

# Izveštaj za projekat iz predmeta Projektovanje Digitalnih sistema

Aleksandar Vuković 2018/3034

Poslednji put promenjeno: 26. 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 peč:

```
terminal
```

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

Ispis peča sors koda 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 kompajlirati pečovan *QEMU*:

terminal

```
$ mkdir -p bin/arm
$ cd bin/arm
$ ../../configure --target-list=arm-softmmu --enable-sdl --with-sdlabi=2.0
--enable-tools --enable-fdt --enable-libnfs --audio-drv-list=alsa
$ make -j4
```

Ispis kompilatora:

terminal

```
GEN      config-host.h
GEN      trace/generated-tcg-tracers.h
GEN      trace/generated-helpers-wrappers.h
GEN      trace/generated-helpers.h
GEN      trace/generated-helpers.c
GEN      module_block.h
CC       block.o
LINK     qemu-nbd
LINK     qemu-img
LINK     qemu-io
GEN      arm-softmmu/config-target.h
CC       arm-softmmu/hw/misc/pds-mmsens.o
CC       arm-softmmu/hw/misc/pds-i2csens.o
CC       arm-softmmu/hw/arm/vexpress.o
GEN      trace/generated-helpers.c
LINK     arm-softmmu/qemu-system-arm
```

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 razvojnom 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_pds03_defconfig
```

```
CONFIG_TARGET_VEXPRESS_PDS03=y
CONFIG_SYS_TEXT_BASE=0x80800000
```

```
...
```

Nova ploča za razvoj bi trebalo da ima zaseban *header* fajl *vexpress\_ca9x4\_pds03.h*. Novi fajl je dobijen kopiranjem koda iz *./include/configs/vexpress\_common.h* i menjanjem sledećih makroa.

```
./include/configs/vexpress_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))

#define PDS_VE_MM_SENSOR (V2M_PA_CS7 + V2M_PERIPH_OFFSET(24))
#define PDS_VE_I2C      (V2M_PA_CS7 + V2M_PERIPH_OFFSET(8))
...
```

Potrebno je i konfigurisati pre instalacije u-boot-a fajl *../board/arm ltd/vexpress\_pds03* dobijen kopiranjem *../board/arm ltd/vexpress\_ca9*.

Naredne linije kod su dodate u fajlove *board/arm ltd/vexpress\_pds03/Kconfig* i *board/arm ltd/vexpress\_pds03/Makefile*

```
./board/arm ltd/vexpress_pds03/Kconfig
if TARGET_VEXPRESS_PDS03

config SYS_BOARD
    default "vexpress"

config SYS_VENDOR
    default "arm ltd"

config SYS_CONFIG_NAME
    default "vexpress_pds03"

endif

...
```

Zameniti objekat koji se kompajlira:

```
./board/arm ltd/vexpress_pds03/Makefile
obj-$(CONFIG_TARGET_VEXPRESS_PDS03) += vexpress_tc2.o
```

U fajl *arch/arm/Kconfig* dodati:

```
./arch/arm/Kconfig

config TARGET_VEXPRESS_PDS03
    bool "Support vexpress_pds03"
    select CPU_V7A
    select PL011_SERIAL

source "board/arm ltd/vexpress_pds03/Kconfig"
```

terminal

```
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- \
vexpress_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 li već kompajlirani u-boot može da se ponovo kompajlira za drugu arhitekturu.

terminal

```
===== WARNING =====
This board does not use CONFIG_DM_MMC. Please update
the board to use CONFIG_DM_MMC before the v2019.04 release.
Failure to update by the deadline may result in board removal.
See doc/driver-model/MIGRATION.txt for more info.
=====
```

Ovo je *warning* koji se ispisuje ne vezano za izabranu platformu.

Peč za u-boot se koristi:

terminal

```
$ patch -p1 < u-boot-2019.01.patch
```

Fajlovi koji se menjaju prilikom primene peč.

terminal

```
patching file arch/arm/Kconfig
patching file board/armltd/vexpress_pds03/.built-in.o.cmd
patching file board/armltd/vexpress_pds03/Kconfig
patching file board/armltd/vexpress_pds03/MAINTAINERS
patching file board/armltd/vexpress_pds03/Makefile
patching file board/armltd/vexpress_pds03/vexpress_common.c
patching file board/armltd/vexpress_pds03/.vexpress_common.o.cmd
patching file board/armltd/vexpress_pds03/vexpress_common.su
patching file board/armltd/vexpress_pds03/vexpress_tc2.c
patching file configs/vexpress_pds03_defconfig
patching file include/configs/vexpress_pds03.h
```

## 4 Kernel, moduli drajvera, rootfs

Fajlovi `./linux-*/arch/arm/boot/dts/vexpress-v2m-ca9-pds03.dtsi` i `./linux-*/arch/arm/boot/dts/vexpress-v2p-ca9-pds03.dts` su dobijeni kopiranjem `vexpress-v2m.dtsi` i `vexpress-v2p-ca9.dts`, dodavanjem sledećeg koda:

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

pds03_mmsensor@18000 {
    compatible = "pds03,mmsensor";
    reg = <0x18000 0x1000>;
    interrupts = <27>;
};
pds03_i2c: i2c@8000 {
    compatible = "arm,versatile-i2c";
    reg = <0x08000 0x1000>;
    #address-cells = <1>;
    #size-cells = <0>;
};

...
```

i izmenom sledećih linija koda:

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

...

v2m_serial0: uart@e000 { // changed
    compatible = "arm,pl011", "arm,primecell";
    reg = <0x0e000 0x1000>; // changed
    interrupts = <5>;
    clocks = <&v2m_osccclk2>, <&smbclk>;
    clock-names = "uartclk", "apb_pclk";
};

v2m_timer01: timer@15000 { // changed
    compatible = "arm,sp804", "arm,primecell";
    reg = <0x15000 0x1000>; // changed
    interrupts = <2>;
    clocks = <&v2m_sysctl 0>, <&v2m_sysctl 1>, <&smbclk>;
    clock-names = "timclken1", "timclken2", "apb_pclk";
};

...
```

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

```
./linux-*/arch/arm/boot/dts/vexpress-v2p-pds03.dts

#include "vexpress-v2m-pds03.dtsi"

...
```

Fajlovi sa ekstenzijom `.dtb` su binarni fajlovi.

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

```
./linux-*/arch/arm/boot/dts/Makefile

dtb-$(CONFIG_ARCH_VEXPRESS) += \
    vexpress-v2p-ca5s.dtb \
    vexpress-v2p-ca9.dtb \
    vexpress-v2p-pds03.dtb \ % added line
    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.

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

```
./linux-*/drivers/char/Makefile

obj-$(CONFIG_PDS03_MMSENSOR) += mmsensor_03.o

...
```

```
./linux-*/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- menuconfig
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- multi_v7_defconfig
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
```

Peč za kernel se koristi:

terminal

```
$ patch -p1 < linux-5.0.2.patch
```

Fajlovi koji se menjaju.

terminal

```
patching file arch/arm/boot/dts/Makefile
patching file arch/arm/boot/dts/vexpress-v2m-pds03.dtsi
patching file arch/arm/boot/dts/vexpress-v2p-pds03.dts
patching file .clang-format
patching file .cocciconfig
patching file drivers/char/etf-pds-03.c
patching file drivers/char/Kconfig
patching file drivers/char/Makefile
patching file .get_maintainer.ignore
patching file .gitattributes
patching file .gitignore
patching file .mailmap
```

## 5 Root fajlsistem

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

Za pravljenje ovog fajl sistema koristi se BusyBox.

terminal

```
% $ wget http://busybox.net/downloads/busybox-1.30.1.tar.bz2
$ cd busybox-1.30.1/
```

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

terminal

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

Eksportovanje promenljivih

terminal

```
$ export SYSROOT=$(arm-linux-gnueabi-hf-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.

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=~/.rootfs install
```

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

## 6 Pokretanje korisničke aplikacije

Deo rezultata pokretanja skripte ./make\_rootfs.h



terminal

```
...  
  
90191 blocks  
Image Name:  
Created:      Tue Sep 15 18:37:00 2020  
Image Type:   ARM Linux RAMDisk Image (gzip compressed)  
Data Size:    16696676 Bytes = 16305.35 KiB = 15.92 MiB  
Load Address: 00000000  
Entry Point:  00000000
```

Potrebno je pokrenuti skriptu `./tap_start.sh` za mrežni interfejs, pre pokretanja qemu emulatora.

`./tap_start.sh`

```
sudo tuncctl -u $(whoami) -t tap0  
sudo ifconfig tap0 192.168.10.1  
sudo route add -net 192.168.10.0 netmask 255.255.255.0 dev tap0  
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Rezultat pokretanja skripte `./make_sd.h`

terminal

```
add map loop8p1 (253:0): 0 131072 linear 7:8 2048  
add map loop8p2 (253:1): 0 1964032 linear 7:8 133120  
total 25M  
drwxr-xr-x  2 root root  16K Dec 31  1969 .  
drwxr-xr-x 20 root root  4.0K Sep 11 08:39 ..  
-rwxr-xr-x  1 root root  16M Sep 15 18:40 uRamdisk  
-rwxr-xr-x  1 root root  14K Sep 15 18:40 vexpress-v2p-pds03.dtb  
-rwxr-xr-x  1 root root  8.4M Sep 15 18:40 zImage  
loop deleted : /dev/loop8
```

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 \  
-net nic -net tap,ifname=tap0,script=no
```

Komande koje se pokreću u u-boot -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-pds03.dtb  
=> fatload mmc 0:1 88080000 uRamdisk  
  
=> bootz 82000000 88080000 88000000
```

Pokretanje aplikacije:

terminal

```
ifup eth0  
httpd -p 80 -h /www  
/home/pds03_app
```

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>*.