

ARM

Projekt za STM32H7 vgrajen sistem

CubeIDE, VSCode

STM32H750B-DK Discovery razvojni sistem

- Arm® Cortex® core-based microcontroller with 128 Kbytes (STM32H750XBH6) of Flash memory and 1 Mbyte of RAM, in TFBGA240+25 package

- 4.3" RGB interface LCD with touch panel connector

- Ethernet compliant with IEEE-802.3-2002, and POE

- USB OTG FS with Micro-AB connector

- SAI audio codec

- One ST-MEMS digital microphone

- 2 x 512-Mbit Quad-SPI NOR Flash memory

- 128-Mbit SDRAM

- 4-Gbyte on-board eMMC

- 1 user and reset push-button

- Fanout daughterboard

- 2 x FDCANs

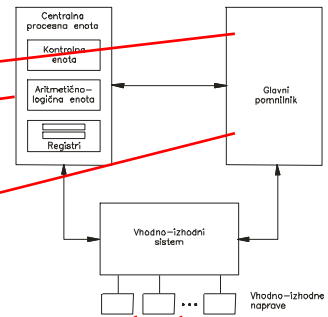
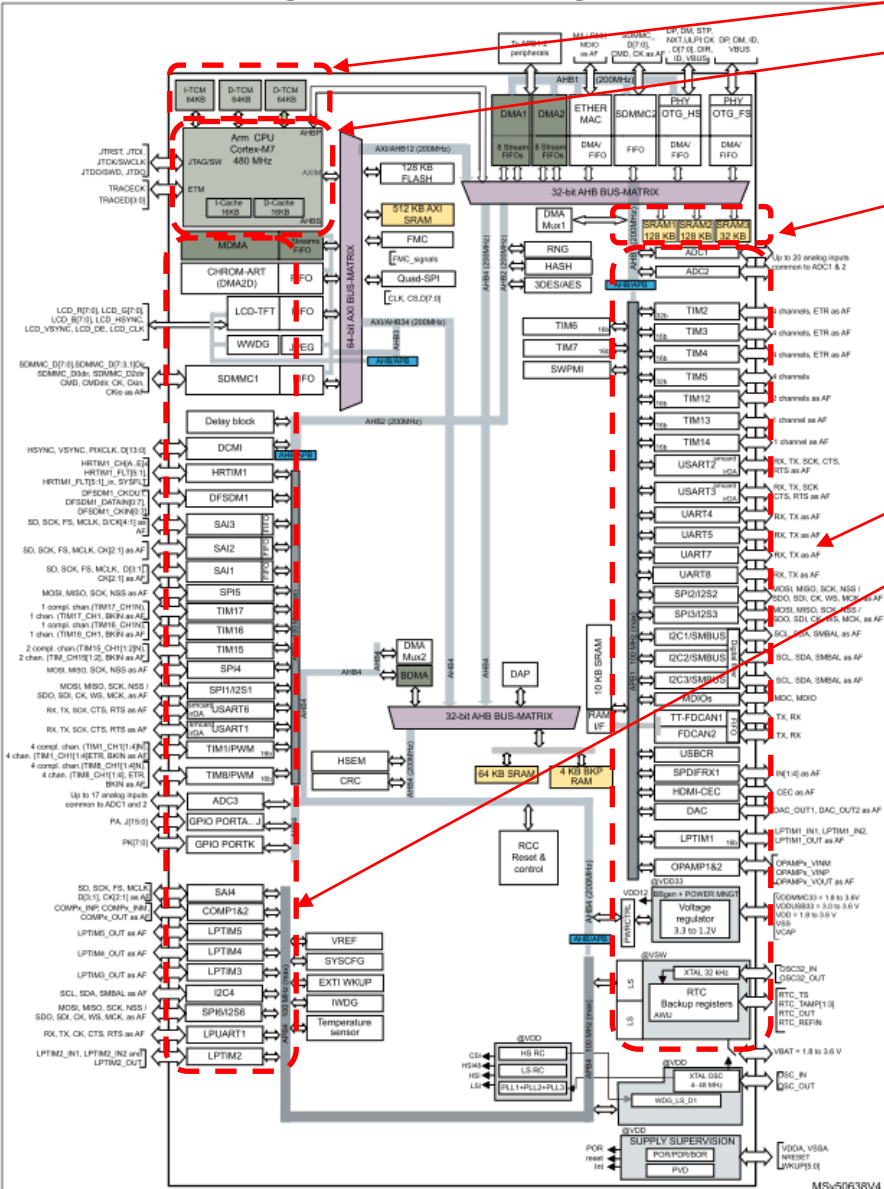
- Board connectors:

- USB FS Micro-AB connectors
- ST-LINK Micro-B USB connector
- USB power Micro-B connector
- Ethernet RJ45
- Stereo headset jack including analog microphone input
- Audio header for external speakers
- Arduino™ Uno V3 expansion connectors
- STMod+



<https://www.st.com/en/evaluation-tools/stm32h750b-dk.html>

STM32H750XB



Delo na STM32H7 razvojnem sistemu

Priključitev :

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

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

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



```
IDE CubelDEWorkspace - stm32h7-asm/Core/Src/Main.s - STM32CubelDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer x
CubelDE_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
      STM32CubelDE_Adv_Debug
      STM32F4_Discovery_VIN_Projects
Main.s x 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

Delo na STM32H7 razvojnem sistemu

Priključitev :

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

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

- **začetne nastavitve** ([startup_stm32f407vgtx.s](#)) :
 - Pustimo default nastavitve:
 - 64MHz frekvenca urinega signala
 - (višja poveča porabo!)
 - izklop predpomnilnikov
 - **inicializacija sklada** oz. SP – kazalca na sklad
- **dodajanje vsebine** ([Main.s](#)):
 - podatki/operandi:
 - dodamo v .data sekcijo, končamo z .align
 - program (dodamo v .text sekcijo) :
 - **dodamo** od oznake **main:** naprej
 - na koncu programa je **mrtva zanka** (**__end: b __end**)
 - **podprograme** dodamo za mrtvo zanko



Inicializacija sistema – začetno stanje

startup_stm32f407vgtx.s :

g_pfnVectors:

```
.word _estack
.word Reset_Handler
.word NMI_Handler
.word HardFault_Handler
.word MemManage_Handler
.word BusFault_Handler
.word UsageFault_Handler
.word 0
.word 0
.word 0
.word 0
.word SVC_Handler
.word DebugMon_Handler
.word 0
.word PendSV_Handler
.word SysTick_Handler
```

ARM Cortex M – Vektorska tabela

Vector Table	Vector address (initial)
Interrupt#239 vector	0x000003FC
Interrupt#31 vector	0x000000BC
Interrupt#1 vector	0x00000044
Interrupt#0 vector	0x00000040
SysTick vector	0x0000003C
PendSV vector	0x00000038
Not used	0x00000034
Debug Monitor vector	0x00000030
SVC vector	0x0000002C
Not used	0x00000028
Not used	0x00000024
Not used	0x00000020
SecureFault (ARMv8-M Mainline)	0x0000001C
Usage Fault vector	0x00000018
Bus Fault vector	0x00000014
MemManage vector	0x00000010
HardFault vector	0x0000000C
NMI vector	0x00000008
Reset vector	0x00000004
MSP initial value	0x00000000

Inicializacija sistema – začetno stanje

startup_stm32f407vgtx.s :

Reset_Handler:

```
ldr    sp, =_estack    /* set stack pointer */
```

```
/* Copy the data segment initializers from flash to SRAM */
```

CopyDataInit: ...

FillZerobss: ...

```
// Initialize DWT counters - added for cycle measurements
```

```
...
```

```
/* Call the application's entry point.*/
```

```
bl    main  
bx    lr
```

Glavni program, →

podprogrami →

Main.s :

main:

```
b1 INIT_IO      // Priprava za kontrolo LED
diode
```

loop:

```
b1 LED_ON      // Vklop LED diode
```

```
mov r0,#500
```

```
b1 DELAY // Zakasnitev: r0 x 1msec
```

```
b1 LED_OFF     // Izlop LED diode
```

```
mov r0,#500
```

```
b1 DELAY // Zakasnitev: r0 x 1msec
```

```
b loop         // skok na vrstico loop:
```

```
__end: b __end
```


spremenljivke.



Main.s :

```
// Start of data section
```

```
.data
```

```
.align
```

```
STEV1: .word 0x10 // 32-bitna spr.
```

```
STEV2: .word 0x40 // 32-bitna spr.
```

```
VSOTA: .word 0 // 32-bitna spr.
```

```
LEDSTAT: .word 0 // LED Status
```