



۲۹

ARM

بنا کردن

second
ثانیه

تایمر بلادرنگ (یا حقیقی) Real Time Timer

RTT

یک ثانیه ۳۲ بیت که با تقسیم از کلاک هسته (۳۲۷۶۸ KHz) کلام کند

بیشتر ها جمع RTT :

mode reg. RTTC-RTMR ۳۲ بیت

۱۵-۰۰

RTPRES می شود

الف - ۱۶ بیت اول

فرکانس RTT برابر فرکانس کلاک هسته ۳۲۷۶۸ Hz تقسیم بر مقدار RTPRES

$$f_{RTT} = \frac{32768 \text{ Hz}}{RTPRES}$$

تلاشه اگر مقدار RTPRES برابر می شود

$$f_{RTT} = \frac{32768}{2^{16}}$$

۱۵ Hz یعنی هر یک ثانیه مقدار RTT یک افزایش باید

$$f_{RTT} = \frac{32768 \text{ Hz}}{RTPRES} \Rightarrow RTPRES = \frac{32768 \text{ Hz}}{1 \text{ Hz}} = 32768 = 0x8000$$

یعنی اگر مقدار RTPRES برابر ۰x8000 تنظیم شود هر یک ثانیه یکبار RTT افزایش می یابد

تذکره در RTT وقفه داریم تحت عنوان وقفه افزایشی RTTINC بطوریکه اگر RTT

یک واحد افزایشی باید این وقفه فراخوانی می شود (بعد از توضیح داده خواهد شد) با این کار می توان یک ساعت حقیقی

طراحی کرد، هر چند بدون وقفه نمی توان این کار را انجام داد

(در حالت $f_{RTT} = 1 \text{ Hz}$)

19

$$RTPRES = \frac{2074112}{2} = 16384 = 0x4000$$

$$f_{RTT} = 1\text{KHz}$$

(25)

در هر ثانیه ۲۰۰۰ افزایش برابر RTT داریم. (در ثانیه ۲ بار یک RTT موجود)

(فعال ساز وقفه هشدار) (RTTC-RTMR.16)

Alarm Interrupt Enable

ALMIEN بیت ۱۶: فعال

وقفه هشدار: وقفه هشدار داریم. بطوریکه مقدار تایمر بلاذنب «RTTC-RTVR» value

هر لحظه با مقدار رجیستر هشدار تایمر بلاذنب «RTTC-RTAR» Alarm مقایسه شود در صورت

برابر در صورت فعال بودن این وقفه (بیت بود بیت «ALMIEN»)

ALMS از رجیستر RTTC-RTSR یک شده و وقفه فراخوانی می شود.

Increment Interrupt Enable

(فعال ساز وقفه افزایش) RTTINCEN بیت ۱۷: فعال

در صورت یک شدن این بیت وقفه افزایش فعال می شود و در صورتیکه RTT یک واحد افزایش یابد

RTTINC در رجیستر RTTC-RTSR یک شده و وقفه افزایش فراخوانی می شود.

Reset فعال، $f_{RTT} = 1\text{KHz}$

RTTRST بیت ۱۸: فعال

اگر یک شود باید Reset تایمر RTT می شود و مجدداً شروع می شود.

تذکره: برای ~~تنظیم~~ RTT به عبارت ابتدای Reset می شود.

2- رجیستر 32 بیت RTTC-RTAR Alarm

رجیستر هشدار می باشد. استفاده از وقفه هشدار (مربوط به این رجیستر مقدار دهانه دارد).
AT91C-BASE-RTTC → RTTC-RTAR = value

3- رجیستر مقدار تایمر RTT ۳۲ بیت RTTC-RTVR

محتوای این رجیستر مقدار تایمر RTT می باشد که می توان مقدار این رجیستر را در هر لحظه توسط این رجیستر خواند.
unsigned int
K = AT91C-BASE-RTTC → RTTC-RTVR;



RTTC - RTSR ^{status}

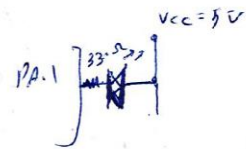
۴۱

۴- رجیستر وضعیت - تایمر RTT

دوبت اول (بیت منبر) ALMS و (بیت یلم) RTTINC
بیت های هر رجیستر دو وقته ذکر شده باشند.

برنامه کار می کند که با یک تایمر RTT (پلاگین) بطوریکه هر ثانیه یکبار PA.1
(یک لایحه ۸MHz) یک ثانیه روشن و یک ثانیه خاموش باشد.

```
#include <Atmel / io at91sam7s64.h>
#define off AT91C_BASE_PIOA -> PIO_SODR = (1 << 1);
#define ON AT91C_BASE_PIOA -> PIO_ODR = (1 << 1);
unsigned int x;
void main(void)
{
    start-up
    speed-up
    AT91C_BASE_PIOA -> PIO_PCR = (1 << AT91C_ID_PIOA);
    AT91C_BASE_RTTC -> RTTC_RTMR = 0x48000;
    AT91C_BASE_PIOA -> PIO_PER = (1 << 1);
    AT91C_BASE_PIOA -> PIO_OER = (1 << 1);
    off;
    while(1)
    {
        x = AT91C_BASE_RTTC -> RTTC_RTVR;
        if(x % 2 == 0)
            off;
        else
            on;
    }
}
```



فعال ساز
لایحه رجیستر

f = 1Hz
divisor = 48000

در هر ثانیه یکبار روشن و یکبار خاموش می شود.
در هر ثانیه یکبار روشن و یکبار خاموش می شود.

تایمر: هر ثانیه یکبار از LED روشن می شود و یکبار خاموش می شود.
در هر ثانیه یکبار از LED روشن می شود و یکبار خاموش می شود.

تمرین : برنامه ساعت را بدون وقفه یکبار RTT بنویسید و نتیجه در LCD نمایش داده شود.

۴۴

```
#include < ATmel / ioat91sam7s64.h >
#include "Lcd.h"
#include <stdio.h>
unsigned char smgh, saat[10];
// unsigned int s;
void main()
{
    start_up
    speed_up
    PCCR
    AT91C_BASE_PMC -> PMC_PRR = (1 << AT91C_ID_PIOA);
    AT91C_BASE_RTC -> RTTC_RTMR = 0x48000;
    Lcd_PinInit(0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
    Lcd_init();
    AT91C_BASE_RTC -> RTTC_RTMR = 0x48000;
    Lcd_clear();
    while(1)
    {
        s = AT91C_BASE_RTC -> RTTC_RTMR;
        if (s == 0) { s = 0; m++; AT91C_BASE_RTC -> RTTC_RTMR = 0x48000; }
        if (m == 60) { m = 0; h++; AT91C_BASE_RTC -> RTTC_RTMR = 0x48000; }
        if (h == 24) { h = 0; }
        if (x == 23) { x = 0; }
        if (m == 59) { m = 0; }
        if (s == 59) { s = 0; }
        h = 0;
        AT91C_BASE_RTC -> RTTC_RTMR = 0x48000; }
        Lcd_gotoxy(0, 0);
        printf(saat, "%02d : %02d : %02d", h, m, s);
        Lcd_puts(saat);
    }
```

نکته: در این برنامه تنها در هر RTT یکبار مقدار

(۷) $glcd_display$: اگر مقدار از ۵۱۲ کمتر باشد عدد را ۵۱۲ قرار می‌دهد و اگر از ۵۱۲ بیشتر باشد عدد را ۵۱۲ قرار می‌دهد. (تقریباً به ۵۱۲ محدود می‌کند)

مقدار ۵۱۲ را به ۵۱۲ محدود می‌کند و اگر از ۵۱۲ بیشتر باشد عدد را ۵۱۲ قرار می‌دهد و اگر از ۵۱۲ کمتر باشد عدد را ۵۱۲ قرار می‌دهد.

۴۳

(۸) $glcd_shift_down$: شیفต์ تصاویر به سمت پایین.

۱- سمت چپ

۲- سمت راست

۳- کل صفحه

۹ ≤ عدد ≤ ۱ : سرعت حرکت تصاویر به پایین عدد ۵ سرعت متوسط

(۹) $glcd_shift_up$: شیفต์ به بالا

(۱۰) $glcd_setup_pix$: روشن کردن یک پیکسل مشخص از تصویر در حال نمایش.

۱۲۷
۶۳

(۱۱) $glcd_clip_pix$: خاموش کردن یک پیکسل از تصویر در حال نمایش.

۵ پیکسل : یک تصویر را در Led نمایش دهیم و بعد از یک ثانیه مناسب آن را به سمت چپ یا راست حرکت دهیم.

ARM