

Izveštaj za projekat iz predmeta Projektovanje Digitalnih sistema

Aleksandar Vuković 2018/3034

Poslednji put promenjeno: 7. septembar 2020

1 Opis posla

Potrebno je realizovati softversku podršku za pristup mernoj stanici preko mreže. Merna stanica je implementirana na razvojnom sistemu koji je baziran na modifikovanoj Versatile Express V2P CA9 ploči i sadrži dva senzora, memorijski mapirani senzor za merenje napona i I2C senzor za merenje temperature. Da bi se omogućio pristup tim senzorima potrebno je prvo portovati U-Boot i Linuks operativni sistem na dostupni razvojni sistem i kreirati odgovarajući minimalni root fajlsistem. Nakon toga potrebno je razviti odgovarajući drajver za memorijski mapirani senzor. Takođe, potrebno je razviti korisničku aplikaciju koja će:

- komunicirati sa razvijenim drajverom memorijski mapiranog senzora
- komunicirati sa I2C senzorom putem *i2c-dev* iz korisničkog prostora
- izmerene podatke smeštati u bazu podataka

Dostupna je web aplikacija koju treba kopirati na zadatu lokaciju na root fajlsistemu. Web aplikacija prilikom pristupa web client-a čita podatke iz baze i prikazuje ih u vidu grafika.

2 Početno okruženje i priprema

Početno okruženje se dobija pokretanjem skripte pod nazivom *pds10-priprema.sh* u željenom direktorijumu na host mašini gde će se nalaziti svi alati i programi od značaja. Ovim dobijamo direktorijume **Linux kernel**, **u-boot**, **qemu**, **toolchain**, **struktura fajl sistema**, koje je potrebno instalirati i konfigurisati.

Prekopirati *pds03.patch* u qemu direktorijum sa sors kodom, i primeniti *patch*:

```
terminal
```

```
$ patch -p1 < pds03.patch
```

Patch-ovan sors kod u QEMU-u.

```
terminal
```

```
patching file hw/arm/vexpress.c
patching file hw/misc/Makefile.objs
patching file hw/misc/pds-i2csens.c
patching file hw/misc/pds-mmsens.c
```

Potrebno ući u **../bin/arm/** i ponovo make-ovati qemu, možda prvo očistiti prethodne fajlove.

terminal

```
$ mkdir -p bin/arm
$ cd bin/arm
$ make clean
$ ../../configure -target-list=arm-softmmu -enable-sdl -with-sdlabi=2.0
-enable-tools -enable-fdt -enable-libnfs -audio-driv-list=alsa
$ make
```

Neophodno je promeniti *environment* varijablu, pre korišćenja qemu-a.

terminal

```
$ export PATH = ~/Downloads/qemu/bin/arm/arm-softmmu:$PATH
```

Qemu je sada podešen za traženu ploču, i sad je potrebno odgovarajuće promene primeniti na module drajvere, kernel-a, u-boot-a ...

3 Bootloader u-boot podešavanja

Pre instaliranja *bootloader*-a neophodno je prolagoditi ga rayvojnom okruženju/ploči. Što znači da ima svoj defconfig fajl unutar direktorijum `/u-boot-*/configs/` koji se dobija kopiranjem `vexpress_ca9x4_defconfig` u `vexpress_ca9x4_pds03_defconfig` i izmeniti ga dodavanjem sledećih linija:

```
./u-boot-*/configs/vexpress_ca9x4_pds03_defconfig
CONFIG_TARGET_VEXPRESS_CA9X4_PDS03=y
CONFIG_SYS_TEXT_BASE=0x80800000
...
```

Nova ploča za razvoj bi trebalo da ima zaseban *header* fajl `vexpress_ca9x4_pds03.h`. Nov fajl je dobijen kopiranjem koda iz `./include/configs/vexpress_common.h` i menjanjem narednih makroa.

```
./include/configs/vexpress_ca9x4_pds03.h
#define V2M_BASE      0x80000000
#define V2M_MMCI      (V2M_PA_CS7 + V2M_PERIPH_OFFSET(5))
#define V2M_UART0     (V2M_PA_CS7 + V2M_PERIPH_OFFSET(14))
#define V2M_TIMER01    (V2M_PA_CS7 + V2M_PERIPH_OFFSET(21))
...
```

Potrebno je i konfigurisati pre instalacije u-boot-a fajl `../board/arm ltd/vexpress_ca9_pds03` dobijen kopiranjem `../board/arm ltd/vexpress_ca9`.

Naredne linije kod su dodate u fajlove `board/arm ltd/vexpress_ca9_pds03/Kconfig` i `board/arm ltd/vexpress_ca9_pds03/Makefile`:

```
./board/armltd/vexpress_ca9_pds03/Kconfig
```

```
if TARGET_VEXPRESS_CA9X4_PDS03

config SYS_BOARD
    default "vexpress"

config SYS_VENDOR
    default "armltd"

config SYS_CONFIG_NAME
    default "vexpress_ca9x4_pds03"

endif

...
```

```
./board/armltd/vexpress_ca9_pds03/Makefile
```

```
obj-$(CONFIG_TARGET_VEXPRESS_CA9X4_PDS03) += vexpress_tc2.o
```

U fajl arch/arm/Kconfig dodati:

```
./arch/arm/Kconfig
```

```
config TARGET_VEXPRESS_CA9X4_PDS03
    bool "Support vexpress_ca9_pds03"
    select CPU_V7A
    select PL011_SERIAL

source "board/armltd/vexpress_ca9_pds03/Kconfig"
```

terminal

```
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- \
vexpress_ca9x4_pds03_defconfig
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
```

Za proveru stare ploče ...

terminal

```
$ make CROSS_COMPILE=arm-linux-gnueabi- vexpress_ca9x4_defconfig
$ make CROSS_COMPILE=arm-linux-gnueabi-
```

Da bi već kompajlirani u-boot mogao ponovo da se kompajlira za drugu arhitekturu.

Fajlovi ./arch/arm/boot/dts/vexpress-v2m-ca9-pds03.dts i ./arch/arm/boot/dts/vexpress-v2p-ca9-pds03.dts su dobijeni kopiranjem vexpress-v2m.dts i vexpress-v2p-ca9.dts i dodavanjem sledećeg koda.

```
./arch/arm/boot/dts/vexpress-v2m-ca9-pds03.dtsi
```

```
pds03_mmsensor@18000 {
    compatible = "pds03,mmsensor";
    reg = <0x18000 0x1000>;
    interrupts = <27>;
};

pds03_i2c: i2c@8000 {
    compatible = "arm,versatile-i2c";
    reg = <0x8000 0x1000>;
    #address-cells = <1>;
    #size-cells = <0>;
};

...
```

Dodati u `./arch/arm/boot/dts/vexpress-v2p-ca9-pds03.dts`:

```
Makefile
```

```
#include "vexpress-v2m-pds03.dtsi"

...
```

A u Makefile-u dodati novu ploču za koju treba napraviti dtb fajl (device tree za novu razvojnu ploču):

```
Makefile
```

```
dtb-$(CONFIG_ARCH_VEXPRESS) += \
    vexpress-v2p-ca5s.dtb \
    vexpress-v2p-ca9.dtb \
    vexpress-v2p-pds03.dtb \ % (samo ovo se dodaje)
    vexpress-v2p-ca15-tc1.dtb \
    vexpress-v2p-ca15_a7.dtb

...
```

Potrebno je omogućiti I2C podršku. To se radi tako što se omogući (linux) menuconfig -> Device Drivers -> I2C Support -> I2C Hardware Bus Support -> Versatile Arm Realview I2C bus support.

4 Kernel, moduli drajvera, rootfs

Za drajver su implementirane funkcije čitanja, pisanja, pokretanja, zaustavljanja, preko sysfs fajlova. Kod za drajver i podešavanja potrebno je ubaciti u `drivers/char/` direktorijum.

Napisati kod za modul drajvera (demo) ...
za kompajliranje

```
./rootfs/drivers/char/Makefile
```

```
obj-$(CONFIG_PDS03_MMSENSOR) += mmsensor_03.o

...
```

```
./rootfs/drivers/char/Kconfig

config PDS03_MMSENSOR
    tristate "PDS03 Memory Mapped Sensor driver"
    depends on ARCH_VEXPRESS
    default y
    help
        This is driver for Memory Mapped Sensor

...
```

Ponovno instaliranje kernel-a sa izvršenim izmenama.

terminal

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- multi_v7_defconfig
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
```

Pravljenje patch-a za u boot i kernel za upotrebu na odbrani ...

U postojeći fajlsistem potrebno je dodati direktorijum /www i u njega prekopirati folder cgi-bin dobijen u postavci zadatka Ovo verovatno u rootfs ...

Za pravljenje ovog fajl sistema koristi se BusyBox

Moraju da bude exportovano sranje u PATH-u.

terminal

```
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- defconfig
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- menuconfig
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- install
```

Za pravljenje root fajlsistem-a koristi se BusyBox, koji instalira osnovni rootfs dovoljan za ovaj projekat. Kada se instalira potrebno je dodati pakete sqlite, i2c-tools za komunikaciju.

pre instaliranja potrebno je u menuconfig podesiti instalacioni direktorijum. /rootfs
Eksportovanje promenljivih

terminal

```
$ export SYSROOT=$(arm-linux-gnueabi-gcc -print-sysroot)
$ export STAGING=/home/aleksandarv/rootfs
```

Instalacija sqlite, kros kompajlira se za arm-linux arhitekturu.

terminal

```
wget http://www.sqlite.org/2015/sqlite-autoconf-3081101.tar.gz
tar xf sqlite-autoconf-3081101.tar.gz
cd sqlite-autoconf-3081101

CC=arm-linux-gnueabi-gcc ./configure --host=arm-linux-gnueabi --prefix=/usr
```

terminal

```
$ make DESTDIR=$SYSROOT install  
$ make DESTDIR=$STAGING install
```

Dodavanje neophodnih paketa u emulirani operativni sistem.

terminal

```
$ cp -a $SYSROOT/usr/lib/libsqlite3.so.0 $STAGING/lib/  
$ cp -a $SYSROOT/usr/lib/libsqlite3.so.0.8.6 $STAGING/lib/  
$ cp -a $SYSROOT/lib/libdl.so.2 $STAGING/lib/  
$ cp -a $SYSROOT/lib/libdl-2.28.so $STAGING/lib/  
$ cp -a $SYSROOT/lib/libpthread.so.0 $STAGING/lib/  
$ cp -a $SYSROOT/lib/libpthread-2.28.so $STAGING/lib/
```

Sledeće što je potrebno jeste da se dohvati i instalira paket i2c-tools.

Za šta služi ovaj paket na emuliranom sistemu?

terminal

```
$ wget https://git.kernel.org/pub/scm/utils/i2c-tools/  
/i2c-tools.git/snapshot/i2c-tools-4.1.tar.gz  
$ tar xf i2c-tools-4.1.tar.gz  
$ cd i2c-tools-4.1  
  
$ make CC=arm-linux-gnueabi-gcc  
$ make PREFIX=/usr DESTDIR=~/.Downloads/rootfs install
```

5 Pokretanje korisničke aplikacije

Za pokretanje emulatora potrebno je napraviti image koji bi bio boot-ovan.

Skripta koja se inicijalizuje na image-u:

terminal

```
setenv bootargs "root=/dev/mem rdinit=/sbin/init console=ttyAMA0"  
fatload mmc 0:1 82000000 zImage  
fatload mmc 0:1 88000000 vexpress-v2p-ca9-pds03.dtb  
fatload mmc 0:1 88080000 uRamdisk  
  
bootz 82000000 88080000 88000000
```

Pokretanje qemu emulatora :

terminal

```
$ qemu-system-arm -M vexpress-pds03 -m 1G -kernel u-boot-2019.01/u-boot \  
-nographic -drive file=sd.img,format=raw,if=sd \  
-device loader,file=u-boot-commands.img,addr=0x100000,force-raw -net nic \  
-net tap,ifname=tap0,script=no
```

Kada prođe ova komanda testiranje korsiničke aplikacije.

Ako je aplikacija uspešno pokrenuta na host računaru se može pristupiti podacima iz pretraživača preko lokalne adrese <http://192.168.10.101/cgi-bin/samples>.