

Organizacija računalnikov

Laboratorijske vaje

R. Rozman 2025

Vsebina vaj

2 tematska sklopa :

1. Programiranje v zbirnem jeziku ARM (4 vaje)

- Ponovitev RA, razširitev, nadgradnja:
 - Delo z biti, sklad, podprogrami

Primeri in osnove tudi v jeziku C (neobvezno)

2. Sistemske naprave v zbirniku (7 vaj) - (*STM32H750B-DK Discovery, STM32F4 Discovery, FRI-SMS*)

- Paralelni vhod/izhod: (G)PIO
- Časovniki: TIM, SysTick, TC
- Serijske povezave: U(S)ART, USB VCOM port, DBGU
- Prekinitve, prekinitveni krmilnik, Mini RTOS (AIC)

3. MiMo lab. vaja (vmes)

2 obvezni in 2 neobvezni domači nalogi

1. **MiMo**, osnovna (obv.), MiMo dodatno delo (neobv.)
 2. **ARM, STM32 - aplikacija** (obv.), ARM, STM-Dodatna (razširitve, aplikacija, senzorji) (neobv.)
- Vsako dodatno delo šteje za višjo oceno !!!

Vsebinska vaj

Primer zadnje vaje – Mini RTOS :

1. Mini RTOS

- Preprost operacijski sistem za več procesov
 - 1000x v sekundi preklopi med procesoma
- Task0: Blink LED1
- Task1: Blink LED2

TASK1_Start:

```
b1 LED2_OFF
mov r0,#1
mov r1,'#+'
b1 ITM_Send
mov r0,'#1'
b1 SEND_UART

mov r0,#500
b1 DELAY // Zakasnitev SW Delay: r0 x 1msec

b1 LED2_ON
mov r0,#1
mov r1,'#-'
b1 ITM_Send

mov r0,#500
b1 DELAY // Zakasnitev SW Delay: r0 x 1msec

b TASK1_Start
```

TASK0_Start:

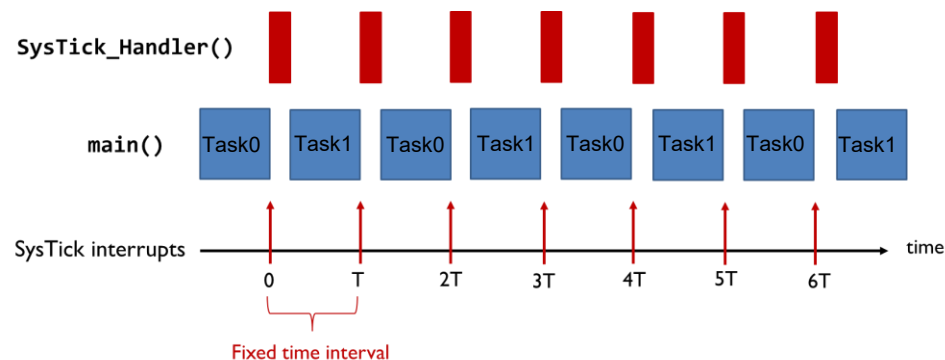
```
b1 LED1_ON
mov r0,#0
mov r1,'#1'
b1 ITM_Send
mov r0,'#0'
b1 SEND_UART

mov r0,#500
b1 DELAY // Zakasnitev SW Delay: r0 x 1msec

b1 LED1_OFF
mov r0,#0
mov r1,'#0'
b1 ITM_Send

mov r0,#500
b1 DELAY // Zakasnitev SW Delay: r0 x 1msec

b TASK0_Start
```



Ocenjevanje OR

- Vaje prispevajo **50%** h končni oceni in morajo biti opravljene:
 - **Pozitivne domače naloge** (obvezni del, približno za oceno do 8),
 - **Dodatne domače naloge** (neobvezni del – višja ocena).
 - Se prišteje obveznemu delu
 - *2022: Vzporedno uvajanje STM32F4, STM32H7 Discovery*
 - *2023: Vzporedno: STM32H7, STM32F4, FRI-SMS*
 - *2024,25: Osrednja platforma: STM32H7, (STM32F4, FRI-SMS)*
- Ustni izpit 50%

Spletni simulator cpulator – 1.sklop

- <https://cpulator.01xz.net/?sys=arm>
- začetni projekt OR:
 - <https://cpulator.01xz.net/?sys=arm&loadasm=share/s8zU3xx.s>

```
.text
.org 0x20

@spremenljivke
STEVI1: .word 0x10
STEVI2: .word 0x40
REZ: .word 0

.align

.global _start

_start:

    adr r0, STEVI1
    ldr r1, [r0]

    adr r0, STEVI2
    ldr r2, [r0]

    add r3, r1, r2

    adr r0, REZ
    str r3, [r0]

end:    b    end
```

The screenshot displays the cpulator web interface. The top navigation bar includes buttons for 'Stopped', 'Step Into', 'Step Over', 'Step Out', 'Continue', 'Stop', 'Restart', and 'Reload'. Below this is a 'File' dropdown and a 'Help' button. The main interface is divided into several panels:

- Registers:** A table showing the current state of ARM registers. The 'r0' register is highlighted with a value of 00000058.
- Disassembly (Ctrl-D):** A table showing the assembly code being executed. The address 00000024 is selected, and the instruction is 'ldr r0, [pc, #24] ; 0x40'.
- Memory (Ctrl-M):** A table showing the memory contents and ASCII values. The address 00000000 is selected, and the memory content is 00000000.

The bottom panel shows the 'Messages' section, which contains the following text:

```
Link: arm-altera-eabi-ld --script build_arm.ld -e _start -u _start -o work/asmETdoww.s.elf work/asmETdoww.s.o
Compile succeeded.

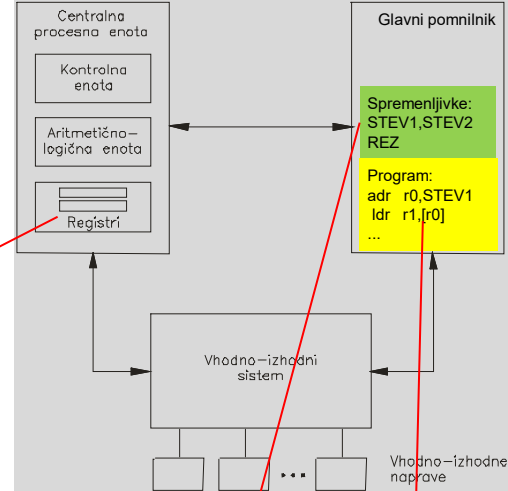
Compiling...
Code and data loaded from ELF executable into memory. Total size is 96 bytes.
Assemble: arm-altera-eabi-as -mfloat-abi=soft -march=armv7-a -mcpu=cortex-a9 -mfpu=neon-fp16 --gdwarf2 -o work/asmreEAFP.s.o work/asmreEAFP.s
Link: arm-altera-eabi-ld --script build_arm.ld -e _start -u _start -o work/asmreEAFP.s.elf work/asmreEAFP.s.o
Compile succeeded.
```

RA - Praktično delo: vsota dveh števil

<https://cpulabor.01xz.net/?sys=arm&loadasm=share/s8zU3xx.s>

Zbirni jezik	Opis ukaza	Strojni jezik
adr r0, stev1	R0 ← nasl. stev1	0xE24F0014
ldr r1, [r0]	R1 ← M[R0]	0xE5901000
adr r0, stev2	R0 ← nasl. stev2	0xE24F0018
ldr r2, [r0]	R2 ← M[R0]	0xE5902000
add r3, r2, r1	R3 ← R1 + R2	0xE0823001
adr r0, rez	R0 ← nasl. rez	0xE24F0020
str r3, [r0]	M[R0] ← R3	0xE5803000

RA - Ponovitev



Stopped

Step Into F2 Step Over Ctrl-F2 Step Out Shift-F2 Continue F3 Stop F4 Restart Ctrl-R Reload Ctrl-Shift-L File Help

Registers

Refresh

r0	00000000
r1	00000000
r2	00000000
r3	00000000
r4	00000000
r5	00000000
r6	00000000
r7	00000000
r8	00000000
r9	00000000
r10	00000000
r11	00000000
r12	00000000
sp	00000000
lr	00000000
pc	0000002c
cpsr	000001d3 NZCVI SVC
spsr	00000000 NZCVI ?

Disassembly (Ctrl-D)

Go to address, label, or register: 00000000

Address	Opcode	Disassembly
00000020	00000010	STEV1: andeq r0, r0, r0,
00000024	00000040	STEV2: andeq r0, r0, r0,
00000028	00000000	REZ: andeq r0, r0, r0
0000002c	e24f0014	9 .align
00000030	e5901000	11 .global _start
00000034	e24f0018	12 _start: adr r0, STEV1
00000038	e5902000	14 _start: adr r0, 0x20 (0x20: enc
0000003c	e0813002	15 ldr r1, [r0]
00000040	e24f0020	17 adr r0, STEV2
00000044	e5803000	18 ldr r2, [r0]
00000048	eaffffff	20 add r3, r1, r2
		22 adr r0, REZ
		23 str r3, [r0]
		26 end: b er
		end: b 0x48 (0x48: enc

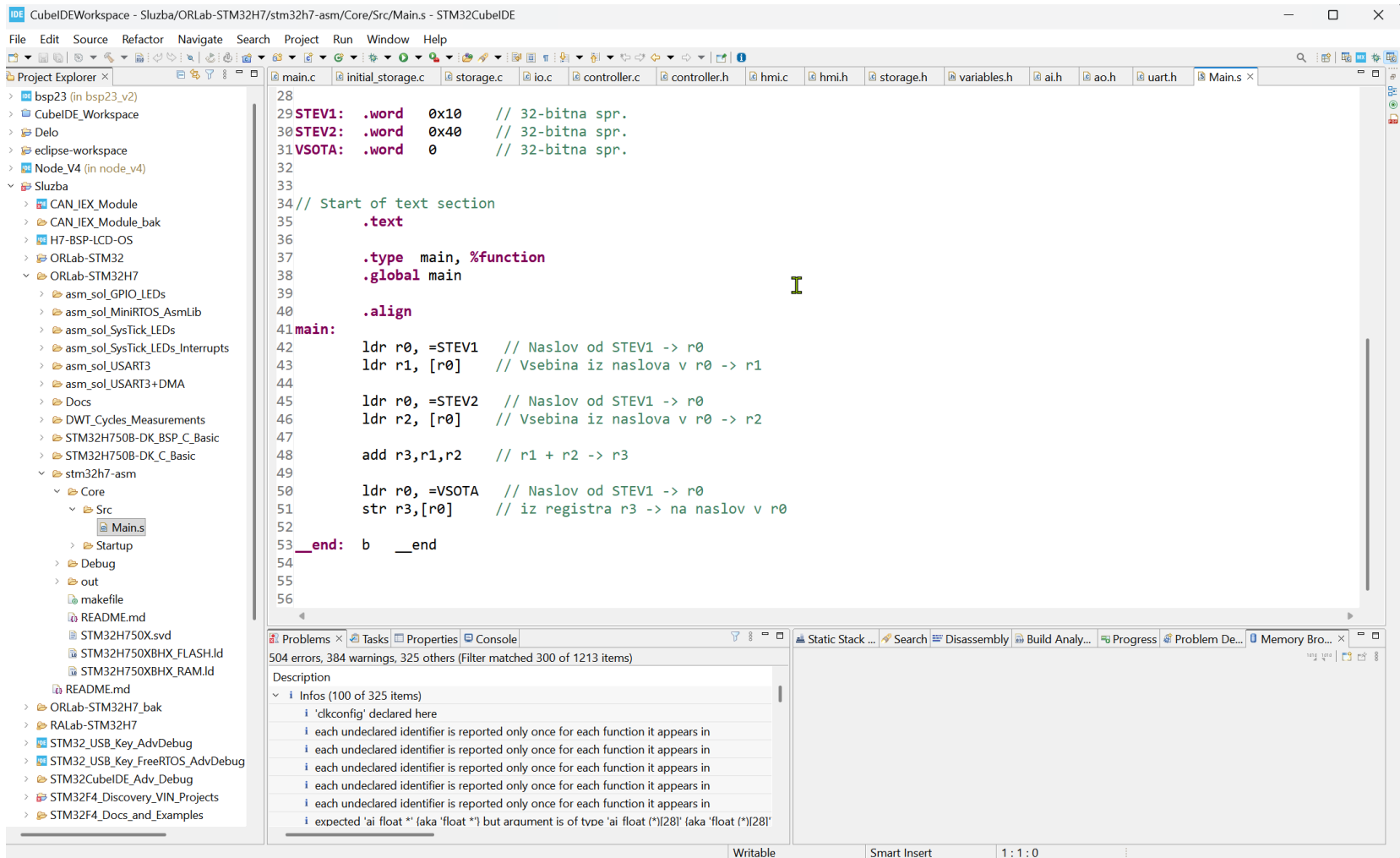
Memory (Ctrl-M)

Go to address, label, or register:

Address	Memory contents and ASCII
00000000	00000000 00000000 00000000 00000000
00000010	00000000 00000000 00000000 00000000
00000020	00000010 00000040 00000000 e24f0014
00000030	e5901000 e24f0018 e5902000 e0813002
00000040	e24f0020 e5803000 eaffffff 00000000
00000050	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000060	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000070	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000080	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000090	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
000000a0	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
000000b0	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
000000c0	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
000000d0	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
000000e0	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
000000f0	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000100	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000110	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000120	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000130	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000140	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000150	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
00000160	aaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa

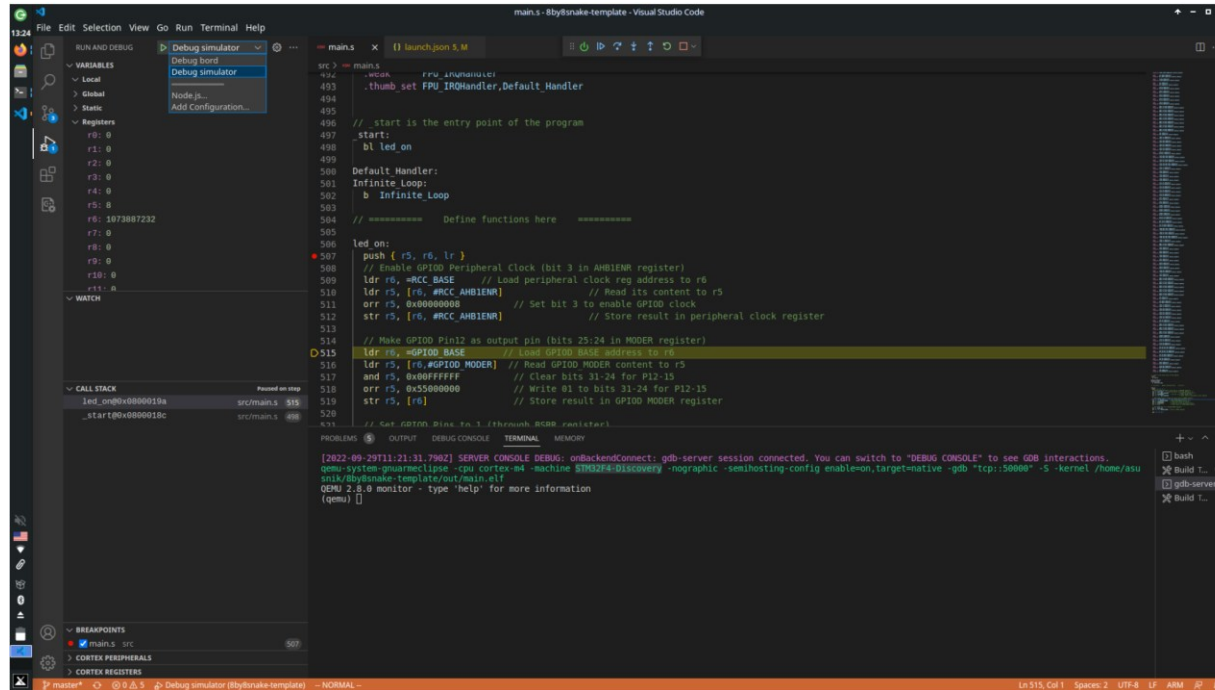
Razvojno okolja 2. sklop: CubeIDE

Delo s ploščami: STM32H7, STM32F4, STM32XX



Razvojno okolja 2. sklop: VSCode

*Delo s ploščami: STM32H7, STM32F4, STM32XX
(ni „uradno“ podprt, je pa zanimiv)*



Instalirate dodatek



STM32CubeIDE for Visual Studio Code

STMicroelectronics [st.com](https://www.st.com) | 135,747 | ★★★★★ (35)

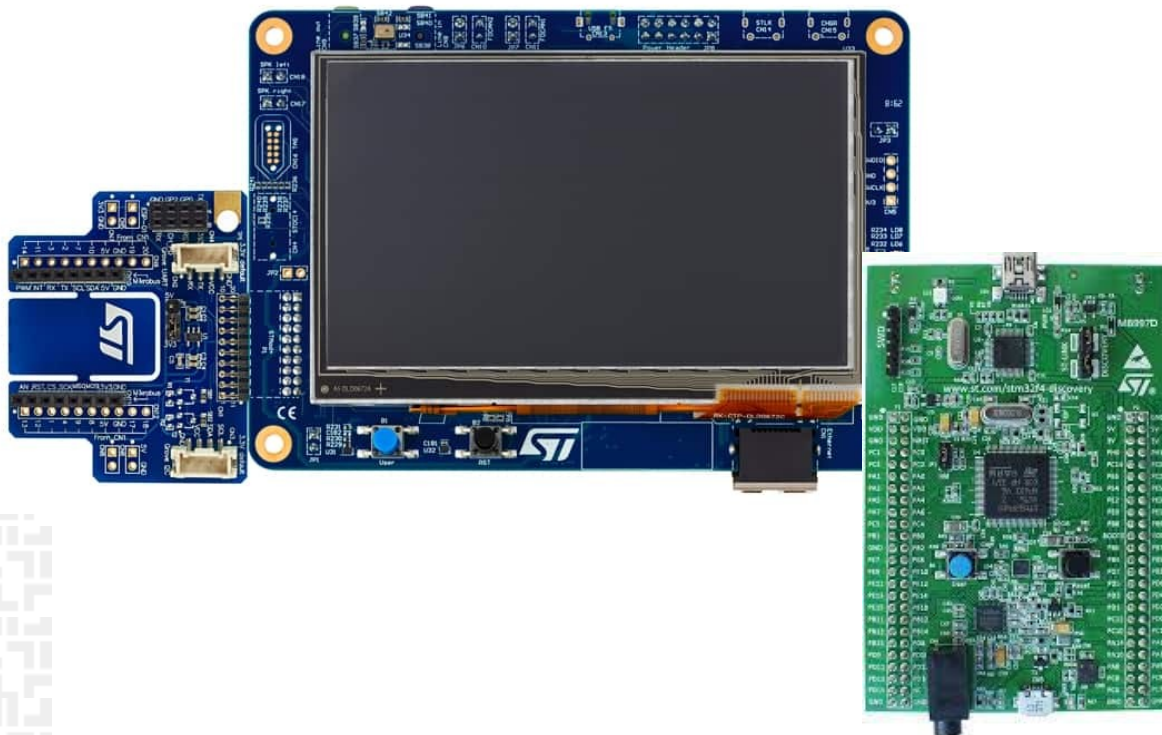
STM32 embedded development support added to Visual Studio Code.

Uninstall

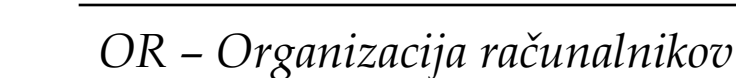
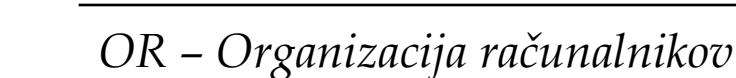
Switch to Pre-Release Version

☒ Auto Update

V 2. sklopu – delo s sistemi STM32H7, STM32F4 , FRI SMS

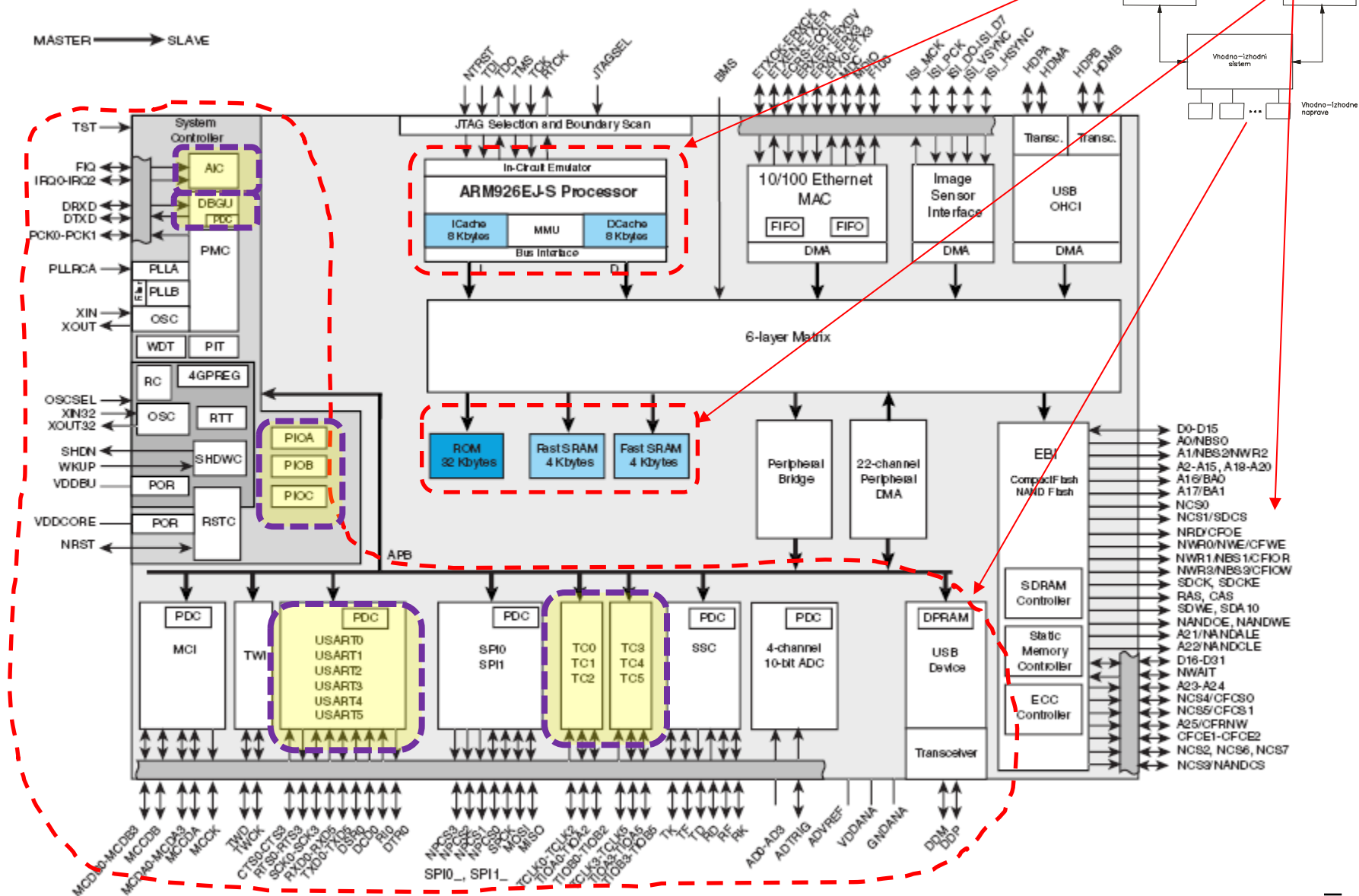


OR – Organizacija računalnikov





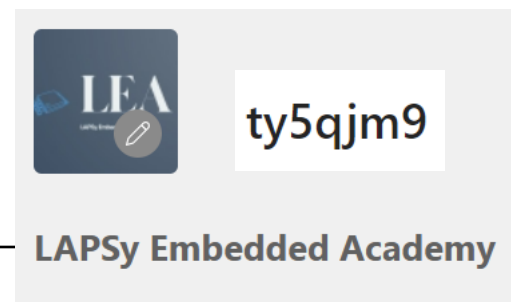
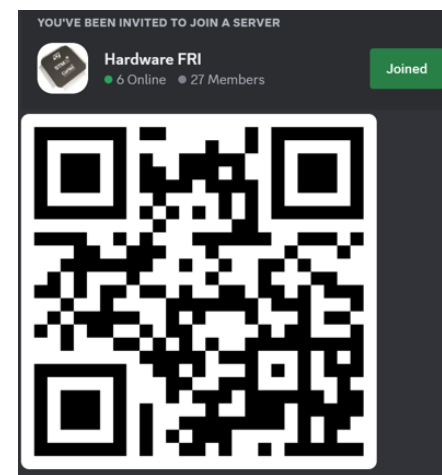
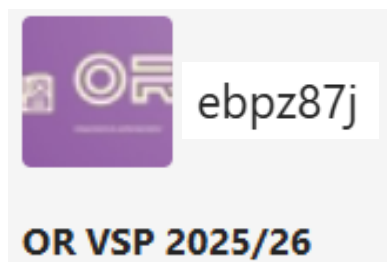
FRISMS: PIO, Časovnik, DBGU UART



Iščete vsebine vaj, dodatne izzive, pomoč, ideje za projekte ?

Dobrodošli:

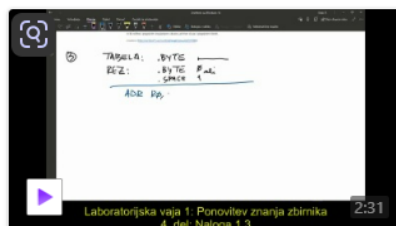
- e-učilnica
- MS Teams
 - **Komunikacija** (govorilne ure, pogovor, pomoč),
 - OneNote zvezek z zapiski predavanj
- Discord(<https://discord.gg/nmzjQU7me7>)
- LEA - Lapsy Embedded Academy
 - Projekt spletnega izobraževalnega portala z vsebinami RA, OR in VIN (ter ostalimi projekti, diplomami, tečaji, ...)
 - <https://unilj.sharepoint.com/sites/LAPSYEmbeddedAcademy/>



LEA - Lapsy Embedded Academy

- <https://unilj.sharepoint.com/sites/LAPSYEmbeddedAcademy/>

Programiranje v zbirniku ARM II (OR LAB I)

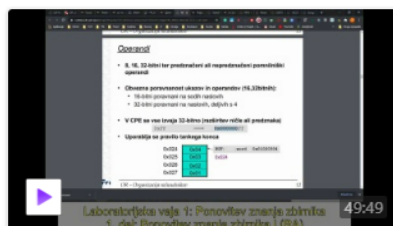


LAPSY Embedded Academy

_OR LAB 01-04 Programiranje v zbirniku II (OR LAB)_Krajši izseki.mp4



Rozman, Robert
Edited April 13, 2024

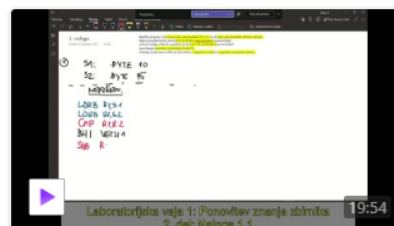


LAPSY Embedded Academy

OR LAB 01.1 Ponovitev znanja zbirnika_Ponovitev znanja zbirnika I...



Rozman, Robert
Edited March 19, 2024

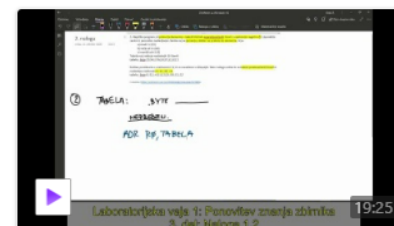


LAPSY Embedded Academy

OR LAB 01.2 Ponovitev znanja zbirnika_Naloga 1.1.mp4



Rozman, Robert
Edited March 19, 2024



LAPSY Embedded Academy

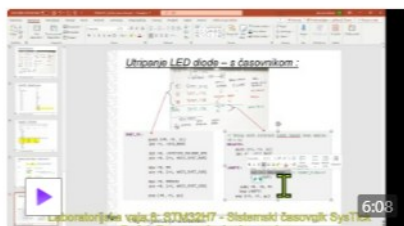
OR LAB 01.3 Ponovitev znanja zbirnika_Naloga 1.2.mp4



Rozman, Robert
Edited March 19, 2024

Programiranje v zbirniku ARM III - STM32H7 (OR LAB II)

See all

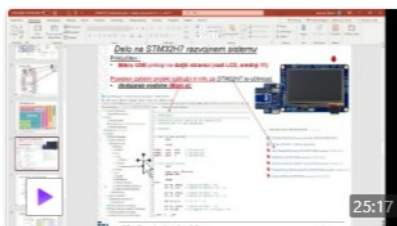


LAPSY Embedded Academy

_OR LAB 06-12 Programiranje v zbirniku III (OR-LAB II-STM32H7)_Krajši...



Rozman, Robert
Edited April 13, 2024



LAPSY Embedded Academy

OR LAB 06.1 STM32H7 - Osnovni projekt, GPIO in LED I_Predstavitev...



Rozman, Robert
Edited March 20, 2024



LAPSY Embedded Academy

OR LAB 06.2 STM32H7 - Osnovni projekt, GPIO in LED I_Projekt v zbirnik...



Rozman, Robert
Edited March 20, 2024



LAPSY Embedded Academy

OR LAB 06.3 STM32H7 - Osnovni projekt, GPIO in LED I_Osnovni projekt ...



Rozman, Robert
Edited March 20, 2024