

RCC a strom hodin

Reset and Clock Control (RCC)

Naším dalším tématem je obvod Reset and Clock Control (RCC). Začneme resetovacím obvodem a jeho funkcí. Existují tři typy resetu, definované jako reset systému, reset napájení a reset záložní domény.

Reset napájení se generuje, když nastane jedna z následujících událostí:

1. Brown-out Reset (BOR).
2. při ukončení pohotovostního režimu.
3. při ukončení režimu vypnutí.

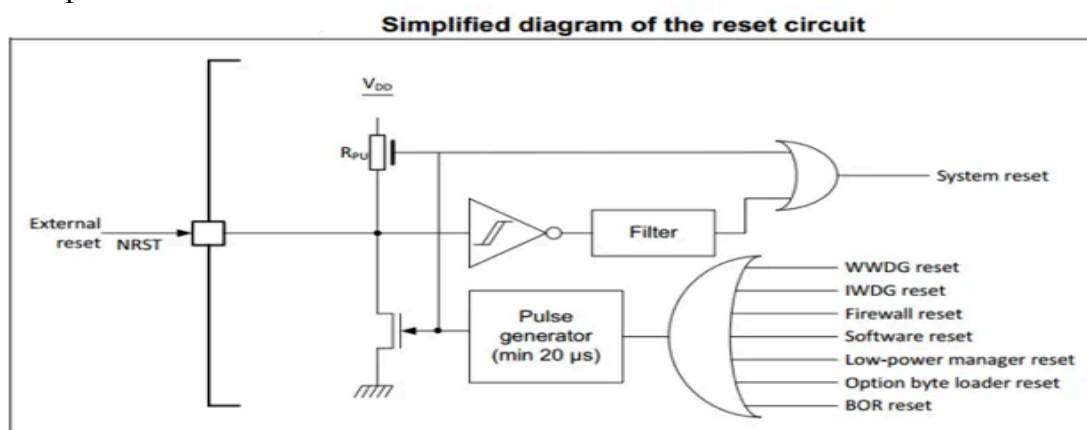
Reset systému nastaví všechny registry na jejich resetované hodnoty, pokud není v popisu registru uvedeno jinak. Zdroj resetování lze identifikovat kontrolou příznaků resetování v registru Control/Status, RCC_CSR.

Resetování systému je generováno, když nastane jedna z následujících událostí:

1. Nízká úroveň na kolíku NRST (externí reset)
2. Událost hlídání okna (resetování WWDG)
3. Událost nezávislého hlídače (resetování IWDG)
4. Událost brány firewall (resetování brány FIREWALL)
5. Reset softwaru (reset SW)
6. Reset zabezpečení režimu nízké spotřeby
7. Resetování zavaděče bajtů možností
8. Brown-out reset
9. Softwarový reset: Bit SYSRESETREQ v Cortex®-M4 Application Interrupt and Reset Control Register musí být nastaven tak, aby vynutil reset softwaru na zařízení.

Resetování záložní domény má dva specifické resetovací signály. Reset záložní domény se vygeneruje, když nastane jedna z následujících událostí:

1. Reset softwaru, spuštěný nastavením bitu BDRST v řídicím registru záložní domény (RCC_BDCR).
2. Napájení V nebo V zapnuto, pokud byly oba zdroje předtím vypnuty. Reset záložní domény ovlivní pouze oscilátor LSE, RTC, záložní registry a řídicí registr domény RCC Backup.



Strom hodin

K řízení systémových hodin (SYSCLK) lze použít čtyři různé zdroje hodin:

1. Hodiny RC oscilátoru HSI16 (vysokorychlostní interní) 16 MHz
2. MSI (multispeed internal) RC hodiny oscilátoru
3. Hodiny oscilátoru HSE, od 4 do 48 MHz
4. PLL hodiny

MSI se používá jako zdroj systémových hodin po spuštění z Resetu, nakonfigurovaného na 4 MHz.

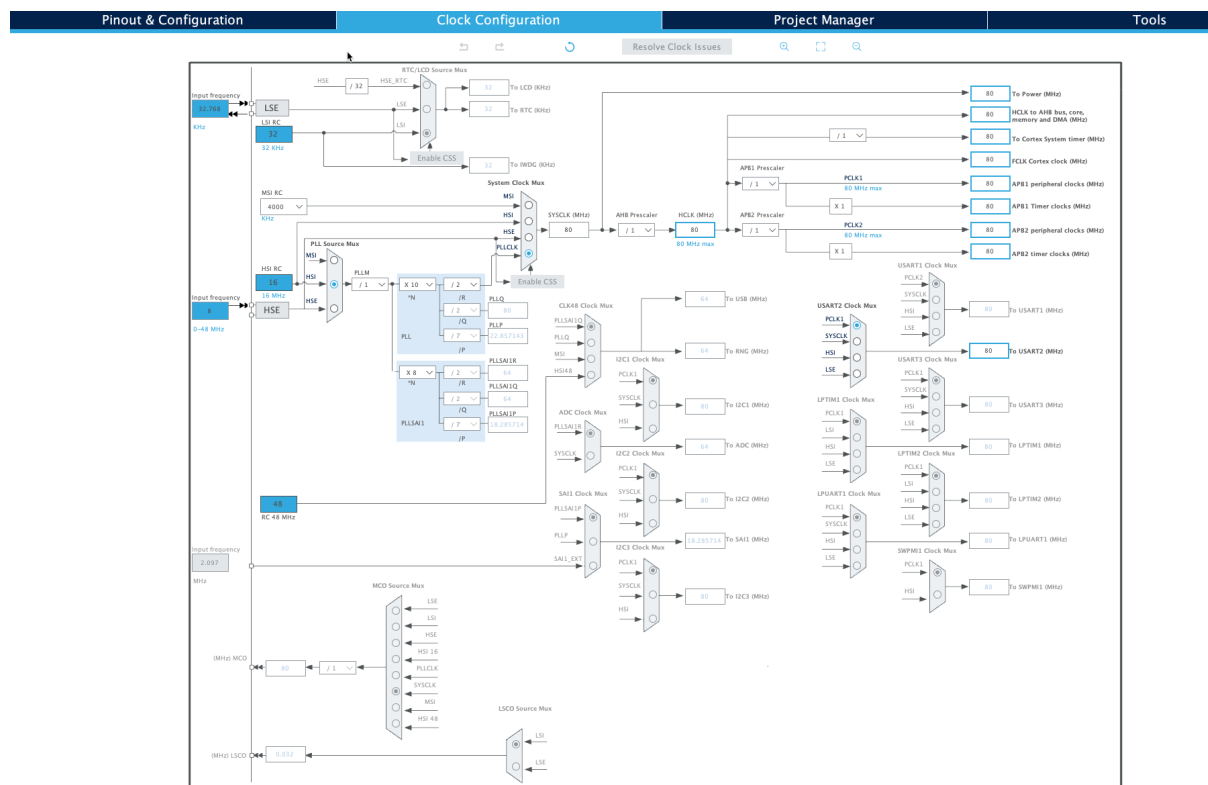
Zařízení mají následující dodatečné zdroje hodin:

- 32 kHz nízkorychlostní interní RC (LSI RC), který řídí nezávislý watchdog a volitelně RTC používaný pro automatické probuzení z režimů Stop a Standby.
- Nízkorychlostní externí krystal 32,768 kHz (LSE crystal), který volitelně řídí hodiny reálného času (RTCCLK).
- RC 48 MHz interní zdroje hodin (HSI48) pro potenciální řízení USB FS, SDMMC a RNG.

Každý zdroj hodin lze zapnout nebo vypnout nezávisle, když se nepoužívá, pro optimalizaci spotřeby energie. CubeMX použijeme v kroku konfigurace stromu hodin. Kde můžeme nastavit požadovanou rychlost SYSCLK (až 80MHz pro L432KC nebo až 72MHz pro F103C8). Můžeme také nastavit rychlost hodin pro různá periferní zařízení, sběrnice a hardware USB.

Celý systém hodin (Clock) lze nastavit buď graficky v STM32CubeMX nebo textově v main.c (main.cpp)

1. Graficky:



Obrázek 1 Strom hodin v STM32CubeMX

2. Textově v main.c

```
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /** Initializes the RCC Oscillators according to the specified parameters
    * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLM = 1;
    RCC_OscInitStruct.PLL.PLLN = 10;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV7;
    RCC_OscInitStruct.PLL.PLLQ = RCC_PLLQ_DIV2;
    RCC_OscInitStruct.PLL.PLLR = RCC_PLLR_DIV2;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
    /** Initializes the CPU, AHB and APB buses clocks
    */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_4) != HAL_OK)
    {
        Error_Handler();
    }
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART2;
    PeriphClkInit.Usart2ClockSelection = RCC_USART2CLKSOURCE_PCLK1;
    if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
    {
        Error_Handler();
    }
    /** Configure the main internal regulator output voltage
    */
    if (HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR_VOLTAGE_SCALE1) != HAL_OK)
    {
        Error_Handler();
    }
}
```