

*Organizacija računalnikov*

*Laboratorijske vaje*

*R. Rozman 2022*

# Vsebina vaj

- **Programiranje v zbirnem jeziku ARM**
  - ponovitev, razširitev, nadgradnja
  - ~~Prvo preverjanje (predvidoma druga polovica novembra)~~
- **Vhodno izhodne naprave, FRI-SMS (2021: STM32F4 Discovery, 2022:STM32H750-DK2 Discovery)**
  - Paralelni vhod/izhod (PIO)
  - Časovniki (TC)
  - Zaporedni vmesnik (UART, DBGU, USB Vcom)
  - Prekinitve, prekinitveni krmilnik (AIC) \*
  - ~~Zaključno preverjanje – zadnji teden semestra~~

priključitev osnovnih V/I enot, osnove jezika C (neobvezno)
- **2 obvezni in 2 neobvezni domači nalogi**
  - MiMo, osnovna (obv.), MiMo dodatki (neobv.)
  - ARM,STM-Aplikacija (obv.), npr. oddajnik Morsejeve abecede
  - ARM,STM-Dodatna (razširitve, aplikacija, senzorji) (neobv.)
- **Neobvezni predmetni seminar po dogovoru z asistentom**
  - v večjem obsegu, poseben dogovor

# Ocenjevanje\*

- Vaje prispevajo **50%** h končni oceni in morajo biti opravljene:
  - **Pozitivne domače naloge** (obvezni del),
  - **Dodatne domače naloge** (neobvezni del – višja ocena).
    - Se prišteje obveznemu delu
  - ~~preverjanja morajo biti vsaj 50% oz. 100 točk :~~
    - ~~dve preverjanji (90 + 110)~~
      - ~~možnih 200 točk~~
      - ~~vsaj 100 točk na obeh skupaj~~
    - *2022: Vzporedno uvajanje STM32F4, STM32H7 Discovery*

*\* Zaradi Covid situacije se lahko še prilagodi*

# Razvojno okolje WinIDEA



The screenshot displays the WinIDEA development environment with the following components:

- Project Workspace:** Shows a project named 'sample.lcf' with source files 'user.s' and 'crt0.s'.
- Source Code (.text):**

```
.text
NIZ1: .asciz "Timi Zajc je svetovni prvak v smucarskih skokih!"
NIZ2: .space 20
.align
NASLOVI_R: .space 40
.align 1
STEVILO_R: .space 2
```
- Disassembly:** Shows assembly instructions for the '\_start' symbol:

```
00007C004: sub r0, pc, #7C
00004F504: sub r5, pc, #4F
00003C604: sub r6, pc, #3C
000018704: sub r7, pc, #18
```
- Registers:** Lists registers R0 through R8, all containing 00000000.
- Memory Dump (0x0000000):**

```
00000000: 23 00 00 EA 22 00 00 EA #... "..."
00000008: 21 00 00 EA 20 00 00 EA !... "..."
00000010: 1F 00 00 EA 1E 00 00 EA .....
00000018: 1D 00 00 EA 1C 00 00 EA .....
00000020: 54 69 6D 69 20 5A 61 6A Timi Zaj
00000028: 63 20 6A 65 20 73 76 65 c ie sve
```
- Output Window:** Shows successful compilation and linking of 'sample.elf'.
- Status Bar:** Displays 'Ready' and 'STOPPED'.

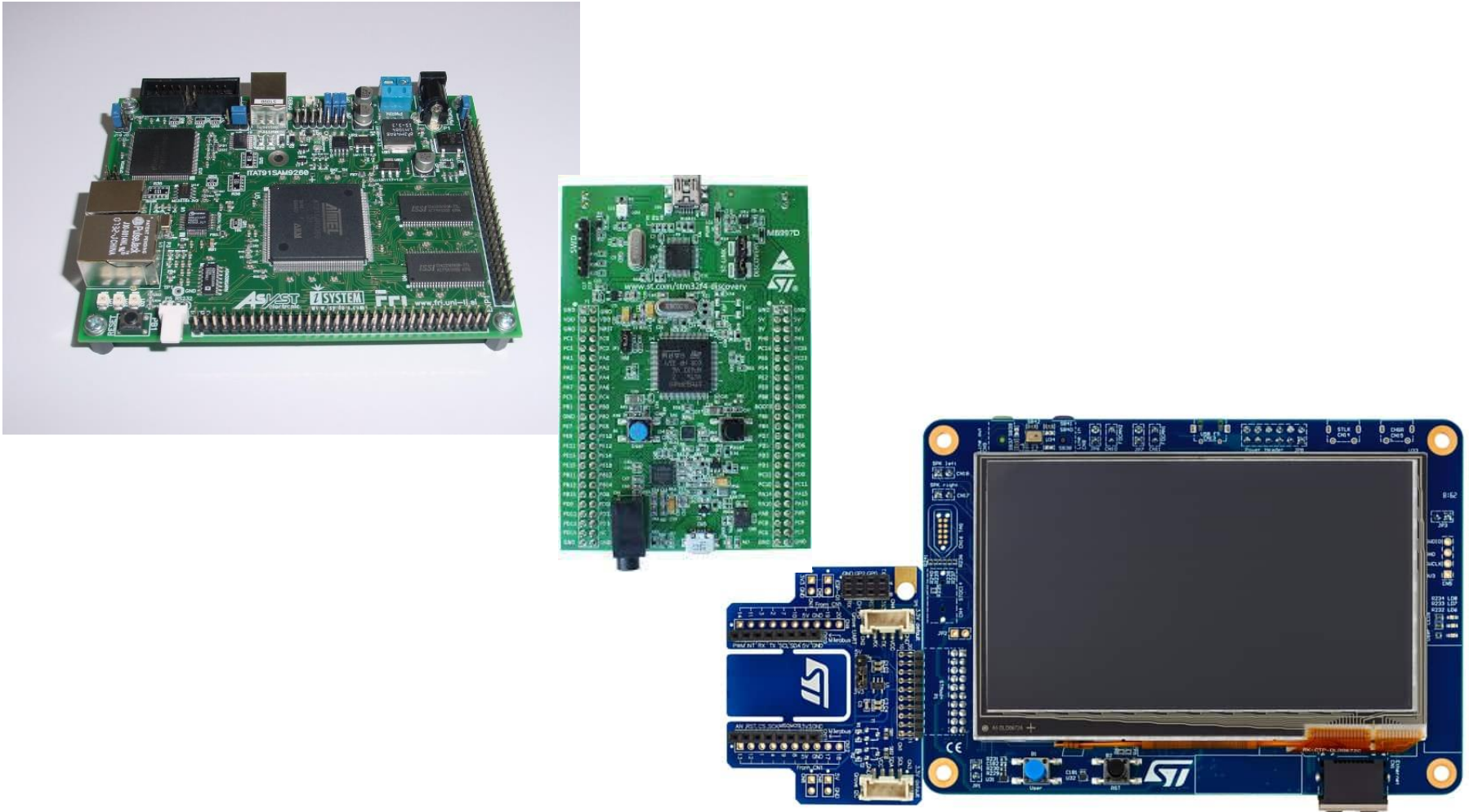


# 2022: še VSCode, CubeIDE

```
src > main.s
492 .weak FPU_IRQHandler
493 .thumb_set FPU_IRQHandler,Default_Handler
494
495
496 // start is the entry point of the program
497 start:
498     bl Led_on
499
500 Default_Handler:
501 Infinite_Loop:
502     b Infinite_Loop
503
504 // ===== Define functions here =====
505
506 Led_on:
507     push { r5, r6, lr }
508     // Enable GPIO Peripheral Clock (bit 3 in AHB1ENR register)
509     ldr r6, =RCC_BASE // Load peripheral clock reg address to r6
510     ldr r5, [r6, #RCC_AHB1ENR] // Read its content to r5
511     orr r5, #0x00000008 // Set bit 3 to enable GPIO clock
512     str r5, [r6, #RCC_AHB1ENR] // Store result in peripheral clock register
513
514 // Make GPIO Pin12 as output pin (bits 25:24 in MODER register)
515     ldr r6, =GPIO_BASE // Load GPIO BASE address to r6
516     ldr r5, [r6, #GPIO_MODER] // Read GPIO MODER content to r5
517     and r5, #0x0FFFFFFF // Clear bits 31-24 for P12-15
518     orr r5, #0x55000000 // Write 01 to bits 31-24 for P12-15
519     str r5, [r6] // Store result in GPIO MODER register
520
521 // Set GPIO Pin12 to 1 (through BSRR register)
```

```
template.s - STM32CubeIDE
54
55 _start:
56 // Enable GPIO Peripheral Clock (bit 3 in AHB1ENR register)
57 ldr r6, = RCC_AHB1ENR // Load peripheral clock reg address to r6
58 ldr r5, [r6] // Read its content to r5
59 orr r5, #0x00000008 // Set bit 3 to enable GPIO clock
60 str r5, [r6] // Store result in peripheral clock register
61
62 // Make GPIO Pin12 as output pin (bits 25:24 in MODER register)
63 ldr r6, = GPIO_MODER // Load GPIO MODER register address to r6
64 ldr r5, [r6] // Read its content to r5
65 bic r5, #0x3000000 // Clear bits 24, 25 for P12
66 orr r5, #0x01000000 // Write 01 to bits 24, 25 for P12
67 str r5, [r6] // Store result in GPIO MODER register
68
69 // Set GPIO Pin12 to 1 (bit 12 in ODR register)
70 ldr r6, = GPIO_ODR // Load GPIO output data register
71 ldr r5, [r6] // Read its content to r5
72 orr r5, #0x1000 // write 1 to pin 12
73 str r5, [r6] // Store result in GPIO output data register
74
75 // Set GPIO Pin12 to 0 (bit 12 in ODR register)
76 ldr r6, = GPIO_ODR // Load GPIO output data register
77 ldr r5, [r6] // Read its content to r5
78 bic r5, #0x1000 // write 0 to pin 12
79 str r5, [r6] // Store result in GPIO output data register
80
81 loop:
82     nop // No operation. Do nothing.
83     b loop // Jump to loop
84
```

# V drugem delu – delo s sistemi FRI SMS, STM32F4, STM32H7



# Naprave: PIO, Časovnik, DBGU UART

