Rīgas Tehniska Universitāte

Datorvadības, automātikas un datortehnikas institūts

Datoru tīklu un sistēmas tehnoloģijas katedra

Mikroprocesoru sistēmu izstrāde un apkalpošana

atskaites par 1, 2 laboratorijas darbiem

Romans Taranovs

Vitālijs Hodiko

1.kurss 2.grupa

Rīga, 2013

1.Teorētiska daļa

LPC-2478STK izstrādes komplekts izmanto LPC-2478 mikro-kontrolieri no NXP, kas atbalsta vairākus seriālos interfeisus, piemēram, kā USB, UART, COM. Izstrādes plate ir aprīkota ar audio ieejam un izejam un mp3 dekodētāju, ciparu akselerometru, JTAG, Ethernet, skārienjutīgu ekrānu un t.t.. Salīdzinot LPC-2478STK plati ar mums iepriekš-zināmam platēm(msp430, charon2), tā ir daudz jaudīgāka, kas ļauj to izmantot daudz sarežģītākiem uzdevumiem. Ka piemērs, to var izmantot kā vārteju, kas apkopos datus no sensoru tikla un apstrādās un\vai pārraidīs tos tālāk globālajā\lokālajā tiklā. Ka sensoru to izmatot nav izdevīgi, to plašu iespēju dēļ, kas nav savienojami ar energoefektivitāti.

Platē ir iespējams instalēt Linux operētājsistēmu, taču neliela flash atmiņas daudzuma dēļ(512 KB), to ir jāpārkompilē, lai tās svērtu daudz mazāk par to skaitli. Demo versijā Linux(uCLinux) izmērs ir 3.3MB, un tāpēc to var palaist tikai šādi:

1)Izmantojot USB flash atmiņu;

2)Izmantojot SD/MMC karti;

3)caur Ethernet;

4)Ar seriāla interfeisa starpniecību.

Tas tiek ierakstīts SDRAM atmiņa, kuras apjoms ir 64 MB, kas ir diezgan daudz. Lai atvieglot uClinux palaišanu platē ir iebūvēts uBoot ielādētais un izmantojot attiecīgas komandas to var ātri palaist(vai mazliet modificēt kodu un tas palaidīsies automātiski).

Lai palaistu bināros failus platē tos ir jāielādē tajā caur Ethernet pieslēgumu izmantojot, piemēram, tftp rīku. Dotajā linux versija tikai /var mape var veikt izmaiņas, tostarp arī ielādēt programmas.

1.Praktiska daļa

Praktiskas daļas ietvaros bija apskatītas divas ielādes metodes: no USB(ir līdzīga SD\MMC) un caur Ethernet. Tāpat ka iepriekšēja reize host ir mašīna ar uzinstalētu Linux OS.

Lai varētu palaist uCLinux no usb, no sākuma vajag ielādēt USB trīs nepieciešamos failus: romfs_5.img, vmlinux.bin. Palaist uCLinux var izpildot šādu komandu secību:

```
lpc-2478-stk# usb start
lpc-2478-stk# fatload usb 0 0xa0800000 romfs_5.img
lpc-2478-stk# fatload usb 0 0xa0008000 vmlinux.bin
lpc-2478-stk# go a0008000
```

Cita iespēja ir ielādēt uCLinux no datora izmantojot Ethernet. Taču šeit var rasties dažās grūtības, jo ir nepieciešams speciāls kabelis, kurā ir krustoti noteikti vadi. Lai apietu šo problēmu var izmantot ruteri un pieslēgt plati pie tā. Pēc tam vajag nokonfigurēt ftp serveru. Vispirms, ar šo komandu instalējam nepieciešamos rīkus:

```
sudo apt-get install xinetd tftpd tftp
```

Tagad vajag izveidot tftp konfigurācijas failu

```
sudo vi /etc/xinetd.d/tftp
```

ar šādu saturu:

```
service tftp
{
      protocol = udp                    #komunikācijas protokols
      port = 6999                       #ports kura klausīties
      socket_type = dgram
      wait = yes
      user = nobody
      server = /usr/sbin/in.tftpd
      server_args = /home/tftp          #direktorija kurai pieslegsies
      disable = no
}
```

Un izveidojam direktoriju, kura glabāsies uCLinux faili:

```
sudo mkdir /home/tftp
```

un mainām pieejas privilēģijas

```
sudo chmod -R 777 /home/tftp              #atļaujam visiem rakstīt, lasīt un palaist
sudo chown -R nobody:nogroup /home/tftp   #mainām grupu un īpašnieku
```

Un restartejam xinet, lai izmaiņas stātos spēkā

```
sudo restart xinetd
```

Pēdējais solis, ko aizpilda, lai varētu sākt uClinux ielādi platē ir jāpārbauda ip adreses. Ja tie

neatbilst reāliem, tad konfigurējam tos ar šādam komandām:

```
setenv gatewayip 192.168.1.1          # uzstādām vārtejas IP-adresi
setenv ipaddr 192.168.1.20            # uzstādām izstrādes plates IP-adresi
setenv serverip 192.168.1.100         # uzstādām servera IP-adresi
saveenv                               #un saglabājam izmaiņas
```

Tagad ielādējam nepieciešamos failus un palaižam uClinux:

```
tftpboot 0xa0800000 romfs_5.img
tftpboot 0xa0008000 vmlinux.bin
go a0008000
```

Lai nevajadzētu katru reizi manuāli palaist uCLinux var nokonfigurēt U-Boot bootcmd parametru, lai tas palaistos automātiski. Piemēram, lai automātiski palaist uCLinux no USB, vajag izpildīt šādas komandas:

```
setenv bootcmd 'usb start; fatload usb 0 0xa0800000 romfs_5.img; fatload usb
0 0xa0008000 vmlinux.bin; go a0008000'
saveenv
```

Secinājumi

Pirmā laboratorijas darba ietvaros mēs iepazināmies ar LPC2478-STK izstrādes plati un dažam tas specifiskam īpašībām, tostarp ir dažas ielādes iespējas, tehniskais nodrošinājums, operētājsistēmas ierobežojumi, ka arī plates ierobežojumi.

Manuprāt, viena no vājam vietām ir mazs iebūvētas flash atmiņas apjoms(2KB) salīdzinājumā ar SDRAM(64MB), kaut arī pastāv iespēja izmantot ārējo flash\sdmmc karti. Cita neērtība, kas būtība ir pati galvena, ir nepieciešamība pēc DB9 RS232 kabeļa, kurš tiek izmantots plates kontrolei, caur konsoli. Pirmkārt, tas ir neērti, un otrkārt, tagad, ne visos datoros šādi porti eksistē.

Salīdzinot abus izmantotus ielādes\palaišanas metodes ir acīmredzami, ka visvienkāršākais ir nokonfigurēt automātisku uClinux ielādi no ārēja datu nesēja(flash, sdmmc) un palaišanu.

2. Praktiska daļa

Lai varētu palaist izpildāmo failu uz LPC2478-STK platēs to vajag nokompilēt ar tai paredzētu kroskompilatoru(angl. Cross-compiler), ko var atrast olimex vietnē. Viena no izplatītākam izstrādes valodām ir C, kas arī tika izmantotā vienkāršas programmas izveide:

```
#include <stdio.h>
int main(int argc, char** argv){
      printf("STKllo world from box!\n");
      return 0;
}
```

Kā redzams, koda netika izmantotas specifiskas darbības ar plates reģistriem, bet tikai "tīra" c valodas darbības.

Kroskompilatora instalēšana nav pārāk sarežģīta lieta, taču arī tur var rasties problēmas nepieredzējušam cilvēkam. Kroskompilators ir rīks, kas dod iespēju ģenerēt izpildāmo failu uz vienas arhitektūras mašīnas, jeb hosta (angl.host), lai to varētu palaist uz citas arhitektūras mērķa-mašīnas(angl. Target). Šajā darba es izmantoju http://www.henjab.se/ piedāvāto risinājumu.

Lai varētu nokompilēt kodu platei, vispirms vajag nokompilēt uClinux, kā rezultāta iegūsim kroskompilatoru.

No sākuma izveidojam izstrādes darba direktoriju.

```
sudo mkdir /opt/arm && cd /opt/arm
```

Savedam kartība nepieciešamos rīkus.

```
tar -xvzf .../Utils/arm-linux-tools-20061213.tar.gz   # atrodas uz diska
tar -xvzf .../uClinux/uClinux-dist-lpc_2478_stk-20081007.tgz      # komplekta ar plati
```

Ja genromfs nav komplekta, tad to jāielādē atsevišķi.

```
wget      http://www.uclinux.org/pub/uClinux/uclinux-elf-tools/gcc-3/uclinux-tools-20040603/genromfs-0.5.1.tar.gz
 tar -xvzf genromfs-0.5.1.tar.gz
cd genromfs-0.5.1 && make && cd ..
```

Konfigurējam ceļus līdz rīkiem un attīrām direktorijas pirms kompilācijas.

```
export PATH=/opt/arm/usr/local/bin:/opt/arm/genromfs-0.5.1:$PATH
cd uClinux-dist-lpc_2478_stk
make distclean
rm -f tools/ucfront-g++
rm -f tools/ucfront-gcc
rm -f tools/ucfront-ld
```

rm -f tools/cksum

Pienācis laiks nokompilēt uClinux, bet vispirms konfigurēsim to.

make menuconfig

Un uzstādām NXP/LPC2468 kā Vendor/Product izvēloties Vendor/Product Selection —> Vendor -> NXP. Kā vienīga alternatīva LPC2468 būs uzstādīts automātiski. Izejam un palaižam kompilācijas komandas.

make dep && make

Ja kompilācijas laika neradās kļūdas, tad rezultāta images direktorija ir daži faili, romfs*.img un vmlinux*.bin. Pievienojam uClinux rīku direktoriju PATH mainīgam.

export PATH=/opt/arm/uClinux-dist-lpc_2478_stk/tools:$PATH

Lai to nevajadzētu darīt katru reizi pārstartējot datoru var visus ceļus pievienot ./bashrc skripta.

Jaunizveidotos failus var izmantot uClinux palaišanai no plates oriģinālo vietā.

Izveidojot Makefile failu ar šādu saturu:

```
CC=ucfront-gcc arm-linux-gcc
 LD=ucfront-gcc arm-linux-gcc
CFLAGS=-msoft-float -D__PIC__ -fpic -msingle-pic-base -Dlinux -D__linux__
-Dunix -D__uClinux__
 LDFLAGS=-Wl,--fatal-warnings -Wl,-elf2flt -msoft-float -L/opt/arm/uClinux-
dist-lpc_2478_stk/uClibc/lib
 SOURCES=my_source_filename.c
 OBJECTS=$(SOURCES:.c=.o)
TARGET=source_file_out_name
 .PHONY=all
 all: $(TARGET)
%.o : %.c
  $(CC) $(CFLAGS) -c $<
$(TARGET): $(OBJECTS)
  $(LD) $(LDFLAGS) -o $@ $(OBJECTS) -lc
.PHONY=clean
 clean:
  \rm -f *.o
  \rm -f *~
```

Un novietojot to kopā ar izejas kodu, var termināla palaist make komandu, kā rezultāta saņemsim mums nepieciešamo bināro failu. Lai to palaist vispirms vajag to ielādēt platē /var mapē. To izdarām izmantojot iepriekš nokonfigurētu tftp serveru.

Palaižam uClinux un ieslēdzam tīklu platē

```
/sbin/ifconfig eth0 192.168.1.20 up
```

Pārbaudām vai ir savienojums ar serveru

```
ping -c 1 192.168.1.100
```

Ieladējam bināro failu platē un palaižam to.

```
cd /var
tftp -g -r binary_filename 192.168.1.100
chmod +x binary_filename                    # uzstādām izpildāma faila karogu
./binary_filename
STKllo world from box!
```

Secinājumi

Šīs darbs salīdzinājumā ar pirmo aizņēma daudz vairāk laikā, galvenokārt, tāpēc ka vajadzēja ģenerēt jaunus uClinux failus, kas ir samēra grūti bez iepriekšējas pieredzes. Šajā solī bija daudz kļūdu, kurus vispirms vajadzēja saprast un pēc tam izlabot, taču ar katru nākamo reizi palika vieglāk. Šo kļūdu iemesls ir dažu bibliotēku trūkums, kurus vajadzēja atsevišķi uzstādīt. Programmas kompilēšana un uzrakstīšana bija triviāls uzdevums.

Rezultāta bija iegūta prasme uClinux un tam paredzētu bināru failu kompilācijā.