

# EOS Uebung 02

Aufgabe 1) Zeichnen des fertigen Systems:

Bootloader (mtdblock0)	0x0-0x0007FFFF
Kernel (mtdblock1)	0x00080000-0x0047FFFF
FileSystem (mtdblock2)	0x00480000-0x01FFFFFF

## Aufgabe 2)

a) Hochladen des Kernels auf das System:

```
host # cp /root/pxadev/images/uImage /root/pxadev/rootfs/home/root
target # dd if=/home/root/uImage of=/dev/mtdblock1
```

b) Erstellen des Filesystems über den angegebenen Befehl:

```
target # mkfs.ext2 /dev/mtdblock2
```

c) Füllen des Filesystems mit den Dateien:

```
target # mkdir /flashfs
target # mount /dev/mtdblock2 /flashfs
host # tar -xjf rootfs_min2.tbz2
host # cp -r rootfs /root/pxadev/rootfs/
target # cp -a /. /flashfs
```

d) Starten von Flash:

Beim Starten den Autobootvorgang stoppen und die Environment-Variable bootargs ändern:

```
printenv → bootargs root=/dev/nfs ip=:::::eth0: console=ttyS0,9600n8
setenv bootargs root=/dev/mtdblock2 console=ttyS0,9600n8
saveenv
boot
```

## Aufgabe 3)

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
```

```
int main()
{
    int gpio0 = open("/dev/gpio0", O_RDWR);

    if(gpio0 == 0)
    {
        printf("open of gpio0 did not work!! \n");
    }
}
```

```

        return -1;
    }

    char led_ctrl;

    while(1)
    {
        if(read(gpio0, &led_ctrl, 1) != sizeof(led_ctrl))
        {
            printf("read of gpio0 did not work!! \n");
            return -2;
        }

        if(write(gpio0, &led_ctrl, 1) != sizeof(led_ctrl))
        {
            printf("write of gpio0 did not work!! \n");
            return -3;
        }
    }
    return 0;
}

```

#### Aufgabe 4)

```
cp gpio_test /home/root/gpio
```

#### Aufgabe 5)

Wurde mit Erfolg ausgearbeitet wie in der Angabe definiert.