

MXE11: run UNIX on a microcontroller

V1.71 (c) 2017-2018 Joerg Wolfram

1 Start and Stop

Overall, the image file contains 3 disk images. All images fit in 8 megabytes and thus on the AT45DB642 Dataflash. The first disk image (/dev/rk0) is always mounted to / (root directory), the other 2 images can only ever be mounted, preferably to /mnt. The mount binary is located in the /etc folder, so the call must be:

```
/etc/mount /dev/rk1 /mnt
```

and to un-mount:

```
/etc/umount /dev/rk1
```

At the controller level, there is a cache that is designed to prevent frequent write actions. The write cache is partly relatively small and buffers just 64 sectors with 64K controller RAM. But that's enough to keep periodic write accesses in the idle state from the storage medium.

After starting, a message briefly appears, during which you can enter the setup menu by pressing the space bar twice. Currently, only the (possibly existing) RTC can be set here. Later, other things like Image transfer via the serial Interface. Then the current version will be displayed and below it a prompt (@).

Here you have to specify the kernel to start. The used Mini-Unix can choose between 2 kernels. „rkmx” starts the unchanged original kernel, while „mx” starts my slightly adjusted one. In fact, it changes just the way how the date and time are determined at the start. While „mx” takes the current time of the RTC, the „rkmx” kernel reads the last modification time of the superblock on Disk1. In addition, the modified kernel can handle multiple terminals in the emulator.

During startup, the drive LED will light up steadily to turn off. If the medium was not found, the Drive LED flashes and the terminal displays **MXE11: NO DRIVE**. In this case, it must be controlled whether the flash medium is correctly connected or the SD card is contacted. Only the physical presence is affected checked the medium, not the content.

After that, the Sys LED is activated and tries to boot from the medium. while the first sector is read into the RAM and jumped to the beginning of the read memory areas. If there is no valid boot sector, in most cases the emulator stopps with a **User Halt** and the status LED starts flashing.

After successfull start You will get a login prompt. Login can be done as **root** or as user **mx**e, both with no password.

To shut down the PDP11 must be shut down. The command resp. the program of the same name **halt**, which first executes a sync() and then the write cache in the controller to the storage medium back. Finally, the status LED goes off and the power supply can be switched off.

Attention: Don **NEVER** disconnect power or reset the controller befor the system is shut down. Otherwise the file system probably will be corrupted and cannot be repaired.

```
#halt
Cleanup cache .....
MXE11: SYSTEM HALTED
```

2 Modified and additional programs

2.1 Die Shell (sh)

the shell (**sh**) currently contains 2 special features:

1. Apart from the usual Unix V6 **chdir** command, **cd** can be used
2. With the experimental **prompt** command (toggling) the current path will be showed at the command prompt

2.2 Shut down the system (halt)

This program comes from me and runs only on the MXE11. It is used to shut down the system. First a `sync()` is executed, then the magic value 0x1234 is written to the memory cell 0xFABC. This causes the emulator to clear the cache and stop the system. After a stop the status LED will be switched off.

2.3 MIPS Benchmark (mips)

This benchmark often runs the machine command **INC R4** one after the other. As parameter is indicated how many millions of times the command should be executed. If you enter here, for example 100 on, can be easily removed from the elapsed time to determine the MIPS value. For more accurate results, the command should be using **time** to be started.

2.4 Towers from Hanoi Benchmark (hanoi)

This benchmark rearranges a 10-pack stack of slices. The parameter indicates the number of sorts, a good value is 1000. For exact results, the command should be using **time** The number of loops per second is then $1000 / \text{measured time}$.