STM32时钟系统

1，认识时钟树（掌握）

1.1，什么是时钟?

简单来说，时钟是具有周期性的脉冲信号，最常用的是占空比50%的方波

时钟是单片机的脉搏，搞懂时钟走向及关系，对单片机使用至关重要!

1.2，认识时钟树（F1）

1.3，认识时钟树（F4）

1.4，认识时钟树（F7）

1.5，认识时钟树（H7）

2，配置系统时钟（掌握）

2.1，系统时钟配置步骤

1，配置HSE_VALUE　　告诉HAL库外部晶振频率，stm32xxxx_hal_conf.h

2，调用SystemInit()函数（可选）　　在启动文件中调用， 在system_stm32xxxx.c定义

3，选择时钟源，配置PLL　　通过HAL_RCC_OscConfig()函数设置

4，选择系统时钟源，配置总线预分频　　通过HAL_RCC_ClockConfig()函数设置

5，配置扩展外设时钟（可选）　　通过HAL_RCCEx_PeriphCLKConfig()函数设置

3 + 4 + 5 = sys_stm32_clock_init()

2.2，外设时钟使能和失能

我们要使用某个外设，必需先使能该外设时钟！！！

HAL库使能某个外设时钟的方法，如：

```
__HAL_RCC_GPIOA_CLK_ENABLE();     /* 使能 GPIOA 时钟 */
```

HAL库禁止某个外设时钟的方法，如：

```
__HAL_RCC_GPIOA_CLK_DISABLE();     /* 禁止 GPIOA 时钟 */
```

2.3，sys_stm32_clock_init 函数（F1）

1， HAL_StatusTypeDef HAL_RCC_OscConfig(RCC_OscInitTypeDef *RCC_OscInitStruct)

```
typedef struct
{
  uint32_t OscillatorType;        /* 选择需要配置的振荡器 */
  uint32_t HSEState;              /* HSE 状态 */
  uint32_t HSEPredivValue;        /* HSE 预分频值 */
  uint32_t LSEState;              /* LSE 状态 */
  uint32_t HSIState;              /* HSI状态 */
  uint32_t HSICalibrationValue;   /* HSI 校准值 */
  uint32_t LSIState;              /* LSI 状态 */
  RCC_PLLInitTypeDef PLL;         /* PLL 结构体 */
}RCC_OscInitTypeDef;
```

```
typedef struct
{
  uint32_t PLLState;      /* PLL 状态 */
  uint32_t PLLSource;     /* PLL 时钟源 */
  uint32_t PLLMUL;        /* PLL 倍频系数 */
}RCC_PLLInitTypeDef;
```

2，HAL_StatusTypeDef HAL_RCC_ClockConfig(RCC_ClkInitTypeDef *RCC_ClkInitStruct, uint32_t FLatency)

```
typedef struct
{
  uint32_t ClockType;             /* 要配置的时钟（SYSCLK/HCLK/PCLK1/PCLK2）  */
  uint32_t SYSCLKSource;          /* 系统时钟源 */
  uint32_t AHBCLKDivider;         /* AHB  时钟预分频系数 */
  uint32_t APB1CLKDivider;        /* APB1 时钟预分频系数 */
  uint32_t APB2CLKDivider;        /* APB2 时钟预分频系数 */
}RCC_ClkInitTypeDef;
```

```
uint32_t FLatency

#define FLASH_LATENCY_0 0x00000000U               /* FLASH 0个等待周期 */
#define FLASH_LATENCY_1 FLASH_ACR_LATENCY_0       /* FLASH 1个等待周期 */
#define FLASH_LATENCY_2 FLASH_ACR_LATENCY_1       /* FLASH 2个等待周期 */
```

2.4，sys_stm32_clock_init 函数（F4/F7）

1， HAL_StatusTypeDef HAL_RCC_OscConfig(RCC_OscInitTypeDef *RCC_OscInitStruct)

```
typedef struct
{
  uint32_t OscillatorType;        /* 选择需要配置的振荡器 */
  uint32_t HSEState;              /* HSE 状态 */
  uint32_t LSEState;              /* LSE 状态 */
  uint32_t HSIState;              /* HSI 状态 */
  uint32_t HSICalibrationValue;   /* HSI 校准微调值，范围0x0~0x1F */
  uint32_t LSIState;              /* LSI 状态 */
  RCC_PLLInitTypeDef PLL;         /* PLL 结构体 */
}RCC_OscInitTypeDef;
```

```
typedef struct
{
  uint32_t PLLState;      /* PLL 状态 */
  uint32_t PLLSource;     /* PLL 时钟源 */
  uint32_t PLLM;          /* PLL 分频系数 M */
  uint32_t PLLN;          /* PLL 倍频系数 N */
  uint32_t PLLP;          /* PLL 分频系数 P */
  uint32_t PLLQ;          /* PLL 分频系数 Q */
}RCC_PLLInitTypeDef;
```

2，HAL_StatusTypeDef HAL_RCC_ClockConfig(RCC_ClkInitTypeDef *RCC_ClkInitStruct, uint32_t FLatency)

```
typedef struct
{
  uint32_t ClockType;             /* 要配置的时钟（SYSCLK/HCLK/PCLK1/PCLK2）  */
  uint32_t SYSCLKSource;          /* 系统时钟源 */
  uint32_t AHBCLKDivider;         /* AHB  时钟预分频系数 */
  uint32_t APB1CLKDivider;        /* APB1 时钟预分频系数 */
  uint32_t APB2CLKDivider;        /* APB2 时钟预分频系数 */
}RCC_ClkInitTypeDef;
```

```
uint32_t FLatency

#define FLASH_LATENCY_0 FLASH_ACR_LATENCY_0WS     /* FLASH 0个等待周期 */
#define FLASH_LATENCY_1 FLASH_ACR_LATENCY_1WS     /* FLASH 1个等待周期 */
#define FLASH_LATENCY_2 FLASH_ACR_LATENCY_2WS     /* FLASH 2个等待周期 */
...
#define FLASH_LATENCY_15 FLASH_ACR_LATENCY_15WS   /* FLASH 15个等待周期 */
```

2.5，sys_stm32_clock_init 函数（H7）

1， HAL_StatusTypeDef HAL_RCC_OscConfig(RCC_OscInitTypeDef *RCC_OscInitStruct)

```
typedef struct
{
  uint32_t OscillatorType;        /* 选择需要配置的振荡器 */
  uint32_t HSEState;              /* HSE 状态 */
  uint32_t LSEState;              /* LSE 状态 */
  uint32_t HSIState;              /* HSI 状态 */
  uint32_t HSICalibrationValue;   /* HSI 校准微调值 */
  uint32_t LSIState;              /* LSI 状态 */
  uint32_t HSI48State;            /* HSI48 状态 */
  uint32_t CSIState;              /* CSI 状态 */
  uint32_t CSICalibrationValue;   /* CSI 校准微调值 */
  RCC_PLLInitTypeDef PLL;         /* PLL 结构体 */
}RCC_OscInitTypeDef;
```

```
typedef struct
{
  uint32_t PLLState;    /* PLL1 状态 */
  uint32_t PLLSource;   /* PLL1 时钟源 */
  uint32_t PLLM;        /* PLL1 分频系数 M */
  uint32_t PLLN;        /* PLL1 倍频系数 N */
  uint32_t PLLP;        /* PLL1 分频系数 P */
  uint32_t PLLQ;        /* PLL1 分频系数 Q */
  uint32_t PLLR;        /* PLL1 分频系数 R */
  uint32_t PLLRGE;      /* PLL1 时钟输入范围 */
  uint32_t PLLVCOSEL;   /* PLL1 时钟输出范围 */
  uint32_t PLLFRACN;    /* PLL1 VCO 乘数因子的小数部分 */
}RCC_PLLInitTypeDef;
```

2，HAL_StatusTypeDef HAL_RCC_ClockConfig(RCC_ClkInitTypeDef *RCC_ClkInitStruct, uint32_t FLatency)

```
typedef struct
{
  uint32_t ClockType;             /* 要配置的时钟 */
  uint32_t SYSCLKSource;          /* 系统时钟源 */
  uint32_t SYSCLKDivider;         /* SYSCLK 分频系数 */
  uint32_t AHBCLKDivider;         /* AHB  时钟预分频系数 */
  uint32_t APB3CLKDivider;        /* APB3 时钟预分频系数 */
  uint32_t APB1CLKDivider;        /* APB1 时钟预分频系数 */
  uint32_t APB2CLKDivider;        /* APB2 时钟预分频系数 */
  uint32_t APB4CLKDivider;        /* APB4 时钟预分频系数 */
}RCC_ClkInitTypeDef;
```

```
uint32_t FLatency

#define FLASH_LATENCY_0 FLASH_ACR_LATENCY_0WS     /* FLASH 0个等待周期 */
#define FLASH_LATENCY_1 FLASH_ACR_LATENCY_1WS     /* FLASH 1个等待周期 */
#define FLASH_LATENCY_2 FLASH_ACR_LATENCY_2WS     /* FLASH 2个等待周期 */
...
#define FLASH_LATENCY_15 FLASH_ACR_LATENCY_15WS   /* FLASH 15个等待周期 */
```

2，HAL_StatusTypeDef HAL_RCCEx_PeriphCLKConfig(RCC_PeriphCLKInitTypeDef *PeriphClkInit)

```
typedef struct
{
  uint32_t PeriphClockSelection;          /* 要配置的扩展时钟 */
  RCC_PLL2InitTypeDef PLL2;               /* PLL2 时钟配置结构体 */
  RCC_PLL3InitTypeDef PLL3;               /* PLL3 时钟配置结构体 */
  uint32_t FmcClockSelection;             /* FMC 时钟源 */
  uint32_t QspiClockSelection;            /* QSPI 时钟源 */
  uint32_t Usart234578ClockSelection;     /* USART2/3/4/5/7/8 时钟源 */
  uint32_t Usart16ClockSelection;         /* USART1/6 时钟源 */
  uint32_t UsbClockSelection;             /* USB 时钟源 */
  ...省略...
}RCC_PeriphCLKInitTypeDef;
```

```
typedef struct
{
  uint32_t PLL2M;          /* PLL2 分频系数 M */
  uint32_t PLL2N;          /* PLL2 倍频系数 N */
  uint32_t PLL2P;          /* PLL2 倍频系数 P */
  uint32_t PLL2Q;          /* PLL2 倍频系数 Q */
  uint32_t PLL2R;          /* PLL2 倍频系数 R */
  uint32_t PLL2RGE;        /* PLL2 时钟输入范围 */
  uint32_t PLL2VCOSEL;     /* PLL2 时钟输出范围 */
  uint32_t PLL2FRACN;      /* PLL2 VCO 乘数因子的小数部分 */
}RCC_PLL2InitTypeDef;
```

3，课堂总结（了解）