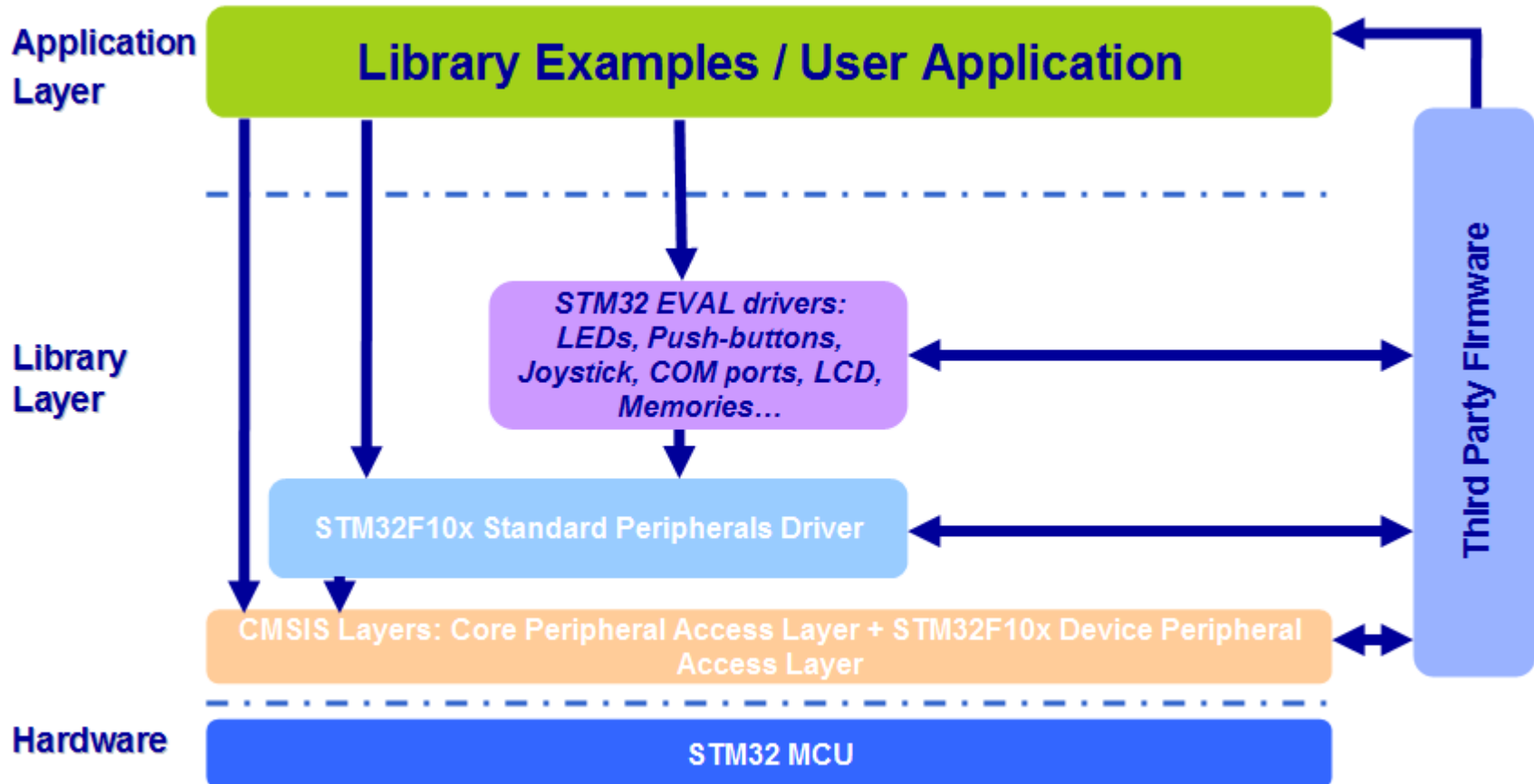


STM32F10x - Cortex M3

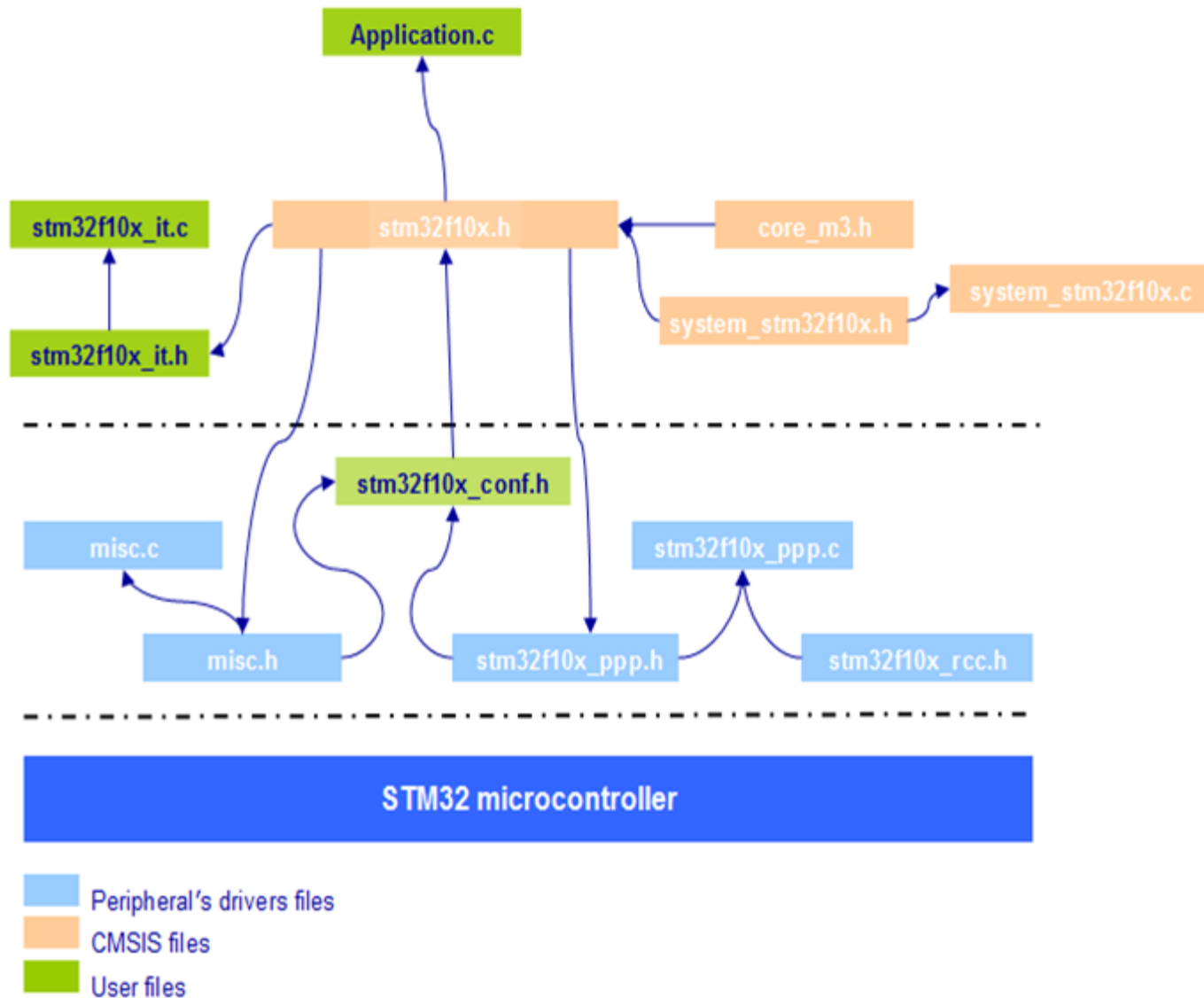
Software

STM32F10x Standard Peripherals Library

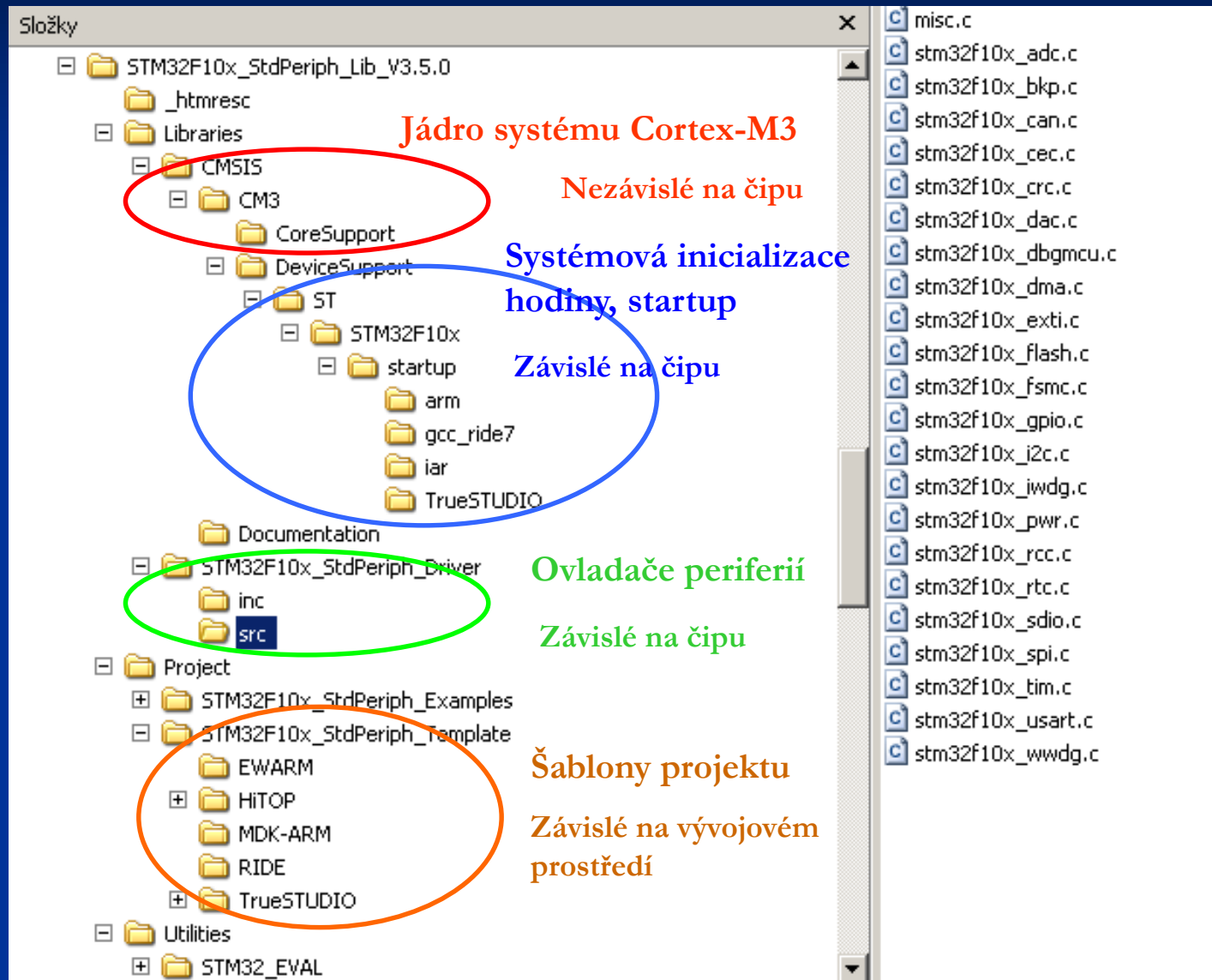
STM32 - Softwarová hierarchie



STM32 - Struktura knihovních souborů



Knihovna STM32F10x Standard Peripherals Library



CMSIS

Cortex Microcontroller Software Interface Standard

- Knihovna CMSIS dodávána přímo společností ARM Ltd.
- Nezávislá na konkrétním čipu výrobce
- Určena pro práci s přímými registry jádra Cortex-Mx
- Obsahuje definici registrů a funkcí pro interní periférie jádra Cortex-Mx
- Definice výjimek a přerušení a funkcí pro práci s přerušením

S řadou **STM32F10x** (STM32F103xx, 107xx, ...) aktuálně dodávána knihovna periférií

STM32F10x_StdPeriph_Lib_V3.5.0

Tato knihovna zahrnuje **CMSIS V1.30** - říjen 2009



Aktuální stabilní verze CMSIS je **CMSIS V4.5.0** – říjen 2015 (M0, M3, M4, M7)

Beta verze: **CMSIS V5.0.0 -Beta4**

Používané verze CMSIS

S řadou **STM32F10x** (STM32F103xx, 107xx, ...) aktuálně dodávána knihovna periférií **STM32F10x_StdPeriph_Lib_V3.5.0**, obsahující **CMSIS V1.30** - říjen 2009

ALE !!!

Aktuální firmware **STM32_Nucleo_FW_V1.2.1** dodávaný pro kit **Nucleo103RB** obsahuje knihovnu periférií **STM32F10x_StdPeriph_Lib_V3.6.x**, s verzí **CMSIS V3.20** - únor 2013 (podpora jader M0, M3, M4)

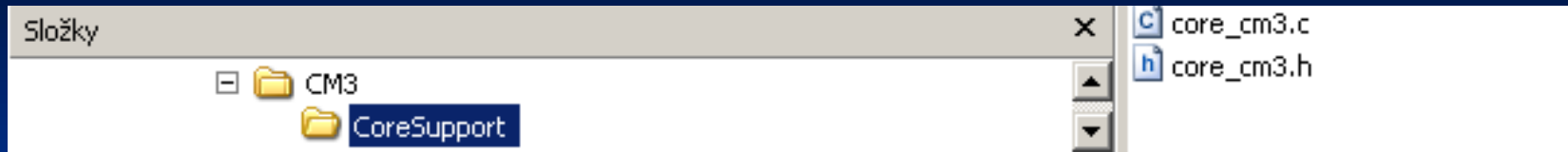
Keil MDK V5.xy obsahuje ve svých implicitních knihovnách knihovnu periférií **STM32F10x_StdPeriph_Lib_V3.5.0** a **CMSIS Vi.jk** – závislé na konkrétní instalaci verze balíku CMSIS (podpora jader M0, M3, M4, M7).

!!! Soubor **stm32f10x.h** je proprietárním souborem prostředí Keil MDK **V5.xx** - obsahuje podporu konfiguračního souboru **RTE_Components.h** generovanou proprietárním wizardem Keil a liší se od standardně dodávaných knihoven STMicroelectronics. Veškeré definice **#define** v tomto souboru jsou "natvrdo" a nelze je modifikovat.

Stejně tak nelze zasahovat do implicitních nastavení cest k hlavičkovým souborům knihoven ***.h** a modifikovat jejich umístění.

Jedinou možností „ruční“ konfigurace je přímá modifikace odpovídajícího *.pds c souboru

CMSIS - Core Peripheral Access Layer



core_cm3.h

- definice adres, struktur, symbolických jmen a bitových polí registrů periférií integrovaných v jádře Cortex-M3:

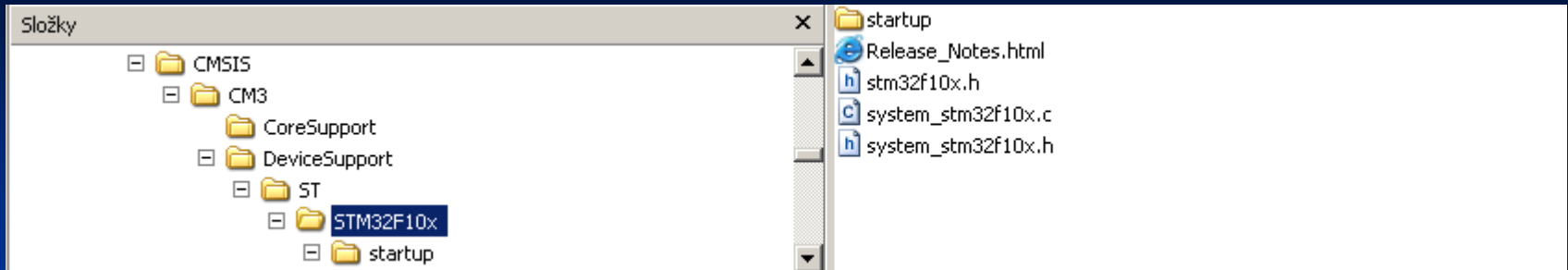
NVIC, SCB, SYSTIC, MPU, ITM, Core Debug

- implementace knihovních funkcí pro **NVIC, SYSTIC, ITM**
- definice maker a funkcí pro povolení maskovatelných přerušení a wrap funkcí vybraných instrukcí (NOP, WFI, WFE, SEV, ISB, DSB, DMB, CLREX)
- deklarace (případně už i in-line definice - závislé na kompilátoru) funkcí pro přístup k PSP, MSP, CONTROL, PRIMASK, FAULTMASK, BASEPRI a užití instrukcí typu REV, RBIT, LDREX, STREX

core_cm3.c

- definice funkcí pro přístup k PSP, MSP, CONTROL, PRIMASK, FAULTMASK, BASEPRI a užití instrukcí typu, REV, RBIT, LDREX, STREX

CMSIS - Device Support (STM32F10x) I



system_stm32f10x.c

- definice funkce `void SystemInit(void)` - kompletní nastavení zdrojových hodin mikrokontroléru a všech PLL závěsů. Funkce volána z assemblerového modulu startup před voláním funkce main.
- definice globální proměnné `unsigned int SystemCoreClock` - frekvence hodin [Hz] systémové AHB sběrnice (72 MHz pro Connectivity Line) - možno libovolně použít v aplikaci.
- nutno explicitně definovat (odkomentovat) požadovanou frekvenci systémových hodin `SYSClk_FREQ_xxxxx`, jinak bude defaultně použit HSI s frekvencí 8 MHz:

`#define SYSClk_FREQ_72MHz 72000000` (pro Connectivity Line)

- definice funkce `void SystemCoreClockUpdate(void)` - nutno volat vždy, když jsou měněny za běhu programu hodiny jádra - update `SystemCoreClock`

CMSIS - Device Support (STM32F10x) II

stm32f10x.h

- jediný hlavičkový soubor, který je třeba includovat do uživatelské aplikace
 - definice struktur, bázových adres, registrů jejich bitových polí pro každou periférii
 - definice enumeračních typů (TRUE, FALSE, ENABLE, DISABLE, ERROR, SUCCESS)
-

Nezbytná uživatelská konfigurace:

`#define STM32F10X_CL`

Nutno vybrat !!! Defaultně není definováno nic a překladač hlásí chybu

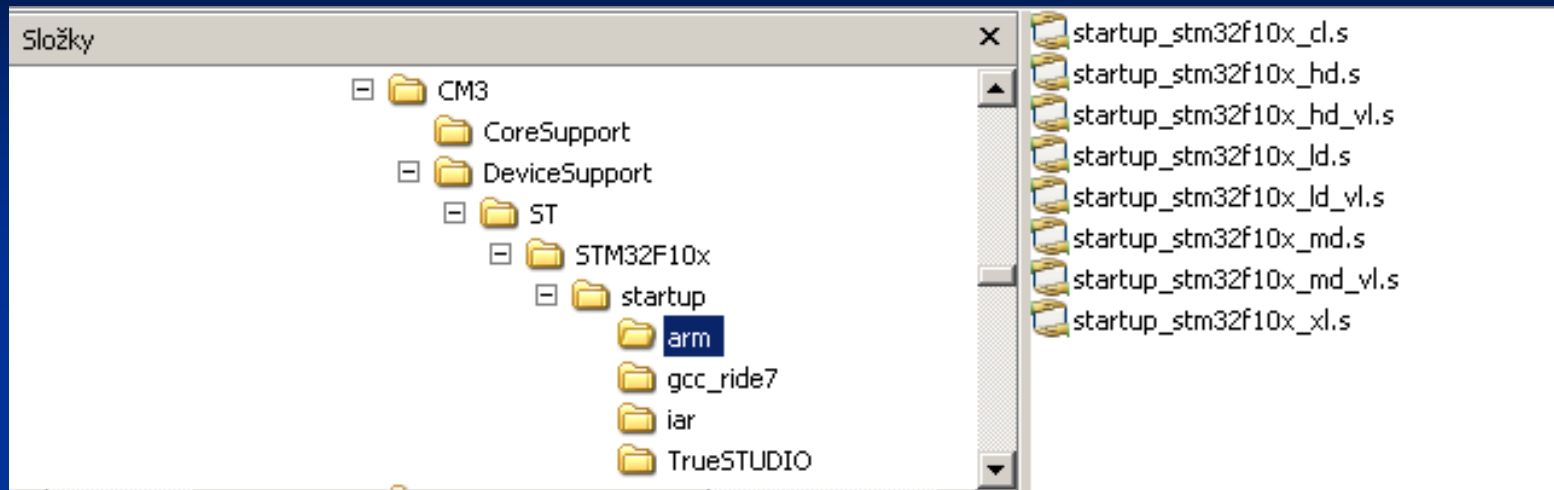
`#define USE_STDPERIPH_DRIVER`

Definuje, zda budou použity ST knihovny, či přímý přístup do registrů

`#define HSE_VALUE xxxx`

Předefinovat v případě, že je použit jiný kmitočet HSE oscilátoru než default (25 MHz pro CL, 8 MHz pro ostatní)

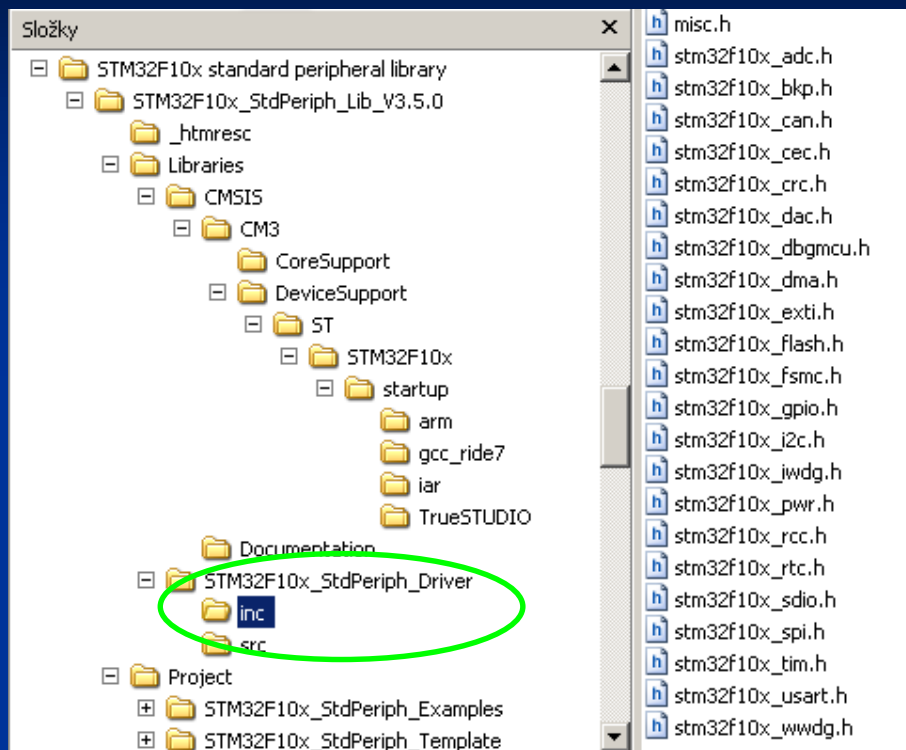
CMSIS - Device Support (STM32F10x) III



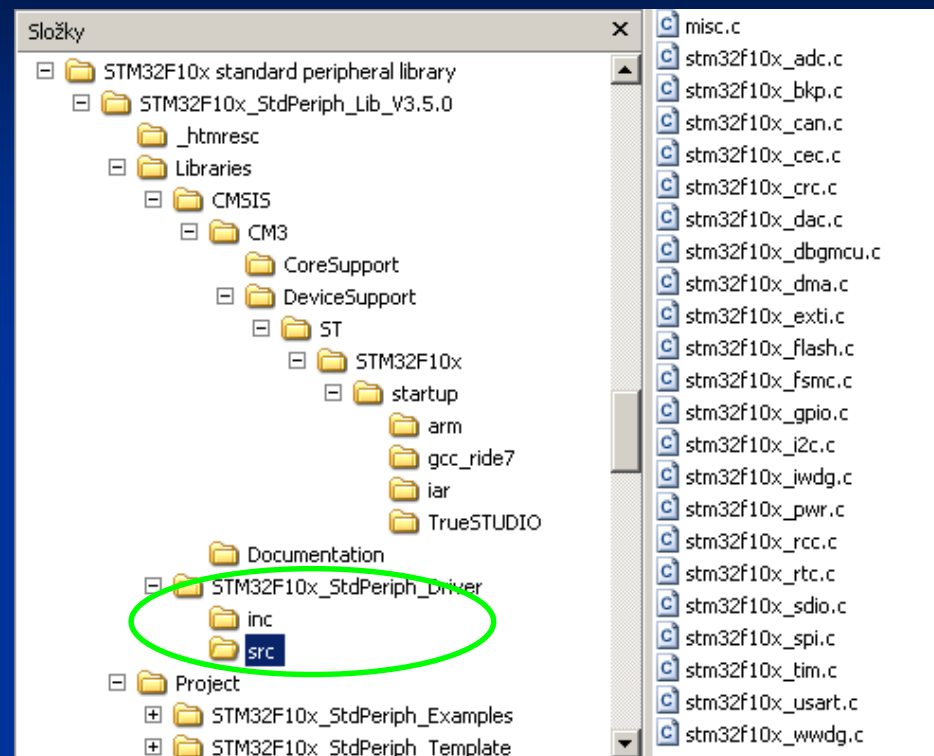
startup_stm32f10x_cl.s

- inicializace přerušovacích vektorů - koresponduje se jmény obslužných funkcí v `stm32f10_it.c`
- volá funkci `system_init()` z modulu `system_stm32f10x.c` pro nastavení hodin
- inicializuje ukazatele zásobníků
- předává řízení uživatelské aplikaci - funkci `main()`

STM32F10x_StdPeriph_Driver



stm32f10x_ppp.h

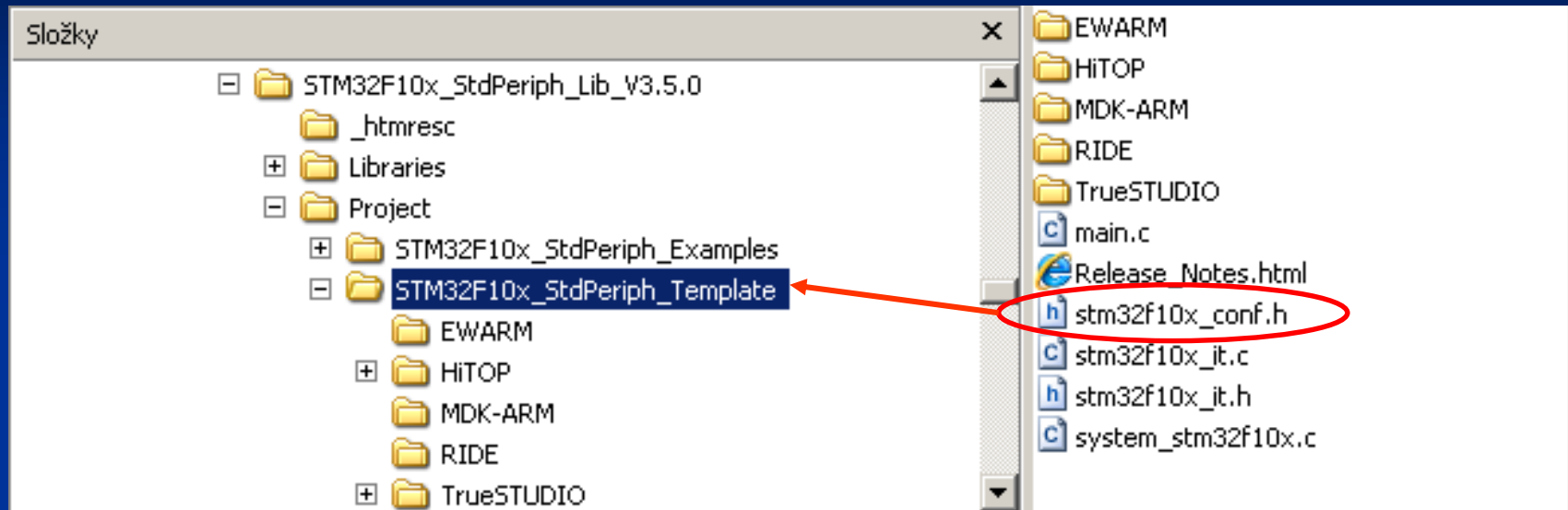


stm32f10x_ppp.c

Každá integrovaná periférie mikrokontroléru je podporována unifikovanou sadou funkcí.

Konkrétní periférii je nutné povolit v konfiguračním souboru **stm32f10x_conf.h**

Konfigurace periférií



Konfigurační soubor **stm32f10x_conf.h** není součástí knihovny, ale projektové šablony, dodávané s balíkem knihovny.

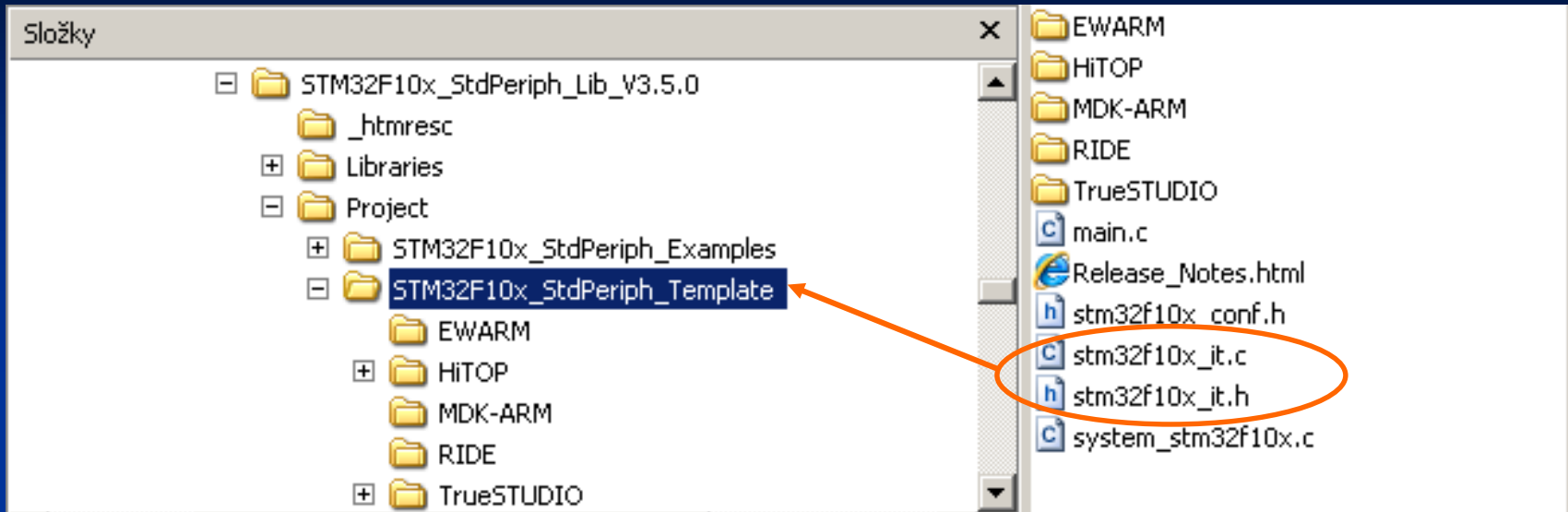
stm32f10x_conf.h

```
/* Includes -----*/
/* Uncomment/Comment the line below to enable/disable peripheral header file inclusion */
/* #include "stm32f10x_adc.h" */
/* #include "stm32f10x_bkp.h" */
/* #include "stm32f10x_can.h" */
/* #include "stm32f10x_cec.h" */
/* #include "stm32f10x_crc.h" */
/* #include "stm32f10x_dac.h" */
/* #include "stm32f10x_dbgmcu.h" */
/* #include "stm32f10x_dma.h" */
/* #include "stm32f10x_exti.h" */
/* #include "stm32f10x_flash.h" */
/* #include "stm32f10x_fsmc.h" */
#include "stm32f10x_gpio.h" /**/
/* #include "stm32f10x_i2c.h" */
/* #include "stm32f10x_iwdg.h" */
/* #include "stm32f10x_pwr.h" */
#include "stm32f10x_rcc.h" /**/
/* #include "stm32f10x_rtc.h" */
/* #include "stm32f10x_sdio.h" */
/* #include "stm32f10x_spi.h" */
/* #include "stm32f10x_tim.h" */
```

Fragment konfiguračního souboru, povolující použití **GPIO** portů.

Inicializace **RCC - Reset Clock Control** je nezbytná pro inicializaci - povolení hodin periférie a reset periférie.

Obsluha výjimek a přerušení I.



Hlavičkový **stm32f10x_it.h** a implementační soubor **stm32f10x_it.c** pro obsluhu výjimek a přerušení nejsou součástí knihovny, ale projektové šablony, dodávané s balíkem knihovny.

stm32f10x_it.c obsahuje prázdné předdefinované funkce pro obsluhu výjimek a interruptů jádra a univerzální předpis pro obecná externí přerušení periférií:

```
void PPP_IRQHandler(void)
```

```
{  
{  
}
```

↖ Nahradiť konkrétním názvem zdroje, shodně s definicí startup kódu

Každý používaný interrupt musí být v tomto souboru doimplementován !!!

Obsluha výjimek a přerušení II.

```
_Vectors      DCD      __initial_sp          ; Top of Stack
              DCD      Reset_Handler       ; Reset Handler
              DCD      NMI_Handler         ; NMI Handler
              DCD      HardFault_Handler   ; Hard Fault Handler
              DCD      MemManage_Handler   ; MPU Fault Handler
              DCD      BusFault_Handler     ; Bus Fault Handler
              DCD      UsageFault_Handler   ; Usage Fault Handler
              DCD      0                    ; Reserved
              DCD      0                    ; Reserved
              DCD      0                    ; Reserved
              DCD      0                    ; Reserved
              DCD      SVC_Handler          ; SVC Call Handler
              DCD      DebugMon_Handler    ; Debug Monitor Handler
              DCD      0                    ; Reserved
              DCD      PendSV_Handler       ; PendSV Handler
              DCD      SysTick_Handler      ; SysTick Handler

; External Interrupts
              DCD      WWDG_IRQHandler      ; Window Watchdog
              DCD      PVD_IRQHandler        ; PVD through EXTI Line detect
              DCD      TAMPER_IRQHandler      ; Tamper
              DCD      RTC_IRQHandler         ; RTC
              DCD      FLASH_IRQHandler       ; Flash
              DCD      RCC_IRQHandler         ; RCC
              DCD      EXTI0_IRQHandler       ; EXTI Line 0
              DCD      EXTI1_IRQHandler       ; EXTI Line 1
              DCD      EXTI2_IRQHandler       ; EXTI Line 2
              DCD      EXTI3_IRQHandler       ; EXTI Line 3
              DCD      EXTI4_IRQHandler       ; EXTI Line 4
              DCD      DMA1_Channel1_IRQHandler ; DMA1 Channel 1
              DCD      DMA1_Channel2_IRQHandler ; DMA1 Channel 2
              DCD      DMA1_Channel3_IRQHandler ; DMA1 Channel 3
              DCD      DMA1_Channel4_IRQHandler ; DMA1 Channel 4
              DCD      DMA1_Channel5_IRQHandler ; DMA1 Channel 5
              DCD      DMA1_Channel6_IRQHandler ; DMA1 Channel 6
              DCD      DMA1_Channel7_IRQHandler ; DMA1 Channel 7
              DCD      ADC1_2_IRQHandler     ; ADC1 and ADC2
              DCD      CAN1_TX_IRQHandler    ; CAN1 TX
```

Výjimky a přerušení jádra Cortex-M3 jsou v souboru `stm32f10x_it.c` defaultně obsloženy prázdnými funkcemi

Příklad obsluhy externího interruptu:

```
void EXTI0_IRQHandler(void)
{
}
}
```


STM32F10x Exceptions-Interrupts

Exception number ⁽¹⁾	IRQ number ⁽¹⁾	Exception type	Priority	Vector address or offset ⁽²⁾
1	-	Reset	-3, the highest	0x00000004
2	-14	NMI	-2	0x00000008
3	-13	Hard fault	-1	0x0000000C
4	-12	Memory management fault	Configurable ⁽³⁾	0x00000010
5	-11	Bus fault	Configurable ⁽³⁾	0x00000014
6	-10	Usage fault	Configurable ⁽³⁾	0x00000018
7-10	-	-	-	Reserved
11	-5	SVCall	Configurable ⁽³⁾	0x0000002C
12-13	-	-	-	Reserved
14	-2	PendSV	Configurable ⁽³⁾	0x00000038
15	-1	SysTick	Configurable ⁽³⁾	0x0000003C
16-83	0-67	Interrupt (IRQ)	Configurable ⁽⁴⁾	0x00000040 and above ⁽⁵⁾

CMSIS knihovny

31	30	29	28	27	26	25	24	23	16	15	10	9	8	0
N	Z	C	V	Q	IC/IT	T	Reserved				IC/IT		ISR_NUMBER	

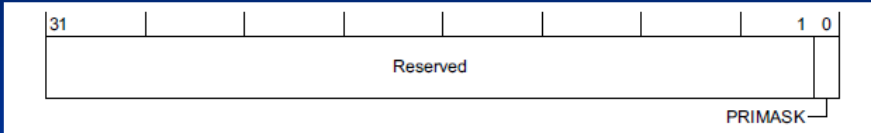
Reserved

PSR (IPSR) registr

Exception number	IRQ number	Offset	Vector
83	67	0x014C	IRQ67
.	.	.	.
.	.	.	.
.	.	.	.
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
13		0x0038	Reserved
12			Reserved for Debug
11	-5		SVCall
10		0x002C	Reserved
9			
8			
7			
6	-10		Usage fault
5	-11	0x0018	Bus fault
4	-12	0x0014	Memory management fault
3	-13	0x0010	Hard fault
2	-14	0x000C	NMI
1		0x0008	Reset
		0x0004	Initial SP value
		0x0000	

Nižší IRQ číslo - vyšší HW priorita

Cortex-M3 Exceptions-Interrupts: Registry jádra

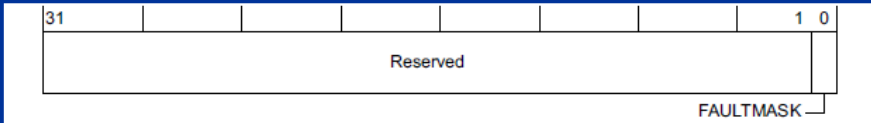


CPSID i, CPSIE i

__disable_irq(), __enable_irq()

CMSIS - core_cm3.h, core_cm3.c

PRIMASK = 1 Blokuje aktivaci všech výjimek (přerušení) s nastavitelnou prioritou

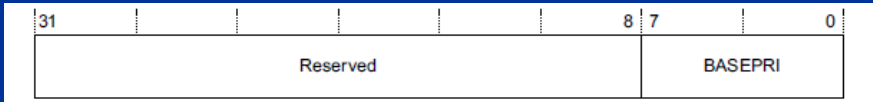


CPSID f, CPSIE f

__disable_fault_irq(), __enable_fault_irq()

CMSIS - core_cm3.h, core_cm3.c

FAULTMASK = 1 Blokuje aktivaci všech výjimek (přerušení) kromě NMI



BASEPRI (bity 7654) - nenulová hodnota blokuje aktivaci všech výjimek (přerušení) se stejnou nebo nižší prioritní úrovní (stejným nebo vyšším číslem), než je hodnota **BASEPRI**

STM32F10x Exceptions-Interrupts Priority

Handler	Field
Memory management fault	PRI_4
Bus fault	PRI_5
Usage fault	PRI_6
SVCall	PRI_11
PendSV	PRI_14
SysTick	PRI_15

SCB_SHPR1 registr

SCB_SHPR2 registr

SCB_SHPR3 registr

Nastavení priority:

4-bitové pole [7..4]
prioritních registrů

Rozmezí: 0 - 15

0 nejvyšší

15 nejnižší

	31	24	23	16	15	8	7	0								
IPR20	Reserved				Reserved				Reserved				IP [20]			
⋮																
IPRm	IP [m + 3]				IP [m + 2]				IP [m + 1]				IP [m]			
⋮																
IPR0	IP [3]				IP [2]				IP [1]				IP [0]			

NVIC_IPRx registry

x = 0 .. 16

CSMIS:

NVIC_IP[i] i = 0 .. 67

Nekonfiguruje-li SW priority, mají všechny konfigurovatelné priority hodnotu 0

STM32F10x Exceptions - Interrupts Priority

Preemptivní priorita

U nekonfigurovaných priorit mají všechny konfigurovatelné priority hodnotu 0. Při současném požadavku dvou IRQ_x a IRQ_y má přednost požadavek s nižším indexem - HW priorita - IRQ0 je obsloužen před IRQ1.

Má-li IRQ0 přiřazenu nižší prioritu (vyšší číselnou hodnotu) než IRQ1, má v případě současného požadavku přednost IRQ1.

Je-li procesor v přerušení, může být stávající **exception handler** přerušen požadavkem s vyšší nastavenou prioritou - nižším číslem v prioritním registru.

Přijde-li v obsluze přerušení požadavek na další přerušení se stejnou prioritou, stávající obsluha není přerušena a nový požadavek je akceptován až po návratu ze stávající obsluhy.

Nastavení rozložení priority

SCB_AIRCR Application Interrupt and Reset Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEYSTAT[15:0](read)/ VECTKEY[15:0](write)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANESS	Reserved				PRIGROUP			Reserved					SYS RESET REQ	VECT CLR ACTIVE	VECT RESET
r					rw	rw	rw						w	w	w

PRIGROUP [2:0]	Interrupt priority level value, PRI_M[7:4]			Number of	
	Binary point ⁽¹⁾	Group priority bits	Subpriority bits	Group priorities	Sub priorities
0b011	0bxxxx	[7:4]	None	16	None
0b100	0bxxx.y	[7:5]	[4]	8	2
0b101	0bxx.yy	[7:6]	[5:4]	4	4
0b110	0bx.yyy	[7]	[6:4]	2	8
0b111	0b.yyyy	None	[7:4]	None	16

Group priority - pouze tyto bity řídí preemptivní prioritu na základě výše popsané filozofie

Subpriority - při současném požadavku několika IRQ určuje pořadí jejich zpracování (nižší číslo - vyšší subpriorita)

V případě současného požadavku několika IRQx se stejnou Group a Subpriority rozhoduje HW priorita - nižší číslo x má přednost

NVIC - ostatní registry

Interrupts	CMSIS array elements ⁽¹⁾				
	Set-enable	Clear-enable	Set-pending	Clear-pending	Active Bit
0-31	ISER[0]	ICER[0]	ISPR[0]	ICPR[0]	IABR[0]
32-63	ISER[1]	ICER[1]	ISPR[1]	ICPR[1]	IABR[1]
64-67	ISER[2]	ICER[2]	ISPR[2]	ICPR[2]	IABR[2]

NVIC_ISERx Interrupt Set Enable Register

[illegible]

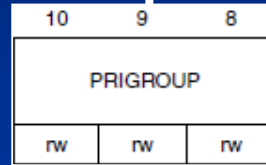
NVIC_ISERx Interrupt Clear Enable Register

[illegible]

CMSIS funkce pro práci s interrupty

`void NVIC_SetPriorityGrouping(uint32_t PriorityGroup) → SCB_AIRCR`

Hodnota 3 ..7



`void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority) → NVIC_IP[i]`

`void NVIC_EnableIRQ(IRQn_Type IRQn) → NVIC_ISERx`

`void NVIC_DisableIRQ(IRQn_Type IRQn) → NVIC_ICERx`