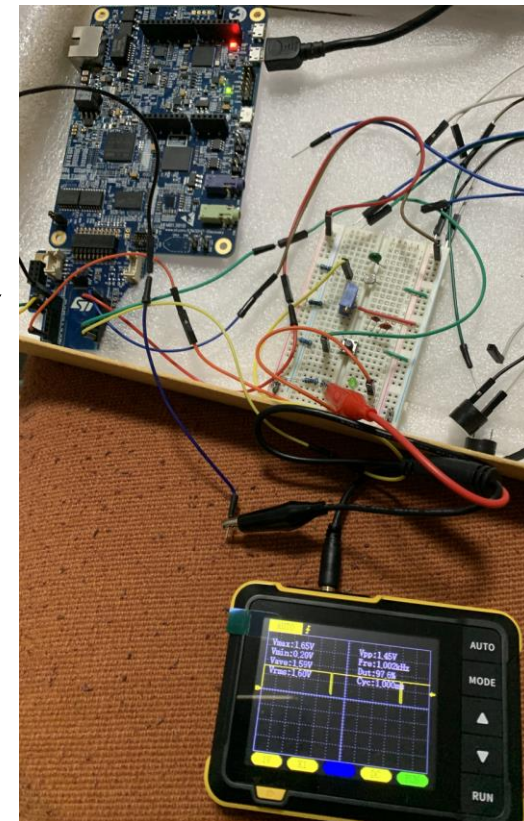


ARM

*Praktični projekt za STM32H7 vgrajen sistem
(informativno, dodatno gradivo)*

Utripanje LED diode, Brenčoč

Arduino IDE + STM32 razširitev



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

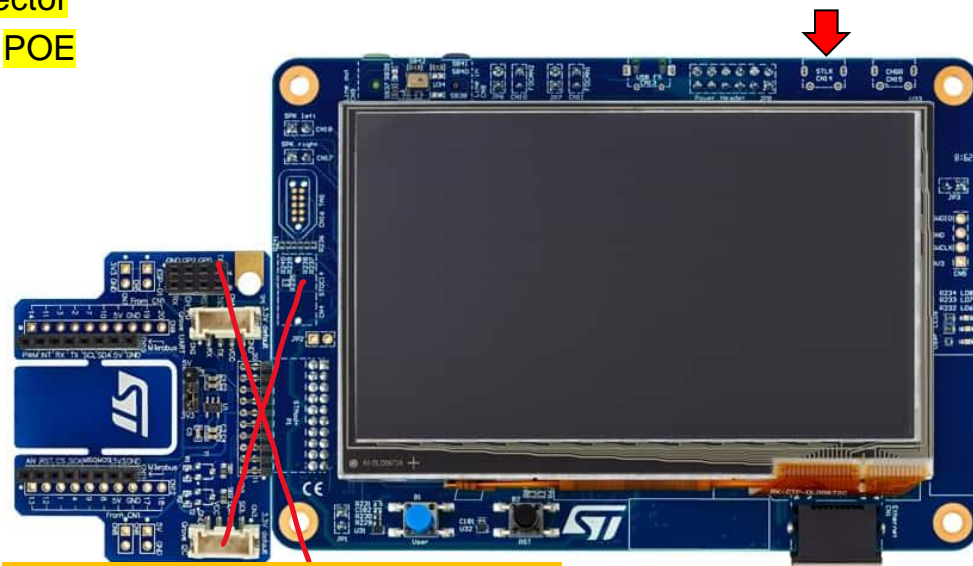
- 1 user and reset push-button

- Fanout daughterboard

- 2 x FDCANs

- Board connectors:

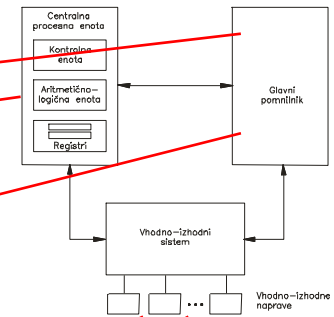
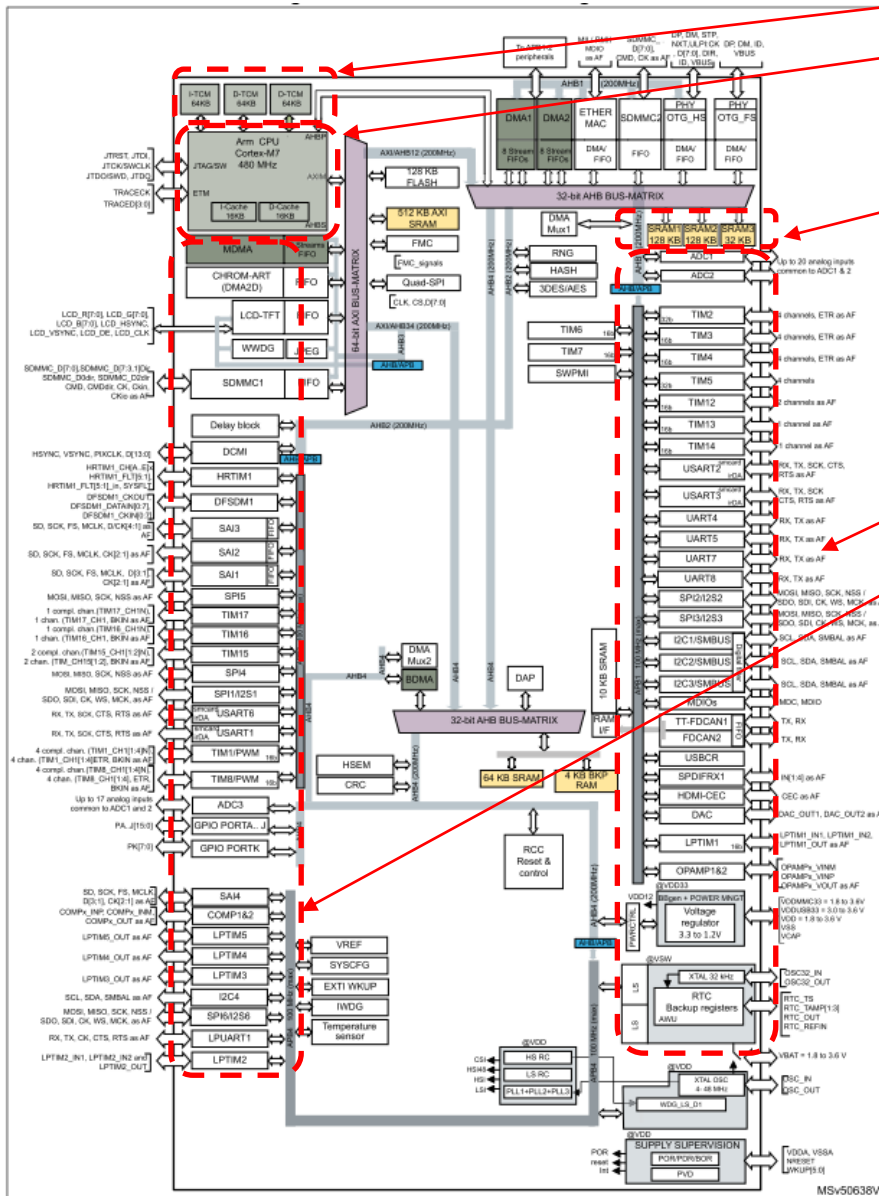
- USB FS Micro-AB connectors
- ST-LINK Micro-B USB connector
- USB power Micro-B connector
- Ethernet RJ45
- Stereo headset jack including analog microphone input
- Audio header for external speakers
- Arduino™ Uno V3 expansion connectors
- STMod+



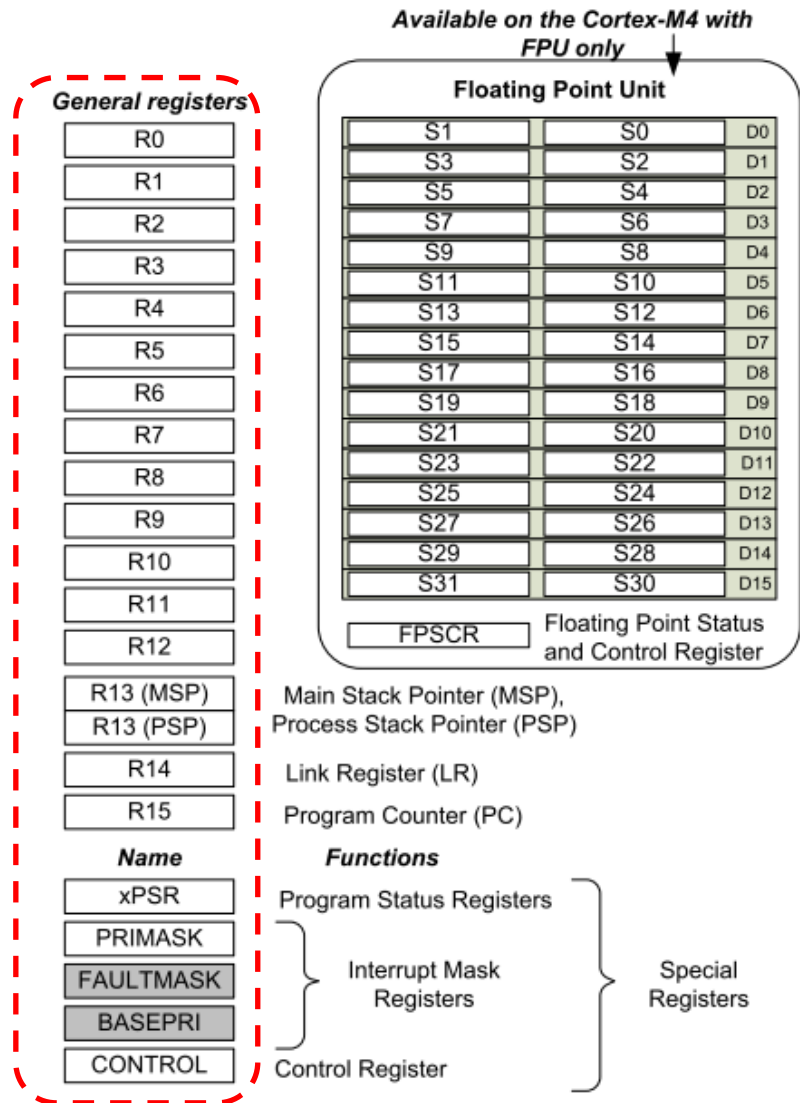
Prikaz napačne povezave na sliki!

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

STM32H750XB

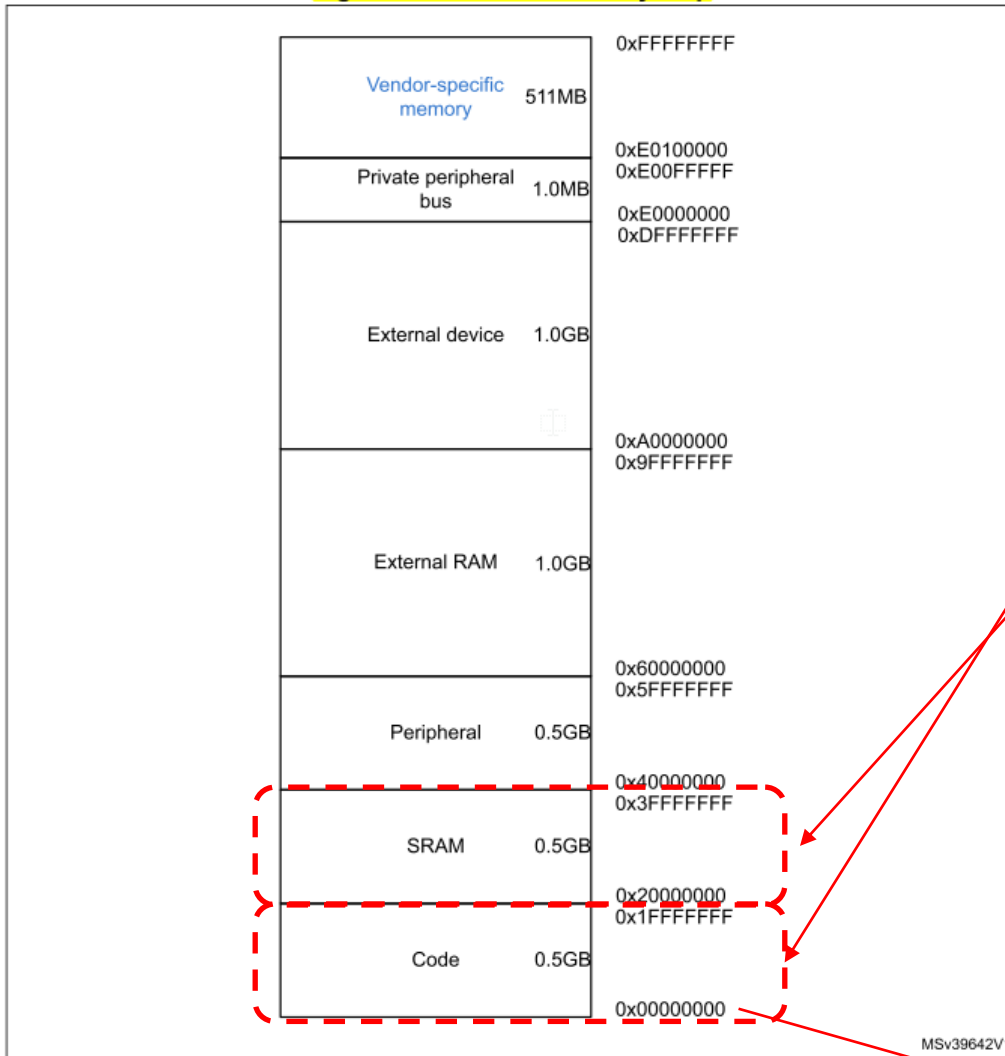


ARM Cortex M – Programski model



ARM Cortex M – Pomnilniški (naslovni) prostor

Figure 8. Processor memory map



Opis razporeda pomnilnikov – linker (povezovalnik)

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
}
    
```

SysTick vector	1	0x0000003C
PendSV vector	1	0x00000038
Not used		0x00000034
Debug Monitor vector	1	0x00000030
SVC vector	1	0x0000002C
Not used		0x00000028
Not used		0x00000024
Not used		0x00000020
SecureFault (ARMv8-M Mainline)	1	0x0000001C
Usage Fault vector	1	0x00000018
Bus Fault vector	1	0x00000014
MemManage vector	1	0x00000010
HardFault vector	1	0x0000000C
NMI vector	1	0x00000008
Reset vector	1	0x00000004
MSP initial value		0x00000000

ARM Cortex M – Vektorska tabela – začetek delovanja

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

Naslov 4 („Reset vector“) vsebuje naslov 1. ukaza „inicializacijskega“ dela, torej začetna koda se nahaja v podprogramu `Reset_Handler`.

```
.section .text.Reset_Handler
    .weak Reset_Handler
    .type Reset_Handler, %function
```

Reset_Handler:

```
...
    ldr    sp, =_estack    /* set stack pointer */

/* Call the application's entry point.*/

    bl    main

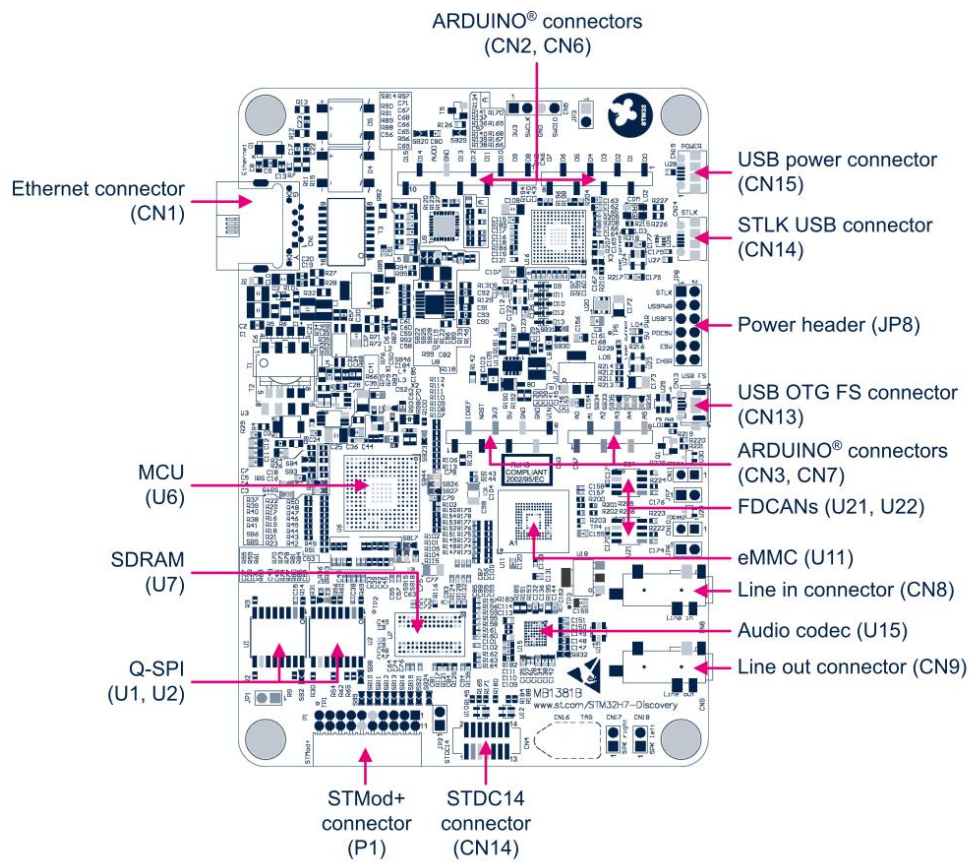
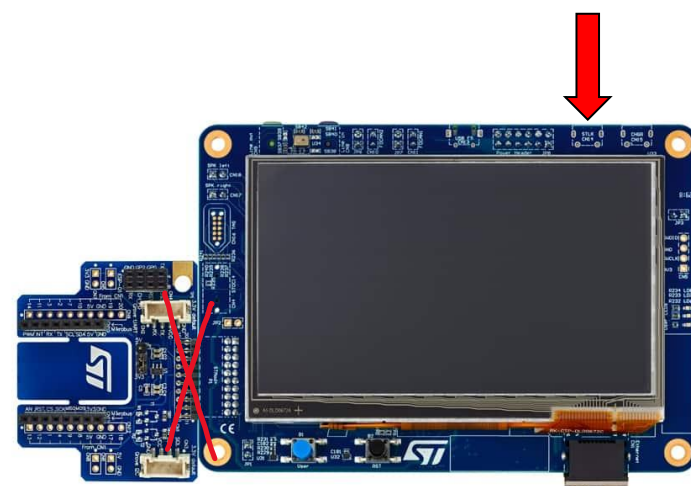
    bx    lr
```

Delo na plošči STM32H750B-DK

Priključitev :

- **Micro USB** prikllop nad zaslonom (**puščica!**), zasveti zaslon

Figure 5. STM32H745I-DISCO and STM32H750B-DK Discovery board bottom layout



Delo na plošči STM32H750B-DK

Compile

Upload

```
STM32H750_Basic_GPIO_SpeedTest_Serial | Arduino 1.8.19
File Edit Sketch Tools Help

STM32H750_Basic_GPIO_SpeedTest_Serial
// #include "stm32h750.h"
2
3 int i,ms,RegisterCMs,RegisterASMMs,WriteFastMs,WriteMs;
4 int j = 1;
5
6 //                RX    TX
7 HardwareSerial Serial13(PB_11, PB_10);
8
9
10
11 void setup() {
12     // put your setup code here, to run once:
13     pinMode(PD_12, OUTPUT);
14     pinMode(PD_13, OUTPUT);
15
16     pinMode(PI_13, OUTPUT);
17     pinMode(PJ_2, OUTPUT);
18
19     Serial13.begin(115200);
20 }
21
22
23 void loop() {
24     // put your main code here, to run repeatedly:
25
26     Serial13.print("----- New series of tests : ");
27     Serial13.println(j);
28
29     // 1. test - I/O access through Arduino library API
30     i=0;
31     Serial13.print("digitalWrite [ms]:");
32     ms=millis();
33     while (i<1000000) {
34         // statement(s)
35         digitalWrite(PD 12. HIGH);
36
```

Posebna distribucija za STM32H750B-DK (e-učilnica) :

- **Arduino IDE okolje s STM32 razširitvijo**
 - prenosna („portable“) instalacija (na D:\RAVINOR)
 - https://github.com/LAPSYLAB/STM32Duino_for_H7/releases/tag/V0.1
 - omogoča:
 - enostavno programiranje
 - uporabo osnovnih vhodno-izhodnih naprav
 - LED diodi
 - GPIO vhodi/izhodi
 - PWM (krmiljenje brenčaća – melodije)
 - https://github.com/LAPSYLAB/STM32Duino_for_H7

The screenshot shows the Windows Device Manager window with 'Upravitelj naprav' (Device Manager) open. Under 'Vrata (COM in LPT)' (Ports), 'STMMicroelectronics STLink Virtual COM Port (COM3)' is selected and highlighted in yellow. A red arrow points from this entry to a terminal window. The terminal window displays the following output:

```
----- New series of tests : 234
digitalWrite [ms]:4410
digitalWriteFast [ms]:750
Registers-C [ms]:750
Registers-ASM [ms]:750
----- New series of tests : 235
digitalWrite [ms]:4410
digitalWriteFast [ms]:750
Registers-C [ms]:750
Registers-ASM [ms]:750
----- New series of tests : 236
digitalWrite [ms]:4410
digitalWriteFast [ms]:750
Registers-C [ms]:750
Registers-ASM [ms]:750
```


Delo na plošči STM32H750B-DK - Nastavitve

Compile

Upload

```
STM32H750_Basic_GPIO_SpeedTest_Serial | Arduino 1.8.19
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Generic STM32H7 Series" >
Board part number: "Generic H750XBHx" >
U(S)ART support: "Enabled (generic 'Serial')" >
USB support (if available): "None" >
USB speed (if available): "Low/Full Speed" >
Optimize: "Debug (-Og)" >
Debug symbols and core logs: "None" >
C Runtime Library: "Newlib Nano (default)" >
Upload method: "STM32CubeProgrammer (SWD)" >
Port >
Get Board Info
pinMode(PJ_2, OUTPUT);
Serial3.begin(115200);
}
void loop() {
// put your main code here, to run repeatedly:
Serial3.print("----- New series of tests : ");
Serial3.println(j);
// 1. test - I/O access through Arduino library API
i=0;
Serial3.print("digitalWrite [ms]:");
ms=millis();
while (i<1000000) {
// statement(s)
digitalWrite(PD 12, HIGH);
}
```

Posebna distribucija za STM32H750B-DK (e-učilnica) :

- **Arduino IDE okolje s STM32 razširitvijo**
 - prenosna („portable“) instalacija (na D:\RAVINOR)
 - https://github.com/LAPSYLAB/STM32Duino_for_H7/releases/tag/V0.1
 - omogoča:
 - enostavno programiranje
 - uporabo osnovnih vhodno-izhodnih naprav
 - LED diodi
 - GPIO vhodi/izhodi
 - PWM (krmiljenje brenčaća – melodije)
 - https://github.com/LAPSYLAB/STM32Duino_for_H7

Upravitelj naprav

Datoteka Dejanje Pogled Pomoč

- ▼ Vrata (COM in LPT)
 - Standard Serial over Bluetooth link (COM5)
 - Standard Serial over Bluetooth link (COM6)
 - STMMicroelectronics STLink Virtual COM Port (COM3)

```
COM3
----- New series of tests : 234
digitalWrite [ms]:4410
digitalWriteFast [ms]:750
Registers-C [ms]:750
Registers-ASM [ms]:750
----- New series of tests : 235
digitalWrite [ms]:4410
digitalWriteFast [ms]:750
Registers-C [ms]:750
Registers-ASM [ms]:750
----- New series of tests : 236
digitalWrite [ms]:4410
digitalWriteFast [ms]:750
Registers-C [ms]:750
Registers-ASM [ms]:750
```

Osnovni program za utripanje LED diod in serijsko komunikacijo

STM32H750 Basic LED Serial.ino

```
int i;
//                RX    TX
HardwareSerial Serial3(PB_11, PB_10);

void setup() {
  // put your setup code here, to run once:
  // initialize digital pins PI13,PJ2 as outputs.
  pinMode(PI_13, OUTPUT);
  pinMode(PJ_2, OUTPUT);
  Serial3.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(PI_13, HIGH); // turn the LED on (HIGH is the voltage level)
  digitalWrite(PJ_2, LOW);   // turn the LED off by making the voltage LOW

  delay(1000);               // wait for a second
  i++;

  digitalWrite(PI_13, LOW); // turn the LED off by making the voltage LOW
  digitalWrite(PJ_2, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second

  Serial3.print("Hello World: ");
  Serial3.println(i);
}
```

Sestavine programa:

- int i;
 - Deklaracija spremenljivke (integer)
- setup () { ... }:
 - se izvede enkrat ob zagonu
- loop () { ... }:
 - se izvaja ves čas

Funkcije:

pinMode (nastavitev nožice vhod/izhod)
digitalWrite (nastavitev stanja nožice 1/0)
delay (n) – počakaj n ms
Serial3 – objekt serijske povezave
Print (izpis)
Println (izpis in nova vrstica)

Osnovni program za krmiljenje GPIO nožic (1/0)

STM32H750 Basic GPIO SpeedTest Serial.ino

```
pinMode(PD_12, OUTPUT);

digitalWrite(PD_12, HIGH);
digitalWrite(PD_12, LOW);

digitalWriteFast(PD_12, HIGH);
digitalWriteFast(PD_12, LOW);

//Set D12 (HIGH)
GPIOID->BSRR = 0b0001000000000000 << 16; //move to upper 16 bits
//Set D12 (LOW)
GPIOID->BSRR = 0b0001000000000000; //move to upper 16 bits

asm (
    " ldr.w r0, = 0x58020C18          \n" // GPIOID->BSRR = (0x58020C00 + 0x18 ) SetReset Register
    " ldr r3,  = 0b0001000000000000 << 16 \n" // Bit 12
    " str r3, [r0]                  \n" // Set port high

    " ldr r3,  = 0b0001000000000000      \n" // Bit 12
    " str r3, [r0]                  \n" // Set port low
    : : : "r0", "r3");
```

Krmiljenje GPIO nožic na različnih nivojih

- **Visokonivojski programski jezik:**

- Arduino API funkcije (digitalWrite)
- STM32 API funkcije (digitalWriteFast)

- **Direktno krmiljenje GPIO:**

- pisalni dostop do registrov – višji nivo
- pisalni dostop do registrov – zbirnik

----- New series of tests : 519

digitalWrite [ms]:4410

digitalWriteFast [ms]:750

Registers-C [ms]:750

Registers-ASM [ms]:750

Funkcije:

digitalWrite

(Arduino API)

digitalWriteFast

(STM32 HAL API)

GPIOID->BSRR =

(direktni vpis – C++)

str r3, [r0]

(direktni vpis – zbirnik)

Osnovni program za krmiljenje brenčača s PWM signalom

STM32H750 Basic PWM Melody.ino

```
void setup() {
  // put your setup code here, to run once:
  // initialize digital pins PI13,PJ2 as outputs.
  pinMode(PA_3, OUTPUT);

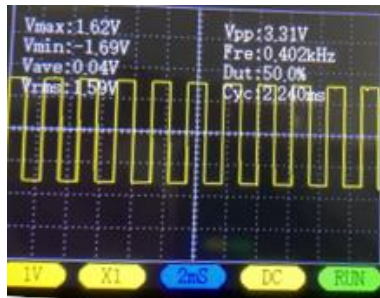
  Serial3.begin(115200);
}

void loop() {

  // Write PWM signal at frequency to pin
  analogWriteFrequency(Frekvenca_Hz); // Set PMW period to Note Hz
  analogWrite(PA_3, 128); // Start PWM on Pin, at Frekvenca_Hz with 50% duty cycle
  delay(noteDuration);

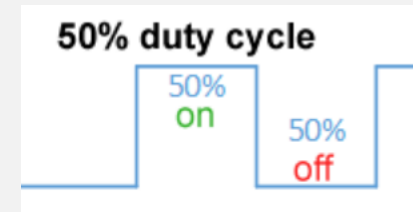
  // stop the tone playing:
  analogWrite(PA_3, 0); // Start PWM on Pin, at melody[thisNote] Hz with 0% duty cycle
  // to distinguish the notes, set a minimum time between them.
  int pauseBetweenNotes = noteDuration * 0.3;
  delay(pauseBetweenNotes);

  ...
}
```



Krmiljenje pravokotnega signala, ki ga priključimo na brenčač:

- Polovico periode visoko, polovico periode nizko stanje
- Spreminjamo frekvenco oziroma periodo signala, da dobimo posamezne tone.



```
-----
Note (#0) [Hz]:349   Duration:250
Note (#1) [Hz]:392   Duration:250
Note (#2) [Hz]:440   Duration:125
Note (#3) [Hz]:392   Duration:250
Note (#4) [Hz]:440   Duration:125
Note (#5) [Hz]:466   Duration:250
```

```
itches.h:
/*****
 * Public Constants
 *****/

...
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
```

Funkcije:

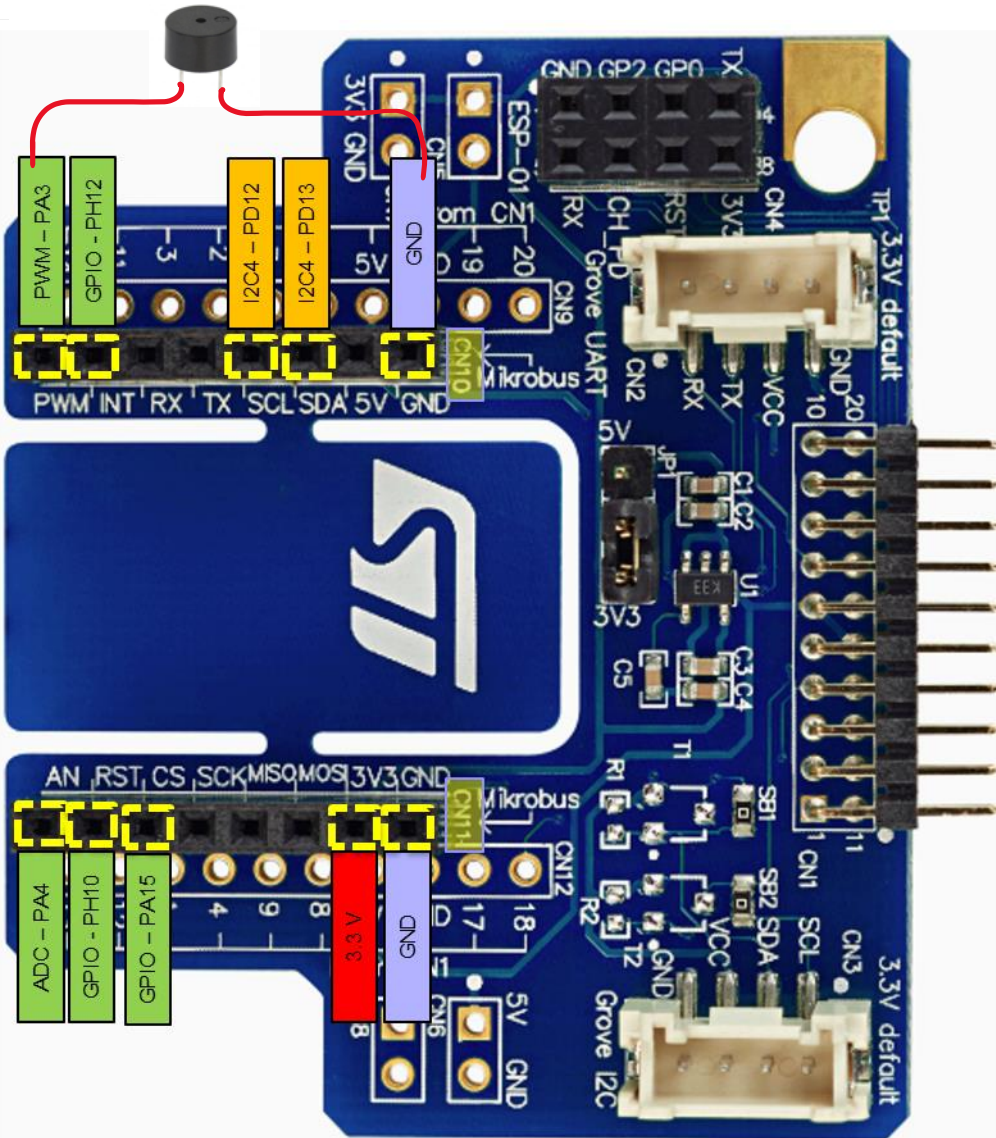
`analogWriteFrequency` (določi frekvenco)
`analogWrite` (pin, 50% duty)

Frekvenca se določi vedno pred klicem `analogWrite`

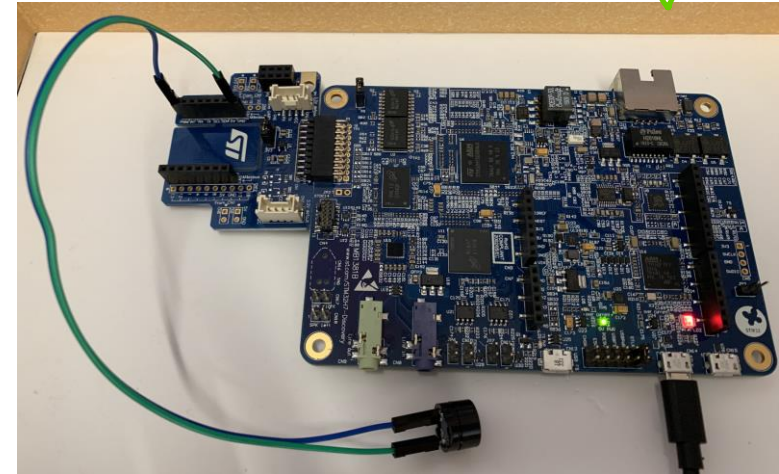
Izzivi

- **Utripanje LED diod**
 - Skrajšanje časa vklopa/izklopa
 - Spreminjanje razmerja trajanja vklopa/izklopa („dimmer“, „PWM-dimmer“)
- **Krmiljenje GPIO nožic na različnih nivojih**
 - Visokonivojski programski jezik:
 - Arduino API funkcije (digitalWrite)
 - STM32 API funkcije (digitalWriteFast)
 - Direktno krmiljenje GPIO:
 - pisalni dostop do registrov – višji nivo
 - pisalni dostop do registrov – zbirnik
- **Uporaba razširitvene plošče:**
 - vklop/izklop nožic na razširitveni plošči
 - tvorba PWM signala za krmiljenje brenčača
 - priklop brenčača na PA3 & GND
 - toni, melodije, ...

STM32H750B – DISCOVERY StMod+ konektor



Pravilna priključitev ✓



Neppravilna priključitev



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

Prikaz napačne povezave na sliki!

Izzivi – Arduino IDE&STM32Duino programiranje

Language Reference

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure.

Functions

For controlling the Arduino board and performing computations.

Digital I/O

digitalRead()
digitalWrite()
pinMode()

Analog I/O

analogRead()
analogReference()
analogWrite()

Zero, Due & MKR Family

analogReadResolution()
analogWriteResolution()

Advanced I/O

noTone()

Math

abs()
constrain()
map()
max()
min()
pow()
sq()
sqrt()

Trigonometry

cos()
sin()
tan()

Characters

isAlpha()

Random Numbers

random()
randomSeed()

Bits and Bytes

bit()
bitClear()
bitRead()
bitSet()
bitWrite()
highByte()
lowByte()

External Interrupts

attachInterrupt()
detachInterrupt()

Reference > Libraries > Stm32duino examples

STM32duino Examples

Other

Provides several examples for the Arduino core for STM32 MCUs.

Arduino STM32 core, libraries and examples are available here: <https://github.com/stm32duino>

Author: several

Maintainer: stm32duino

[Read the documentation](#)

Compatibility

This library is compatible with the **stm32** architecture.

- <https://www.arduino.cc/reference/en/>
- veliko ostalih spletnih virov
- veljajo tudi posebnosti uporabe knjižnice STM32Duino
 - <https://www.arduino.cc/reference/en/libraries/stm32duino-examples/>