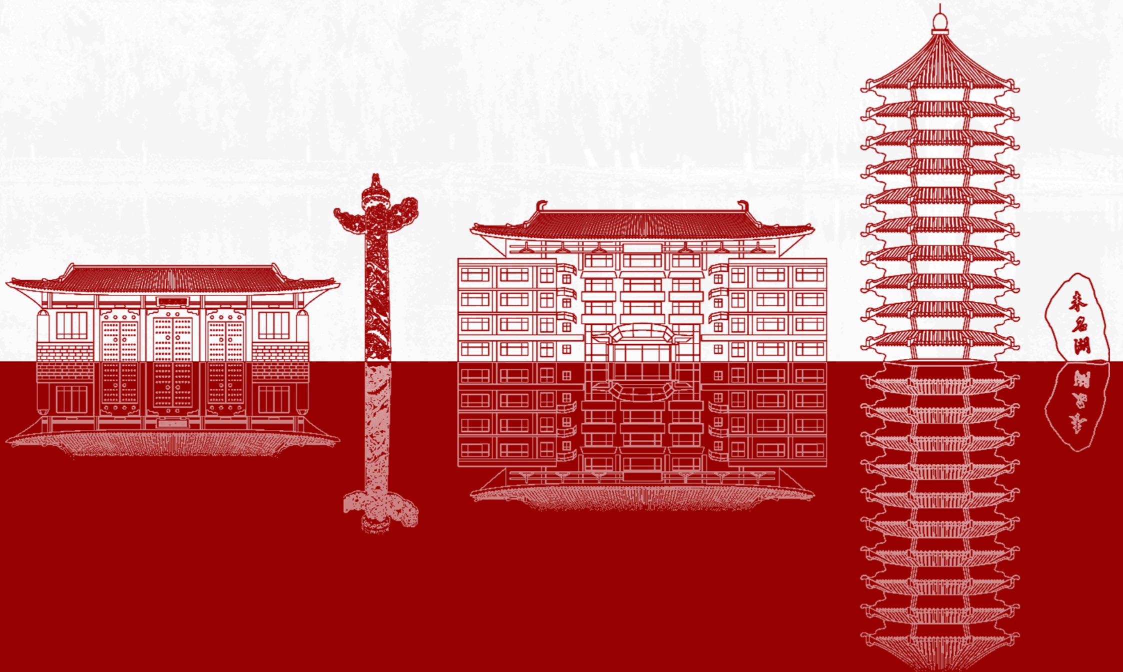
北京大學
PEKING UNIVERSITY

PaddlePi 实验介绍

(2.1) RTC: Real-Time Clock



彭伟桀

2024.10.23

RTC: Real-Time Clock/实时时钟

功能

- 查询当前时间
- 设置闹钟，闹钟到时触发中断
- 配置每日、每时、每分、每秒触发中断

其他信息

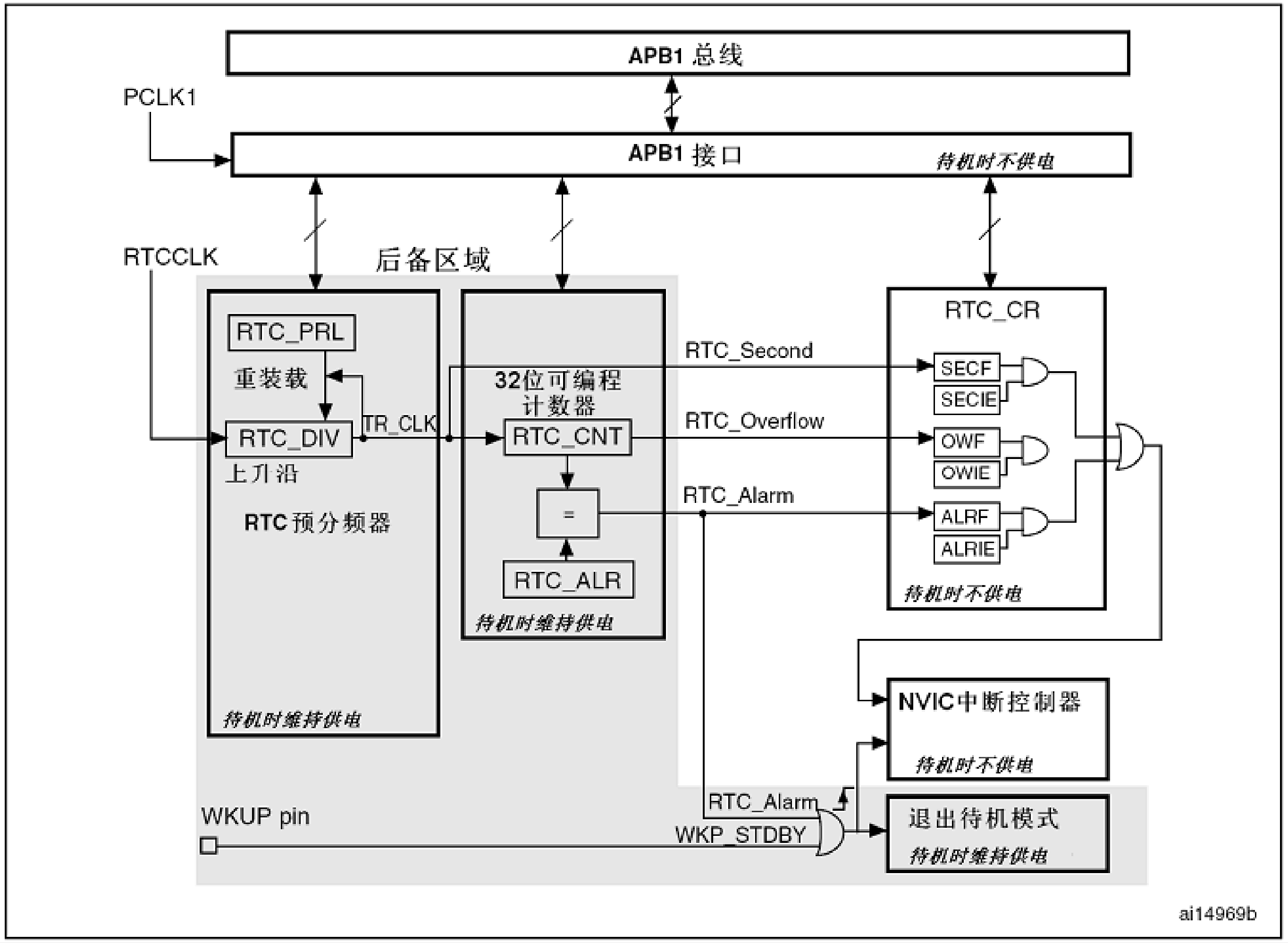
- 可以直接读出计数器计数值，最小刻度为外部晶振的单个周期
- 上电/复位后数据清零
- 内存映射 `Bus: APB1, Addr: 0x50460000, Length: 0x10000`
- 具体寄存器定义可以查阅手册，参考 `lib/drivers` 目录下 `rtc.c` 和 `include/rtc.h` 的实现
- 我们只使用 SDK 提供的封装好的 API

RTC 原理

RTC 内部就是一个计数器，与外部晶振连接，每周期计数器加一

根据配置进行单位和偏移量的转换读出现在的时间

每周期通过控制/比较逻辑来决定是否触发中断



K210 RTC API

> in file `lib/drivers/include/rtc.h`

```c
int rtc_init(void);
int rtc_timer_set(int year, int month, int day, int hour, int minute, int second);
int rtc_timer_get(int *year, int *month, int *day, int *hour, int *minute, int *second);
int rtc_alarm_set(int year, int month, int day, int hour, int minute, int second);
int rtc_tick_irq_register(bool is_single_shot, rtc_tick_interrupt_mode_t mode, plic_irq_callback_t callback, void *ctx, uint8_t priority);
int rtc_alarm_irq_register(bool is_single_shot, rtc_mask_t mask, plic_irq_callback_t callback, void *ctx, uint8_t priority);
```

- 初始化：`rtc_init`
- 设置/获取当前时间：`rtc_timer_set` / `rtc_timer_get`
- 设置闹钟：`rtc_alarm_set`
- 注册中断回调函数：`rtc_tick_irq_register` / `rtc_alarm_irq_register`

示例代码详解

> in file `src/rtc/main.c`

```c
int main(void) {
    // ...
    rtc_init();
    rtc_timer_set(2018, 9, 12, 23, 30, 50);
    rtc_alarm_set(2018, 9, 12, 23, 31, 00);
```

- 初始化 RTC
- 设置当前时间为 `2018/09/12 23:30:50`
- 设置闹钟时间为 `2018/09/12 23:31:00`

示例代码详解

> in file `src/rtc/main.c`

```c
void get_date_time(bool alarm)
{
    int year;
    int month;
    int day;
    int hour;
    int minute;
    int second;
    rtc_timer_get(&year, &month, &day, &hour, &minute, &second);
    if (!alarm)
        printf("%4d-%02d-%02d %02d:%02d:%02d\n", year, month, day, hour, minute, second);
    else
        printf("Alarm at --> %4d-%02d-%02d %02d:%02d:%02d\n", year, month, day, hour, minute, second);
}
```

- 辅助函数，获取并打印当前时间

示例代码详解

> in file `src/rtc/main.c`

```c
int on_timer_interrupt(void *ctx)
{
    get_date_time(false);
    return 0;
}

int main(void) {
    // ...
    rtc_tick_irq_register(
        false,
        RTC_INT_SECOND,
        on_timer_interrupt,
        NULL,
        1
    );
```

```c
int rtc_tick_irq_register(
    bool is_single_shot, rtc_tick_interrupt_mode_t mode,
    plic_irq_callback_t callback, void *ctx, uint8_t priority);
typedef enum
{
    RTC_INT_SECOND, /*!< 0: Interrupt every second */
    RTC_INT_MINUTE, /*!< 1: Interrupt every minute */
    RTC_INT_HOUR,   /*!< 2: Interrupt every hour */
    RTC_INT_DAY,    /*!< 3: Interrupt every day */
    RTC_INT_MAX     /*!< Max count of this enum*/
} rtc_tick_interrupt_mode_t;
```

- 注册 tick 中断回调函数，可触发多次
- 每秒中断一次，打印当前时间

示例代码详解

> in file `src/rtc/main.c`

```c
int on_alarm_interrupt(void *ctx)
{
    get_date_time(true);
    return 0;
}
int main(void) {
    // ...
    rtc_alarm_irq_register(false,
        (rtc_mask_t) {
            .second = 0, /* Second mask */
            .minute = 1, /* Minute mask */
            .hour = 0,   /* Hour mask */
            .week = 0,   /* Week mask */
            .day = 0,    /* Day mask */
            .month = 0,  /* Month mask */
            .year = 0,   /* Year mask */
        },
        on_alarm_interrupt, NULL, 1);
```

```c
int rtc_alarm_irq_register(bool is_single_shot, rtc_mask_t mask,
    plic_irq_callback_t callback, void *ctx, uint8_t priority);
 /* @param  mask  The alarm compare mask for RTC alarm interrupt
  *       (rtc_mask_t) {
  *           .second = 1, Set this mask to compare Second
  *           .minute = 0, Set this mask to compare Minute
  *           .hour = 0,   Set this mask to compare Hour
  *           .week = 0,   Set this mask to compare Week
  *           .day = 0,    Set this mask to compare Day
  *           .month = 0,  Set this mask to compare Month
  *           .year = 0,   Set this mask to compare Year
  *       } */
```

- 注册 alarm 中断回调函数，可触发多次
- 当分钟和设定值相等时触发中断

示例功能演示

- 所用编译下载命令如下：

```
# in `build` directory
cmake .. -DPROJ=rtc -DTOOLCHAIN=/opt/kendryte-toolchain/bin
make -j
kflash -b 3000000 -t rtc.bin
```

- 编译程序并下载运行
- 可以看到如右图所示的输出
- 程序每秒打印一次当前时间
- 当分钟数与闹钟设定相同时，额外打印一次

```
--- Miniterm on /dev/ttyUSB0  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
RTC Tick and Alarm Test
Compiled in Oct 22 2024 17:32:19
2018-09-12 23:30:51
2018-09-12 23:30:52
2018-09-12 23:30:53
2018-09-12 23:30:54
2018-09-12 23:30:55
2018-09-12 23:30:56
2018-09-12 23:30:57
2018-09-12 23:30:58
2018-09-12 23:30:59
Alarm at --> 2018-09-12 23:31:00
2018-09-12 23:31:00
Alarm at --> 2018-09-12 23:31:01
2018-09-12 23:31:01
Alarm at --> 2018-09-12 23:31:02
2018-09-12 23:31:02
Alarm at --> 2018-09-12 23:31:03
2018-09-12 23:31:03
Alarm at --> 2018-09-12 23:31:04
2018-09-12 23:31:04
--- exit ---
```

请大家批评指正