

Zadanie: Dokończyć konfigurację na karcie SD dla u-boota i kernela

1. Dodajemy kernela oraz devicetree na kartę SD:

```
$ cp <KERNEL_DIR>/arch/arm64/boot/dts/broadcom/bcm2711-rpi-4-b.dtb <MOUNT>/boot
$ cp <KERNEL_DIR>/arch/arm64/boot/Image <MOUNT>/boot
$ sync
```

2. Zawartość partycji boot powinna wyglądać tak:

```
$ ls <MOUNT>/boot/
bcm2711-rpi-4-b.dtb  bootcode.bin  config.txt  Image  start4.elf  u-boot.bin
```

3. Włączamy płytkę i wchodzimy do u-boota

4. Wykonujemy następujące polecenia:

```
fatload mmc 0:1 ${kernel_addr_r} Image
setenv bootargs "8250.nr_uarts=1 console=ttyS0,115200 root=/dev/mmcblk0p2 rw
rootwait earlyprintk init=/bin/sh"
booti ${kernel_addr_r} - ${fdt_addr}
```

Obserwujemy co się dzieje.

- Pauza -

Piszemy bootscript dla u-boota (co ma zrobić po wstaniu)

```
$ cat << EOF > boot_cmd.txt
fatload mmc 0:1 \${kernel_addr_r} Image
setenv bootargs "8250.nr_uarts=1 console=ttyS0,115200 root=/dev/mmcblk0p2 rw
rootwait earlyprintk init=/bin/sh"
booti \${kernel_addr_r} - \${fdt_addr}
EOF
```

Dodatkowe informacje: <https://u-boot.readthedocs.io/en/v2021.04/usage/booti.html>

Tworzymy z pliku txt plik który może przeczytać uboot

```
$ <UBOOT_DIR>/tools/mkimage -A arm64 -O linux -T script -C none -d boot_cmd.txt
boot.scr
```

Kopiujemy na partycję

```
$ sudo cp boot.scr /mnt/boot/
$ sync
```

Wsadzamy do portu SD na Raspberry i obserwujemy co pokaże się na serialu. System powinien załadować się automatycznie.

Wykorzystaj również komendę fatinfo.