Vhodno izhodne naprave

Laboratorijska vaja 4 - VP 3
STM32-CubeIDE projekt, USART, GPIO (LED, tipka), BSP

VIN projekt - STM32-CubeIDE projekt, USART, GPIO (LED, tipka), BSP

- STM32H7, STM32F4 Discovery board in ostale platforme

- Osnovna projekta v CubeIDE:
  - CubeMX (HAL knjižnica, BSP) – STM32H7, STM32F4
    - CubeMX – osnovni projekt (GPIO – tipka, LED diode, USART)
    - BSP – osnovni projekt (Touch, LCD)
  - STM32H7:
    - Touch panel (I2C, registri, …)
  - STM32F4:
    - USB VComPort

VIN projekt - STM32-CubeIDE projekt, USART, GPIO (LED, tipka), BSP

- STM32H7, STM32F4 Discovery board in ostale platforme

- Osnovna projekta v CubeIDE:
  - CubeMX (HAL knjižnica, BSP) – STM32H7, STM32F4
    - CubeMX – osnovni projekt (GPIO – tipka, LED diode, USART)
    - BSP – osnovni projekt (Touch, LCD)
  - STM32H7:
    - Touch panel (I2C, registri, …)
  - STM32F4:
    - USB VComPort

ARM Cortex M – ISA

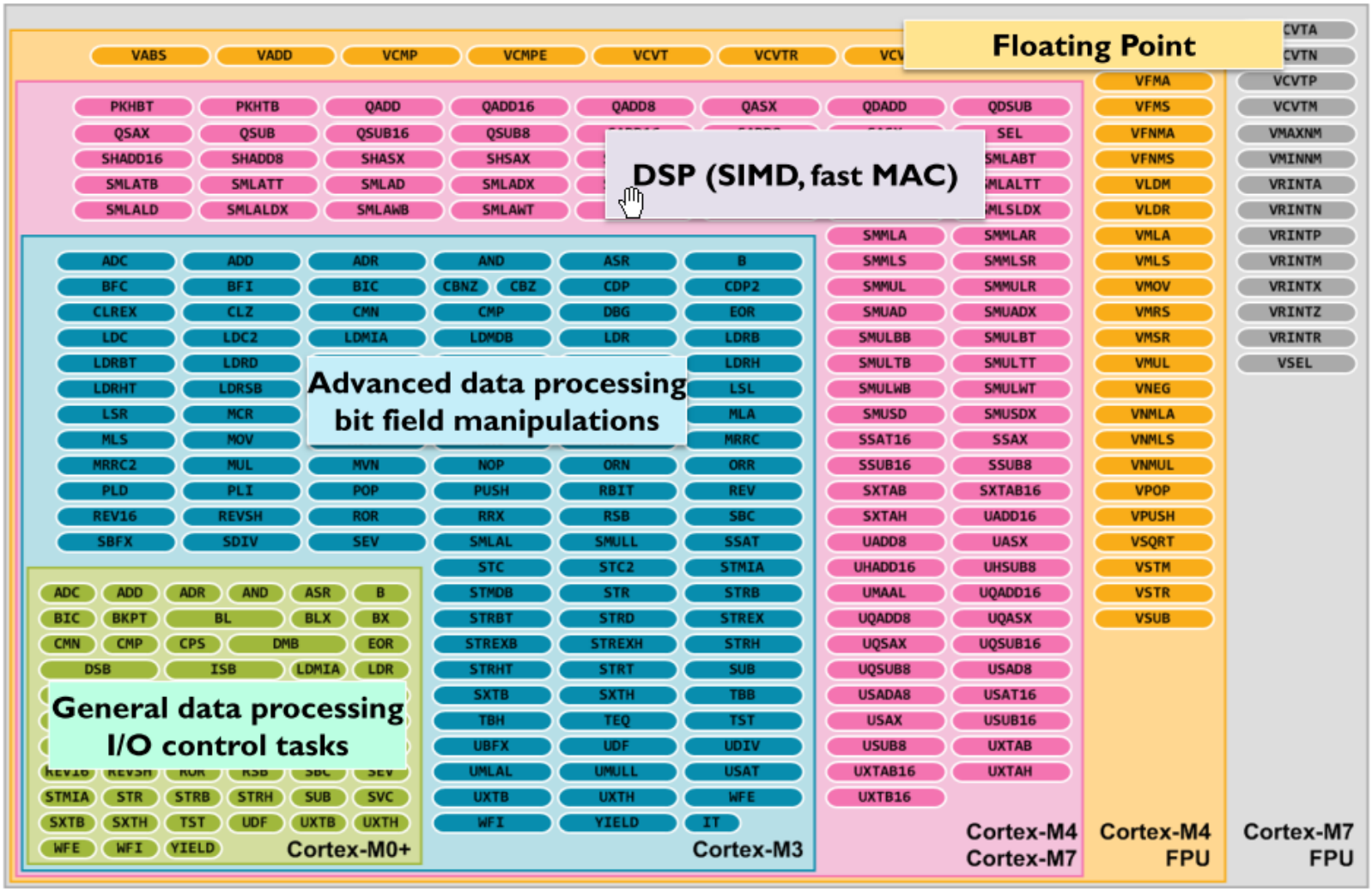
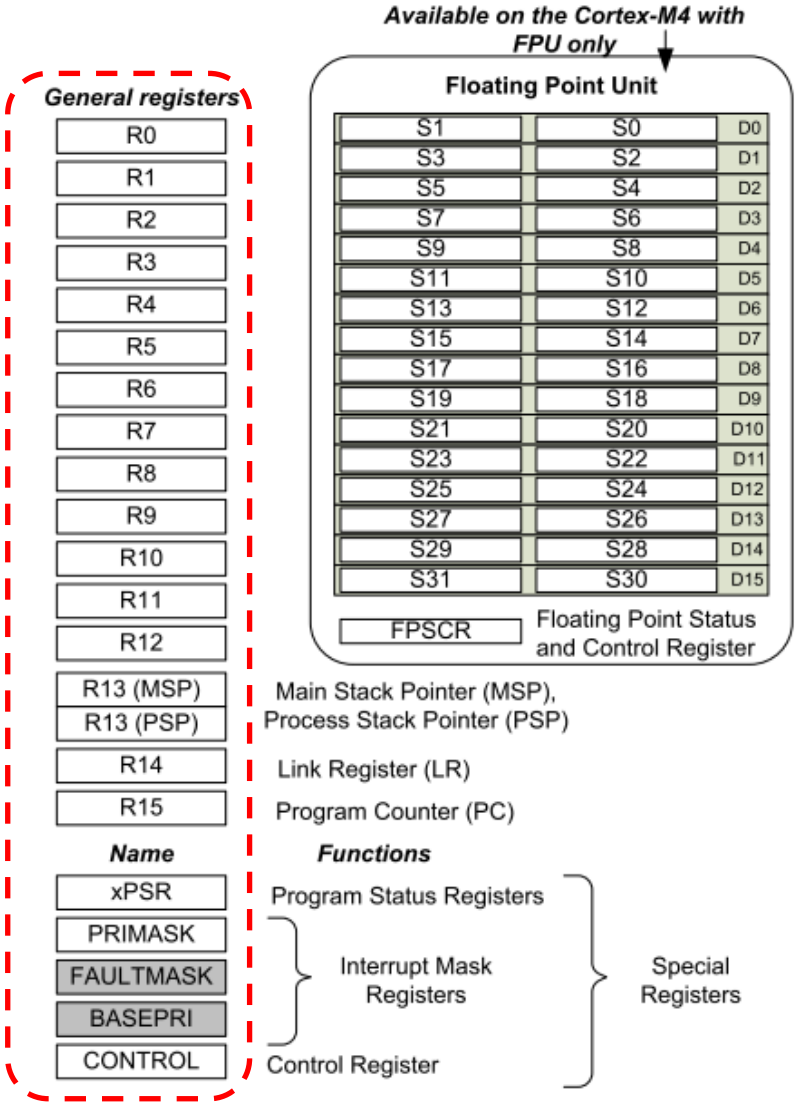


Figure 3: Instruction Set support in the Cortex-M processors

ARM Cortex M – Programski model



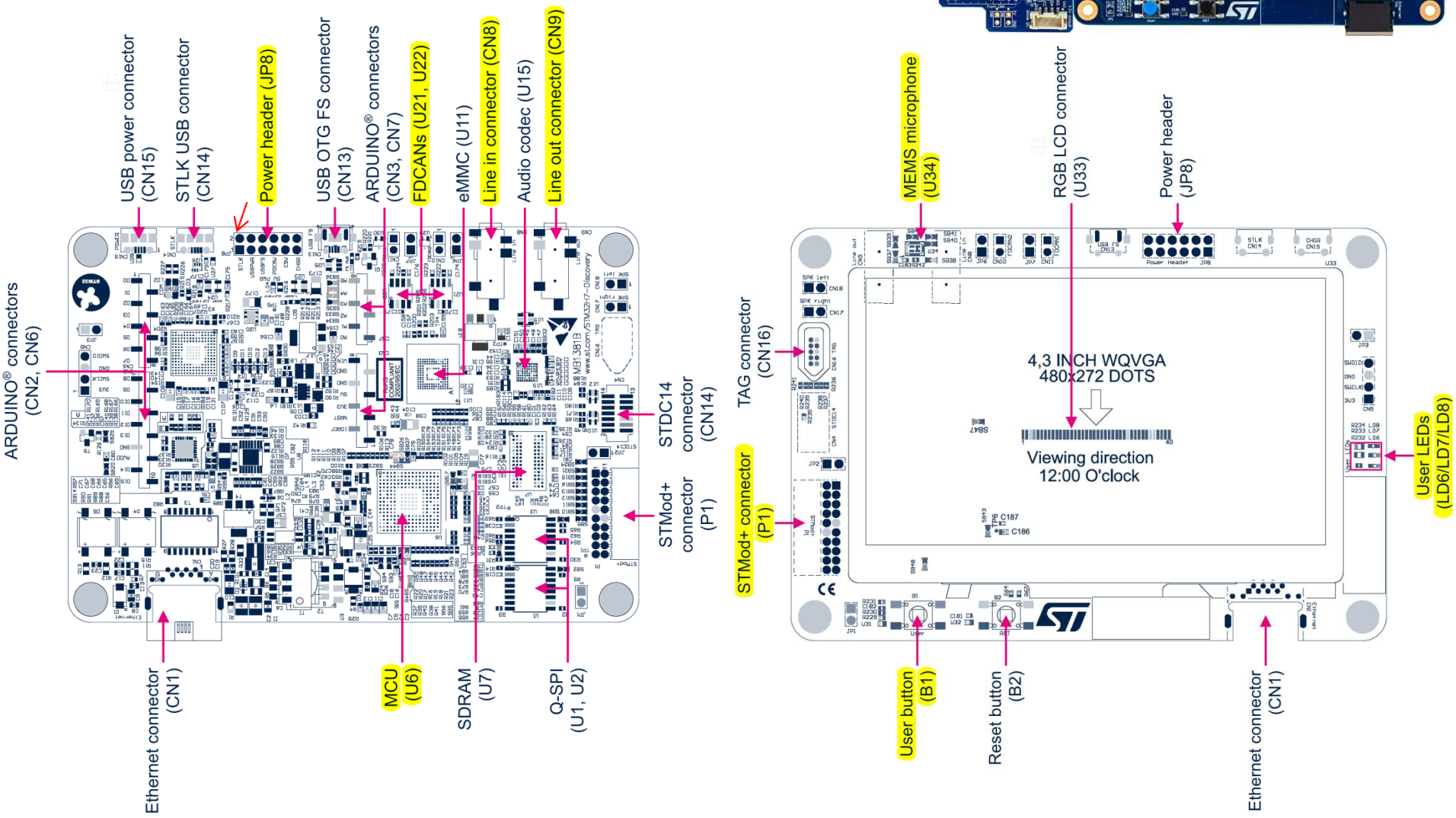
VIN Projekt – Osnovna platforma

STM32H750B-DK Discovery razvojni sistem

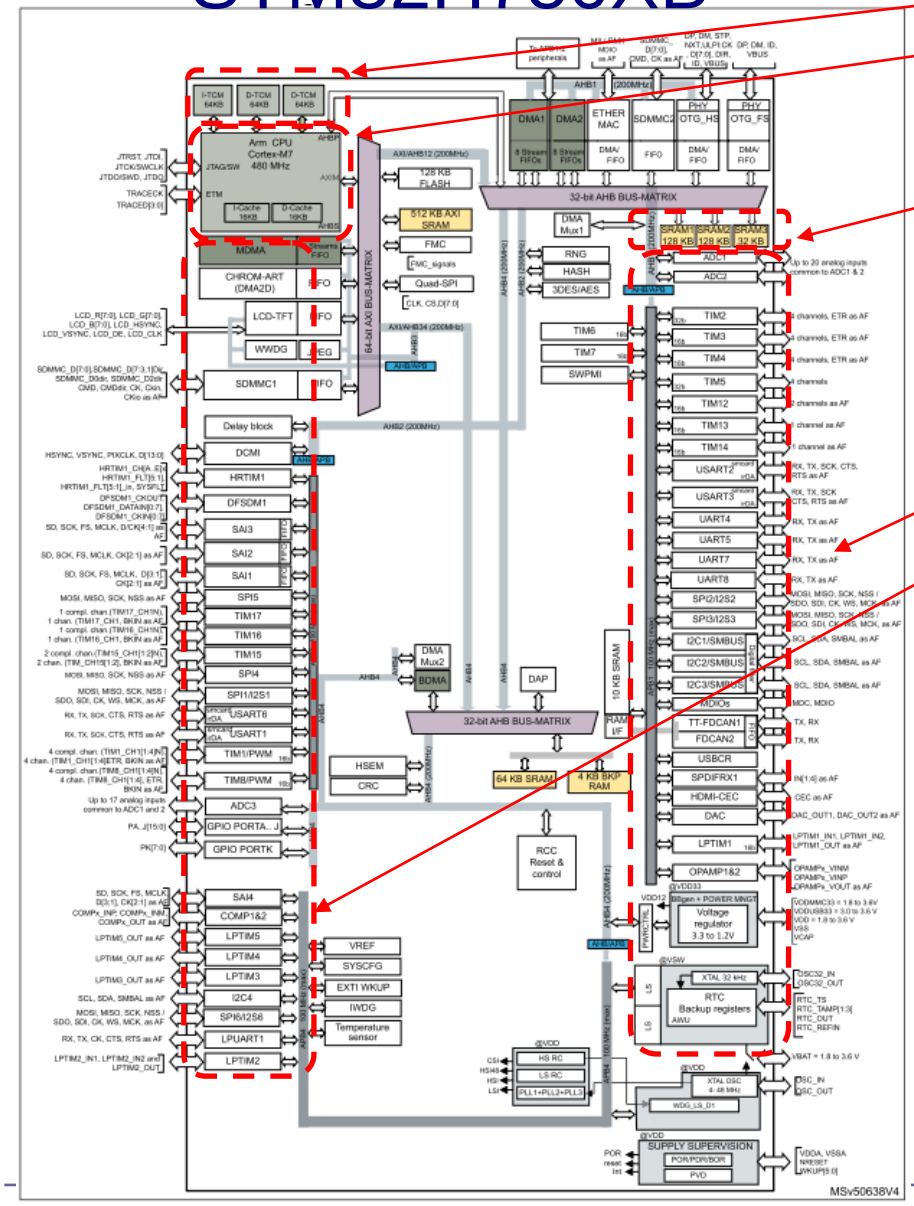
- Arm® Cortex® core-based microcontroller with 128 Kbytes (STM32H750XBH6) of Flash memory and 1 Mbyte of RAM, in TFBGA240+25 package
- 4.3” RGB interface LCD with touch panel connector
- Ethernet compliant with IEEE-802.3-2002, and POE
- USB OTG FS with Micro-AB connector
- SAI audio codec
- One ST-MEMS 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+

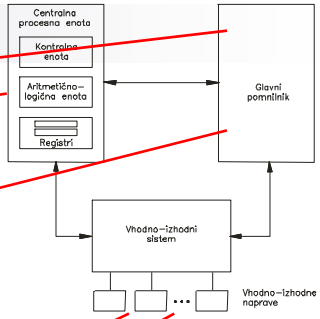
https://www.st.com/en/evaluation-tools/stm32h750b-dk.html

STM32H750B-DK Discovery razvojni sistem



STM32H750XB





VIN Projekt – Osnovna platforma

STM32F407 ST Discovery

STM Discovery F4 (Cortex M4)

•STM32F407VGT6 microcontroller featuring 32-bit Arm® Cortex®-M4 with FPU core, 1-Mbyte Flash memory and 192-Kbyte RAM in an LQFP100 package

•USB OTG FS

•ST MEMS 3-axis accelerometer

•ST-MEMS audio sensor omni-directional digital microphone

•Audio DAC with integrated class D speaker driver

•User and reset push-buttons

•Eight LEDs:

- •LD1 (red/green) for USB communication
- •LD2 (red) for 3.3 V power on
- •Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)

•Board connectors:

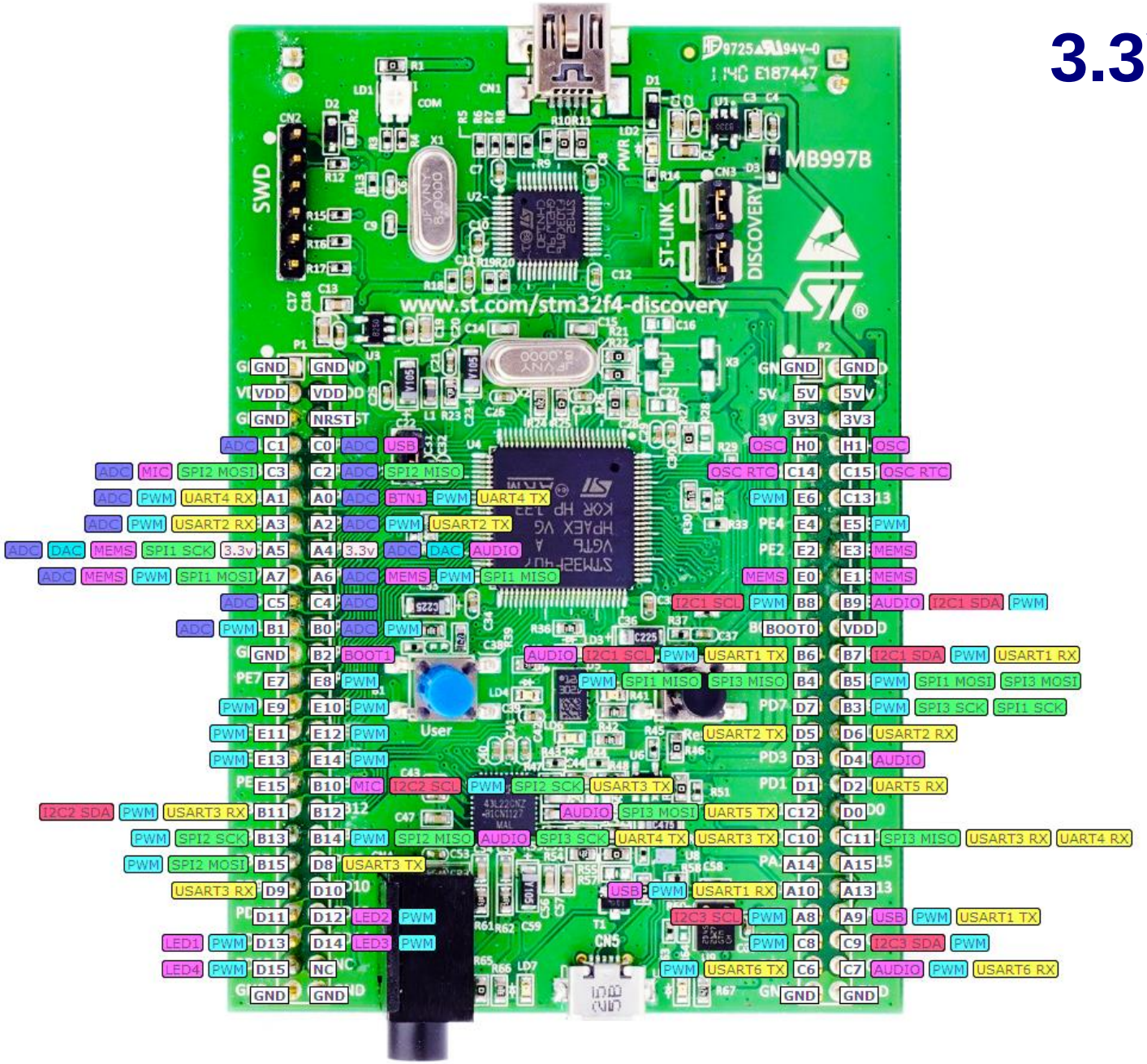
- •USB with Micro-AB
- •Stereo headphone output jack
- •2.54 mm pitch extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing

•External application power supply: 3 V and 5 V

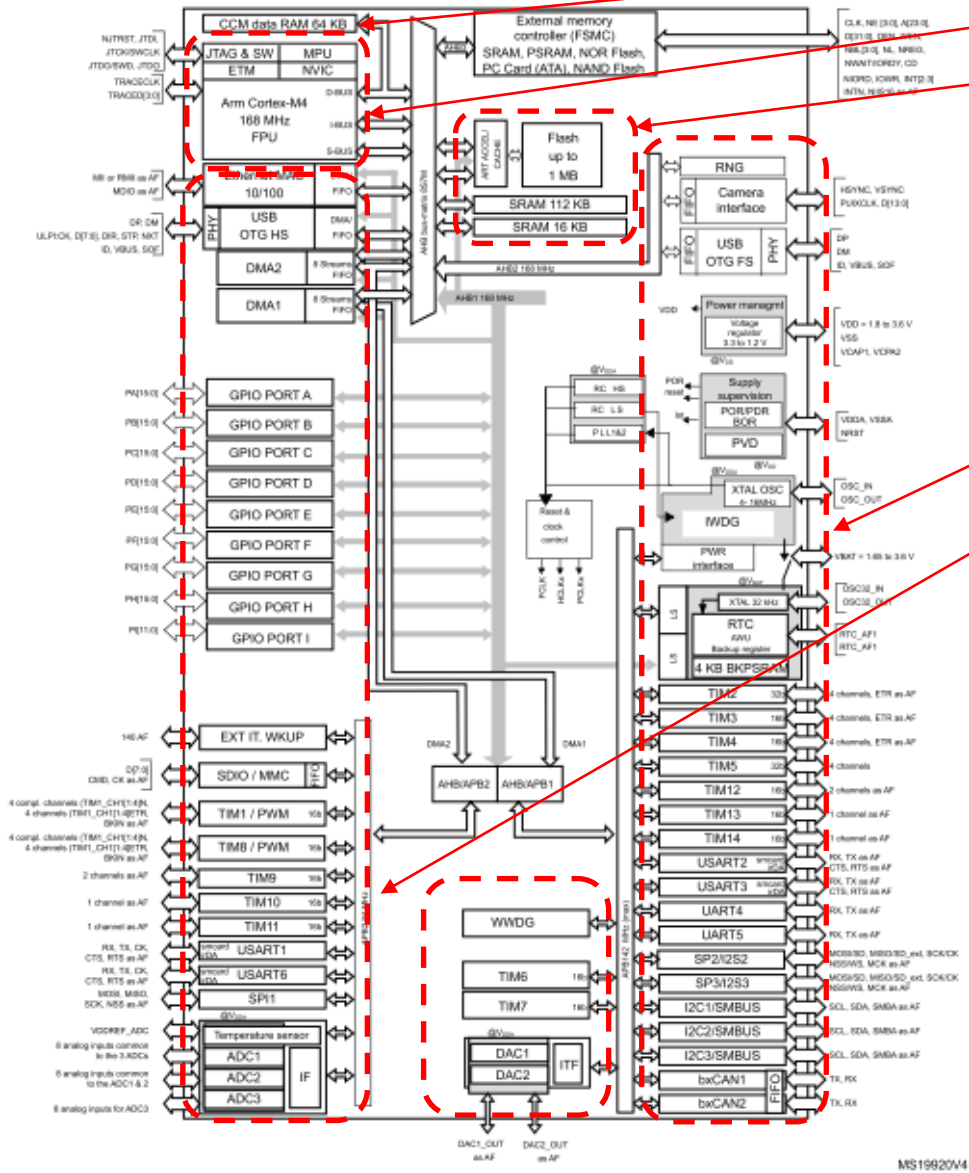
STM32

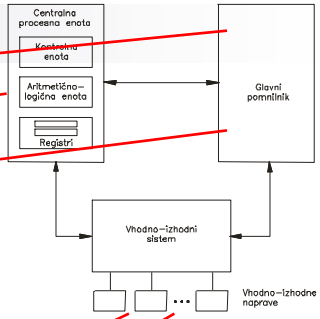
STM32F4DISCOVERY

3.3V !!!



STM32F407VG





Delo na STM32F4 razvojnem sistemu

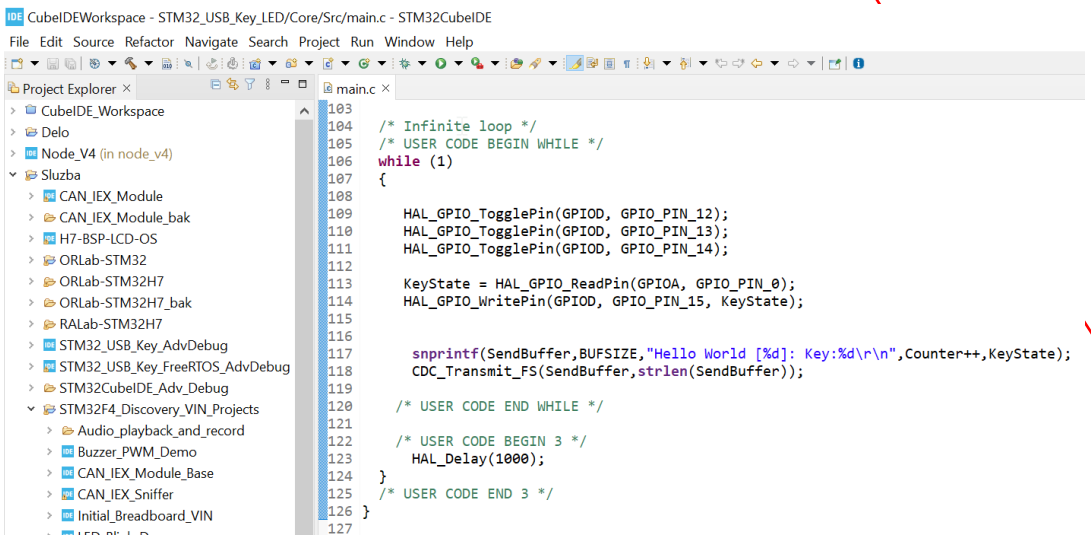
Mikro USB VCom-port

Priključitev :

- **Mini USB** priklop na **krajši stranici**, svetita rdeči **LED** diodi

Poseben začetni projekt za *STM32F4* (e-učilnica) :

- ***dodajanje vsebine (main.c):***



Lastni viri :

https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects

https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples

https://github.com/LAPSyLAB/ORLab-STM32

VIN Projekt – Ostale platforme

STEVAL-MKSBOX1V1 SensorTile.box razvojni sistem

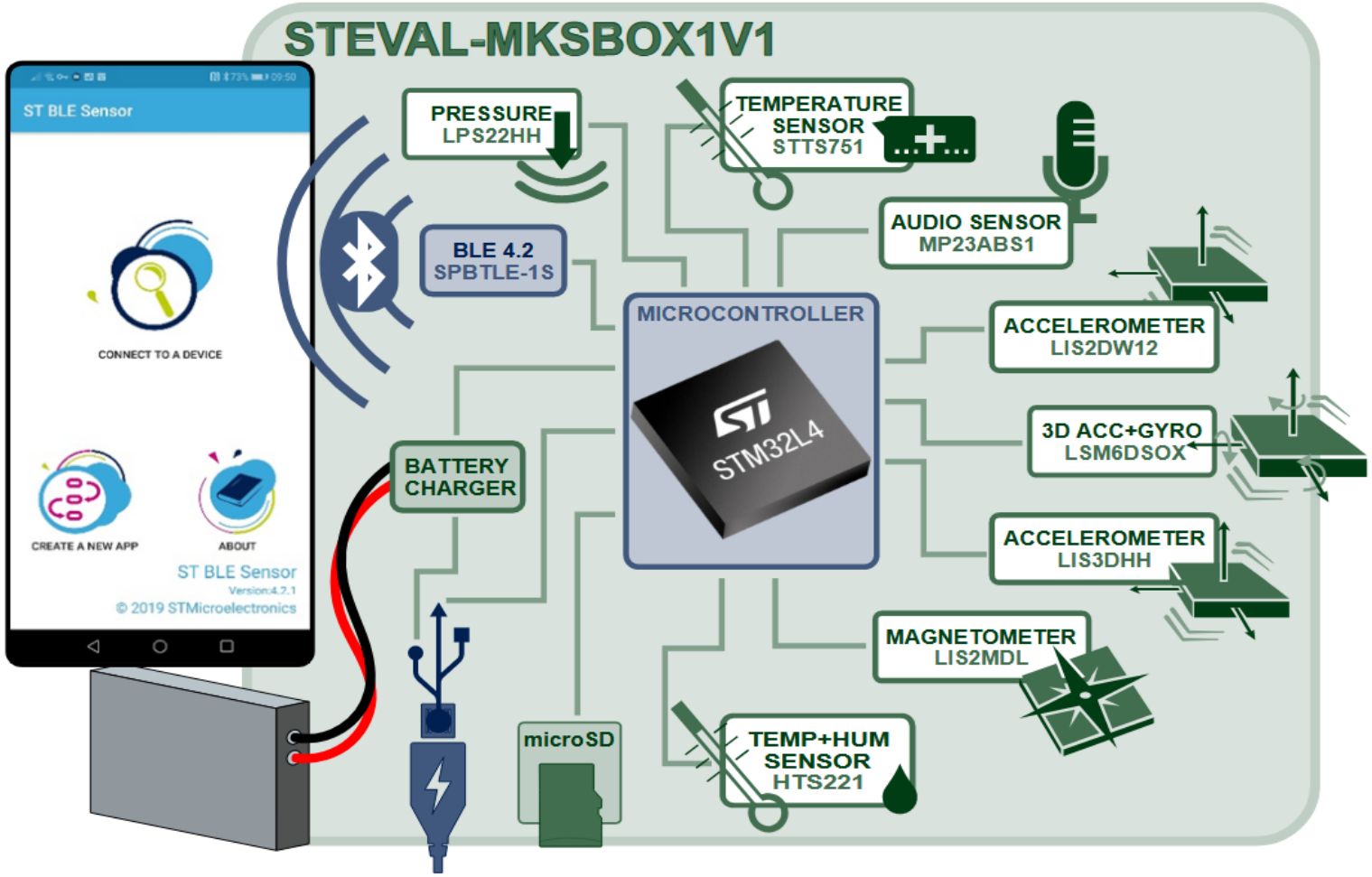
All features

- Easy-to-use app with immediate functionality for the following motion and environmental sensor applications:
  - Pedometer optimized for belt positioning
  - Baby crying detection with Cloud AI learning
  - Barometer / environmental monitoring
  - Vehicle / goods tracking
  - Vibration monitoring
  - Compass and inclinometer
  - Sensor data logger
- Expert Mode with additional sensor app parameter settings
- Compact board with the following high precision sensors:
  - Digital temperature sensor (STTS751)
  - 6-axis inertial measurement unit (LSM6DSOX)
  - 3-axis accelerometers (LIS2DW12 and LIS3DHH)
  - 3-axis magnetometer (LIS2MDL)
  - Altimeter / pressure sensor (LPS22HH)
  - Microphone / audio sensor (MP23ABS1)
  - Humidity sensor (HTS221)
- Ultra-low-power ARM Cortex-M4 microcontroller with DSP and FPU (STM32L4R9)
- Bluetooth application processor v5.2 (BlueNRG-M2) which replaces the SPBTLE-1S Bluetooth Smart connectivity v4.2 module of the board previous batches

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

VIN Projekt – Ostale platforme

STEVAL-MKSBOX1V1 SensorTile.box razvojni sistem

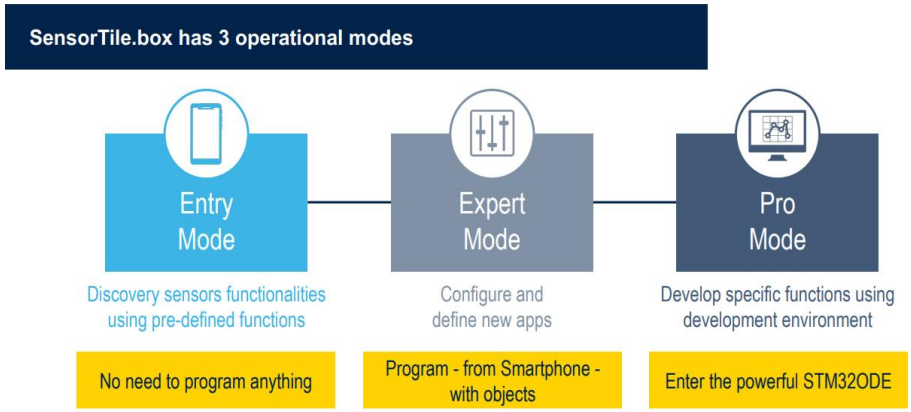


https://www.st.com/en/evaluation-tools/stm32h750b-dk.html

VIN Projekt – Ostale platforme

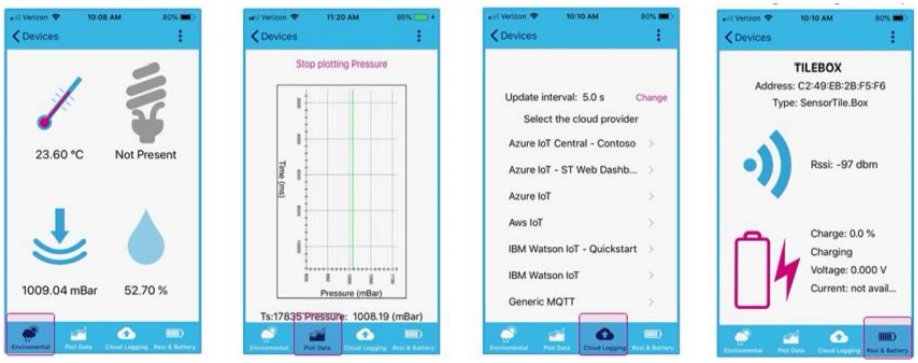
STEVAL-MKSBOX1V1 SensorTile.box razvojni sistem

The IoT made easy



KAJ VSE OMOGOČA APLIKACIJA?

V realnem času s pomočjo senzorjev zaznava stanje v okolju, risanje grafov na podlagi podatkov, shranjevanje podatkov v oblak - cloud logging.



VGRAJENE APLIKACIJE

Primeri vgrajenih aplikacij, dostopnih v Entry Mode načinu aplikacije STE BLE Sensor:

- Preprosta aplikacija za štetje korakov s pomočjo merilnika pospeška
- Zaznavanje otroškega joka
  - zaznavanje otroškega joka preko mikrofona, prižig LED diode/opozorilo na aplikaciji, ko je jok zaznan
- Barometrska aplikacija
  - omogoča dodatno ponastavljanje senzorjev za temperaturo, tlak in vlago, prejete meritve prikaže grafično v obliki
- Sledenje prevažanim dobrinam, stanje v vozilu
  - s primernimi tipali (npr. merilnik vlage) merimo pogoje, katerim bi bili potencialno izpostavljeni izdelki, prevažani v nekem vozilu
- Vibration monitoring
  - določimo neko sprejemljivo meritev tresljajev, ki jih zaznamo z merilnikom pospeška - o morebitnih odstopanjih obvestimo uporabnika in zabeležimo dogodek
- Compass and inclinometer
  - beleženje sprememb v orientaciji SensorTile.box-a tekom časa, grafični prikaz meritev (npr. spremembe v smeri neba, naklon)

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

VIN Projekt – Ostale platforme

STM32F769 ST Discovery

STM Discovery F7 (Cortex M7)

•STM32F769NIH6 microcontroller featuring 2 Mbytes of Flash memory and 512+16+4 Kbytes of RAM, in BGA216 package

•On-board ST-LINK/V2-1 supporting USB reenumeration capability

•USB ST-LINK functions: virtual COM port, mass storage, debug port

•4" capacitive touch LCD display with MIPI® DSI connector (on STM32F769I-DISCO only)

•SAI audio codec

•Two audio line jacks, one for input and one for output

•Stereo speaker outputs

•Four ST MEMS microphones on DFSDM inputs

•Two SPDIF RCA input and output connectors

•Two push-buttons (user and reset)

•512-Mbit Quad-SPI Flash memory

•128-Mbit SDRAM

•Connector for microSD card

•Wi-Fi or Ext-EEP daughterboard connector

•USB OTG HS with Micro-AB connector

•Ethernet connector compliant with IEEE-802.3-2002

•Arduino™Uno V3 connectors

STM32

aws qualified device Internet of things

arm MBED Enabled

https://www.st.com/en/evaluation-tools/32f769idiscovery.html

VIN Projekt – Ostale platforme

ST Discovery STM32MP157C

STM Discovery MP1 (2xCortex A7 + 1xCortex M4)

•STM32MP157 Arm®-based ***dual Cortex®-A7 32 bits + Cortex®-M4 32 bits MPU*** in TFBGA361 package

•4-Gbit DDR3L, 16 bits, 533 MHz

•1-Gbps Ethernet (RGMII) compliant with IEEE-802.3ab

•USB OTG HS

•Audio codec

•4 user LEDs

•2 user and reset push-buttons, 1 wake-up button

•5 V / 3 A USB Type-CTM power supply input (not provided)

•Board connectors:

- •Ethernet RJ454 × USB Host Type-AUSB Type-CTM DRPMIPI DSISMHDMI®Stereo headset jack including analog microphone inputmicroSDTM cardGPIC expansion connector (Raspberry Pi® shields capability)

•ARDUINO® Uno V3 expansion connectors

•STM32CubeMP1 and full mainline open-source Linux® STM32 MPU OpenSTLinux Distribution (such as STM32MP1Starter) software and examples

•4" TFT 480 × 800 pixels with LED backlight, MIPI DSISM interface, and capacitive touch panel

•Wi-Fi® 802.11b/g/n

•Bluetooth® Low Energy 4.1

STM32MP1

STM32

VIN projekt - STM32-CubeIDE projekt, USART, GPIO (LED, tipka), BSP

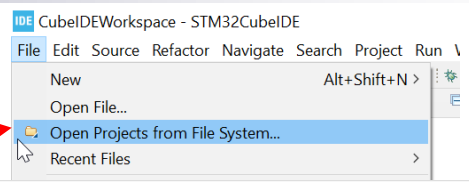
- STM32H7, STM32F4 Discovery board in ostale platforme

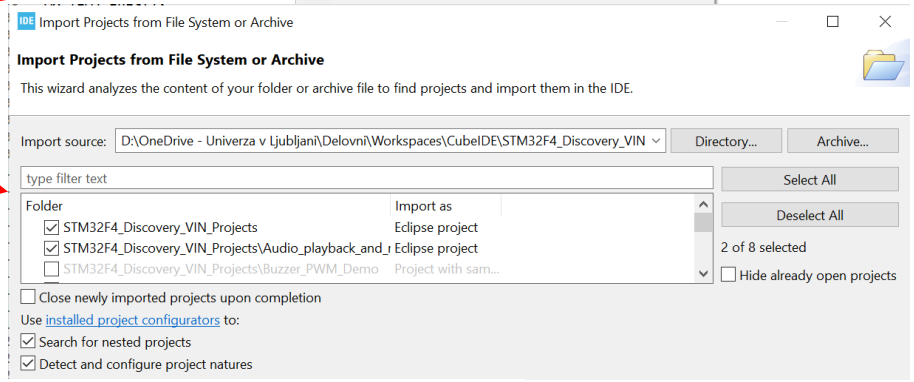
- Osnovna projekta v CubeIDE:
  - CubeMX (HAL knjižnica, BSP) – STM32H7, STM32F4
    - CubeMX – osnovni projekt (GPIO – tipka, LED diode, USART)
    - BSP – osnovni projekt (Touch, LCD)
  - STM32H7:
    - Touch panel (I2C, registri, …)
  - STM32F4:
    - USB VComPort

CubeIDE: delo na STM32 sistemih

Vzpostavitev začetnega projekta :

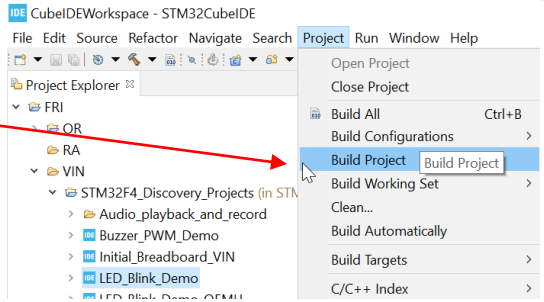
- **Uvoz obstoječega (BSP)**
  - Open projects from File System
  - Select project(s)
- **Nov projekt CubeMX ->**
  (v nadaljevanju)

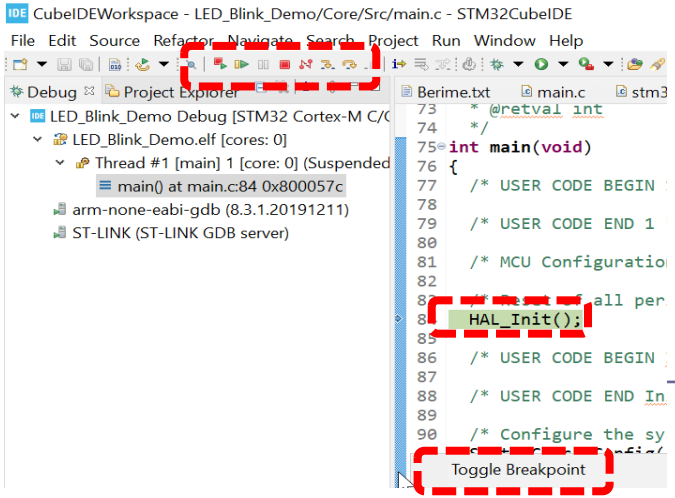


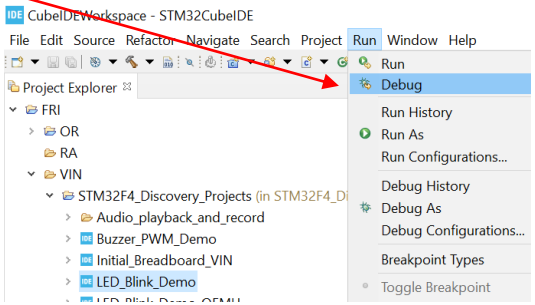


Prevajanje, zagon :

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



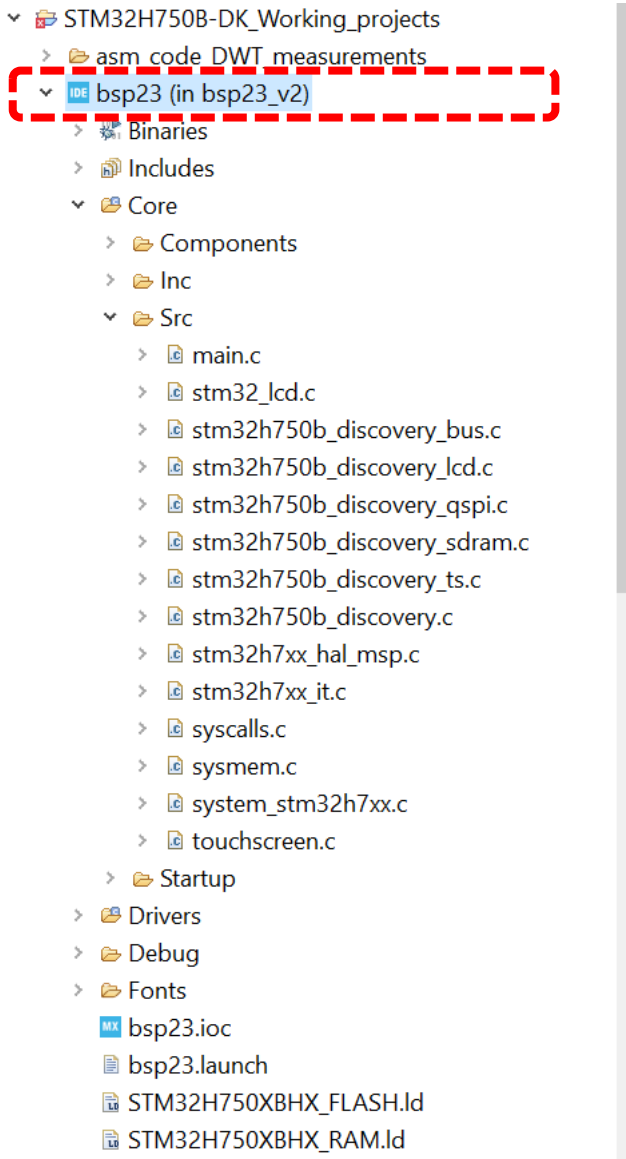




CubeIDE: delo na STM32 sistemih

Kopiranje projekta :

- **Kopiranje CubeIDE projekta z CubeMX .ioc datoteko**
  - 1) Edit > Copy (obstoječi projekt).
  - 2) Edit > Paste (nova lokacija).
  - 3) Preimenuj .ioc datoteko.
  - 4) Zbriši Debug.launch datoteko.
  - 5) Project > Clean.
  - 6) Generiraj kodo s CubeMX.
  - 7) Project > Build Project.
  - 8) Debug As Stm32 Application.
  - 9) Debug aplikacije.
  -
- **Kopiranje osnovnih CubeIDE asm,BSP C projekta**
  - 1) Edit > Copy (obstoječi projekt).
  - 2) Edit > Paste (nova lokacija).
  - 3) Delete the Debug.launch file.
  - 4) Project > Clean.
  - 5) Project > Build Project.
  - 6) Debug As Stm32 Application.
  - 7) And debug the application
  - 8) Add breakpoint on first instruction if necessary



Osnovni projekt CubeIDE – GPIO – različni nivoji programiranja

Baremetal - zbirnik

```
INIT_IO:
  push {r5, r6, lr}
    // Enable GPIOD Peripheral Clock (bit 3 in AHB1ENR register)
    ldr r6, = RCC_AHB1ENR       // Load peripheral clock reg address to r6
    ldr r5, [r6]                // Read its content to r5
    orr r5, 0x00000008          // Set bit 3 to enable GPIOD clock
    str r5, [r6]                // Store result in peripheral clock register

    // Make GPIOD Pin12 as output pin (bits 25:24 in MODER register)
    ldr r6, =GPIOD_BASE       // Load GPIOD BASE address to r6
    ldr r5, [r6,#GPIOD_MODER]  // Read GPIOD_MODER content to r5
    and r5, 0x00FFFFFF          // Clear bits 31-24 for P12-15
    orr r5, 0x55000000          // Write 01 to bits 31-24 for P12-15
    str r5, [r6]                // Store result in GPIOD MODER register
  pop {r5, r6, pc}


LED_ON:
    push {r5, r6, lr}
    // Set GPIOD Pins to 1 (through BSSR register)
    ldr     r6, =GPIOD_BASE       // Load GPIOD BASE address to r6
    mov     r5, #LEDs_ON
    str     r5, [r6,#GPIOD_BSSR] // Write to BSRR register
    pop {r5, r6, pc}

LED_OFF:
    push {r5, r6, lr}
    // Set GPIOD Pins to 0 (through BSSR register)
    ldr     r6, =GPIOD_BASE       // Load GPIOD BASE address to r6
    mov     r5, #LEDs_OFF
    str     r5, [r6,#GPIOD_BSSR] // Write to BSRR register
    pop {r5, r6, pc}
```

https://github.com/LAPSyLAB/ORLab-STM32/tree/main/GPIO_LEDs

RA, OR

Baremetal - C

```
  /* USER CODE BEGIN 2 */


  RCC->AHB1ENR |= 0x08;
// Enable clock for GPIOD
  GPIOD->MODER |= 0x01000000;        //
MODE Register: bit 12 == out

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    GPIOD->ODR ^= 0x1000;            //
Toggle PD12

/* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    for (int i=0; i<0x1000000; i++) {};
// waste some time
  }
  /* USER CODE END 3 */
```

https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/LED_GPIO_C_Baremetal_C

VIN

HAL - C

```
/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {

HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);

  /* USER CODE END WHILE */

   /* USER CODE BEGIN 3 */
 HAL_Delay(1000);
 }
 /* USER CODE END 3 */
```

```
void HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx,
uint16_t GPIO_Pin)
{
  uint32_t odr;

  /* Check the parameters */
  assert_param(IS_GPIO_PIN(GPIO_Pin));

  /* get current Ouput Data Register value
*/
  odr = GPIOx->ODR;

  /* Set selected pins that were at low
level, and reset ones that were high */
  GPIOx->BSRR = ((odr & GPIO_Pin) <<
GPIO_NUMBER) | (~odr & GPIO_Pin);
}
```

https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/LED_Blink_Demo

Spletni viri

Full range of STM32H7 training courses available online

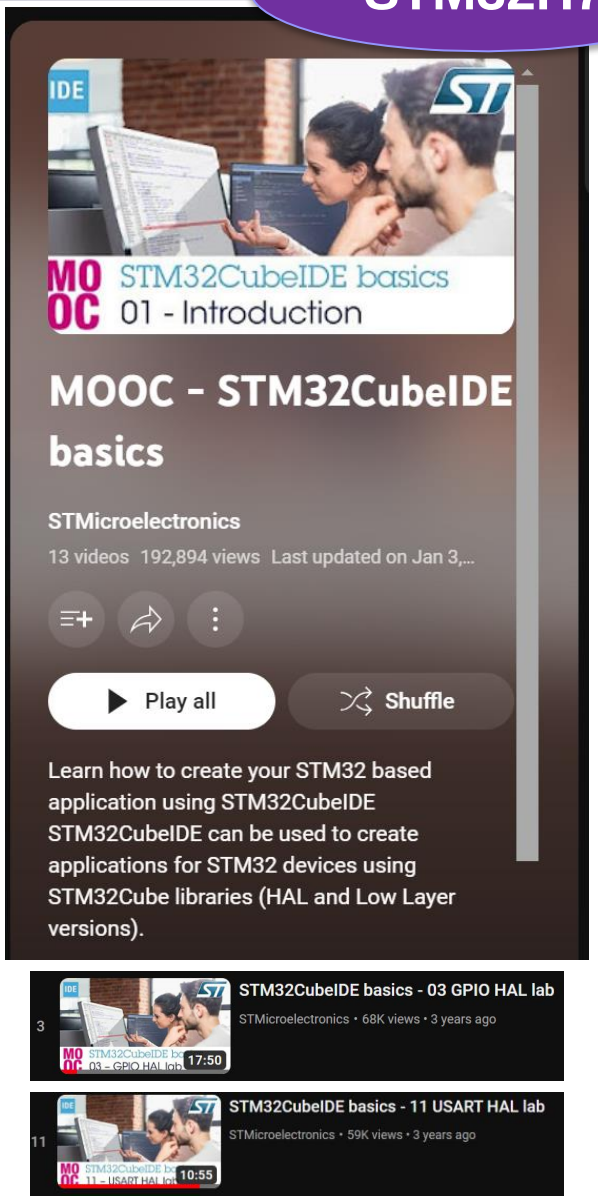
Full range of STM32H7 training courses available online

ST offers a full range of training courses in both **ePresentation** and **PDF** format for the STM32H7 series of High Performance MCUs.
These courses provide helpful instructions and specific information on how to design applications that take advantage of the STM32H7's performance capabilities.
More than 60 specific training modules focus on teaching the skills and knowledge for getting the most performance from STM32H7 MCUs for your applications.

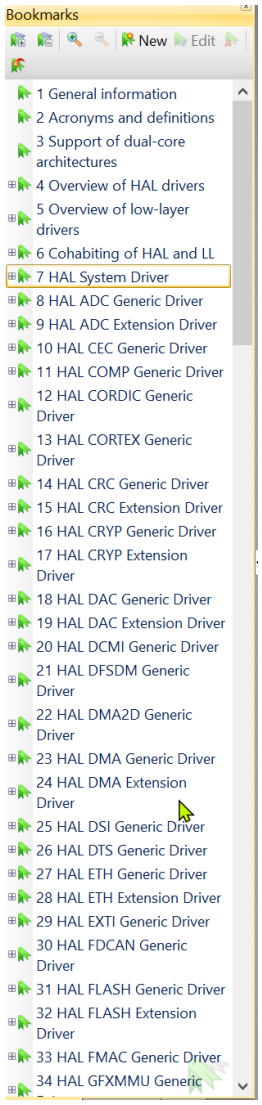
Z naslova <https://www.st.com/content/st_com/en/support/learning/stm32-education/stm32-online-training/stm32h7-online-training.html>

BSP STM32H750B-DK Component

Z naslova <https://github.com/STMicroelectronics/stm32h750b-dk-bsp>



Vir: UM2217 - HAL and Low-layer drivers



Programska knjižnica

Vsebuje predpripravljene funkcije za delo s sistemskimi in V/I napravami.

life.augmented

UM2217

User manual

Description of STM32H7 HAL and low-layer drivers

This section contains the following APIs:

- HAL_Init()
- HAL_DeInit()
- HAL_MspInit()
- HAL_MspDeInit()
- HAL_InitTick()

This section contains the following APIs:

- HAL_USART_Transmit()
- HAL_USART_Receive()
- HAL_USART_TransmitReceive()
- HAL_USART_Transmit_IT()
- HAL_USART_Receive_IT()
- HAL_USART_TransmitReceive_IT()
- HAL_USART_Transmit_DMA()
- HAL_USART_Receive_DMA()
- HAL_USART_TransmitReceive_DMA()

This section contains the following APIs:

- HAL_IncTick()
- HAL_GetTick()
- HAL_GetTickPrio()
- HAL_SetTickFreq()
- HAL_GetTickFreq()
- HAL_Delay()
- HAL_SuspendTick()
- HAL_ResumeTick()
- HAL_GetHalVersion()

35.2.4 IO operation functions

This section contains the following APIs:

- HAL_GPIO_ReadPin()
- HAL_GPIO_WritePin()
- HAL_GPIO_TogglePin()
- HAL_GPIO_LockPin()
- HAL_GPIO_EXTI_IRQHandler()
- HAL_GPIO_EXTI_Callback()

This section contains the following APIs:

- HAL_I2C_Init()
- HAL_I2C_DeInit()
- HAL_I2C_MspInit()
- HAL_I2C_MspDeInit()
- HAL_I2C_RegisterCallback()
- HAL_I2C_UnRegisterCallback()
- HAL_I2C_RegisterAddrCallback()
- HAL_I2C_UnRegisterAddrCallback()

Vir: RM0433 Reference manual

Tehnični opis MCU, vseh naprav

Vsebuje podrobnejše informacije za delo s sistemskimi in V/I napravami.

life.augmented

RM0433
Reference manual

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

11 General-purpose I/Os (GPIO)

11.1 Introduction

Each general-purpose I/O port has four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR), two 32-bit data registers (GPIOx_IDR and GPIOx_ODR) and a 32-bit set/reset register (GPIOx_BSRR). In addition all GPIOs have a 32-bit locking register (GPIOx_LCKR) and two 32-bit alternate function selection registers (GPIOx_AFRH and GPIOx_AFRL).

Delo na STM32 razvojnih sistemih

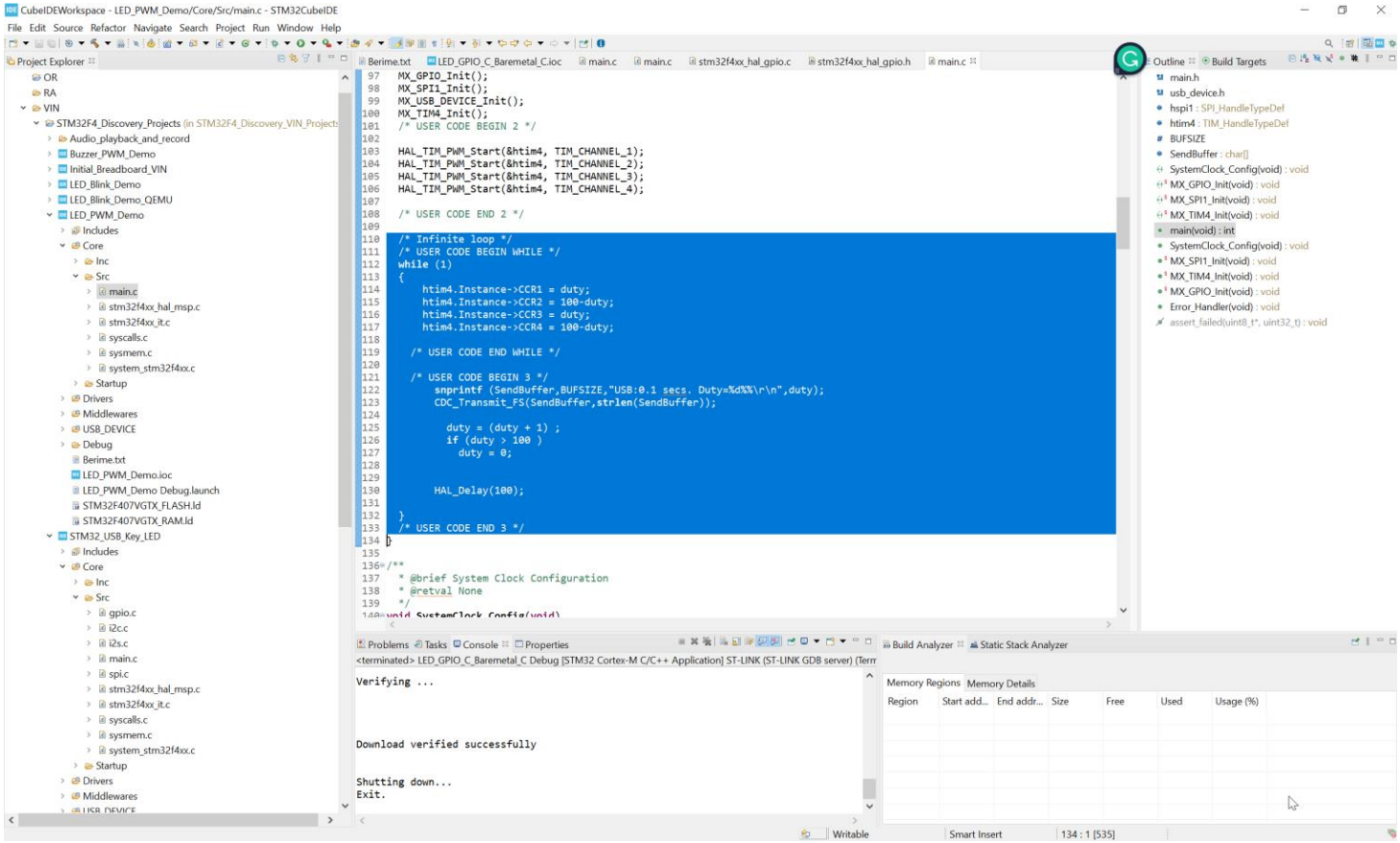
Mikro USB priklop na daljši stranici (srednji !!!)

Priključitev :

- **Mini USB** priklop na **krajši stranici**, svetita rdeči **LED** diodi

STM32 CubeIDE

- https://www.st.com/en/development-tools/stm32cubeide.html



Mini USB

Mikro USB VCom-port

VIN projekt - STM32-CubeIDE projekt, USART, GPIO (LED, tipka), BSP

- STM32H7, STM32F4 Discovery board in ostale platforme

- Osnovna projekta v CubeIDE:
  - CubeMX (HAL knjižnica, BSP) – STM32H7, STM32F4
    - CubeMX – osnovni projekt (GPIO – tipka, LED diode, USART)
    - BSP – osnovni projekt (Touch, LCD)
  - STM32H7:
    - Touch panel (I2C, registri, …)
  - STM32F4:
    - USB VComPort

Delo na STM32H7 razvojnem sistemu

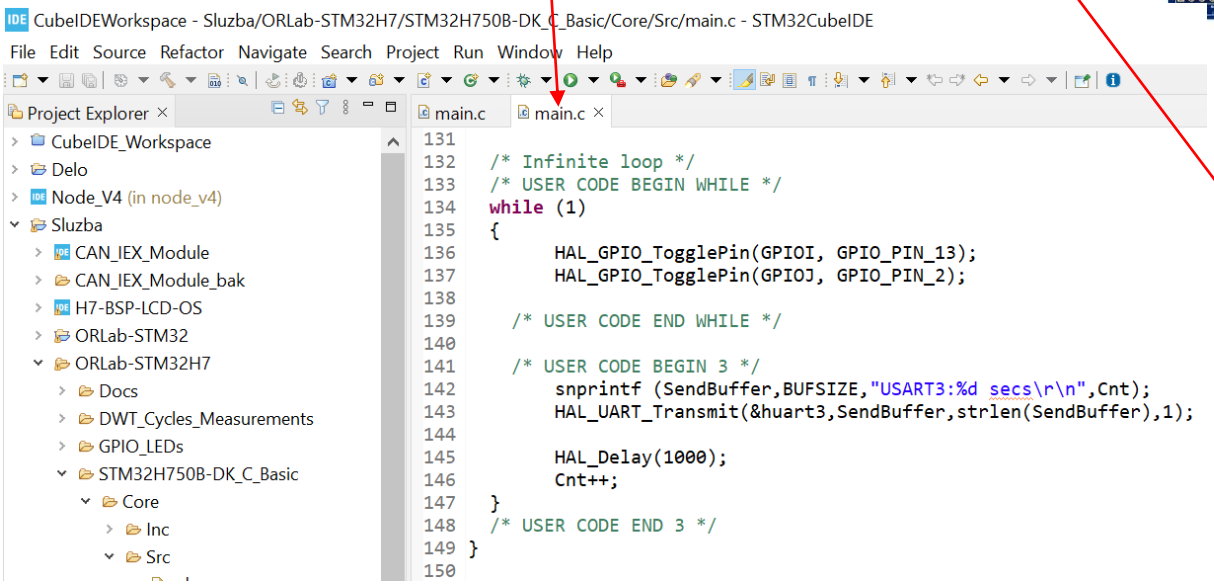
Mikro USB priklop na daljši stranici (srednji !!!)

Priključitev :

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

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

- ***dodajanje vsebine (main.c):***



Lastni viri :

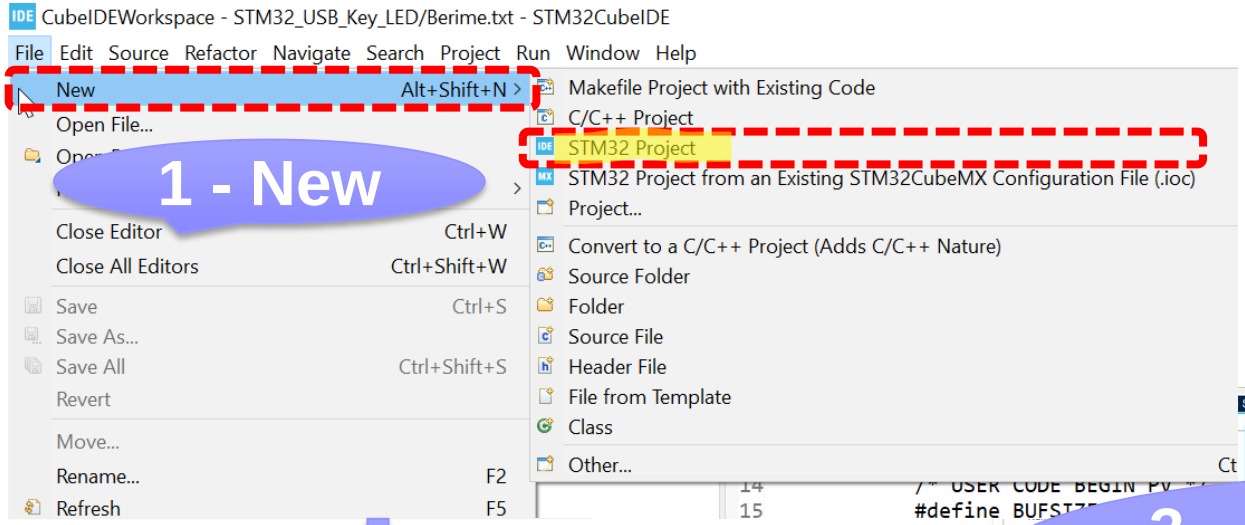
https://github.com/LAPSyLAB/STM32H7_Discovery_VIN_Projects

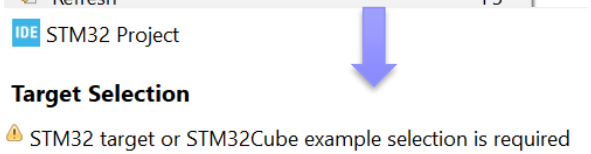
https://github.com/LAPSyLAB/ORLab-STM32H7

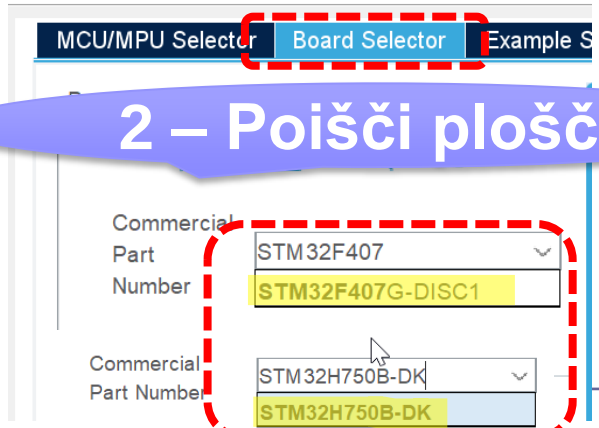
STM32H7

CubeIDE – Vzpostavitev novega projekta s CubeMX

Nov projekt :

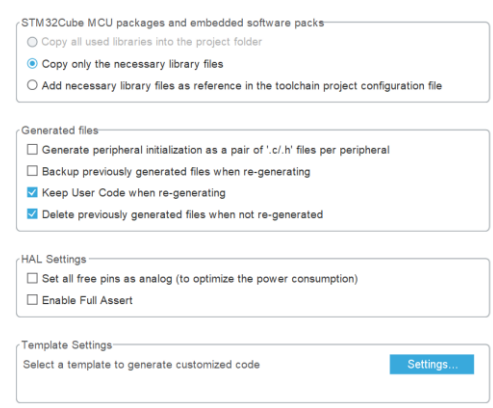


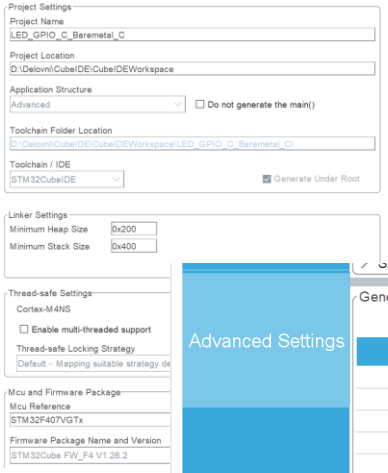


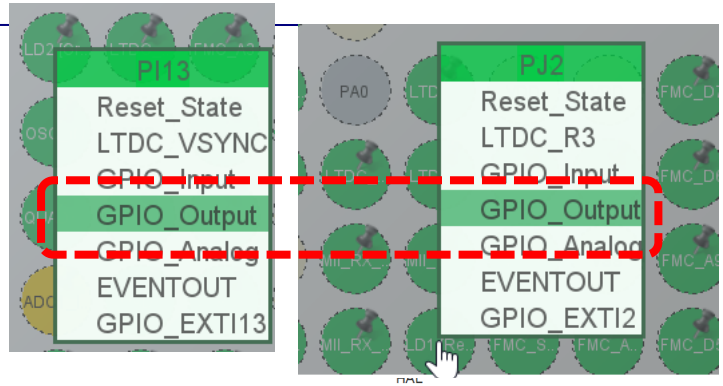


Osnovni projekt CubeIDE – CubeMX

Konfiguracija : priključki, knjižnice STM32H7







Advanced Settings

Generated Function Calls

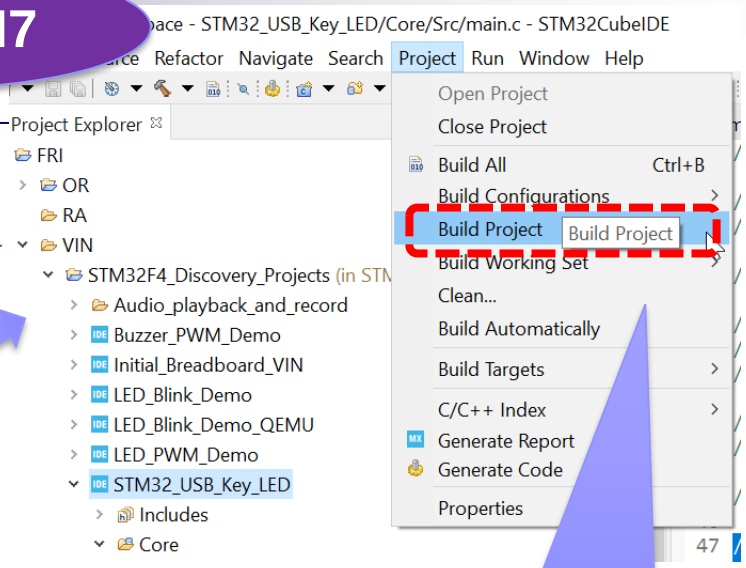
Generate Code	Rank	Function Name	Peripheral Instance Name	Do Not Generate Function Call
☑	1	SystemClock_Config	RCC	☐
☑	2	MX_GPIO_Init	GPIO	☐
☑	3	MX_ADC1_Init	ADC1	☐
☑	4	MX_ADC2_Init	ADC2	☐
☑	5	MX_ADC3_Init	ADC3	☐
☑	6	MX_ETH_Init	ETH	☐
☑	7	MX_FDCAN1_Init	FDCAN1	☐
☑	8	MX_FDCAN2_Init	FDCAN2	☐
☑	9	MX_FMC_Init	FMC	☐
☑	10	MX_LTDC_Init	LTDC	☐
☑	11	MX_QUADSPI_Init	QUADSPI	☐
☑	12	MX_RTC_Init	RTC	☐
☑	13	MX_SAI2_Init	SAI2	☐
☑	14	MX_SDMMC1_MMC_Init	SDMMC1	☑

4 – Preveri nastavitve

Osnovni projekt CubeMX – USB Virtual COM

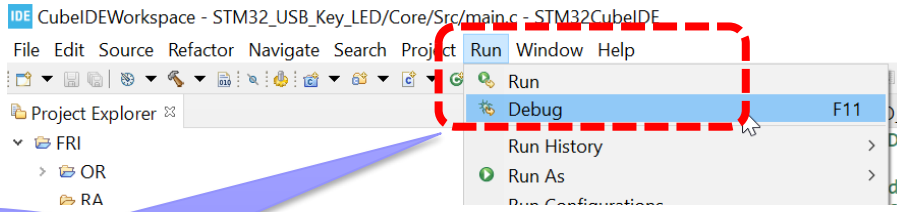
Program : za pošiljanje po USB Virtual COM Port (USART3)

```
    /* Private variables ---------------------------------------------------------*/

    /* USER CODE BEGIN PV */
    #define     BUFSIZE 256
    char        SendBuffer[BUFSIZE];
    int         Counter;
    /* USER CODE END PV */


/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {
   /* USER CODE END WHILE */

   /* USER CODE BEGIN 3 */
   snprintf (SendBuffer,BUFSIZE,"USART3:%d secs\r\n",Counter);
   HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),100);

   HAL_Delay(1000);
   Counter++;

 }
 /* USER CODE END 3 */
```



6 – Build project

5 – UART koda

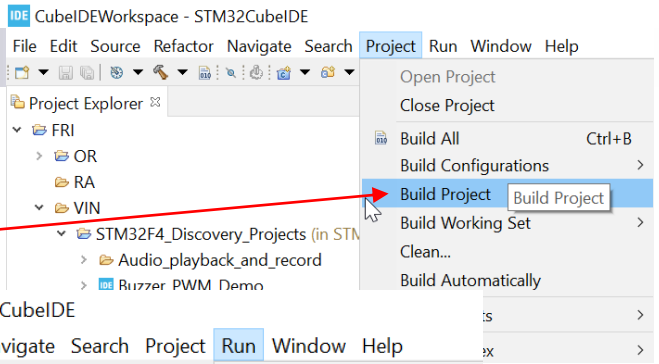


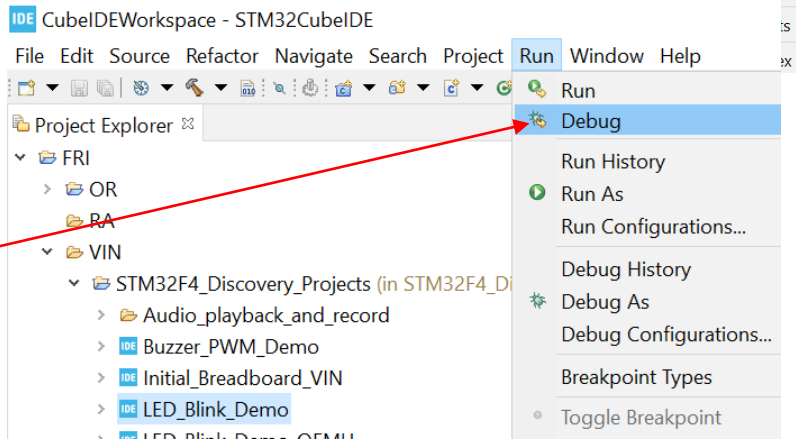
7 – Debug project

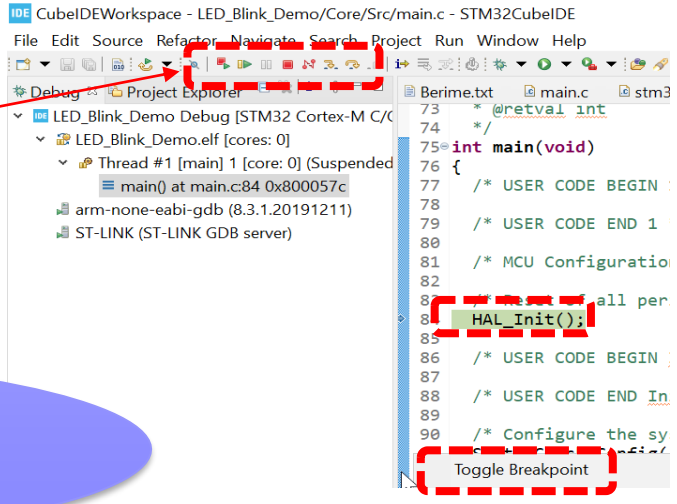
CubeIDE – Zagon, debug

Prevajanje, zagon :

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



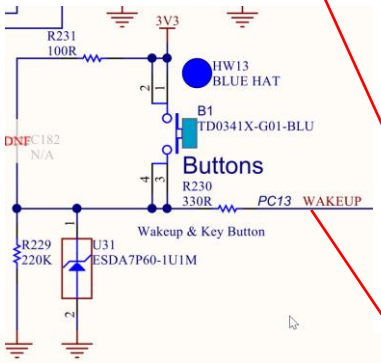




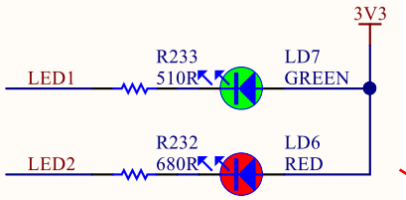
Build <->Debug project, …

STM32H7

GPIO Krmilnik



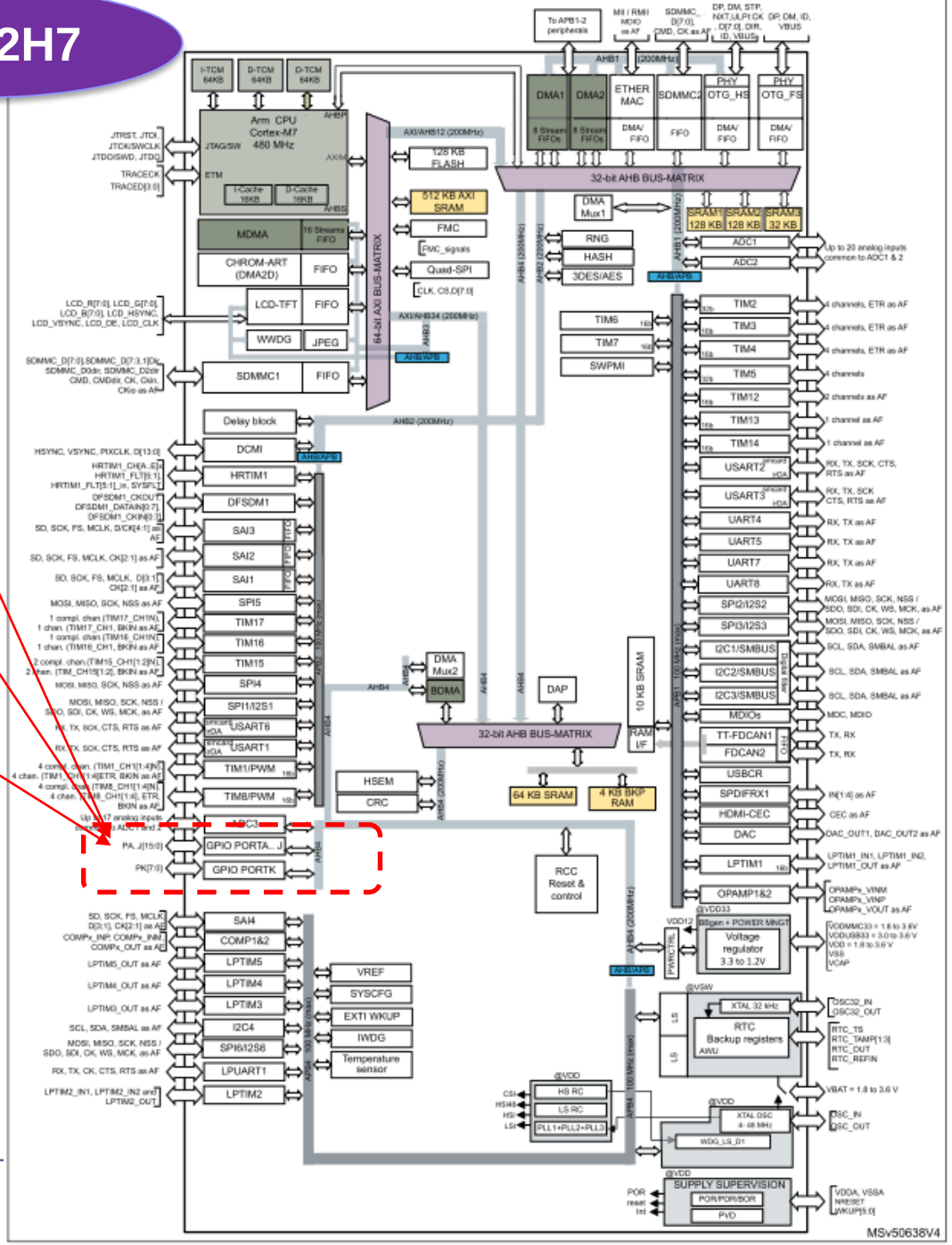
LEDs



U6B
STM32H750XBH6

T8	PG0	PI0	A16	LCD_G5
U8	PG1	PI1	A15	LCD_G6
H16	PG2	PI2	B15	ARD_D12
H15	PG3	PI3	C14	STMOD#8-MOSIs
H14	PG4	PI4	A4	SAI2_MCLKA
G14	PG5	PI5	A3	SAI2_SCKA
G15	PG6	PI6	A2	SAI2_SDA
F16	PG7	PI7	B3	SAI2_FSA
F15	PG8	PI8	E4	ARD_D7
A10	PG9	PI9	E2	LCD_VSYNC
A9	PG10	PI10	F3	MII_RX_ER
B9	PG11	PI11	F4	STMOD#18
C9	PG12	PI12	H1	LCD_HSYNC
D9	PG13	PI13	H2	LED2
D8	PG14	PI14	H3	LCD_CLK
D6	PG15	PI15	P5	LCD_R0
	PJ0		P6	LCD_R2
	PJ1		T6	LED1
	PJ2		U6	LCD_R4

LED: rdeča PI13, zelena PJ2



Osnovni projekt CubeIDE – GPIO – tipka, LED diode

HAL - C

```
/* USER CODE BEGIN PV */
#define BUFSIZE 256
char SendBuffer[BUFSIZE];
int Counter;
int KeyState=0;

/* USER CODE END PV */
```

```
/* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {

    HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_13);

    KeyState = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13);
    HAL_GPIO_WritePin(GPIOJ, GPIO_PIN_2, KeyState);


    snprintf(SendBuffer,BUFSIZE,"Hello World [%d]: Key:%d\r\n",Counter++,KeyState);
    HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),100);

    HAL_Delay(1000);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
```

5 – GPIO

6 – USART COM Port

UM2217

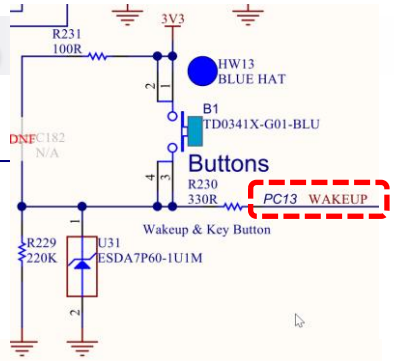
User manual

Description of STM32H7 HAL and low-layer drivers

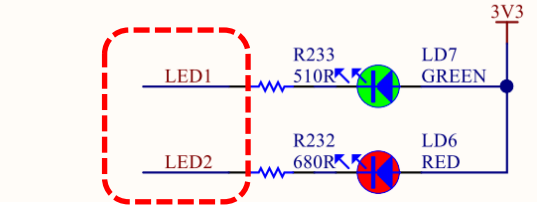
35.2.4 **IO operation functions**

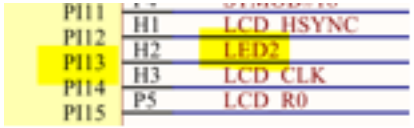
This section contains the following APIs:

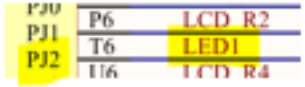
- *HAL_GPIO_ReadPin()*
- *HAL_GPIO_WritePin()*
- *HAL_GPIO_TogglePin()*
- *HAL_GPIO_LockPin()*
- *HAL_GPIO_EXTI_IRQHandler()*
- *HAL_GPIO_EXTI_Callback()*



LEDs





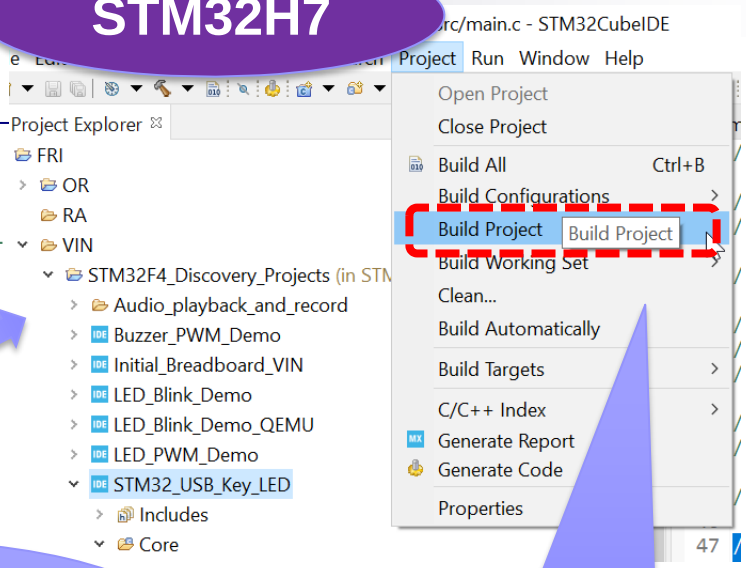


Osnovni projekt CubeIDE – USB Virtual COM Port

STM32H7

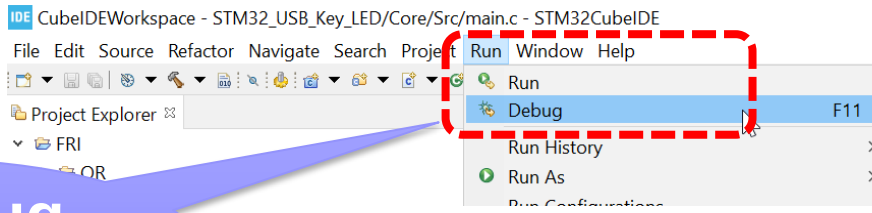
Program : za pošiljanje po USB Virtual COM Port

```
/* Private variables ---------------------------------------------------------

/* USER CODE BEGIN PV */
#define BUFSIZE 256
char SendBuffer[BUFSIZE];
int Counter;
int KeyState=0;

/* USER CODE END PV */

/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {

   HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_13);

   KeyState = HAL_GPIO_ReadPin(GPIOC      _PIN_13);
   HAL_GPIO_WritePin(GPIOJ, GPIO      _2, KeyState);


   snprintf(SendBuffer,B    SIZE,"Hello World [%d]: Key:%d\r\n",Counter++,KeyState);
   HAL_UART_Transmit(    huart3,SendBuffer,strlen(SendBuffer),100);

   HAL_Delay(1000);
   /* USER CODE END WHILE */

   /* USER CODE BEGIN 3 */
  }
 /* USER CODE END 3 */
```



7 – Delay

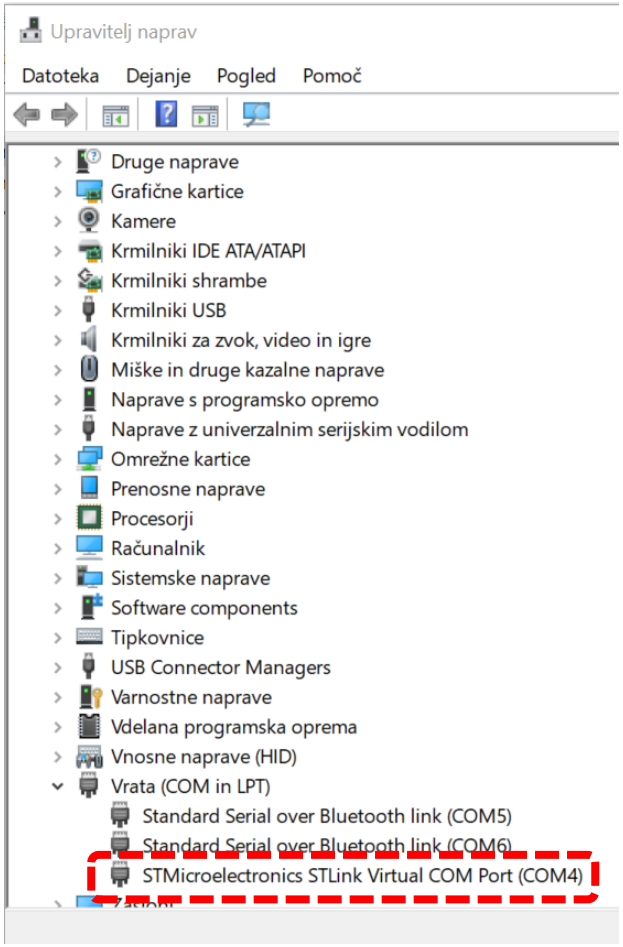
8 – Build project



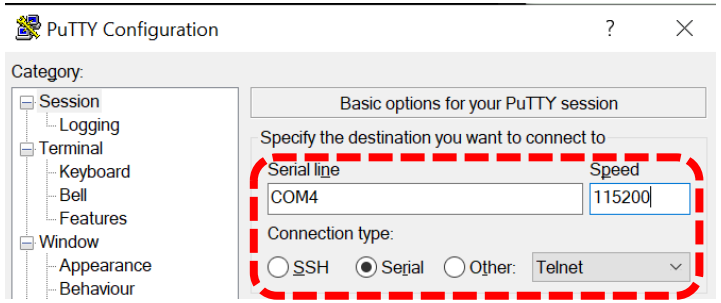
9 – Debug project

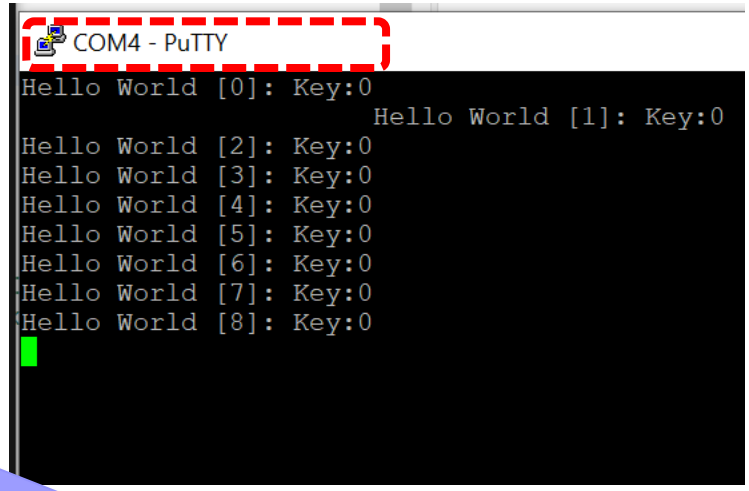
Osnovni projekt CubeIDE – USB Virtual COM Port (USART3 na STM strani)

Program : sprejem na PC strani (povezava z Micro-USB kablom)



https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe





10 – Test project

VIN projekt - STM32-CubeIDE projekt, USART, GPIO (LED, tipka), BSP

- STM32H7, STM32F4 Discovery board in ostale platforme

- Osnovna projekta v CubeIDE:
  - CubeMX (HAL knjižnica, BSP) – STM32H7, STM32F4
    - CubeMX – osnovni projekt (GPIO – tipka, LED diode, USART)
    - BSP – osnovni projekt (Touch, LCD)
  - STM32H7:
    - Touch panel (I2C, registri, …)
  - STM32F4:
    - USB VComPort

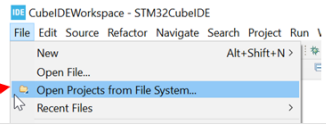
BSP projekt CubeIDE – Touch, LCD screen demo

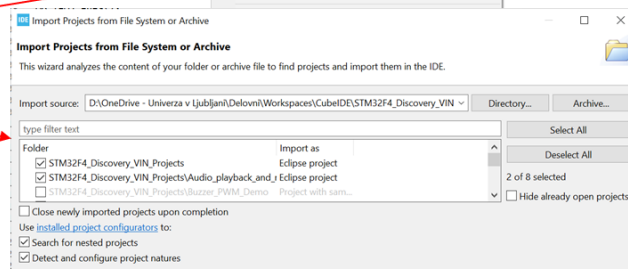
Program : Touch Screen Demo (I2C4)

```
int main(void)
{
/* Enable the CPU Cache */
CPU_CACHE_Enable();

/* STM32H7xx HAL library initialization:*/
HAL_Init();

/* Configure the system clock to 400 MHz */
SystemClock_Config();

BSP_LCD_Init(0, LCD_ORIENTATION_LANDSCAPE);
UTIL_LCD_SetFuncDriver(&LCD_Driver);

Touchscreen_demo();
/* Wait For User inputs */
```

Vzpostavitev začetnega projekta :

- **Uvoz obstoječega**
  - Open projects from File System
  - Select project(s)

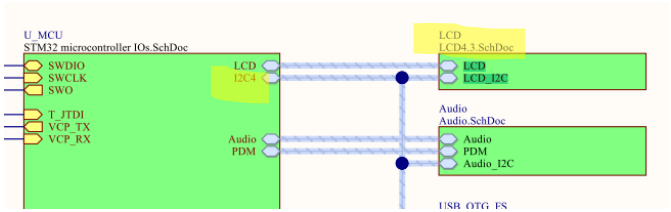


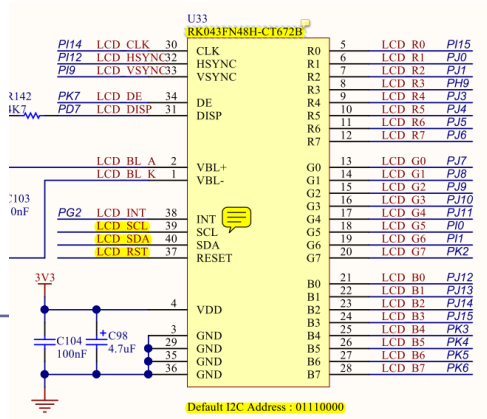



```
void Touchscreen_demo(void)
{
uint16_t x1, y1;
uint8_t state = 0;
uint32_t ts_status = BSP_ERROR_NONE;
uint32_t x_size, y_size;


BSP_LCD_GetXSize(0, &x_size);
BSP_LCD_GetYSize(0, &y_size);


hTS.Width = x_size;
hTS.Height = y_size;
hTS.Orientation =TS_SWAP_XY ;
hTS.Accuracy = 5;


/* Touchscreen initialization */
ts_status = BSP_TS_Init(0, &hTS);
```

```
int32_t BSP_TS_Init(uint32_t Instance, TS_Init_t *TS_Init)
{
int32_t ret = BSP_ERROR_NONE;


if((Instance >=TS_INSTANCES_NBR) || (TS_Init->Width == 0U) ||(
TS_Init->Width > TS_MAX_WIDTH) ||\
(TS_Init->Height == 0U) ||( TS_Init->Height > TS_MAX_HEIGHT) ||\
(TS_Init->Accuracy > TS_MIN((TS_Init->Width), (TS_Init->Height))))
{
ret = BSP_ERROR_WRONG_PARAM;
}
else
{
if(FT5336_Probe(Instance) != BSP_ERROR_NONE)
```

```
int32_t BSP_I2C4_ReadReg(uint16_t DevAddr, uint16_t Reg,
uint8_t *pData, uint16_t Length)
{
int32_t ret;

if(I2C4_ReadReg(DevAddr, Reg, I2C_MEMADD_SIZE_8BIT,
pData, Length) == 0)
{
ret = BSP_ERROR_NONE;
}else{
if( HAL_I2C_GetError(&hbus_i2c4) == HAL_I2C_ERROR_AF){
ret = BSP_ERROR_BUS_ACKNOWLEDGE_FAILURE;
}else{
ret = BSP_ERROR_PERIPH_FAILURE;
}
}
return ret;
}
```





VIN projekt - STM32-CubeIDE projekt, USART, GPIO (LED, tipka), BSP

- STM32H7, STM32F4 Discovery board in ostale platforme

- Osnovni projekt CubeIDE in CubeMX
  - CubeMX (HAL knjižnica, BSP) – STM32H7, STM32F4
    - GPIO – nivoji programiranja
    - GPIO – tipka, LED diode
  - STM32H7:
    - Touch panel (I2C, registri, …)
  - STM32F4:
    - USB VComPort

Delo na STM32F4 razvojnem sistemu

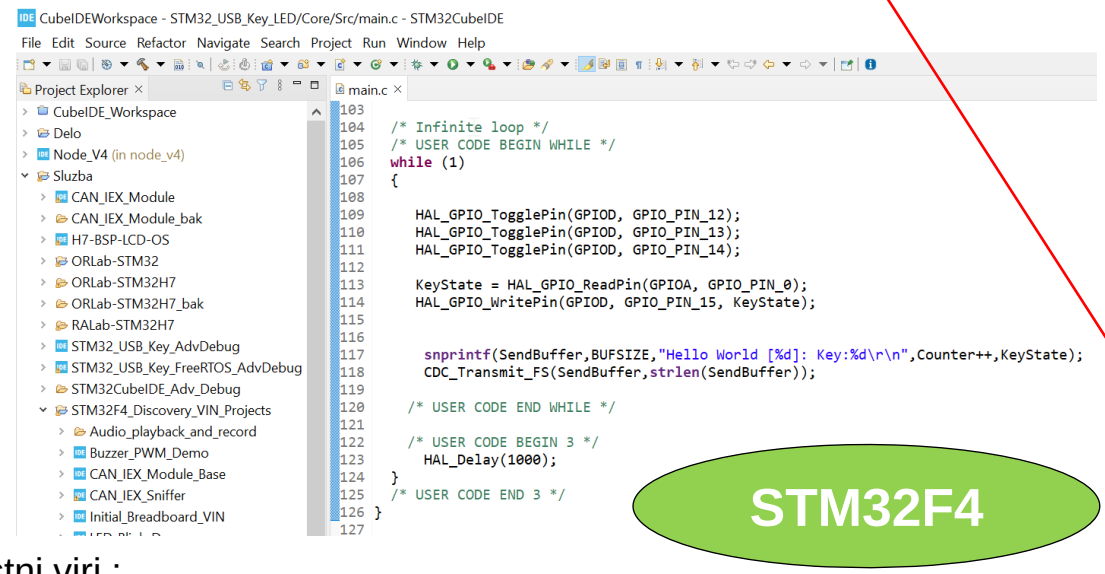
Priključitev :

- **Mini USB** priklop na **krajši stranici**, svetita rdeči **LED** diodi

Mikro USB VCom-port

Poseben začetni projekt za *STM32F4* (e-učilnica) :

- ***dodajanje vsebine (main.c):***



STM32F4

Lastni viri :

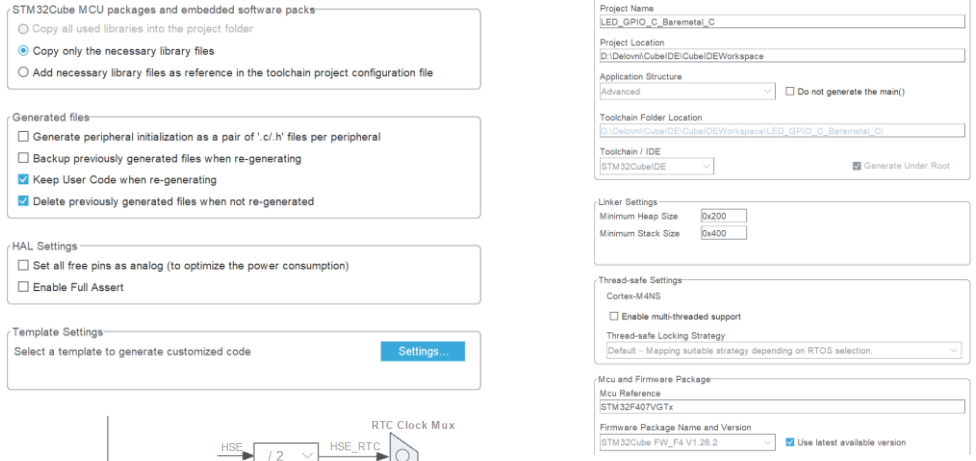
https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects

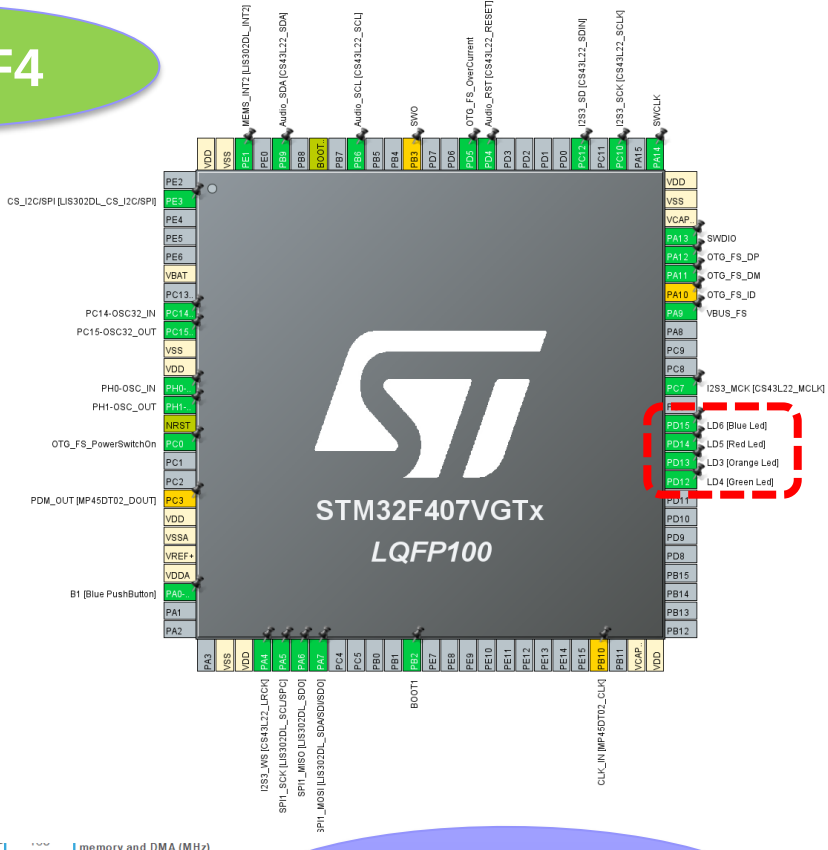
https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples

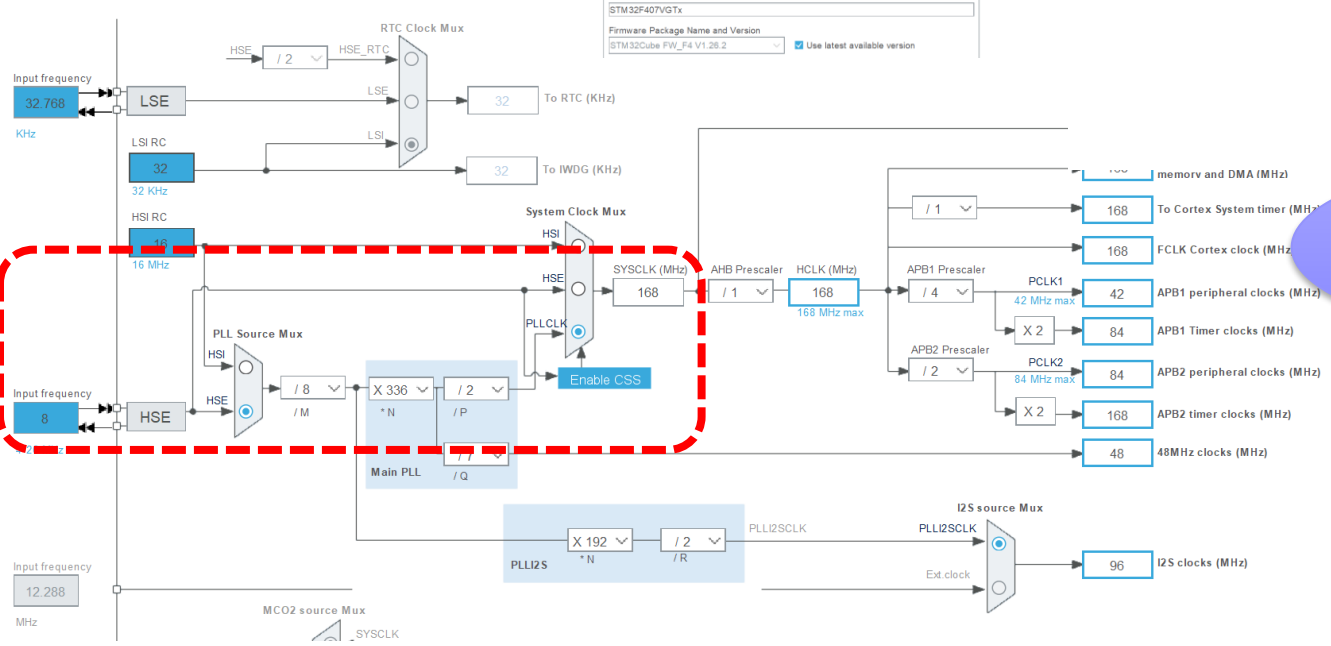
https://github.com/LAPSyLAB/ORLab-STM32

Osnovni projekt CubeIDE – CubeMX

Konfiguracija : priključki, knjižnice STM32F4



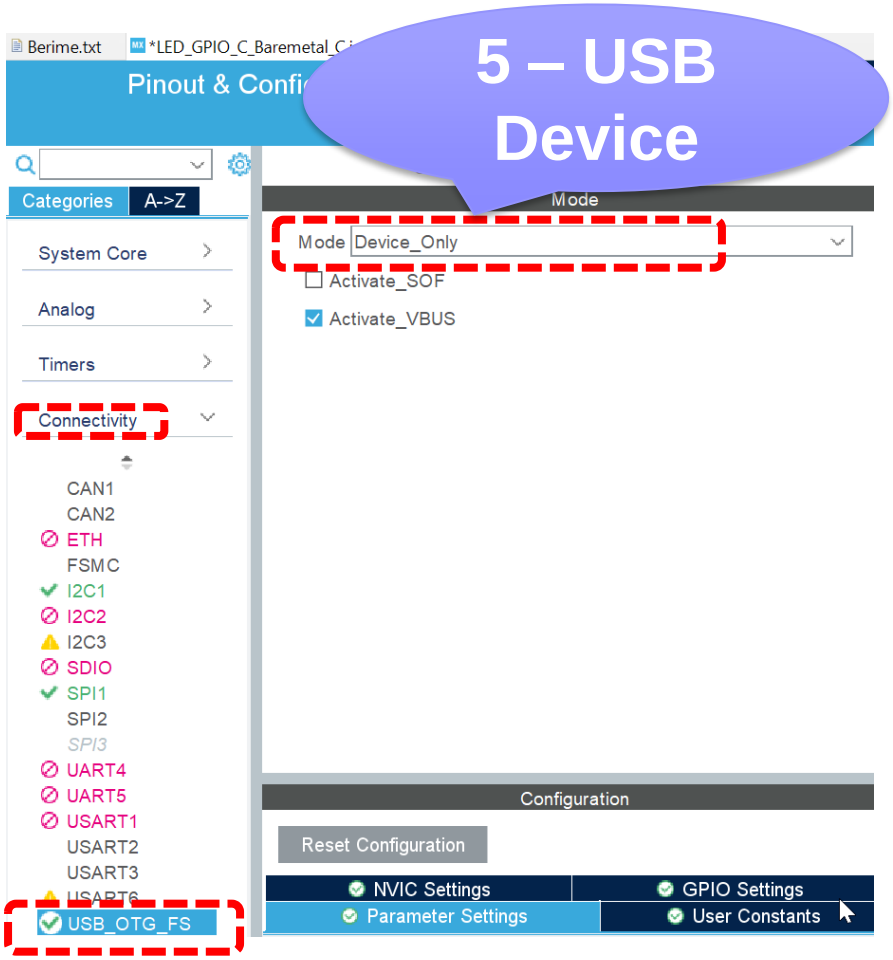


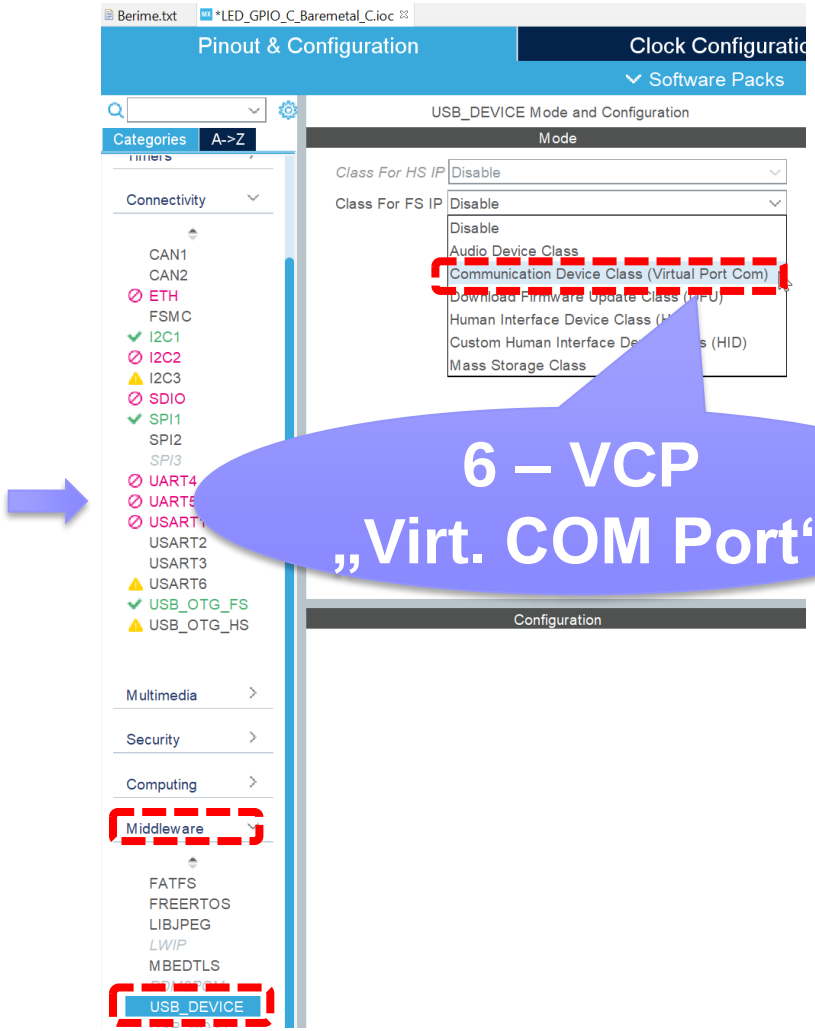


4 – Preveri nastavitve

Osnovni projekt CubeIDE – USB Virtual COM Port

Konfiguracija : USB Device,CDC Class = Virtual COM Port



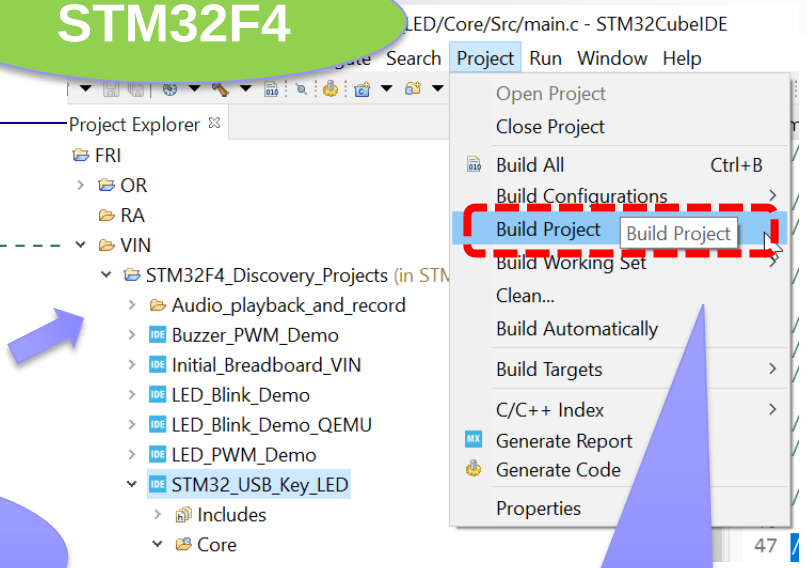


STM32F4

Program : za pošiljanje po USB Virtual COM Port

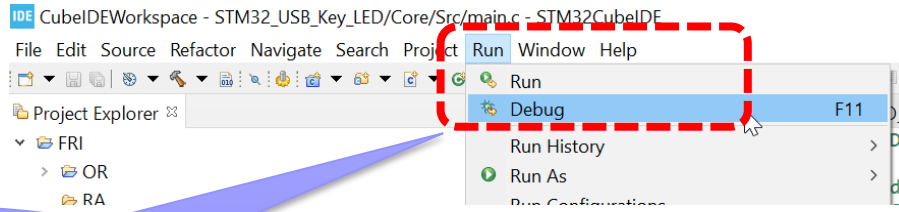
```
/* Private variables ---------------------------------------------------------*/

/* USER CODE BEGIN PV */
#define     BUFSIZE 256
char        SendBuffer[BUFSIZE];
int         Counter;
/* USER CODE END PV */
```

7 – USB VCP koda



8 – Build project

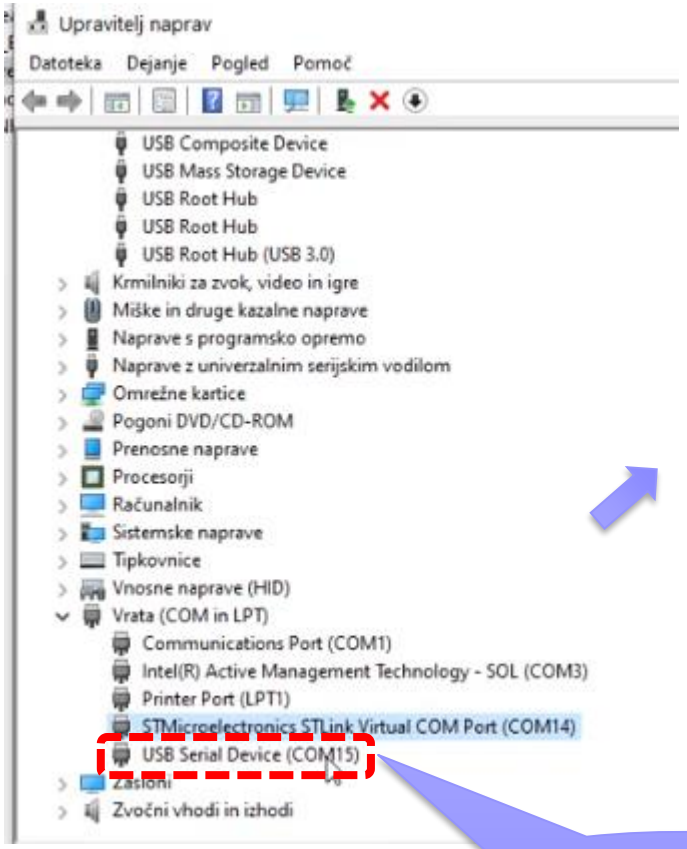
```
/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {
         snprintf(SendBuffer,BUFSIZE,"Hello World [%d]\r\n",Counter++);
         CDC_Transmit_FS(SendBuffer,strlen(SendBuffer));
 /* USER CODE END WHILE */

 /* USER CODE BEGIN 3 */
         HAL_Delay(1000);
 }
 /* USER CODE END 3 */
```

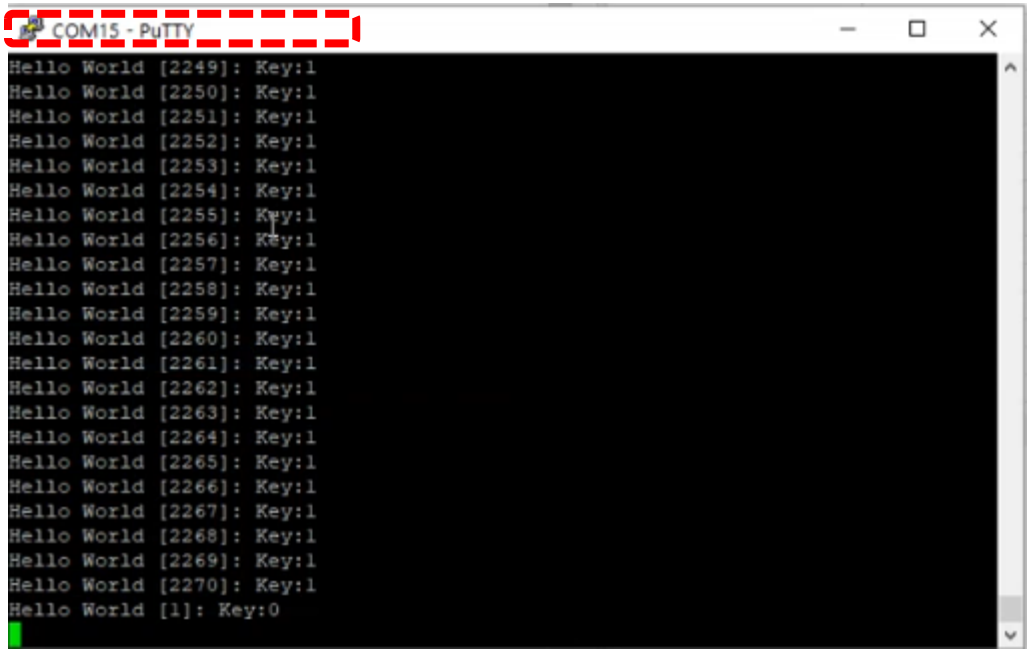


9 – Debug project

Osnovni projekt CubeIDE – USB Virtual COM Port

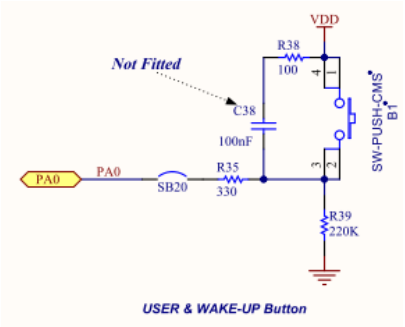
Program : sprejem na PC strani (povezava z Micro-USB kablom)

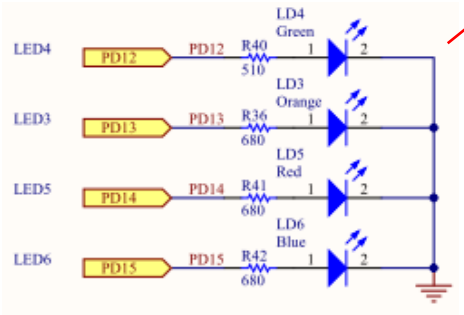


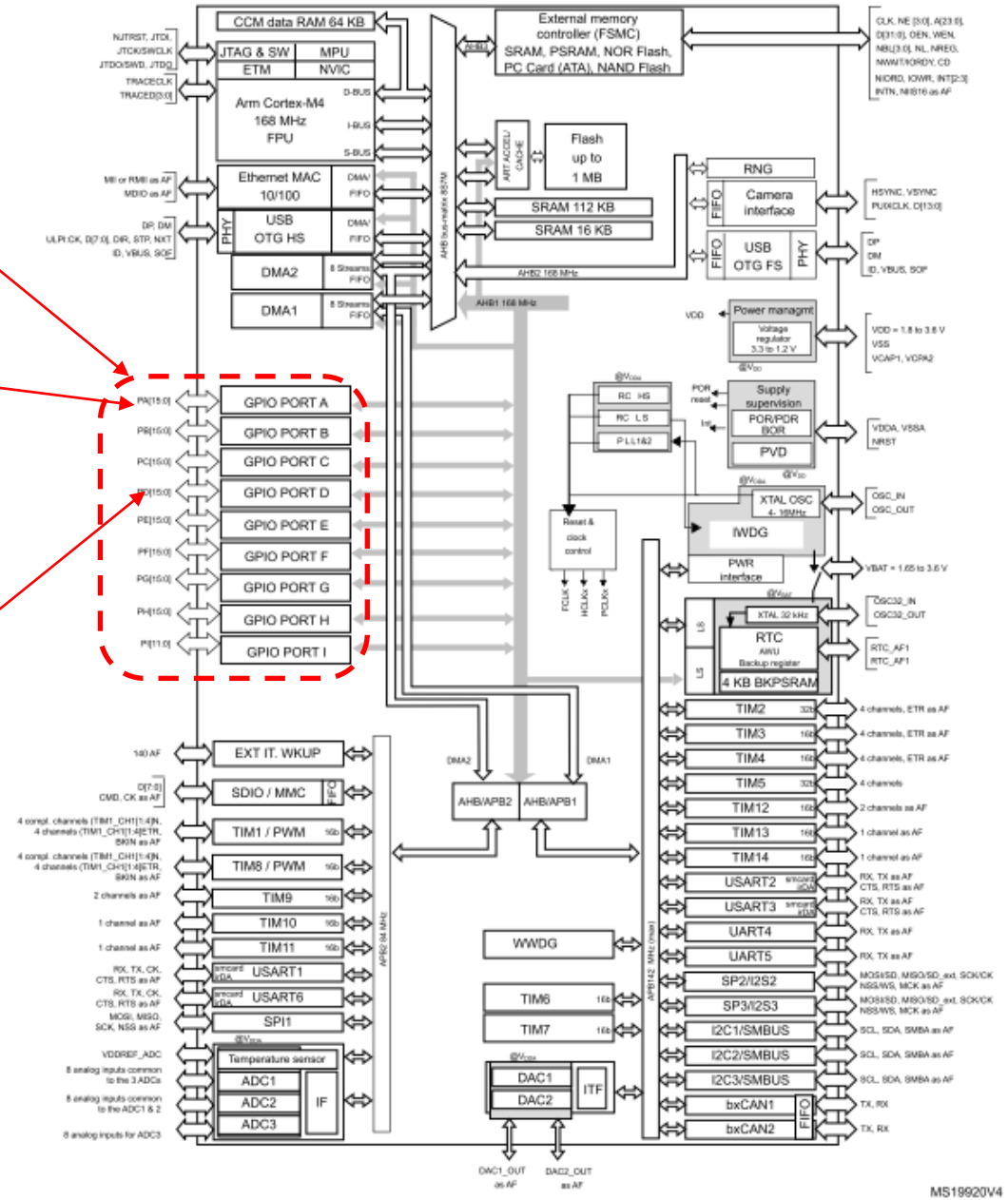


10 – Test project

GPIO Krmilnik



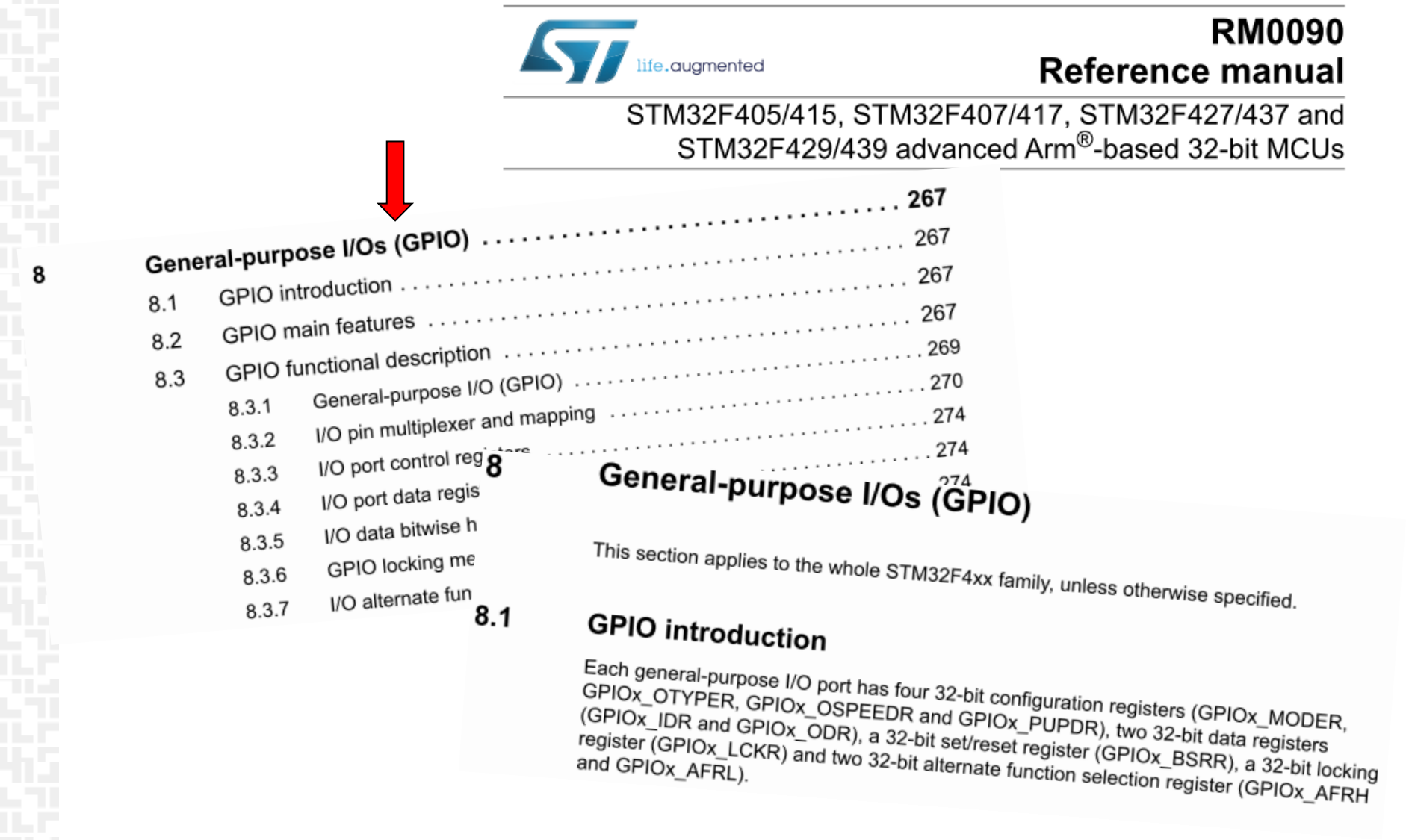




STM32F4

Vir: RM0090 Reference manual

STM32F4



life.augmented

RM0090
Reference manual

STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs

8 General-purpose I/Os (GPIO)

This section applies to the whole STM32F4xx family, unless otherwise specified.

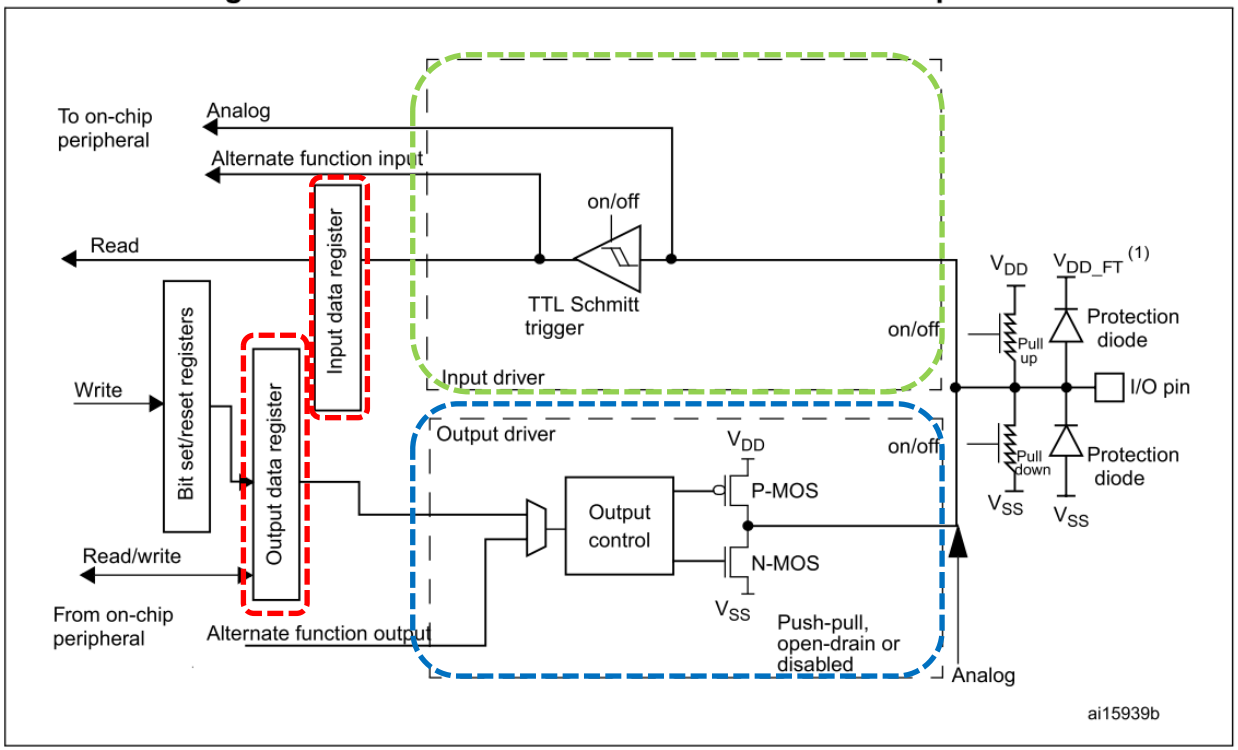
8.1 GPIO introduction

Each general-purpose I/O port has four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR), two 32-bit data registers (GPIOx_IDR and GPIOx_ODR), a 32-bit set/reset register (GPIOx_BSRR), a 32-bit locking register (GPIOx_LCKR) and two 32-bit alternate function selection register (GPIOx_AFRH and GPIOx_AFRL).

GPIO krmilnik – vhod/izhod

STM32F4

Figure 25. Basic structure of a five-volt tolerant I/O port bit



Potrebni koraki za krmiljenje izhoda:

1. RCC_AHB1ENR(Peripheral Clock Register): b_3=1 .. Port D Enable
2. **MODER (Mode Register): 01: General purpose output mode**
3. Default vrednosti že ustrezne v registrih :
   OTYPER (Output TYPE Register): 0: Output push-pull (reset state)
   OSPEEDR (Output SPEED Register): 00 – Low speed (reset state)
   PUPDR (Pull Up/Down Register): 00 – No pull (reset state)
4. določi stanje izhoda s pisanjem v ODR ali BSRR (nastavljamo na 1/0)

Osnovni projekt CubeIDE – GPIO – različni mogoči nivoji programiranja

STM32F4

Baremetal - zbirnik

```
INIT_IO:
  push {r5, r6, lr}
    // Enable GPIOD Peripheral Clock (bit 3 in AHB1ENR register)
    ldr r6, = RCC_AHB1ENR       // Load peripheral clock reg address to r6
    ldr r5, [r6]                // Read its content to r5
    orr r5, 0x00000008          // Set bit 3 to enable GPIOD clock
    str r5, [r6]                // Store result in peripheral clock register

    // Make GPIOD Pin12 as output pin (bits 25:24 in MODER register)
    ldr r6, =GPIOD_BASE       // Load GPIOD BASE address to r6
    ldr r5, [r6,#GPIOD_MODER]  // Read GPIOD_MODER content to r5
    and r5, 0x00FFFFFF          // Clear bits 31-24 for P12-15
    orr r5, 0x55000000          // Write 01 to bits 31-24 for P12-15
    str r5, [r6]                // Store result in GPIOD MODER register
  pop {r5, r6, pc}


LED_ON:
    push {r5, r6, lr}
    // Set GPIOD Pins to 1 (through BSSR register)
    ldr     r6, =GPIOD_BASE       // Load GPIOD BASE address to r6
    mov     r5, #LEDs_ON
    str     r5, [r6,#GPIOD_BSSR] // Write to BSRR register
    pop {r5, r6, pc}

LED_OFF:
    push {r5, r6, lr}
    // Set GPIOD Pins to 0 (through BSSR register)
    ldr     r6, =GPIOD_BASE       // Load GPIOD BASE address to r6
    mov     r5, #LEDs_OFF
    str     r5, [r6,#GPIOD_BSSR] // Write to BSRR register
    pop {r5, r6, pc}
```

https://github.com/LAPSyLAB/ORLab-STM32/tree/main/GPIO_LEDs

Baremetal - C

```
  /* USER CODE BEGIN 2 */


  RCC->AHB1ENR |= 0x08;
// Enable clock for GPIOD
  GPIOD->MODER |= 0x01000000;        //
MODE Register: bit 12 == out

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    GPIOD->ODR ^= 0x1000;            //
Toggle PD12

/* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    for (int i=0; i<0x1000000; i++) {};
// waste some time
  }
  /* USER CODE END 3 */
```

https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/LED_GPIO_C_Baremetal_C

Potrebni koraki za krmiljenje izhoda:

1. RCC_AHB1ENR(Peripheral Clock Register): b_3=1 .. Port D Enable
2. **MODER (Mode Register): 01: General purpose output mode**
3. Default vrednosti že ustrezne v registrih :
   OTYPER (Output TYPE Register): 0: Output push-pull (reset state)
   OSPEEDR (Output SPEED Register): 00 – Low speed (reset state)
   PUPDR (Pull Up/Down Register): 00 – No pull (reset state)
4. določi stanje izhoda s pisanjem v ODR ali BSRR (nastavljamo na 1/0)

HAL - C

```
/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {

HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);

  /* USER CODE END WHILE */

   /* USER CODE BEGIN 3 */
 HAL_Delay(1000);
 }
 /* USER CODE END 3 */
```

```
void HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx,
uint16_t GPIO_Pin)
{
  uint32_t odr;

  /* Check the parameters */
  assert_param(IS_GPIO_PIN(GPIO_Pin));

  /* get current Ouput Data Register value
*/
  odr = GPIOx->ODR;

  /* Set selected pins that were at low
level, and reset ones that were high */
  GPIOx->BSRR = ((odr & GPIO_Pin) <<
GPIO_NUMBER) | (~odr & GPIO_Pin);
}
```

https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/LED_Blink_Demo

Osnovni projekt CubeIDE – GPIO – tipka, LED diode

HAL - C

```
/* USER CODE BEGIN PV */
#define BUFSIZE 256
char SendBuffer[BUFSIZE];
int Counter;
int KeyState=0;

/* USER CODE END PV */



/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {

        HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);
        HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_13);
        HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_14);

        KeyState = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, KeyState);


         snprintf(SendBuffer,BUFSIZE,"Hello World [%d]: Key:%d\r\n",Counter++,KeyState);
         CDC_Transmit_FS(SendBuffer,strlen(SendBuffer));

   /* USER CODE END WHILE */

   /* USER CODE BEGIN 3 */
 HAL_Delay(1000);
 }
 /* USER CODE END 3 */
```

UM1725

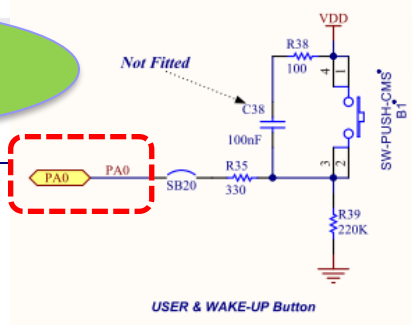
User manual

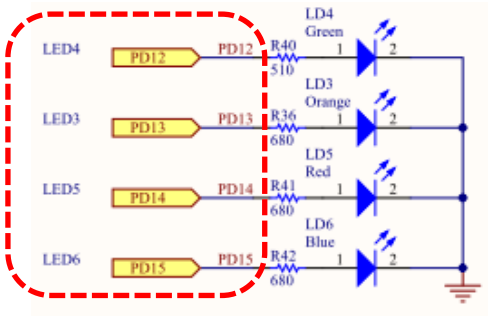
Description of STM32F4 HAL and low-layer drivers

31.2.4 IO operation functions

This section contains the following APIs:

- *HAL_GPIO_ReadPin()*
- *HAL_GPIO_WritePin()*
- *HAL_GPIO_TogglePin()*
- *HAL_GPIO_LockPin()*
- *HAL_GPIO_EXTI_IRQHandler()*
- *HAL_GPIO_EXTI_Callback()*





https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/STM32_USB_Key_LED