RTC LIBRARY

Generated by Doxygen 1.9.6

Chapter 1

RTC Library Library Integration Guide

1.1 Step 1:

Download the `rtc.c` and `rtc.h` files from the provided source or from your own implementation.

1.2 Step 2:

Add the `rtc.c` file to your project's source folder and the `rtc.h` file to your project's include folder.

1.3 Step 3:

In your main program, add the following include statement to include the RTC library header file:

```
#include "rtc.h"
```

1.4 Step 4:

Initialize the RTC module using STM32CubeIDE GUI. Configure RTC clock source and enable RTC in the main initialization code or in the Clock Configuration function (SystemClock_Config).

1.5 Step 5:

In your code, call the `RTC_SetTime()` function to set the RTC time. The function takes the `RTC_HandleTypeDef` structure and three parameters for hours, minutes, and seconds. The function returns a `HAL_StatusTypeDef` value indicating the success or failure of the operation.

```
uint8_t hours = 12;
uint8_t minutes = 30;
uint8_t seconds = 0;
HAL_StatusTypeDef status = RTC_SetTime(&hrtc, hours, minutes, seconds);
if (status != HAL_OK) {
    // Handle error
}
```

1.6 Step 6:

In your code, call the RTC_GetTime() function to get the RTC time. The function takes the RTC_HandleTypeDef structure and three pointers to uint8_t variables for hours, minutes, and seconds. The function returns a HAL_StatusTypeDef value indicating the success or failure of the operation.

```
uint8_t hours, minutes, seconds;
HAL_StatusTypeDef status = RTC_GetTime(&hrtc, &hours, &minutes, &seconds);
if (status != HAL_OK) {
    // Handle error
}
```

1.7 Step 7:

Use the values of the hours, minutes, and seconds variables as needed in your program.

Note: The RTC module needs an external battery to retain the time when the device is powered off. The battery must be connected to the Vbat pin of the STM32 microcontroller.

1.8 Example:

Below is an example:

```
    #include "rtc.h"

int main(void) {
    RTC_HandleTypeDef hrtc;
    uint8_t hours = 12;
    uint8_t minutes = 30;
    uint8_t seconds = 0;
    HAL_StatusTypeDef status = RTC_SetTime(&hrtc, hours, minutes, seconds);
    if (status != HAL_OK) {
        // Handle error
    }

    uint8_t read_hours, read_minutes, read_seconds;
    status = RTC_GetTime(&hrtc, &read_hours, &read_minutes, &read_seconds);
    if (status != HAL_OK) {
        // Handle error
    }

    // Use the read hours, minutes and seconds values
    // ...

    while (1) {
        // Main loop
    }
}
```

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

File Documentation

3.1 rtc.c File Reference

RTC Library Source.

```
#include "rtc.h"
```

Functions

- HAL_StatusTypeDef RTC_SetTime (RTC_HandleTypeDef ∗hrtc, uint8_t hours, uint8_t minutes, uint8_t seconds)

  Set the RTC time.
- HAL_StatusTypeDef RTC_GetTime (RTC_HandleTypeDef ∗hrtc, uint8_t ∗hours, uint8_t ∗minutes, uint8_t ∗seconds)

  Get the RTC time.

3.1.1 Detailed Description

RTC Library Source.

Author

Travimadox Webb @position Embedded Software Engineer @company Imperium LLC

Date

7th May 2023

Note

IMPORTANT: Initialize the RTC module using STM32CubeIDE GUI. Configure RTC clock source and enable RTC in the main initialization code or in the Clock Configuration function (SystemClock_Config).

3.1.2 Function Documentation

3.1.2.1 RTC_GetTime()

```
HAL_StatusTypeDef RTC_GetTime (
            RTC_HandleTypeDef * hrtc,
            uint8_t * hours,
            uint8_t * minutes,
            uint8_t * seconds )
```

Get the RTC time.

Parameters

hrtc	Pointer to an RTC_HandleTypeDef structure that contains the configuration information for the specified RTC
hours	Pointer to an uint8_t variable to store the hours value
minutes	Pointer to an uint8_t variable to store the minutes value
seconds	Pointer to an uint8_t variable to store the seconds value

Returns

HAL status (HAL_OK, HAL_ERROR, HAL_BUSY, or HAL_TIMEOUT)

3.1.2.2 RTC_SetTime()

```
HAL_StatusTypeDef RTC_SetTime (
            RTC_HandleTypeDef * hrtc,
            uint8_t hours,
            uint8_t minutes,
            uint8_t seconds )
```

Set the RTC time.

Parameters

hrtc	Pointer to an RTC_HandleTypeDef structure that contains the configuration information for the specified RTC
hours	Hours value to set (0-23)
minutes	Minutes value to set (0-59)
seconds	Seconds value to set (0-59)

Returns

HAL status (HAL_OK, HAL_ERROR, HAL_BUSY, or HAL_TIMEOUT)

3.2 rtc.h File Reference

RTC Library Header.

```
#include "stm32f0xx_hal.h"
```

Functions

- HAL_StatusTypeDef RTC_SetTime (RTC_HandleTypeDef ∗hrtc, uint8_t hours, uint8_t minutes, uint8_t seconds)

  Set the RTC time.
- HAL_StatusTypeDef RTC_GetTime (RTC_HandleTypeDef ∗hrtc, uint8_t ∗hours, uint8_t ∗minutes, uint8_t ∗seconds)

  Get the RTC time.

3.2.1 Detailed Description

RTC Library Header.

Author

Travimadox Webb @position Embedded Software Engineer @company Imperium LLC

Date

7th May 2023

3.2.2 Function Documentation

3.2.2.1 RTC_GetTime()

```
HAL_StatusTypeDef RTC_GetTime (
            RTC_HandleTypeDef ∗ hrtc,
            uint8_t ∗ hours,
            uint8_t ∗ minutes,
            uint8_t ∗ seconds )
```

Get the RTC time.

Parameters

hrtc	Pointer to an RTC_HandleTypeDef structure that contains the configuration information for the specified RTC
hours	Pointer to an uint8_t variable to store the hours value
minutes	Pointer to an uint8_t variable to store the minutes value
seconds	Pointer to an uint8_t variable to store the seconds value

Returns

HAL status (HAL_OK, HAL_ERROR, HAL_BUSY, or HAL_TIMEOUT)

3.2.2.2 RTC_SetTime()

```
HAL_StatusTypeDef RTC_SetTime (
            RTC_HandleTypeDef * hrtc,
            uint8_t hours,
            uint8_t minutes,
            uint8_t seconds )
```

Set the RTC time.

Parameters

hrtc	Pointer to an RTC_HandleTypeDef structure that contains the configuration information for the specified RTC
hours	Hours value to set (0-23)
minutes	Minutes value to set (0-59)
seconds	Seconds value to set (0-59)

Returns

HAL status (HAL_OK, HAL_ERROR, HAL_BUSY, or HAL_TIMEOUT)

3.3 rtc.h

Go to the documentation of this file.

```

#ifndef __RTC_H
#define __RTC_H

#include "stm32f0xx_hal.h"

HAL_StatusTypeDef RTC_SetTime(RTC_HandleTypeDef *hrtc, uint8_t hours, uint8_t minutes, uint8_t seconds);

HAL_StatusTypeDef RTC_GetTime(RTC_HandleTypeDef *hrtc, uint8_t *hours, uint8_t *minutes, uint8_t *seconds);

#endif // __RTC_H
```

Index