Embedded System Software

개발 환경 설정

Dept. of Computer Science and Engineering
Sogang University, Seoul, KOREA

실습 환경

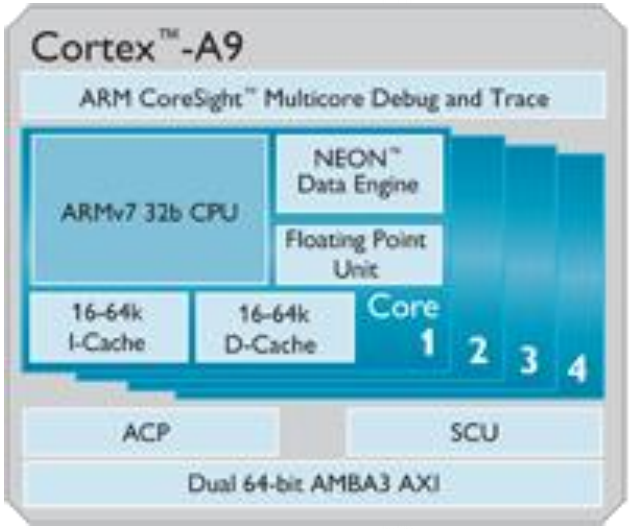
- **Host PC : Linux (Ubuntu 16.04) System on x64 architecture**
- **Target Board : Embedded Linux System on ARM architecture**
- **Host PC에서 제작된 소스 코드를 ARM용 실행 파일로 컴파일하기 위해서는 ARM용 toolchain(컴파일러, 링커, 라이브러리 등)을 설치해야 한다.**

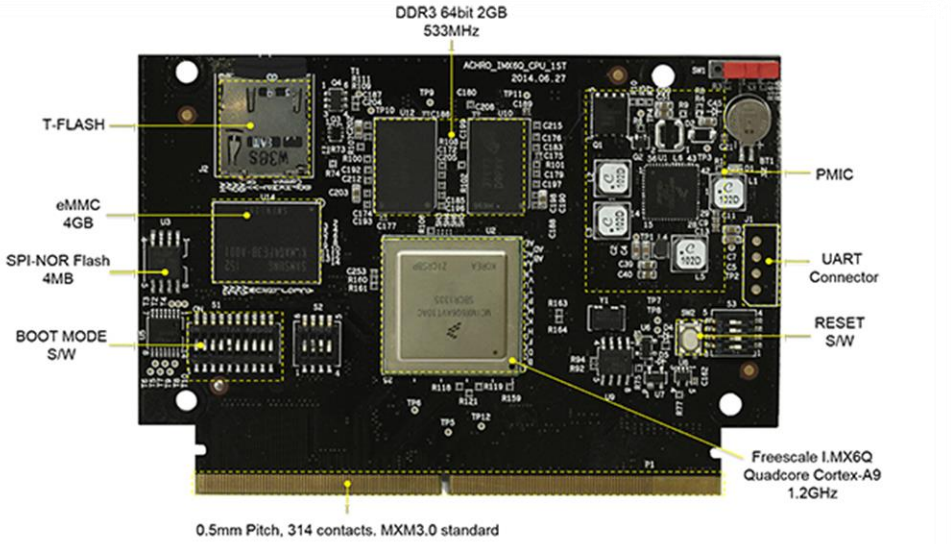
Target Board

- Freescale I.MX6Q Cortex-A9 Dual core
- USB 2.0, SATA 1.0/2.0/3.0 Interface
- MIPI(Mobile Industry Processor Interface)
- EMMC 4.4 / T-FLASH
- WiFi / Bluetooth / GPS
- Ethernet 10/100M bps
- HDMI mirror screen
- 10 points touchscreen
- Flexcan

Target Board

- **ARM Cortex-A9 기반의 듀얼코어 프로세서**

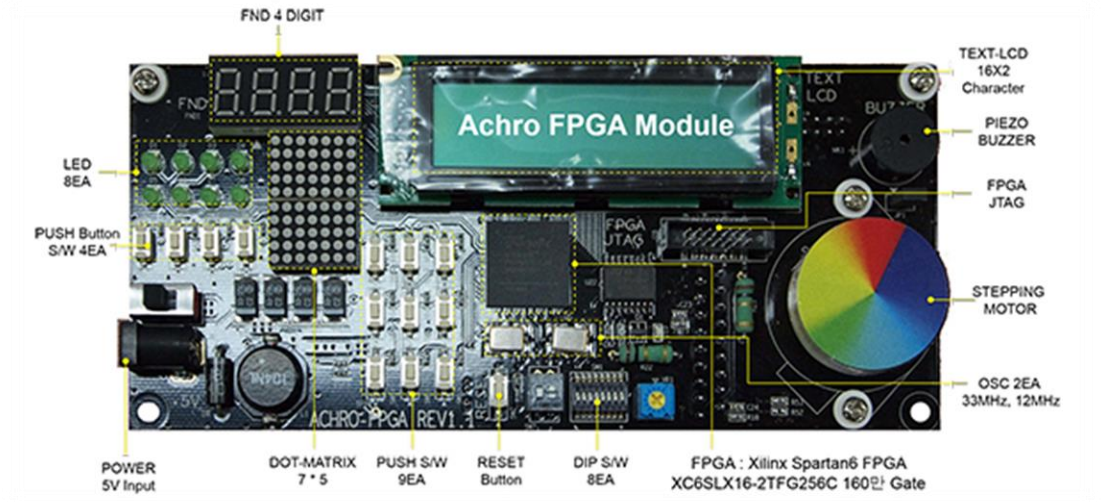




- 멀티코어를 지원 - 최대 쿼드코어까지 구성이 가능
- NEON SMID의 강화 - NEON코어를 직접 연산 유닛에 내장
- 外비순차적 명령어 처리를 도입
- L2 Cache 도입

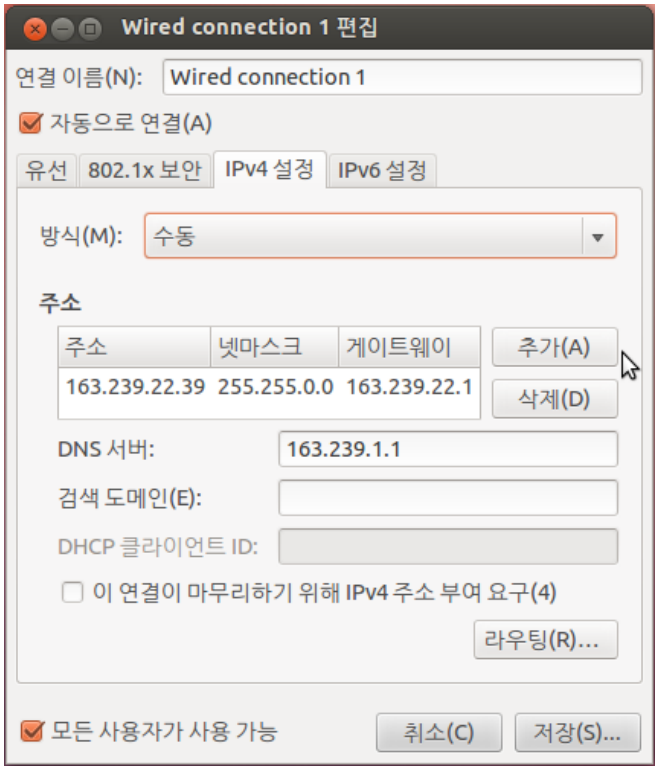
Target Board

- FPGA module (Devices)
  - LED 8
  - DOT-MATRIX
  - FND
  - BUZZER
  - DIP S/W
  - PUSH S/W
  - STEPPING MOTOR



네트워크 설정

- **Edit Connections→ Edit → IPv4 Settings → Method: Manual → Addresses: Add →**
  - **Address:163.239.22.***
  - **Netmask:255.255.0.0**
  - **Gateway:163.239.22.1**
  - **DNS servers:8.8.8.8**



네트워크 설정

- 터미널 접속 후 확인
  - **ping www.google.com**

```
root@ubuntu:~# ping 168.126.63.1
PING 168.126.63.1 (168.126.63.1) 56(84) bytes of data.
64 bytes from 168.126.63.1: icmp_req=1 ttl=58 time=1.69 ms
64 bytes from 168.126.63.1: icmp_req=2 ttl=58 time=2.10 ms
64 bytes from 168.126.63.1: icmp_req=3 ttl=58 time=1.17 ms
```

  - **ifconfig → ifconfig eth0 up**

```
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1d:92:82:89:e6
          inet addr:163.239.22.39  Bcast:163.239.255.255  Mask:255.255.0.0
          inet6 addr: fe80::21d:92ff:fe82:89e6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:72782 errors:0 dropped:512 overruns:0 frame:0
          TX packets:15362 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28345986 (28.3 MB)  TX bytes:1593793 (1.5 MB)
          Interrupt:44

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1854 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1854 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:173667 (173.6 KB)  TX bytes:173667 (173.6 KB)
```

슈퍼유저 계정 생성

- 터미널 실행
- 슈퍼유저 계정 생성

  sudo passwd root

```
root@ubuntu:~# sudo passwd root
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
root@ubuntu:~#
```

- **sudo –s** 를 입력하고 **root**의 암호를 입력하여 **root** 권한 얻기
- **apt-get update**

시작시 root 계정으로 로그인

- vi /usr/share/lightdm/lightdm.conf.d/50-ubuntu.conf
- greeter-show-manual-login=true에 추가

```
1 [SeatDefaults]
2 user-session=ubuntu
3 greeter-show-manual-login=true
4
```

- 저장 후 우분투 로그아웃 후 root 계정으로 로그인
- 에러 발생할 경우
  - vi /root/.profile
  - 맨 밑 줄 # 후, tty –s && mesg n 추가

repository 변경

- VIM 설치 : apt-get install vim
- apt를 통한 다운로드의 경우 초기 경로가 외국 서버로 되어 있어서 속도가 느림 -> ftp.daumkakao.com 으로 변경
- 직접 수정
  - sudo vi /etc/apt/sources.list
  - :%s/kr.archive.ubuntu.com/ftp.daumkakao.com/g
  - :%s/security.Ubuntu.com/ftp.daumkakao.com/g
- apt-get update

개발툴 설치

- **TFTP**
  - **apt-get install tftp tftpd-hpa**
  - **vi /etc/default/tftpd-hpa**
    - 수정

```
# /etc/default/tftpd-hpa

RUN_DAEMON="yes"
OPTIONS="-l -s /tftpboot"

TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
```

  - **mkdir /tftpboot**
  - **sudo service tftpd-hpa restart**

개발툴 설치

- 우분투 방화벽 설정
  - **ufw disable / ufw enable**

  (target 기기와의 통신 위해서는 disable 필요)

- **TFTP test**
  - **echo "TFTP test" > /tftpboot/test.txt**
  - **tftp localhost**
  - **tftp> get test.txt**
  - **tftp> quit**
  - **cat test.txt**

개발툴 설치

- **TFTP(앞에 것으로 안되었을 때)**
  - **apt-get install tftp tftpd xinetd**
  - **vi /etc/xinetd.d/tftp**
    - 수정

```
service tftp
{
    protocol        =   udp
        port        =   69
        socket_type  =   dgram
        wait        =   yes
        user        =   nobody
        server      =   /usr/sbin/in.tftpd
        server_args =    /tftpboot
        disable     =   no
}
```

  - **mkdir /tftpboot**
  - **service xinetd stop**
  - **service xinetd start**

개발툴 설치

- **USB 드라이버 다운로드**
  - **apt-get install libusb-dev**
- **smdk-usbdl이 되지 않을 때**
  - **apt-get install libusb-0.1-4:i386**
- **32bit 호환 패키지 설치**
  - **apt-get install lib32z1**

개발툴 설치

- **Cross Compiler**
  - **dpkg-reconfigure –plow dash**
    - No 선택
  - **mkdir /root/temp**
  - **cp arm-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 /root/temp**
  - **sync**
  - **cd /root/temp**
  - **mkdir /opt/toolchains**
  - **tar jxvf arm-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 –C /opt/toolchains**

개발툴 설치

- **vi /root/.bashrc**
  - 수정

```
 87 # ~/.bash_aliases, instead of adding them here directly.
 88 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
 89
 90 if [ -f ~/.bash_aliases ]; then
 91     . ~/.bash_aliases
 92 fi
 93
 94 # enable programmable completion features (you don't need to enable
 95 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
 96 # sources /etc/bash.bashrc).
 97 #if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
 98 #    . /etc/bash_completion
 99 #fi
100
101 # Cross Compiler
102 export CROSS_COMPILE=arm-none-linux-gnueabi-
103 export PATH=/opt/toolchains/arm-2014.05/bin:$PATH
104 export ARCH=arm
```

- **.bashrc 갱신**
  - source /root/.bashrc

개발툴 설치

- **CrossCompiler 버전 확인**
  - arm-none-linux-gnueabi-gcc –v
- **CrossCompiler 버전이 확인되지 않을 때**
  - $ sudo apt-get install git gnupg flex bison gperf build-essential
  - $ sudo apt-get install zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev
  - $ sudo apt-get install libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386
  - $ sudo apt-get install libglapi-mesa-lts-saucy:i386
- **리눅스 재부팅**

개발툴 설치

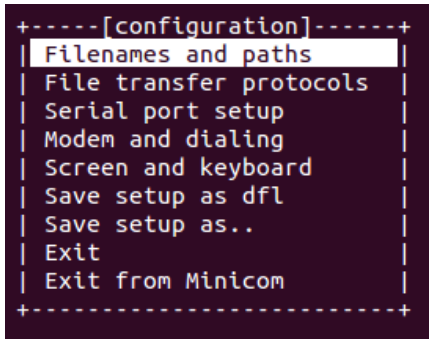
- **CrossCompile**
  - **arm-none-linux-gnueabi-gcc –static –o hello hello.c**
  - **file ./hello**

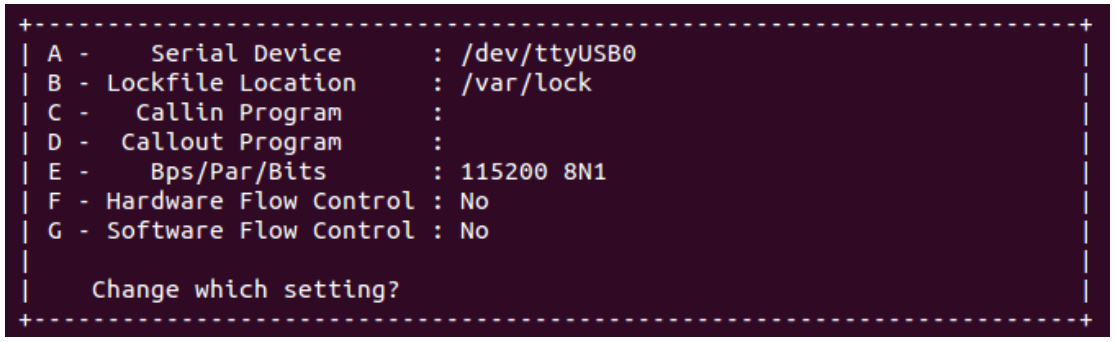
```
root@ubuntu:~# file ./hello
./hello: ELF 32-bit LSB  executable, ARM, EABI5 version 1 (SYSV), statically lin
ked, for GNU/Linux 2.6.16, not stripped
```

```
root@ubuntu:~# arm-none-linux-gnueabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-none-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/opt/toolchains/arm-2014.05/bin/../libexec/gcc/arm-none-linux-gnueabi/4.8.3/lto-wrapper
Target: arm-none-linux-gnueabi
Configured with: /scratch/maciej/arm-linux-2014.05-rel/src/gcc-4.8-2014.05/configure --build=i686-pc-linux-gnu --hos
t=i686-pc-linux-gnu --target=arm-none-linux-gnueabi --enable-threads --disable-libmudflap --disable-libssp --disable
-libstdcxx-pch --enable-extra-sgxxlite-multilibs --with-arch=armv5te --with-gnu-as --with-gnu-ld --with-specs='%{sav
e-temps: -fverbose-asm} %{funwind-tables|fno-unwind-tables|mabi=*|ffreestanding|nostdlib:;:-funwind-tables} -D__CS_S
OURCERYGXX_MAJ__=2014 -D__CS_SOURCERYGXX_MIN__=5 -D__CS_SOURCERYGXX_REV__=29' --enable-languages=c,c++ --enable-shar
ed --enable-lto --enable-symvers=gnu --enable-__cxa_atexit --with-pkgversion='Sourcery CodeBench Lite 2014.05-29' --
with-bugurl=https://sourcery.mentor.com/GNUToolchain/ --disable-nls --prefix=/opt/codesourcery --with-sysroot=/opt/c
odesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/maciej/arm-linux-2014.05-rel/install/opt/codes
ourcery/arm-none-linux-gnueabi/libc --with-gmp=/scratch/maciej/arm-linux-2014.05-rel/obj/pkg-2014.05-29-arm-none-lin
ux-gnueabi/arm-2014.05-29-arm-none-linux-gnueabi.extras/host-libs-i686-pc-linux-gnu/usr --with-mpfr=/scratch/maciej/
arm-linux-2014.05-rel/obj/pkg-2014.05-29-arm-none-linux-gnueabi/arm-2014.05-29-arm-none-linux-gnueabi.extras/host-li
bs-i686-pc-linux-gnu/usr --with-mpc=/scratch/maciej/arm-linux-2014.05-rel/obj/pkg-2014.05-29-arm-none-linux-gnueabi/
arm-2014.05-29-arm-none-linux-gnueabi.extras/host-libs-i686-pc-linux-gnu/usr --with-isl=/scratch/maciej/arm-linux-20
14.05-rel/obj/pkg-2014.05-29-arm-none-linux-gnueabi/arm-2014.05-29-arm-none-linux-gnueabi.extras/host-libs-i686-pc-l
inux-gnu/usr --with-cloog=/scratch/maciej/arm-linux-2014.05-rel/obj/pkg-2014.05-29-arm-none-linux-gnueabi/arm-2014.0
5-29-arm-none-linux-gnueabi.extras/host-libs-i686-pc-linux-gnu/usr --disable-libgomp --disable-libitm --enable-libat
omic --disable-libssp --enable-poison-system-directories --with-build-time-tools=/scratch/maciej/arm-linux-2014.05-r
el/install/opt/codesourcery/arm-none-linux-gnueabi/bin --with-build-time-tools=/scratch/maciej/arm-linux-2014.05-rel
/install/opt/codesourcery/arm-none-linux-gnueabi/bin SED=sed
Thread model: posix
gcc version 4.8.3 20140320 (prerelease) (Sourcery CodeBench Lite 2014.05-29)
```

개발툴 설치

- **minicom**
  - **apt-get install minicom**
  - **minicom 환경 설정**
    - minicom –s
    - Serial port setup
    - 통신포트: /dev/ttyUSB0(/dev/ttyUSB0)(a)
    - 통신속도: 115200 8N1(e → q)
    - flow control: hard → no / soft → no





  - save setup as dfl 선택 후 exit

파일 전송

- **TFTP**
  - **보드 ip setting (보드에서)**
    - ifconfig eth0 TARGETIPADDRESS
    - ifconfig eth0 up
    - ping HOSTIPADDRESS

```
# ifconfig eth0 163.239.22.9
# ifconfig eth0 up
# ping 163.239.22.39
PING 163.239.22.39 (163.239.22.39) 56(84) bytes of data.
64 bytes from 163.239.22.39: icmp_seq=1 ttl=64 time=1.24 ms
64 bytes from 163.239.22.39: icmp_seq=2 ttl=64 time=0.505 ms
```

  - **파일 전송 (보드에서)**
    - tftp –r FILENAME –g HOSTIPADDRESS
    - 파일은 host pc의 /tftpboot 폴더안에 존재해야한다
    - 보드에서의 현재 디렉토리가 /이면 안됨.
    - /data/local/tmp 폴더 추천(data가 사라지지 않음)
    - 실행파일 전송 후, chmod 수정 필요할 수 있음