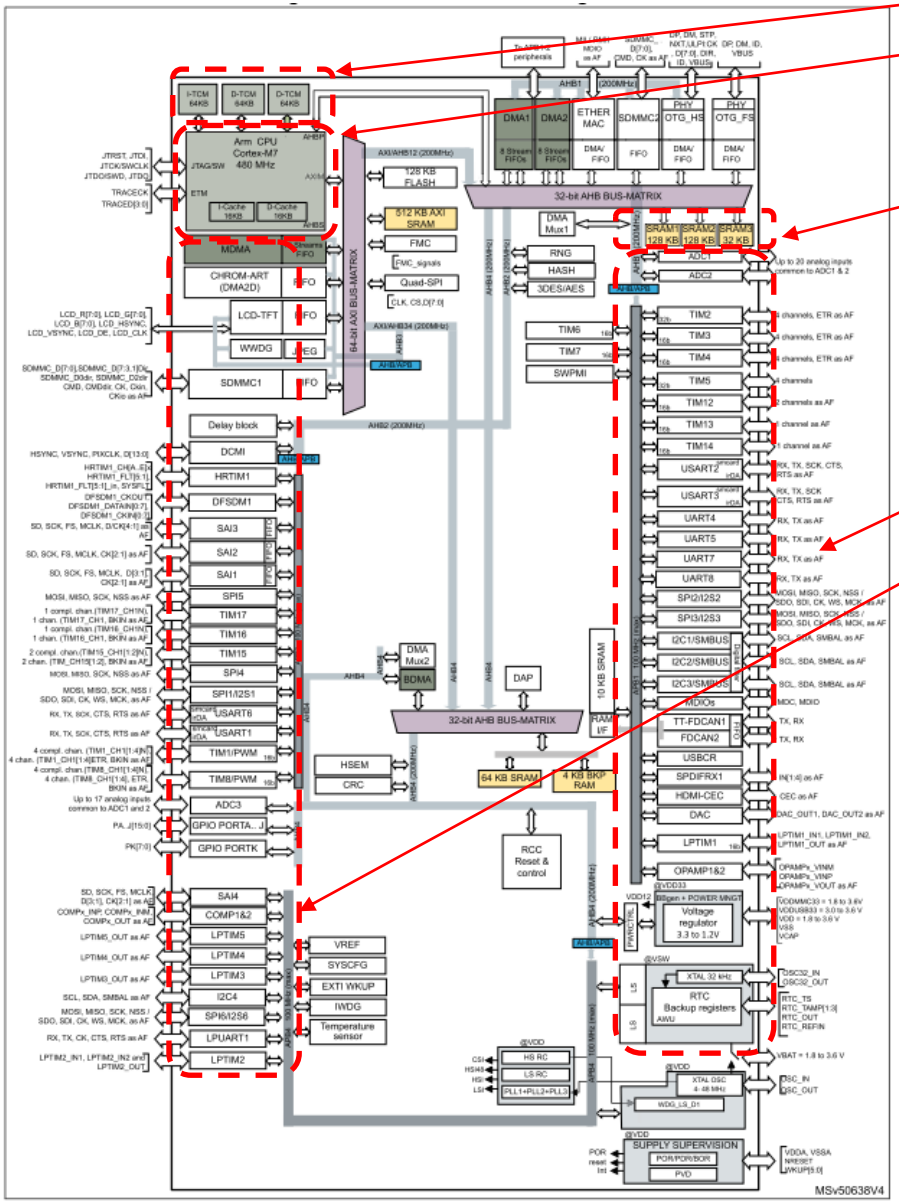
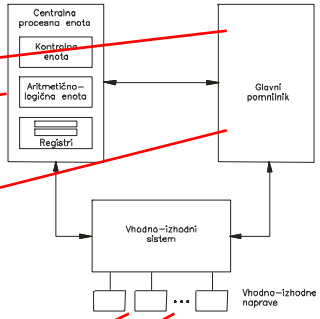
STM32H7

Vhodno / izhodne naprave

Prekinitve

STM32H750XB





Delo na STM32H7 razvojnem sistemu

Priključitev :

- **Mikro USB** priklop na **daljši stranici (nad LCD, srednji !!!)**

Poseben začetni projekt (github) in info za *STM32H7* (e-učilnica):

- ***dodajanje vsebine (Main.s):***

CubeIDEWorkspace - stm32h7-asm/Core/Src/Main.s - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- CubeIDE_Workspace
  - stm32f4-asm-qemu
- Delo
  - ARM9Template
  - stm32f4-asm (in STM32AsmTemplate)
  - ARM9Template.zip
- Node_V4 (in node_v4)
- Sluzba
  - CAN_IEX_Module
  - ORLab-STM32H7
    - stm32h7-asm
      - Binaries
      - Includes
      - Core
        - Src
          - Main.s
        - Startup
          - startup_stm32h750xbhx.s
      - Debug
      - out
      - makefile
      - README.md
      - STM32H750X.svd
      - STM32H750XBHX_FLASH.ld
      - STM32H750XBHX_RAM.ld
    - README.md
  - RALab-STM32H7
    - stm32h7-asm_RA_LED
    - README.md
  - STM32_USB_Key_AdvDebug
  - STM32_USB_Key_FreeRTOS_AdvDebug
  - STM32CubeIDE_Adv_Debug
  - STM32F4_Discovery_VIN_Projects

Main.s × startup_stm32h750xbhx.s

```
12
13 ////////////////////////////////////////////////////////////////////////////////
14 // Definitions
15 ////////////////////////////////////////////////////////////////////////////////
16 // Definitions section. Define all the registers and
17 // constants here for code readability.
18
19 // Constants
20
21
22 // Start of data section
23         .data
24
25         .align
26
27 STEV1:  .word   0x10    // 32-bitna spr.
28 STEV2:  .word   0x40    // 32-bitna spr.
29 VSOTA:  .word   0       // 32-bitna spr.
30
31
32 // Start of text section
33         .text
34
35         .type  main, %function
36         .global main
37
38         .align
39 main:
40         ldr r0, =STEV1   // Naslov od STEV1 -> r0
41         ldr r1, [r0]     // Vsebina iz naslova v r0 -> r1
42
43         ldr r0, =STEV2   // Naslov od STEV1 -> r0
44         ldr r2, [r0]     // Vsebina iz naslova v r0 -> r2
45
46         add r3,r1,r2     // r1 + r2 -> r3
47
48         ldr r0, =VSOTA   // Naslov od STEV1 -> r0
49         str r3,[r0]      // iz registra r3 -> na naslov v r0
50
51 __end:  b    __end
52
```

--------- Razvojni sistem STM32H750-DK ----------

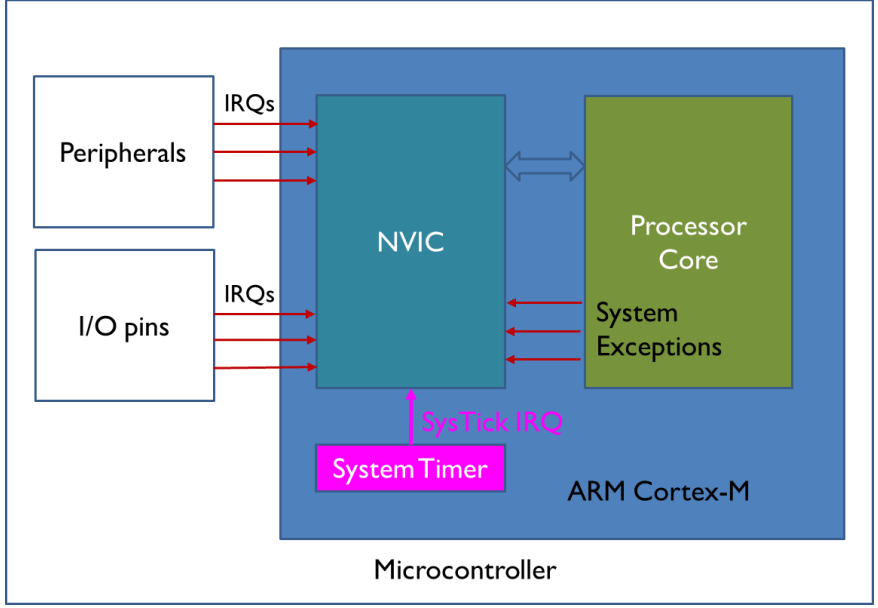
- STM32H750B-DK Discovery kit with STM32H750XB MCU
- ORLab-STM32H7 - GitHub repozitorij
- User Manual Discovery kit stm32h750xb Uploaded 11/11/22, 10.15
- DataSheet_stm32h750xb Uploaded 11/11/22, 10.16
- Reference Manual rm0433-stm32h750xb Uploaded 11/11/22, 10.17
- Programming_Manual_pm0253-stm32h750xb Uploaded 11/11/22, 10.17
- Errata_es0396-stm32h750xb Uploaded 11/11/22, 10.19

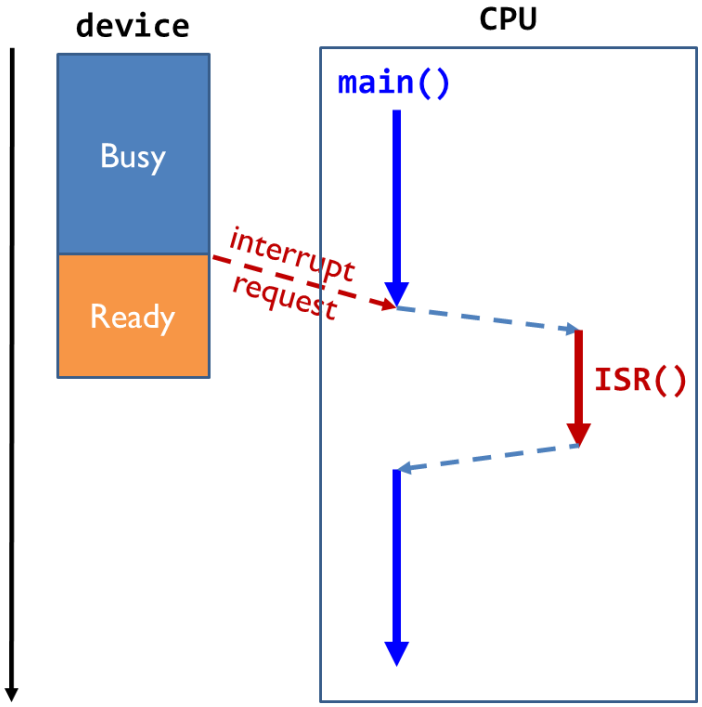
Prekinitve

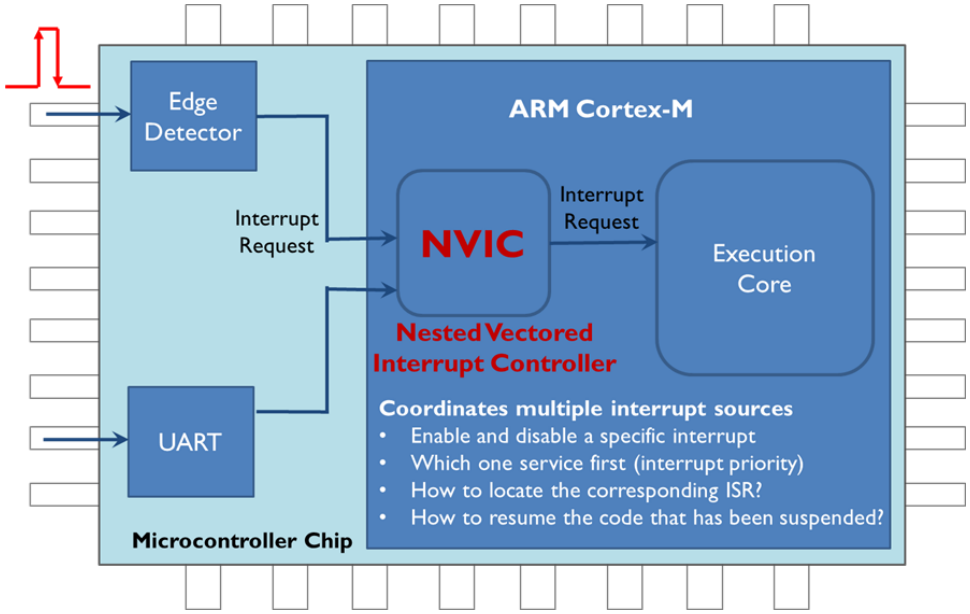
Optimizacija programske opreme za ustrezne reakcije na dogodke v realnem času:

- **dogodkovno vodene reakcije**
- **ciklične prekinitve s časovniki**

Prožijo se ustrezni Prekinitveno servisni podprogrami – PSP (ang. ISR)







Veliko virov (naprav) povezanik na krmilnik (NVIC), ki naprej sporoča zahteve na CPE

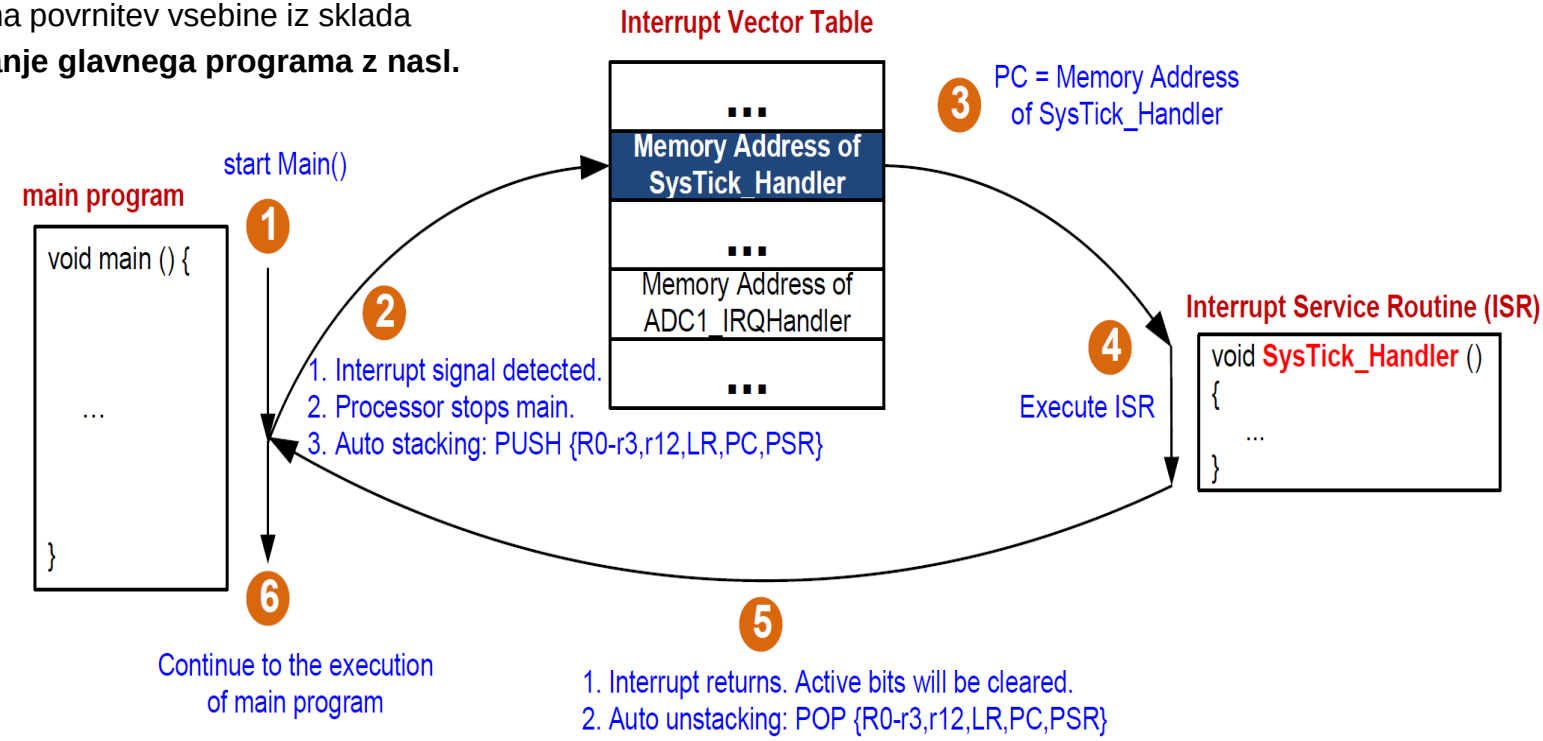
Prekinitve – izvedba, programiranje

Izvedba prekinitev:

1. **Glavni program se izvaja**
2. **Zazna se zahteva po prekinitvi:**
   - prekine se glavni program
   - delna shranitev registrov na sklad
3. **v PC se naloži naslov ustreznega PSPja**
4. **PSP se izvede**
5. **Vrnitev iz PSP :**
   - Brisanje oznak aktivnosti prekinitve
   - Delna povrnitev vsebine iz sklada
6. **Nadaljevanje glavnega programa z nasl. ukazom**

Uporaba prekinitev – splošen postopek:

- **Nastavitve NVIC krmilnika (vklop)**
- **Nastavitve V/I naprave (vklop)**
- **Pisanje PSP**
- **Vklop odziva na prekinitve**



Vira: Reference & Programming manuals

life.augmented

PM0253
Programming manual

STM32F7 Series and STM32H7 Series Cortex®-M7 processor programming manual

2.1.3 Core registers

The processor core registers are

Figure 2. Processor core registers

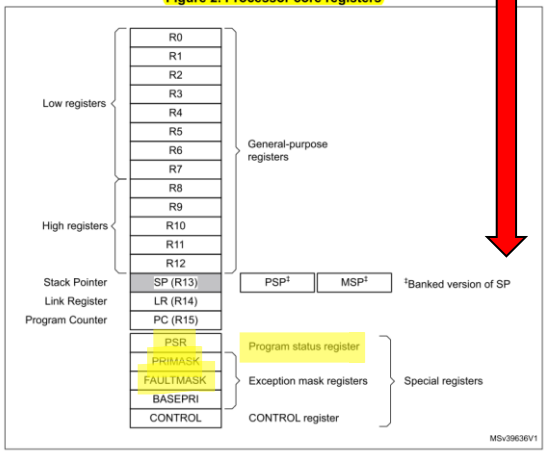


Table 2. Core register set summary

Register	Name	Type(1)	Required privilege(2)	Reset value	Description
General-purpose registers	R0-R12	RW	Either	Unknown	*General-purpose registers on page 21.*
Interrupt program status register	IPSR	RO	Privileged	0x00000000	*Interrupt program status register on page 23*
Execution program Status register	EPSR	RO	Privileged	0x01000000(3)	*Execution program status register on page 24*
Priority mask register	PRIMASK	RW	Privileged	0x00000000	*Priority mask register on page 25*
Fault mask register	FAULTMASK	RW	Privileged	0x00000000	*Fault mask register on page 26*
Base priority mask register	BASEPRIS	RW	Privileged	0x00000000	*Priority mask register on page 25*
Control register	CONTROL	RW	Privileged	0x00000000	*CONTROL register on page 27*

4.2 Nested Vectored Interrupt Controller

This section describes the NVIC and the registers it uses. The NVIC supports:

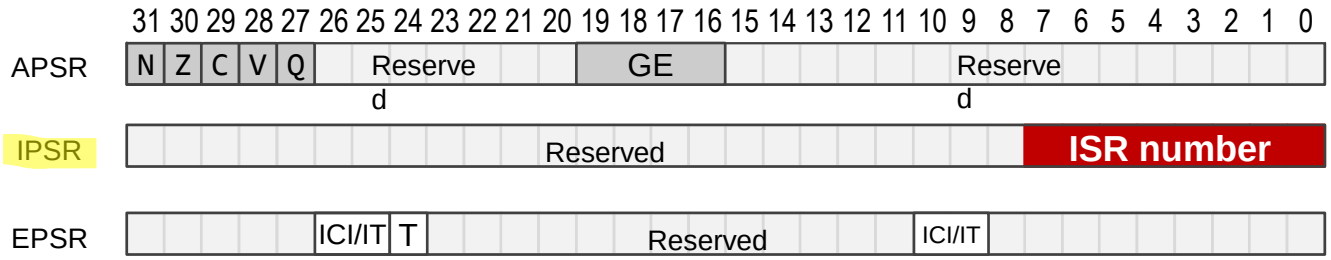
- 1 to 240 interrupts.
- A programmable priority level of 0-255 for each interrupt. A higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Level and pulse detection of interrupt signals.
- Dynamic reprioritization of interrupts.
- Grouping of priority values into group priority and subpriority fields.
- Interrupt tail-chaining.
- An external *Non Maskable Interrupt* (NMI)

Table 40. NVIC register summary

Address	Name	Type	Required privilege	Reset value	Description
0xE000E100-0xE000E11C	NVIC_ISER0-NVIC_ISER7	RW	Privileged	0x00000000	*Interrupt set-enable registers on page 185*
0XE000E180-0xE000E19C	NVIC_ICER0-NVIC_ICER7	RW	Privileged	0x00000000	*Interrupt clear-enable registers on page 186*
0XE000E200-0xE000E21C	NVIC_ISPR0-NVIC_ISPR7	RW	Privileged	0x00000000	*Interrupt set-pending registers on page 186*
0XE000E280-0xE000E29C	NVIC_ICPR0-NVIC_ICPR7	RW	Privileged	0x00000000	*Interrupt clear-pending registers on page 187*
0xE000E300-0xE000E31C	NVIC_IABR0-NVIC_IABR7	RW	Privileged	0x00000000	*Interrupt active bit registers on page 188*
0xE000E400-0xE000E4EF	NVIC_IPR0-NVIC_IPR59	RW	Privileged	0x00000000	*Interrupt priority registers on page 188*
0xE000EF00	STIR	WO	Configurable(1)	0x00000000	*Software trigger interrupt register on page 189*

Program Status Register - PSR

3 statusni registri : Application PSR (APSR), Interrupt PSR (IPSR), Execution PSR (EPSR)



so združeni v statusnem registru (PSR):

8-bit interrupt number in **PSR**
Range: **`0 - 255`**

Prekinitve – IPSR Statusni register

Interrupt program status register

The IPSR contains the exception type number of the current *Interrupt Service Routine* (ISR). See the register summary in *Table 5 on page 24* for its attributes. The bit assignments are:

Table 5. IPSR bit assignments

Bits	Name	Function
[31:9]	-	Reserved
[8:0]	ISR_NUMBER	This is the number of the current exception: 0 = Thread mode. 1 = Reserved. 2 = NMI. 3 = HardFault. 4 = MemManage. 5 = BusFault 6 = UsageFault 7-10 = Reserved 11 = SVCall. 12 = Reserved for debug 13 = Reserved 14 = PendSV. 15 = SysTick. 16 = IRQ0. . . 256 = IRQ239. see *Exception types on page 39* for more information.

Maskiranje prekinitve – PRIMASK register

Priority mask register

The PRIMASK register prevents the activation of all exceptions with a configurable priority.
See the register summary in *Table 7* for its attributes. The bit assignments are

Figure 4. PRIMASK bit assignments:

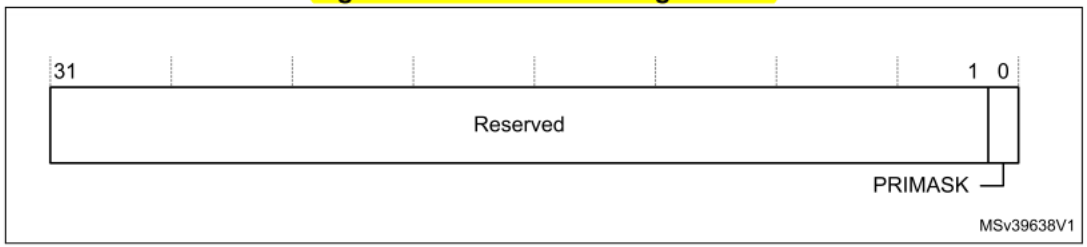


Table 7. PRIMASK register bit assignments

Bits	Name	Function
[31:1]	-	Reserved.
[0]	PRIMASK	Prioritizable interrupt mask: 0 = No effect. 1 = Prevents the activation of all exceptions with configurable priority.

Maskiranje prekinitev – FAULTMASK register

Fault mask register

The FAULTMASK register prevents activation of all exceptions except for *Non Maskable Interrupt* (NMI). See the register summary in *Table 8 on page 26* for its attributes. The bit assignments are

Figure 5. FAULTMASK bit assignments

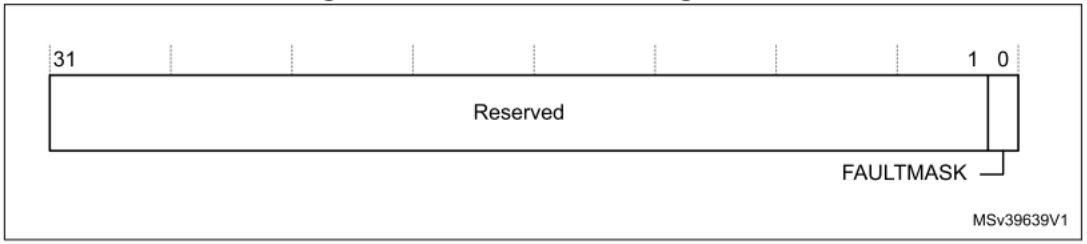


Table 8. FAULTMASK register bit assignments

Bits	Name	Function
[31:1]	-	Reserved.
[0]	FAULTMASK	Prioritizable interrupt mask: 0 = No effect. 1 = Prevents the activation of all exceptions except for NMI.

The processor clears the FAULTMASK bit to 0 on exit from any exception handler except the NMI handler.

Prekinitve – ICSR register

4.3.3 Interrupt control and state register

The ICSR:

- provides:
  - A set-pending bit for the *Non Maskable Interrupt* (NMI) exception.
  - Set-pending and clear-pending bits for the PendSV and SysTick exceptions.
- indicates:
  - The exception number of the exception being processed.
  - Whether there are preempted active exceptions.
  - The exception number of the highest priority pending exception
  - Whether any interrupts are pending.

Table 53. ICSR bit assignments

Bits	Name	Type	Function
[31]	NMIPENDSET	RW	NMI set-pending bit. Write: 0: No effect. 1: Changes NMI exception state to pending. Read: 0: NMI exception is not pending. 1: NMI exception is pending. Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it registers a write of 1 to this bit, and entering the handler clears this bit to 0. A read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.
[30:29]	-	-	Reserved.
[28]	PENDSVSET	RW	PendSV set-pending bit. Write: 0: No effect. 1: Changes PendSV exception state to pending. Read: 0: PendSV exception is not pending. 1: PendSV exception is pending. Writing 1 to this bit is the only way to set the PendSV exception state to pending.
[27]	PENDSVCLR	WO	PendSV clear-pending bit. Write: 0: No effect. 1: Removes the pending state from the PendSV exception.
[26]	PENDSTSET	RW	SysTick exception set-pending bit. Write: 0: No effect. 1: Changes SysTick exception state to pending. Read: 0: SysTick exception is not pending. 1: SysTick exception is pending.

4.3 System control block

The *System Control Block* (SCB) provides system implementation information, and system control. This includes configuration, control, and reporting of the system exceptions. The system control block registers are:

Table 50. Summary of the system control block registers

Address	Name	Type	Required privilege	Reset value	Description
0xE000E008	ACTLR	RW	Privileged	0x00000000	*Auxiliary control register on page 193*
0xE000ED00	CPUID	RO	Privileged	0x410FC270	*CPUID base register on page 194*
0xE000ED04	ICSR	RW(1)	Privileged	0x00000000	*Interrupt control and state register on page 194*
0xE000ED08	VTOR	RW	Privileged	Unknown	*Vector table offset register on page 197*

Figure 26. ICSR bit assignments

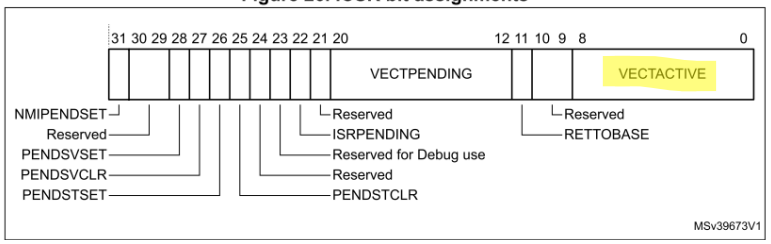


Table 53. ICSR bit assignments (continued)

Bits	Name	Type	Function
[25]	PENDSTCLR	WO	SysTick exception clear-pending bit. Write: 0: No effect. 1: Removes the pending state from the SysTick exception. This bit is WO. On a register read its value is `Unknown`.
[24]	-	-	Reserved.
[23]	Reserved for Debug use	RO	This bit is reserved for Debug use and reads-as-zero when the processor is not in Debug.
[22]	ISRPENDING	RO	Interrupt pending flag, excluding NMI and Faults: 0: Interrupt not pending. 1: Interrupt pending.
[21]	-	-	Reserved.
[20:12]	VECTPENDING	RO	Indicates the exception number of the highest priority pending enabled exception: 0: No pending exceptions. Nonzero: The exception number of the highest priority pending enabled exception. The value indicated by this field includes the effect of the BASEPRI and FAULTMASK registers, but not any effect of the PRIMASK register.
[11]	RETTOBASE	RO	Indicates whether there are preempted active exceptions: 0: There are preempted active exceptions to execute. 1: There are no active exceptions, or the currently-executing exception is the only active exception.
[10:9]	-	-	Reserved.
[8:0]	VECTACTIVE(1)	RO	Contains the active exception number: 0: Thread mode 1: The exception number(1) of the currently active exception. Subtract 16 from this value to obtain the CMSIS IRQ number required to index into the Interrupt Clear-Enable, Set-Enable, Clear-Pending, Set-Pending, or Priority Registers, see *Table 5 on page 24*.

1. This is the same value as IPSR bits[8:0], see *Interrupt program status register on page 23*.

NVIC – Prekinitveni krmilnik

Table 42. ISER bit assignments

Bits	Name	Function
[31:0]	SETENA	Interrupt set-enable bits. Write: 0: No effect. 1: Enable interrupt. Read: 0: Interrupt disabled. 1: Interrupt enabled.

Table 43. ICER bit assignments

Bits	Name	Function
[31:0]	CLRENA	Interrupt clear-enable bits. Write: 0: No effect. 1: Disable interrupt. Read: 0: Interrupt disabled. 1: Interrupt enabled.

Table 45. ICPR bit assignments

Bits	Name	Function
[31:0]	CLRPEND	Interrupt clear-pending bits. Write: 0: No effect. 1: Removes pending state an interrupt. Read: 0: Interrupt is not pending. 1: Interrupt is pending.

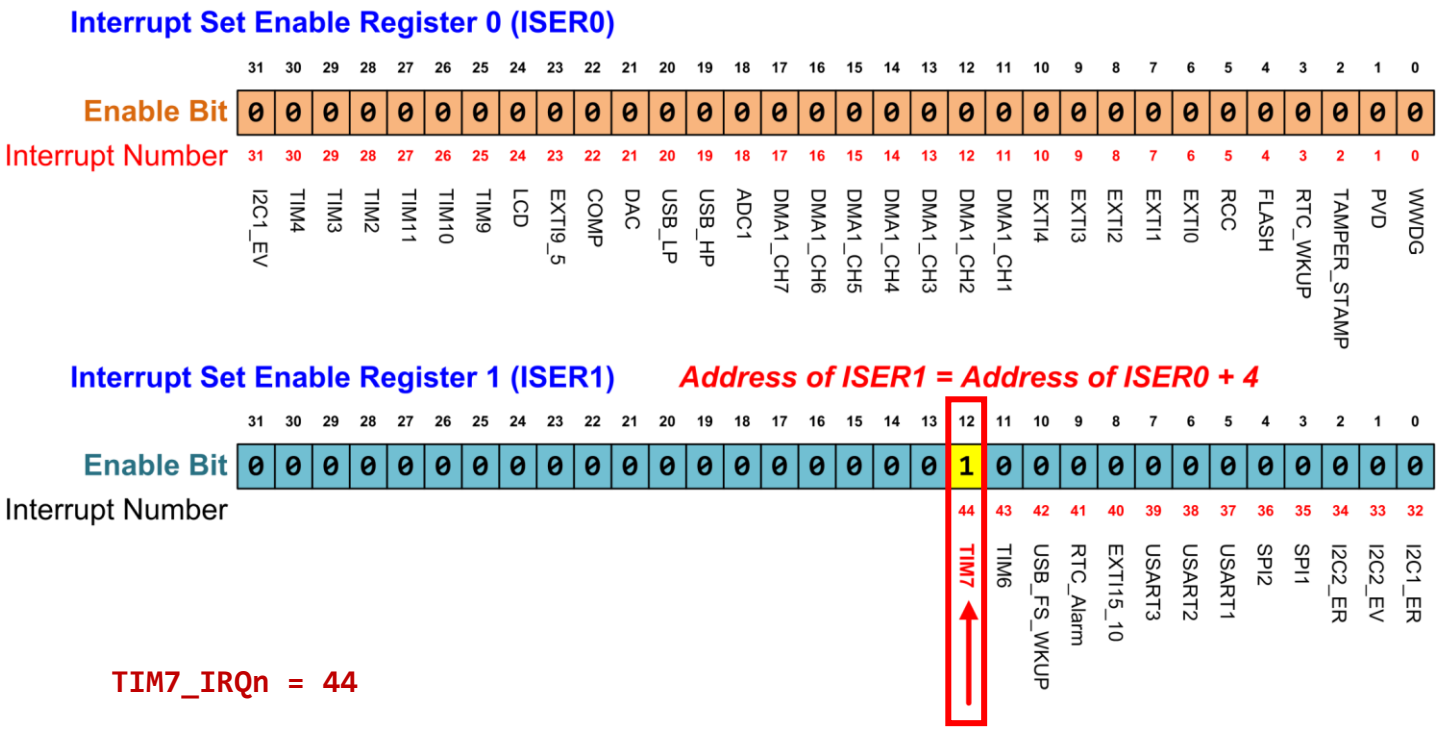
Table 46. IABR bit assignments

Bits	Name	Function
[31:0]	ACTIVE	Interrupt active flags: 0: Interrupt not active. 1: Interrupt active.

Table 40. NVIC register summary

Address	Name	Type	Required privilege	Reset value	Description
0xE000E100-0xE000E11C	NVIC_ISER0-NVIC_ISER7	RW	Privileged	0x00000000	*Interrupt set-enable registers on page 185*
0XE000E180-0xE000E19C	NVIC_ICER0-NVIC_ICER7	RW	Privileged	0x00000000	*Interrupt clear-enable registers on page 186*
0XE000E200-0xE000E21C	NVIC_ISPR0-NVIC_ISPR7	RW	Privileged	0x00000000	*Interrupt set-pending registers on page 186*
0XE000E280-0xE000E29C	NVIC_ICPR0-NVIC_ICPR7	RW	Privileged	0x00000000	*Interrupt clear-pending registers on page 187*
0xE000E300-0xE000E31C	NVIC_IABR0-NVIC_IABR7	RW	Privileged	0x00000000	*Interrupt active bit registers on page 188*
0xE000E400-0xE000E4EF	NVIC_IPR0-NVIC_IPR59	RW	Privileged	0x00000000	*Interrupt priority registers on page 188*
0xE000EF00	STIR	WO	Configurable(1)	0x00000000	*Software trigger interrupt register on page 189*

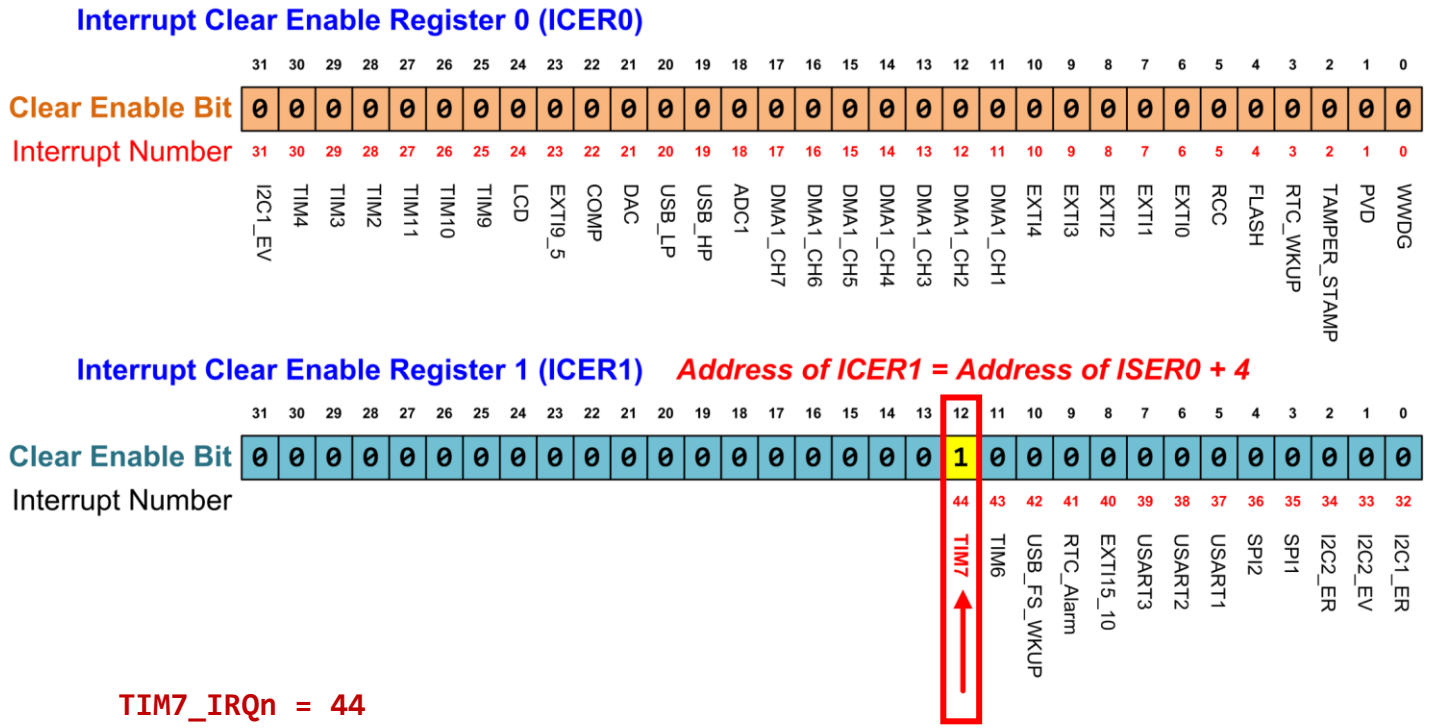
NVIC –ISER: Registri za omogočanje prekinitev




```
TIM7_IRQn = 44

NVIC->ISER[1] = 1 << 12;     // Enable Timer 7 interrupt
```

NVIC – ICER: Registri za onemogočanje prekinitev

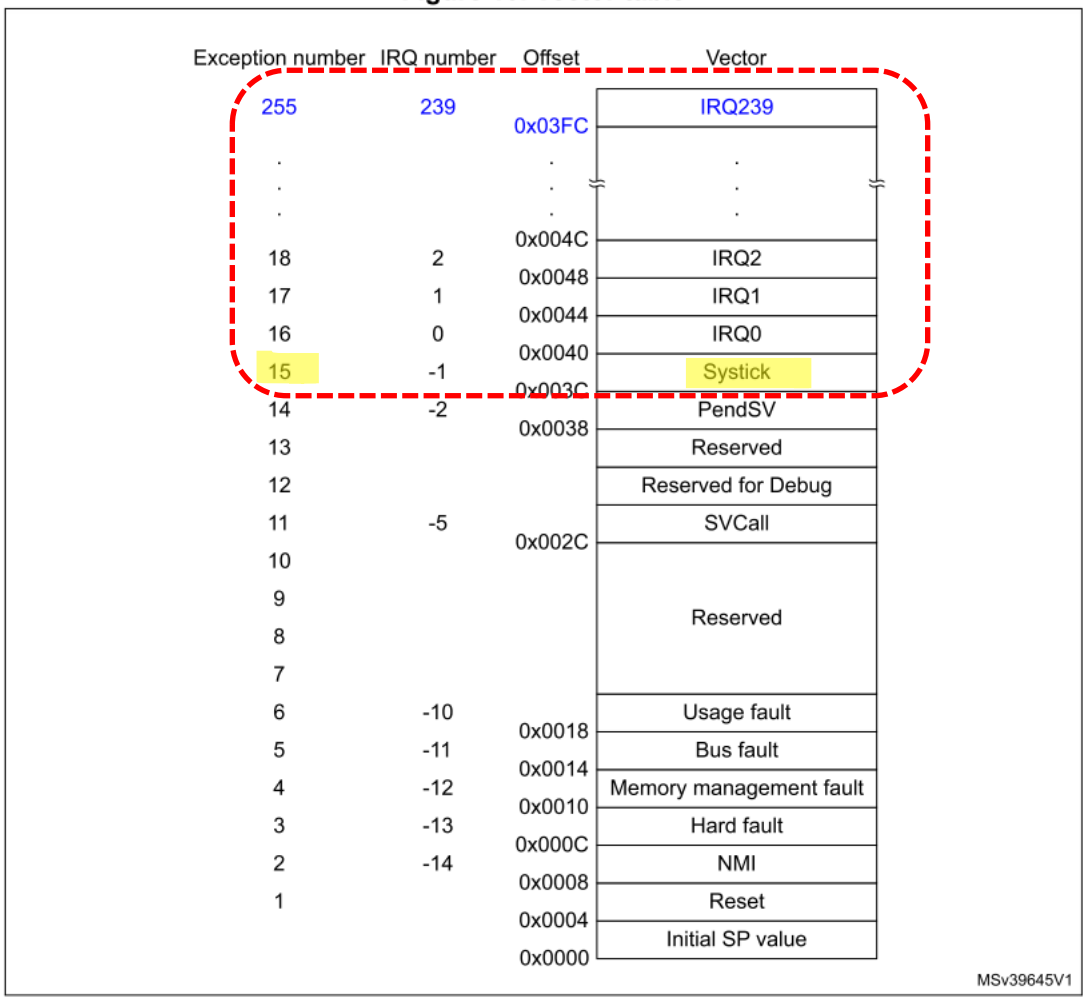



```
TIM7_IRQn = 44

NVIC->ICER[1] = 1 << 12;      // Diable Timer 7 interrupt
```

Prekinitve - vektorska tabela

Figure 10. Vector table



Prekinitve – ukaz CPS

3.12.2 CPS

Change Processor State.

Syntax

`CPS`*`effect iflags`*

Where:

effect Is one of:

- `IE` Clears the special purpose register.
- `ID` Sets the special purpose register.

iflags Is a sequence of one or more flags:

- `i` Set or clear PRIMASK.
- `f` Set or clear FAULTMASK.

Examples

```
CPSID i  ; Disable interrupts and configurable fault handlers (set
         ; PRIMASK)
CPSID f  ; Disable interrupts and all fault handlers (set FAULTMASK)
CPSIE i  ; Enable interrupts and configurable fault handlers (clear
         ; PRIMASK)
CPSIE f  ; Enable interrupts and fault handlers (clear FAULTMASK)
```

Vira: Reference & Programming manuals

life.augmented

RM0433

Reference manual

STM32H742, STM32H743/753 and STM32H750 Value line advanced Arm®-based 32-bit MCUs

19 Nested vectored interrupt controller (NVIC)

19.1 NVIC features

The NVIC includes the following features:

- up to 150 maskable interrupt channels for STM32H7xxx (not including the 16 interrupt lines of Cortex®-M7 with FPU)
- 16 programmable priority levels (4 bits of interrupt priority are used)
- low-latency exception and interrupt handling
- power management control
- implementation of system control registers

19.1.2 Interrupt and exception vectors

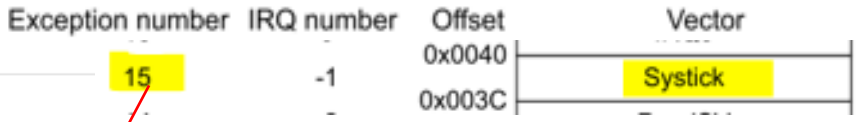
The exception vectors connected to the NVIC are the following: reset, NMI, HardFault, MemManage, Bus Fault, UsageFault, SVCall, DebugMonitor, PendSV, SysTick.

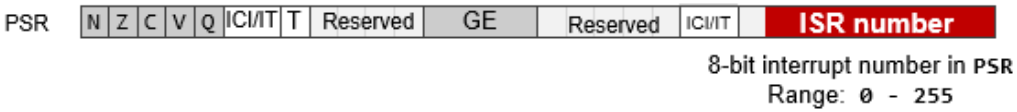
Table 143. NVIC(1)

Signal	Priority	NVIC position	Acronym	Description	Address offset
-	-	-	-	Reserved	0x0000 0000
-	-3	-	Reset	Reset	0x0000 0004
-	-2	-	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000 0008
-	-1	-	HardFault	All classes of fault	0x0000 000C
-	0	-	MemManage	Memory management	0x0000 0010
-	1	-	BusFault	Prefetch fault, memory access fault	0x0000 0014
-	2	-	UsageFault	Undefined instruction or illegal state	0x0000 0018
-	-	-	-	Reserved	0x0000 001C-0x0000 002B
-	3	-	SVCall	System service call via SWI instruction	0x0000 002C
-	4	-	DebugMonitor	Debug monitor	0x0000 0030
-	-	-	-	Reserved	0x0000 0034
-	5	-	PendSV	Pendable request for system service	0x0000 0038
-	6	-	SysTick	System tick timer	0x0000 003C
wwdg1_it	7	0	WWDG1	Window Watchdog interrupt	0x0000 0040

CubeIDE – Registers okno

Name	Value
General Registers	
r0	603979776 (Decimal)
r1	0xe000ed08
r2	0x40000001
r3	0x0
r4	0x2000002c
r5	0x0
r6	0x0
r7	0x0
r8	0x0
r9	0x0
r10	0x0
r11	0x0
r12	0x0
sp	0x2001ffcc
lr	0xfffffff9
pc	0x24000386
xpsr	0x6100000f





CubeIDE – SFR okno

Register	Address	Value
Cortex_M7		
Cache		
Control		
ACTLR	0xe000e008	0x1000
ICSR	0xe000ed04	0x80f
NMIPENDSET	[31:1]	0x0
PENDSVSET	[28:1]	0x0
PENDSVCLR	[27:1]	0x0
PENDSTSET	[26:1]	0x0
PENDSTCLR	[25:1]	0x0
ISRPREEMPT	[23:1]	0x0
ISRPENDING	[22:1]	0x0
VECTPENDING	[12:9]	0x0
RETTOBASE	[11:1]	0x1
VECTACTIVE	[0:9]	0xf