

Introduktion till Olimex LPC-2478-STK

Kjell Enblom

Lysator

September 2009

Copyright © 2009 Kjell Enblom. This document is covered by the GNU Free Documentation License,
Version 1.1 or later.

Introduktion till Olimex LPC-2478-STK

Olimex LPC-
2478-STK

Kjell Enblom

Denna presentation, dokumentation och exempelkod finns på
<http://www.lysator.liu.se/~kjell-e/embedded/olimex/>

Presentationen finns även på:
<http://www.lysator.liu.se/~kjell-e/tekla/linux/dokument.html>

Introduktion till Olimex LPC-2478-STK

Olimex LPC-
2478-STK

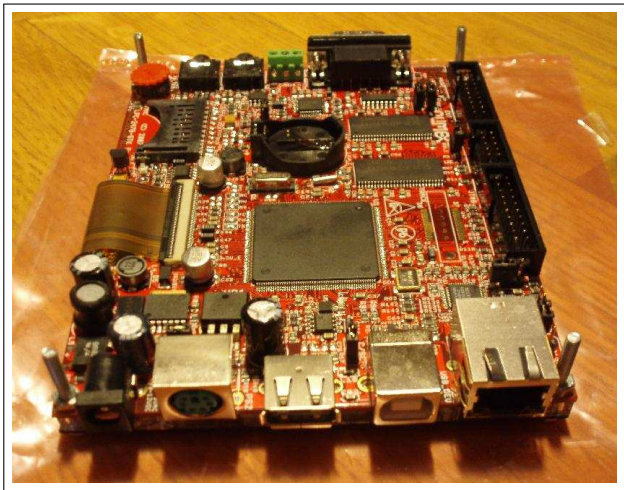
Kjell Enblom

- LPC-2478-STK är ett utvecklingskort för inbyggda system.
- Inbyggda system är datorer som sitter inbyggda i apparater; tvättmaskiner, mikrovågsugnar, bilar, digitalboxar för TV, mobiltelefoner, handdatorer etc.
- Inbyggda system har oftast begränsade resurser som t.ex. relativt liten mängd minne, ingen hårddisk, litet flashminne, långsammare CPU än i moderna arbetsstationer och servrar etc.
- Inbyggda system kan t.ex. ha en mängd med I/O-portar, exempelvis för serieportar, I^2C -bussar, I^2S -bussar, CAN-bussar etc.
- Inbyggda system kan sakna minnesskydd.
- Många riktigt små inbyggda system saknar operativsystem.

Introduktion till Olimex LPC-2478-STK

Olimex LPC-
2478-STK

Kjell Enblom



Introduktion till Olimex LPC-2478-STK

Olimex LPC-
2478-STK

Kjell Enblom

- Avsikten med denna introduktion är att lära sig så pass mycket om Olimexkortet att man snabbt kan komma igång och börja labba med detta kort.
- Innehåll:
 - Data om kortet
 - Spänningsmatning
 - Seriekonsolen
 - Grunderna i bootloadern U-Boot
 - Boota från USB
 - Boota från SD-kort
 - Boota via tftp-server
 - Kompilera om uClinux
 - Boota med nfs-rootfilsystem
 - Kompilera program inom uClinux byggsystem och utanför
 - Bilder

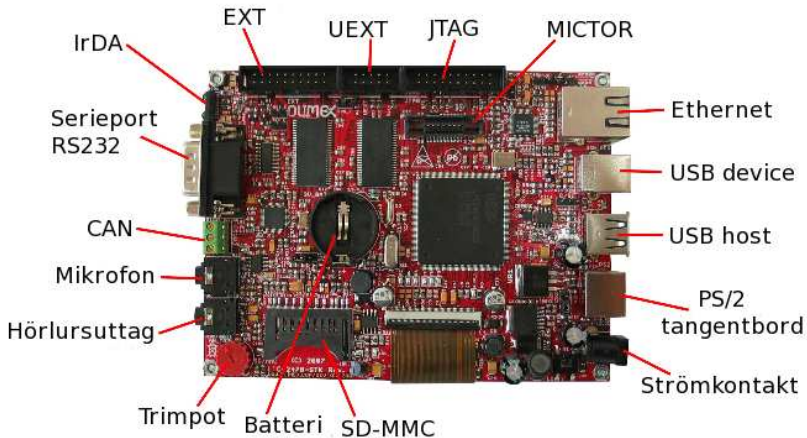
Data i korthet:

- ARM7-CPU (ARM7TDMI-S, little endian).
- 3.5 tums TFT-skärm med backlight och touchscreen (320x240 punkter)
- MP3-dekoder
- 64 MB SDRAM
- 512 KB on-chip flash
- Plats för SD/MMC-kort.
- CAN-buss
- JTAG-anlutning
- Ethernetanslutning
- Serieport (9-pol DSUB)
- USB host och USB device
- U-boot boot-loader

Introduktion

Olimex LPC-
2478-STK

Kjell Enblom



Introduktion - spänningsmatning och serieport

Olimex LPC-
2478-STK

Kjell Enblom

- Olimex LPC-2478-STK kan matas med antingen växelspänning eller likspänning
 - AC 6-9 V eller
 - DC 9-12 V.
- Polariteten på kontakten spelar ingen roll då det sitter en likriktarbrygga efter kontakten. Denna matar sedan spänningsstabilisatorerna som ger 5V respektive 3.3V.
- Clas Ohlsson batterieliminators 32-2314 fungerar utmärkt.
- Serieporten är inställd på 115200 bps 8 databitar ingen paritet.
- För att ansluta seriporten till en annan dator behövs en nollmodemkabel som korsar bland annat stiften 2 och 3.

Introduktion - 9-pol seriekontakt

Olimex LPC-
2478-STK

Kjell Enblom

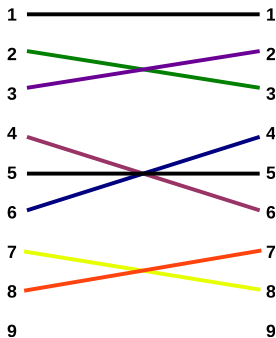
Den 9-poliga Dsub seriekontakten ser ut enligt följande:

- 1 - Frame ground (N/U)
- 2 - Rx Data
- 3 - Tx Data
- 4 - DTR
- 5 - Signal Ground
- 6 - DSR
- 7 - RTS
- 8 - CTS
- 9 - N/U

Introduktion - nollmodemkabel

Olimex LPC-
2478-STK

Kjell Enblom



- Nollmodemkabel (9-pol hona-hona) finns att köpa på välsorterade elektronikbutiker som t.ex. Clasohlsson.
- USB till serieport finns även det att köpa på välsorterade elektronikbutiker.

Blockschema för den integrerade kretsens LPC2478

Olimex LPC-2478-STK

Kjell Enblom

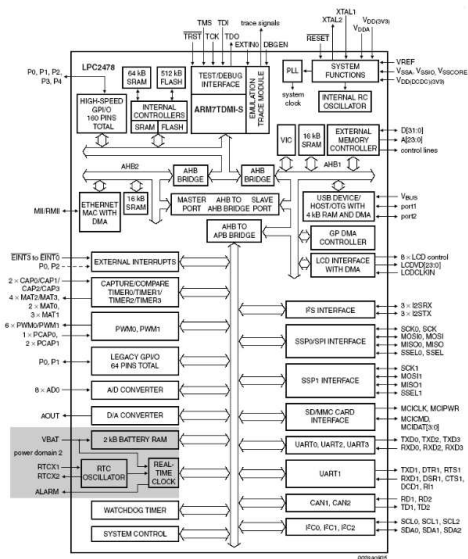
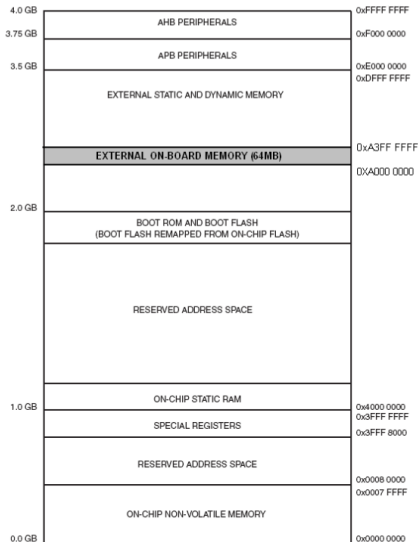


Fig. 1. LPC2478 block diagram

Minneslayout

Olimex LPC-
2478-STK

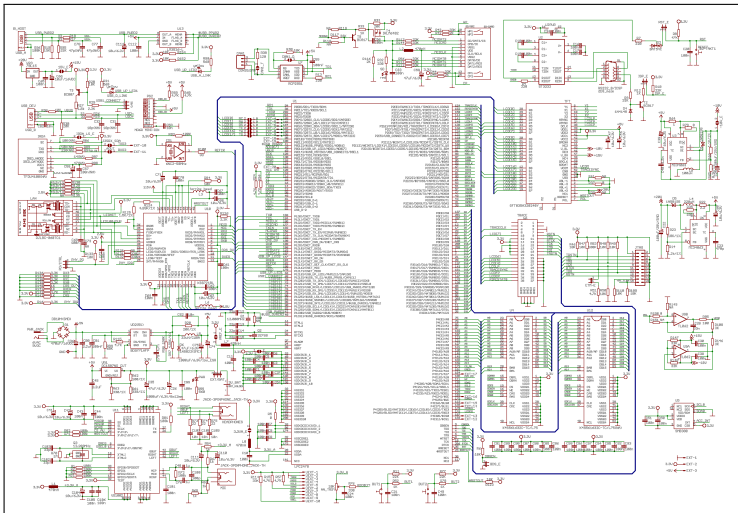
Kjell Enblom



Schema

Olimex LPC-
2478-STK

Kjell Enblom



Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- Bootloadern på Olimexkortet är Das U-Boot (U-Boot).
- Olimexkortet använder serieportskonsol på vilken den kommunicerar i 115200 bps, 8 databitar, ingen paritet.
- Anslut serieporten och starta t.ex. kermit eller picocom.
- Exempel:
picocom -b 115200 /dev/ttyUSB0
- Anslut ström till kortet.
- Innan U-Boot har hunnit börja boota tryck på någon tangent. U-Boot har en timeout på 3 sekunder som standard.
- Alla inställningar inklusive timeout lagras i variabler i U-Boot.

Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- Skriv **help** eller **?** för att lista alla kommandon.
- Det går även att skriva **help kommando** för att få mer hjälp om ett specifikt kommando.
- **printenv** visar alla variabler och deras värden.

```
bootargs=root=/dev/ram initrd=0xa0800000,4000k console=ttyS0,115200N8
bootcmd=run usb_boot
bootdelay=3
baudrate=115200
tftp_boot=tftpboot a0008000 linux.bin;tftpboot a1800000 romfs.bin;go a0008000
nand_boot=nand read 0xa0008000 0x0 0x220000;nand read 0xa0800000 0x400000 0x2200
usb_boot=usb start;fatload usb 0 0xa0800000 romfs_5.img;fatload usb 0 0xa0008000
mmc_boot=mmc;fatload mmc 0 a0008000 linux.bin;fatload mmc 0 a1800000 romfs.bin;g
update_uboot=tftpboot a1000000 u-boot.bin;protect off 0 2ffff;erase 0 2ffff;cp.b
update_nand=nand erase;tftpboot a1000000 linux.bin;nand write a1000000 0 200000;
ipaddr=192.168.0.158
netmask=255.255.255.0
serverip=192.168.0.240
ethaddr=00:de:ad:b0:05:03
console=serial
stdin=serial
stdout=serial
stderr=serial
```

Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- Olimexkortet kan ladda filer från USB-minne SD/MMC-kort eller från en tftp-server.
- USB och SD/MMC anges på följande sätt:
 - **usb 0** anger första USB-enheten med fat-filsystem (vfat).
 - **mmc 0** anger första SD-kortet med fat-filsystem (vfat).
- **fatls <interface> <dev[:part]> [directory]** - listar filer.
- **fatload <interface> <dev[:part]> <addr> <filename> [bytes]** - laddar en fil.
- addr är en minnessadress.
- Exempel:
fatls usb 0
- Exmpel:
fatload usb 0 0xa0800000 romfs_5.img

Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- För att kunna accessa USB-minnen måste man först aktivera USB-stödet.
- USB startas med **usb start**
- USB-stödet stoppas igen med **usb stop**
- USB-partitioner visas med **usb part**
- **help usb** ger mer hjälp om vad man kan göra med USB i U-Boot.

Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- **tftpboot [loadAddress] [[hostIPAddr:]bootfilename]** - laddar en fil från en tftp-server och lägger den på angiven minnesadress.
- Exempel:
**tftpboot 0xa0008000
192.168.0.1:/linux-install/olimax/vmlinux.bin**
- **go addr [arg ...]** - kör det som finns på minnesadress addr.
- Exempel:
go 0xa0008000

Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- **run var [...]** - Kör kommandona i variabeln var.
- Exempel:
 - Variabeln `tftp_boot` innehåller följande:

```
tftpboot 0xa0800000  
192.168.0.1:/linux-install/olimax/romfs_5.img;  
tftpboot 0xa0008000  
192.168.0.1:/linux-install/olimax/vmlinux.bin;go a0008000
```
 - **run tftp_boot**
- Det går att betrakta variabeln som ett shellsript som man kör.

Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- Ändra innehåll på en variabel görs med:
setenv variabel värde
- Exempel:
**setenv bootargs 'root=/dev/ram
initrd=0xa0800000,4000k console=ttyS0,115200N8'**
- Det går att spara alla variablers innehåll till flash med
saveenv
- Kommandot **bdinfo** visar lite mer information om kortet
(print Board Info structure).
- **ping host** skickar pingpaket till angiven host. Exempel:
ping 192.168.0.1

Grunderna i U-Boot

Olimex LPC-
2478-STK

Kjell Enblom

- Standardalternativet för U-Boot är att köra innehållet i variabeln bootcmd, d.v.s. utföra **run bootcmd**
- För Olimex-kortet innebär det att filerna romfs_5.img och vmlinux.bin läses in från ett VFAT-formatterat USB-minne.
- romfs_5.img innehåller ett romfs med root-filsystem.
- vmlinux.bin innehåller en linux-kärna.
- Dessa två filer finns i katalogen Images på den medföljande CDn till Olimexkortet.
- I samma katalog finns även u-boot-bin.hex som är en U-Boot image.
- Med U-boot går det att flash om den flash som finns på kortet.

U-Boot - boota från USB

Olimex LPC-
2478-STK

Kjell Enblom

- För att boota från USB behöver man först starta USB-stödet, ladda in root-filsystemet, ladda kärnan och slutligen köra igång kärnan.
- Notera att USB-minnet måste innehålla ett VFAT-filsystem.
 - **usb start**
 - **fatload usb 0 0xa0800000 romfs_5.img**
 - **fatload usb 0 0xa0008000 vmlinux.bin**
 - **go a0008000**
- Dessa inställningar finns som standard i variabeln `usb_boot`.
- I och med det går det att köra **run usb_boot** för att boota från USB.

U-Boot - boota från SD-kort

Olimex LPC-
2478-STK

Kjell Enblom

- För att boota från SD-kort behöver man först starta mmc, ladda in kärnan, ladda root-filsystemet och slutligen köra igång kärnan.
- Notera att SD-kortet måste innehålla ett VFAT-filsystem.
 - **mmc**
 - **fatload mmc 0 a0008000 linux.bin**
 - **fatload mmc 0 a1800000 romfs.bin**
 - **go a0008000**
- Dessa inställningar finns som standard i variabeln `mmc_boot`.
- I och med det går det att köra **run mmc_boot** för att boota från USB.

U-Boot - boota via tftp

Olimex LPC-
2478-STK

Kjell Enblom

- För att ladda filerna från en tftp-server och boota från dessa filer behövs en fungerande tftp-server med de aktuella filerna och en fungerande ethernetförbindelse.
- Nedan visas inställningarna för en tftp-server som startas från xinetd.

```
service tftp
{
    socket_type           = dgram
    protocol              = udp
    wait                  = yes
    user                  = root
    server                 = /usr/sbin/in.tftpd
    server_args           = -s /tftpboot
    disable               = no
    per_source            = 11
    cps                   = 100 2
    flags                 = IPv4
}
```

- tftp-servern ovan chrootas till katalogen /tftpboot

U-Boot - boota via tftp

Olimex LPC-
2478-STK

Kjell Enblom

- Filerna i följande exempel är placerade i katalogen /tftpboot/linux-install/olimex/ på host-datorn.
- Host-datorn har här IP-adress 192.168.0.1.
- Olimexkortet är target.
- Här laddas först filen med root-filsystemet, därefter laddas kärnan och slutligen körs kärnan.
 - **tftpboot 0xa0800000**
192.168.0.1:/linux-install/olimex/romfs_5.img
 - **tftpboot 0xa0008000**
192.168.0.1:/linux-install/olimex/vmlinux.bin
 - **go a0008000**

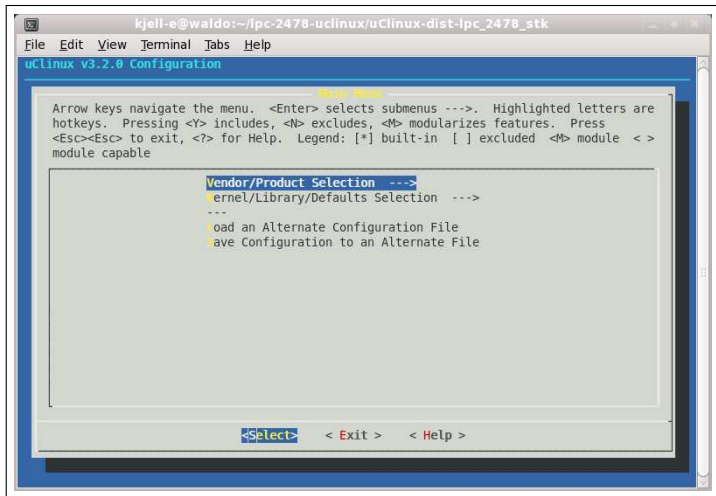
- Med Olimex följer uClinux.
- uClinux används framförallt på system som saknar MMU.
- Ingen MMU ger:
 - Inget virtuellt minne (VM) och ingen swap.
 - Inget minnesskydd. Det gör att en process kan skriva sönder en annan process minne.
 - Inget tmpfs.
- Eftersom systemet saknar MMU så stöds endast binärformatet Flat (bFLT) för program. Alla övriga binärformat för program unyttjar funktioner i VM.
- Kompilatorn som man använder för att kompilera uClinux och program med måste ha stöd för Flat. Den kan även behöva ha stöd för position-independent code (PIC) och execute-in-place (XIP).
- Stacken allokeras vid kompilering och kan inte växa under exekvering.

- Det går att ändra på stackstorleken med programmet **flthdr**
- Exempel:
**lpc-2478-
uclinux/buildroot/toolchain_build_arm/elf2flt/flthdr
-s 20k program**
- Heapen fungerar också annorlunda i uClinux då minnet allokeras från en global minnespol.

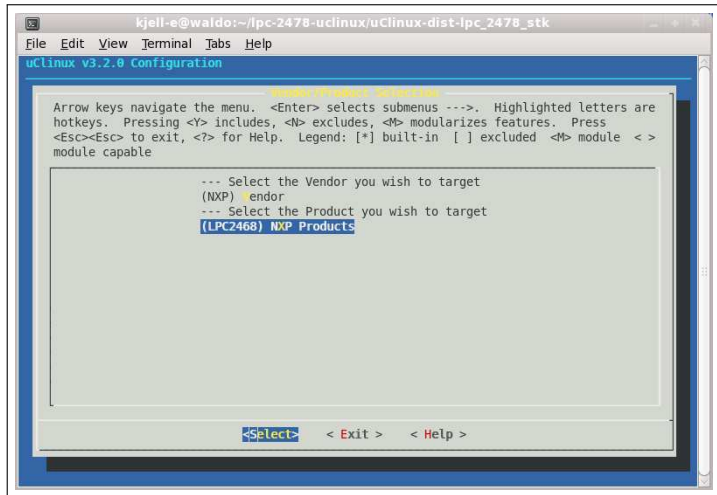
- Följande beskriver hur man kompilerar om uClinux (buildroot, kärna etc).
 - Skapa ett katalogträd,
mkdir -p ~/lpc-2478-uclinux/snapgear-cross
 - Gå ner i katalogträdet och packa upp toolchain för ARM (korskompilator), **cd ~/lpc-2478-uclinux/snapgear-cross**
 - **tar xzf**
/mnt/cdrom/Utils/arm-linux-tools-20061213.tar.gz
 - Sätt PATH till att inkludera katalogen med detta toolchain, **PATH=~/lpc-2478-uclinux/snapgear-cross/usr/local/bin:\$PATH**

- forts.
 - Nu är det dags att packa upp uClinux (inklusive källkoden till kärnan) och konfigurera och kompilera den.
 - Börja med att gå till katalogen `lpc-2478-uclinux`,
`cd ~/lpc-2478-uclinux/`
 - Packa upp tar-arkivet med källkoden,
`tar xzf /mnt/cdrom/uClinux/uClinux-dist-lpc_2478_stk-20081007.tgz`
 - Gå ner i katalogen och konfigurera uClinux.
`cd uClinux-dist-lpc_2478_stk`
`make menuconfig`
 - Se till att sätta Vendor till NXP och Product till LPC2468.

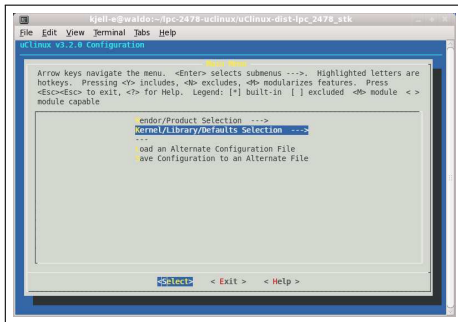
Gå in på 'Vendor/Product Selection'.



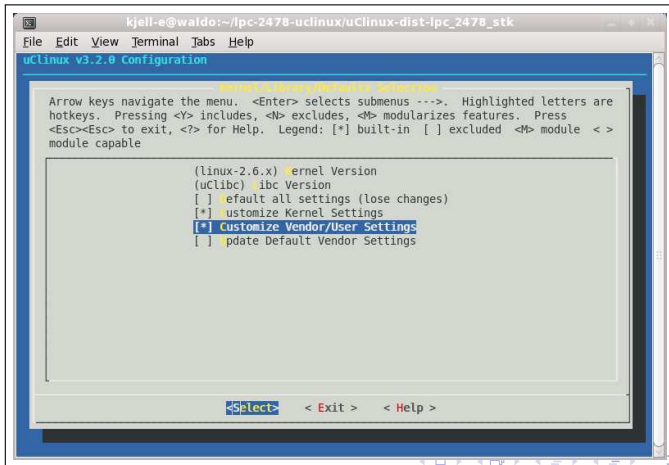
Sätt 'Vendor' till 'NXP' och 'Product' till 'LPC2468'.



- Forts.
 - För att konfigurera kärnan, busybox, etc. gå in på 'Kernel/Library/Defaults Selection' och välj 'Customize Kernel Settings' för kärnan och 'Customize Vendor/User Settings' för busybox och applikationer.
 - När menuconfig avslutas kommer den att gå vidare till konfigurationen för kärna respektive applikationer.



Aktivera 'Customize Kernel Settings' för att konfigurera kärnan och 'Customize Vendor/User Settings' för att konfigurera busybox och applikationer.



- Forts.
 - Nu är det dags att kompilera uClinux.
 - För att kompilera uClinux och bygga romfs behövs programmet genromfs.
 - Kompilera med: **make**
(toolchainet måste vara med i din PATH, se sida 28).
 - Om kompileringen gick bra finns ett träd med rootfilssystemet i katalogen romfs och de färdiga filerna med rootfilssystem och kärna i katalogen images.

- Om du bygger en egen kärna måste du kompilera den med **make ulmage**
- Du behöver även ha **genromfs** installerad på utvecklingsdatorn.

U-Boot - NFS-rootfilesystem

Olimex LPC-
2478-STK

Kjell Enblom

- För att kunna boota och montera root-filsystemet från en NFS-server behöver den medföljande kärnan patchas.
- Gå till
~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/linux-2.6.x/
- Spara nedanstående patch i filen nfsroot.patch

```
--- fs/nfs/oldfile.c      2009-06-24 18:13:46.000000000 +0200
+++ fs/nfs/file.c        2009-06-24 18:13:49.000000000 +0200
@@ -281,9 +281,13 @@

         status = nfs_revalidate_mapping(inode, file->f_mapping);
         if (!status) {
+##ifndef CONFIG_MMU
+
+         status = generic_file_mmap(file, vma);
+##else
+
+         vma->vm_ops = &nfs_file_vm_ops;
+         vma->vm_flags |= VM_CAN_NONLINEAR;
+         file_accessed(file);
+##endif
         }
         return status;
     }
 }
```

- Applicera patchen med **patch -p0 < nfsroot.patch**

U-Boot - NFS-rootfilesystem

Olimex LPC-
2478-STK

Kjell Enblom

- Därefter behöver kärnan konfigureras. Gör antingen som beskrivits tidigare eller kör **make menuconfig** i katalogen där du har källkoden till kärnan.
- Gå till 'Networking' – > 'Networking options' och aktivera TCP/IP-stöd och 'kernel level autoconfiguration' och eventuellt 'DHCP support' och 'BOOTP support'.
- Gå till 'File systems' – > 'Network File Systems' och aktivera 'NFS file system support', 'Provide NFSv3 client support', 'Root file system on NFS'.
- Gå slutligen in på 'Boot options' och sätt 'Default kernel command string' till

```
root=/dev/nfs console=ttyS0,115200N8 rw nfsroot=192.168.0.1:/export/olimax/root
```

```
ip=192.168.0.158:192.168.0.1:192.168.0.1:255.255.255.0:olimax::off
```

```
rootpath=/export/olimax/root/ init=/bin/sh
```

U-Boot - NFS-rootfilesystem

Olimex LPC-
2478-STK

Kjell Enblom

- Kompilera uClinux och installera den nya kärnan.
cd ..
make
cp images/vmlinux.bin
/tftpboot/linux-install/olimax/vmlinux-nfs.bin
- Starta kortet och ladda kärnan:
tftpboot 0xa0008000
192.168.0.1:/linux-install/olimax/vmlinux-nfs.bin
- Kör igång kärnan så att datorn bootar:
go 0xa0008000
- Om allt fungerar ska Olimexkortet boota och montera root-filsystemet från NFS-servern 192.168.0.1.
- Root-filsystemet kan kopieras från katalogen romfs.
cp -a
~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/romfs
/export/olimax/root

U-Boot - NFS-rootfilesystem

- Nedan visas en nedklippt skärmdump på en boot med nfs-rootfilesystem.

```
lpc-2478-stk # tftpboot 0xa0008000 192.168.0.1:/linux-install/olimex/vmlinux.bin
emac: check_phy - (22, 1619)
emac: link status = 100Mbps, full duplex
emac: MAC address = 0:de:ad:b0: 5: 3
TFTP from server 192.168.0.1; our IP address is 192.168.0.158
Filename '/linux-install/olimex/vmlinux-nfs.bin'.
Load address: 0xa0008000
Loading: #####
done
Bytes transferred = 2236072 (221ea8 hex)
lpc-2478-stk # go 0xa0008000
## Starting application at 0xA0008000 ...
úLinux version 2.6.24.2-uc0 (kjell-e@waldo.dyndns.org) (gcc version 3.4.4) #14 S
CPU: NXP-LPC2468 [24680000] revision 0 (ARMvundefined/unknown), cr=a0229ec0
Machine: Olimex LPC-2478-STK
Warning: bad configuration page, trying to continue
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 16256
Kernel command line: root=/dev/nfs console=ttyS0,115200N8 rw nfsroot=192.168.0.1
PID hash table entries: 256 (order: 8, 1024 bytes)
LPC22XX Clocking Fin=12000000Hz Fcco=288000000Hz M=11 N=0
Fcclk=57600000 PCLKSEL=55515555 11555455
Console: colour dummy device 80x30
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 64MB = 64MB total
```

U-Boot - NFS-rootfilesystem

Olimex LPC-
2478-STK

Kjell Enblom

```
Memory: 62628KB available (2000K code, 181K data, 92K init)
Mount-cache hash table entries: 512
net_namespace: 64 bytes
NET: Registered protocol family 16
eth0: Link down.
eth0: LPC22XX ethernet at 0xffe00000 int=21 10-FullDuplex (00:1a:f1:00:00:f6)
eth0: Micrel PHY at 1
block2mtd: version $Revision: 1.30 $
TCP cubic registered
NET: Registered protocol family 1
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
eth0: Link down.
IP-Config: Complete:
    device=eth0, addr=192.168.0.158, mask=255.255.255.0, gw=192.168.0.1,
    host=olimex, domain=, nis-domain=(none),
    bootserver=192.168.0.1, rootserver=192.168.0.1, rootpath=
Looking up port of RPC 100003/2 on 192.168.0.1
Looking up port of RPC 100005/1 on 192.168.0.1
VFS: Mounted root (nfs filesystem).
```

```
BusyBox v1.00 (2008.10.07-02:27+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.
```

```
#
#
# ls
bin    dev    etc    home  lib    mnt    proc  sbin  tmp    usr    var
```


Addera en applikation till uClinux

Olimex LPC-2478-STK

Kjell Enblom

- För att lägga till en ny applikation till uClinux behöver man skapa en katalog och lägga till några rader i ett par filer.
- Om projektet heter foo så är de berörda filerna följande filer och kataloger:
 - Katalogen
~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/user/foo/
och dess projektfiler.
 - En makefil i katalogen foo.
 - En rad för projektet foo i makefilen ~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/user/Makefile
 - En kort hjälptext i ~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/config/Configure.help. Denna hjälptext är till för konfigurationen av uClinux.
 - En, eller flera, rader i filen ~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/config/config.in för att kunna välja att bygga projektet när man bygger om uClinux.

Addera en applikation till uClinux

Olimex LPC-
2478-STK

Kjell Enblom

- Skapa katalogen
~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/user/foo/
och kopiera in projektfilerna dit.
- Skapa en Makefile i projektkatalogen. Nedan visas ett exempel:

```
EXEC = foo
OBJS = foo.o

all: $(EXEC)

$(EXEC): $(OBJS)
    $(CC) $(LDFLAGS) -o $@ $(OBJS) $(LDLIBS)

romfs:
    $(ROMFSINST) /bin/$(EXEC)

clean:
    -rm -f $(EXEC) *.elf *.gdb *.o
```

Addera en applikation till uClinux

Olimex LPC-
2478-STK

Kjell Enblom

- Om projektet består av flera programfiler som ska byggas bör Makefile se ut enligt nedan:

```
EXECS = foo bar
OBJS = foo.o bar.o

all: $(EXECS)

$(EXECS): $(OBJS)
    $(CC) $(LDFLAGS) -o $@ $@.o $(LDLIBS)

romfs:
    $(ROMFSINST) -e CONFIG.USER.FOO.FOO           /bin/foo
    $(ROMFSINST) -e CONFIG.USER.FOO.BAR           /bin/bar
```

- Sjäklart kan makefilen vara mer komplex än de två ovanstående enkla exemplen.

Addera en applikation till uClinux

Olimex LPC-2478-STK

Kjell Enblom

- Lägg till en rad i `~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/user/Makefile` för projektet `foo`.

```
dir_$ (CONFIG_USER_FOO_FOO) += foo
```

- Denna rad adderar katalogen `foo` till det som ska byggas.
- Raderna med projektkataloger för projekt som ska byggas behöver inte vara i bokstavsordning men det ger en bättre ordning.

```
dir_$ (CONFIG_USER_FNORD_HTTPD) += fnord
dir_$ (CONFIG_USER_FLASHW_FLASHW) += flashw
dir_$ (CONFIG_USER_FLATFSD_FLATFSD) += flatfsd
dir_$ (CONFIG_USER_FLTHDR_FLTHDR) += flthdr
dir_$ (CONFIG_USER_FOO_FOO) += foo
dir_$ (CONFIG_USER_FREESWAN) += freeswan
dir_$ (CONFIG_USER_FROB_LED_FROB_LED) += frob-led
dir_$ (CONFIG_USER_FROX_FROX) += frox
dir_$ (CONFIG_USER_FSWCERT_FSWCERT) += fswcert
dir_$ (CONFIG_USER_FTP_FTP_FTP) += ftp
```

Addera en applikation till uClinux

Olimex LPC-2478-STK

Kjell Enblom

- Lägg till en hjälptext i filen `~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/config/Configure.help` för projektet `foo`.

```
CONFIG_USER_FCONFIG_FCONFIG
```

```
A program that lets you manipulate your RedBoot  
configuration from Linux.
```

```
CONFIG_USER_FOO_FOO
```

```
This program does fooley things to your bars.
```

```
CONFIG_USER_GETTYD_GETTYD
```

```
Another getty program.  
Approx. binary size: 16k
```

- Notera att textraderna för hjälptexten måste inledas med exakt 2 mellanslag.
- Alla rader måste vara kortare än 70 tecken.
- Tomrader är inte tillåtna.

Addera en applikation till uClinux

Olimex LPC-
2478-STK

Kjell Enblom

- Slutligen behövs en eller flera rader i filen `~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/config/config.in` så att det går att välja att bygga projektet när man gör **make menuconfig** för att konfigurera uClinux.
- Exempel:

```
bool 'foo'                                CONFIG_USER_FOO_FOO
```

- Lägg till denna rad under lämplig menykategori exempelvis 'Miscellaneous Applications'

```
mainmenu_option next_comment
comment 'Miscellaneous Applications'

bool '7za'                                CONFIG_USER_P7ZIP_7ZA
bool 'a60'                                CONFIG_USER_LANG_A60
if [ "$CONFIG_USER_LANG_A60" = "y" ]; then
    bool 'examples'                        CONFIG_USER_LANG_A60_EGS
    bool 'tests'                           CONFIG_USER_LANG_A60_TEST
fi
bool 'flthdr'                              CONFIG_USER_FLTHDR_FLTHDR
bool 'foo'                                  CONFIG_USER_FOO_FOO
bool 'frob-led'                             CONFIG_USER_FROB_LED_FROB_LED
bool 'gdbreplay'                           CONFIG_USER_GDB_GDBREPLAY
```

Addera en applikation till uClinux

Olimex LPC-
2478-STK

Kjell Enblom

- Upprepningen av FOO i CONFIG_USER_FOO_FOO är ifall projektet består av flera program som ska kompileras.
- Om projektet foo har programmen foo och bar som kan byggas var för sig kan man vilja ha separata rader i konfigurationen för dessa.

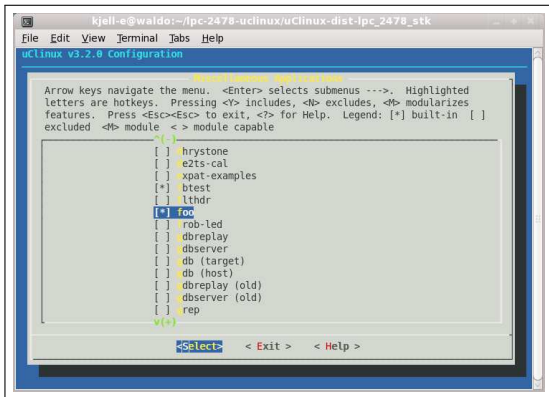
```
bool 'foo'          CONFIG_USER_FOO_FOO
bool 'bar'          CONFIG_USER_FOO_BAR
```

- Gå till katalogen
~/lpc-2478-uclinux/uClinux-dist-lpc_2478_stk/ och kör
make menuconfig.
- Gå in på 'Kernel/Library/Defaults Selection' och aktivera
'Customize Vendor/User Settings'
- När menuconfig avslutas kommer den att gå vidare till
konfigurationen för applikationer.
- I den nya konfigurationen gå in på 'Miscellaneous
Applications' och aktivera det nya projektet.

Addera en applikation till uClinux

Olimex LPC-2478-STK

Kjell Enblom



- Efter att konfigurationen är klar bygg uClinux med **make menuconfig**
- Det färdiga programmet/programmen finns nu i filen romfs_5.img och i katalogen romfs/bin.

Kompilera program till kortet utanför uClinux

Olimex LPC-
2478-STK

Kjell Enblom

- För att kunna kompilera program för Olimexkortet utan att använda uClinuxs byggsystem behöver man ändra sin PATH till att innehålla följande:

```
PATH=~/.lpc-2478-uclinux/uClinux-dist-lpc-2478-stk/tools:\  
~/lpc-2478-uclinux/snapgear-cross/usr/local/bin/:$PATH
```

- Se sedan till att definiera CC, CXX, CFLAGS, CXXFLAGS, LDFLAGS etc i makefilen enligt följande:

```
CC=ucfront-gcc arm-linux-gcc
```

```
CXX=ucfront-g++ arm-linux-g++
```

```
CFLAGS=-Os -g -fomit-frame-pointer -pipe \  
-msoft-float -fno-common -fno-builtin -Wall \  
-DEMBED -D__PIC__ -fpic -msingle-pic-base \  
-Dlinux -D__linux__ -Dunix -D__uClinux__
```

```
LDFLAGS=-Wl,--fatal-warnings -Wl,-elf2flt \  
-msoft-float -D__PIC__ -fpic -msingle-pic-base
```

```
CXXFLAGS=-Os -g -fomit-frame-pointer -pipe \  
-msoft-float -fno-common -fno-builtin -Wall \  
-DEMBED -D__PIC__ -fpic -msingle-pic-base \  
-Dlinux -D__linux__ -Dunix -D__uClinux__ \  
-nostdinc++ -fno-exceptions
```

Kompilera program till kortet utanför uClinux

Olimex LPC-
2478-STK

Kjell Enblom

■ Exempel på en Makefile, här utan g++.

```
CC=ucfront-gcc arm-linux-gcc

CFLAGS=-Os -g -fomit-frame-pointer -pipe \
-msoft-float -fno-common -fno-builtin -Wall \
-DEMBED -D__PIC__ -fpic -msingle-pic-base \
-Dlinux -D__linux__ -Dunix -D__uClinux__

LDFLAGS=-Wl,--fatal-warnings -Wl,-elf2flt \
-msoft-float -D__PIC__ -fpic -msingle-pic-base

EXEC = foo
OBJS = foo.o

all: $(EXEC)

$(EXEC): $(OBJS)
$(CC) $(LDFLAGS) -o $@ $(OBJS) $(LDLIBS)

romfs:
$(ROMFSINST) /bin/$(EXEC)

clean:
-rm -f $(EXEC) *.elf *.gdb *.o *
```

- I/O är minnesmappad på Olimex LPC2478-STK.
- Ett register styr PIN-funktion, d.v.s vilken funktion en viss pinne ska ha (PINSEL).
- Andra register styr datariktning (in eller ut). (IODIR, FIODIR)
- Andra register används för att sätta en utgång till 0 respektive 1 (skriva data). (IOSET, IOCLR, FIOSET, FIOCLR).
- För att läsa data används in-registren (IOPIN, FIOPIN).
- Alla registren är 32 bitar stora.

Lysdiodstyrning via GPIO

Olimex LPC-
2478-STK

Kjell Enblom

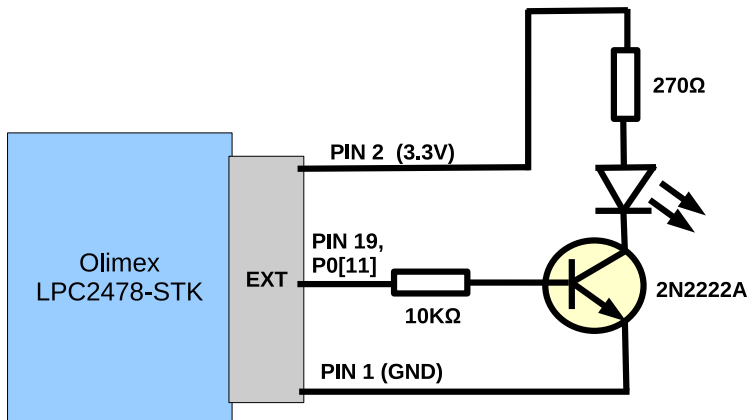
- För att t.ex. ställa om pinne 100 på LPC2478 (pin 19 på EXT-kontakten) behöver man göra följande:
 - Sätta bit 22 och 23 på adress 0xE002C000 (PINSEL0). Detta för att ställa in den till att vara P0[11]. Samma pinne kan vara RXD2/SCL2/MAT3[1].
 - Sätta bit 11 på adress 0xE0028008 (IOODIR) för att deklarerera att P0[11] ska vara en utgång.
- Utgången sätts till 1 genom att skriva 1 till bit 11 på adress 0xE0028004 (IOOSET).
- Utgången sätts utgången till 0 genom att skriva 1 till bit 11 på adress 0xE002800C (IOOCLR).
- För mer detaljer se dokumentationen `user.manual.lpc24xx.pdf`.

Lysdiodstyrning via GPIO

Olimex LPC-
2478-STK

Kjell Enblom

Exempel på koppling för att blinka med en lysdiod.



blink.c - program för att blinka med en lysdiod

Olimex LPC-
2478-STK

Kjell Enblom

```
/* Flash a LED connected to PIN 19, P0[11], on EXT connector */
/* Olimex LPC2478-STK */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <inttypes.h>

/* PINSEL0 Pin Control */
/* PINSEL0 register controls the functions of the pins */
#define PINSEL0 ((uint32_t volatile *)0xE002C000)

/* Register is used to define data direction */
#define IOODIR ((uint32_t volatile *)0xE0028008)

/* Used to set data bits */
#define IO0SET ((uint32_t volatile *)0xE0028004)

/* Used to clear data bits */
#define IO0CLR ((uint32_t volatile *)0xE002800C)

void ledInit(void);
void blink();
void toggle_led(void);
```

blink.c, forts.

Olimex LPC-
2478-STK

Kjell Enblom

```
int main(int argc, char ** argv)
{
    printf("\n"); /* DO NOT REMOVE THIS LINE */
    ledlnit();
    blink();
    exit(EXIT.SUCCESS);
}

void ledlnit(void)
{
    /* initialize LED, P0[11] on Olimex LPC2478-STK card */
    PINSEL0 &= ~((1 << 22) | (1 << 23)); /* Set bit 22:23 to 0 to define
        pin as GPIO p0[11] */
    IO0DIR |= (1 << 11); /* Declare P0.11 as output */
}

void blink()
{
    int i;
    for(i = 0; i < 121; i++)
    {
        toggle_led();
        usleep(100000);
    }
}
```

blink.c, forts.

Olimex LPC-
2478-STK

Kjell Enblom

```
void toggle_led(void)
{
    /* toggle LED, P0[11] on Olimex LPC2478-STK card */

    /* IOSET GPIO Port Output Set register. This register controls the */
    /* state of R/W 0x0 IOSET - 0xE002 8004 output pins in conjunction */
    /* with the IOCLR register. Writing ones IO1SET - 0xE002 8014 produces */
    /* highs at the corresponding port pins. Writing zeroes has no effect. */

    /* Set pin P0[11] to 0 when IO0CLR is 1 */
    /* Set pin Po[11] to 1 when IO0SET is 1 */

    static int LED_ON = 1;

    if (LED_ON) {
        LED_ON = 0;
        IO0CLR |= (1 << 11);
    }
    else {
        LED_ON = 1;
        IO0SET |= (1 << 11);
    }
}
```


Fast GPIO

Olimex LPC-
2478-STK

Kjell Enblom

- FIO0-FIO4 styrs av:
 - PINSEL, exempel: PINSEL9 för P4.
 - FIODIR, exempel: FIO4DIR för P4.
 - FIOSET, exempel: FIO4SET för P4.
 - FIOCLR, exempel: FIO4CLR för P4.

- Notera att det står fel i `user.manual.lpc24xx.pdf` i tabell 159 på sidan 198. Där står:
 - FIO0DIR - 0x3FFF C000
 - FIO1DIR - 0x3FFF C020
 - FIO2DIR - 0x3FFF C040
 - FIO2DIR - 0x3FFF C060
 - FIO2DIR - 0x3FFF C080

- De två sista ska vara FIO3DIR och FIO4DIR.

Blinka med lysdiod på P4[31]

Olimex LPC-
2478-STK

Kjell Enblom

- För att blinka med en lysdiod på P4[31], pin 12 på EXT-kontakten behövs följande deklarerationer:

```
#define PINSEL9      (*((uint32_t volatile *)0xE002C024))  
#define FIO4DIR     (*((uint32_t volatile *)0x3FFFC080))  
#define FIO4SET     (*((uint32_t volatile *)0x3FFFC098))  
#define FIO4CLR     (*((uint32_t volatile *)0x3FFFC09C))
```

- bit 30 och 31 på PINSEL9 ska sättas till 0:

```
PINSEL9 &= ~((1 << 30) | (1 << 31));
```

- Bit 31 på FIO4DIR sätts till 1 för att deklarera den som utgång:

```
FIO4DIR |= (1 << 31);
```

blink2.c

Motsvarande funktioner för blink2.c blir:

```
#define PINSEL9          (*((uint32_t volatile *)0xE002C024))
#define FIO4DIR         (*((uint32_t volatile *)0x3FFFC080))
#define FIO4SET         (*((uint32_t volatile *)0x3FFFC098))
#define FIO4CLR         (*((uint32_t volatile *)0x3FFFC09C))

void ledInit(void)
{
    /* initialize LED, P4[31] on Olimex LPC2478-STK card */
    PINSEL9 &= ~((1 << 30) | (1 << 31)); /* Set bit 30:31 to 0 to define
                                          pin as GPIO p4[31] */
    FIO4DIR |= (1 << 31); /* Declare P4.31 as output */
}

void toggle_led(void)
{
    /* toggle LED, P4[31] on Olimex LPC2478-STK card */
    /* Set pin P4[31] to 0 when IO0CLR is 1 */
    /* Set pin P4[31] to 1 when IO0SET is 1 */

    static int LED_ON = 1;

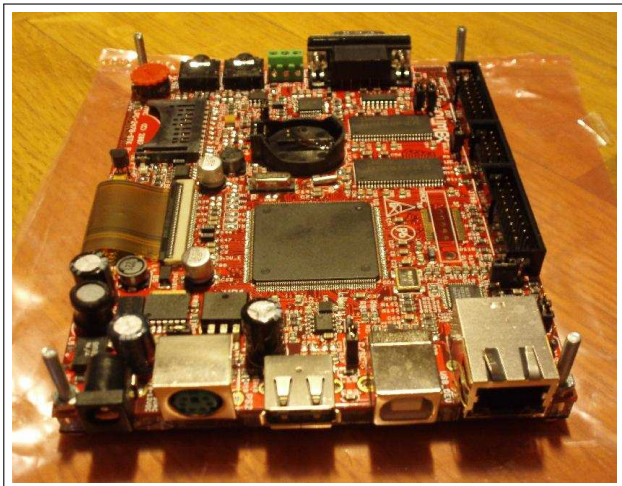
    if (LED_ON) {
        LED_ON = 0;
        FIO4CLR |= (1 << 31);
    }
    else {
        LED_ON = 1;
        FIO4SET |= (1 << 31);
    }
}
}
```

Bilder på Olimexkortet

Olimex LPC-
2478-STK

Kjell Enblom

Komponentsidan av Olimex LPC-2478-STK.

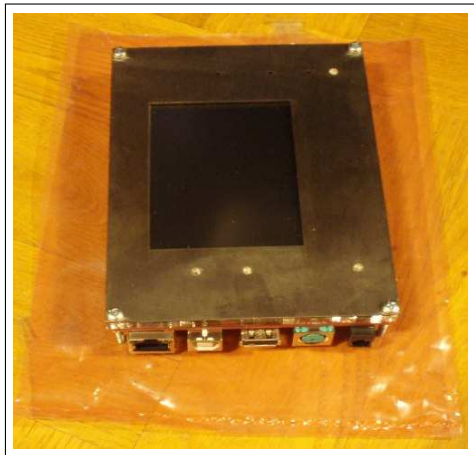


Bilder på Olimexkortet

Olimex LPC-
2478-STK

Kjell Enblom

Ovansidan (display-sidan).

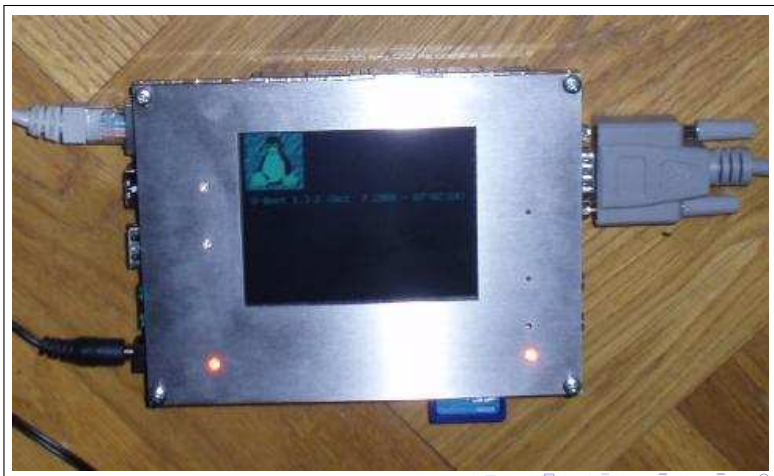


Bilder på Olimexkortet

Olimex LPC-
2478-STK

Kjell Enblom

U-Boot har laddats och visar en pingvin.

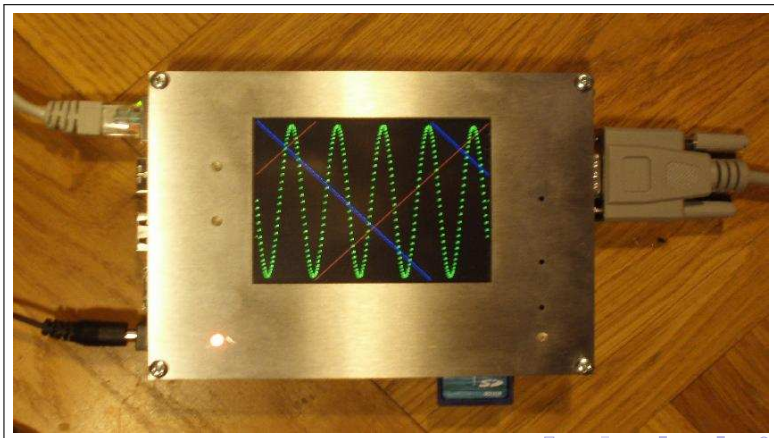


Bilder på Olimexkortet

Olimex LPC-
2478-STK

Kjell Enblom

Här körs testprogrammet fbtest som använder framebufferten för att visa grafiska kurvor.



Om du tänker skaffa ett eget Olimex LPC-2478-STK-kort behövs följande:

- Olimex LPC-2478-STK utvecklingskort.
- Spänningsaggregat/batterieliminatör 9V DC eller 12V DC.
- USB till serieportsadapter.
- Nollmodemkabel 9pol hona-hona.
- SD-kort.
- Nätverkssladd (TP-kabel).
- USB-minne.
- USB-sladdar, för att kunna ansluta kortet som en deviceenhet till en dator respektive för att kunna ansluta olika enheter till Olimexkortet.
- Hörlurar.
- Mikrofon.

- Linux Device Drivers, 3rd edition, Alessandro Rubini, O'Reilly online-version finns på <http://lwn.net/Kernel/LDD3/> (pdf)
- Building Embedded Linux Systems, Second Edition, Philippe Gerum, Karim Yaghmour, Jon Masters, Gilad Ben-Yossef, August 2008, O'Reilly.
- Programming Embedded Systems: With C and GNU Development Tools, 2nd Edition, Michael Barr och Anthony Massa, 2006, O'Reilly.
- Designing Embedded Hardware, John Catsoulis, 2005, O'Reilly.
- Embedded Linux Primer A Practical Real-World Approach, Christopher Hallinan, 2006, Prentice Hall. (2nd edition kommer ut januari 2010.)

- uClinux, <http://www.uclinux.org/>
- Manual till U-Boot, <http://www.denx.de/wiki/DULG/Manual>
- Manual till NXP LPC2478, <http://www.standardics.nxp.com/support/documents/microcontrollers/pdf/user.manual.lpc24xx.pdf>
- Embedded Linux Developer Forum, <http://www.ucdot.org/>
- SparkFun Electronics, <http://forum.sparkfun.com/>
- Busybox, <http://www.busybox.net>
- uClibc, <http://www.uclibc.org>
- uClinux for Linux Programmers (Linux Journal), <http://www.linuxjournal.com/article/7221>
- Linux on ARM Wiki, <http://www.linux-arm.com/>
- Arm, <http://www.arm.com/>

Frågor

Olimex LPC-
2478-STK

Kjell Enblom

- Frågor?