

Vaja 5
RTC (real time clock) in zapis v
pomnilnik FLASH

Na prikazovalniku LCD izpisujte trenutni čas (ure:minute:sekunde) in čas zadnjega pritiska tipke T0. V primeru izklopa napajanja sistema naj se informacije o pritisku T0 ohrani (zapis v FLASH). Tipke T1, T2 in T3 uporabite za nastavljanje trenutnega časa.

Uporaba RTC (real time clock) – glej rtc.c, rtc.h :

rtc_init(); / mogoči RTC, onemogoči prekinitve, ki bi jih zahtevala ta periferna enota, in preveri doslednost datuma */*

Branje časa RTC:

*void get_time_and_date(int *hour, int *min, int *sec, int *day_of_week, int *day_of_year, int *day, int *month, int *year); /* argumenti so kazalci => naslovi spr. tipa int */*

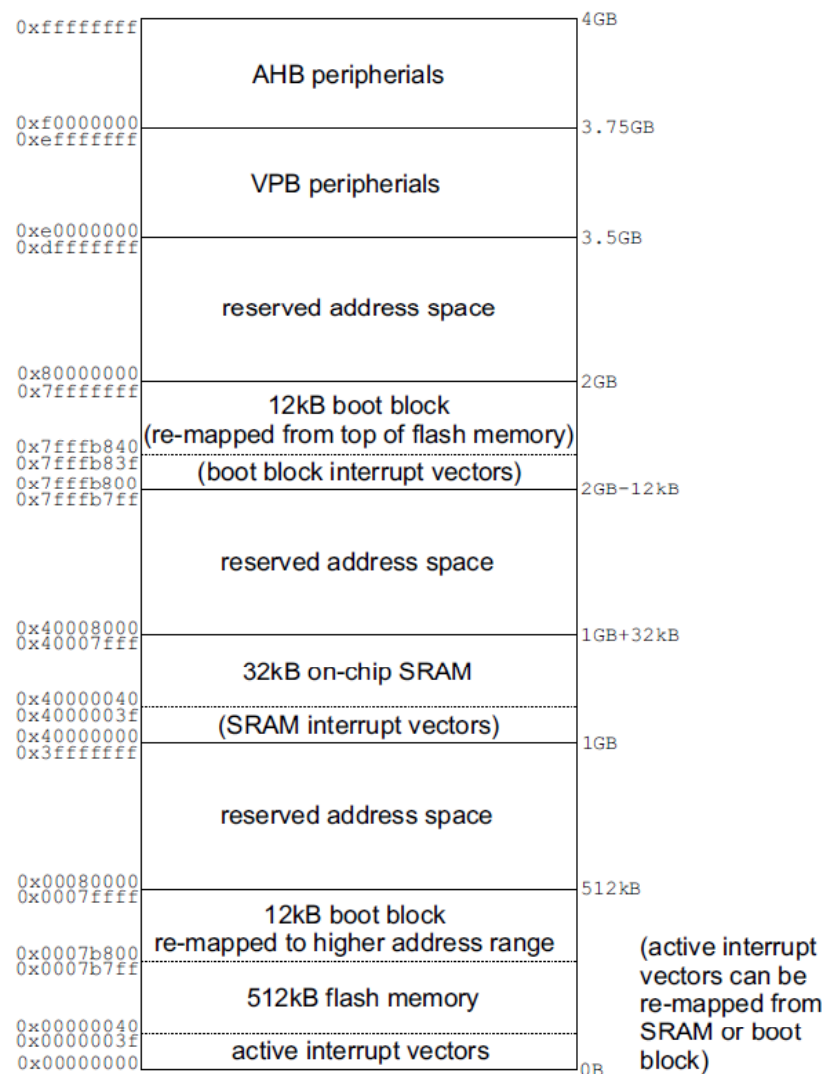
Klic funkcije:

*int hour, min, sec, day_of_week, day_of_year, day, month, year;
get_time_and_date(&hour, &min, &sec, &day_of_week, &day_of_year, &day, &month, &year);*

Nastavitev časa in datuma:

*void set_time(int hour, int min, int sec);
void set_date(int day, int month, int year);*

Zapis v FLASH – glej iap.c, iap.h (In Application Programming):



Slika B.1: Razdelitev naslovnega prostora mikrokontrolerka LPC2138

sektor	velikost	naslovni prostor
0	4kB	0x00000000 - 0x00000fff
1	4kB	0x00001000 - 0x00001fff
2	4kB	0x00002000 - 0x00002fff
3	4kB	0x00003000 - 0x00003fff
4	4kB	0x00004000 - 0x00004fff
5	4kB	0x00005000 - 0x00005fff
6	4kB	0x00006000 - 0x00006fff
7	4kB	0x00007000 - 0x00007fff
8	32kB	0x00008000 - 0x0000ffff
9	32kB	0x00010000 - 0x00017fff
10	32kB	0x00018000 - 0x0001ffff
11	32kB	0x00020000 - 0x00027fff
12	32kB	0x00028000 - 0x0002ffff
13	32kB	0x00030000 - 0x00037fff
14	32kB	0x00038000 - 0x0003ffff
15	32kB	0x00040000 - 0x00047fff
16	32kB	0x00048000 - 0x0004ffff
17	32kB	0x00050000 - 0x00057fff
18	32kB	0x00058000 - 0x0005ffff
19	32kB	0x00060000 - 0x00067fff
20	32kB	0x00068000 - 0x0006ffff
21	32kB	0x00070000 - 0x00077fff
22	4kB	0x00078000 - 0x00078fff
23	4kB	0x00079000 - 0x00079fff
24	4kB	0x0007a000 - 0x0007afff
25	4kB	0x0007b000 - 0x0007bfff
26	4kB	0x0007c000 - 0x0007cfff

Tabela B.13: Razdelitev pomnilnika flash po sektorjih

Zapis v FLASH – glej iap.c, iap.h (In Application Programming):

```
int *end_of_code(); /* poišče 1. prosto mesto v FLASH-u */
```

```
int find_sector( int *address, int **sector, int *length);
```

```
/* poišče v katerem sektorju pomnilnika se nahaja address:
```

```
    address ... naslov celice, za katero želimo ugotoviti v katerem sektorju se nahaja,
```

```
    sector ... kazalec na naslov prve celice v izbranem sektorju
```

```
    length ... kazalec na dolžino izbranega sektorja
```

```
*/
```

Ne smemo pisat v sektor, kjer je še koda, ampak v 1. naslednjega => ta je cel prost

V FLASH zapisujemo 0, z brisanjem se vpišejo 1 => če vpišemo 0 in hočemo nato vpisati 1, moramo cel sektor najprej pobrisati in na novo vpisati le 0

Brisanje sektorjev:

```
int erase_sectors(int first, int last);    /* brišemo lahko le 1 ali več sektorjev hkrati =>
                                           indeksi od first do last */
```

Pisanje v FLASH:

```
int write_flash(int *to, int *from, int length);
/*      to ... na 256B poravnan naslov v pomnilniku flash (zadnji bajt naslova je 0x00),
        kamor bomo podatke zapisali,
    from ... na 32 bitov poravnan naslov, kjer se nahajajo podatki za zapis in
    length ... količina podatkov, ki jih želimo zapisati (možne vrednosti so: 256, 512,
        1024 in 4096).
*/
```

