

망고100 보드로 놀아보자-7

Include/mango100.h 설명

U-boot \$BOARDNAME.h 파일 분석

- #vi include/configs/ 에 위치
- #make mango100_config 명령 수행 mango100.h 이름

```
echo "/* Automatically generated - do not edit */" >>config.h  
echo "#include <configs/$1.h>" >>config.h
```

- Booting Device, 메모리, 프롬프트 이름 등, u-boot 소스에서 가장 중요한 파일임
- Define한 Value값은 전체 시스템에 영향을 미침

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_S5PC100    1        /* in a SAMSUNG S5PC100 SoC */
#define CONFIG_S5PC1XX    1        /* in a SAMSUNG S5PC1XX Family */
#define CONFIG_MANGO100 1

// #define CONFIG_S5PC100_EVT1
#define CONFIG_S5PC100_EVT2

#define BOOT_ONENAND      0x1
#define BOOT_NAND        0x2
#define BOOT_MMCSDB      0x3
// S5PC100 CPU 는 Memory MAP에서 DRAM 영역 시작 정의
#define MEMORY_BASE_ADDRESS 0x20000000

#if defined(CONFIG_S5PC100_EVT2)
#define CONFIG_MEMORY_UPPER_CODE // u-boot code 위에 STACK, Malloc 메모리 사용
#define CONFIG_MMC // MMC Booting 지원
#define CONFIG_MOVINAND //
#endif

/* input clock of PLL */ // 망고100 은 Crystal 12Mhz가 달려있습니다.
#define CONFIG_SYS_CLK_FREQ 12000000 /* the Mango100 has 12MHz input clock */
```

U-boot \$BOARDNAME.h 파일 분석

```
#undef CONFIG_ENABLE_MMU //MMU 미지원 Physical ADDR 사용
```

```
#ifdef CONFIG_ENABLE_MMU
#define virt_to_phys(x) virt_to_phy_smdkc100(x)
#else
#define virt_to_phys(x) (x)
#endif
```

```
#define CONFIG_MEMORY_UPPER_CODE
//#undef CONFIG_MEMORY_UPPER_CODE
```

```
#undef CONFIG_USE_IRQ /* we don't need IRQ/FIQ stuff */
```

```
#define CONFIG_INCLUDE_TEST
```

```
#define CONFIG_ZIMAGE_BOOT //커널 zImage 이미지 지원:압축
```

```
#define CONFIG_IMAGE_BOOT
```

```
#define BOARD_LATE_INIT
```

```
#define CONFIG_SETUP_MEMORY_TAGS
```

```
#define CONFIG_CMDLINE_TAG
```

```
#define CONFIG_INITRD_TAG
```

```
if (*(ulong*)(addr + 9*4) == LINUX_ZIMAGE_MAGIC) {
    printf("Boot with zImage\n");
    addr = virt_to_phys(addr);
    hdr = (image_header_t *)addr;
    hdr->ih_os = IH_OS_LINUX;
    hdr->ih_ep = ntohl(addr);

    memmove (&images.legacy_hdr_os_copy, hdr, sizeof(image_header_t));
}
```

U-boot \$BOARDNAME.h 파일 분석

```
/*
 * Architecture magic and machine type
 */
#define MACH_TYPE          1826 //kernel machine type과 일치해야 함
#define UBOOT_MAGIC       (0x43090000 | MACH_TYPE)

/* Power Management is enabled */
#define CONFIG_PM          smdkc100 MACH_SMDKC100 SMDKC100
                          (kernel소스)/arch/arm/tools/mach-types에 위치함

#define CONFIG_DISPLAY_CPUINFO //CPU 정보 제공
#define CONFIG_DISPLAY_BOARDINFO //보드 정보 제공

#undef CONFIG_SKIP_RELOCATE_UBOOT
#undef CONFIG_USE_NOR_BOOT

/*
 * Size of malloc() pool
 */
#define CFG_MALLOC_LEN     (CFG_ENV_SIZE + 1024*1024) //0x40000+1MB
#define CFG_GBL_DATA_SIZE  128 /* size in bytes reserved for initial data */

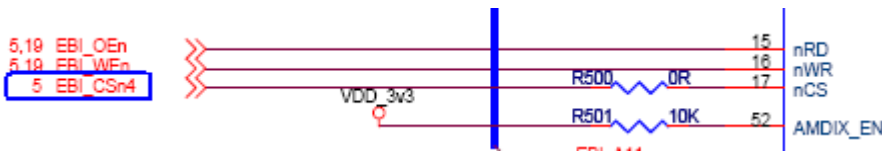
#define CFG_STACK_SIZE     512*1024 /* STACK Size 정의 */
```

smdkc100 MACH_SMDKC100 SMDKC100
(kernel소스)/arch/arm/tools/mach-types에 위치함

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_DRIVER_SMC911X 1 /* we have a SMSC9220 on-board */

#ifdef CONFIG_DRIVER_SMC911X
#define CONFIG_DRIVER_SMC911X_16_BIT
#undef CONFIG_DRIVER_CS8900
#define CONFIG_DRIVER_SMC911X_BASE 0xA0000300
#else
#define CONFIG_DRIVER_CS8900 1 /* we have a CS8900 on-board */
#define CS8900_BASE 0x18800300
#define CS8900_BUS16 1 /* the Linux driver does accesses as shorts */
#endif
```



이더넷이 CS Bank4에 연결되어 있으므로 0xA0000000 Address 값을 가진다. [7:0]bit 값은 0 값을 가져야 한다.

0x9800_0000	0xA000_0000	128MB	SMC Bank 3
0xA000_0000	0xA800_0000	128MB	SMC Bank 4
0xA800_0000	0xB000_0000	128MB	SMC Bank 5

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_SERIAL2      1      /*mango100 uart 1번이 DEBUG Port 임*/
#define CONFIG_USB_OHCI
#undef CONFIG_USB_STORAGE
#define CONFIG_S3C_USBD

// #define USBD_DOWN_ADDR      0xc0000000
#define USBD_DOWN_ADDR      0x20008000

/*****
* RTC
*****/

/* allow to overwrite serial and ethaddr */
#define CONFIG_ENV_OVERWRITE

#define CONFIG_BAUDRATE      115200 //Baurate 115200 설정
```

Usb otg 활성화하여 usb download 지원

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_BOOTDELAY      3/*u-boot가 실행 후 커널 이미지를 load하기 전에 3초 기다린다.*/
```

```
// MMC2 boot
```

```
#define CONFIG_BOOTARGS      "root=/dev/mmcblk0p3 rw rootfstype=ext3 console=ttySAC1,115200 rootdelay=1"
```

```
// gnome boot
```

```
//#define CONFIG_BOOTARGS      "ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rw console=ttySAC1,115200"
```

```
// android boot
```

```
//#define CONFIG_BOOTARGS      "ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rw init=/init console=ttySAC1,115200"
```

```
#define CONFIG_ETHADDR       00:40:5c:26:0a:5b
```

```
#define CONFIG_NETMASK       255.255.255.0
```

```
#define CONFIG_IPADDR        192.168.1.121 /* mango board IP 주소*/
```

```
#define CONFIG_SERVERIP      192.168.1.2 /* Host PC IP 주소*/
```

```
#define CONFIG_GATEWAYIP     192.168.1.1 /*Gateway IP 주소*/
```

Bootargs 에서

Root : 디바이스 이름

rootfstype: 사용할 root filesystem 타입(ubifs,yaffs2,jffs2,ext3, 등)

Rw :Read ,Write 가능 하도록

Console: uart 설정

Init: 커널에서 처음 실행하는 process

Ubi.mtd: MTD Partition Block Number

U-boot \$BOARDNAME.h 파일 분석

```
#define CFG_LONGHELP          /* undef to save memory */
#define CFG_PROMPT           "MANGO100 # " /* Monitor Command Prompt */
#define CFG_CBSIZE           256          /* Console I/O Buffer Size */
#define CFG_PBSIZE           384          /* Print Buffer Size */
#define CFG_MAXARGS          16           /* max number of command args */
#define CFG_BARGSIZE         CFG_CBSIZE   /* Boot Argument Buffer Size */

#define CFG_MEMTEST_START    MEMORY_BASE_ADDRESS /* memtest works on */
#define CFG_MEMTEST_END     MEMORY_BASE_ADDRESS + 0xfe00000 /* 256MB in DRAM */

#undef CFG_CLKS_IN_HZ        /* everything, incl board info, in Hz */

#define CFG_LOAD_ADDR        MEMORY_BASE_ADDRESS /* default load address */

/* the PWM Timer 4 uses a counter of 15625 for 10 ms, so we need */
/* it to wrap 100 times (total 1562500) to get 1 sec. */
#define CFG_HZ                1562500      // at PCLK 50MHz

/* valid baudrates */
#define CFG_BAUDRATE_TABLE   { 9600, 19200, 38400, 57600, 115200 }

/*-----
 * Stack sizes
 *
 * The stack sizes are set up in start.S using the settings below
 */
```

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_IDENT_STRING " for MANGO100"

/* total memory required by uboot */
#define CFG_UBOOT_SIZE (2*1024*1024)

/* base address for uboot */
#ifdef CONFIG_ENABLE_MMU
#define CFG_UBOOT_BASE 0xcfe00000
#else
#define CFG_UBOOT_BASE 0x2fe00000 (2*1024*1024)
#endif

#define CFG_PHY_UBOOT_BASE MEMORY_BASE_ADDRESS + 0xfe0000

#define CFG_ENV_OFFSET 0x0007C000
```



U-boot \$BOARDNAME.h 파일 분석

```
/* NAND configuration */
#define CFG_MAX_NAND_DEVICE 1 /*mango100 Nand device 1개*/
#define CFG_NAND_BASE (0xE7200000) /*Nand Control Register*/
#define NAND_MAX_CHIPS 1 // Chip 갯수

#define NAND_DISABLE_CE() (NFCONT_REG |= (1 << 1))
#define NAND_ENABLE_CE() (NFCONT_REG &= ~(1 << 1))
#define NF_TRANSRnB() do { while(!(NFSTAT_REG & (1 << 0))); } while(0)

#define CFG_NAND_SKIP_BAD_DOT_I 1 /* ".i" read skips bad blocks */
#define CFG_NAND_WP 1
#define CFG_NAND_YAFFS_WRITE 1 /* support yaffs write */

#define CFG_NAND_HWECC //ECC 체크를 HW 처리
// #define CONFIG_NAND_BL1_8BIT_ECC
#undef CFG_NAND_FLASH_BBT

#define CONFIG_BOOTCOMMAND "movi read kernel 20008000;bootm 20008000"
// #define CONFIG_BOOTCOMMAND "nand read 20008000 80000 300000;bootm 20008000"

#define CONFIG_NAND
#define CONFIG_BOOT_NAND
#define CONFIG_BOOT_MOVINAND
#define CFG_ENV_IS_IN_AUTO
```