```
bootcmd=if test ${boot_fit} -eq 1; then run update_to_fit; fi; run findfdt; run
init_console; run envboot; run distro_bootcmd
bootdelay=0
baudrate=115200
arch=arm
cpu=armv7
board=am335x
board_name=am335x
vendor=ti
soc=am33xx

#CONFIG_EXTRA_ENV_SETTINGS
#DEFAULT_LINUX_BOOT_ENV
loadaddr=0x82000000
kernel_addr_r=0x82000000
fdtaddr=0x88000000
fdt_addr_r=0x88000000
rdaddr=0x88080000
ramdisk_addr_r=0x88080000
scriptaddr=0x80000000
pxefile_addr_r=0x80100000
bootm_size=0x10000000
boot_fdt=try

#DEFAULT_MMC_TI_ARGS
mmcdev=0
mmcrootfstype=ext4 rootwait
finduuid=part uuid ${devtype} ${bootpart} uuid
args_mmc=run finduuid;setenv bootargs console=${console}
    ${cape_uboot}
    root=PARTUUID=${uuid} ro
    rootfstype=${mmcrootfstype}
    ${uboot_detected_capes}
    ${cmdline}
args_mmc_old=setenv bootargs console=${console}
    ${optargs}
    ${cape_uboot}
    root=${oldroot} ro
    rootfstype=${mmcrootfstype}
    ${uboot_detected_capes}
    ${cmdline}
args_mmc_uuid=setenv bootargs console=${console}
    ${optargs}
    ${cape_uboot}
    root=UUID=${uuid} ro
    rootfstype=${mmcrootfstype}
    ${uboot_detected_capes}
    ${cmdline}
args_uenv_root=setenv bootargs console=${console}
    ${optargs}
    ${cape_uboot}
    root=${uenv_root} ro
    rootfstype=${mmcrootfstype}
    ${uboot_detected_capes}
    ${cmdline}
args_netinstall=setenv bootargs ${netinstall_bootargs}
    ${optargs}
    ${cape_uboot}
    root=gcdev/ram rw
    ${uboot_detected_capes}
    ${cmdline}
script=boot.scr
scriptfile=${script}
loadbootscript=load ${devtype} ${bootpart} ${loadaddr} ${scriptfile};
bootscript=echo Running bootscript from mmc${bootpart} ...;
    source ${loadaddr}
bootenvfile=uEnv.txt
bootenv=uEnv.txt
importbootenv=echo Importing environment from ${devtype} ...;
```

```
    env import -t ${loadaddr} ${filesize}
loadbootenv=load ${devtype} ${bootpart} ${loadaddr} ${bootenvfile}
loadimage=load ${devtype} ${bootpart} ${loadaddr} ${bootdir}gc${bootfile}
loadrd=load ${devtype} ${bootpart} ${rdaddr} ${bootdir}gc${rdfile}; setenv rdsize ${filesize}
loadfdt=echo loading ${fdtdir}gc${fdtfile} ...; load ${devtype} ${bootpart} ${fdtaddr} ${fdtdir}/${fdtfile}
loadoverlay=echo uboot_overlays: loading ${actual_uboot_overlay} ...;
    load ${devtype} ${bootpart} ${rdaddr} ${actual_uboot_overlay};
    fdt addr ${fdtaddr}; fdt resize ${fdt_buffer};
    fdt apply ${rdaddr}; fdt resize ${fdt_buffer};
virtualloadoverlay=if test -e ${devtype} ${bootpart} ${fdtdir}gcoverlays/${uboot_overlay}; then
            setenv actual_uboot_overlay ${fdtdir}gcoverlays/${uboot_overlay};
            run loadoverlay;
        else
            if test -e ${devtype} ${bootpart} gclib/firmware/${uboot_overlay}; then
                setenv actual_uboot_overlay gclib/firmware/${uboot_overlay};
                run loadoverlay;
            else
                if test -e ${devtype} ${bootpart} ${uboot_overlay}; then
                    setenv actual_uboot_overlay ${uboot_overlay};
                    run loadoverlay;
                else
                    echo uboot_overlays: unable to find [${devtype} ${bootpart} ${uboot_overlay}]...;
                fi;
            fi;
        fi;
failumsboot=echo; echo FAILSAFE: U-Boot UMS (USB Mass Storage) enabled, media now available over the usb slave port ...;
    ums 0 ${devtype} 1;
envboot=mmc dev ${mmcdev};
    if mmc rescan; then
        echo SDgcMMC found on device ${mmcdev};
        if run loadbootscript; then
            run bootscript;
        else
            if run loadbootenv; then
                echo Loaded env from ${bootenvfile};
                run importbootenv;
            fi;
            if test -n $uenvcmd; then
                echo Running uenvcmd ...;
                run uenvcmd;
            fi;
        fi;
    fi;
mmcloados=run args_mmc;
    if test ${boot_fdt} = yes || test ${boot_fdt} = try; then
        if run loadfdt; then
            if test -n ${uname_r}; then
                bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr};
            else
                bootz ${loadaddr} - ${fdtaddr};
            fi;
        else
            if test ${boot_fdt} = try; then
                bootz;
            else
                echo WARN: Cannot load the DT;
            fi;
        fi;
    else
        bootz;
    fi;
mmcboot=mmc dev ${mmcdev};
    setenv devnum ${mmcdev};
    setenv devtype mmc;
```

```
    if mmc rescan; then
        echo SDgcMMC found on device ${mmcdev};
        if run loadimage; then
            if test ${boot_fit} -eq 1; then
                run loadfit;
            else
                run mmcloados;
            fi;
        fi;
    fi;

#DEFAULT_FIT_TI_ARGS
boot_fit=0
fit_loadaddr=0x90000000
fit_bootfile=fitImage
update_to_fit=setenv loadaddr ${fit_loadaddr}; setenv bootfile ${fit_bootfile}
loadfit=run args_mmc; bootm ${loadaddr}#${fdtfile};

bootpart=0:2
bootdir=gcboot
bootfile=zImage
board_eeprom_header=undefined
fdtfile=undefined
console=ttyO0,115200n8
partitions=
    uuid_disk=${uuid_gpt_disk};
    name=bootloader,start=384K,size=1792K,
        uuid=${uuid_gpt_bootloader};
    name=rootfs,start=2688K,size=-,uuid=${uuid_gpt_rootfs}
optargs=
ramroot=gcdev/ram0 rw
ramrootfstype=ext2
spiroot=gcdev/mtdblock4 rw
spirootfstype=jffs2
spisrcaddr=0xe0000
spiimgsize=0x362000
spibusno=0
spiargs=setenv bootargs console=${console}
    ${optargs}
    root=${spiroot}
    rootfstype=${spirootfstype}
ramargs=setenv bootargs console=${console}
    ${optargs}
    root=${ramroot}
    rootfstype=${ramrootfstype}
loadramdisk=load mmc ${mmcdev} ${rdaddr} ramdisk.gz
spiboot=echo Booting from spi ...;
    run spiargs;
    sf probe ${spibusno}:0;
    sf read ${loadaddr} ${spisrcaddr} ${spiimgsize};
    bootz ${loadaddr}
pb_eeprom_hdr=
    mw 82001000 ee3355aa;
    mw 82001004 35333341;
    mw 82001008 4c474250
serverip=192.168.1.1
ipaddr=192.168.1.2
if_netconsole=ping $serverip
start_netconsole=
    setenv ncip $serverip;
    setenv bootdelay 10;
    setenv stdin serial,nc;
    setenv stdout serial,nc;
    setenv stderr serial,nc;
    version
preboot=run if_netconsole start_netconsole
eeprom_program=
    if test $board_eeprom_header = bbb_blank; then
        run eeprom_dump; run eeprom_blank; run eeprom_bbb_header; run eeprom_dump;
```

```
reset; fi;
    if test $board_eeprom_header = bbbl_blank; then
        run eeprom_dump; run eeprom_blank; run eeprom_bbb_header; run eeprom_bbbl_footer; run eeprom_dump; reset; fi;
    if test $board_eeprom_header = bbbw_blank; then
        run eeprom_dump; run eeprom_blank; run eeprom_bbb_header; run eeprom_bbbw_footer; run eeprom_dump; reset; fi;
    if test $board_eeprom_header = pocketbeagle_blank; then
        run eeprom_dump; run eeprom_blank; run eeprom_pocketbeagle; run eeprom_dump; reset; fi;
    if test $board_eeprom_header = bbgg_blank; then
        run eeprom_dump; run eeprom_blank; run eeprom_bbb_header; run eeprom_bbgg_footer; run eeprom_dump; reset; fi;
    if test $board_eeprom_header = beaglelogic_blank; then
        run eeprom_dump; run eeprom_blank; run eeprom_beaglelogic; run eeprom_dump; reset; fi;
ramboot=echo Booting from ramdisk ...;
    run ramargs;
    bootz ${loadaddr} ${rdaddr} ${fdtaddr}
findfdt=
    echo board_name=[$board_name] ...;
    if test $board_name = A335BLGC; then
        setenv fdtfile am335x-beaglelogic.dtb; fi;
    if test $board_name = A335BONE; then
        setenv fdtfile am335x-bone.dtb; fi;
    if test $board_name = A335BNLT; then
        echo board_rev=[$board_rev] ...;
        if test $board_rev = GH01; then
            setenv fdtfile am335x-boneblack.dtb;
        elif test $board_rev = BBG1; then
            setenv fdtfile am335x-bonegreen.dtb;
        elif test $board_rev = BP00; then
            setenv fdtfile am335x-pocketbone.dtb;
        elif test $board_rev = GW1A; then
            setenv fdtfile am335x-bonegreen-wireless.dtb;
        elif test $board_rev = GG1A; then
            setenv fdtfile am335x-bonegreen-gateway.dtb;
        elif test $board_rev = AIA0; then
            setenv fdtfile am335x-abbbi.dtb;
        elif test $board_rev = EIA0; then
            setenv fdtfile am335x-boneblack.dtb;
        elif test $board_rev = ME06; then
            setenv fdtfile am335x-bonegreen.dtb;
        elif test $board_rev = OS00; then
            setenv fdtfile am335x-osd3358-sm-red.dtb;
        else
            setenv fdtfile am335x-boneblack.dtb;
        fi;
    fi;
    if test $board_name = A335PBGL; then
        setenv fdtfile am335x-pocketbeagle.dtb; fi;
    if test $board_name = BBBW; then
        setenv fdtfile am335x-boneblack-wireless.dtb; fi;
    if test $board_name = BBG1; then
        setenv fdtfile am335x-bonegreen.dtb; fi;
    if test $board_name = BBGW; then
        setenv fdtfile am335x-bonegreen-wireless.dtb; fi;
    if test $board_name = BBGG; then
        setenv fdtfile am335x-bonegreen-gateway.dtb; fi;
    if test $board_name = BBBL; then
        setenv fdtfile am335x-boneblue.dtb; fi;
    if test $board_name = BBEN; then
        setenv fdtfile am335x-sancloud-bbe.dtb; fi;
    if test $board_name = OS00; then
        setenv fdtfile am335x-osd3358-sm-red.dtb; fi;
    if test $board_name = A33515BB; then
        setenv fdtfile am335x-evm.dtb; fi;
    if test $board_name = A335X_SK; then
        setenv fdtfile am335x-evmsk.dtb; fi;
```

```
265     if test $board_name = A335_ICE; then
266         setenv fdtfile am335x-icev2.dtb; fi;
267     if test $fdtfile = undefined; then
268         setenv board_name A335BNLT;
269         setenv board_rev EMMC;
270         setenv fdtfile am335x-bonegreen.dtb;
271     fi;
272 init_console=
273     if test $board_name = A335_ICE; then
274         setenv console ttyO3,115200n8;
275     elif test $board_name = A335BLGC; then
276         setenv console ttyO4,115200n8;
277     else
278         setenv console ttyO0,115200n8;
279     fi;
280 
281 #EEWIKI_NFS
282 server_ip=192.168.1.100
283 gw_ip=192.168.1.1
284 netmask=255.255.255.0
285 hostname=
286 device=eth0
287 autoconf=off
288 root_dir=gchome/userid/targetNFS
289 tftp_dir=
290 nfs_options=,vers=3
291 nfsrootfstype=ext4 rootwait fixrtc
292 nfsargs=setenv bootargs console=${console}
293     ${optargs}
294     ${cape_uboot}
295     root=gcdev/nfs rw
296     rootfstype=${nfsrootfstype}
297     nfsroot=${nfsroot}
298     ip=${ip}
299     ${cmdline}
300 nfsboot=echo Booting from ${server_ip} ...;
301     setenv nfsroot ${server_ip}:${root_dir}${nfs_options};
302     setenv ip ${client_ip}:${server_ip}:${gw_ip}:${netmask}:${hostname}:${device}:$-
    {autoconf};
303     setenv autoload no;
304     setenv serverip ${server_ip};
305     setenv ipaddr ${client_ip};
306     tftp ${loadaddr} ${tftp_dir}${bootfile};
307     tftp ${fdtaddr} ${tftp_dir}dtbsgc${fdtfile};
308     run nfsargs;
309     bootz ${loadaddr} - ${fdtaddr}
310 nfsboot_uname_r=echo Booting from ${server_ip} ...;
311     setenv nfsroot ${server_ip}:${root_dir}${nfs_options};
312     setenv ip ${client_ip}:${server_ip}:${gw_ip}:${netmask}:${hostname}:${device}:$-
    {autoconf};
313     setenv autoload no;
314     setenv serverip ${server_ip};
315     setenv ipaddr ${client_ip};
316     tftp ${loadaddr} ${tftp_dir}vmlinuz-${uname_r};
317     tftp ${fdtaddr} ${tftp_dir}dtbsgc${uname_r}/${fdtfile};
318     run nfsargs;
319     bootz ${loadaddr} - ${fdtaddr}
320 
321 #EEWIKI_BOOT
322 boot=${devtype} dev ${mmcdev};
323     if ${devtype} rescan; then
324         gpio set 54;
325         setenv bootpart ${mmcdev}:1;
326         if test -e ${devtype} ${bootpart} gcetc/fstab; then
327             setenv mmcpart 1;
328         fi;
329         echo Checking for: gcuEnv.txt ...;
330         if test -e ${devtype} ${bootpart} gcuEnv.txt; then
331             if run loadbootenv; then
```

```
                gpio set 55;
                echo Loaded environment from gcuEnv.txt;
                run importbootenv;
            fi;
            echo Checking if uenvcmd is set ...;
            if test -n ${uenvcmd}; then
                gpio set 56;
                echo Running uenvcmd ...;
                run uenvcmd;
            fi;
            echo Checking if client_ip is set ...;
            if test -n ${client_ip}; then
                if test -n ${dtb}; then
                    setenv fdtfile ${dtb};
                    echo using ${fdtfile} ...;
                fi;
                gpio set 56;
                if test -n ${uname_r}; then
                    echo Running nfsboot_uname_r ...;
                    run nfsboot_uname_r;
                fi;
                echo Running nfsboot ...;
                run nfsboot;
            fi;
        fi;
        echo Checking for: gc${script} ...;
        if test -e ${devtype} ${bootpart} gc${script}; then
            gpio set 55;
            setenv scriptfile ${script};
            run loadbootscript;
            echo Loaded script from ${scriptfile};
            gpio set 56;
            run bootscript;
        fi;
        echo Checking for: gcboot/${script} ...;
        if test -e ${devtype} ${bootpart} gcboot/${script}; then
            gpio set 55;
            setenv scriptfile gcboot/${script};
            run loadbootscript;
            echo Loaded script from ${scriptfile};
            gpio set 56;
            run bootscript;
        fi;
        echo Checking for: gcboot/uEnv.txt ...;
        for i in 1 2 3 4 5 6 7 ; do
            setenv mmcpart ${i};
            setenv bootpart ${mmcdev}:${mmcpart};
            if test -e ${devtype} ${bootpart} gcboot/uEnv.txt; then
                gpio set 55;
                load ${devtype} ${bootpart} ${loadaddr} gcboot/uEnv.txt;
                env import -t ${loadaddr} ${filesize};
                echo Loaded environment from gcboot/uEnv.txt;
                if test -n ${dtb}; then
                    echo debug: [dtb=${dtb}] ... ;
                    setenv fdtfile ${dtb};
                    echo Using: dtb=${fdtfile} ...;
                fi;
                echo Checking if uname_r is set in gcboot/uEnv.txt...;
                if test -n ${uname_r}; then
                    gpio set 56;
                    setenv oldroot gcdev/mmcblk${mmcdev}p${mmcpart};
                    echo Running uname_boot ...;
                    run uname_boot;
                fi;
            fi;
        done;
    fi;

#EEWIKI_UNAME_BOOT
```

```
uname_boot=
    setenv bootdir gcboot;
    setenv bootfile vmlinuz-${uname_r};
    if test -e ${devtype} ${bootpart} ${bootdir}gc${bootfile}; then
        echo loading ${bootdir}gc${bootfile} ...;
        run loadimage;
        setenv fdtdir gcboot/dtbs/${uname_r};
        echo debug: [enable_uboot_overlays=${enable_uboot_overlays}] ... ;
        if test -n ${enable_uboot_overlays}; then
            echo debug: [enable_uboot_cape_universal=${enable_uboot_cape_universal}] ... ;
            if test -n ${enable_uboot_cape_universal}; then
                echo debug: [uboot_base_dtb_univ=${uboot_base_dtb_univ}] ... ;
                if test -n ${uboot_base_dtb_univ}; then
                    echo uboot_overlays: [uboot_base_dtb=${uboot_base_dtb_univ}] ... ;
                    if test -e ${devtype} ${bootpart} ${fdtdir}gc${uboot_base_dtb_univ}; then
                        setenv fdtfile ${uboot_base_dtb_univ};
                        echo uboot_overlays: Switching too: dtb=${fdtfile} ...;
                        setenv cape_uboot bone_capemgr.uboot_capemgr_enabled=1;
                    else
                        echo debug: unable to find [${uboot_base_dtb_univ}] using [${uboot_base_dtb}] instead ... ;
                        echo debug: [uboot_base_dtb_univ=${uboot_base_dtb}] ... ;
                        if test -n ${uboot_base_dtb}; then
                            echo uboot_overlays: [uboot_base_dtb=${uboot_base_dtb}] ... ;
                            if test -e ${devtype} ${bootpart} ${fdtdir}gc${uboot_base_dtb}; then
                                setenv fdtfile ${uboot_base_dtb};
                                echo uboot_overlays: Switching too: dtb=${fdtfile} ...;
                            fi;
                        fi;
                    fi;
                fi;
            else
                echo debug: [uboot_base_dtb_univ=${uboot_base_dtb}] ... ;
                if test -n ${uboot_base_dtb}; then
                    echo uboot_overlays: [uboot_base_dtb=${uboot_base_dtb}] ... ;
                    if test -e ${devtype} ${bootpart} ${fdtdir}gc${uboot_base_dtb}; then
                        setenv fdtfile ${uboot_base_dtb};
                        echo uboot_overlays: Switching too: dtb=${fdtfile} ...;
                    fi;
                fi;
            fi;
        fi;
        if test -e ${devtype} ${bootpart} ${fdtdir}gc${fdtfile}; then
            run loadfdt;
        else
            setenv fdtdir gcusr/lib/linux-image-${uname_r};
            if test -e ${devtype} ${bootpart} ${fdtdir}gc${fdtfile}; then
                run loadfdt;
            else
                setenv fdtdir gclib/firmware/${uname_r}/device-tree;
                if test -e ${devtype} ${bootpart} ${fdtdir}gc${fdtfile}; then
                    run loadfdt;
                else
                    setenv fdtdir gcboot/dtb-${uname_r};
                    if test -e ${devtype} ${bootpart} ${fdtdir}gc${fdtfile}; then
                        run loadfdt;
                    else
                        setenv fdtdir gcboot/dtbs;
                        if test -e ${devtype} ${bootpart} ${fdtdir}gc${fdtfile}; then
                            run loadfdt;
                        else
```

```
                                setenv fdtdir gcboot/dtb;
                                if test -e ${devtype} ${bootpart} ${fdtdir}gc${fdtfile}; then
                                    run loadfdt;
                                else
                                    setenv fdtdir gcboot;
                                    if test -e ${devtype} ${bootpart} ${fdtdir}gc${fdtfile}; then
                                        run loadfdt;
                                    else
                                        if test -e ${devtype} ${bootpart} ${fdtfile}; then
                                            run loadfdt;
                                        else
                                            echo; echo unable to find [dtb=${fdtfile}] did you name it correctly? ...;
                                            run failumsboot;
                                        fi;
                                    fi;
                                fi;
                            fi;
                        fi;
                    fi;
                fi;
            fi;
            if test -n ${enable_uboot_overlays}; then
                setenv fdt_buffer 0x60000;
                if test -n ${uboot_fdt_buffer}; then
                    setenv fdt_buffer ${uboot_fdt_buffer};
                fi;
                echo uboot_overlays: [fdt_buffer=${fdt_buffer}] ... ;
                if test -n ${uboot_silicon}; then
                    setenv uboot_overlay ${uboot_silicon};
                    run virtualloadoverlay;
                fi;
                if test -n ${uboot_model}; then
                    setenv uboot_overlay ${uboot_model};
                    run virtualloadoverlay;
                fi;
                if test -n ${disable_uboot_overlay_adc}; then
                    echo uboot_overlays: uboot loading of [BB-ADC-00A0.dtbo] disabled by gcboot/uEnv.txt [disable_uboot_overlay_adc=1]...;
                else
                    setenv uboot_overlay BB-ADC-00A0.dtbo;
                    run virtualloadoverlay;
                fi;
                if test -n ${uboot_overlay_addr0}; then
                    if test -n ${disable_uboot_overlay_addr0}; then
                        echo uboot_overlays: uboot loading of [${uboot_overlay_addr0}] disabled by gcboot/uEnv.txt [disable_uboot_overlay_addr0=1]...;
                    else
                        setenv uboot_overlay ${uboot_overlay_addr0};
                        run virtualloadoverlay;
                    fi;
                fi;
                if test -n ${uboot_overlay_addr1}; then
                    if test -n ${disable_uboot_overlay_addr1}; then
                        echo uboot_overlays: uboot loading of [${uboot_overlay_addr1}] disabled by gcboot/uEnv.txt [disable_uboot_overlay_addr1=1]...;
                    else
                        setenv uboot_overlay ${uboot_overlay_addr1};
                        run virtualloadoverlay;
                    fi;
                fi;
                if test -n ${uboot_overlay_addr2}; then
                    if test -n ${disable_uboot_overlay_addr2}; then
                        echo uboot_overlays: uboot loading of [${uboot_overlay_addr2}] disabled by gcboot/uEnv.txt [disable_uboot_overlay_addr2=1]...;
                    else
```

```
522                     setenv uboot_overlay ${uboot_overlay_addr2};
523                     run virtualloadoverlay;
524                 fi;
525             fi;
526             if test -n ${uboot_overlay_addr3}; then
527                 if test -n ${disable_uboot_overlay_addr3}; then
528                     echo uboot_overlays: uboot loading of [${uboot_overlay_addr3}]
    disabled by gcboot/uEnv.txt [disable_uboot_overlay_addr3=1]...;
529                 else
530                     setenv uboot_overlay ${uboot_overlay_addr3};
531                     run virtualloadoverlay;
532                 fi;
533             fi;
534             if test -n ${uboot_overlay_addr4}; then
535                 setenv uboot_overlay ${uboot_overlay_addr4};
536                 run virtualloadoverlay;
537             fi;
538             if test -n ${uboot_overlay_addr5}; then
539                 setenv uboot_overlay ${uboot_overlay_addr5};
540                 run virtualloadoverlay;
541             fi;
542             if test -n ${uboot_overlay_addr6}; then
543                 setenv uboot_overlay ${uboot_overlay_addr6};
544                 run virtualloadoverlay;
545             fi;
546             if test -n ${uboot_overlay_addr7}; then
547                 setenv uboot_overlay ${uboot_overlay_addr7};
548                 run virtualloadoverlay;
549             fi;
550             if test -n ${uboot_emmc}; then
551                 if test -n ${disable_uboot_overlay_emmc}; then
552                     echo uboot_overlays: uboot loading of [${uboot_emmc}] disabled
    by gcboot/uEnv.txt [disable_uboot_overlay_emmc=1]...;
553                 else
554                     setenv uboot_overlay ${uboot_emmc};
555                     run virtualloadoverlay;
556                 fi;
557             fi;
558             if test -n ${uboot_video}; then
559                 if test -n ${disable_uboot_overlay_video}; then
560                     echo uboot_overlays: uboot loading of [${uboot_video}] disabled
    by gcboot/uEnv.txt [disable_uboot_overlay_video=1]...;
561                 else
562                     if test -n ${disable_uboot_overlay_audio}; then
563                         echo uboot_overlays: uboot loading of [${uboot_video}]
    disabled by gcboot/uEnv.txt [disable_uboot_overlay_audio=1]...;
564                         setenv uboot_overlay ${uboot_video_naudio};
565                         run virtualloadoverlay;
566                     else
567                         setenv uboot_overlay ${uboot_video};
568                         run virtualloadoverlay;
569                     fi;
570                 fi;
571             fi;
572             if test -n ${uboot_wireless}; then
573                 if test -n ${disable_uboot_overlay_wireless}; then
574                     echo uboot_overlays: uboot loading of [${uboot_wireless}]
    disabled by gcboot/uEnv.txt [disable_uboot_overlay_wireless=1]...;
575                 else
576                     setenv uboot_overlay ${uboot_wireless};
577                     run virtualloadoverlay;
578                 fi;
579             fi;
580             if test -n ${uboot_overlay_pru}; then
581                 setenv uboot_overlay ${uboot_overlay_pru};
582                 run virtualloadoverlay;
583             fi;
584             if test -n ${uboot_overlay_pru_add}; then
585                 setenv uboot_overlay ${uboot_overlay_pru_add};
```

```
                run virtualloadoverlay;
            fi;
            if test -n ${dtb_overlay}; then
                setenv uboot_overlay ${dtb_overlay};
                echo uboot_overlays: [dtb_overlay=${uboot_overlay}] ... ;
                run virtualloadoverlay;
            fi;
            if test -n ${uboot_detected_capes}; then
                echo uboot_overlays: [uboot_detected_capes=${uboot_detected_capes_addr0}${uboot_detected_capes_addr1}${uboot_detected_capes_addr2}${uboot_detected_capes_addr3}] ... ;
                setenv uboot_detected_capes uboot_detected_capes=${uboot_detected_capes_addr0}${uboot_detected_capes_addr1}${uboot_detected_capes_addr2}${uboot_detected_capes_addr3};
            fi;
        else
            echo uboot_overlays: add [enable_uboot_overlays=1] to gcboot/uEnv.txt to enable...;
        fi;
        setenv rdfile initrd.img-${uname_r};
        if test -e ${devtype} ${bootpart} ${bootdir}gc${rdfile}; then
            echo loading ${bootdir}gc${rdfile} ...;
            run loadrd;
            if test -n ${netinstall_enable}; then
                run args_netinstall; run message;
                echo debug: [${bootargs}] ... ;
                echo debug: [bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr}] ... ;
                bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr};
            fi;
            if test -n ${uenv_root}; then
                run args_uenv_root;
                echo debug: [${bootargs}] ... ;
                echo debug: [bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr}] ... ;
                bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr};
            fi;
            if test -n ${uuid}; then
                run args_mmc_uuid;
                echo debug: [${bootargs}] ... ;
                echo debug: [bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr}] ... ;
                bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr};
            fi;
            run args_mmc_old;
            echo debug: [${bootargs}] ... ;
            echo debug: [bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr}] ... ;
            bootz ${loadaddr} ${rdaddr}:${rdsize} ${fdtaddr};
        else
            if test -n ${uenv_root}; then
                run args_uenv_root;
                echo debug: [${bootargs}] ... ;
                echo debug: [bootz ${loadaddr} - ${fdtaddr}] ... ;
                bootz ${loadaddr} - ${fdtaddr};
            fi;
            run args_mmc_old;
            echo debug: [${bootargs}] ... ;
            echo debug: [bootz ${loadaddr} - ${fdtaddr}] ... ;
            bootz ${loadaddr} - ${fdtaddr};
        fi;
    fi;

#EEPROM_PROGRAMMING
    eeprom_dump=i2c dev 0;
    i2c md 0x50 0x00.2 20;

eeprom_blank=i2c dev 0;
    i2c mw 0x50 0x00.2 ff;
    i2c mw 0x50 0x01.2 ff;
    i2c mw 0x50 0x02.2 ff;
    i2c mw 0x50 0x03.2 ff;
    i2c mw 0x50 0x04.2 ff;
```

```
    i2c mw 0x50 0x05.2 ff;
    i2c mw 0x50 0x06.2 ff;
    i2c mw 0x50 0x07.2 ff;
    i2c mw 0x50 0x08.2 ff;
    i2c mw 0x50 0x09.2 ff;
    i2c mw 0x50 0x0a.2 ff;
    i2c mw 0x50 0x0b.2 ff;
    i2c mw 0x50 0x0c.2 ff;
    i2c mw 0x50 0x0d.2 ff;
    i2c mw 0x50 0x0e.2 ff;
    i2c mw 0x50 0x0f.2 ff;
    i2c mw 0x50 0x10.2 ff;
    i2c mw 0x50 0x11.2 ff;
    i2c mw 0x50 0x12.2 ff;
    i2c mw 0x50 0x13.2 ff;
    i2c mw 0x50 0x14.2 ff;
    i2c mw 0x50 0x15.2 ff;
    i2c mw 0x50 0x16.2 ff;
    i2c mw 0x50 0x17.2 ff;
    i2c mw 0x50 0x18.2 ff;
    i2c mw 0x50 0x19.2 ff;
    i2c mw 0x50 0x1a.2 ff;
    i2c mw 0x50 0x1b.2 ff;
    i2c mw 0x50 0x1c.2 ff;
    i2c mw 0x50 0x1d.2 ff;
    i2c mw 0x50 0x1e.2 ff;
    i2c mw 0x50 0x1f.2 ff;

eeprom_bbb_header=i2c dev 0;
    i2c mw 0x50 0x00.2 aa;
    i2c mw 0x50 0x01.2 55;
    i2c mw 0x50 0x02.2 33;
    i2c mw 0x50 0x03.2 ee;
    i2c mw 0x50 0x04.2 41;
    i2c mw 0x50 0x05.2 33;
    i2c mw 0x50 0x06.2 33;
    i2c mw 0x50 0x07.2 35;
    i2c mw 0x50 0x08.2 42;
    i2c mw 0x50 0x09.2 4e;
    i2c mw 0x50 0x0a.2 4c;
    i2c mw 0x50 0x0b.2 54;

eeprom_bbbl_footer=
    i2c mw 0x50 0x0c.2 42;
    i2c mw 0x50 0x0d.2 4c;
    i2c mw 0x50 0x0e.2 41;
    i2c mw 0x50 0x0f.2 32;

eeprom_bbbw_footer=
    i2c mw 0x50 0x0c.2 42;
    i2c mw 0x50 0x0d.2 57;
    i2c mw 0x50 0x0e.2 41;
    i2c mw 0x50 0x0f.2 35;

eeprom_bbgg_footer=
    i2c mw 0x50 0x0c.2 47;
    i2c mw 0x50 0x0d.2 47;
    i2c mw 0x50 0x0e.2 31;
    i2c mw 0x50 0x0f.2 41;

eeprom_pocketbeagle=
    i2c mw 0x50 0x00.2 aa;
    i2c mw 0x50 0x01.2 55;
    i2c mw 0x50 0x02.2 33;
    i2c mw 0x50 0x03.2 ee;
    i2c mw 0x50 0x04.2 41;
    i2c mw 0x50 0x05.2 33;
    i2c mw 0x50 0x06.2 33;
    i2c mw 0x50 0x07.2 35;
```

```
719       i2c mw 0x50 0x08.2 50;
720       i2c mw 0x50 0x09.2 42;
721       i2c mw 0x50 0x0a.2 47;
722       i2c mw 0x50 0x0b.2 4c;
723       i2c mw 0x50 0x0c.2 30;
724       i2c mw 0x50 0x0d.2 30;
725       i2c mw 0x50 0x0e.2 41;
726       i2c mw 0x50 0x0f.2 32;
727
728   eeprom_beaglelogic=
729       i2c mw 0x50 0x00.2 aa;
730       i2c mw 0x50 0x01.2 55;
731       i2c mw 0x50 0x02.2 33;
732       i2c mw 0x50 0x03.2 ee;
733       i2c mw 0x50 0x04.2 41;
734       i2c mw 0x50 0x05.2 33;
735       i2c mw 0x50 0x06.2 33;
736       i2c mw 0x50 0x07.2 35;
737       i2c mw 0x50 0x08.2 42;
738       i2c mw 0x50 0x09.2 4c;
739       i2c mw 0x50 0x0a.2 47;
740       i2c mw 0x50 0x0b.2 43;
741       i2c mw 0x50 0x0c.2 30;
742       i2c mw 0x50 0x0d.2 30;
743       i2c mw 0x50 0x0e.2 30;
744       i2c mw 0x50 0x0f.2 41;
745
746
747   #NANDARGS
748   mtdids=" CONFIG_MTDIDS_DEFAULT "
749   mtdparts=" CONFIG_MTDPARTS_DEFAULT "
750   nandargs=setenv bootargs console=${console}
751       ${optargs}
752       root=${nandroot}
753       rootfstype=${nandrootfstype}
754   nandroot=ubi0:rootfs rw ubi.mtd=NAND.file-system,2048
755   nandrootfstype=ubifs rootwait=1
756   nandboot=echo Booting from nand ...;
757       run nandargs;
758       nand read ${fdtaddr} NAND.u-boot-spl-os;
759       nand read ${loadaddr} NAND.kernel;
760       bootz ${loadaddr} - ${fdtaddr}
761
762   #NETARGS
763   static_ip=${ipaddr}:${serverip}:${gatewayip}:${netmask}:${hostname}
764       ::off
765   nfsopts=nolock
766   rootpath=gcexport/rootfs
767   netloadimage=tftp ${loadaddr} ${bootfile}
768   netloadfdt=tftp ${fdtaddr} ${fdtfile}
769   netargs=setenv bootargs console=${console}
770       ${optargs}
771       root=gcdev/nfs
772       nfsroot=${serverip}:${rootpath},${nfsopts} rw
773       ip=dhcp
774   netboot=echo Booting from network ...;
775       setenv autoload no;
776       dhcp;
777       run netloadimage;
778       run netloadfdt;
779       run netargs;
780       bootz ${loadaddr} - ${fdtaddr}
781
782   #DFUARGS
783   #DFU_ALT_INFO_EMMC
784   dfu_alt_info_emmc=
785       rawemmc raw 0 3751936;
786       boot part 1 1;
787       rootfs part 1 2;
```

```
788       MLO fat 1 1;
789       MLO.raw raw 0x100 0x100;
790       u-boot.img.raw raw 0x300 0x1000;
791       u-env.raw raw 0x1300 0x200;
792       spl-os-args.raw raw 0x1500 0x200;
793       spl-os-image.raw raw 0x1700 0x6900;
794       spl-os-args fat 1 1;
795       spl-os-image fat 1 1;
796       u-boot.img fat 1 1;
797       uEnv.txt fat 1 1
798
799   #DFU_ALT_INFO_MMC
800   dfu_alt_info_mmc=
801       boot part 0 1;
802       rootfs part 0 2;
803       MLO fat 0 1;
804       MLO.raw raw 0x100 0x100;
805       u-boot.img.raw raw 0x300 0x1000;
806       u-env.raw raw 0x1300 0x200;
807       spl-os-args.raw raw 0x1500 0x200;
808       spl-os-image.raw raw 0x1700 0x6900;
809       spl-os-args fat 0 1;
810       spl-os-image fat 0 1;
811       u-boot.img fat 0 1;
812       uEnv.txt fat 0 1
813
814   #DFU_ALT_INFO_RAM
815   dfu_alt_info_ram=
816       kernel ram 0x80200000 0x4000000;
817       fdt ram 0x80f80000 0x80000;
818       ramdisk ram 0x81000000 0x4000000
819
820   #DFU_ALT_INFO_NAND
821   dfu_alt_info_nand=
822       SPL part 0 1;
823       SPL.backup1 part 0 2;
824       SPL.backup2 part 0 3;
825       SPL.backup3 part 0 4;
826       u-boot part 0 5;
827       u-boot-spl-os part 0 6;
828       kernel part 0 8;
829       rootfs part 0 9
830
831   #BOOTENV
832   #BOOTENV_SHARED_MMC
833   mmc_boot=if mmc dev ${devnum}; then devtype=mmc; run scan_dev_for_boot_part; fi
834   #BOOTENV_SHARED_USB
835   boot_net_usb_start=usb start
836   usb_boot=usb start; if usb dev ${devnum}; then devtype=usb; run
      scan_dev_for_boot_part; fi
837   #BOOTENV_SHARED_EFI
838   boot_efi_binary=
839       if fdt addr ${fdt_addr_r}; then
840           bootefi bootmgr ${fdt_addr_r};
841       else
842           bootefi bootmgr ${fdtcontroladdr};
843       fi;
844       load ${devtype} ${devnum}:${distro_bootpart}
845           ${kernel_addr_r} efigcboot/"bootarm.efi";
846       if fdt addr ${fdt_addr_r}; then
847           bootefi ${kernel_addr_r} ${fdt_addr_r};
848       else
849           bootefi ${kernel_addr_r} ${fdtcontroladdr};
850       fi
851
852   load_efi_dtb=
853       load ${devtype} ${devnum}:${distro_bootpart}
854           ${fdt_addr_r} ${prefix}${efi_fdtfile}
855
```

```
efi_dtb_prefixes=gc /dtb/ /dtb/current/
scan_dev_for_efi=
    setenv efi_fdtfile ${fdtfile};
   if test -z${fdtfile}" -a -n${soc}"; then
        setenv efi_fdtfile ${soc}-${board}${boardver}.dtb;
    fi;

    for prefix in ${efi_dtb_prefixes}; do
        if test -e ${devtype}
                ${devnum}:${distro_bootpart}
                ${prefix}${efi_fdtfile}; then
            run load_efi_dtb;
        fi;
    done;
    if test -e ${devtype} ${devnum}:${distro_bootpart}
                efigcboot/"bootarm.efi"; then
            echo Found EFI removable media binary
                efigcboot/"bootarm.efi";
            run boot_efi_binary;
            echo EFI LOAD FAILED: continuing...;
    fi;
    setenv efi_fdtfile

boot_prefixes=gc /boot/
boot_scripts=boot.scr.uimg boot.scr
boot_script_dhcp=boot.scr.uimg
#BOOTENV_BOOT_TARGETS
boot_targets=mmc0 legacy_mmc0 mmc1 legacy_mmc1 pxe dhcp

boot_syslinux_conf=extlinuxgcextlinux.conf
boot_extlinux=
    sysboot ${devtype} ${devnum}:${distro_bootpart} any
        ${scriptaddr} ${prefix}${boot_syslinux_conf}

scan_dev_for_extlinux=
    if test -e ${devtype}
            ${devnum}:${distro_bootpart}
            ${prefix}${boot_syslinux_conf}; then
        echo Found ${prefix}${boot_syslinux_conf};
        run boot_extlinux;
        echo SCRIPT FAILED: continuing...;
    fi

boot_a_script=
    load ${devtype} ${devnum}:${distro_bootpart}
        ${scriptaddr} ${prefix}${script};
    source ${scriptaddr}

scan_dev_for_scripts=
    for script in ${boot_scripts}; do
        if test -e ${devtype}
                ${devnum}:${distro_bootpart}
                ${prefix}${script}; then
            echo Found U-Boot script
                ${prefix}${script};
            run boot_a_script;
            echo SCRIPT FAILED: continuing...;
        fi;
    done

scan_dev_for_boot=
    echo Scanning ${devtype}
            ${devnum}:${distro_bootpart}...;
    for prefix in ${boot_prefixes}; do
        run scan_dev_for_extlinux;
        run scan_dev_for_scripts;
    done;
    #SCAN_DEV_FOR_EFI
    run scan_dev_for_efi;
```

```
925
926
927  scan_dev_for_boot_part=
928      part list ${devtype} ${devnum} -bootable devplist;
929      env exists devplist || setenv devplist 1;
930      for distro_bootpart in ${devplist}; do
931          if fstype ${devtype}
932                  ${devnum}:${distro_bootpart}
933                  bootfstype; then
934              run scan_dev_for_boot;
935          fi;
936      done;
937      setenv devplist
938
939  #BOOT_TARGET_DEVICES(BOOTENV_DEV)
940  bootcmd_mmc0=devnum=0; run mmc_boot
941  bootcmd_legacy_mmc0=gpio clear 56; gpio clear 55; gpio clear 54; gpio set 53;
     setenv devtype mmc; setenv mmcdev 0; setenv bootpart 0:1 ; run boot
942  bootcmd_mmc1=devnum=1; run mmc_boot
943  bootcmd_legacy_mmc1=gpio clear 56; gpio clear 55; gpio clear 54; gpio set 53;
     setenv devtype mmc; setenv mmcdev 1; setenv bootpart 1:1 ; run boot
944  bootcmd_pxe=run boot_net_usb_start; dhcp; if pxe get; then pxe boot; fi
945  bootcmd_dhcp=run boot_net_usb_start; if dhcp ${scriptaddr} ${boot_script_dhcp};
     then source ${scriptaddr}; fi;setenv efi_fdtfile ${fdtfile}; if test -z${fdtfile}" -
     a -n${soc}"; then setenv efi_fdtfile ${soc}-${board}${boardver}.dtb; fi; setenv
     efi_old_vci ${bootp_vci};setenv efi_old_arch ${bootp_arch};setenv bootp_vci
     PXEClient:Arch:00010:UNDI:003000;setenv bootp_arch 0xa;if dhcp ${kernel_addr_r};
     then tftpboot ${fdt_addr_r} dtbgc${efi_fdtfile};if fdt addr ${fdt_addr_r}; then
     bootefi ${kernel_addr_r} ${fdt_addr_r}; else bootefi ${kernel_addr_r} $
     {fdtcontroladdr};fi;fi;setenv bootp_vci ${efi_old_vci};setenv bootp_arch $
     {efi_old_arch};setenv efi_fdtfile;setenv efi_old_arch;setenv efi_old_vci;
946
947
948  distro_bootcmd=
949      for target in ${boot_targets}; do
950          run bootcmd_${target};
951      done
```