

U-Boot walk

[TS-WXL](#)

Take a walk because I entered the U-Boot prompt

Environment variable

```
Marvell>> printenv
bootargs=console=ttyS0,115200 mtdparts=physmapped-flash.0:32m(root)
baudrate=115200
loads_echo=0
ipaddr=192.168.11.150
serverip=192.168.11.1
rootpath=/mnt/ARM_FS/
netmask=255.255.255.0
fw_image_base=0x0
fw_image_size=0x0
console = console = ttyS0,115200
Casset = min
MALLOC_len = 1
ethprime = egiga0
bootargs_end=:::DB78xx0:eth0:none
standalone=fsload 0x20000000 ${image_name};setenv bootargs ${console} root=/dev/mtdblock0 rw ip=${ipaddr}:${serverip}${bootargs_end}; bootm 0x20000000;
ethmtu = 1500
eth1mtu = 1500
ethact = egiga0
ethaddr = 00: 1D: 73: E7: 23: B3
eth1addr = 00: 1D: 73: E7: 23: B4
buffalo_ver = BOOTVER = 2.08
buffalo_minor_ver=BOOT_MINOR_VER=1.00
build_time=11:57:53
initrd=initrd.buffalo
kernel=uImage.buffalo
bootargs_base=console=ttyS0,115200
bootargs_root=root=/dev/sda2 rw initrd=0x00900040,15M panic=5
def_tftp=tftp 0x02000000 ${kernel}; tftp 0x00900000 ${initrd}; setenv bootargs ${bootargs_base} ${bootargs_root} ${buffalo_ver} tftpboot=yes; bootm 0x02000000
EnableNandBoot=1
stdin=serial
stdout=serial
stderr=serial
mainlineLinux=no
enaMonExt = no
enaFlashBuf=yes
enaCpuStream=yes
enaWrAllo = no
enaFPU=yes
pexMode = RC
disL2Cache = no
setL2CacheWT=no
disL2Prefetch=yes
setL2Size256K=no
disL2Ecc = no
sata_dma_mode=yes
netbsd_en = no
vxworks_en=no
bootdelay=3
disaMvPnp=no
usb0Mode=host
usb1Mode=host
usb2Mode=host
filesize=8DD935
bootcmd= ext2load ide 0:1 0x02000000 /${kernel}; ext2load ide 0:1 0x00900000 /${initrd}; setenv bootargs ${bootargs_base} ${bootargs_root} ${buffalo_ver};

Environment size: 1504/4092 bytes
Marvell>>
```

help

```
Marvell>> help
?          - alias for 'help'
FSdel      - del a file from the Flash MV FS
FSdir      - ls the Flash MV FS
FSformat   - format the Flash MV FS
FSrun      - Load an exe file from the Flash MV FS and run it
FSstftp    - tftp a file to the Flash MV FS
FSftptpe   - tftp an exe file to the Flash MV FS
FStype     - cat file from the Flash MV FS
autoscr    - run script from memory
base       - print or set address offset
bdfinfo    - print Board Info structure
boot       - boot default, i.e., run 'bootcmd'
bootd      - boot default, i.e., run 'bootcmd'
bootext2   dev:boot_part1,boot_part2 addr boot_image linux_dev_name
bootm      - boot application image from memory
bootp      - boot image via network using BootP/TFTP protocol
bubt       - Burn an image on the Boot Flash.
chpart     - change active partition
cmp        - memory compare
```

```

cmpm    - Compare memory
coninfo - print console devices and information
cp      - memory copy
-----
cpumap  - Display CPU memory mapping settings.
crc32   - checksum calculation
date    - get/set/reset date & time
dclk    - Display the MV device CLKs.
ddimm   - Display SPD Dimm Info
dimer_test  change blightness test
diskboot- boot from IDE device
disp_test  LCD TEST
dma      - Perform DMA
echo     - echo args to console
eeprom  - EEPROM sub-system
erase    - erase FLASH memory
ext2load- load binary file from a Ext2 filesystem
ext2ls   - list files in a directory (default /)
fi       - Find value in the memory.
flinfo   - print FLASH memory information
fsinfo   - print information about filesystems
fsload   - load binary file from a filesystem image
fun_test
g        - start application at cached address 'addr'(default addr 0x40000)
gettemp
go       - start application at address 'addr'
help     - print online help
icrc32   - checksum calculation
ide      - IDE sub-system
iloop    - infinite loop on address range
imd      - i2c memory display
iminfo   - print header information for application image
imls     - list all images found in flash
imm      - i2c memory modify (auto-incrementing)
imw      - memory write (fill)
inm      - memory modify (constant address)
iprobe   - probe to discover valid I2C chip addresses
ir       - reading and changing MV internal register values.
itest    - return true/false on integer compare
led_test - loop LED
loadb    - load binary file over serial line (kermit mode)
loads    - load S-Record file over serial line
loop     - infinite loop on address range
ls       - list files in a directory (default /)
map      - Display address decode windows
md       - memory display
me       - PCI master enable
miconput - put data to mincon
mm       - memory modify (auto-incrementing)
mp       - map PCI BAR
mtdparts- define flash/nand partitions
mtest    - simple RAM test
mw       - memory write (fill)
nand     - NAND sub-system
nand_test - nand test
nboot    - boot from NAND device
nfs      - boot image via network using NFS protocol
nm       - memory modify (constant address)
pci      - list and access PCI Configuration Space
phyRead  - Read PCI-E Phy register
pciePhyWrite - Write PCI-E Phy register
phyRead  - Read Phy register
phyWrite - Write Phy register
ping     - send ICMP ECHO_REQUEST to network host
printenv- print environment variables
protect  - enable or disable FLASH write protection
pset hdd(led) [No.] on(off)
rarpboot- boot image via network using RARP/TFTP protocol
reset    - Perform RESET of the CPU
resetenv - Return all environment variable to default.
run      - run commands in an environment variable
saveenv  - save environment variables to persistent storage
se       - PCI Slave enable
POWER_STATUS_REBOOTING          1          0x18
setenv   - set environment variables
sflash  - read, write or erase the external SPI Flash.
sg       - scanning the PHYs status
signature - Get and display disk signature.
sleep    - delay execution for some time
sp       - Scan PCI bus.
spdcpy   - Copy Dimm 0 SPD to Dimm 1 SPD
tftpboot- boot image via network using TFTP protocol
version  - print monitor version
Marvell>>

```

LCD TEST

```

Marvell>> disp_test
temprature get start
<<do_disp_test>>
finish .... Push Disp SW
Marvell>>

```

```

On the liquid crystal,
DISP0 0000000000000000
0000000000000000000000
↓

```

DISP1 1111111111111111
11111111111111111111

~
DISP4 4444444444444444
44444444444444444444
.

led_test

```
Marvell>> led_test
LED test start
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2c0001
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2b0002
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 290004
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 250008
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 1d0010
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = d0020
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ed0040
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ad0080
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2c0100
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2b0200
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 290400
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 250800
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 1d1000
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = d2000
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ed4000
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ad8000
mcon_led_blink    = 230009
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2c0001
mcon_led_blink    = 2b0001
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2b0002
mcon_led_blink    = 2a0002
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 290004
mcon_led_blink    = 280004
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 250008
mcon_led_blink    = 240008
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 1d0010
mcon_led_blink    = 1c0010
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = d0020
mcon_led_blink    = c0020
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ed0040
mcon_led_blink    = ec0040
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ad0080
mcon_led_blink    = ac0080
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2c0100
mcon_led_blink    = 2b0100
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 2b0200
mcon_led_blink    = 2a0200
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 290400
mcon_led_blink    = 280400
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 250800
mcon_led_blink    = 240800
```

```
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = 1d1000
```

```
mcon_led_blink    = 1c1000
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = d2000
mcon_led_blink    = c2000
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ed4000
mcon_led_blink    = ec4000
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ad8000
mcon_led_blink    = ac8000
finish
Marvell>> CPU = MV78
```

Checking DATA BUS OK!
Checking ADDRESS BUS OK!

Reboot started after LED and LCD flicker

If you try again,

```
      :
      :
mcon_led_cpu_mcon = 30ffff
mcon_led_on_off   = ad8000
mcon_led_blink    = ac8000
finish
Marvell>>
```

This time, do not reboot.
Maybe watchdog timer

fun_test

```
Marvell>> fun_test
Fun test start
fun speed = 4c00
fun rpm   = 20
fun speed = 4b01
fun rpm   = 920
fun speed = 4a02
fun rpm   = 1380
fun speed = 4903
fun rpm   = 1910
finish
Marvell>>
```

At first it became quiet, then it became increasingly noisy.

Place

Left for about 5 minutes, no sign of falling.
What was the previous reboot? ??

When I was preparing to blog, I rebooted.



TS-WXL

[Rakuten Ichiba](#)
[Amazon](#)
[Yahoo Shopping](#)
[Livedoor Department Store](#)

[← Enable serial console even after reboot](#)

[Hack of record](#)
[LinkStation / KuroBox trying to hack](#)

[→ ar of standard firmware \(HDD root\)](#)