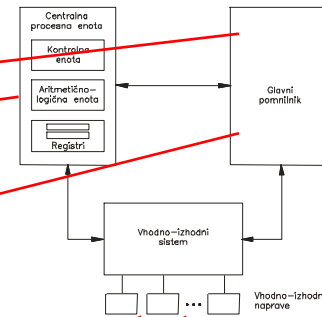
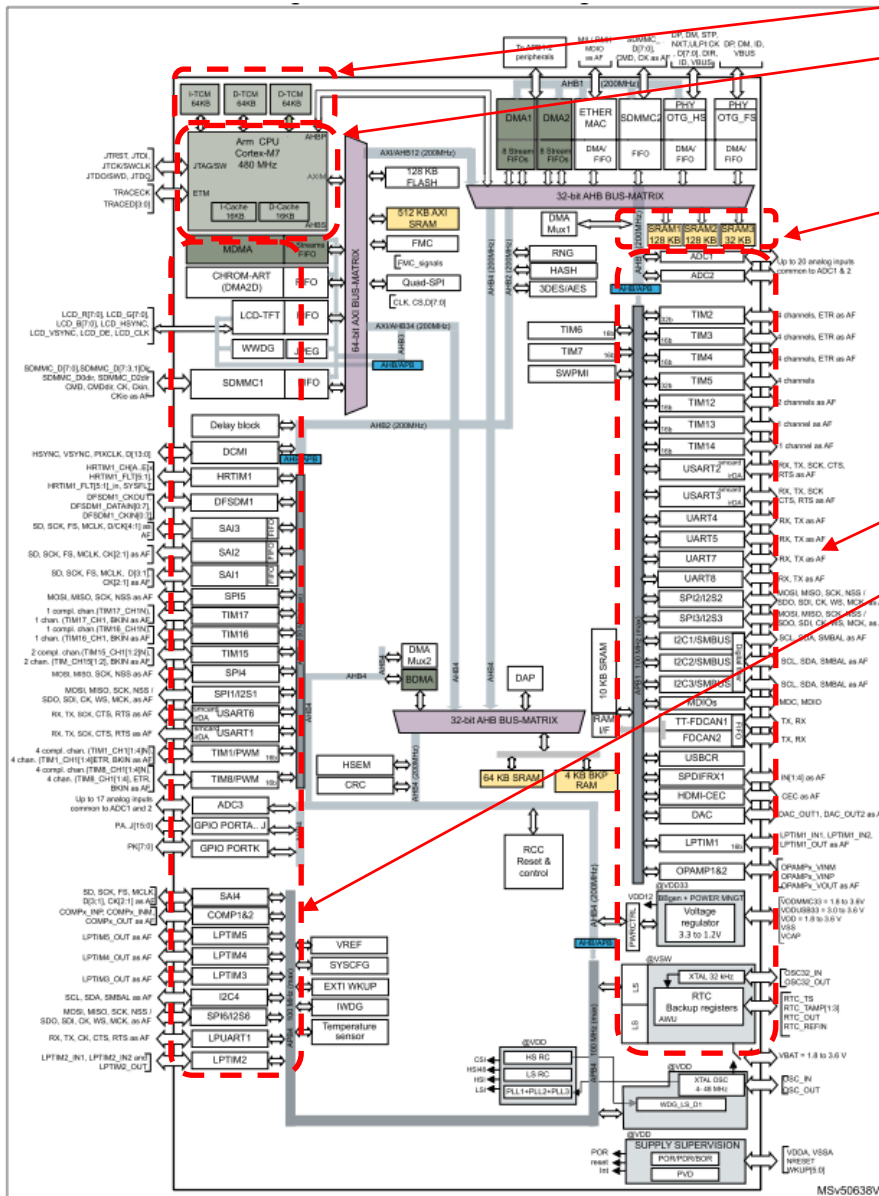


STM32H7

Vhodno / izhodne naprave

CubeIDE : osnovni projekt v C + HAL knjižnica

STM32H750XB



Osnovni projekt CubeIDE – GPIO – različni mogoči nivoji programiranja

Baremetal - zbirnik

```
INIT_IO:
push {r5, r6, lr}
// Enable GPIO 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 GPIO clock
str r5, [r6] // Store result in peripheral clock register

// Make GPIO Pin12 as output pin (bits 25:24 in MODER register)
ldr r6, =GPIO_BASE // Load GPIO BASE address to r6
ldr r5, [r6,#GPIO_MODER] // Read GPIO_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 GPIO MODER register
pop {r5, r6, pc}
```

```
LED_ON:
push {r5, r6, lr}
// Set GPIO Pins to 1 (through BSSR register)
ldr r6, =GPIO_BASE // Load GPIO BASE address to r6
mov r5, #LEDS_ON
str r5, [r6,#GPIO_BSSR] // Write to BSSR register
pop {r5, r6, pc}
```

```
LED_OFF:
push {r5, r6, lr}
// Set GPIO Pins to 0 (through BSSR register)
ldr r6, =GPIO_BASE // Load GPIO BASE address to r6
mov r5, #LEDS_OFF
str r5, [r6,#GPIO_BSSR] // Write to BSSR register
pop {r5, r6, pc}
```

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

RA, OR



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 BSSR (nastavljamo na 1/0)

VIN

Baremetal - C

```
/* USER CODE BEGIN 2 */

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

/* USER CODE END 2 */

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

HAL - C

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_GPIO_TogglePin(GPIO, 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 Output 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](https://github.com/LAPSYLAB/STM32F4_Discovery_VIN_Projects/tree/main/LED_Blink_Demo)

Delo na STM32H7 razvojnem sistemu

Priključitev :

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

Mikro USB priklp na daljši stranici (srednji !!!) ↓

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

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

https://github.com/LAPSYLAB/ORLab-STM32H7/tree/main/STM32H750B-DK_C_Basic



```
CubeIDEWorkspace - Sluzba/ORLab-STM32H7/STM32H750B-DK_C_Basic/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
CubelDE_Workspace
Delo
Node_V4 (in node_v4)
Sluzba
  CAN_IEX_Module
  CAN_IEX_Module_bak
  H7-BSP-LCD-OS
  ORLab-STM32
  ORLab-STM32H7
    Docs
    DWT_Cycles_Measurements
    GPIO_LEDs
    STM32H750B-DK_C_Basic
      Core
        Inc
        Src
main.c
131
132 /* Infinite loop */
133 /* USER CODE BEGIN WHILE */
134 while (1)
135 {
136     HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_13);
137     HAL_GPIO_TogglePin(GPIOJ, GPIO_PIN_2);
138
139     /* USER CODE END WHILE */
140
141     /* USER CODE BEGIN 3 */
142     snprintf (SendBuffer,BUFSIZE,"USART3:%d secs\r\n",Cnt);
143     HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),1).
144
145     HAL_Delay(1000);
146     Cnt++;
147 }
148 /* USER CODE END 3 */
149 }
150
```

- 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

Lastni viri :

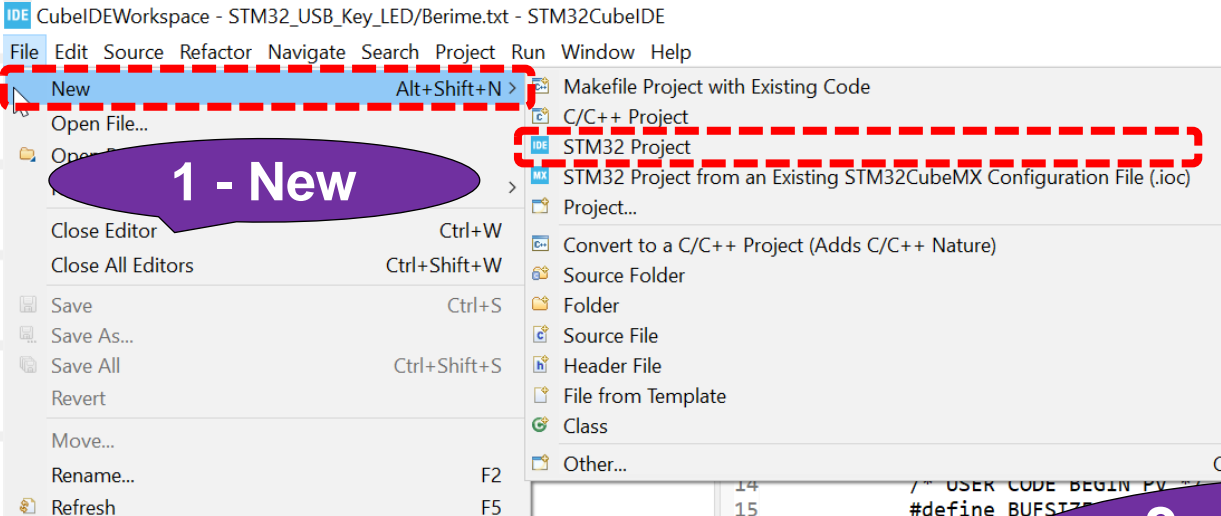
<https://github.com/LAPSYLAB/ORLab-STM32H7>

https://github.com/LAPSYLAB/STM32H7_Discovery_VIN_Projects



CubeIDE – Vzpostavitev novega projekta

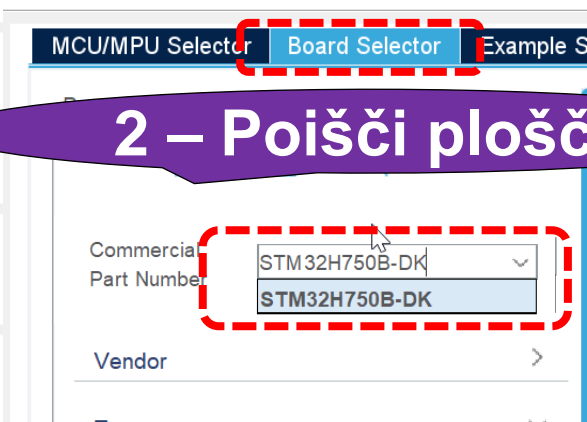
Nov projekt :



STM32 Project

Target Selection

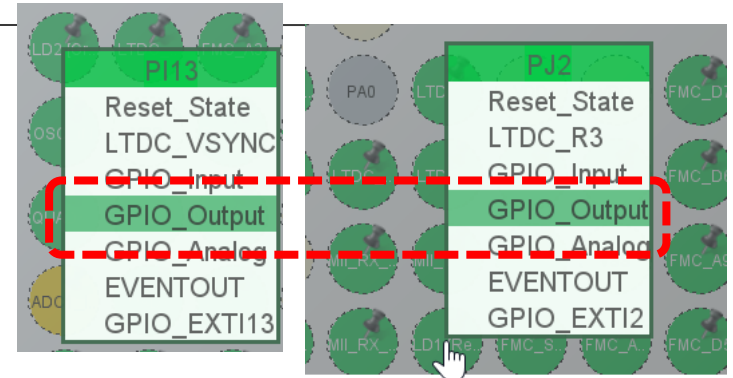
STM32 target or STM32Cube example selection is required



STM32H7

Osnovni projekt CubeIDE – CubeMX

Konfiguracija : priključki, knjižnice STM32H7



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 Settings...

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 to

MCU and Firmware Package

MCU Reference: STM32F407VGTx

Firmware Package Name and Version: STM32Cube FW_F4 V1.26.2

Advanced Settings

Generated Function Calls

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



4 – Preveri nastavitve

STM32H7

Osnovni projekt CubeIDE – USB Virtual COM

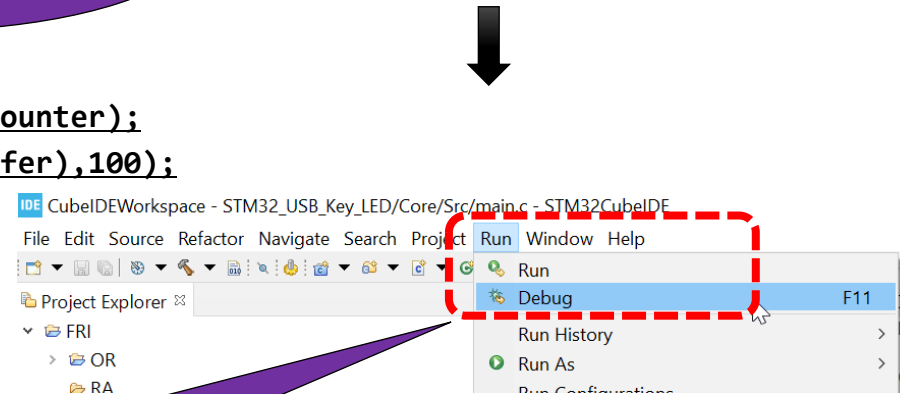
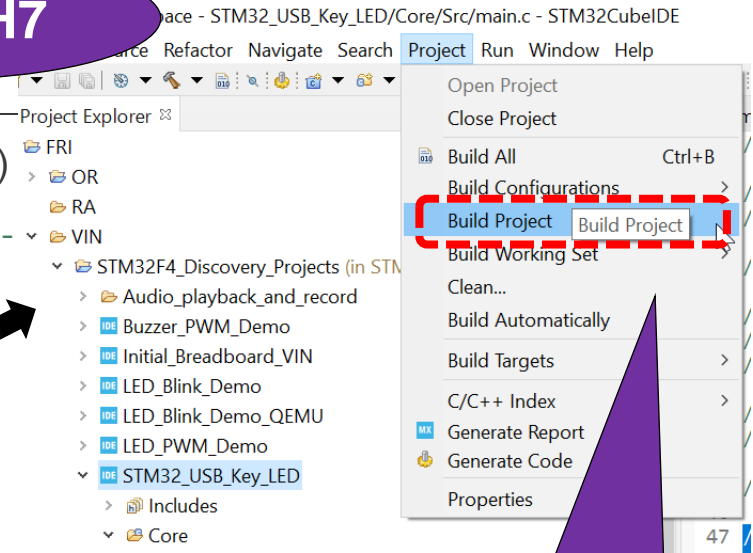
Program : za pošiljanje po USB Virtual COM Port (USART3)

```
/* 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 */
```

5 – UART
koda

6 – Build
project

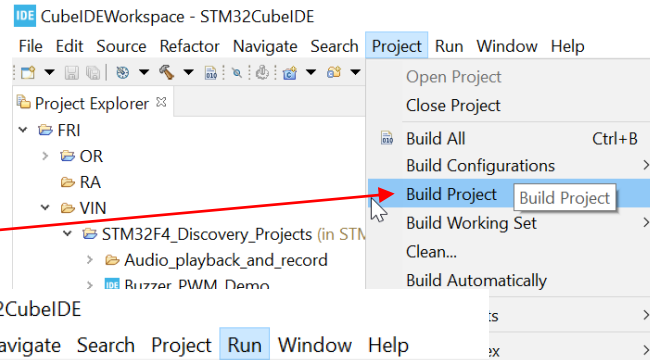
7 – Debug
project



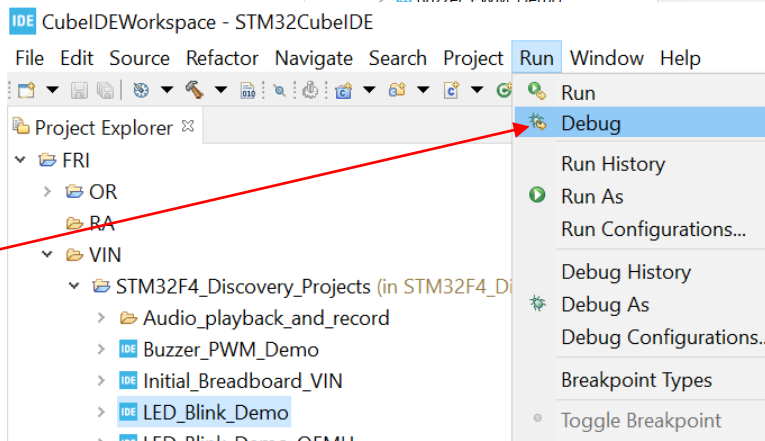
CubeIDE – Zagon, debug

Prevajanje, zagon :

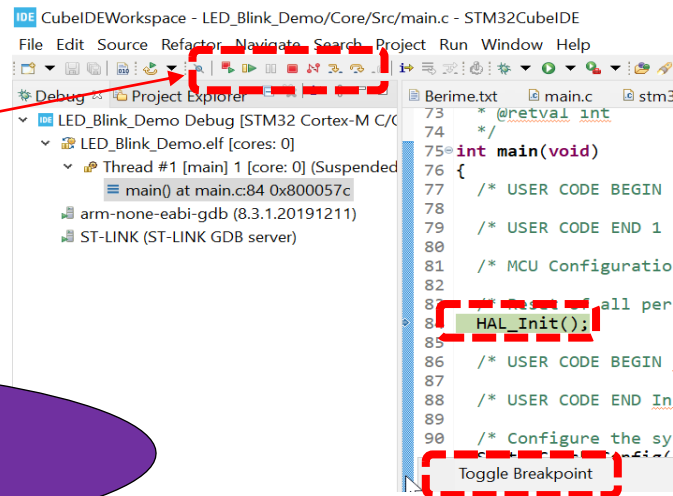
- Project -> Build Project



- Run -> Debug

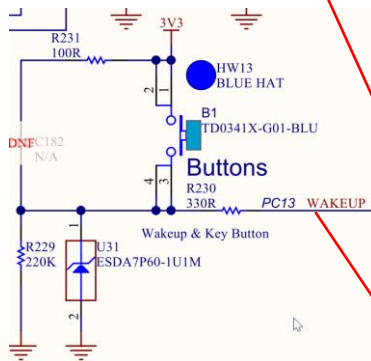


- Step (Into,Over), Breakpoints

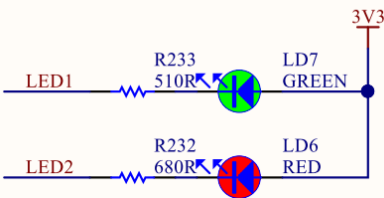


Build <-> Debug
project, ...

GPIO Krmilnik STM32H7



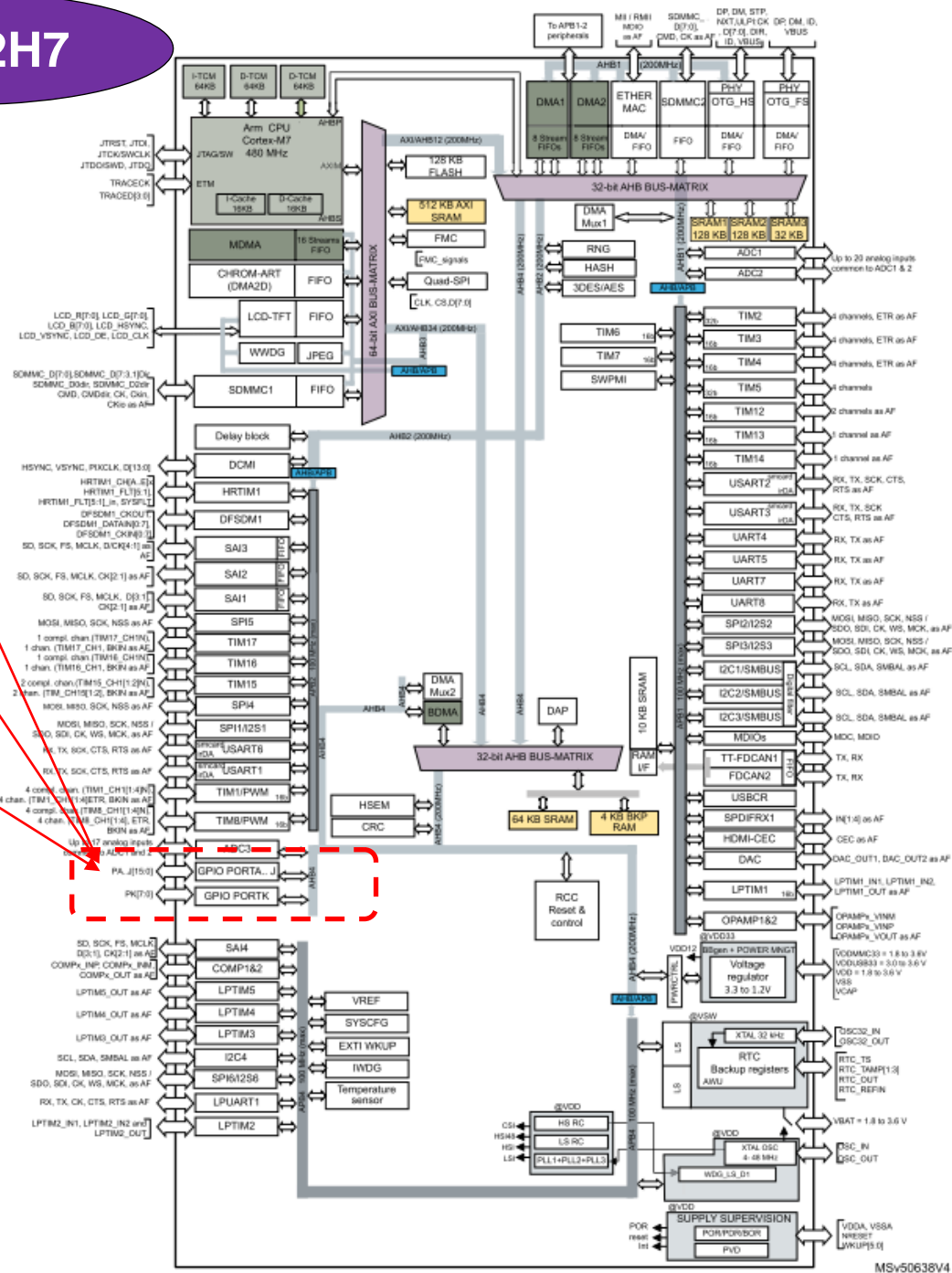
LEDs



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
		PI1	T6	LED1
		PJ2	T16	LCD R4

LED: rdeča PI13, zelena PJ2



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);

    sprintf(SendBuffer, BUFSIZE, "Hello World [%d]: Key:%d\r\n", Counter++, KeyState);
    HAL_UART_Transmit(&uart3, SendBuffer, strlen(SendBuffer), 100);

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

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

```

UM2217

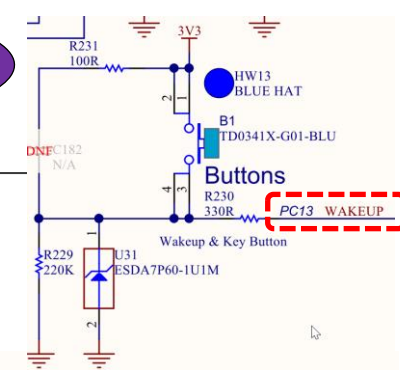
User manual

Description of STM32H7 HAL and low-layer drivers

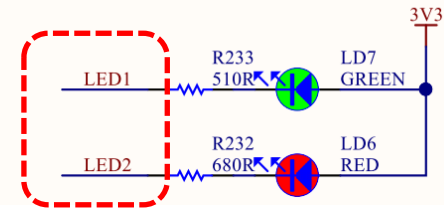
35.2.4 IO operation functions

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

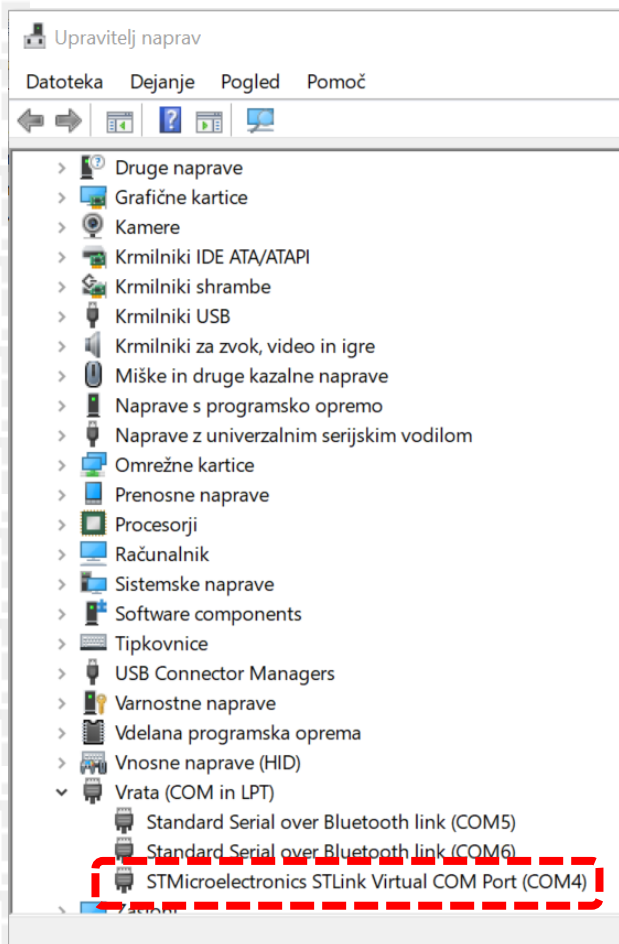
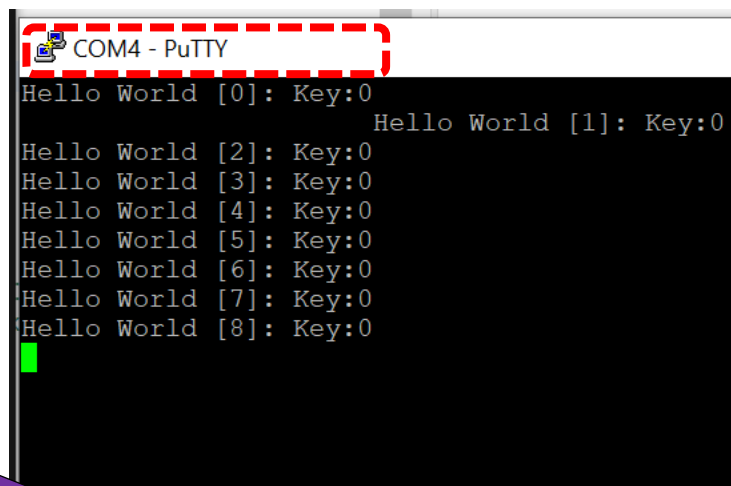
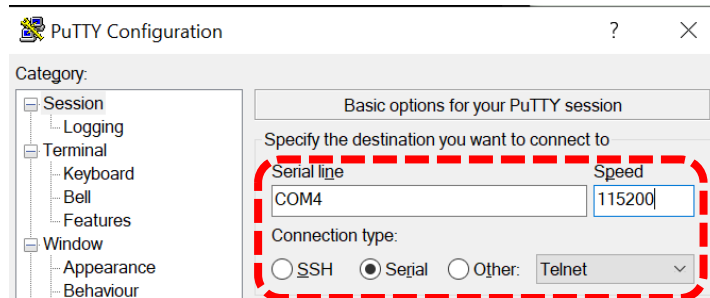


P111	H1	LCD HSYNC
P112	H2	LED2
P113	H3	LCD CLK
P114	P5	LCD R0
P115		

P10	P6	LCD R2
P11	T6	LED1
P12	T16	LCD R4

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