Dreamplug Setup

These directions work on:

- a **Version 10** Dreamplug using
  Uboot 2011.06 (October 15 2011 - 02:02:08) Marvell-Dreamplug
- a **Version 12** Dreamplug using
  Uboot 2011.06 (October 15 2011 - 02:02:08) Marvell-Dreamplug

Dream Plug Version:

> *Date: 2012-03-22*
> *Globalscale Technologies, Inc.*
>
> *To find the correct version of software drivers, please first*
>
> *1) Identify your version of the Dreamplug via the white serial label affixed to the back of the unit. On this label, there is a 4 number sequence such as 1001 or 0901 in small block. The former refers to Version 10 Revision 1 and the latter is Version 9 Revision 1.*
>
> *2) After identifying the version of the Dreamplug, then download the appropriate version Wifi/Kernel/RTFS and associated drivers. V9 Dreamplugs can only use the V9 software payloads.*

To access the console using a JTAG (uart cable and miniUSB cable)

> *minicom –s*
> *Set the Configure properties as follows:*
> *Bits per sec field to 115200*
> *Data bits to 8*
> *Parity to None*
> *Stop bit to 1*
> *Flow Control to None*

Create an SD using an image:

To get an image from the micro sd card, undo screws at base of dplug (under the feet), and unlock microsd.

> *dd if=/dev/[microsd device] conv=sync,noerror bs=64K /[directory to image]/dreamplug.[date].img.gz*
> *and to copy it to another SD card (micro or regular)*
> *dd if=/[directory to image]/dreamplug.img.gz of=/dev/[device] conv=sync,noerror bs=64K*

A clean image of the microsd card can be downloaded from:
ftp://128.128.250.51/pub/dplug/

If there are problems with the SD card, try zero-ing it out, then re-dd-ing :), the image

> *dd if=/dev/zero of=/dev/sdb*

To change the Uboot:

Access the Uboot environment by hitting any key while is it counting down:
"Hit any key to stop autoboot"

The original environment is:

```
Marvell>> printenv
bootargs=console=ttyS0,115200 rw root=/dev/sda2 rootdelay=10
bootcmd=setenv ethact egiga0; ${x_bootcmd_ethernet}; setenv ethact egiga1; ${x_bootcmd_ethernet};
${x_bootcmd_usb}; ${x_bootcmd_kernel}; bootm 0x6400000;
bootdelay=3
baudrate=115200
```

```
x_bootcmd_ethernet=ping 192.168.2.1
x_bootcmd_usb=usb start
ethact=egiga0
x_bootcmd_kernel=ext2load usb 0 0x6400000 uImage
ethaddr=F0:AD:4E:01:42:A0
eth1addr=F0:AD:4E:01:42:A1
stdin=serial
stdout=serial
stderr=serial

Environment size: 460/4092 bytes
```

This needs to be changed to:

```
Marvell>> printenv
bootdelay=3
baudrate=115200
ethact=egiga0
ethaddr=F0:AD:4E:01:42:A0
eth1addr=F0:AD:4E:01:42:A1
bootcmd=run clear_kernel_in_mem; run bootcmd_sdb; run bootcmd_sda
clear_kernel_in_mem=echo Purging kernel in memory; mw 0x6400000 0x0 0x300000
bootcmd_sdb=setenv bootargs console=ttyS0,115200 root=/dev/sdb2 panic=10 rootwait; usb start; run boot_sdb;
boot_sdb=ext2load usb 1 0x6400000 uImage; bootm 0x6400000;
bootcmd_sda=setenv bootargs console=ttyS0,115200 root=/dev/sda2 panic=10 rootwait; usb start; run boot_sda;
boot_sda=ext2load usb 0 0x6400000 uImage; bootm 0x6400000;
stdin=serial
stdout=serial
```

The following command will make the appropriate changes. They can be cut and pasted into the Uboot environment.

First remove unneeded commands:

```
setenv x_bootcmd_usb
setenv x_bootcmd_ethernet
setenv bootargs
setenv x_bootcmd_kernel
setenv bootcmd
```

New Uboot commands:

```
setenv clear_kernel_in_mem 'echo Purging kernel in memory; mw 0x6400000 0x0 0x300000'
setenv boot_sda ext2load 'usb 0 0x6400000 uImage; bootm 0x6400000'
setenv bootcmd_sda 'setenv bootargs console=ttyS0,115200 root=/dev/sda2 panic=10 rootwait; usb start; run boot_sda'
setenv boot_sdb 'ext2load usb 1 0x6400000 uImage; bootm 0x6400000'
setenv bootcmd_sdb 'setenv bootargs console=ttyS0,115200 root=/dev/sdb2 panic=10 rootwait; usb start; run boot_sdb'
setenv bootcmd 'run clear_kernel_in_mem; run bootcmd_sdb; run bootcmd_sda'
```

You can test your changes by typing: boot at the Uboot command line, you will need to redo your changes and then save them in the Uboot environment.

Type saveenv or the changes will not stick through reboots..

Uboot Environment Commands

Set environment variables:

SETENV

- set environment variable 'name' to 'value ...'
setenv name
- delete environment variable 'name'
To modify the U-Boot environment you have to use the setenv command. When called with exactly one argument, it will delete any variable of that name from U-Boot's environment, if such a variable exists. Any storage occupied for such a variable will be automatically reclaimed:

```
=> printenv foo
foo=This is an example value.
=> setenv foo
=> printenv foo
## Error: "foo" not defined
=>
```

When called with more arguments, the first one will again be the name of the variable, and all following arguments will (concatenated by single space characters) form the value that gets stored for this variable. New variables will be automatically created, existing ones overwritten.

```
=> printenv bar
## Error: "bar" not defined
=> setenv bar This is a new example.
=> printenv bar
bar=This is a new example.
=>
```

Remember standard shell quoting rules when the value of a variable shall contain characters that have a special meaning to the command line parser (like the $ character that is used for variable substitution or the semicolon which separates commands). Use the backslash (\) character to escape such special characters.

```
=> setenv cons_opts console=tty0 console=ttyS0,\$(baudrate)
=> printenv cons_opts
cons_opts=console=tty0 console=ttyS0,$(baudrate)
=>
```

There is no restriction on the characters that can be used in a variable name except the restrictions imposed by the command line parser (like using backslash for quoting, space and tab characters to separate arguments, or semicolon and newline to separate commands). Even strange input like "=-/|()+=" is a perfectly legal variable name in U-Boot.
A common mistake is to write
setenv name=value
instead of
setenv name value
There will be no error message, which lets you believe everything went OK, but it didn't: instead of setting the variable name to the value you tried to delete a variable with the name name=value. This is probably not what you intended. Always remember that name and value have to be separated by space and/or tab characters.

SAVEENV
Save environment variables to persistent storage.
All changes you make to the U-Boot environment are made in RAM only. They are lost as soon as you reboot the system. If you want to make your changes permanent you have to use the saveenv command to write a copy of the environment settings to persistent storage, from where they are automatically loaded during startup:

```
=> saveenv
Saving Enviroment to Flash...
Un-Protected 1 sectors
Erasing Flash...
. done
Erased 1 sectors
Writing to Flash... done
Protected 1 sectors
```

=>