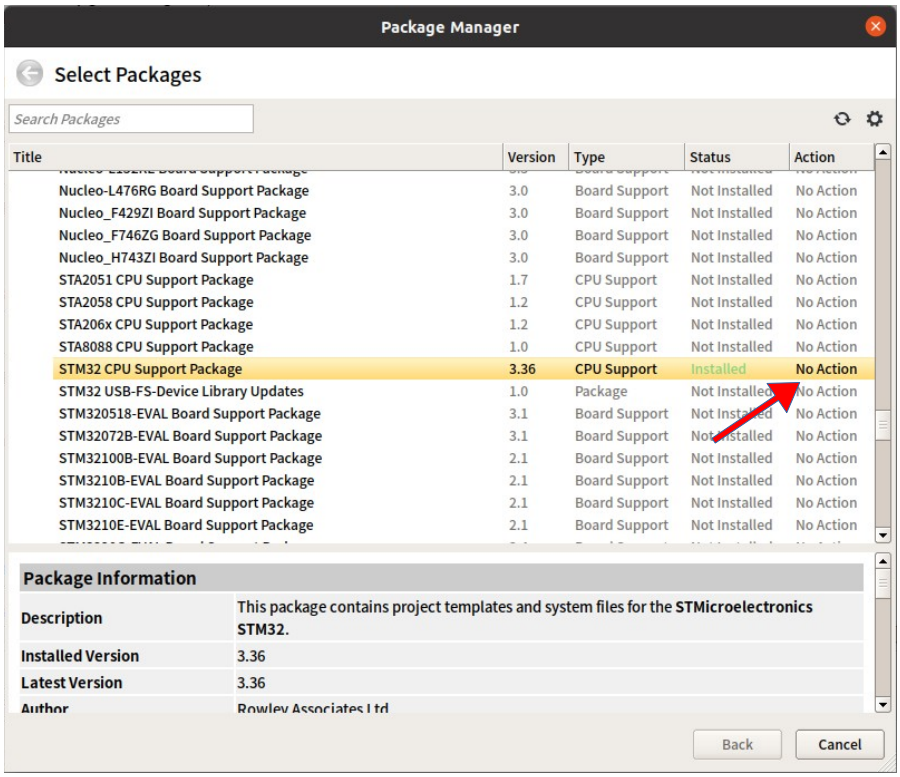
4.5. 평가판

라이센스 구입에 앞서 사용자는 1개월 평가판 사용이 가능합니다. 경험에 의하면 약 2~3회 가량 한 메일 계정에 대하여 평가판 신청이 가능합니다. 평가판은 유료 라이센스 버전과 동일하게 기능의 제약 없이 사용 가능합니다.

기본적으로 라이센스가 설치되지 않으면, 16 kB까지 코드를 컴파일하고 디버깅하는데 유료 라이센스 버전과 동일하게 기능의 제약 없이 사용 가능합니다.

4.6. 패키지 설치

메뉴에서 Tools → Package Manager를 클릭 합니다.



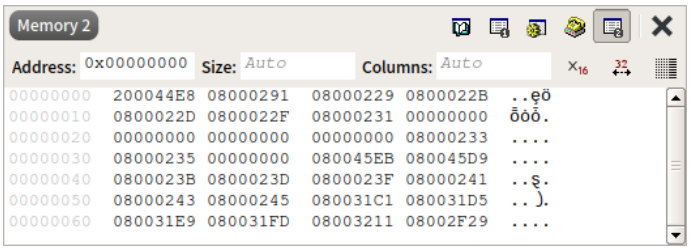
사용하고자 하는 CPU의 지원 패키지를 찾아 더블 클릭 하면 화살표가 가리키는 부분의 상태가 바뀝니다. No Action은 아무것도 하지 않는 상태이고, 설치 되지 않은 패키지를 더블 클릭 할 경우 install로 상태가 바뀝니다. 이때 하단에 나타나는 Next 버튼을 클릭하면 설치가 진행 됩니다.

CPU 지원 패키지가 설치되지 않은 경우 해당 MCU에 대해 사용이 불가능 합니다.

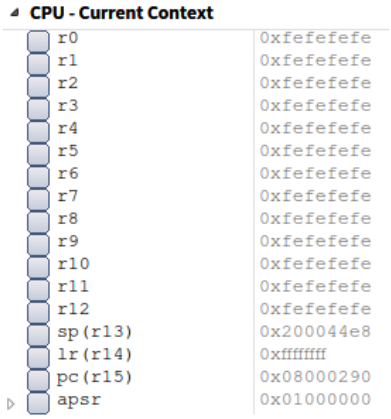
5. 부팅 과정

5.1. reset_handler

리셋이 걸리면 기본적으로 플래시의 0번지에 저장된 4바이트를 Stack Pointer(SP)로 4번지에 저장된 4바이트를 Program Counter(PC) 가져온다.

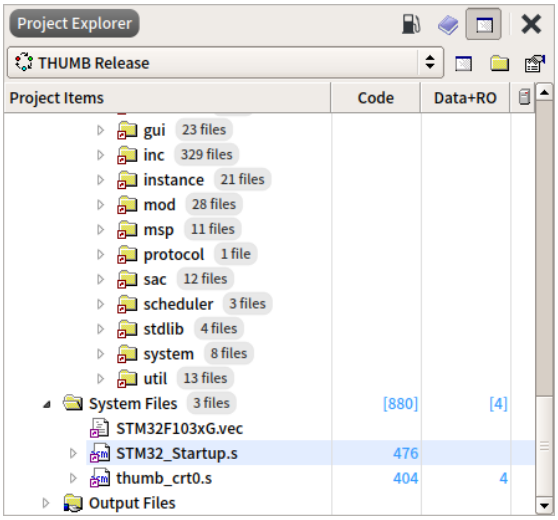


위 그림에 나타난 0번지의 내용은




```
113 reset_handler:

    #ifndef·__NO_SYSTEM_INIT
    ··ldr·r0,·=__RAM_segment_end__
    ··mov·sp,·r0
    ··bl·SystemInit
    #endif
120
    #ifdef·VECTORS_IN_RAM
    ··ldr·r0,·=__vectors_load_start__
    ··ldr·r1,·=__vectors_load_end__
    ··ldr·r2,·=_vectors_ram
    l0:
```



위 그림과 같이 “System Files” 폴더의 STM32_Startup.s 파일에 reset_handler가 들어 있습니다.