in drivers/serial/serial.c

```c
/**
* serial_initialize() - Register all compiled-in serial port drivers
*
* This function registers all serial port drivers that are compiled
* into the U-Boot binary with the serial core, thus making them
* available to U-Boot to use. Lastly, this function assigns a default
* serial port to the serial core. That serial port is then used as a
* default output.
*/
void serial_initialize(void)
{
mpc8xx_serial_initialize();
ns16550_serial_initialize();
pxa_serial_initialize();
s3c24xx_serial_initialize();
s5p_serial_initialize();
mpc512x_serial_initialize();
bfin_serial_initialize();
bfin_jtag_initialize();
uartlite_serial_initialize();
zynq_serial_initialize();
au1x00_serial_initialize();
asc_serial_initialize();
jz_serial_initialize();
mpc5xx_serial_initialize();
mpc8260_scc_serial_initialize();
mpc8260_smc_serial_initialize();
mpc85xx_serial_initialize();
iop480_serial_initialize();
leon2_serial_initialize();
leon3_serial_initialize();
marvell_serial_initialize();
amirix_serial_initialize();
bmw_serial_initialize();
cogent_serial_initialize();
cpci750_serial_initialize();
evb64260_serial_initialize();
ml2_serial_initialize();
sconsole_serial_initialize();
p3mx_serial_initialize();
altera_jtag_serial_initialize();
altera_serial_initialize();
atmel_serial_initialize();
lpc32xx_serial_initialize();
mcf_serial_initialize();
oc_serial_initialize();
sandbox_serial_initialize();
clps7111_serial_initialize();
imx_serial_initialize();
ixp_serial_initialize();
ks8695_serial_initialize();
lh7a40x_serial_initialize();
max3100_serial_initialize();
mxc_serial_initialize();
pl01x_serial_initialize();
sa1100_serial_initialize();
sh_serial_initialize();
arm_dcc_initialize();
mxs_auart_initialize();

serial_assign(default_serial_console()->name);
}
```

serial_initialize()看上去好像什么类型的串口都initialize，但实际上真正initialize的只是其中一种，其他都是空函数。
比如对pegmatite而言，有效的是 `ns16550_serial_initialize()` ,因为

```
void ns16550_serial_initialize(void)
{
#if defined(CONFIG_SYS_NS16550_COM1)
serial_register(&eserial1_device);
#endif
#if defined(CONFIG_SYS_NS16550_COM2)
serial_register(&eserial2_device);
#endif
#if defined(CONFIG_SYS_NS16550_COM3)
serial_register(&eserial3_device);
#endif
#if defined(CONFIG_SYS_NS16550_COM4)
serial_register(&eserial4_device);
#endif
#if defined(CONFIG_SYS_NS16550_COM5)
serial_register(&eserial5_device);
#endif
#if defined(CONFIG_SYS_NS16550_COM6)
serial_register(&eserial6_device);
#endif
}
```

而在pegmatite.h中

```
#define CONFIG_SYS_NS16550_COM1 0xd4030000
```

==>

```
void ns16550_serial_initialize(void)
{
#if defined(CONFIG_SYS_NS16550_COM1)
serial_register(&eserial1_device);
#endif
}
```

对pxa_serial_initialize()而言

```
void pxa_serial_initialize(void)
{
#if defined(CONFIG_FFUART)
serial_register(&serial_ffuart_device);
#endif
#if defined(CONFIG_BTUART)
serial_register(&serial_btuart_device);
#endif
#if defined(CONFIG_STUART)
serial_register(&serial_stuart_device);
#endif
}
```

这里 `CONFIG_FFUART` , `CONFIG_BTUART` and `CONFIG_STUART` pegmatite.h中没有一个定义，自然是空函数。

在arch/arm/lib/board.c中

```
void board_init_r(gd_t *id, ulong dest_addr)
{
ulong malloc_start;
#if !defined(CONFIG_SYS_NO_FLASH)
ulong flash_size;
#endif

gd->flags |= GD_FLG_RELOC;  /* tell others: relocation done */
bootstage_mark_name(BOOTSTAGE_ID_START_UBOOT_R, "board_init_r");

monitor_flash_len = _end_ofs;

/* Enable caches */
enable_caches();

debug("monitor flash len: %08lX\n", monitor_flash_len); ①
board_init();   /* Setup chipselects */
/*
* TODO: printing of the clock inforamtion of the board is now
* implemented as part of bdinfo command. Currently only support for
* davinci SOC's is added. Remove this check once all the board
* implement this.
*/
#ifdef CONFIG_CLOCKS
set_cpu_clk_info(); /* Setup clock information */
#endif
serial_initialize();    ②

debug("Now running in RAM - U-Boot at: %08lx\n", dest_addr);    ③
......
}
```

①
这行的debug()能输出吗？由于串口还未初始化，好像不应该吧？

②
初始化串口

③
串口可以工作了

而board_init_r()则是在arch/arm/lib/crt0.S中被调用的。

```
ENTRY(_main)

/*
* Set up initial C runtime environment and call board_init_f(0).
*/

#if defined(CONFIG_SPL_BUILD) && defined(CONFIG_SPL_STACK)
ldr sp, =(CONFIG_SPL_STACK)
#else
ldr sp, =(CONFIG_SYS_INIT_SP_ADDR)
#endif
bic sp, sp, #7  /* 8-byte alignment for ABI compliance */
sub sp, sp, #GD_SIZE    /* allocate one GD above SP */
bic sp, sp, #7  /* 8-byte alignment for ABI compliance */
mov r9, sp  /* GD is above SP */
mov r0, #0
bl  board_init_f

#if ! defined(CONFIG_SPL_BUILD)

/*
* Set up intermediate environment (new sp and gd) and call
* relocate_code(addr_moni). Trick here is that we'll return
* 'here' but relocated.
*/

ldr sp, [r9, #GD_START_ADDR_SP] /* sp = gd->start_addr_sp */
bic sp, sp, #7  /* 8-byte alignment for ABI compliance */
ldr r9, [r9, #GD_BD]    /* r9 = gd->bd */
sub r9, r9, #GD_SIZE    /* new GD is below bd */

adr lr, here
ldr r0, [r9, #GD_RELOC_OFF] /* r0 = gd->reloc_off */
add lr, lr, r0
ldr r0, [r9, #GD_RELOCADDR] /* r0 = gd->relocaddr */
b   relocate_code   ①
here:

/* Set up final (full) environment */

bl  c_runtime_cpu_setup /* we still call old routine here */

ldr r0, =__bss_start    /* this is auto-relocated! */
ldr r1, =__bss_end  /* this is auto-relocated! */

mov r2, #0x00000000 /* prepare zero to clear BSS */

clbss_l:cmp r0, r1  /* while not at end of BSS */
strlo   r2, [r0]    /* clear 32-bit BSS word */
addlo   r0, r0, #4  /* move to next */
blo clbss_l

bl coloured_LED_init
bl red_led_on

/* call board_init_r(gd_t *id, ulong dest_addr) */
mov     r0, r9                  /* gd_t */
ldr r1, [r9, #GD_RELOCADDR] /* dest_addr */
/* call board_init_r */
ldr pc, =board_init_r   /* this is auto-relocated! */   ②

/* we should not return here. */

#endif

ENDPROC(_main)
```

①
relocate_code()完成u-boot自身的relocate，把自己搬到内存高端去执行

②
调用板级初始化。

由此看出，debug log可以从串口输出还是蛮早的。