

2012 공개SW커뮤니티 지원사업

상용 게임기를 이용한 임베디드 소프트웨어 교육 환경 구성

과제책임자: 이민석

한성대학교

minsuk@hansung.ac.kr

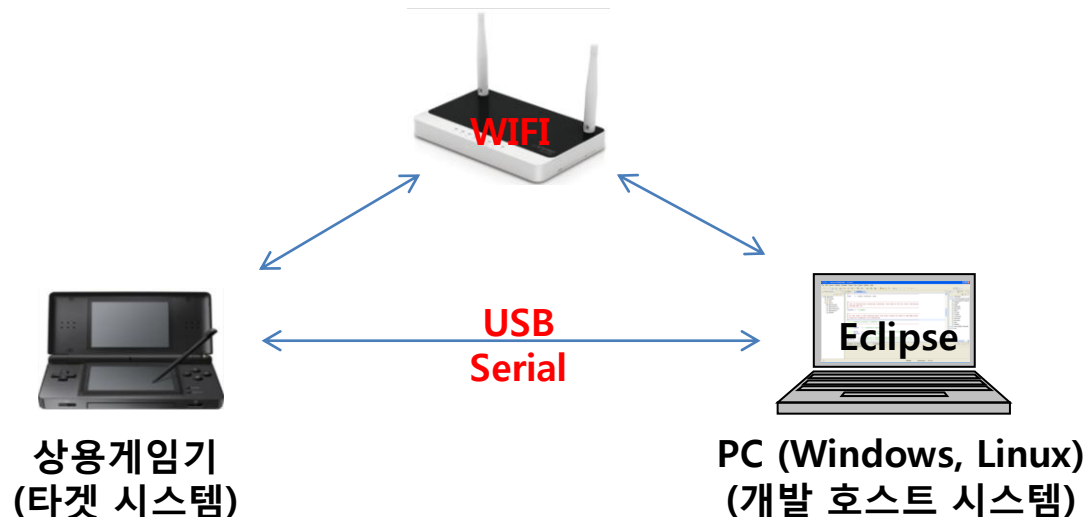
개발 목표

- 프로젝트 주제

- 상용 게임기를 이용한 임베디드 소프트웨어 교육 환경 구성

- 프로젝트 목표

- 닌텐도 DS와 같은 상용 게임기 하드웨어를 이용하여 마이크로 프로세서 또는 임베디드 소프트웨어 교육 실습을 수행하기 위한 개발 환경을 구성



개발 동기

※ 기존 임베디드 시스템 교육 장비(환경)의 문제점 극복



VS



- 기존 임베디드 시스템 교육 환경

1. 기존의 임베디드 시스템 개발 장비는 가격이 매우 비싸다
2. 학생들에게 친근하지 않다
3. 완벽한 개발 환경의 제공이 미흡하다

개발 동기

- 상용 게임기를 이용한 임베디드 시스템 교육 환경
 1. 가격이 저렴하다 (닌텐도 DS-Lite 10만원 내외)
 2. 임베디드 시스템을 처음 배우는 학생도 대부분 익숙하다
 3. Eclipse 기반의 통합 개발 환경 하에서 원격 디버깅을 가능하게 하여 품질 높은 S/W를 개발하기 위해 충분한 환경을 제공한다
 4. 게임기는 다양한 임베디드 장치들을 내장하고 있다

※ 저렴한 가격으로 교육 기관, 동호회, 개인이 현실감 높고, 품질 높은 임베디드 소프트웨어를 개발할 수 있는 환경 제공

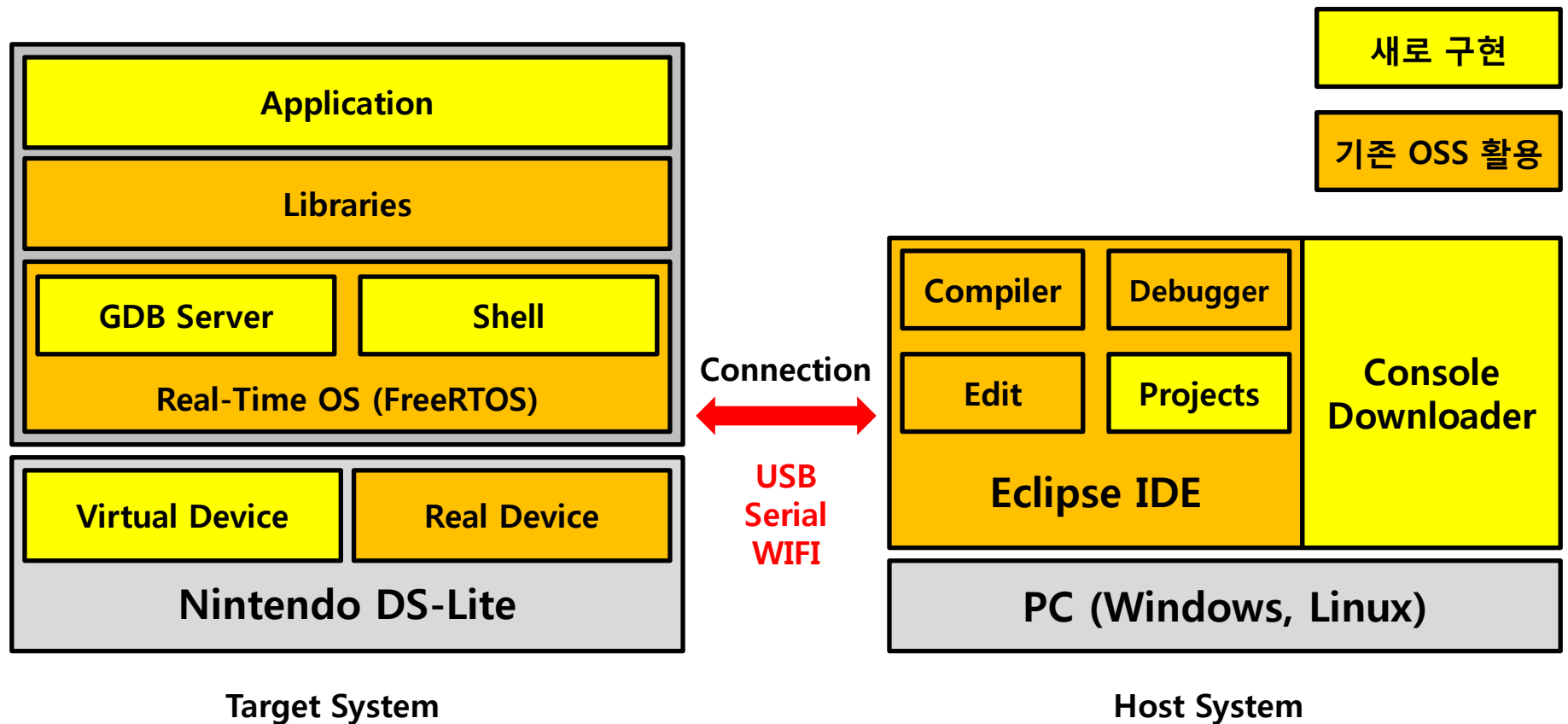
주요 개발 내용 (1 / 2)

- 임베디드 소프트웨어 교육을 위한 실습 환경이 갖추어야 할 여러 가지 요소들을 상용 게임기의 **게임 팩과 같은 형태로 개발 및 설치**
- 게임기로의 프로그램 **다운로드 및 실행**
- 게임기에서의 **실시간 운영체제 환경**을 구현

주요 개발 내용 (2 / 2)

- **원격 디버깅이 가능**하게 하는 게임기용 소프트웨어 구현
- 프로젝트 관리, 소스 편집, 컴파일, 다운로드 및 디버깅을 통합 처리하는 **통합 개발 환경** 구현
- 교육 실습을 위한 **실습용 교재 + 답안**

시스템 구성



시스템 구성

- Target System S/W

 기존 OSS

 새로 구현

- RTOS: FreeRTOS (+ remote debugging)

- Open Source Realtime Operating System : <http://www.freertos.org>

→ OSEK 호환 RTOS로 변경

Trampoline

<http://trampoline.rts-software.org/>

- Libraries: RTOS version of libwifi, libfat, libnds, pa_lib, ...

- Modify OSS to multitasking version : <http://www.devkitpro.org>

- Shell: configuration, download & run, file management

- Virtual device software

- Sample Applications (game...)

실시간 운영체제 (1 / 2)

- **FreeRTOS**

- Open Source Realtime Operating System : <http://www.freertos.org>

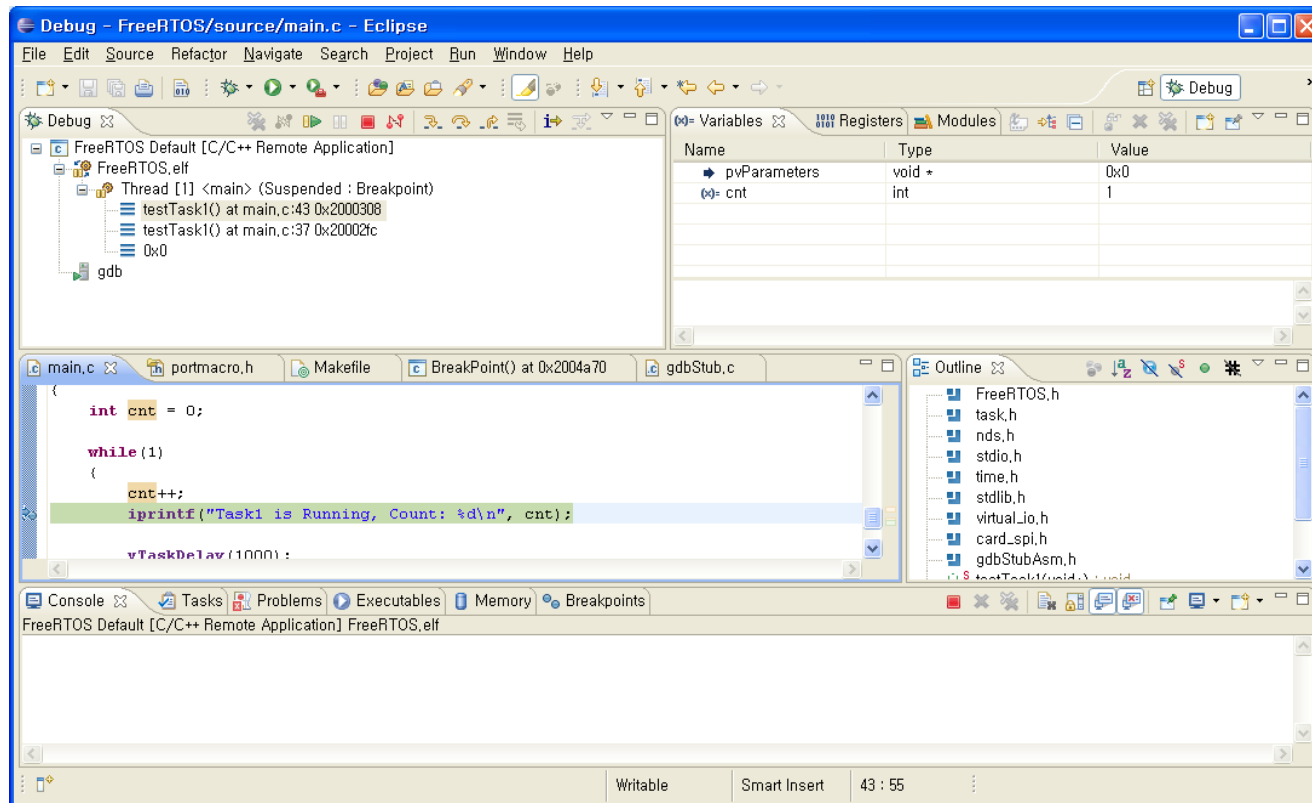


FreeRTOS 멀티 태스킹 실행 화면

- OSEK 호환 RTOS인 **Trampoline** 이식 예정

실시간 운영체제 (2 / 2)

- **Remote Debugging**
 - GDB Stub Module

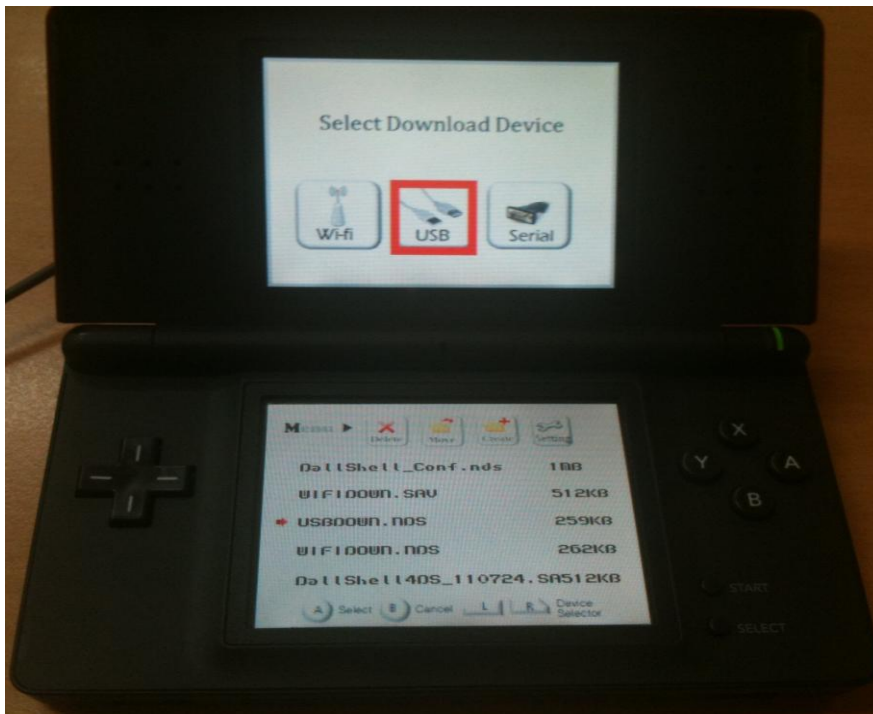


Eclipse 상에서 원격 디버깅

셸 프로그램

- **Shell Program**

- 게임기용 소프트웨어를 통합적으로 실행할 수 있는 기능 제공



초기 화면 - LCD 상단 (다운로드 모드 선택 메뉴)
- LCD 하단 (파일/디렉토리 관리 메뉴)



다운로드 실행



다운로드 환경 설정 실행

가상 디바이스 모듈

- **Virtual Device Module**

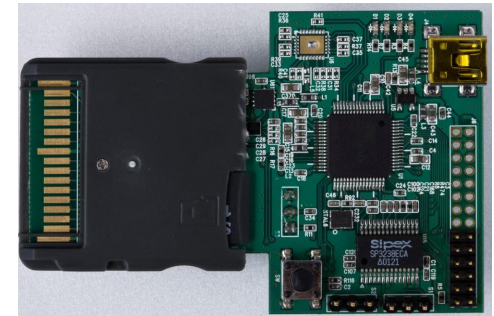
- LCD화면과 터치스크린 상에서 동작하는 하드웨어 시뮬레이터



가상 디바이스 모듈

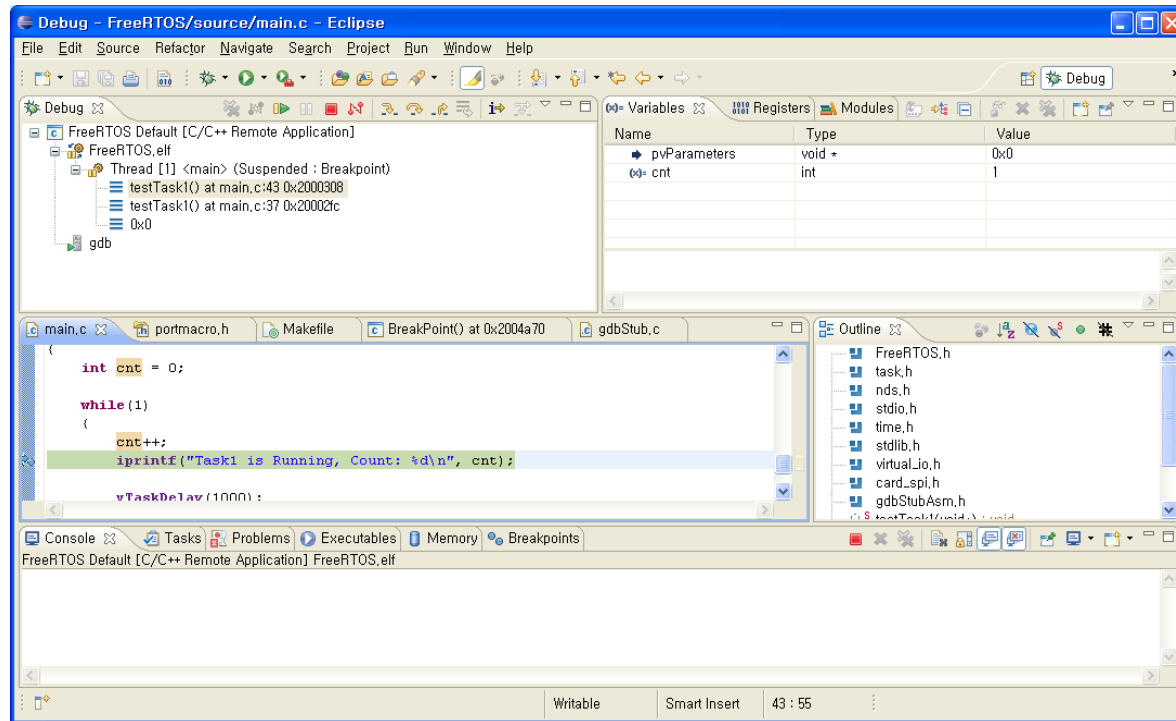
시스템 구성

- **Host System S/W (Windows, Linux)**
 - **Eclipse IDE**: devkitPro
 - <http://www.eclipse.org>, <http://www.devkitpro.org>
 - remote debugging 기능 추가
 - **Console Downloader** (WIFI, USB 이용)
- **Adapter Hardware (optional)**
 - Atmel 32bit CPU
 - 역시 FreeRTOS 사용
 - USB, Serial, JTAG 등 연결성
 - 3D 가속센서, LED, GPIO 등 실제 H/W
 - **추가적인 장치의 확장성을 위해 제작한 H/W Board**



통합 개발 환경

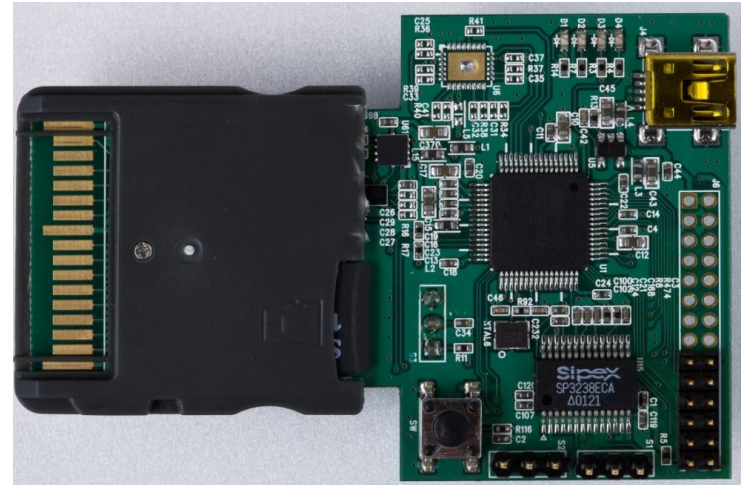
- Eclipse-CDT
- Remote Debugging
- USB / WIFI Download (어댑터 보드가 없는 경우 WIFI 만 이용 가능)



- Ubuntu 기반 Virtual Machine 이미지

어댑터 하드웨어

- Atmel 32 bit CPU + 3D-가속센서 + USB + GPIO + LED + Serial x2 + (지자기 + 자이로 센서)
- Nintendo DS의 Doctor tool R4 Compatible (Upto 8G Micro SD)
- FreeRTOS 기반 Firmware
- USB/Serial로 PC와 Nintendo를 연결
- 어댑터가 없어도 WIFI로 대부분 S/W 기능 이용 가능



프로젝트 활용 방안 (1 / 3)

프로젝트 결과물을 통한 수업 활용 예 (수업 결과물)

– 실험 Material

- 실험 목표 및 과제
- 정답 실행 이미지
 - download 해서 실행

– Example 소스

- FreeRTOS
- libnds
- 과제를 위한 예제 Task

– 실험 과제 Template 소스

sourceforge
SourceForge.net > Find Software > nintendo > SCM Repositories > nint

SCM Repositories – [nintendo](#)

Files shown: 0
Directory revision: 145 (of 154)
Sticky Revision: Set

File	Rev.	Age	Author
Parent Directory			
init_files/	117	3 weeks	uessie
keymatrix/	145	4 days	hilitij
simple-io-1/	115	3 weeks	uessie
simple-io-2/	115	3 weeks	uessie
simple-io-3/	132	2 weeks	sjjskkk

[Download GNU tarball](#)

[SourceForge Help](#) [ViewVC Help](#) [Powered by ViewVC 1.1.6](#)

Copyright © 2010 [Geeknet, Inc.](#) All rights reserved. [Terms of Use](#)

실험 4. Key Matrix 구동

1. 목적

임베디드 시스템에서 다수의 스위치 입력이 필요한 경우가 많다. 휴대폰 또는 전화기에 적어도 12개 이상의 스위치가 필요하며, 일반적으로 컴퓨터에 사용하는 키보드의 경우는 100개 이상의 스위치가 필요하다. 이 실험에서는 각 스위치를 한 비트씩 입력 포트에 연결하는 대신, Key Matrix를 구성함으로써 최소한의 입출력 포트를 이용할 수 있도록 구현된 회로를 소프트웨어적으로 이용하는 방법을 배운다.

또 실험에서는 7 Segment LED를 활용한 실습을 통하여 좀 더 실용적인 임베디드 시스템을 구성해 본다.

2. 하드웨어 설명

실험 환경의 가상 스위치 4개 (왼쪽, SW1 ~ SW4) 와 Key Matrix (오른쪽 1,2,3,4 ~ D,E,F,0)는 그림 1과 같으며 회로는 그림 2와 같다.

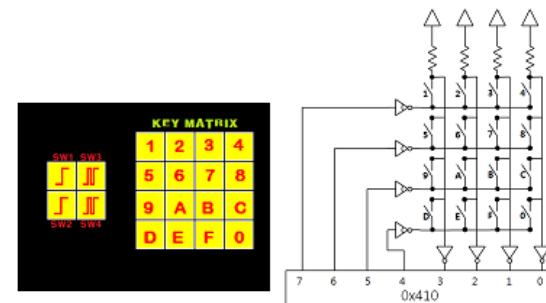


그림 1. 가상 키 그림

그림 2. Key Matrix 모양과 회로

우선 왼쪽의 Push Switch 들은 Nintendo DS의 실제 Key들과 마찬가지로 `readb_virtual_io()`를 이용하여 해당 입력 포트 (0x414 번지, `PUSH_SW`)를 읽음으로써, 해당 스위치를 Bit 단위로 읽을 수 있다. 그림 3과 같이 각 스위치가 비트에 할당되어 있고, 스위치

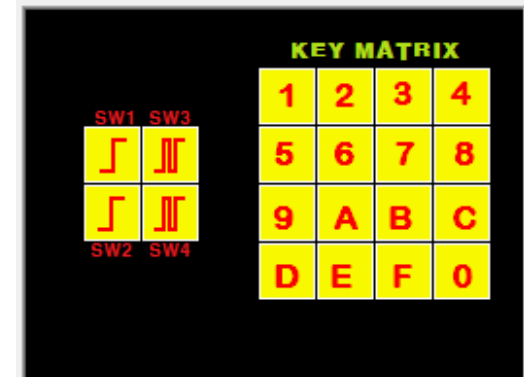
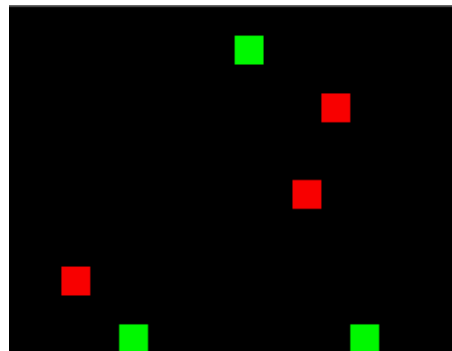
프로젝트 활용 방안 (2 / 3)

- 실험 수업 항목

- Simple I/O 1 – 간단한 입출력
- Simple I/O 2 – State 유지가 필요한 간단한 입출력
- Simple I/O 3 – 복잡한 State Machine이 필요한 입출력
- Key Matrix – Scan에 의한 Key-Matrix 및 7 Segment 구동
- Key Queue – Queue에 의한 Multi-Tasking 입출력 처리
- Ball – LCD에 움직이는 Ball 그리기
- Ball Mutex – RTOS의 Mutex를 이용한 Ball 겹침 해결
- Ball Wall – Mutex를 이용한 Wall과 Ball 움직임

- Term 프로젝트

- LCD, Touch, Sound, 가속센서를 이용한 게임 제작
- (Sound, 가속센서, 2D access sample 제공)



프로젝트 활용 방안 (3 / 3)

- 프로젝트 결과물을 통한 수업 활용 예 (수업 결과물)



커뮤니티 활동

- 버전 관리 및 배포 (github)
 - <https://github.com/hl1itj/nds-ide/>
 - 예전 사이트 : <http://sourceforge.net/projects/nintendo>
- Ubuntu, Windows에 대한 착한 지원
 - Single Installation Package
 - Virtualbox, VMWare용 Preinstalled Virtual Machine 이미지
- Open Source Homebrew 게임 download 연계

장기 발전 방안 (1 / 3)

- 발전 가능성 - 임베디드 시스템 교육 환경 적용
 - 대학, 전문 대학, 전문계 고등학교, 사설 학원에서 활용 가능
 - 현재 한성대학교 컴퓨터공학과 -
'마이크로 프로세서 응용 설계' 과목에 적용하여 수업 중
 - 프로젝트의 결과물로 제공되는 Virtual Device를 바탕으로
 - 교육 실습에 필요한 장치를 S/W로 추가 가능
 - 커뮤니티를 통해 실습용 교재를 공유함으로써,
 - 교육 콘텐츠 증대 및 개선 효과 기대
- ➔ 임베디드 SW 교육에 대한 흥미/성취도 향상 !
- Source Repository를 github 으로 이전 중
 - Visibility 향상, 속도 개선

장기 발전 방안 (2 / 3)

- 발전 가능성 - 개인 개발자 참여 확대

- 현재 유사 사례 많음 (유럽, 미국을 중심으로)

- Homebrew Game 개발 : Nintendo DS, PSP, ...
 - 아직은 개발 도구가 취약 (예, remote debugging 부재)

- 우리 결과물

- 프로젝트의 모든 결과물(SW, HW)은 소스가 공개
 - Eclipse 기반으로 Edit, Build, Remote Debugging 까지 가능
 - 같은 방식으로 다른 게임기 등에도 적용 가능

※ 개인 개발자들은 기존 고가의 교육용 개발 장비 대신

저렴한 상용 임베디드 시스템을 활용하여 기술을 배울 수 있는 기회 !

장기 발전 방안 (3 / 3)

- 발전 가능성 - H/W를 통한 기능 확장

- Serial, USB, GPIO 등으로 외부 장치/회로 연결 가능
- 프로젝트 기본 결과물의 회로 수정으로 하드웨어 추가 가능

(7 Segment LED, Step / DC Motor, Encoder, IR 센서, US 센서, AD/DA Converter 등)

※ 추가적인 하드웨어를 연결할 수 있도록 하고, 간단한 API를 제공함으로써, 기존 고가의 장비들의 기능을 최소한의 비용으로 구현 가능



Q & A

감사합니다.