SevenCore

보고서 / 품의서 / 제안서

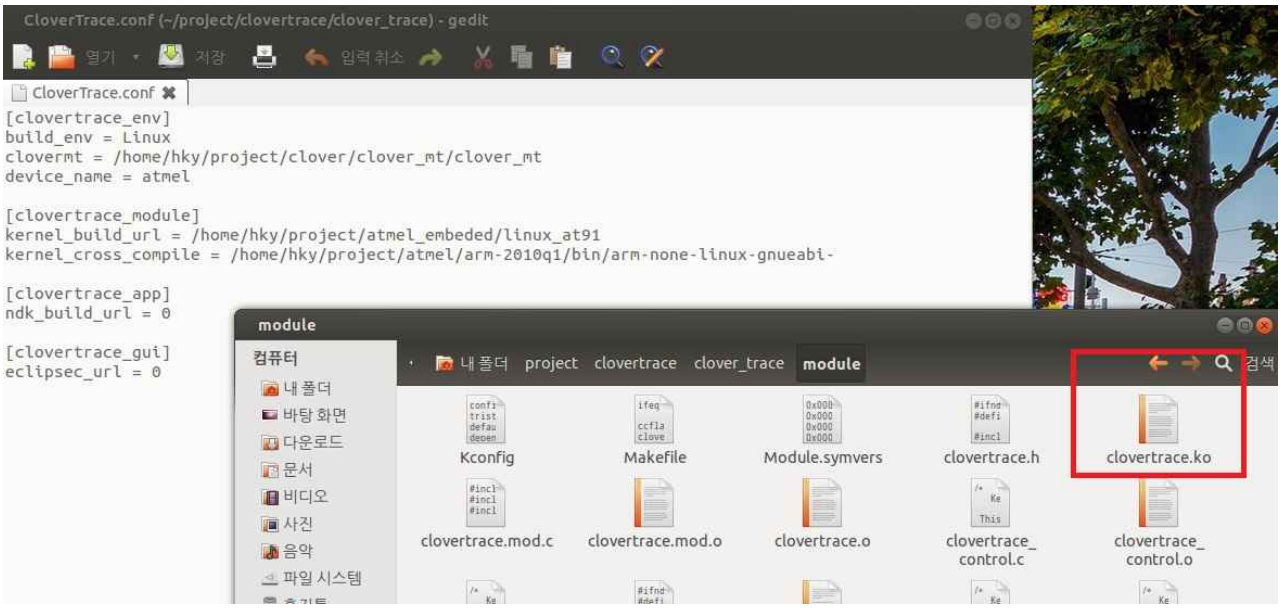
작성자	하권용
작성 일자	2014. 04. 28 (V 1.0)
분 류	■ 기술 □ 제안서 □ 의견제출 □ 구매요청 □ 기타 ■□
제 목	How to use CloverTrace in vanilla linux

Summary

SKT Heart – FPGA용 바닐라 리눅스에서 clovertrace를 사용하기 위하여 작업한 내용들을 설명한 기술문서이다. 즉, 안드로이드가 없는 환경에서 clovertrace를 수행하기 위한 방법을 기술하였다.

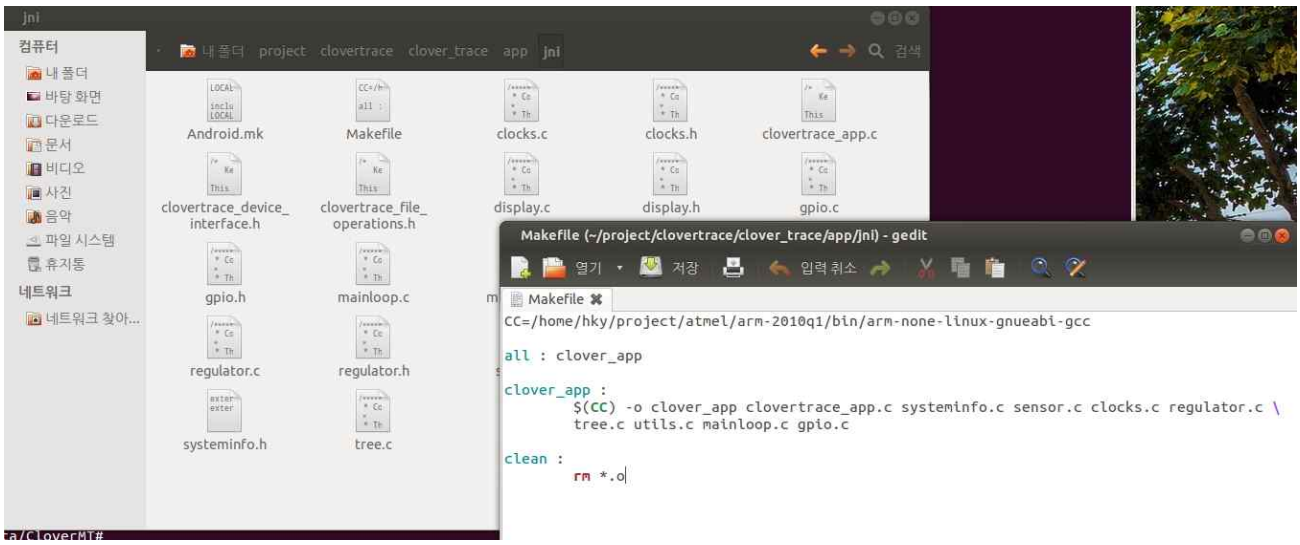
1. CloverTrace 관련 파일 준비

1-1. 해당 리눅스 커널로 빌드된 clovertrace.ko 모듈파일



1-2. Cross Compile된 clovertrace app

- Clovertrace App은 결과파일들(kernel.txt, thread.txt ...)을 생성(dump)하는 역할을 한다.
- 기존엔 NDK를 이용하여 컴파일하고 adb를 통해 동작하였으나
  이전 방식을 사용할 수 없으므로 포팅한다.
  - <clovertrace>/app/jni 폴더에 Maefile 작성

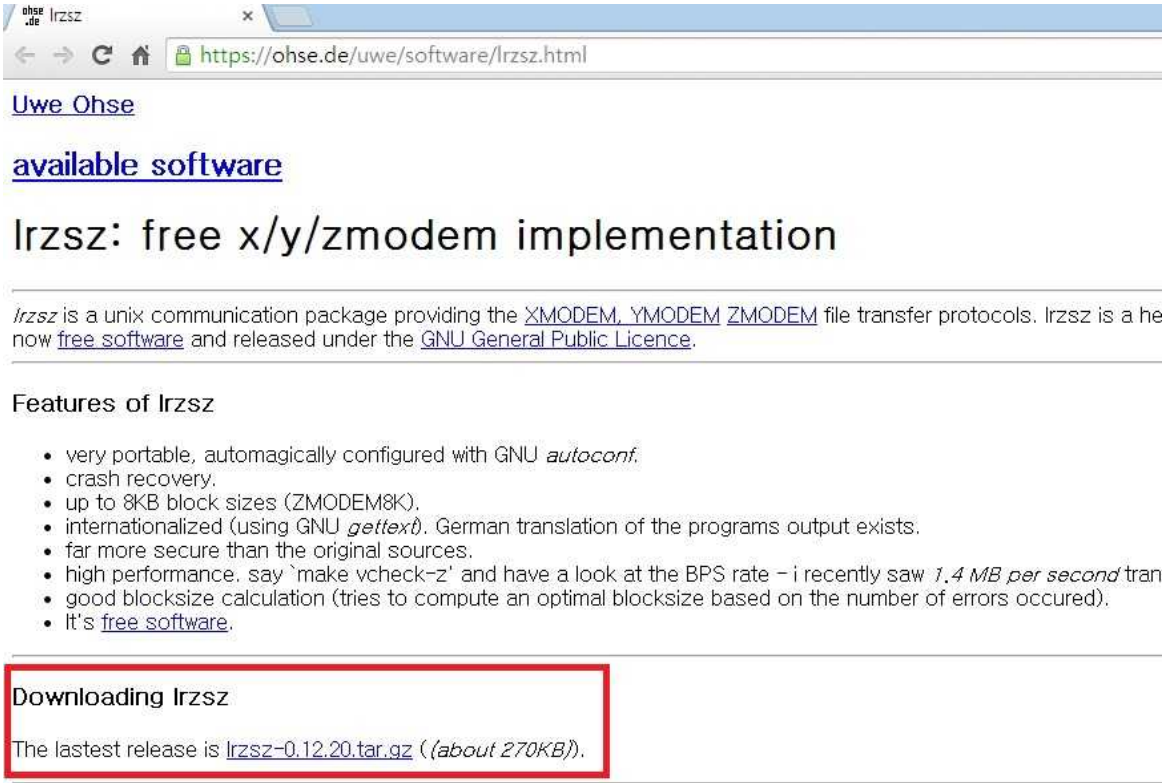



```
CC=/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc

all : clover_app

clover_app :
        $(CC) -o clover_app clovertrace_app.c systeminfo.c sensor.c clocks.c regulator.c ₩
        tree.c utils.c mainloop.c gpio.c

clean :
        rm *.o
```

- make 수행하면 clover_app 바이너리 파일이 생성된다.

1-3. Lrzsz

- Lrzsz는 터미널을 통한 파일전송을 가능하게 해 주는 통신 프로토콜 패키지이다.
- https://ohse.de/uwe/software/lrzsz.html
- 위 사이트에서 lrzsz를 다운받은 후 압축을 해제한다.



- CC=<툴체인path> ./configure --host=arm-none-linux-gnueabi-gcc
  - CC=/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc ./configure --host=arm-none-linux-gnueabi-gcc

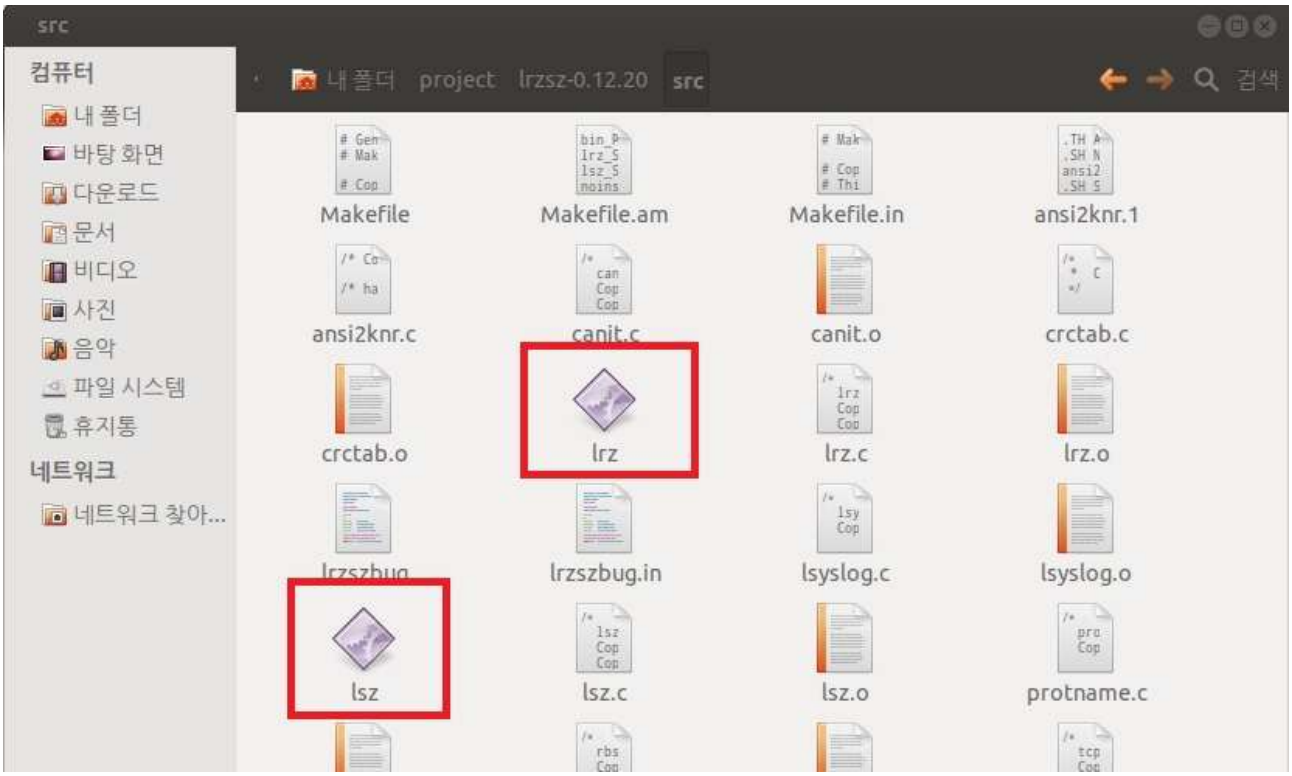
```
checking whether NLS is requested... yes
checking whether included gettext is requested... no
checking for libintl.h... yes
checking for gettext in libc... yes
checking for msgfmt... no
checking whether catgets can be used... no
checking for msgfmt... (cached) no
checking for gmsgfmt... no
checking for xgettext... :
checking for catalogs to be installed...  de
updating cache ./config.cache
creating ./config.status
creating Makefile
creating intl/Makefile
creating lib/Makefile
creating testsuite/Makefile
creating man/Makefile
creating po/Makefile.in
creating src/Makefile
creating debian/rules
creating Specfile
creating systype
creating src/lrzszbug
creating config.h
linking ./intl/libgettext.h to intl/libintl.h
hky@hky-All-Series:~/project/lrzsz-0.12.20$
```

- make

```
hky@hky-All-Series: ~/project/lrzsz-0.12.20
usr/local/share/locale\" -I.. -I. -I../src -I../intl -I../lib  -g -O2 -c zm.c
/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc -DNFGVMIN -DHAVE_CONFIG_H -DLOCALEDIR
usr/local/share/locale\" -I.. -I. -I../src -I../intl -I../lib  -g -O2 -c protname.c
/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc -DNFGVMIN -DHAVE_CONFIG_H -DLOCALEDIR
usr/local/share/locale\" -I.. -I. -I../src -I../intl -I../lib  -g -O2 -c tcp.c
/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc -DNFGVMIN -DHAVE_CONFIG_H -DLOCALEDIR
usr/local/share/locale\" -I.. -I. -I../src -I../intl -I../lib  -g -O2 -c lsyslog.c
lsyslog.c: In function 'lsyslog':
lsyslog.c:67: warning: incompatible implicit declaration of built-in function 'sprintf'
/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc -DNFGVMIN -DHAVE_CONFIG_H -DLOCALEDIR
usr/local/share/locale\" -I.. -I. -I../src -I../intl -I../lib  -g -O2 -c canit.c
/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc -g -O2  -o lrz  lrz.o timing.o zperr.
eadline.o crctab.o rbsb.o zm.o protname.o tcp.o lsyslog.o canit.o ../lib/libzmodem.a ../intl/libintl.a
l
/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc -DNFGVMIN -DHAVE_CONFIG_H -DLOCALEDIR
usr/local/share/locale\" -I.. -I. -I../src -I../intl -I../lib  -g -O2 -c lsz.c
/home/hky/project/atmel/arm-2010q1/bin/arm-none-linux-gnueabi-gcc -g -O2  -o lsz  lsz.o timing.o zperr.
eadline.o crctab.o rbsb.o zm.o protname.o tcp.o lsyslog.o canit.o ../lib/libzmodem.a ../intl/libintl.a
l
make[2]: Leaving directory `/home/hky/project/lrzsz-0.12.20/src'
Making all in po
make[2]: Entering directory `/home/hky/project/lrzsz-0.12.20/po'
make[2]: Leaving directory `/home/hky/project/lrzsz-0.12.20/po'
Making all in man
make[2]: Entering directory `/home/hky/project/lrzsz-0.12.20/man'
make[2]: `all'를 위해 할 일이 없습니다
make[2]: Leaving directory `/home/hky/project/lrzsz-0.12.20/man'
Making all in testsuite
make[2]: Entering directory `/home/hky/project/lrzsz-0.12.20/testsuite'
make[2]: `all'를 위해 할 일이 없습니다
make[2]: Leaving directory `/home/hky/project/lrzsz-0.12.20/testsuite'
make[1]: Leaving directory `/home/hky/project/lrzsz-0.12.20'
hky@hky-All-Series:~/project/lrzsz-0.12.20$
```

- 빌드가 완료되면 /src 폴더에 lsz, lrz 두 바이너리 파일이 생성된다.

```
hky@hky-All-Series:~/project/lrzsz-0.12.20$ cd src
hky@hky-All-Series:~/project/lrzsz-0.12.20/src$ ls
Makefile      ansi2knr.c  crctab.o  lrzszbug     lsz         protname.o  tcp.o     zglobal.h  zperr.c
Makefile.am   canit.c     lrz       lrzszbug.in  lsz.c       rbsb.c      timing.c  zm.c       zperr.o
Makefile.in   canit.o     lrz.c     lsyslog.c    lsz.o       rbsb.o      timing.h  zm.o       zreadline.c
ansi2knr.1    crctab.c    lrz.o     lsyslog.o    protname.c  tcp.c       timing.o  zmodem.h   zreadline.o
hky@hky-All-Series:~/project/lrzsz-0.12.20/src$ file lsz
lsz: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux
 2.6.16, not stripped
hky@hky-All-Series:~/project/lrzsz-0.12.20/src$ file lrz
lrz: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux
 2.6.16, not stripped
hky@hky-All-Series:~/project/lrzsz-0.12.20/src$
```



2. Clovertrace 관련 파일 삽입

1-1. rootfs ramdisk 이미지를 마운트하여 위에서 준비한 파일들을 copy한다.

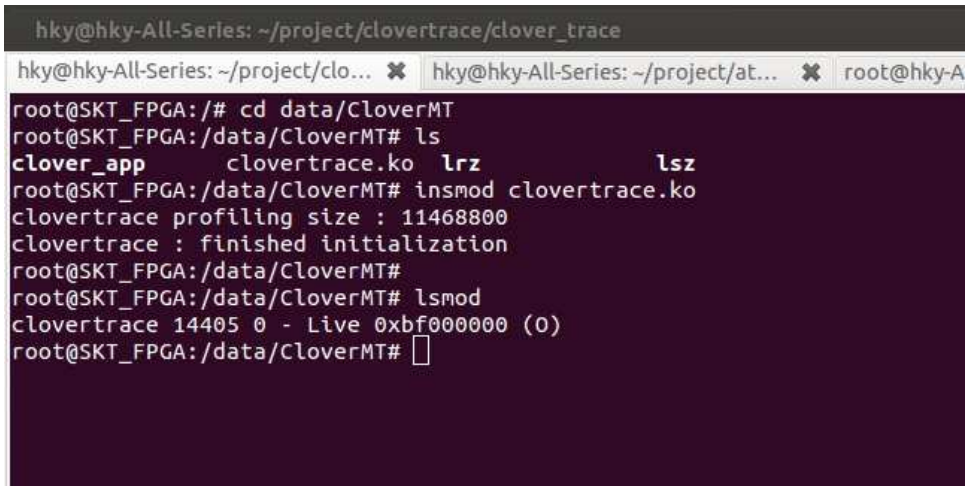
■ /data/CloverMT 폴더에서 작업함.

1-2. Ramdisk 이미지 언마운트 후 보드로 다운로드.

3. CloverTrace 수행하기

여기서부터는 minicom 화면입니다.

1-1. 모듈 인스톨



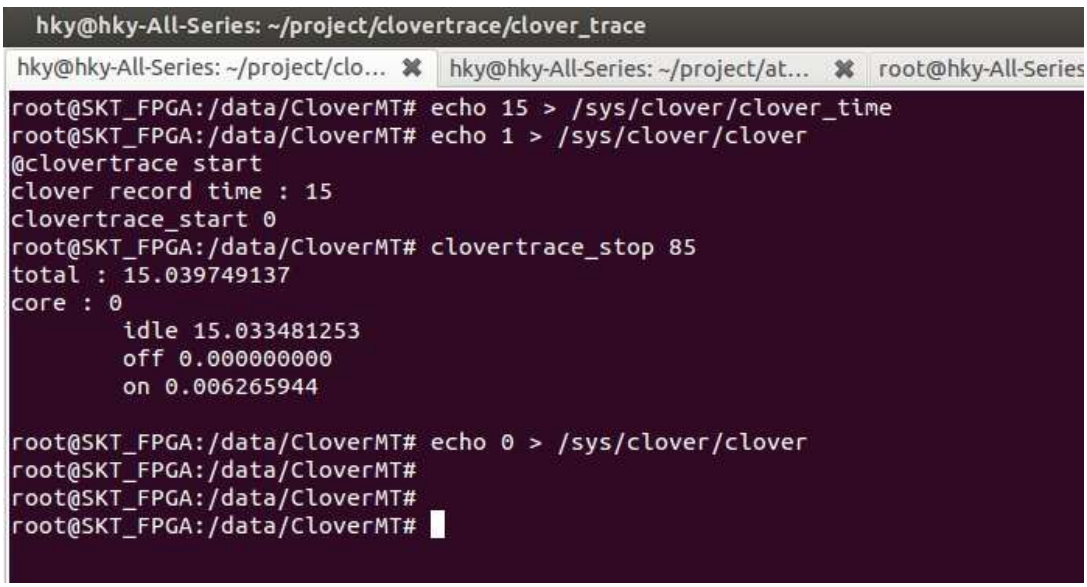
1-2. Clovertrace 타이머에 수행시간(15초) 전달

- echo 15 > /sys/clover/clover_time

1-3. Clovertrace 수행시작 신호 전달

- echo 1 > /sys/clover/clover

1-4. 원하는 작업 수행



1-5. 15초 후 trace가 완료되면 CPU 수행데이터가 출력되고, 수행완료 신호를 입력해준다.

- echo 0 > /sys/clover/clover

4. CloverTrece 결과 확인

1-1. 결과파일 추출

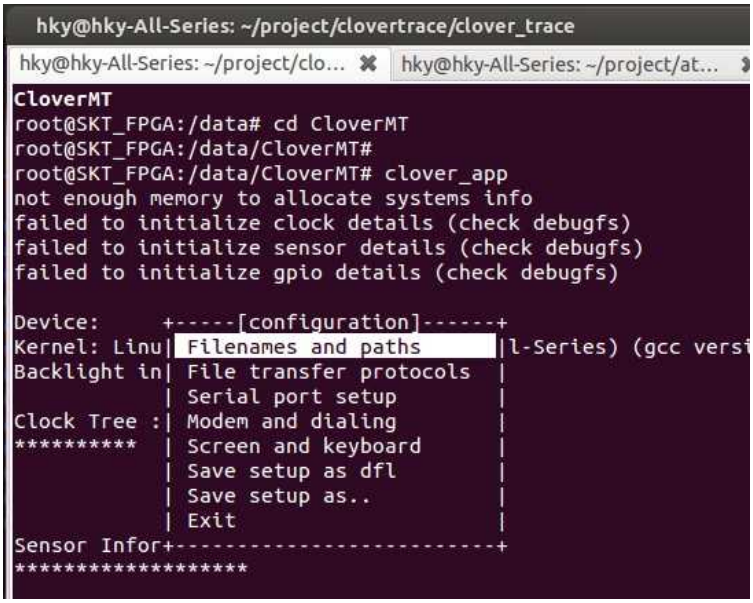
- clover_app 실행

```
root@SKT_FPGA:/data/CloverMT#
root@SKT_FPGA:/data/CloverMT# cd ..
root@SKT_FPGA:/data# ls
CloverMT
root@SKT_FPGA:/data# cd CloverMT
root@SKT_FPGA:/data/CloverMT#
root@SKT_FPGA:/data/CloverMT# clover_app
not enough memory to allocate systems info
failed to initialize clock details (check debugfs)
failed to initialize sensor details (check debugfs)
failed to initialize gpio details (check debugfs)

Device:
Kernel: Linux version 3.6.9 (root@hky-All-Series) (gcc version 4.4.1 (Sourcery G++
Backlight information

Clock Tree :
**********



Sensor Information:
******************


Gpio Tree :
***********
/



Regulator Information:
**********************

clovertrace dump done...
root@SKT_FPGA:/data/CloverMT#
root@SKT_FPGA:/data/CloverMT# cd ..
root@SKT_FPGA:/data# ls
CloverMT            gpioinfo.json       regulatorinfo.json  systeminfo.csv
clockinfo.json      kernel.txt          sensorinfo.json     thread.txt
root@SKT_FPGA:/data#
```

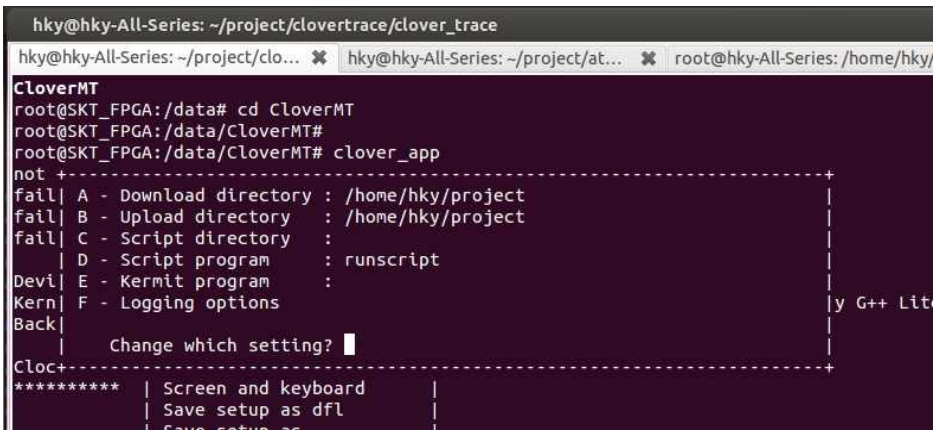
- /data/폴더에 trace 결과파일들이 생성됨.

1-2. 결과파일을 Host로 가져오기 위해 minicom에서 path 설정

- Ctrl+A, O



- Download/Upload directory 설정



1-3. lsz <filename>을 입력하면 해당 파일이 직전에 설정한 폴더로 다운로드 된다.

```
        +----------[zmodem download - Press CTRL-C to quit]-----------+
Device:  |                                                            |
Kernel: L|Receiving: kernel.txt                                       |urcery
Backlight|Bytes received:    2984/   2984   BPS:11107                 |
         |                                                            |
Clock Tre|Transfer complete                                          |
*********|                                                            |
         | READY: press any key to continue...                        |
         +------------------------------------------------------------+

Sensor Information:
*******************


Gpio Tree :
***********
/



Regulator Information:
**********************

clovertrace dump done...
root@SKT_FPGA:/data/CloverMT#
root@SKT_FPGA:/data/CloverMT# cd ..
root@SKT_FPGA:/data# ls
CloverMT            gpioinfo.json       regulatorinfo.json  systeminfo.csv
clockinfo.json      kernel.txt          sensorinfo.json     thread.txt
root@SKT_FPGA:/data# cd CloverMT
root@SKT_FPGA:/data/CloverMT# ls
clover_app      clovertrace.ko  lrz             lsz
root@SKT_FPGA:/data/CloverMT#
root@SKT_FPGA:/data/CloverMT# lsz /data/kernel.txt
**B00
```

1-4. 결과파일을 열어 내용 확인.

kernel.txt (~/project) - gedit

kernel.txt

```
1500    0       0       1170959846      534954521       0       0
1504    377     0       1170959846      534956339       2       0
1507    377     0       1170959846      534957309       8       0
1500    3       0       1170959846      535681309       0       0
1500    0       0       1170959846      535711855       0       0
1500    236     0       1170959846      598619861       0       0
1500    0       0       1170959846      598710649       0       0
1500    236     0       1170959846      658670897       0       0
1500    0       0       1170959846      658726533       0       0
1500    236     0       1170959846      858861220       0       0
1500    0       0       1170959846      858902190       0       0
1500    236     0       1170959847      59049240        0       0
1500    0       0       1170959847      59087422        0       0
1500    236     0       1170959847      259242109       0       0
1500    0       0       1170959847      259278715       0       0
1500    236     0       1170959847      459434614       0       0
1500    0       0       1170959847      459468917       0       0
1500    236     0       1170959847      659622997       0       0
1500    0       0       1170959847      659660573       0       0
1500    236     0       1170959847      859816472       0       0
1500    0       0       1170959847      859852593       0       0
1500    236     0       1170959848      60006067        0       0
```

thread.txt (~/project) - gedit

thread.txt

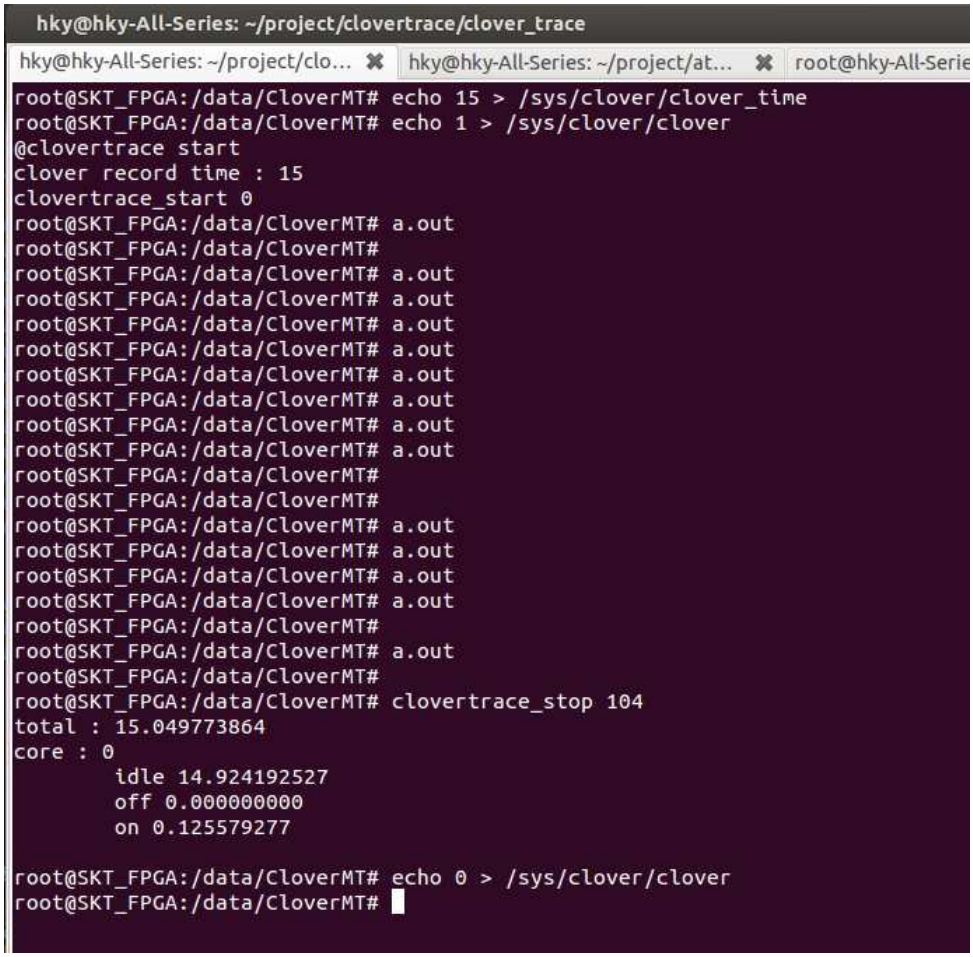
```
*version
1
data-file-overflow=false
clock=thread-cpu
elapsed-time-usec=269540
num-method-calls=4164
clock-call-overhead-nsec=26001
vm=dalvik
*cpus
1
*freqs

*threads
1       1       ini
2       2       kthread
3       3       ksoftirqd/
4       4       kworker/0:
5       5       kworker/0:0
6       6       kworker/u:
7       7       kworker/u:0
8       8       khelpe
9       9       kdevtmpf
```

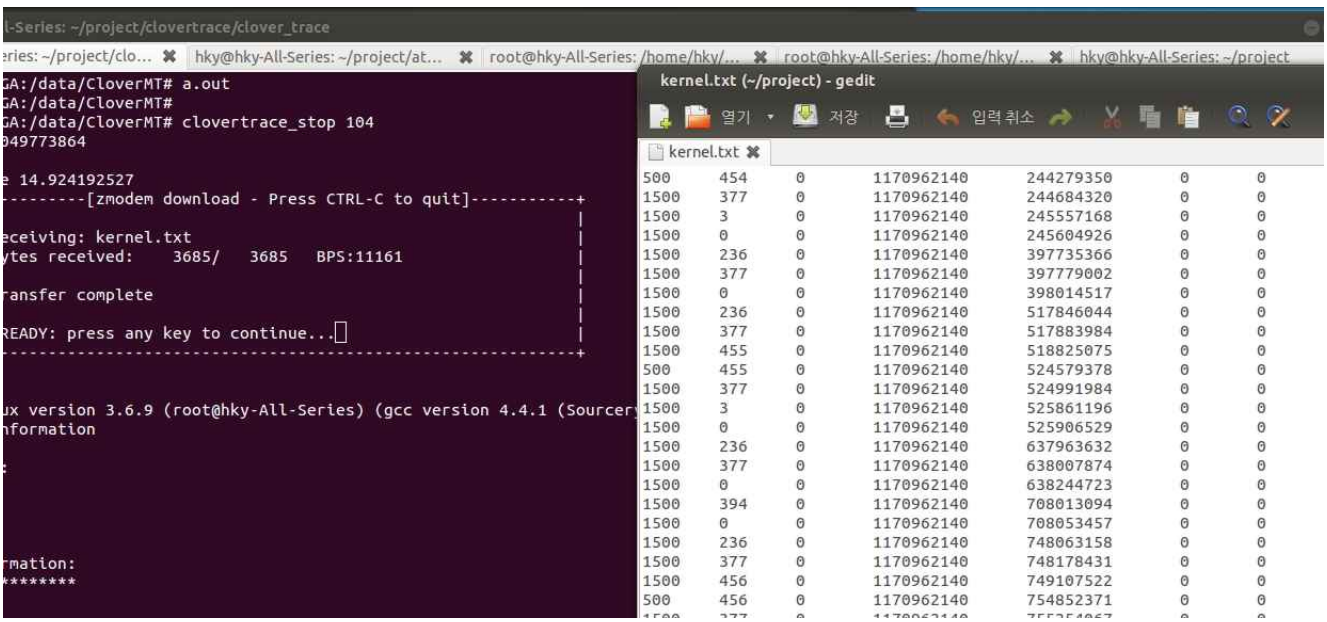
1-5. 결과 증명

■ 테스트 파일

```
#include <unistd.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>

#define FRAME_PATH "/sys/clover/clover_framework2"

void main()
{
	int id = 500;
	long value1;
	long value2;
	int ret = -1;
	int nwr = -1;
	int fd = -1;
	char buf[20];

	fd = open(FRAME_PATH, O_RDWR);
	nwr = sprintf(buf, "%ld ₩n", id);
	ret = write(fd, buf, nwr);
	close(fd);
}
```

■ 실행 시 (500, 0, 0)값을 찍음.

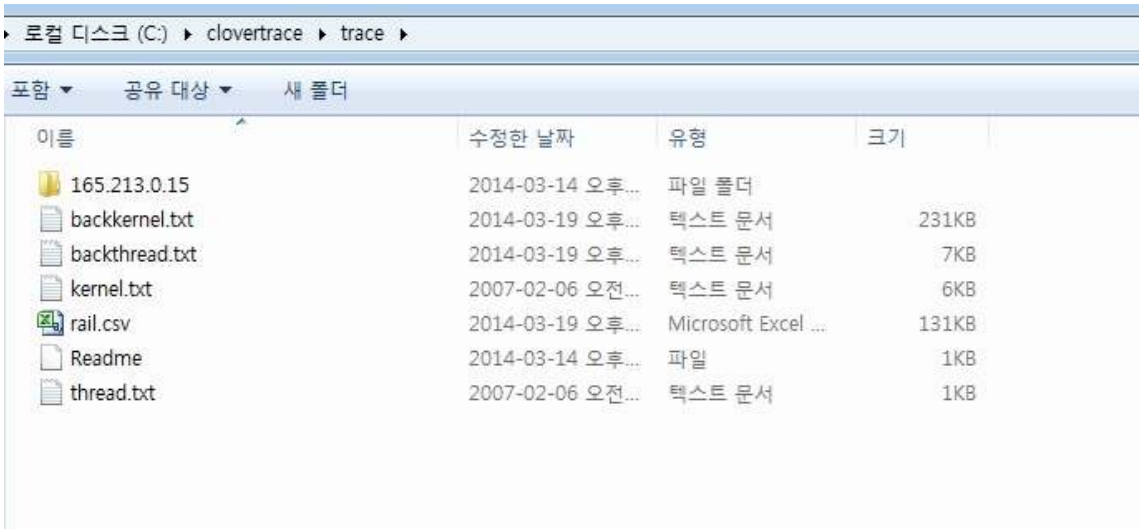




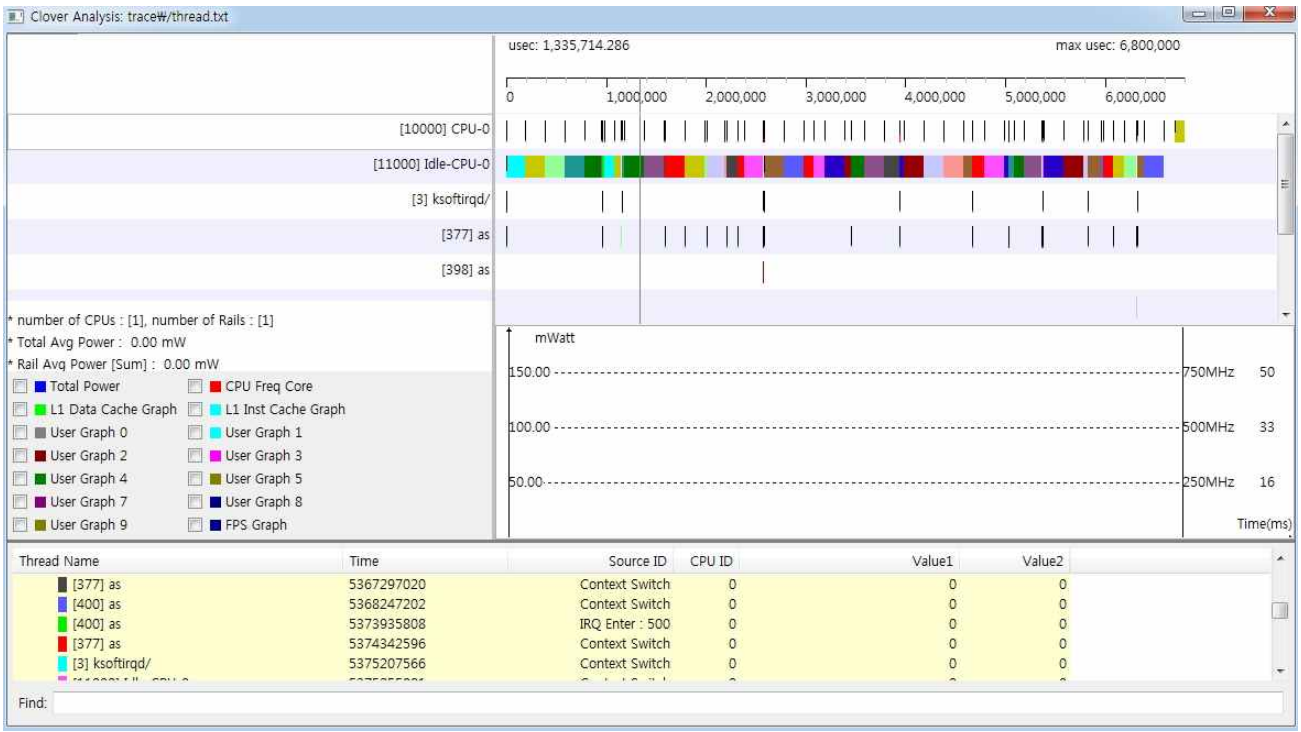
■ Kernel.txt 파일에서 id - 500으로 찍혀나옴.

1-6. GUI로 보기

■ (MS 윈도우) kernel.txt, thread.txt 파일을 Clovertrace/trace 폴더로 옮긴다.

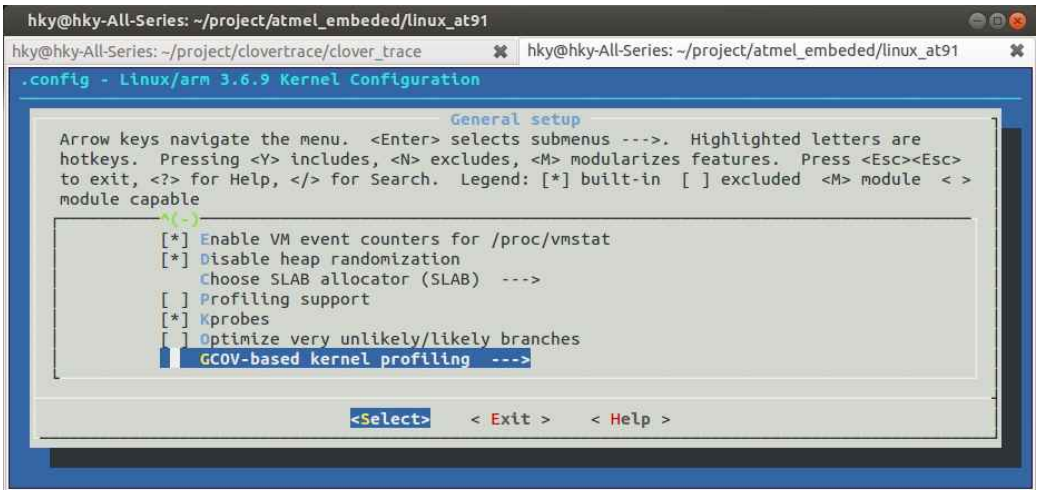


■ cmd창에서 run_gui.bat

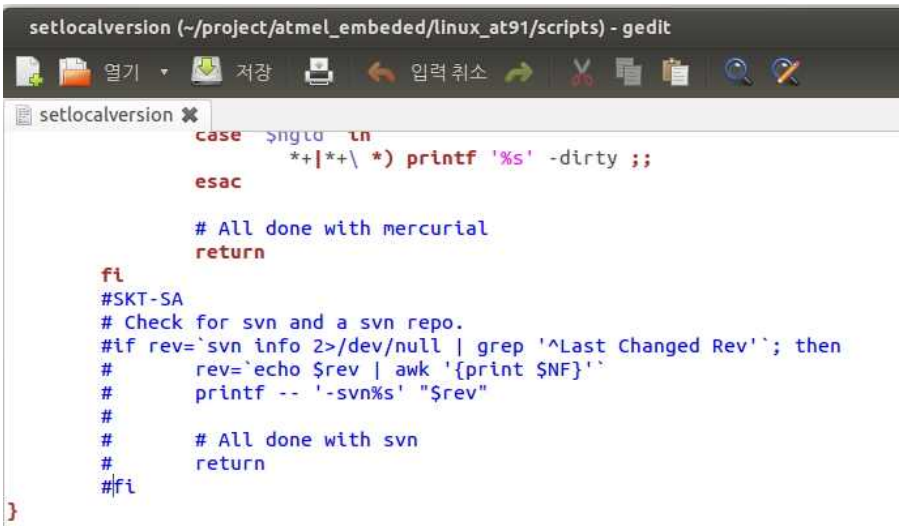


5. Clovertrace 모듈 관련 기타사항

1-1. 모듈이 동작하기 위해선 커널에서 kprobe 옵션 체크해야함.



1-2. 커널의 version masic string 불일치시 모듈 로딩불가 - svn부분 주석처리



1-3. Wake_up_new_task, finish_task_switch 심볼 확인