

ARM

Projekt za STM32H7 vgrajen sistem II. del

CubeIDE

STM32H750B-DK Discovery razvojni sistem

I. del

- 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:

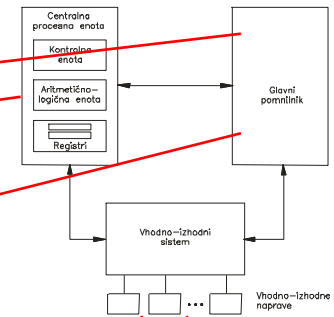
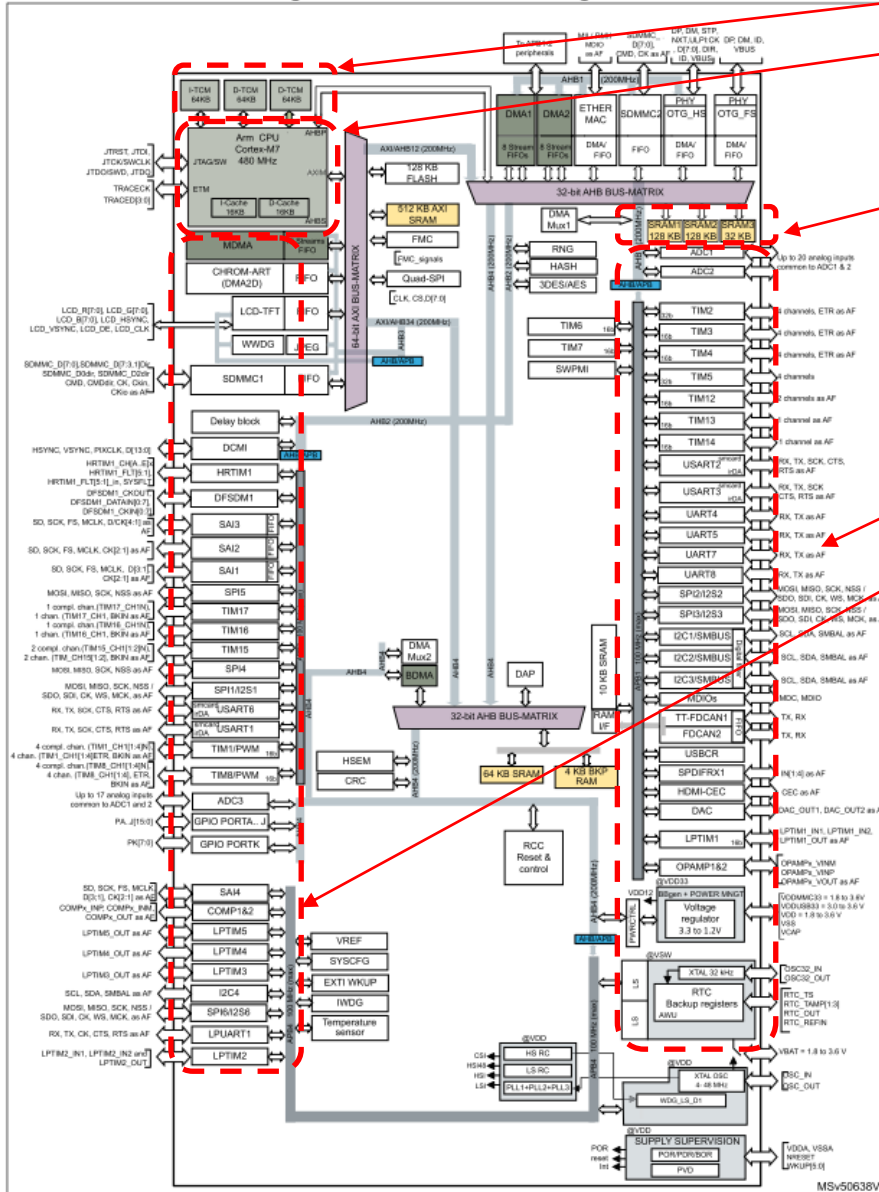
- USB FS Micro-AB connectors
- ST-LINK Micro-B USB connector
- USB power Micro-B connector
- Ethernet RJ45
- Stereo headset jack including analog microphone input
- Audio header for external speakers
- Arduino™ Uno V3 expansion connectors
- STMod+



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

STM32H750XB

I. del



Delo na STM32H7 razvojnem sistemu

I. del

Priključitev :

- **Mikro USB** prikllop 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

I. del

Priključitev :

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

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

- **začetne nastavitve** ([startup_stm32h750xbhx.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_stm32h750xbhx.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_stm32h750xbhx.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
```

Povezovalna skripta („linker script“)

Introduction to the STM32 Memory Model

The memory of STM32 microcontrollers has two main areas :

1. Flash Memory :

- Read-Only
- Begins at address 0x08000000
- Size depends on specific STM32 microcontroller
- Program code is stored here.
- Contains a vector table at address 0x08000004

```
MEMORY
```

```
{
```

```
FLASH(rx):ORIGIN =0x08000000,LENGTH =512K
```

```
SRAM(rwx):ORIGIN =0x20000000,LENGTH =128K
```

```
}
```

2. SRAM Memory :

- Read and Write
- Begins at 0x20000000
- Size depends on specific STM32 microcontroller
- Variables and Stack are stored here

```
SECTIONS
```

```
{
```

```
    .text :
```

```
    {
```

```
        *(.text)    /*merge all .text sections of input files */
```

```
    }> FLASH
```

```
}
```

<https://go.embeddedexpert.io/article1635692200758>

Povezovalna skripta („linker script“) - CubeIDE

Linker skripta določí razpored po pomnilnikih

Introduction to STM32 Memory Model and Linker Scripts

Introduction to the STM32 Memory Model

The memory of STM32 microcontrollers has two main areas :

1. Flash Memory :

- Read-Only
- Begins at address 0x08000000
- Size depends on specific STM32 microcontroller
- Program code is stored here.

- Contains a vector table at address 0x08000004

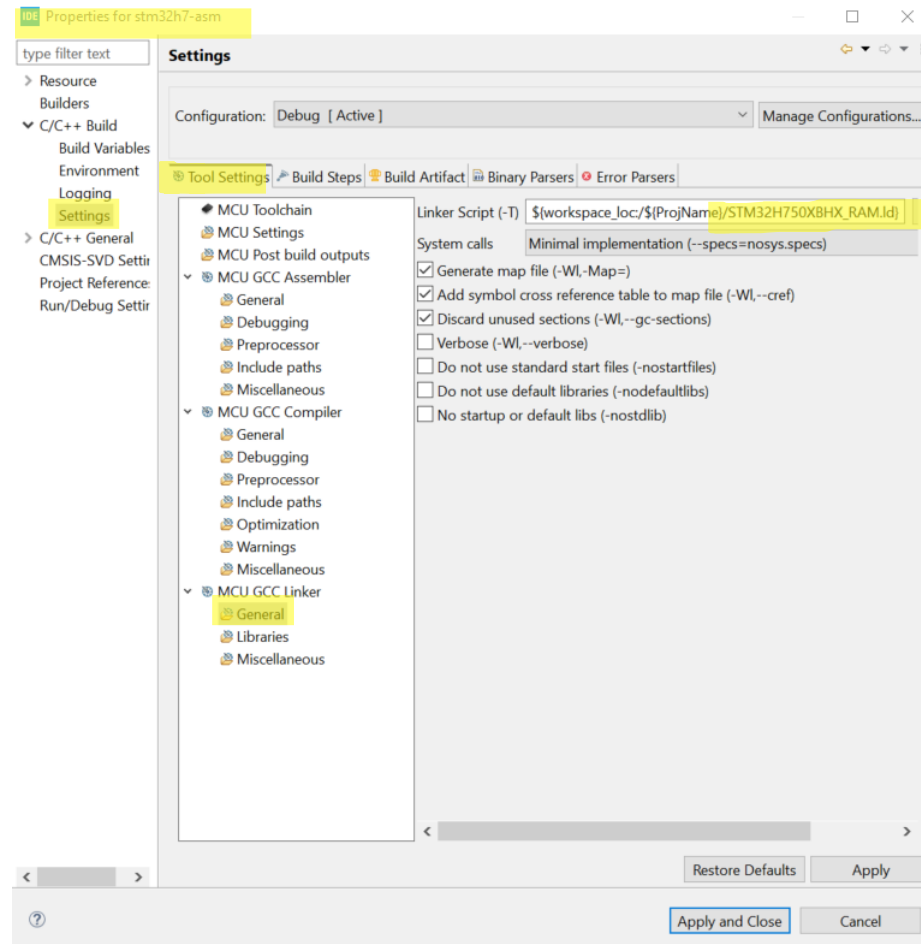
2. SRAM Memory :

- Read and Write
- Begins at 0x20000000
- Size depends on specific STM32 microcontroller
- Variables and Stack are stored here

The 4 features of a linker script :

- Memory Layout** : Available memory types
- Section Definitions** : Placing specific parts at specific locations
- Options** : Commands e.g. ENTRY POINT
- Symbols** : Variables to inject into program at link time

<https://go.embeddedexpert.io/article1635692200758>>



Povezovalna skripta („linker script“) - CubeIDE

STM32H750XBHX FLASH.Id :

```
/* Specify the memory areas */
MEMORY
{
  FLASH (rx)      : ORIGIN = 0x08000000, LENGTH = 128K
  DTCMRAM (xrw)   : ORIGIN = 0x20000000, LENGTH = 128K
  RAM_D1 (xrw)    : ORIGIN = 0x24000000, LENGTH = 512K
  RAM_D2 (xrw)    : ORIGIN = 0x30000000, LENGTH = 288K
  RAM_D3 (xrw)    : ORIGIN = 0x38000000, LENGTH = 64K
  ITCMRAM (xrw)   : ORIGIN = 0x00000000, LENGTH = 64K
}
...
SECTIONS
{
  /* The startup code goes first into FLASH */
  .isr_vector :
  {
    ...
  } >FLASH
  /* The program code and other data goes into FLASH */
  .text :
  {
    ...
  } >FLASH

  /* Initialized data sections goes into RAM, load LMA copy*/
  .data :
  {
    ...
  } >RAM_D1 AT> FLASH
}

```

STM32H750XBHX RAM.Id :

```
/* Specify the memory areas */
MEMORY
{
  RAM_EXEC (xrw)  : ORIGIN = 0x24000000, LENGTH = 128K
  DTCMRAM (xrw)   : ORIGIN = 0x20000000, LENGTH = 128K
  RAM_D2 (xrw)    : ORIGIN = 0x30000000, LENGTH = 288K
  RAM_D3 (xrw)    : ORIGIN = 0x38000000, LENGTH = 64K
  ITCMRAM (xrw)   : ORIGIN = 0x00000000, LENGTH = 64K
}
...
SECTIONS
{
  /* The startup code goes first into FLASH */
  .isr_vector :
  {
    ...
  } >RAM_EXEC
  /* The program code and other data goes into FLASH */
  .text :
  {
    ...
  } >RAM_EXEC

  /* Initialized data sections goes into RAM, load LMA copy*/
  .data :
  {
    ...
  } >DTCMRAM AT> RAM_EXEC
}

```

CubeIDE – Izvedba programa - RAM

STM32H750XBHX RAM.Id:

```
/* Define output sections */
SECTIONS
{
/* The startup code goes first into RAM_EXEC (D1) */
.isr_vector :
{
...
} >RAM_EXEC

/* The program code and other data goes into RAM_EXEC(D1)*/
.text :
{
...
} >RAM_EXEC

/* Initialized data sections go into DTCMRAM from RAM_EXEC*/
.data :
{
...
} >DTCMRAM AT> RAM_EXEC

/* Uninitialized data section goes into DTCMRAM */
.bss :
{
...
} >DTCMRAM

MEMORY /* Specify the memory areas */
{
RAM_EXEC (xrw) : ORIGIN = 0x24000000, LENGTH = 128K
DTCMRAM (xrw) : ORIGIN = 0x20000000, LENGTH = 128K
RAM_D2 (xrw) : ORIGIN = 0x30000000, LENGTH = 288K
RAM_D3 (xrw) : ORIGIN = 0x38000000, LENGTH = 64K
ITCMRAM (xrw) : ORIGIN = 0x00000000, LENGTH = 64K
}
```

Build Analyzer × Static Stack Analyzer Search Disassembly

stm32h7-asm.elf - /stm32h7-asm/Debug - Dec 10, 2023, 10:13:51 PM

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM_EXEC	0x24000000	0x2401ffff	128 KB	126,9 KB	1,1 KB	0.86%
DTCMRAM	0x20000000	0x2001ffff	128 KB	126,45 KB	1,55 KB	1.21%
RAM_D2	0x30000000	0x30047fff	288 KB	288 KB	0 B	0.00%
RAM_D3	0x38000000	0x3800ffff	64 KB	64 KB	0 B	0.00%
ITCMRAM	0x00000000	0x0000ffff	64 KB	64 KB	0 B	0.00%

Build Analyzer × Static Stack Analyzer Search Disassembly

stm32h7-asm.elf - /stm32h7-asm/Debug - Dec 10, 2023, 10:13:51 PM

Memory Regions Memory Details

Selection:436 B

Name	Run address ...	Load address (...)	Size
ITCMRAM	0x00000000		64 KB
DTCMRAM	0x20000000		128 KB
.data	0x20000000	0x24000454	16 B
.bss	0x20000010		28 B
_user_heap_stack	0x2000002c		1,5 KB
RAM_EXEC	0x24000000		128 KB
.data	0x20000000	0x24000454	16 B
isr_vector	0x24000000	0x24000000	664 B
g_pfnVectors	0x24000000	0x24000000	0 B
.text	0x24000298	0x24000298	436 B
.rodata	0x2400044c	0x2400044c	0 B
.init_array	0x2400044c	0x2400044c	4 B
.fini_array	0x24000450	0x24000450	4 B
RAM_D2	0x30000000		288 KB
RAM_D3	0x38000000		64 KB

Potrebno spremeniti naslov vektorske tabele !!!
//Set Vector table addr. to 0x24000000

```
ldr r1, =VTOR
ldr r0, =0x24000000
str r0, [r1]
```

CubeIDE – Izvedba programa - Flash

STM32H750XBHX_FLASH.Id:

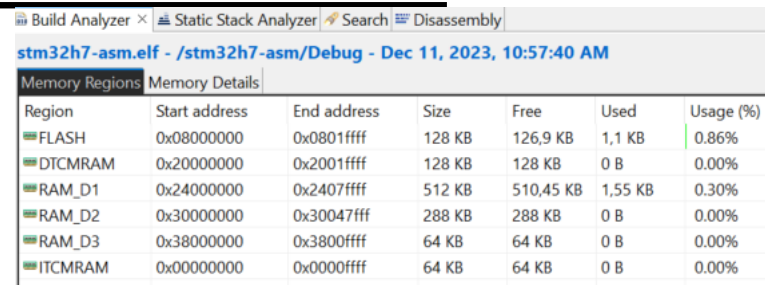
```
/* Define output sections */
SECTIONS
{
/* The startup code goes first into FLASH */
.isr_vector :
{
...
} >FLASH

/* The program code and other data goes into FLASH */
.text :
{
...
} >FLASH

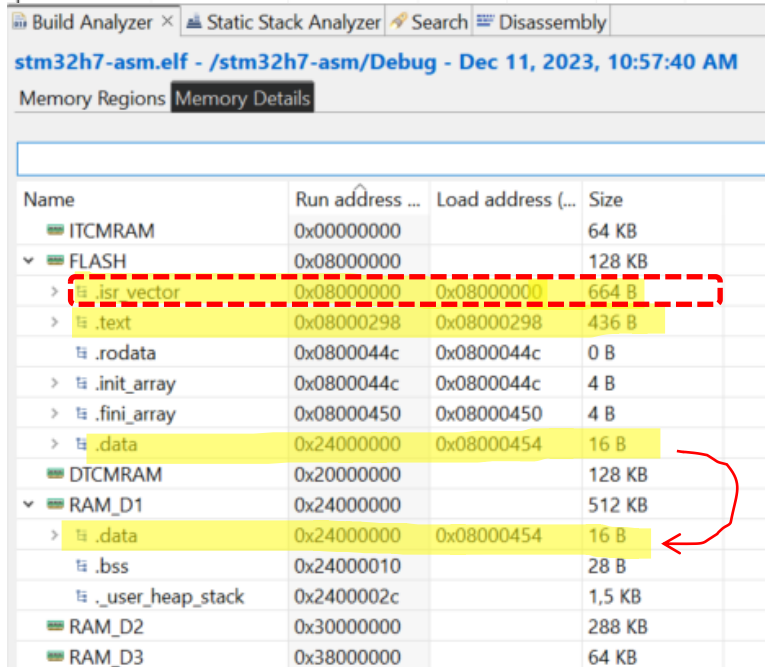
/* Initialized data sections go into RAM_D1 from FLASH*/
.data :
{
...
} >RAM_D1 AT> FLASH

/* Uninitialized data section */
.bss :
{
...
} >RAM_D1
```

```
MEMORY /* Specify the memory areas */
{
FLASH (rx) : ORIGIN = 0x08000000, LENGTH = 128K
DTCMRAM (xrw) : ORIGIN = 0x20000000, LENGTH = 128K
RAM_D1 (xrw) : ORIGIN = 0x24000000, LENGTH = 512K
RAM_D2 (xrw) : ORIGIN = 0x30000000, LENGTH = 288K
RAM_D3 (xrw) : ORIGIN = 0x38000000, LENGTH = 64K
ITCMRAM (xrw) : ORIGIN = 0x00000000, LENGTH = 64K
}
```



Region	Start address	End address	Size	Free	Used	Usage (%)
FLASH	0x08000000	0x0801ffff	128 KB	126,9 KB	1,1 KB	0.86%
DTCMRAM	0x20000000	0x2001ffff	128 KB	128 KB	0 B	0.00%
RAM_D1	0x24000000	0x2407ffff	512 KB	510,45 KB	1,55 KB	0.30%
RAM_D2	0x30000000	0x30047fff	288 KB	288 KB	0 B	0.00%
RAM_D3	0x38000000	0x3800ffff	64 KB	64 KB	0 B	0.00%
ITCMRAM	0x00000000	0x0000ffff	64 KB	64 KB	0 B	0.00%



Name	Run address ...	Load address (...)	Size
ITCMRAM	0x00000000		64 KB
FLASH	0x08000000		128 KB
> .isr_vector	0x08000000	0x08000000	664 B
> .text	0x08000298	0x08000298	436 B
.rodata	0x0800044c	0x0800044c	0 B
.init_array	0x0800044c	0x0800044c	4 B
.fini_array	0x08000450	0x08000450	4 B
.data	0x24000000	0x08000454	16 B
DTCMRAM	0x20000000		128 KB
RAM_D1	0x24000000		512 KB
> .data	0x24000000	0x08000454	16 B
.bss	0x24000010		28 B
.user_heap_stack	0x2400002c		1,5 KB
RAM_D2	0x30000000		288 KB
RAM_D3	0x38000000		64 KB

GitHub repozitorij ORLab-STM32H7

LAPSYLAB / ORLab-STM32H7

ode Issues Pull requests Actions Projects Wiki Security Insights Settings

ORLab-STM32H7 Public

main 1 Branch 0 Tags

Go to file

LAPSYLAB	USART3 solution added	95129
DWT_Cycles_Measurements	Minor	
Docs	Docs added	
STM32H750B-DK_BSP_C_Basic	STM32H750B-DK_BSP_C_Basic added	
STM32H750B-DK_C_Basic	Basic project in C added	
asm_sol_GPIO_LEDs	Minor	
asm_sol_SysTick_LEDs	Minor	
asm_sol_USART3	USART3 solution added	
stm32h7-asm	Minor	
.gitignore	Initial commit	
README.md	Minor	

Projekti:

- stm32h7-asm
 - **Osnovni projekt za zbirnik**
- STM32H750B-DK BSP C Basic
 - **Osnovni projekt z BSP (brez CubeMx)**
- STM32H750B-DK C Basic
 - **Osnovni projekt z CubeMx**
- DWT Cycles Measurements
 - Main.s za meritve hitrosti izvajanja delov kode in preučevanje delovanja cevovoda

Docs:

- **Dokumentacija**

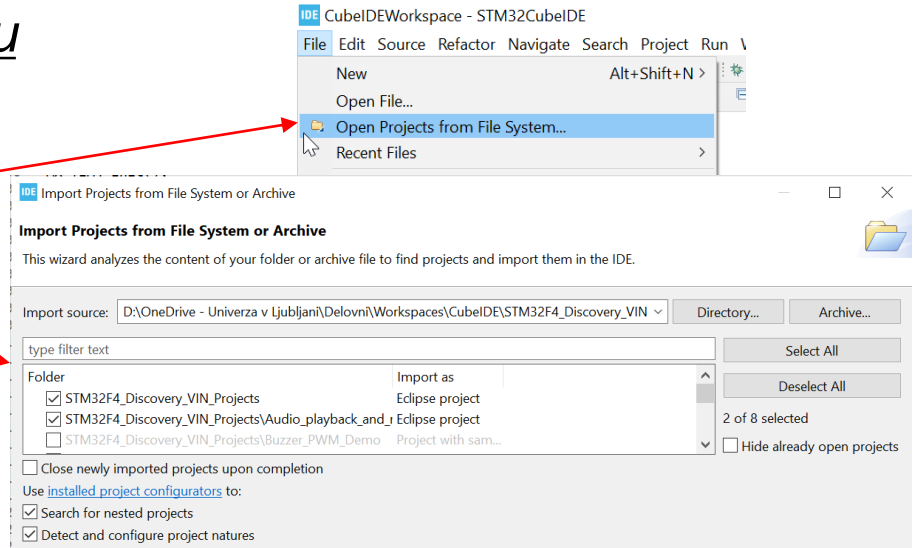
Rešitve:

- asm_sol...
 - Main.s datoteke za rešitev naloge

CubeIDE – Uvoz in delo na projektu

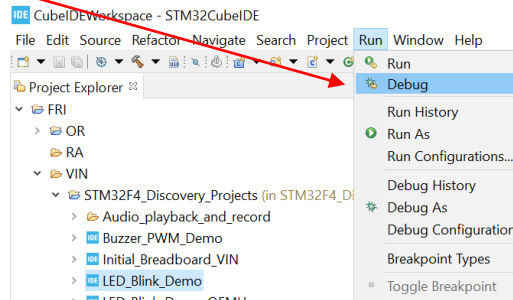
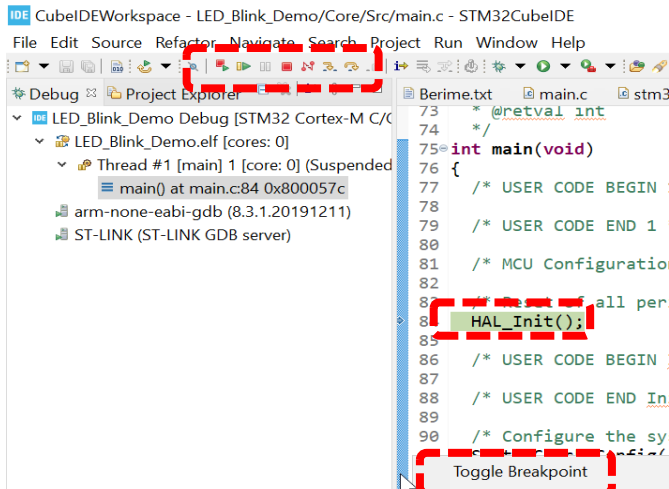
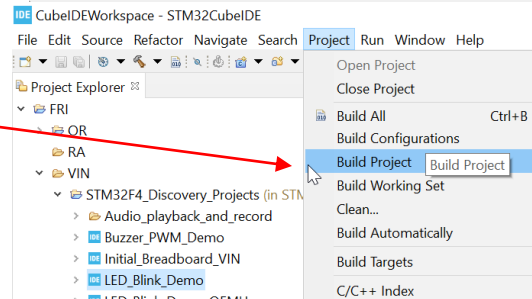
Vzpostavitev začetnega projekta :

- Uvoz obstoječega
 - Open projects from File System
 - Select project(s)
- **Nov projekt Cube MX ->**
(v nadaljevanju)



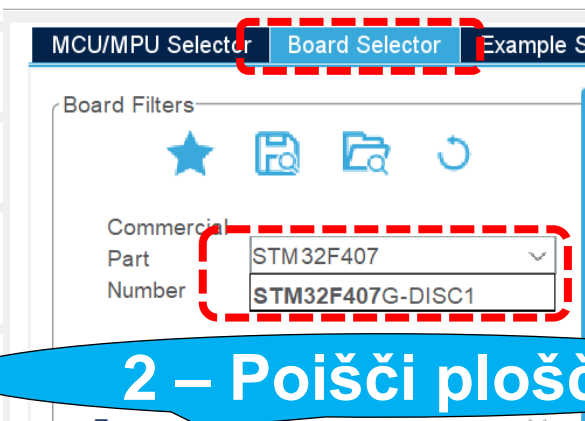
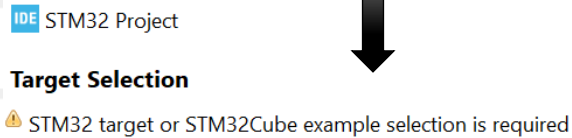
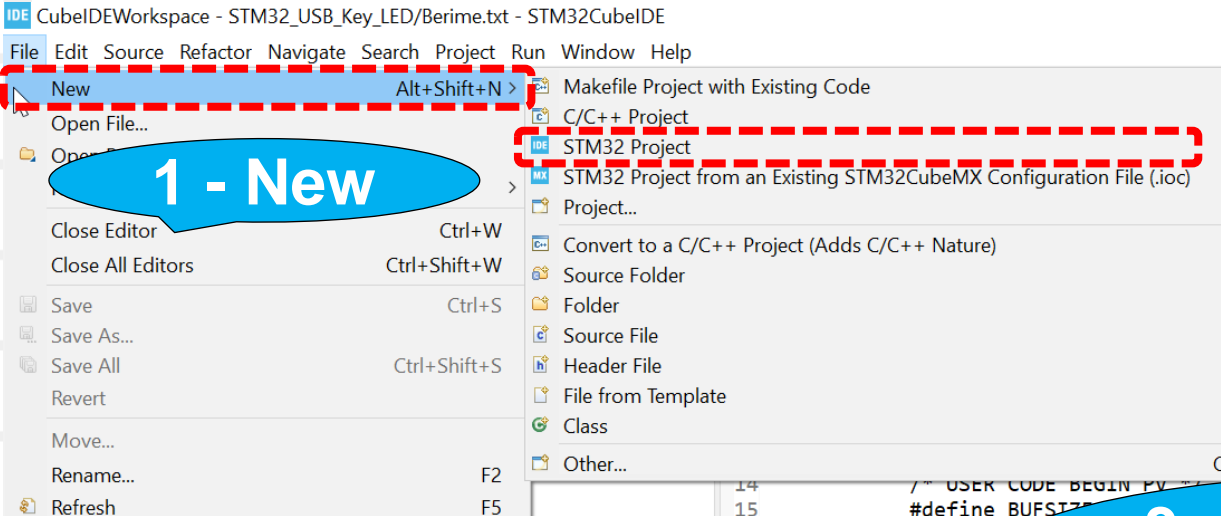
Prevajanje, zagon :

- Project -> Build Project
- Run -> Debug
- Step (Into,Over), Breakpoints



CubeIDE – Vzpostavitev novega projekta I

Nov projekt :



CubeIDE – Vzpostavitev novega projekta II

Konfiguracija : priključki, knjižnice

STM32Cube MCU packages and embedded software packs

- Copy all used libraries into the project folder
- Copy only the necessary library files
- Add necessary library files as reference in the toolchain project configuration file

Generated files

- Generate peripheral initialization as a pair of '.c'/'h' files per peripheral
- Backup previously generated files when re-generating
- Keep User Code when re-generating
- Delete previously generated files when not re-generated

HAL Settings

- Set all free pins as analog (to optimize the power consumption)
- Enable Full Assert

Template Settings

Select a template to generate customized code

Project Settings

Project Name: LED_GPIO_C_Baremetal_C

Project Location: D:\Delovni\CubeIDE\CubeIDEWorkspace

Application Structure: Advanced Do not generate the main()

Toolchain Folder Location: D:\Delovni\CubeIDE\CubeIDEWorkspace\LED_GPIO_C_Baremetal_C

Toolchain / IDE: STM32CubeIDE Generate Under Root

Linker Settings

Minimum Heap Size: 0x200
Minimum Stack Size: 0x400

Thread-safe Settings

Cortex-MANS

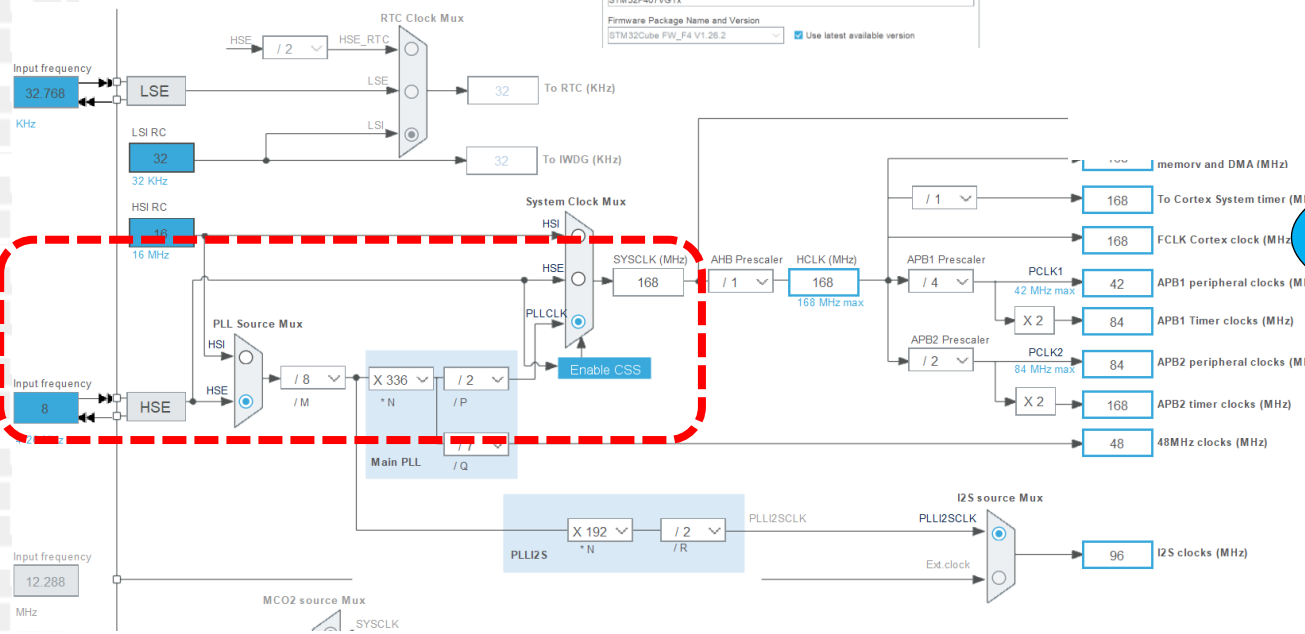
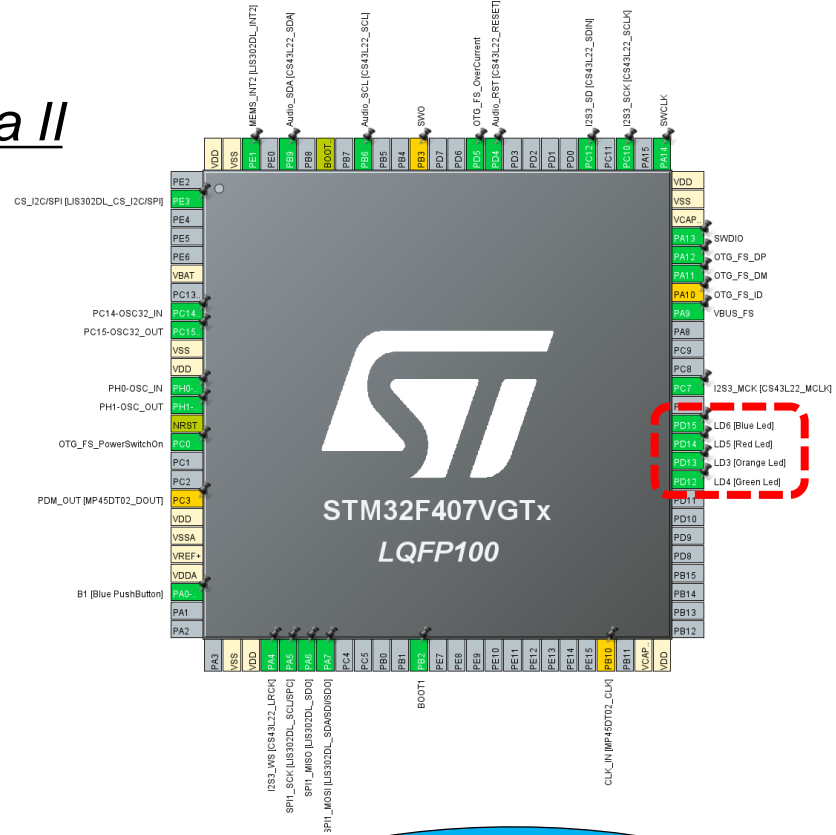
- Enable multi-threaded support

Thread-safe Locking Strategy: Default - Mapping suitable strategy depending on RTOS selection

McU and Firmware Package

McU Reference: STM32F407VGTx

Firmware Package Name and Version: STM32Cube_FW_F4_V1.26.2 Use latest available version



4 – Preveri nastavitve