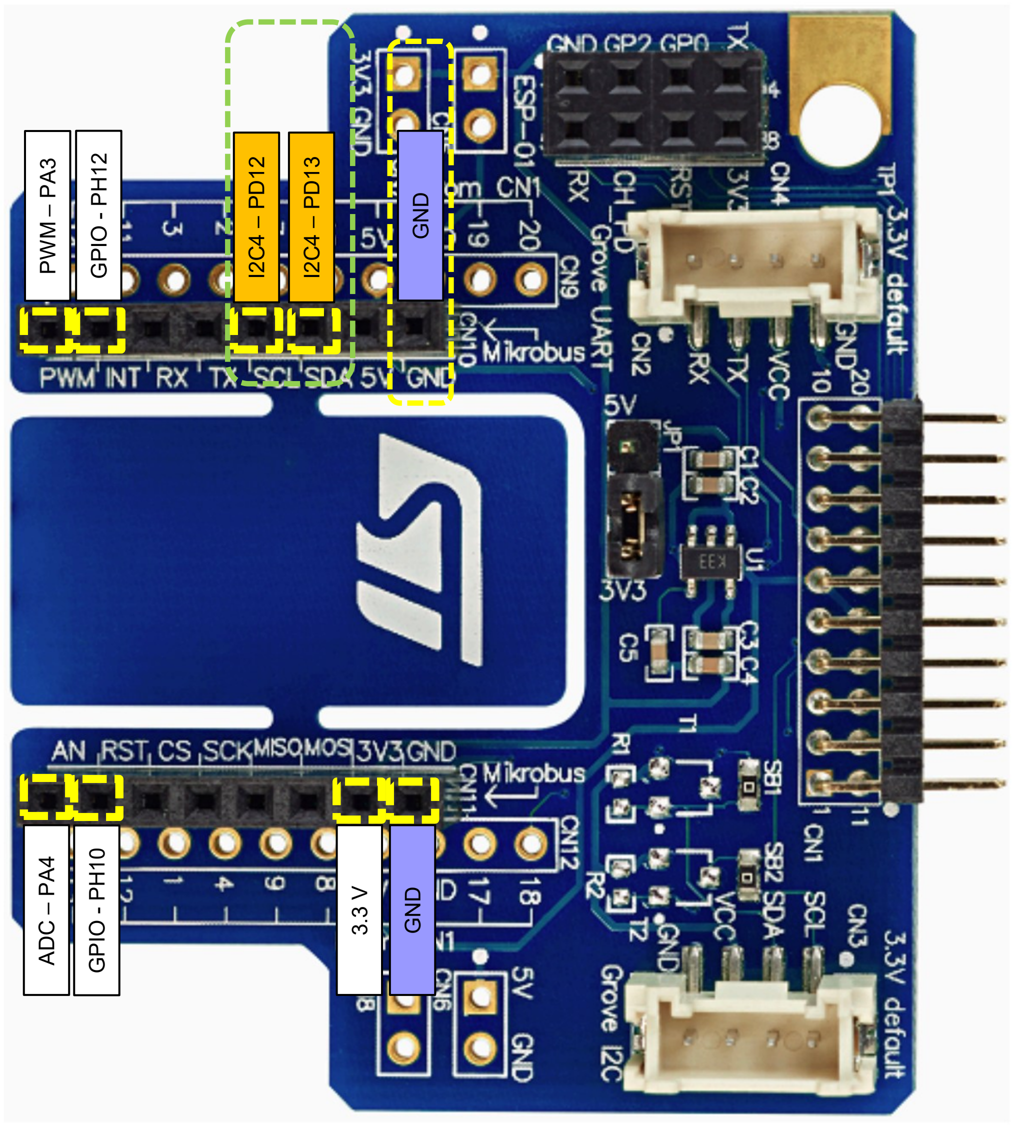
Vhodno izhodne naprave

VP 6: STM32H7 I2C primeri, STM32F4 SPI, pospeškomer, „Air USB mouse“

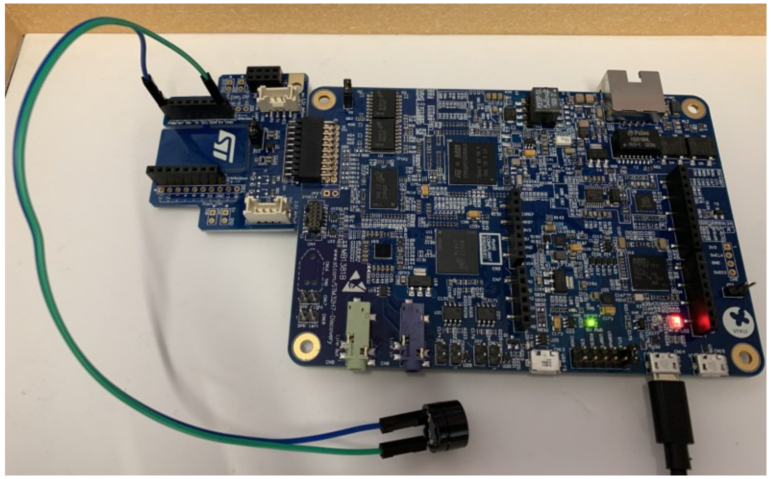
VP 6: STM32H7 I2C primeri, STM32F4 SPI, pospeškomer, „Air USB mouse“

- STM32H7 CubeIDE, I2C (Scan, WM9884, Touch)
- Predstavitev STM32F4
- CubeIDE projekt STM32F4, SPI, pospeškomer:
  - SPI - LIS3DSH pospeškomer
  - USB „Air Mouse“ with STM32F4, SPI1 and LIS3DSH
- Sledenje („tracing“) – CubeMonitor
- Osciloskop - ponovitev

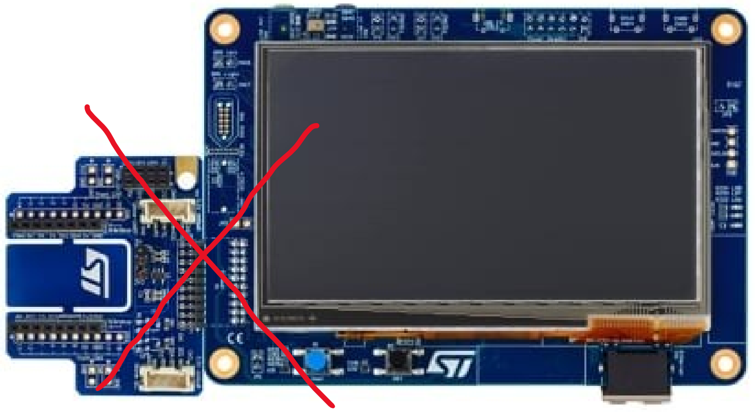
STM32H7 – I2C signali SCL (PD12), SDA (PD13), GND (2x)



Pravilna priključitev

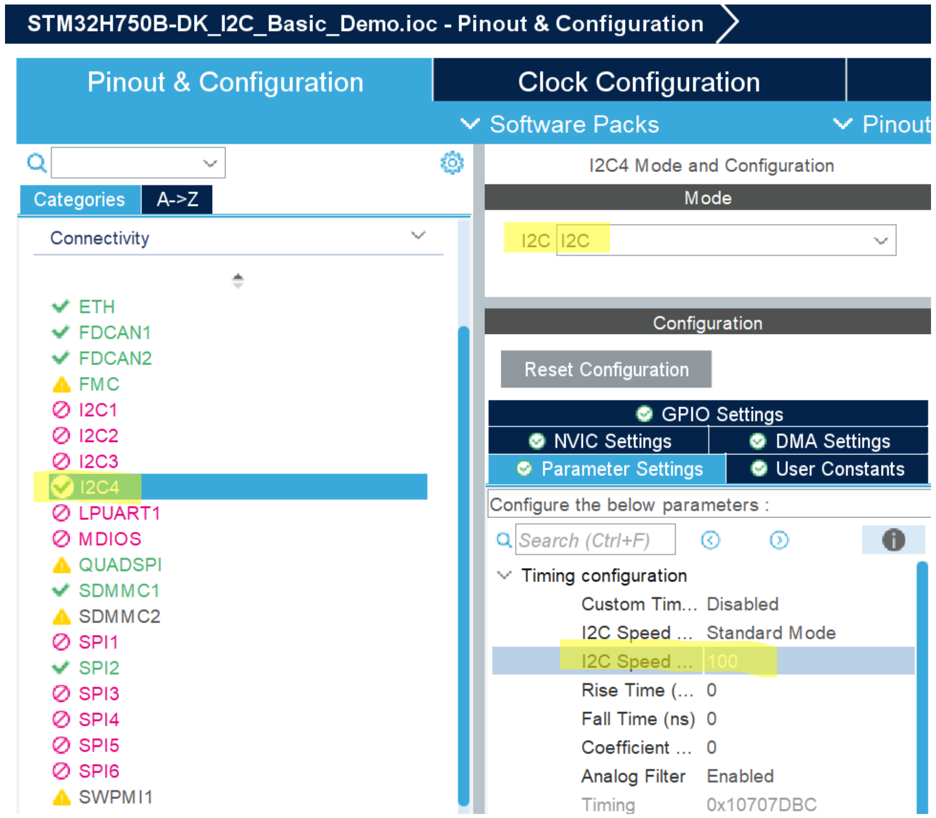


Nepravilna priključitev



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

VP 6 - STM32H7 CubeIDE – CubeMX nastavitve za I2C4



STM32H750B-DK_I2C_Basic_Demo.ioc - Pinout & Configuration

- ✓ ETH
- ✓ FDCAN1
- ✓ FDCAN2
- ⚠ FMC
- ⊘ I2C1
- ⊘ I2C2
- ⊘ I2C3
- ✓ I2C4
- ⊘ LPUART1
- ⊘ MDIOS
- ⚠ QUADSPI
- ✓ SDMMC1
- ⚠ SDMMC2
- ⊘ SPI1
- ✓ SPI2
- ⊘ SPI3
- ⊘ SPI4
- ⊘ SPI5
- ⊘ SPI6
- ⚠ SWPMI1

I2C4 Mode and Configuration

Mode

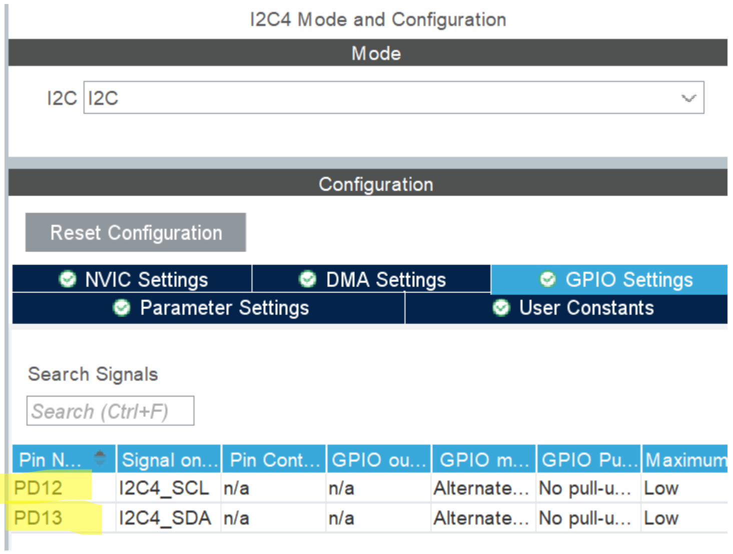
I2C: I2C

Configuration

Configure the below parameters :

Timing configuration

Custom Tim...	Disabled
I2C Speed ...	Standard Mode
I2C Speed ...	100
Rise Time (...	0
Fall Time (ns)	0
Coefficient ...	0
Analog Filter	Enabled
Timing	0x10707DBC



I2C4 Mode and Configuration

Mode

I2C: I2C

Configuration

Pin N...	Signal on...	Pin Cont...	GPIO ou...	GPIO m...	GPIO Pu...	Maximum
PD12	I2C4_SCL	n/a	n/a	Alternate...	No pull-u...	Low
PD13	I2C4_SDA	n/a	n/a	Alternate...	No pull-u...	Low

VP 6 - STM32H7 CubeIDE – I2C scanner

Spremenljivke

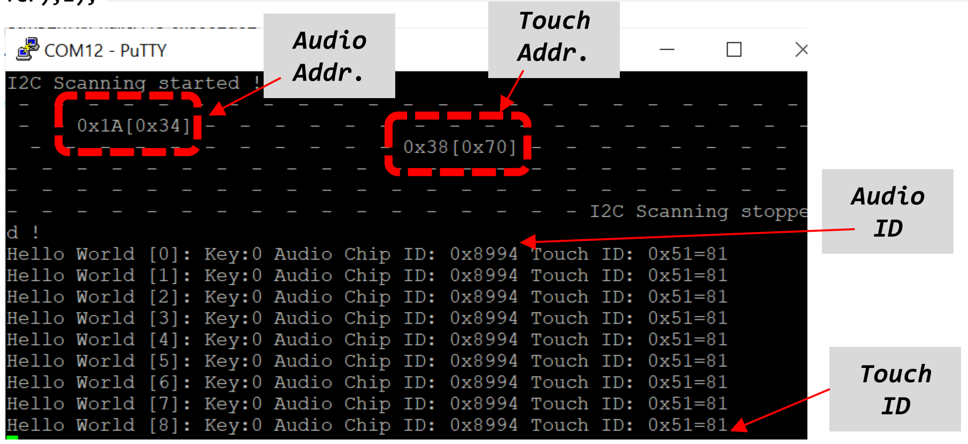
I2C Scan (vseh naprav)

main.c : dodana koda

```c
/* USER CODE BEGIN PV */
#define BUFSIZE 256
charSendBuffer[BUFSIZE];
intCounter;
int KeyState=0;
uint8_t dataBuffer[10];

HAL_StatusTypeDef retval;
uint8_t Answer;
```

```c
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, 1);     // Set LCD_RST to high

 /*-[ I2C Bus Scanning ]-*/
snprintf(SendBuffer,BUFSIZE,"I2C Scanning started !\n\r");
HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),100);

 for(i=1; i<128; i++)
 {
         retval = HAL_I2C_IsDeviceReady(&hi2c4, (uint16_t)(i<<1), 3, 5);
         if (retval != HAL_OK) /* No ACK Received At That Address */
         {
                 HAL_UART_Transmit(&huart3, Space, sizeof(Space), 100);
         }
         else if(retval == HAL_OK)
         {
                 snprintf(SendBuffer,BUFSIZE,"0x%02X[0x%02X]", i, i<<1);

                 HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),1);
         }
 }
snprintf(SendBuffer,BUFSIZE,"I2C Scanning stopped !\n\r");
HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),100);
   /*--[ Scanning Done ]--*/
```

```c
/**
* @brief Checks if target device is ready for communication.
* @note This function is used with Memory devices
* @param hi2c Pointer to a I2C_HandleTypeDef structure that contains
* the configuration information for the specified I2C.
* @param DevAddress Target device address: The device 7 bits address value
* in datasheet must be shifted to the left before calling the interface
* @param Trials Number of trials
* @param Timeout Timeout duration
* @retval HAL status
*/
HAL_StatusTypeDef HAL_I2C_IsDeviceReady(I2C_HandleTypeDef *hi2c, uint16_t
DevAddress, uint32_t Trials, uint32_t Timeout)
```



```
I2C Scanning started !
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
 -   0x1A[0x34] -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
  -  -  -  -  -  -  -  -  -  -  - 0x38[0x70] -  -  -  -  -  -  -  -
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  I2C Scanning stoppe
d !
Hello World [0]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [1]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [2]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [3]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [4]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [5]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [6]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [7]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [8]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
```

STM32H7

STM32H7

VP 6 - STM32H7 CubeIDE, I2C4 read device IDs

main.c : dodana koda

Spremenljivke

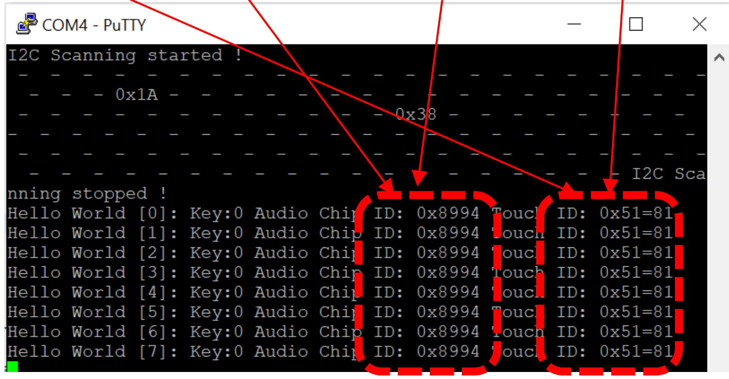
```
/* USER CODE BEGIN PV */
#define BUFSIZE 256
charSendBuffer[BUFSIZE];
intCounter;
int KeyState=0;
uint8_t dataBuffer[10];
int i=0;
uint8_t Space[] = " - ";

HAL_StatusTypeDef retval;
/* USER CODE END PV */
```

Glavna zanka

```
 /* 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);

…

// Reading from address 0x1a register R0 (addr. 0x00) default value should be 0x8994 - Both variations work !
    //dataBuffer[0] = 0; dataBuffer[1] = 0x00;
    //retval = HAL_I2C_Master_Transmit(&hi2c4, (0x1a << 1), dataBuffer, 2, HAL_MAX_DELAY);
    //retval = HAL_I2C_Master_Receive(&hi2c4, (0x1a << 1), dataBuffer, 2, HAL_MAX_DELAY);
    retval = HAL_I2C_Mem_Read(&hi2c4, (0x1a << 1), 0, I2C_MEMADD_SIZE_16BIT,dataBuffer, 2, HAL_MAX_DELAY);

// Reading from address 0x38 register Vendor's Chip ID (addr. 0xA8) default value should be 0x51=81  - Both variations work !
    //dataBuffer[5] = 0xA8;
    //retval = HAL_I2C_Master_Transmit(&hi2c4, (0x38 << 1), &dataBuffer[5], 1, HAL_MAX_DELAY);
    //retval = HAL_I2C_Master_Receive(&hi2c4, (0x38 << 1), &dataBuffer[5], 1, HAL_MAX_DELAY);
    retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0xA8, I2C_MEMADD_SIZE_8BIT,&dataBuffer[5], 1, HAL_MAX_DELAY);


    snprintf(SendBuffer,BUFSIZE,"Hello World [%d]: Key:%d Audio Chip ID: 0x%4x Touch ID: 0x%2x=%d\n\r",Counter++,KeyState,
dataBuffer[0]*256+dataBuffer[1],dataBuffer[5],dataBuffer[5]);
    HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),100);

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

    /* USER CODE BEGIN 3 */

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

Audio ID

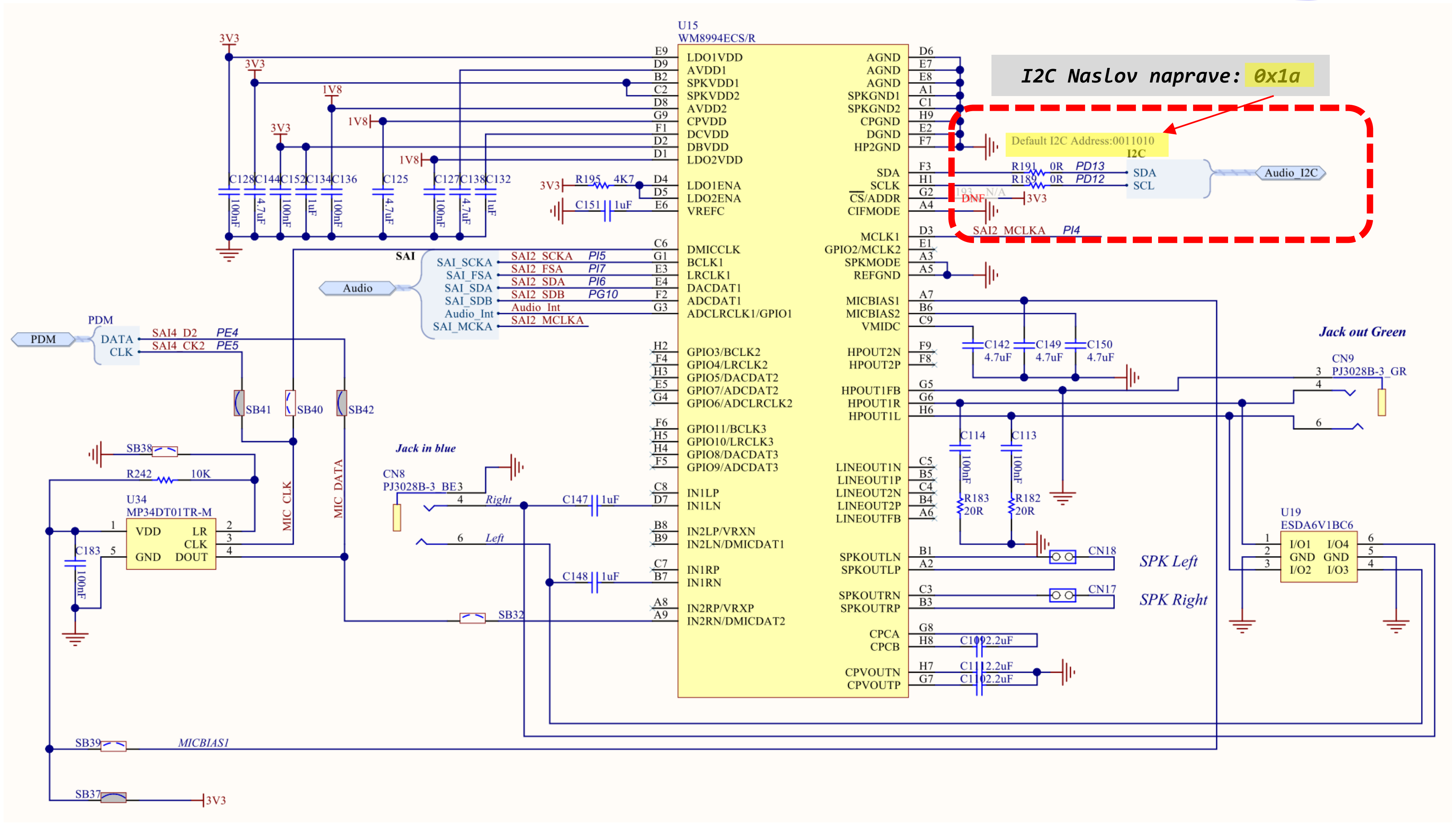
Touch ID



```
I2C Scanning started !
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
  -  -  - 0x1A -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
 -  -  -  -  -  -  -  -  -  -  -  -  - 0x38 -  -  -  -  -  -  -
 -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - I2C Sca
nning stopped !
Hello World [0]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [1]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [2]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [3]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [4]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [5]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [6]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
Hello World [7]: Key:0 Audio Chip ID: 0x8994 Touch ID: 0x51=81
```

VP 6 - STM32H7 I2C Audio WM9884 – vezalna shema

STM32H7



https://github.com/LAPSyLAB/STM32H750B-DK_Docs_and_Examples/blob/main/Documentation/STM32H750B-DK/en.MB1381-H750XB-B01_Schematic.pdf

VP 6 - Audio čip WM8994 na I2C4 vodilu

Gradivo

CIRRUS LOGIC®

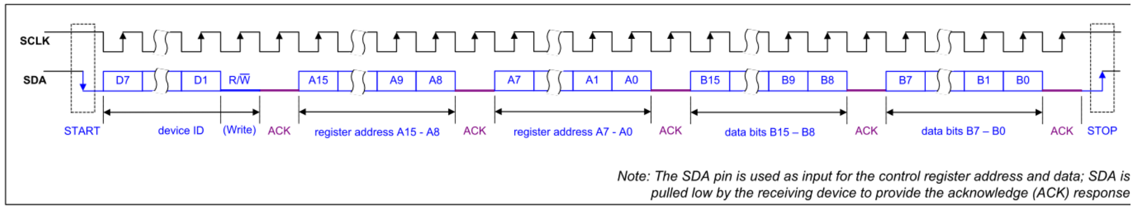
WM8994

Multi-channel Audio Hub CODEC for Smartphones

`16-bitni naslovi in 16-bitni registri !`

2-WIRE (I2C) CONTROL MODE

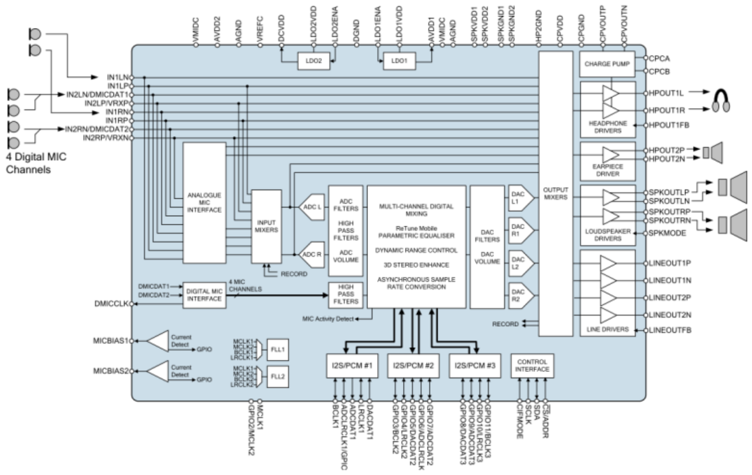
The sequence of signals associated with a single register write operation is illustrated in Figure 72.



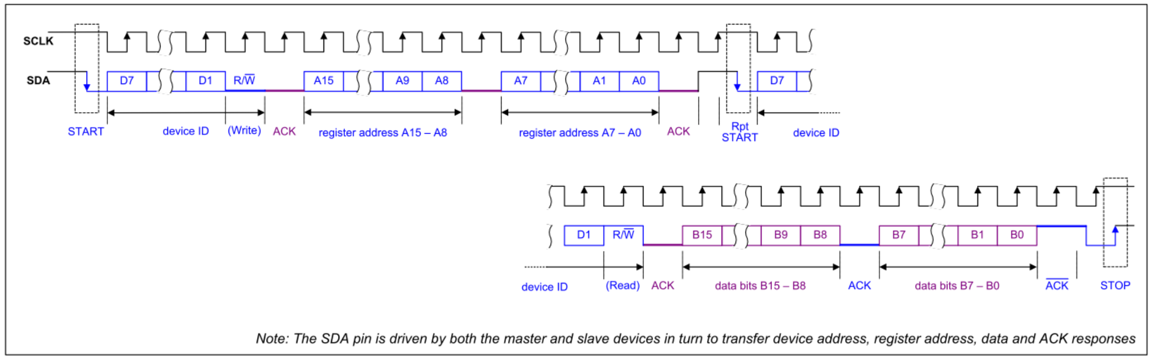
Note: The SDA pin is used as input for the control register address and data; SDA is pulled low by the receiving device to provide the acknowledge (ACK) response

[Brez naslova]

Figure 72 Control Interface 2-wire (I2C) Register Write



The sequence of signals associated with a single register read operation is illustrated in Figure 73.



Note: The SDA pin is driven by both the master and slave devices in turn to transfer device address, register address, data and ACK responses

Figure 73 Control Interface 2-wire (I2C) Register Read

https://github.com/LAPSyLAB/STM32H750B-DK_Docs_and_Examples/blob/main/Documentation/STM32H750B-DK/Audio/WM8994_Rev4.6-unlocked.pdf

STM32H7

REGISTER BITS BY ADDRESS

REGISTER ADDRESS	BIT	LABEL	DEFAULT	DESCRIPTION	REFER TO
R0 (00h) Software Reset	15:0	SW_RESET [15:0]	0000_0000 _0000_000 0	Writing to this register resets all registers to their default state. (Note - Control Write Sequencer registers are not affected by Software Reset.) Reading from this register will indicate device family ID 8994h.	

Register 00h Software Reset

VP 6 - STM32H7 I2C4 read audio device ID

Spremenljivke

main.c : dodana koda

```
// Reading from address 0x1a register R0 (addr. 0x00) default value should be 0x8994 - Both variations work !
//dataBuffer[0] = 0; dataBuffer[1] = 0x00;
//retval = HAL_I2C_Master_Transmit(&hi2c4, (0x1a << 1), dataBuffer, 2, HAL_MAX_DELAY);
//retval = HAL_I2C_Master_Receive(&hi2c4, (0x1a << 1), dataBuffer, 2, HAL_MAX_DELAY);
retval = HAL_I2C_Mem_Read(&hi2c4, (0x1a << 1), 0, I2C_MEMADD_SIZE_16BIT,dataBuffer, 2, HAL_MAX_DELAY);

snprintf(SendBuffer,BUFSIZE,"Hello World [%d]: Key:%d Audio Chip ID: 0x%4x Touch ID: 0x%2x=%d\n\r",Counter++,KeyState,
dataBuffer[0]*256+dataBuffer[1],dataBuffer[5],dataBuffer[5]);
HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),100);
```

Glavna zanka

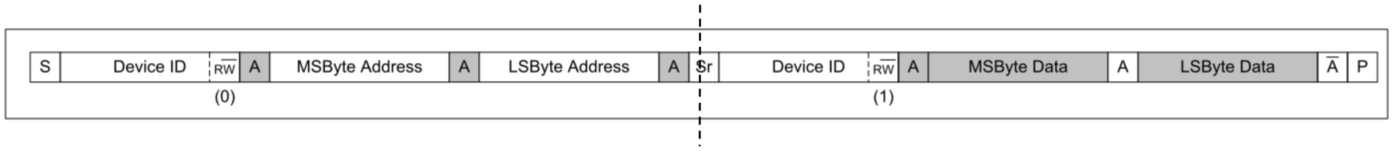
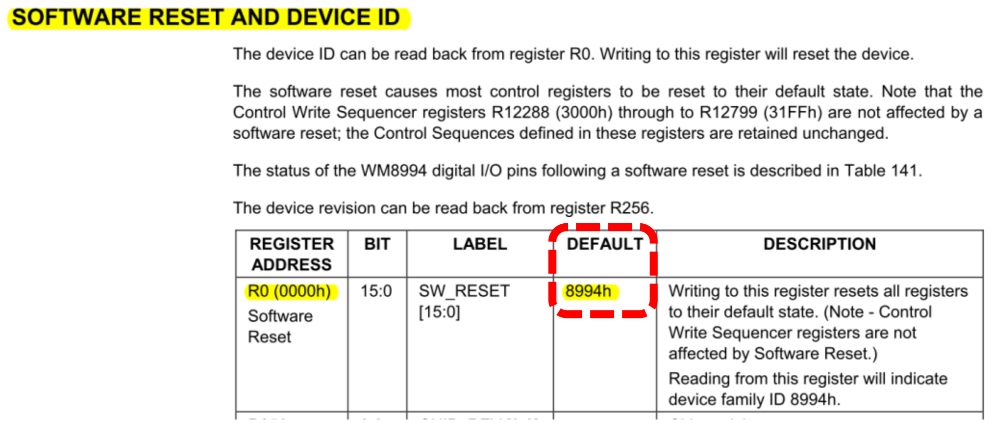


Figure 75 Single Register Read from Specified Address



SOFTWARE RESET AND DEVICE ID

The device ID can be read back from register R0. Writing to this register will reset the device.

The software reset causes most control registers to be reset to their default state. Note that the Control Write Sequencer registers R12288 (3000h) through to R12799 (31FFh) are not affected by a software reset; the Control Sequences defined in these registers are retained unchanged.

The status of the WM8994 digital I/O pins following a software reset is described in Table 141.

The device revision can be read back from register R256.

REGISTER ADDRESS	BIT	LABEL	DEFAULT	DESCRIPTION
R0 (0000h) Software Reset	15:0	SW_RESET [15:0]	8994h	Writing to this register resets all registers to their default state. (Note - Control Write Sequencer registers are not affected by Software Reset.) Reading from this register will indicate device family ID 8994h.

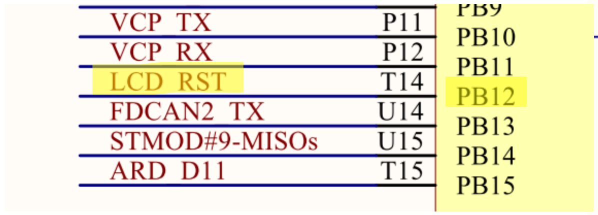
STM32H7

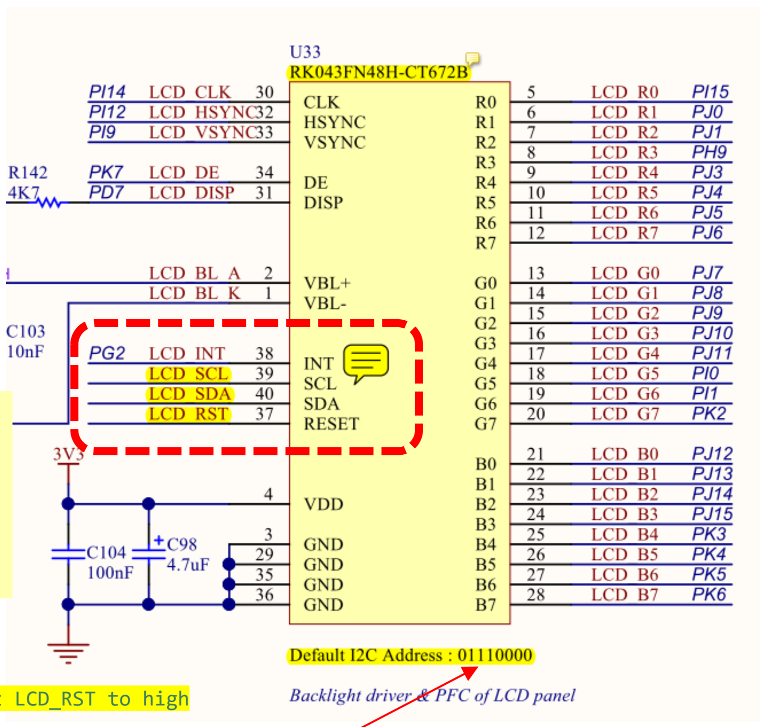
https://github.com/LAPSyLAB/STM32H7_Discovery_VIN_Projects/tree/main/STM32H750B-DK_I2C_Basic_Demo

VP 6 - STM32H7 I2C LCD-Touch RK043FN48H – vezalna shema

STM32H7

LCD RST mora biti 1,
da naprava deluje





```
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, 1); // Set LCD_RST to high
```

I2C Naslov naprave: 0x38 (ali 0x70 s pomikom na levo – sprostimo prostor za R/W bit)

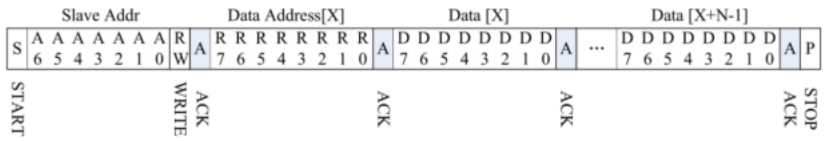
https://github.com/LAPSyLAB/STM32H750B-DK_Docs_and_Examples/blob/main/Documentation/STM32H750B-DK/en.MB1381-H750XB-B01_Schematic.pdf

VP 4 - STM32H7 CubeIDE, I2C LCD-Touch RK043FN48H

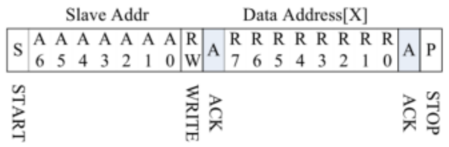
8-bitni naslovi in 8-bitni registri !

1.2 I²C Read/Write Interface description

Write N bytes to I2C slave


Slave Addr
Data Address[X]
Data [X]
Data [X+N-1]
S A6 A5 A4 A3 A2 A1 A0 R/W A R7 R6 R5 R4 R3 R2 R1 R0 A D7 D6 D5 D4 D3 D2 D1 D0 A … D7 D6 D5 D4 D3 D2 D1 D0 A P
START
WRITE
ACK
ACK
ACK
ACK
STOP

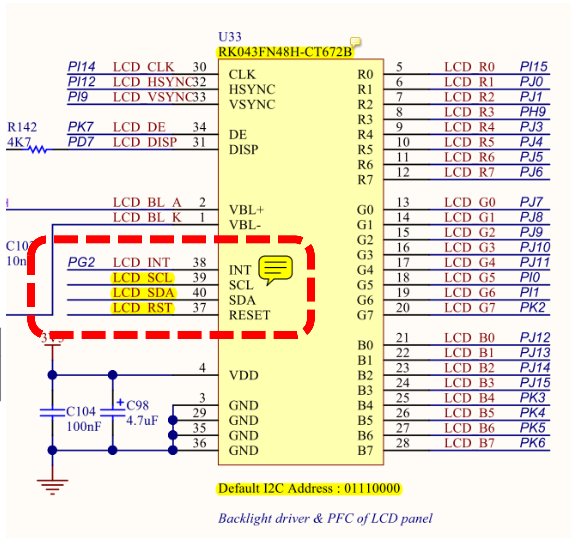
Set Data Address



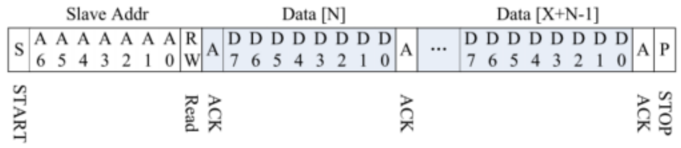
2.1.26 ID_G_FT5201ID

This register describes vendor's chip id

Address	Bit Address	Register Name	Description
A8h	7:0	ID_G_ FT5201ID	R: xx



Read X bytes from I²C Slave


Slave Addr
Data [N]
Data [X+N-1]
S A6 A5 A4 A3 A2 A1 A0 R/W A D7 D6 D5 D4 D3 D2 D1 D0 A … D7 D6 D5 D4 D3 D2 D1 D0 A P
START
Read
ACK
ACK
ACK
STOP

2.1 Work Mode

In this mode the CTP is fully functional as a touch screen controller. Read and write access address is ju logical address which is not enforced by hardware or firmware. Here is the operating mode register map.

Work Mode Register Map

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Host Access
00h	DEVIDE_MODE		Device Mode[2:0]							RW

Device Mode	Val	Description
Work	000b	Read touch point and gesture
Factory	100b	Read raw data

STM32H7

https://github.com/LAPSyLAB/STM32H7_Discovery_VIN_Projects/tree/main/STM32H750B-DK_I2C_Basic_Demo

VP 6 - STM32H7 I2C4 read touch device ID

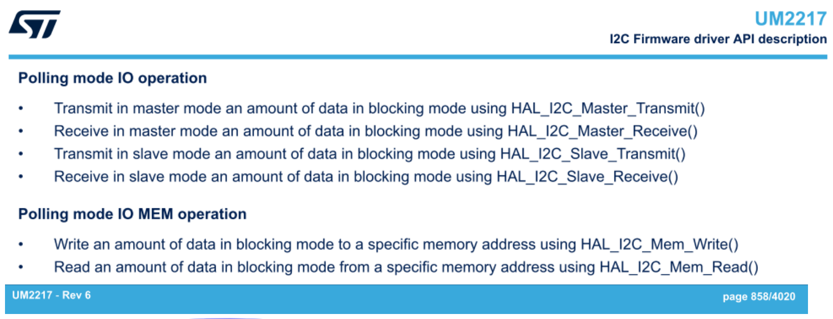
Spremenljivke

```
/* USER CODE BEGIN PV */
#define BUFSIZE 256
charSendBuffer[BUFSIZE];
intCounter;
int KeyState=0;
uint8_t dataBuffer[10];

HAL_StatusTypeDef retval;
/* USER CODE END PV */
```

main.c : dodana koda

```
// Reading from address 0x38 register Vendor's Chip ID (addr. 0xA8) default value should be 0x51=81  - Both variations work !
   //dataBuffer[5] = 0xA8;
   //retval = HAL_I2C_Master_Transmit(&hi2c4, (0x38 << 1), &dataBuffer[5], 1, HAL_MAX_DELAY);
   //retval = HAL_I2C_Master_Receive(&hi2c4, (0x38 << 1), &dataBuffer[5], 1, HAL_MAX_DELAY);
   retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0xA8, I2C_MEMADD_SIZE_8BIT,&dataBuffer[5], 1, HAL_MAX_DELAY);


   snprintf(SendBuffer,BUFSIZE,"Hello World [%d]: Key:%d Audio Chip ID: 0x%4x Touch ID: 0x%2x=%d\n\r",Counter++,KeyState,
dataBuffer[0]*256+dataBuffer[1],dataBuffer[5],dataBuffer[5]);
   HAL_UART_Transmit(&huart3,SendBuffer,strlen(SendBuffer),100);

   HAL_Delay(1000);
```

Glavna zanka


UM2217
I2C Firmware driver API description

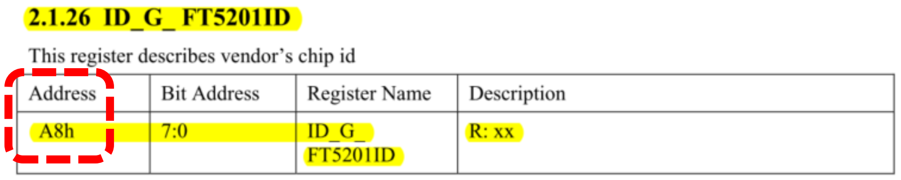
Polling mode IO operation

- Transmit in master mode an amount of data in blocking mode using HAL_I2C_Master_Transmit()
- Receive in master mode an amount of data in blocking mode using HAL_I2C_Master_Receive()
- Transmit in slave mode an amount of data in blocking mode using HAL_I2C_Slave_Transmit()
- Receive in slave mode an amount of data in blocking mode using HAL_I2C_Slave_Receive()

Polling mode IO MEM operation

- Write an amount of data in blocking mode to a specific memory address using HAL_I2C_Mem_Write()
- Read an amount of data in blocking mode from a specific memory address using HAL_I2C_Mem_Read()

UM2217 - Rev 6 page 858/4020


2.1.26 ID_G_ FT5201ID

This register describes vendor's chip id

Address	Bit Address	Register Name	Description
A8h	7:0	ID_G_ FT5201ID	R: xx

STM32H7

https://github.com/LAPSyLAB/STM32H7_Discovery_VIN_Projects/tree/main/STM32H750B-DK_I2C_Basic_Demo

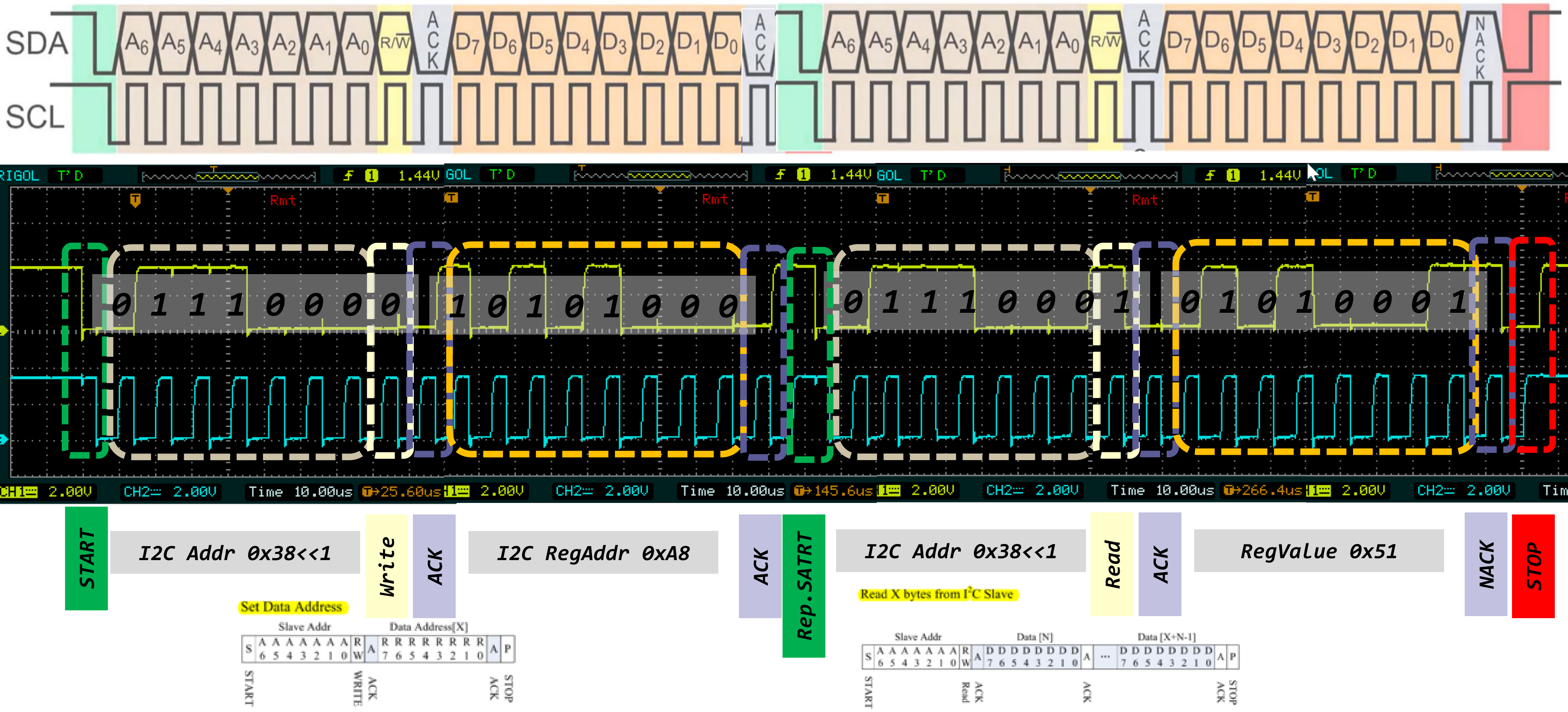
STM32H7

VP 4 - STM32H7 CubeIDE, I2C4 branje

I2C branje

main.c : dodana koda

```
// Reading from address 0x38 register Vendor's Chip ID (addr. 0xA8) default value should be 0x51=81

retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0xA8, I2C_MEMADD_SIZE_8BIT,&dataBuffer[5], 1, HAL_MAX_DELAY);
```



https://github.com/LAPSyLAB/STM32H7_Discovery_VIN_Projects/tree/main/STM32H750B-DK_I2C_Basic_Demo

Primer I2C komunikacije STM32H7 – Touch: zaznava dotika

FocalTech **FT5336GQQ**

True Multi-Touch Capacitive Touch Panel Controller

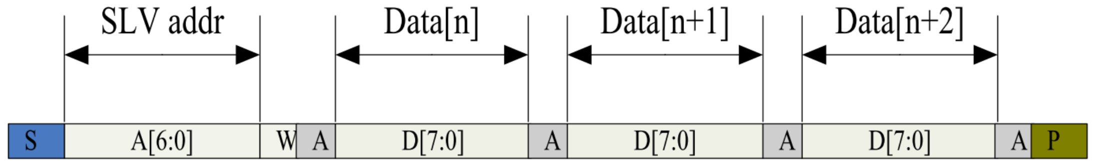


Figure 2-5 I2C master write, slave read

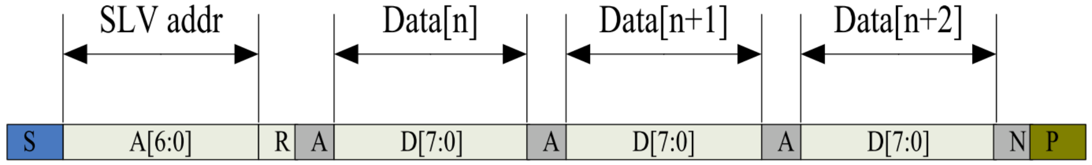


Figure 2-6 I2C master read, slave write

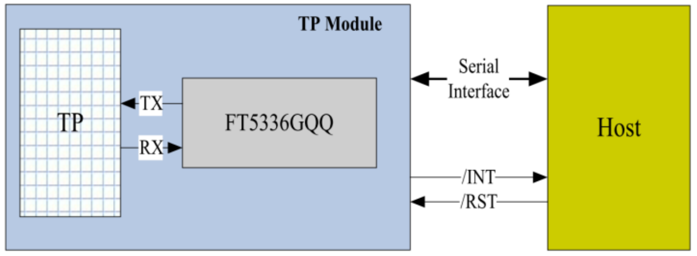
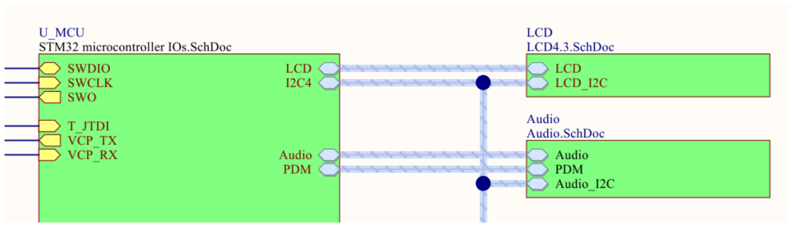
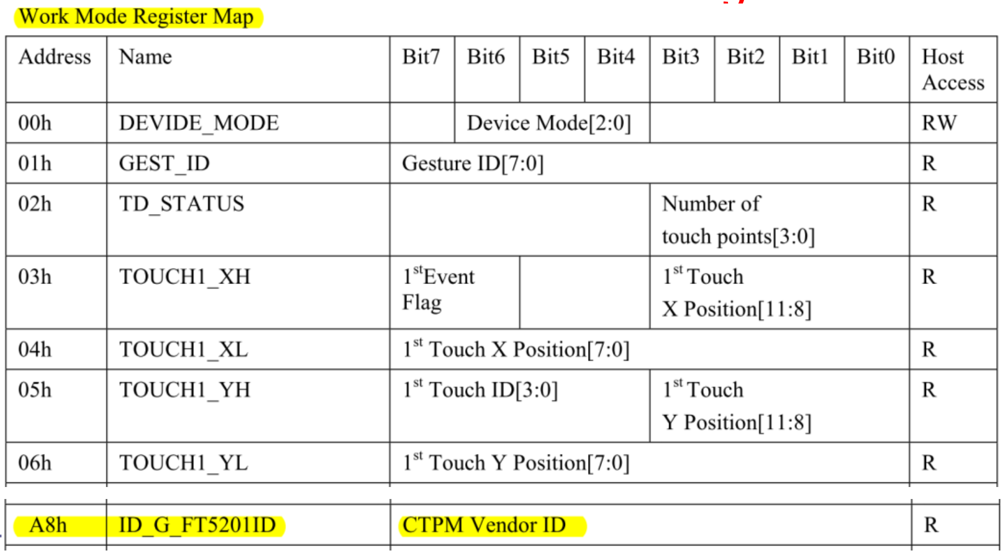


Figure 2-3 Host Interface Diagram

https://github.com/LAPSyLAB/STM32H7_Discovery_VIN_Projects/tree/main/STM32H750B-DK_I2C_Touch_Demo

8-bitni naslovi in registri





Work Mode Register Map

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Host Access
00h	DEVIDE_MODE		Device Mode[2:0]							RW
01h	GEST_ID	Gesture ID[7:0]								R
02h	TD_STATUS					Number of touch points[3:0]				R
03h	TOUCH1_XH	1stEvent Flag				1st Touch X Position[11:8]				R
04h	TOUCH1_XL	1st Touch X Position[7:0]								R
05h	TOUCH1_YH	1st Touch ID[3:0]				1st Touch Y Position[11:8]				R
06h	TOUCH1_YL	1st Touch Y Position[7:0]								R
A8h	ID_G_FT5201ID	CTPM Vendor ID								R

STM32H7

STM32H7

Primer I2C komunikacije STM32H7 – Touch: zaznava dotika

```
// Reading from address 0x38 register Vendor's Chip ID (addr. 0xA8) default value should be 0x51=81

retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0xA8, I2C_MEMADD_SIZE_8BIT,&VendorID, 1, HAL_MAX_DELAY);

retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0x00, I2C_MEMADD_SIZE_8BIT,&DeviceMode, 1, HAL_MAX_DELAY);
retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0x01, I2C_MEMADD_SIZE_8BIT,&Gesture, 1, HAL_MAX_DELAY);
retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0x02, I2C_MEMADD_SIZE_8BIT,&Status, 1, HAL_MAX_DELAY);

retval = HAL_I2C_Mem_Read(&hi2c4, (0x38 << 1), 0x03, I2C_MEMADD_SIZE_8BIT,&dataBuffer, 4, HAL_MAX_DELAY);
if (Status != 0) {
      TouchX = ( (dataBuffer[0] & 0b1111) << 8) + dataBuffer[1];
      TouchY = ( (dataBuffer[2] & 0b1111) << 8) + dataBuffer[3];
} else {
      TouchX = 0;
      TouchY = 0;
}
```

8-bitni naslovi in registri

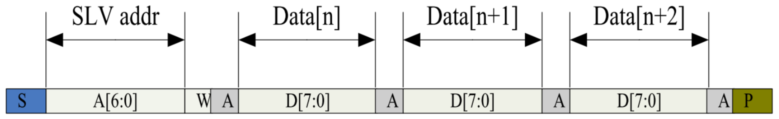


Figure 2-5 I2C master write, slave read

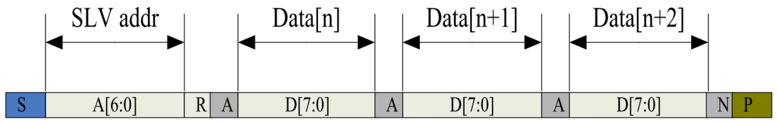
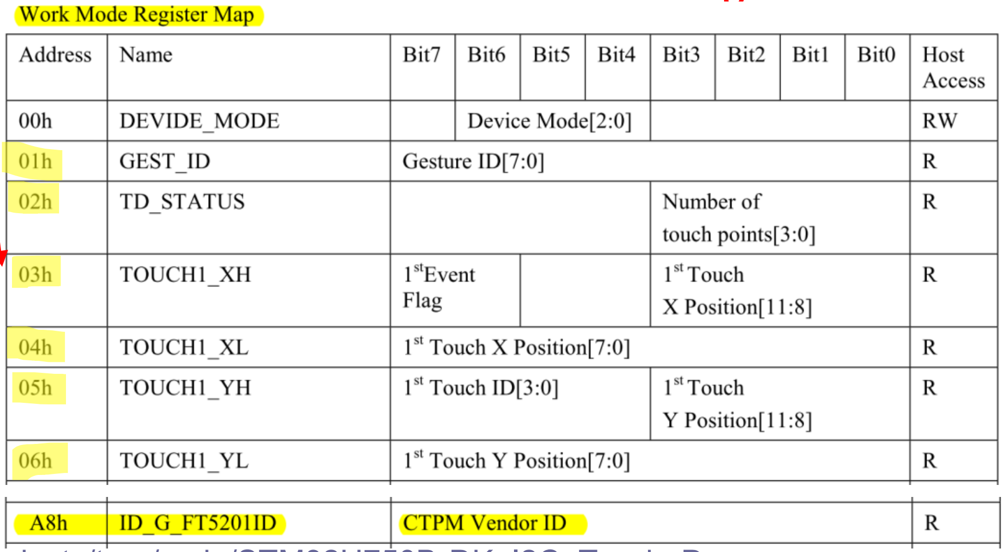


Figure 2-6 I2C master read, slave write



Work Mode Register Map

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Host Access
00h	DEVIDE_MODE		Device Mode[2:0]							RW
01h	GEST_ID	Gesture ID[7:0]								R
02h	TD_STATUS					Number of touch points[3:0]				R
03h	TOUCH1_XH	1st Event Flag				1st Touch X Position[11:8]				R
04h	TOUCH1_XL	1st Touch X Position[7:0]								R
05h	TOUCH1_YH	1st Touch ID[3:0]				1st Touch Y Position[11:8]				R
06h	TOUCH1_YL	1st Touch Y Position[7:0]								R
A8h	ID_G_FT5201ID	CTPM Vendor ID								R

https://github.com/LAPSyLAB/STM32H7_Discovery_VIN_Projects/tree/main/STM32H750B-DK_I2C_Touch_Demo

VP 6: STM32H7 I2C primeri, STM32F4 SPI, pospeškomer, „Air USB mouse“

- STM32H7 CubeIDE, I2C (Scan, WM9884, Touch)
- Predstavitev STM32F4
- CubeIDE projekt STM32F4, SPI, pospeškomer:
  - SPI - LIS3DSH pospeškomer
  - USB „Air Mouse“ with STM32F4, SPI1 and LIS3DSH
- Sledenje („tracing“) – CubeMonitor
- Osciloskop - ponovitev

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:

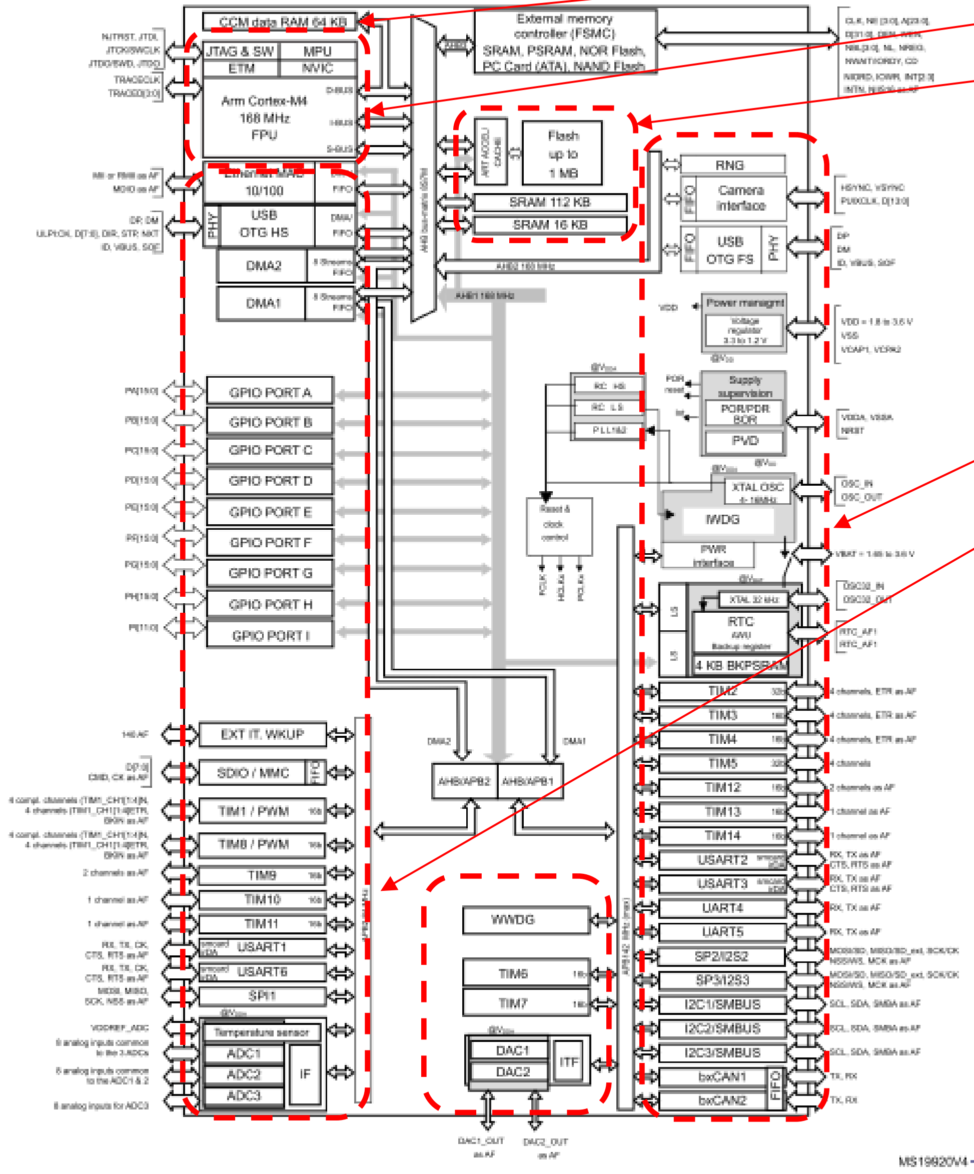
- •USB with Micro-AB
- •Stereo headphone output jack
- •2.54 mm pitch extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing

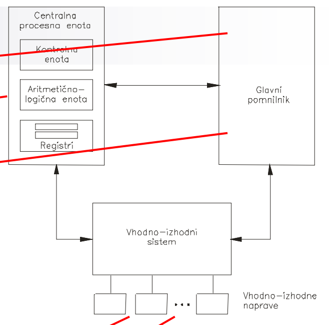
•External application power supply: 3 V and 5 V

STM32

STM32F4

STM32F407VG

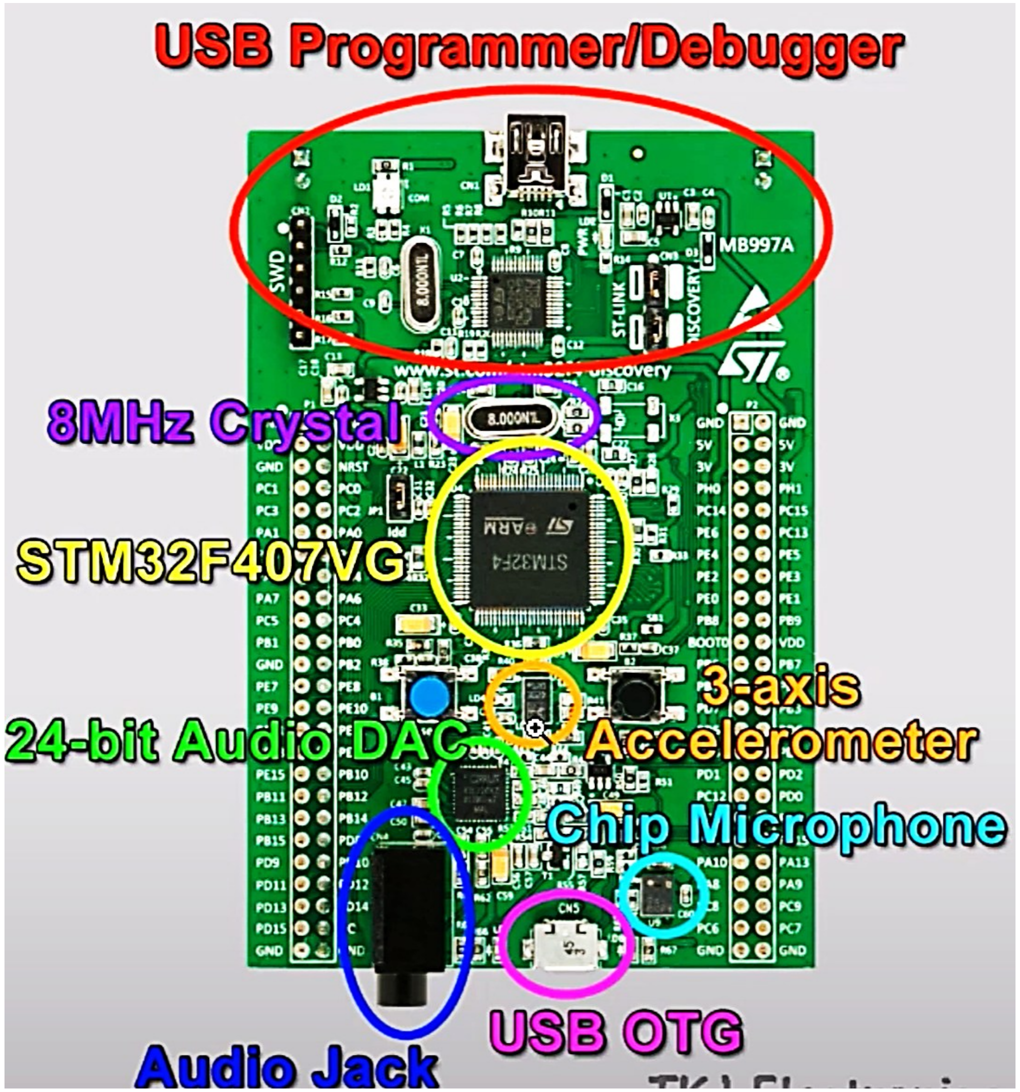




STM32F4

STM32F4DISCOVERY

3.3V !!!



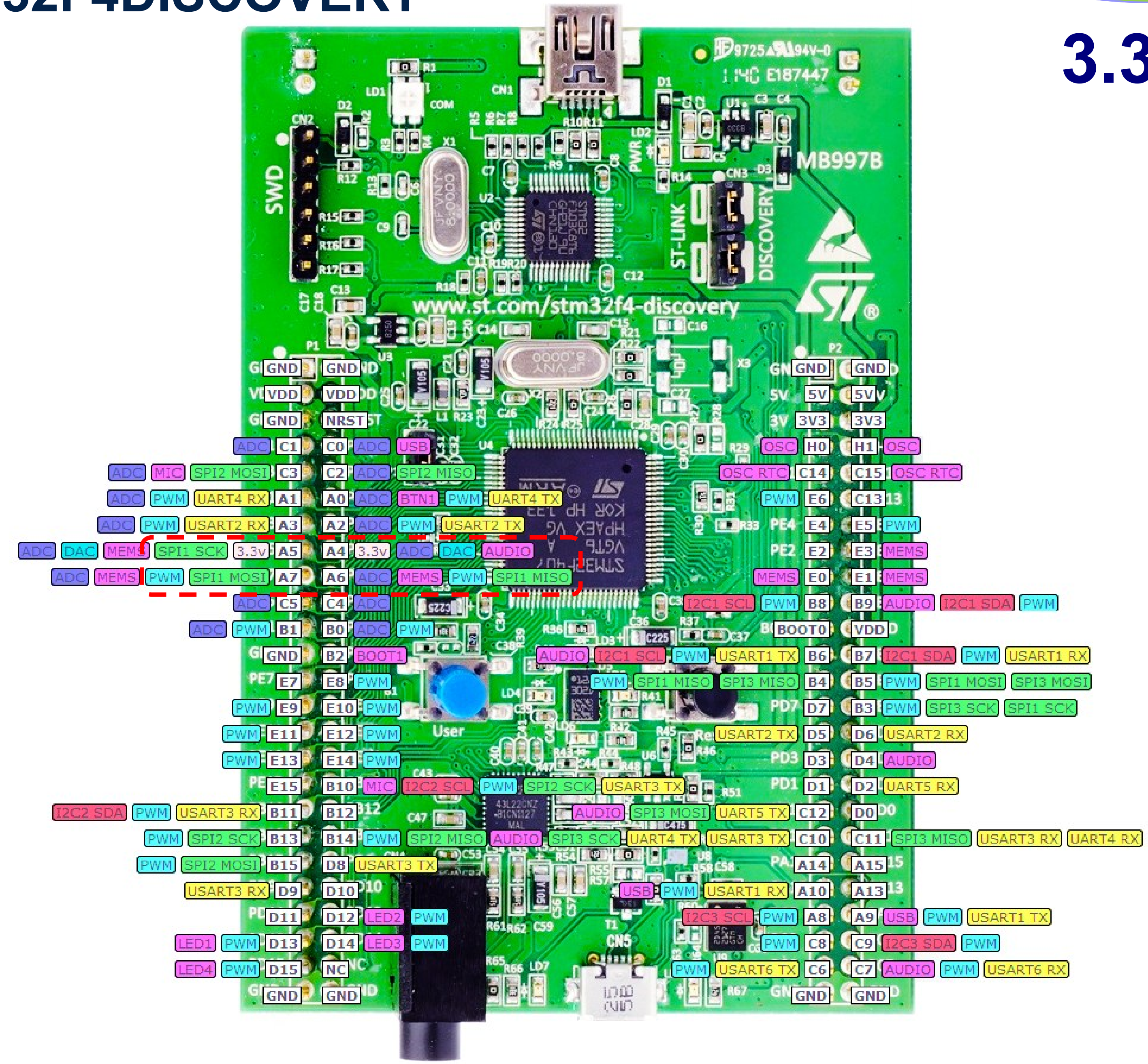
STM32F4DISCOVERY

STM32F4

3.3V !!!

P1

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50



GND GND
VDD VDD
GND NRST
ADC C1 C0 ADC USB
ADC MIC SPI2 MOSI C3 C2 ADC SPI2 MISO
ADC PWM UART4 RX A1 A0 ADC BTN1 PWM UART4 TX
ADC PWM USART2 RX A3 A2 ADC PWM USART2 TX
ADC DAC MEMS SPI1 SCK 3.3v A5 A4 3.3v ADC DAC AUDIO
ADC MEMS PWM SPI1 MOSI A7 A6 ADC MEMS PWM SPI1 MISO
ADC C5 C4 ADC
ADC PWM B1 B0 ADC PWM
GND B2 BOOT1
E7 E8 PWM
PWM E9 E10 PWM
PWM E11 E12 PWM
PWM E13 E14 PWM
E15 B10 MIC I2C2 SCL PWM SPI2 SCK USART3 TX
I2C2 SDA PWM USART3 RX B11 B12
PWM SPI2 SCK B13 B14 PWM SPI2 MISO
PWM SPI2 MOSI B15 D8 USART3 TX
USART3 RX D9 D10
D11 D12 LED2 PWM
LED1 PWM D13 D14 LED3 PWM
LED4 PWM D15 NC
GND GND
GND GND
5V 5V
3V3 3V3
OSC H0 H1 OSC
OSC RTC C14 C15 OSC RTC
PWM E6 C13
E4 E5 PWM
E2 E3 MEMS
MEMS E0 E1 MEMS
I2C1 SCL PWM B8 B9 AUDIO I2C1 SDA PWM
BOOT0 VDD
AUDIO I2C1 SCL PWM USART1 TX B6 B7 I2C1 SDA PWM USART1 RX
PWM SPI1 MISO SPI3 MISO B4 B5 PWM SPI1 MOSI SPI3 MOSI
D7 B3 PWM SPI3 SCK SPI1 SCK
USART2 TX D5 D6 USART2 RX
D3 D4 AUDIO
D1 D2 UART5 RX
AUDIO SPI3 MOSI UART5 TX C12 D0
AUDIO SPI3 SCK UART4 TX USART3 TX C10 C11 SPI3 MISO USART3 RX UART4 RX
A14 A15
USB PWM USART1 RX A10 A13
I2C3 SCL PWM A8 A9 USB PWM USART1 TX
PWM C8 C9 I2C3 SDA PWM
PWM USART6 TX C6 C7 AUDIO PWM USART6 RX
GND GND
SWD
ST-LINK
DISCOVERY
MB997B
www.st.com/stm32f4-discovery
User

P2

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50

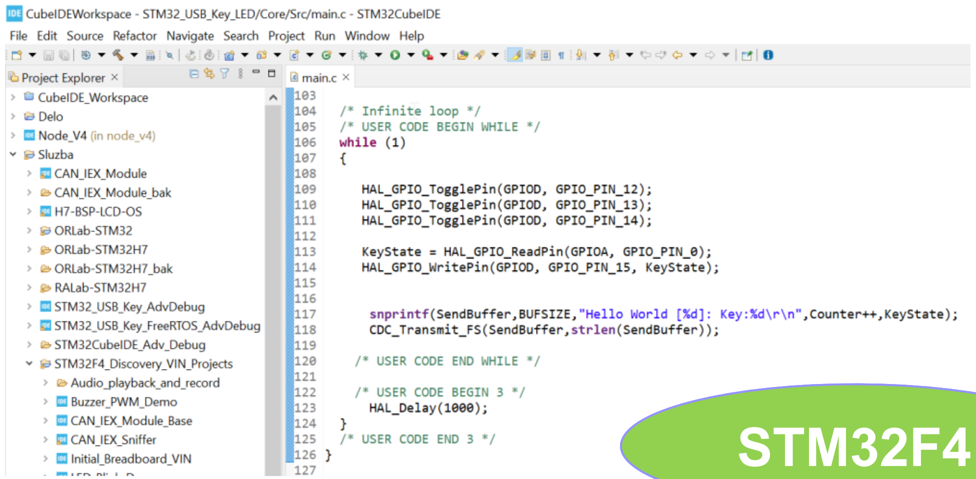
Delo na STM32F4 razvojnem sistemu

Priključitev :

- **Mini USB** priklop na **krajši stranici**, svetita rdeči **LED** diodi
    - napajanje, debug…
- **Micro USB** priklop (VCom port)

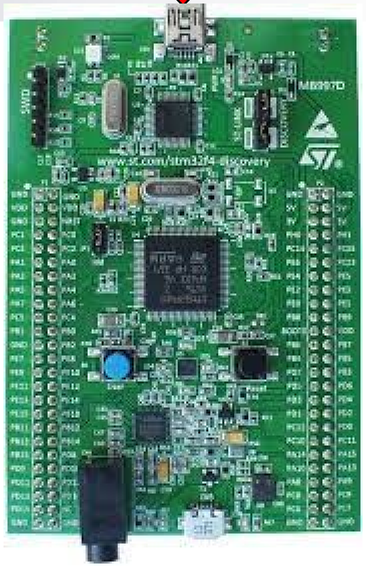
Poseben začetni projekt za *STM32F4* (Github) :

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


CubeIDEWorkspace - STM32_USB_Key_LED/Core/Src/main.c - STM32CubeIDE

```c

  /* 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 */
}

```

STM32F4



Mikro USB
VCom-port

Lastni viri :

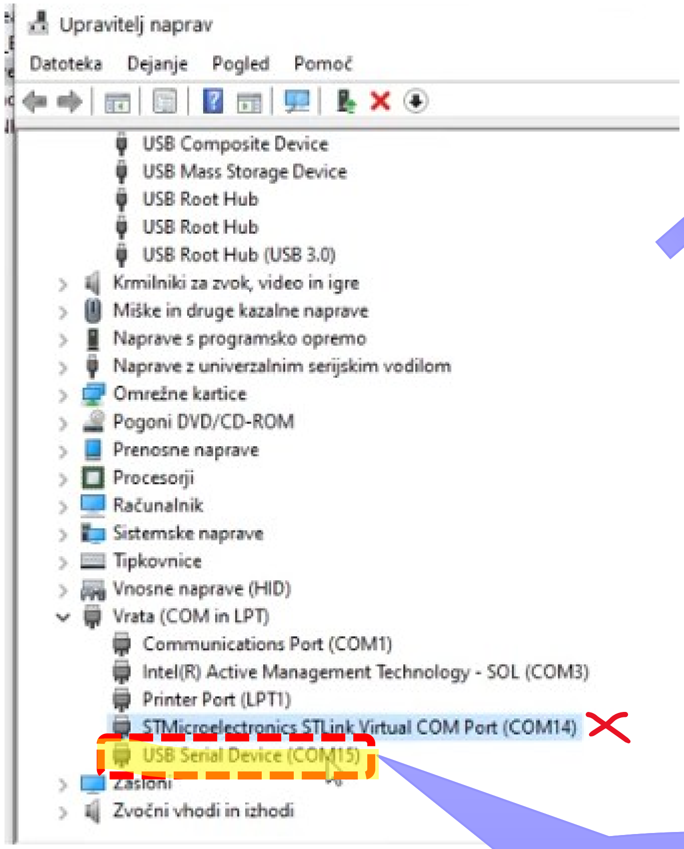
https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects

https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples

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

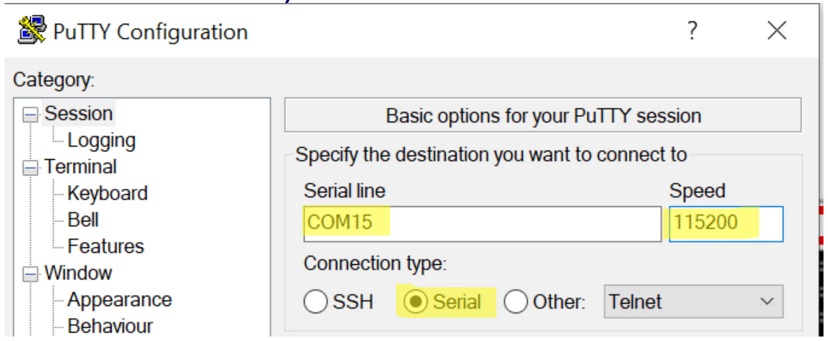
Osnovni projekt CubeIDE – USB Virtual COM Port

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



Upravitelj naprav

- USB Composite Device
- USB Mass Storage Device
- USB Root Hub
- USB Root Hub
- USB Root Hub (USB 3.0)
- Krmilniki za zvok, video in igre
- Miške in druge kazalne naprave
- Naprave s programsko opremo
- Naprave z univerzalnim serijskim vodilom
- Omrežne kartice
- Pogoni DVD/CD-ROM
- Prenosne naprave
- Procesorji
- Računalnik
- Sistemske naprave
- Tipkovnice
- Vnosne naprave (HID)
- Vrata (COM in LPT)
  - Communications Port (COM1)
  - Intel(R) Active Management Technology - SOL (COM3)
  - Printer Port (LPT1)
  - STMicroelectronics STLink Virtual COM Port (COM14)
  - USB Serial Device (COM15)
- Zasloni
- Zvočni vhodi in izhodi



PuTTY Configuration

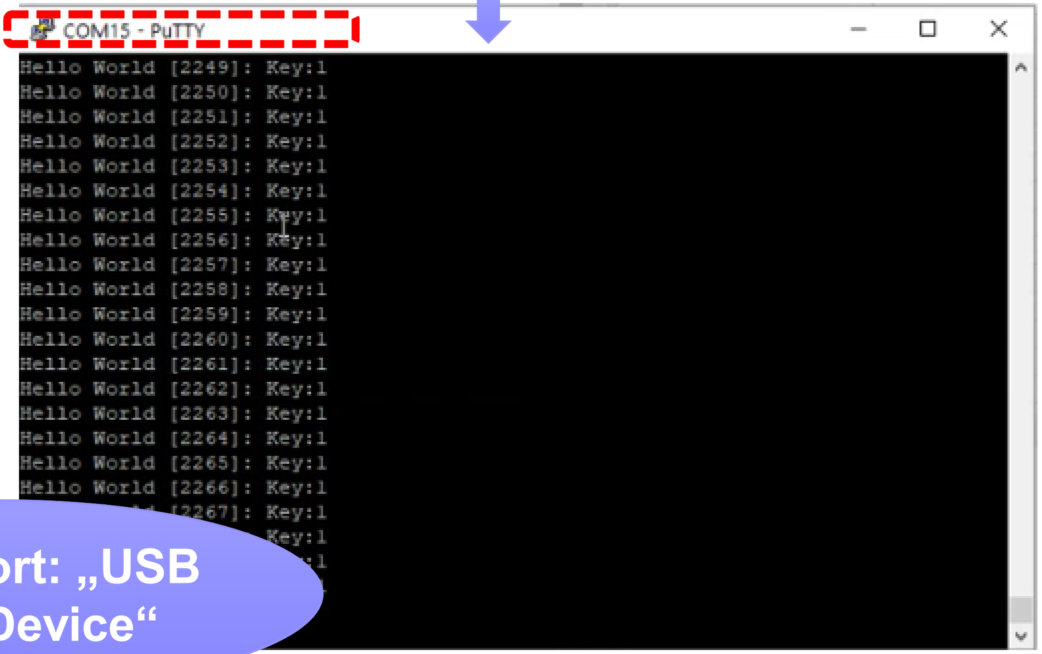
Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM15

Speed: 115200

Connection type: SSH [✓ Serial] Other: Telnet



COM15 - PuTTY

```
Hello World [2249]: Key:1
Hello World [2250]: Key:1
Hello World [2251]: Key:1
Hello World [2252]: Key:1
Hello World [2253]: Key:1
Hello World [2254]: Key:1
Hello World [2255]: Key:1
Hello World [2256]: Key:1
Hello World [2257]: Key:1
Hello World [2258]: Key:1
Hello World [2259]: Key:1
Hello World [2260]: Key:1
Hello World [2261]: Key:1
Hello World [2262]: Key:1
Hello World [2263]: Key:1
Hello World [2264]: Key:1
Hello World [2265]: Key:1
Hello World [2266]: Key:1
            [2267]: Key:1
                    Key:1
```

VCOM port: „USB Serial Device“

VP 6: STM32H7 I2C primeri, STM32F4 SPI, pospeškomer, „Air USB mouse“

- STM32H7 CubeIDE, I2C (Scan, WM9884, Touch)
- Predstavitev STM32F4
- CubeIDE projekt STM32F4, SPI, pospeškomer:
  - SPI - LIS3DSH pospeškomer
  - USB „Air Mouse“ with STM32F4, SPI1 and LIS3DSH
- Sledenje („tracing“) – CubeMonitor
- Osciloskop - ponovitev

VP 6 - STM32 CubeIDE, SPI in LIS3DSH Gradiva

5 Digital main blocks

5.1 State machine

The LIS3DSH embeds two state machines able to run a user defined program.

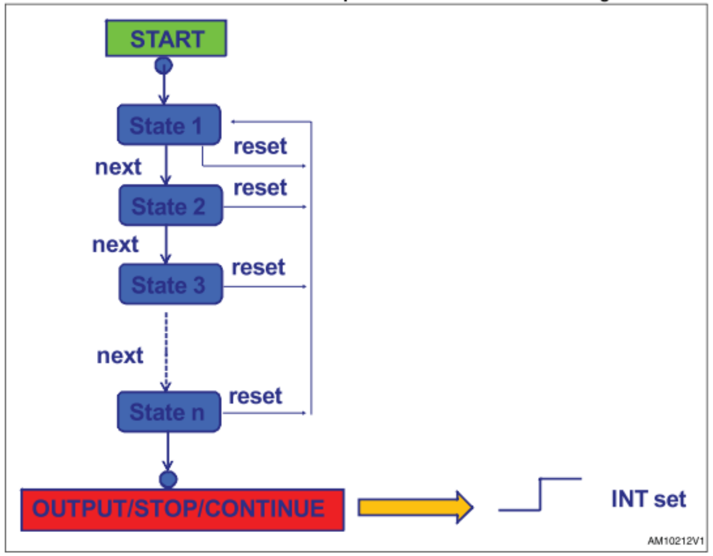
The program is made up of a set of instructions that define the transition to successive states. Conditional branches are possible.

From each state (n) it is possible to have transition to the next state (n+1) or to reset state. Transition to reset point happens when “RESET condition” is true; Transition to the next step happens when “NEXT condition” is true.

Interrupt is triggered when output/stop/continue state is reached.

Each state machine allows to implement gesture recognition in a flexible way, free-fall, wake-up, 4D/6D orientation, pulse counter and step recognition, click/double click, shake/double shake, face-up/face-down, turn/double turn:

Table 8. LIS3DSH state machines: sequence of state to execute an algorithm



SPI - serial peripheral interface

Subject to general operating conditions for Vdd and Top.

SPI slave timing values

ol	Parameter	Value (1)		Unit
		Min.	Max.	
)	SPI clock cycle	100		ns
)	SPI clock frequency		10	MHz
)	CS setup time	6		

I²C - inter IC control interface

Subject to general operating conditions for Vdd and Top.

I²C slave timing values

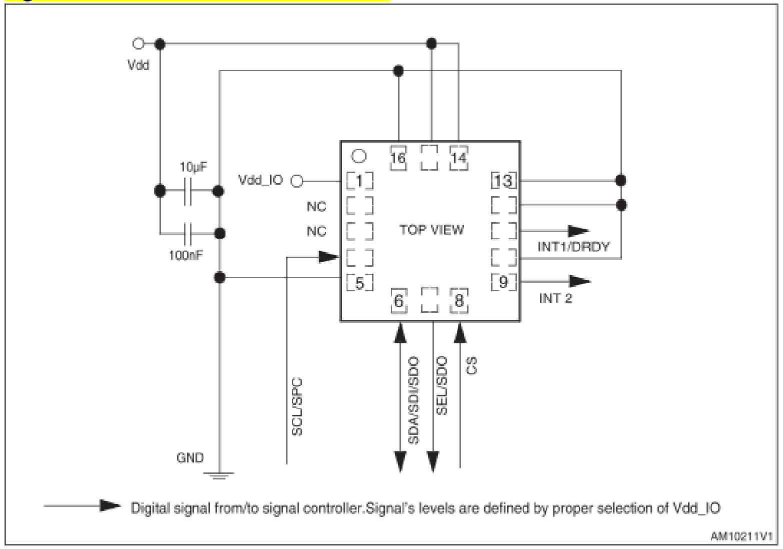
	Parameter	I²C standard mode (1)		I² fast mode (1)		Unit
		Min.	Max.	Mi	Max.	
	SCL clock frequency	0	100	0	400	kHz

Table 7. Absolute maximum ratings

Symbol	Ratings	Maximum value	Unit
Vdd	Supply voltage	-0.3 to 4.8	V

Application hints

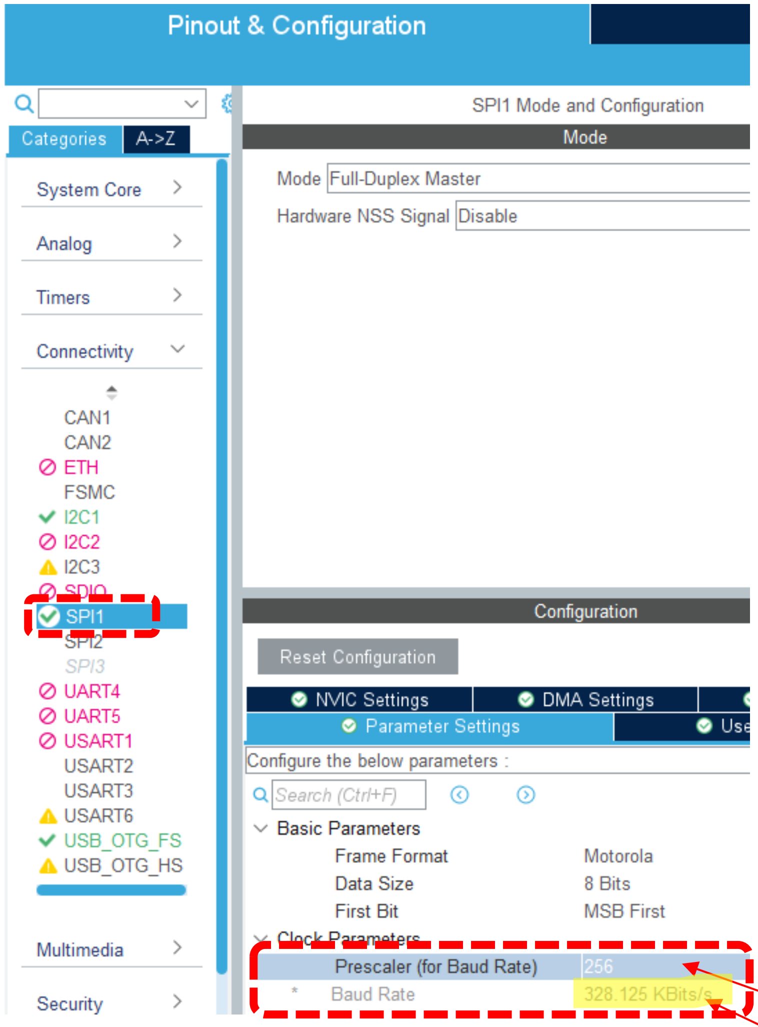
Figure 5. LIS3DSH electrical connection


Vdd
10µF
100nF
Vdd_IO
NC
NC
1
5
6
8
9
13
14
16
TOP VIEW
INT1/DRDY
INT 2
GND
SCL/SPC
SDA/SDI/SDO
SEL/SDO
CS
Digital signal from/to signal controller.Signal's levels are defined by proper selection of Vdd_IO
AM10211V1

https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples/blob/main/STM32F407_Discovery_kit/LIS3DSH.pdf

VP 6 - STM32F4 SPI1 in LIS3DSH

CubeMX nastavitve :



Pinout & Configuration

Categories | A->Z

System Core, Analog, Timers, Connectivity

CAN1, CAN2, ETH, FSMC, I2C1, I2C2, I2C3, SDIO, SPI1, SPI2, SPI3, UART4, UART5, USART1, USART2, USART3, USART6, USB_OTG_FS, USB_OTG_HS

Multimedia, Security

SPI1 Mode and Configuration

Mode

Mode: Full-Duplex Master

Hardware NSS Signal: Disable

Configuration

Reset Configuration

NVIC Settings | DMA Settings | Parameter Settings | User Constants

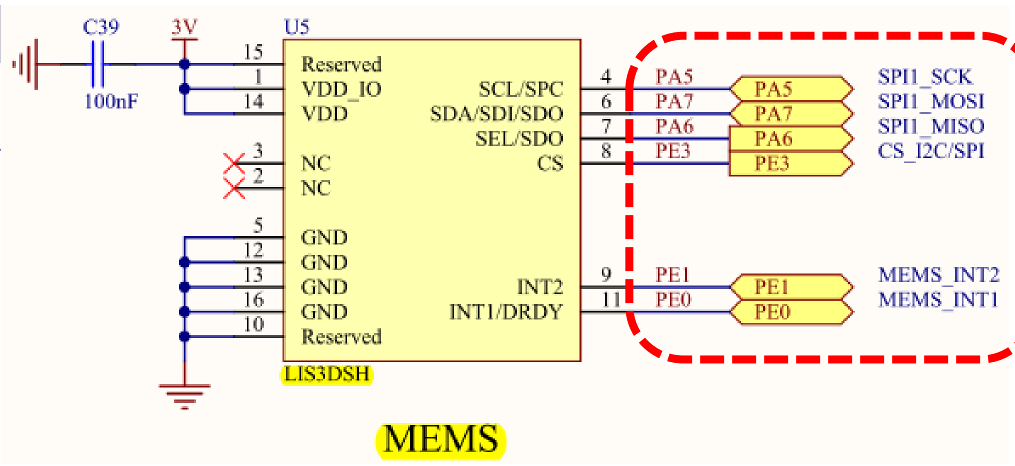
Configure the below parameters :

Basic Parameters

Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First

Clock Parameters

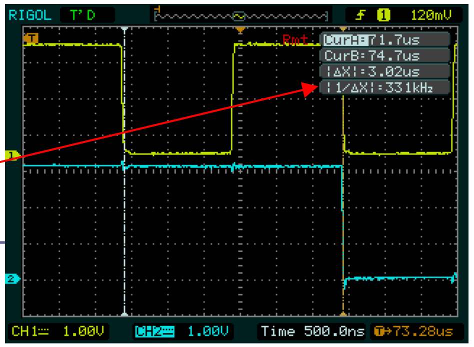
Prescaler (for Baud Rate)	256
Baud Rate	328.125 KBits/s



spi.c:

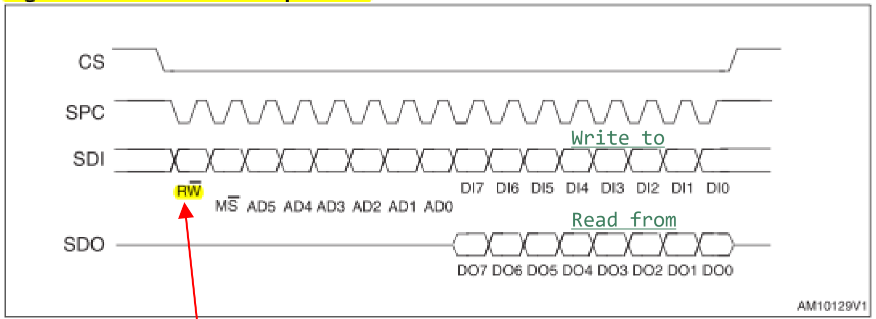
```
/* USER CODE END SPI1_Init 1 */
  hspi1.Instance = SPI1;
  hspi1.Init.Mode = SPI_MODE_MASTER;
  hspi1.Init.Direction = SPI_DIRECTION_2LINES;
  hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
  hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
  hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
  hspi1.Init.NSS = SPI_NSS_SOFT;
  hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_256;
  hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
  hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
  hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
  hspi1.Init.CRCPolynomial = 10;
  if (HAL_SPI_Init(&hspi1) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN SPI1_Init 2 */
```

Spremenimo iz 2 v 256 (počasnejša komunikacija)



VP 6 - STM32 SPI1 in LIS3DSH - program

Figure 6. Read and write protocol


CS
SPC
SDI
Write to
RW
MS AD5 AD4 AD3 AD2 AD1 AD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0
Read from
SDO
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0
AM10129V1

***bit 0*: RW bit**. When 0, the data DI(7:0) is written into the device. When 1, the data DO(7:0) from the device is read. In the latter case, the chip drives **SDO** at the start of bit 8.

***bit 1-7*: address AD(6:0)**. This is the address field of the indexed register.

***bit 8-15*: data DI(7:0) (write mode)**. This is the data that is written into the device (MSb first).

***bit 8-15*: data DO(7:0) (read mode)**. This is the data that is read from the device (MSb first).

8.3 WHO_AM_I (0Fh)

Who_AM_I register.

rozman 26. 04. 2022, 0..
0x3F

Table 19. WHO_AM_I register default value

0	0	1	1	1	1	1	1

```
// Config accelerometer
// Read WHOAMI register
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
outdata[0] = 0x0f | 0x80 ;  // read whoami
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
lis_id = indata[1];
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);

// Write to CTRL register (enable 3 axes meaurements on 25Hz)
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
outdata[0] = 0x20 ;  // switch on axes
outdata[1] = 0x47 ;
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);
```

SPI - serial peripheral interface

Subject to general operating conditions for Vdd and Top.

SPI slave timing values

l	Parameter	Value (1) Min.	Value (1) Max.	Unit
)	SPI clock cycle	100		ns
)	SPI clock frequency		10	MHz
)	CS setup time	6		

Table 7. Absolute maximum ratings

Symbol	Ratings	Maximum value	Unit
Vdd	Supply voltage	-0.3 to 4.8	V

Gradiva

8.5 CTRL_REG4 (20h)

Control register 4.

rozman 26. 04. 2022, 0..
0x47 (25Hz, all axes on)

Table 22. Control register 4

ODR3	ODR2	ODR1	ODR0	BDU	ZEN	YEN	XEN

Table 23. CTRL_REG4 register description

ODR 3:0	Output data rate and power mode selection. Default value:0000 (see *Table 24*)
BDU	Block data update. Default value:0 0:continuos update 1:output registers not updated until MSB and LSB reading)
Zen	Z axis enable. Default value:1 (0:Z axis disabled; 1:Z axis enabled)
Yen	Y axis enable. Default value:1 (0:Y axis disabled; 1:Y axis enabled)
Xen	X axis enable. Default value:1 0=X axis disabled; 1=X axis enabled

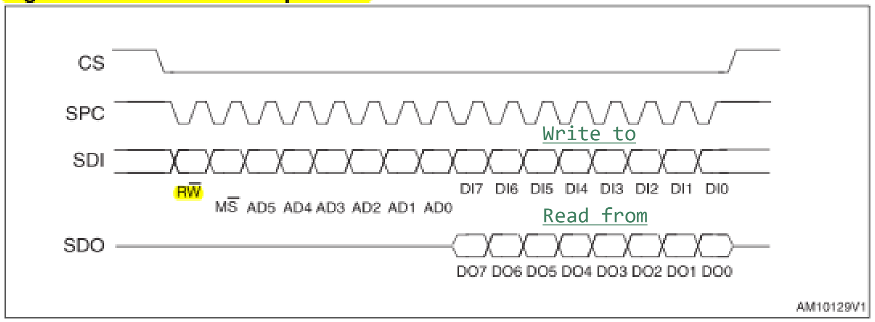
Table 24. CTRL4 ODR configuration

ODR3	ODR2	ODR1	ODR0	ODR selection
0	0	0	0	Power down
0	0	0	1	3.125 Hz
0	0	1	0	6.25 Hz
0	0	1	1	12.5 Hz
0	1	0	0	25 Hz

https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples/blob/main/Documentation/STM32F407_Discovery_kit/LIS3DSH.pdf

VP 6 - STM32 CubeIDE, SPI in LIS3DSH Gradiva

Figure 6. Read and write protocol


CS
SPC
SDI
Write to
RW
MS AD5 AD4 AD3 AD2 AD1 AD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0
Read from
SDO
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0
AM10129V1

bit 0: RW bit. When 0, the data DI(7:0) is written into the device. When 1, the data DO(7:0) from the device is read. In the latter case, the chip drives **SDO** at the start of bit 8.

bit 1-7: address AD(6:0). This is the address field of the indexed register.

bit 8-15: data DI(7:0) (write mode). This is the data that is written into the device (MSb first).

bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

7 Register mapping

Table 16 provides a list of the 8/16-bit registers embedded in the device and the related address:

Table 16. Register address map

Name	Type	Register address		Default	Comment
		Hex	Binary		
INFO1	r	0D	00001101	0010 0001	Information register 1
INFO2	r	0E	00001110	0000 0000	Information register 2
WHO_AM_I	r	OF	00001111	0011 1111	Who I am ID
OUT_X_L	r	28	00101000	0000 0000	Output registers
OUT_X_H	r	29	00101001		
OUT_Y_L	r	2A	00101010		
OUT_Y_H	r	2B	00101011		
OUT_Z_L	r	2C	00101100		
OUT_Z_H	r	2D	00101101		

```
// Read x,y,z axes
 outdata[0] = 0x29 | 0x80  ;  // read x
 HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
 HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
 AccelX = indata[1];

 outdata[0] = 0x2B | 0x80  ;  // read y
 HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
 AccelY = indata[1];

 outdata[0] = 0x2D | 0x80  ;  // read z
 HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
 HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);
 AccelZ = indata[1];
```

8.23 OUT_X (28h - 29h)

X-axis output register.

Table 49. OUT_X_L register default value

0	0	0	0	0	0	0	0

Table 50. OUT_X_H register default value

0	0	0	0	0	0	0	0

https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples/blob/main/Documentation/STM32F407_Discovery_kit/LIS3DSH.pdf

VP 6 - STM32 CubeIDE, SPI in LIS3DSH

main.c : dodana koda

Spremenljivke

```
/* USER CODE BEGIN PV */
#define BUFSIZE 256
charSendBuffer[BUFSIZE];
intCounter;
int KeyState=0;

// Global variables
uint8_t indata[2];
uint8_t outdata[2] = {0,0};
uint8_t lis_id;
int8_t AccelX;
int8_t AccelY;
int8_t AccelZ;

HAL_StatusTypeDef SPIStatus;

/* USER CODE END PV */
```

Glavna zanka

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

// Read x,y,z axes
  outdata[0] = 0x29 | 0x80  ;  // read x
  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
  HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
  AccelX = indata[1];

  outdata[0] = 0x2B | 0x80  ;  // read y
  HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
  AccelY = indata[1];

  outdata[0] = 0x2D | 0x80  ;  // read z
  HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);
  AccelZ = indata[1];

…

snprintf(SendBuffer,BUFSIZE,"Hello World [%d]: Key:%d Duty:%d PWM-Freq:%d PWM-Period:%d
Accel[ID:%02x] X:%04d Y:%d
Z:%04d\r\n",Counter++,KeyState,Duty,NoteFreq,NotePeriod,lis_id,AccelX,AccelY,AccelZ);
  CDC_Transmit_FS(SendBuffer,strlen(SendBuffer));

    /* USER CODE END WHILE */
```

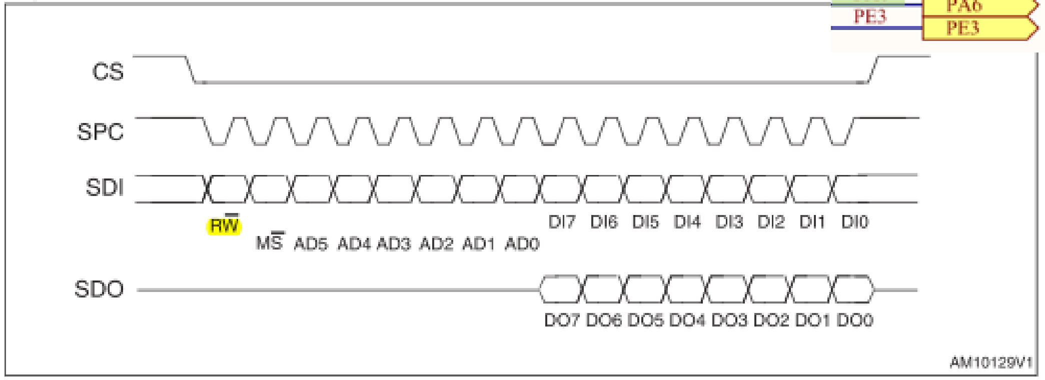
Inicializacija

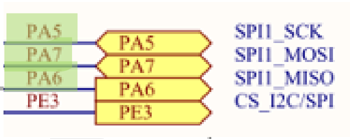
```
  /* USER CODE BEGIN 2 */

  // Config accelerometer
  // Read WHOAMI register
  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
  outdata[0] = 0x0f | 0x80 ;  // read whoami
  HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2,
HAL_MAX_DELAY);
  lis_id = indata[1];
  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);

  HAL_Delay(500);

  // Set CTRL register 0x47 -> [0x20]
  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
  outdata[0] = 0x20 ;  // switch on axes
  outdata[1] = 0x47 ;
  HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2,
HAL_MAX_DELAY);
  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);

  HAL_Delay(500);
  outdata[1] = 0x00 ;

  /* USER CODE END 2 */
```

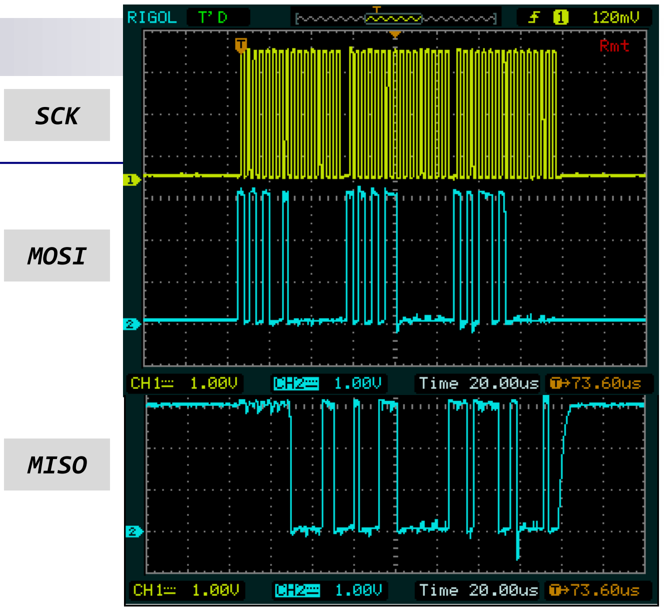
https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/STM32_SPI_LIS302DL_Basic

VP 6 - STM32 CubeIDE, SPI in LIS3DSH - Osciloskop

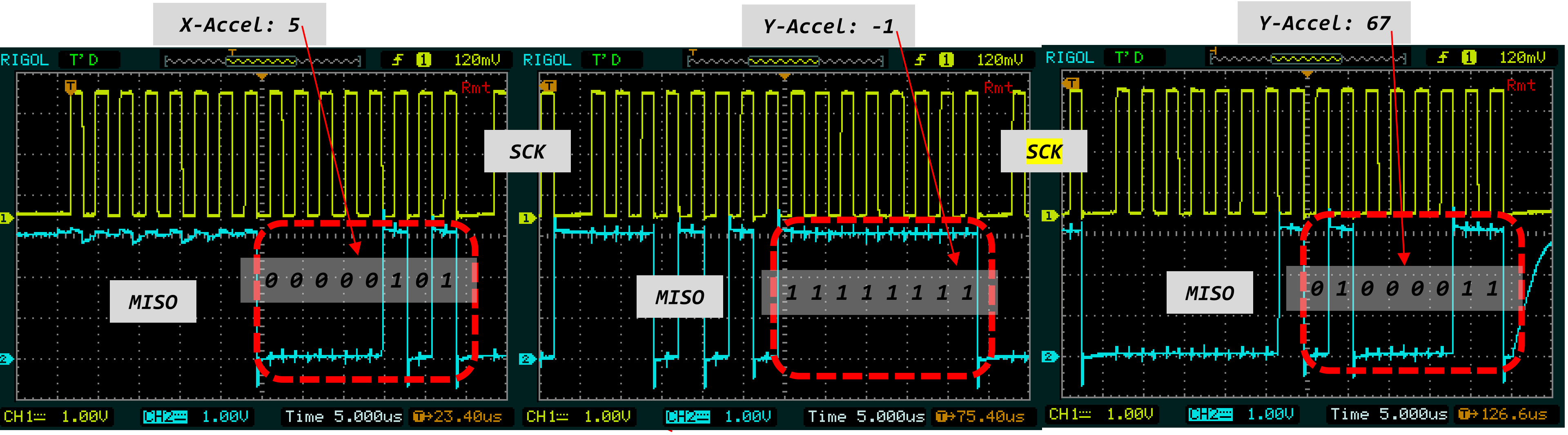
Figure 6. Read and write protocol


CS
SPC
SDI
RW
MS AD5 AD4 AD3 AD2 AD1 AD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0
SDO
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0
AM10129V1





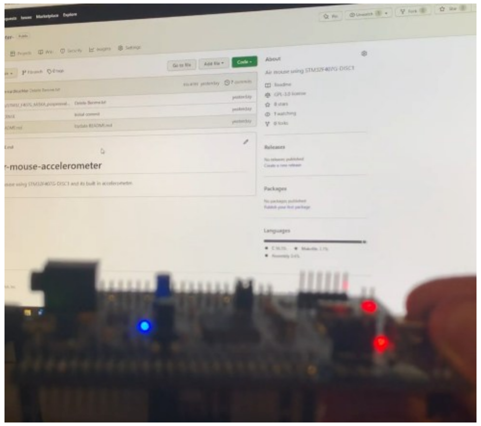
```
Hello World [3530]: Key:0000 Accel[ID:00] X:0005 Y:-1 Z:0066
Hello World [3531]: Key:0000 Accel[ID:00] X:0005 Y:-1 Z:0067
```

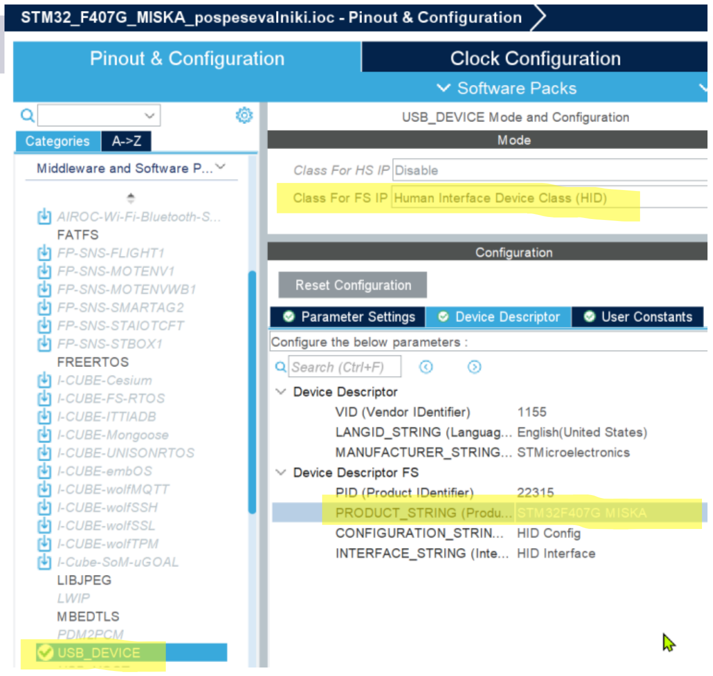

X-Accel: 5
Y-Accel: -1
Y-Accel: 67
SCK
SCK
MISO
MISO
MISO
0 0 0 0 0 1 0 1
1 1 1 1 1 1 1 1
0 1 0 0 0 0 1 1
RIGOL T'D 120mV
CH1 1.00V CH2 1.00V Time 5.000us 23.40us
CH1 1.00V CH2 1.00V Time 5.000us 75.40us
CH1 1.00V CH2 1.00V Time 5.000us 126.6us

AirMouse STM32F4

Beremo pospeškomer in sporočamo premike kazalca na zaslonu preko ustreznega USB HID profila
Avtor: Bernard Kuchler

```
while (1)
{

// Read accel values into AccelX,Y,Z

if (AccelX < min_xval){
newxval = AccelX - min_xval;
} else if (AccelX > max_xval) {
newxval = AccelX - max_xval;
}

if (AccelY < min_yval){
newyval = AccelY - min_yval;
} else if (AccelY > max_yval){
newyval = AccelY - max_yval;
}

if ((newxval > 10) || (newxval <-10)) //Determines the necessary
amount of change in value from the sensor to start moving the mouse
cursor
{
mousehid.mouse_y = (newxval/10); //Divides the value from the
sensor by 10 in order to make a slower acceleration of the mouse
cursor and thereby making it more accurate to use
}
else mousehid.mouse_y = 0;


if ((newyval > 10) || (newyval <-10)) {
mousehid.mouse_x= (newyval)/10;
} else mousehid.mouse_x = 0;
…
USBD_HID_SendReport(&hUsbDeviceFS,&mousehid, sizeof (mousehid));
//Send data to USB
```




STM32_F407G_MISKA_pospesevalniki.ioc - Pinout & Configuration

USB_DEVICE Mode and Configuration

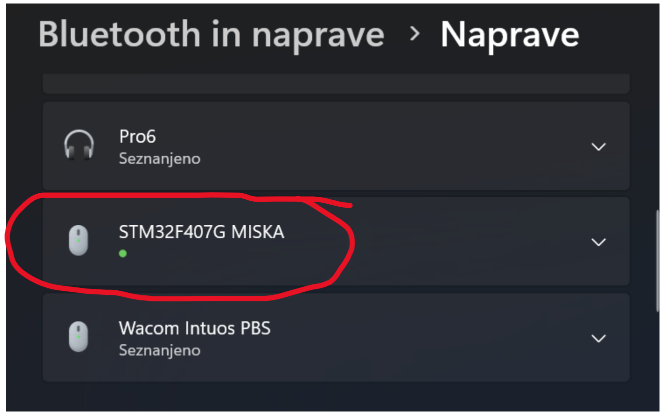
Mode

Class For HS IP: Disable
Class For FS IP: Human Interface Device Class (HID)

Configuration

Configure the below parameters :

Device Descriptor
VID (Vendor IDentifier) 1155
LANGID_STRING (Languag... English(United States)
MANUFACTURER_STRING... STMicroelectronics
Device Descriptor FS
PID (Product IDentifier) 22315
PRODUCT_STRING (Produ... STM32F407G MISKA
CONFIGURATION_STRIN... HID Config
INTERFACE_STRING (Inte... HID Interface


Bluetooth in naprave › Naprave

Pro6
Seznanjeno

STM32F407G MISKA

Wacom Intuos PBS
Seznanjeno

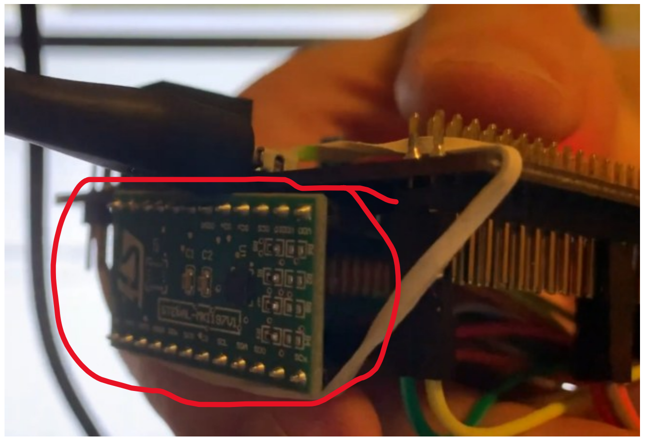
https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples/tree/main/STM32_F407G_MISKA_pospesevalniki

AirMouse STM32F4 + LSM6DSOX

Beremo zunanji pospeškomer in sporočamo premike kazalca na zaslonu preko ustreznega USB HID profila
Avtor: Bernard Kuchler

The LSM6DSOX is a 6-axis IMU (inertial measurement unit) system-in-package featuring a 3-axis digital accelerometer and a 3-axis digital gyroscope, boosting performance at 0.55 mA in high-performance mode and enabling always-on low-power features for an optimal motion experience for the consumer.

https://www.st.com/en/mems-and-sensors/lsm6dsox.html



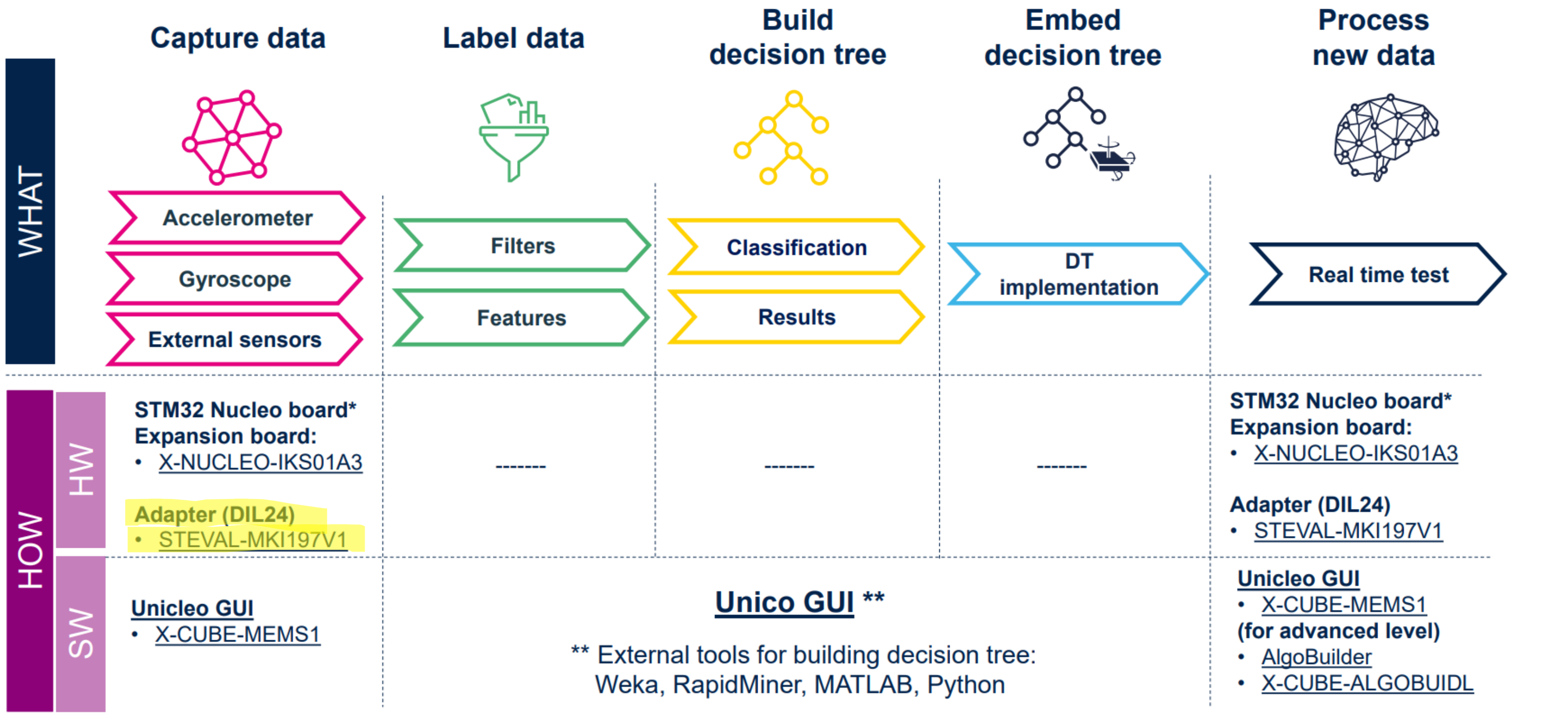
```
//Read gyroscope measurements
outdata[0] = 0x23 | 0x80 ; // read x (pitch), 0x4B
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
GyroX = indata[1];


outdata[0] = 0x25 | 0x80 ; // read y (roll), 0x4D
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
GyroY = indata[1];


outdata[0] = 0x27 | 0x80 ; // read z (yaw), 0x4F
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
GyroZ = indata[1];
```

```
//Read accelerometer measurements
outdata[0] = 0x29 | 0x80 ; // read x, 0x51
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
AccelX = indata[1];
outdata[0] = 0x2B | 0x80 ; // read y, 0x53
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
AccelY = indata[1];
outdata[0] = 0x2D | 0x80 ; // read z, 0x55
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
AccelZ = indata[1];
```

https://github.com/LAPSyLAB/STM32F4_Docs_and_Examples/tree/main/STM32_F407G_MISKA_Gyro

LSM6DSOX

LSM6DSOX - STM32 Nucleo with expansion board



		Capture data	Label data	Build decision tree	Embed decision tree	Process new data
WHAT		Accelerometer Gyroscope External sensors	Filters Features	Classification Results	DT implementation	Real time test
HOW	HW	**STM32 Nucleo board*** **Expansion board:** • X-NUCLEO-IKS01A3 **Adapter (DIL24)** • STEVAL-MKI197V1	-------	-------	-------	**STM32 Nucleo board*** **Expansion board:** • X-NUCLEO-IKS01A3 **Adapter (DIL24)** • STEVAL-MKI197V1
	SW	**Unicleo GUI** • X-CUBE-MEMS1	**Unico GUI** ** ** External tools for building decision tree: Weka, RapidMiner, MATLAB, Python			**Unicleo GUI** • X-CUBE-MEMS1 **(for advanced level)** • AlgoBuilder • X-CUBE-ALGOBUIDL

https://www.st.com/en/mems-and-sensors/lsm6dsox.html

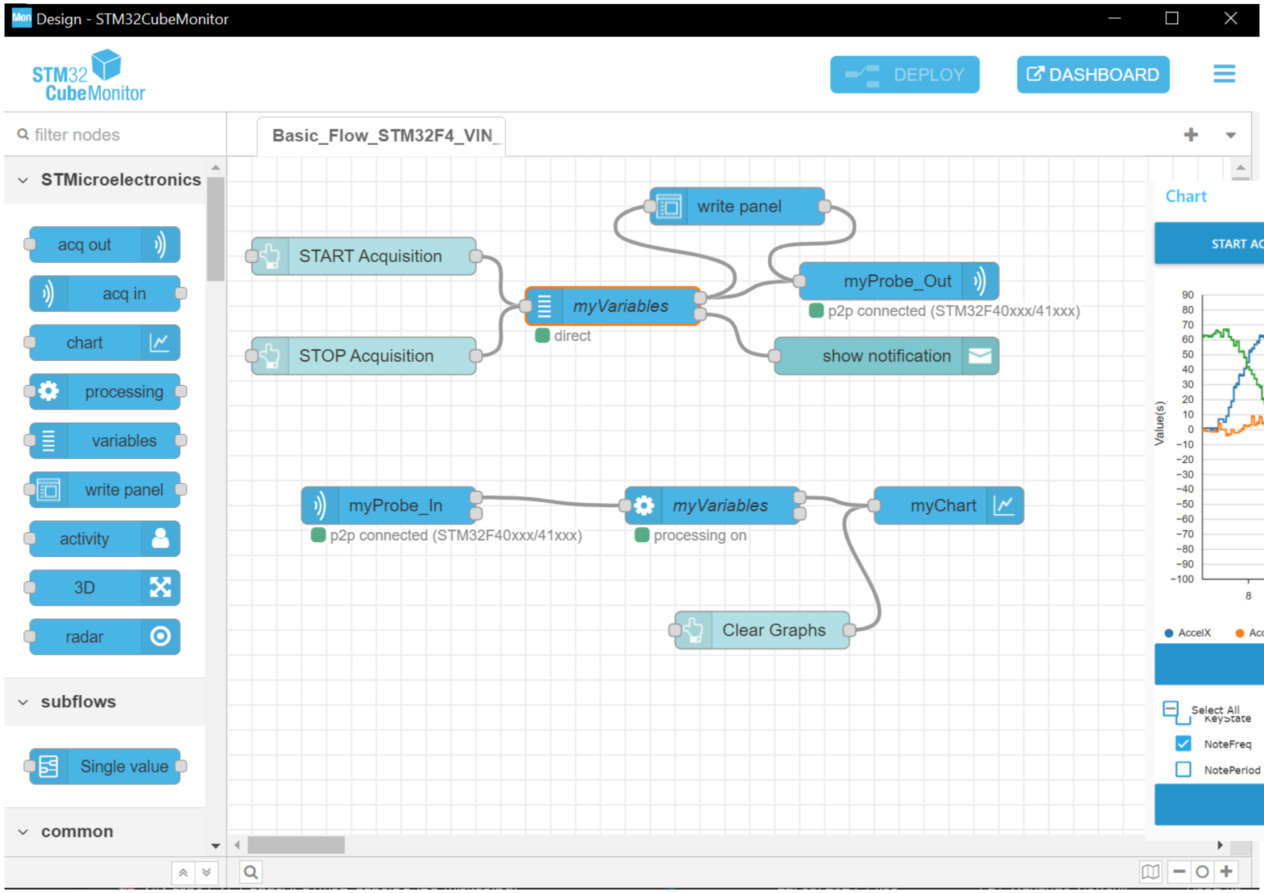
VP 6: STM32H7 I2C primeri, STM32F4 SPI, pospeškomer, „Air USB mouse“

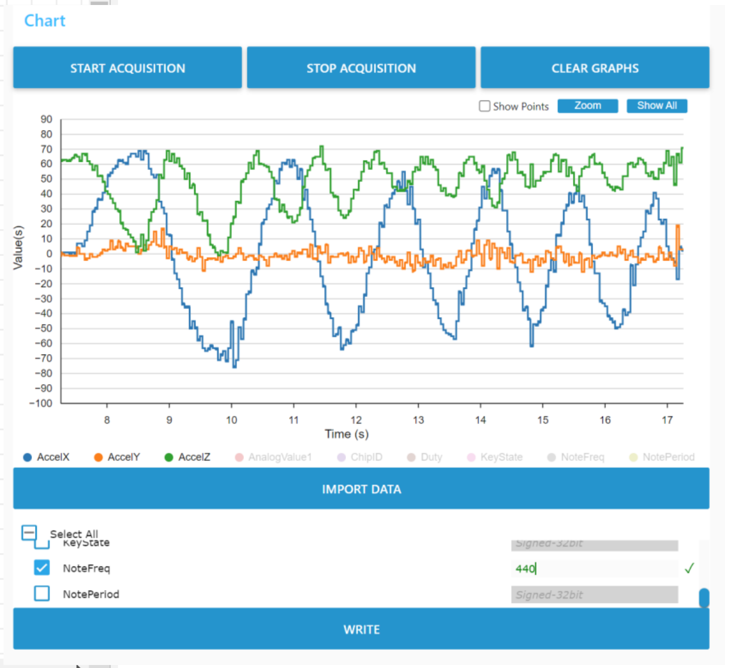
- STM32H7 CubeIDE, I2C (Scan, WM9884, Touch)
- Predstavitev STM32F4
- CubeIDE projekt STM32F4, SPI, pospeškomer:
  - SPI - LIS3DSH pospeškomer
  - USB „Air Mouse“ with STM32F4, SPI1 and LIS3DSH
- Sledenje („tracing“) – CubeMonitor
- Osciloskop - ponovitev

STM32CubeMonitor

STM32CubeMonitor is a tool that allows real-time sampling and visualization of user variables while the application is running. It runs on Windows, Linux or macOS, and provides a browser-based interface.

The user can define their own flow to monitor variables for their STM32 microcontroller-based application. Example design and dashboard views are shown below.





https://wiki.stmicroelectronics.cn/stm32mcu/wiki/Category:STM32CubeMonitor

VIN projekt - VP 6 STM32-CubeIDE projekt

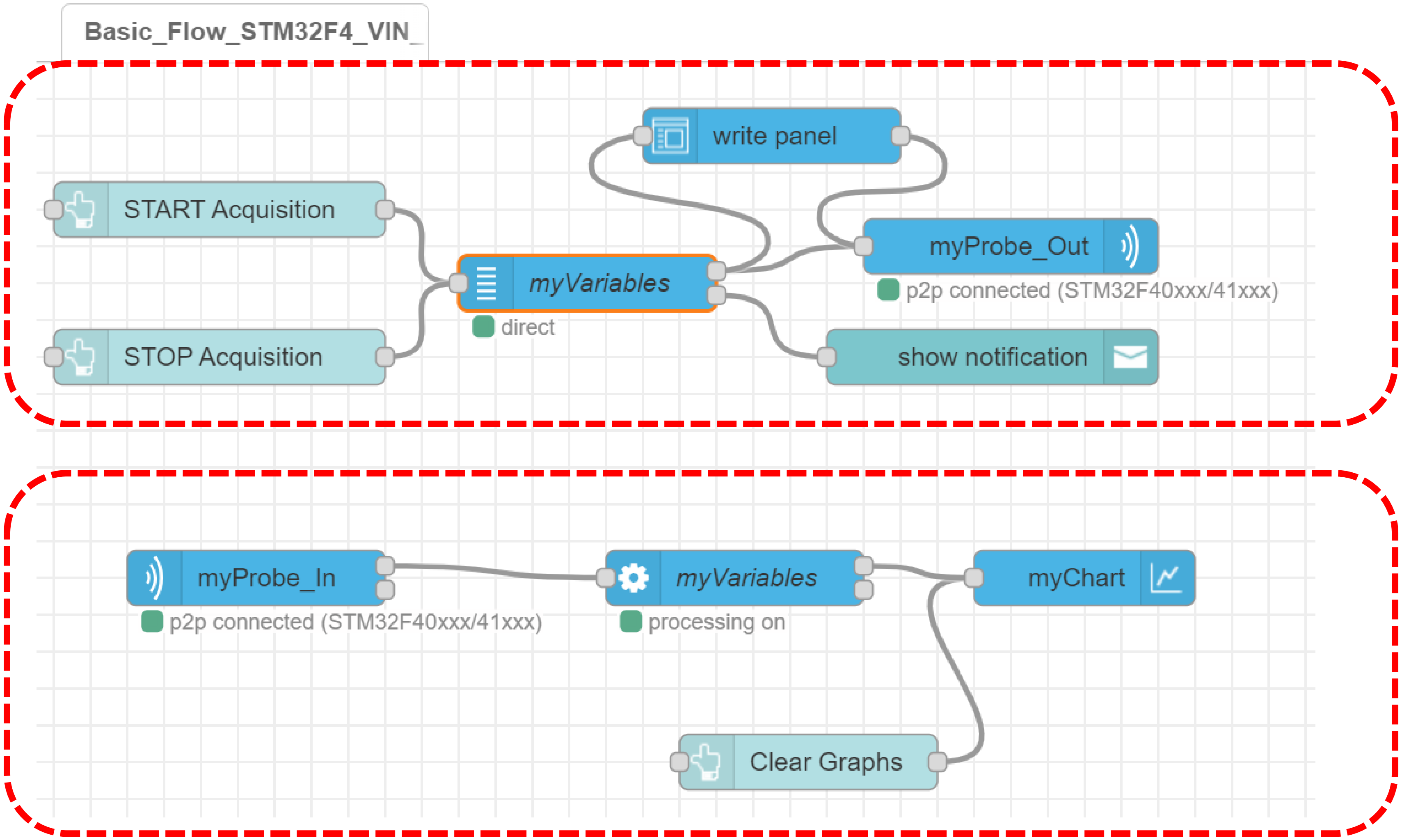
Program : za demonstracijo različnih funkcionalnosti – ADC, PWM – LED, Buzzer, SPI - Accel, I2C - audio

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
        while (1)
        {
        htim4.Instance->CCR1 = Duty;
        htim4.Instance->CCR2 = 100-Duty;
        htim4.Instance->CCR3 = Duty;
        htim4.Instance->CCR4 = 100-Duty;

        KeyState = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);

        // Read x,y,z axes
        outdata[0] = 0x29 | 0x80 ; // read x
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
        HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
        AccelX = indata[1];

        outdata[0] = 0x2B | 0x80 ; // read y
        HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
        AccelY = indata[1];

        outdata[0] = 0x2D | 0x80 ; // read z
        HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);
        AccelZ = indata[1];

        HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
        AnalogValue1 = HAL_ADC_GetValue(&hadc1);
        HAL_ADC_Start(&hadc1);
/* USER CODE END WHILE */
```

```
/* USER CODE BEGIN 3 */
if ( (HAL_GetTick() - TickLast) > 1000) { // Do this each second !
Duty = (Duty + 10) ; // Add 10 if delay 1 sec, add 1 on shorter delay...
if (Duty > 100 )
Duty = 1;


// From Device with address=0x94, Read register with address 0x01 and put value
in ChipID
// DevAddress_0x94, tMemAddress=0x01, MemAddSize=8b, *pData,Size, Timeout);
retval = HAL_I2C_Mem_Read(&hi2c1, 0x94, 0x01, I2C_MEMADD_SIZE_8BIT, &ChipID, 1,
1000);


// Change Period and set 50% duty for buzzer PWM output
NotePeriod = (int)(1000000/NoteFreq); //Already prescaled to 1 MHz
setPWM(htim2, TIM_CHANNEL_1, NotePeriod, NotePeriod/2);


// Print values on USB VComPort
snprintf(SendBuffer,BUFSIZE,"Hello World [%d]: Key:%d Duty:%d PWM-Freq:%d PWM-
Period:%d Accel[ID:%02x] X:%04d Y:%d Z:%04d ChipID:%02x
ADC1:%d\r\n",Counter++,KeyState,Duty,NoteFreq,NotePeriod,lis_id,AccelX,AccelY,A
ccelZ,ChipID,AnalogValue1);
CDC_Transmit_FS(SendBuffer,strlen(SendBuffer));


TickLast = HAL_GetTick(); // Reset counter
};


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

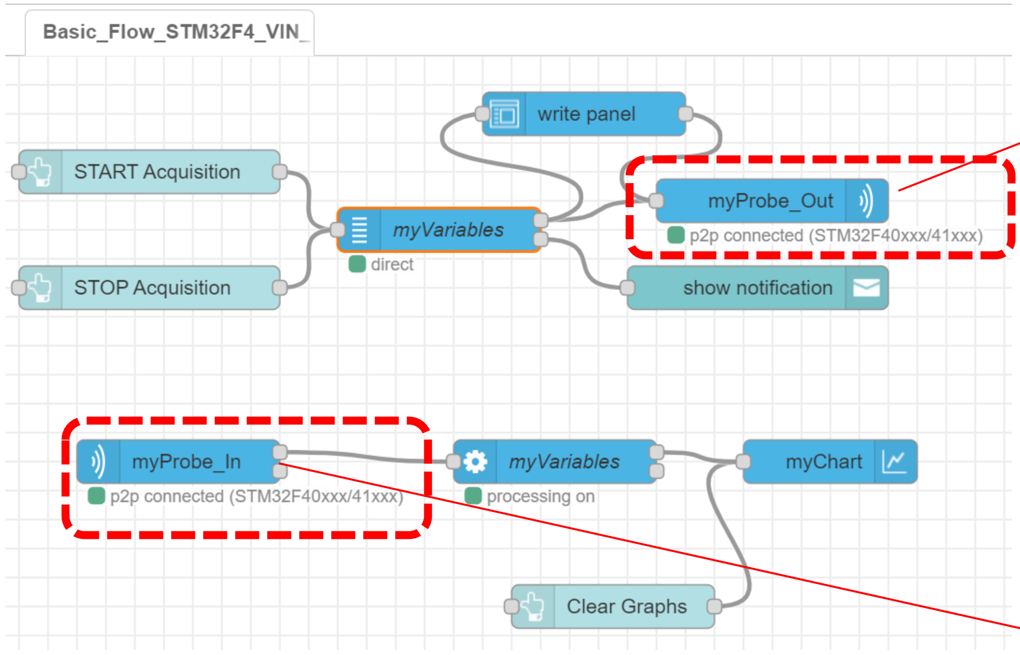
https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/STM32F4_GPIO_PWM_SPI_I2C_C_Demo

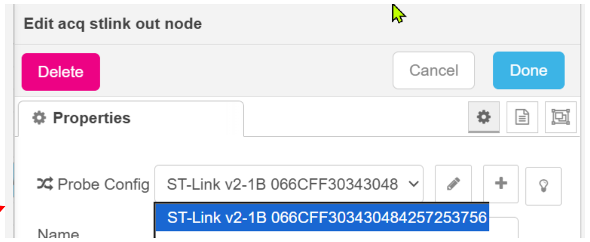
STM32CubeMonitor



https://wiki.stmicroelectronics.cn/stm32mcu/wiki/Category:STM32CubeMonitor

STM32CubeMonitor



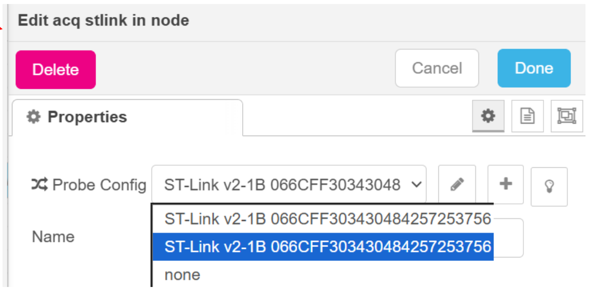


Edit acq stlink out node

Probe Config: ST-Link v2-1B 066CFF30343048

ST-Link v2-1B 066CFF303430484257253756

Name



Edit acq stlink in node

Probe Config: ST-Link v2-1B 066CFF30343048

ST-Link v2-1B 066CFF303430484257253756

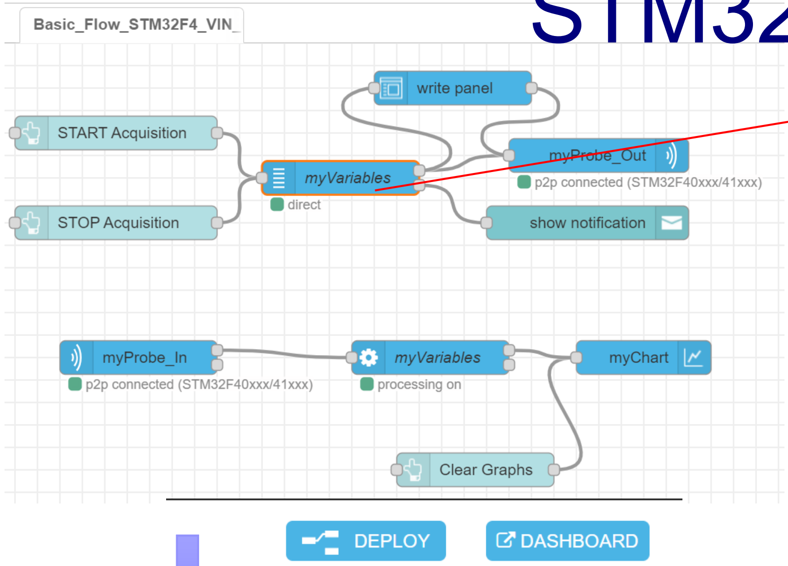
ST-Link v2-1B 066CFF303430484257253756

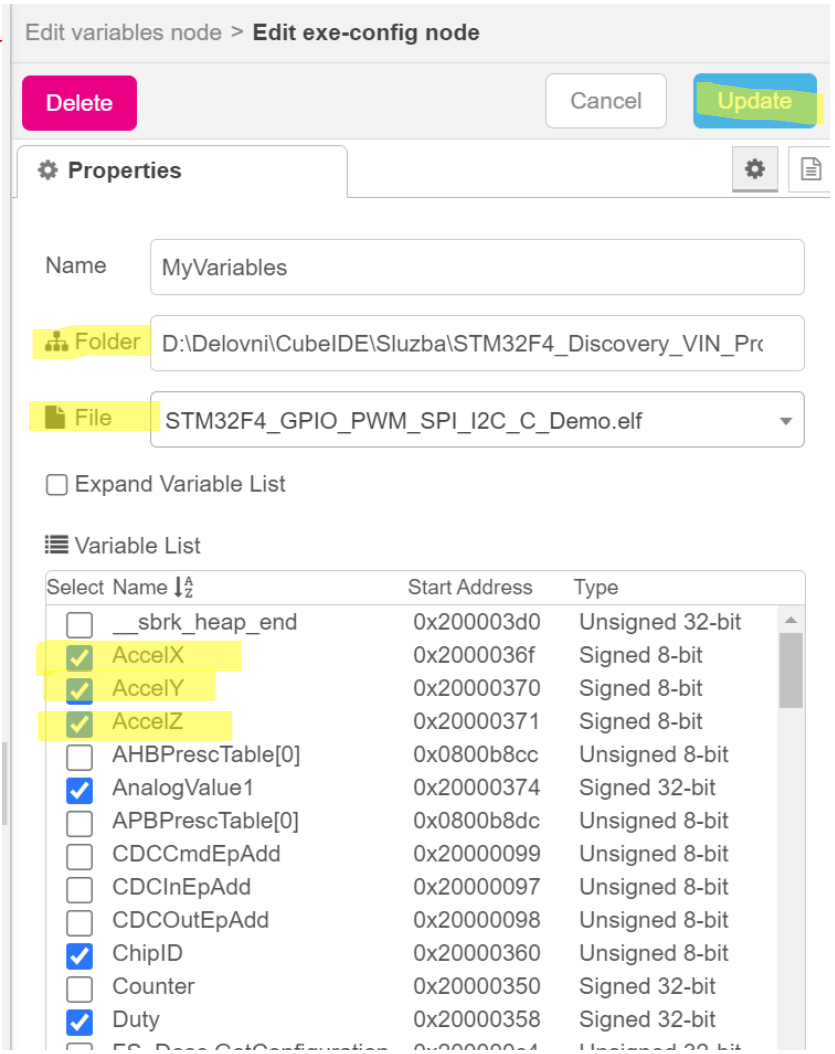
none

Name

https://wiki.stmicroelectronics.cn/stm32mcu/wiki/Category:STM32CubeMonitor

STM32CubeMonitor





Edit variables node > **Edit exe-config node**

Delete | Cancel | Update

Properties

Name: MyVariables

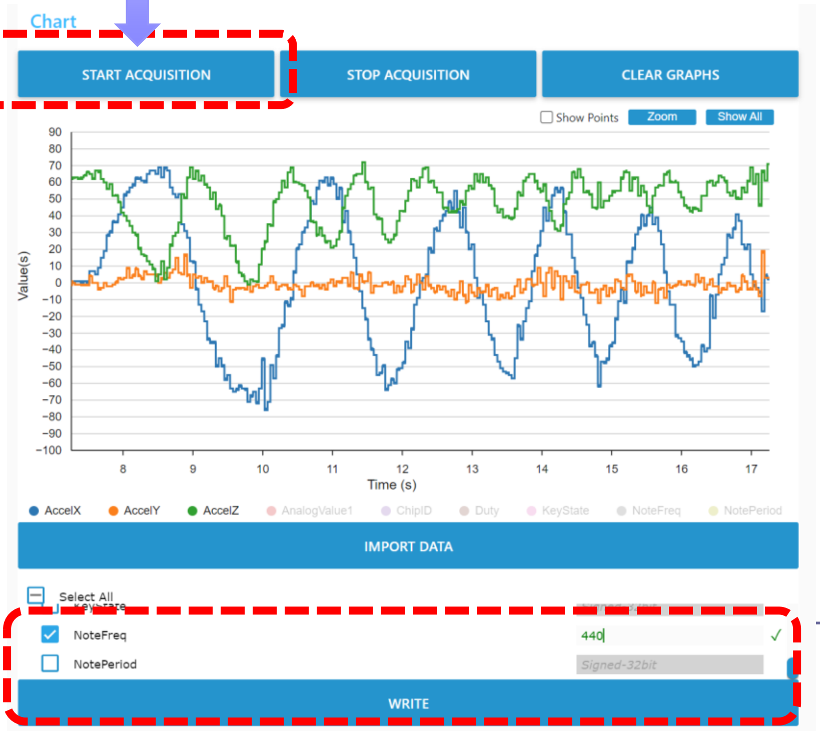
Folder: D:\Delovni\CubeIDE\Sluzba\STM32F4_Discovery_VIN_Pro

File: STM32F4_GPIO_PWM_SPI_I2C_C_Demo.elf

☐ Expand Variable List

Variable List

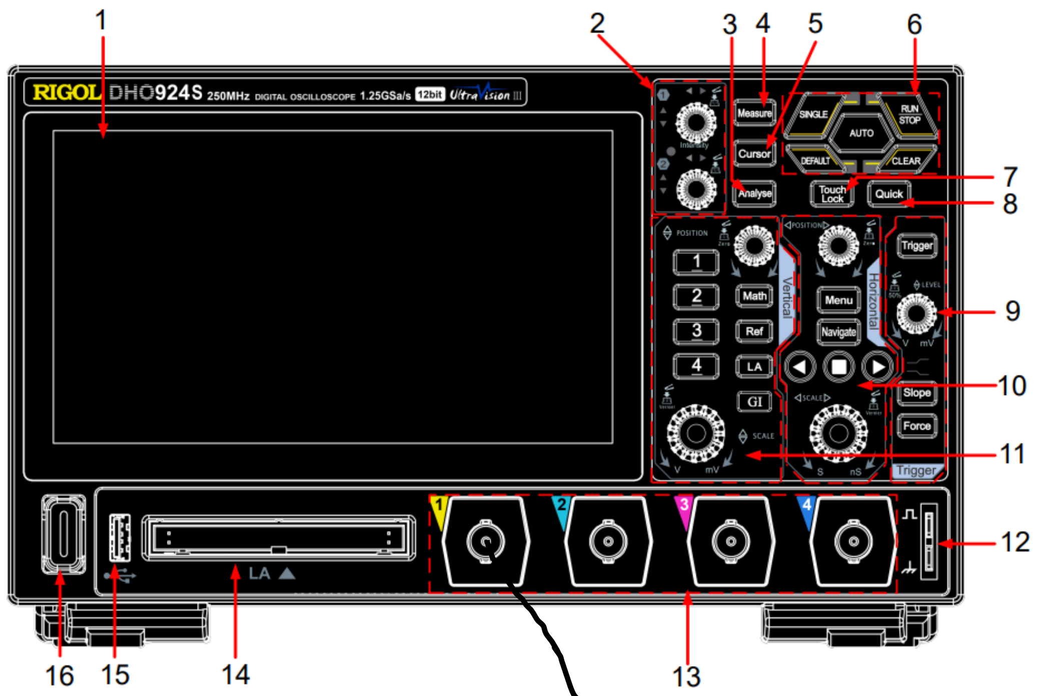
Select	Name	Start Address	Type
☐	__sbrk_heap_end	0x200003d0	Unsigned 32-bit
✓	AccelX	0x2000036f	Signed 8-bit
✓	AccelY	0x20000370	Signed 8-bit
✓	AccelZ	0x20000371	Signed 8-bit
☐	AHBPrescTable[0]	0x0800b8cc	Unsigned 8-bit
✓	AnalogValue1	0x20000374	Signed 32-bit
☐	APBPrescTable[0]	0x0800b8dc	Unsigned 8-bit
☐	CDCCmdEpAdd	0x20000099	Unsigned 8-bit
☐	CDCInEpAdd	0x20000097	Unsigned 8-bit
☐	CDCOutEpAdd	0x20000098	Unsigned 8-bit
✓	ChipID	0x20000360	Unsigned 8-bit
☐	Counter	0x20000350	Signed 32-bit
✓	Duty	0x20000358	Signed 32-bit



VP 6: STM32H7 I2C primeri, STM32F4 SPI, pospeškomer, „Air USB mouse“

- STM32H7 CubeIDE, I2C (Scan, WM9884, Touch)
- Predstavitev STM32F4
- CubeIDE projekt STM32F4, SPI, pospeškomer:
  - SPI - LIS3DSH pospeškomer
  - USB „Air Mouse“ with STM32F4, SPI1 and LIS3DSH
- Sledenje („tracing“) – CubeMonitor
- Osciloskop - ponovitev

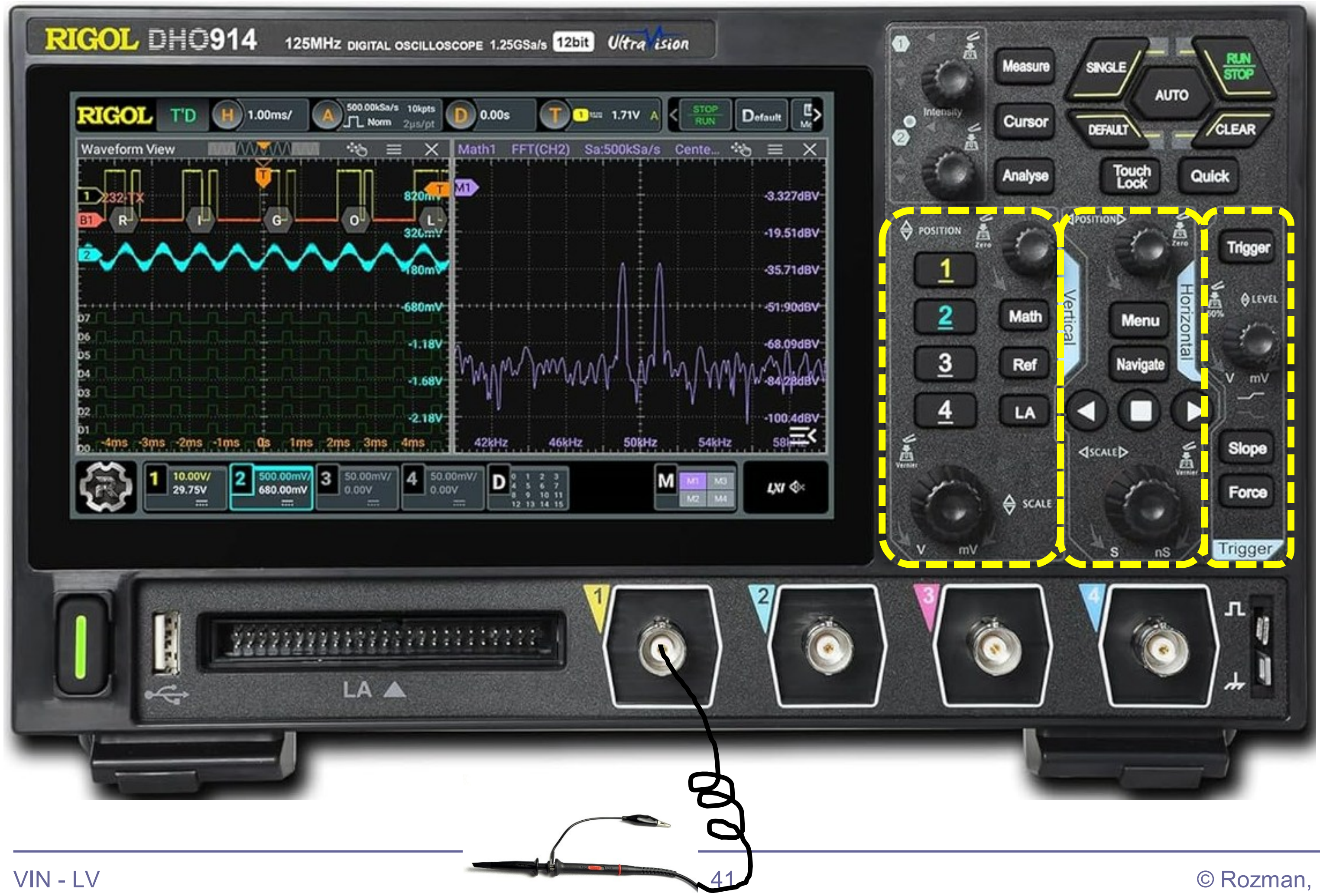
Prednja stran osciloskopa - shema



1	7'' Capacitive Touch Screen	9	Trigger Controls
2	Multipurpose Knobs	10	Horizontal Controls
3	Analyse Key	11	Vertical Controls
4	Measure Key	12	Probe Compensation Signal Output Terminal/Ground Terminal
5	Cursor Key	13	Analog Channel Input Terminals
6	Common Tools Keys	14	Digital Channel Input Terminal
7	Touch Lock Key	15	USB HOST Port
8	Quick Action Key (Self-defined function)	16	Power Key

https://download.rigol.com/en/Manual/Digital%20Oscilloscope/DHO900/DHO900_QuickGuide_EN.pdf

Prednja stran osciloskopa - realna



Prednja stran osciloskopa - kontrole

<u>Y-os (el. napetost)</u>

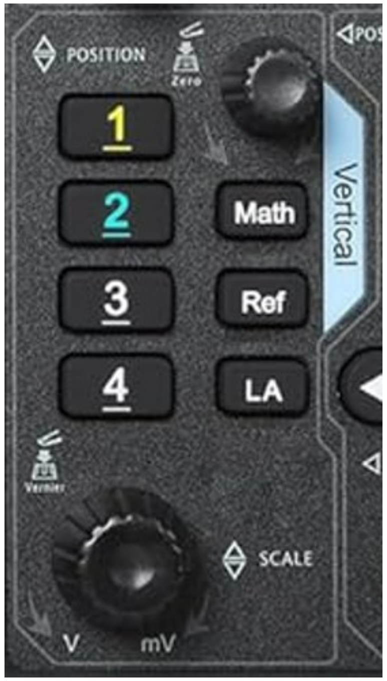
- nastavitev merila [V/razdelek]
- pozicioniranje y-os
- prikaz kanalov da/ne

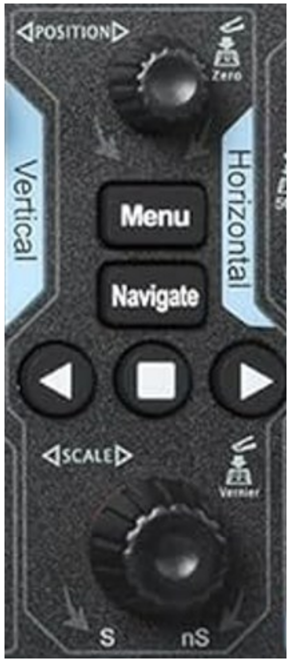
<u>X-os (čas)</u>

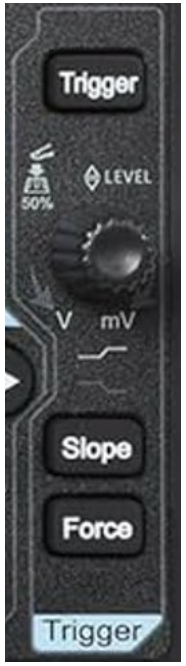
- nastavitev merila [s/razdelek]
- pozicioniranje

<u>Prožilnik</u>

- začetek prikaza
- tipično: poz. fronta in 50%



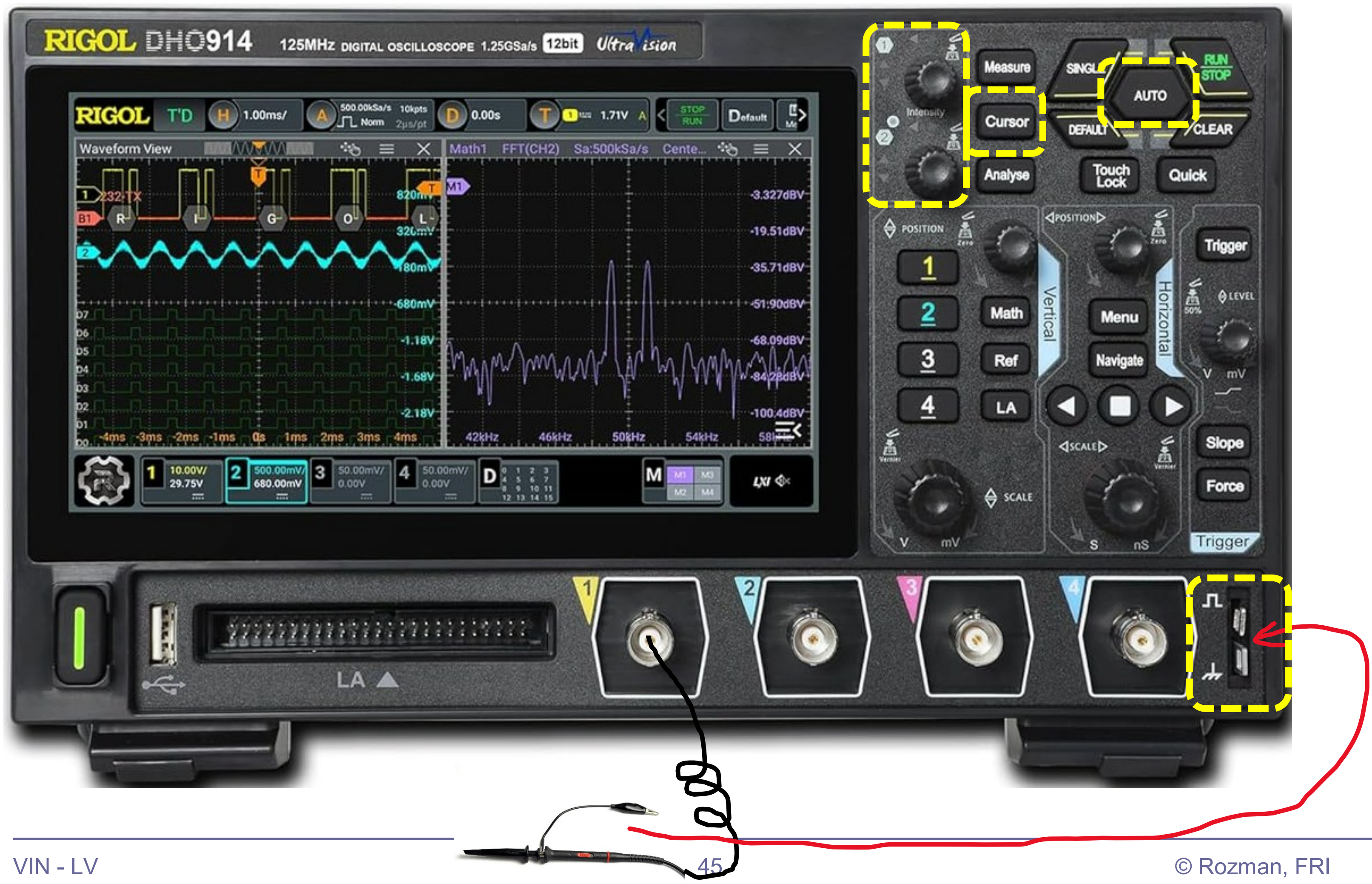




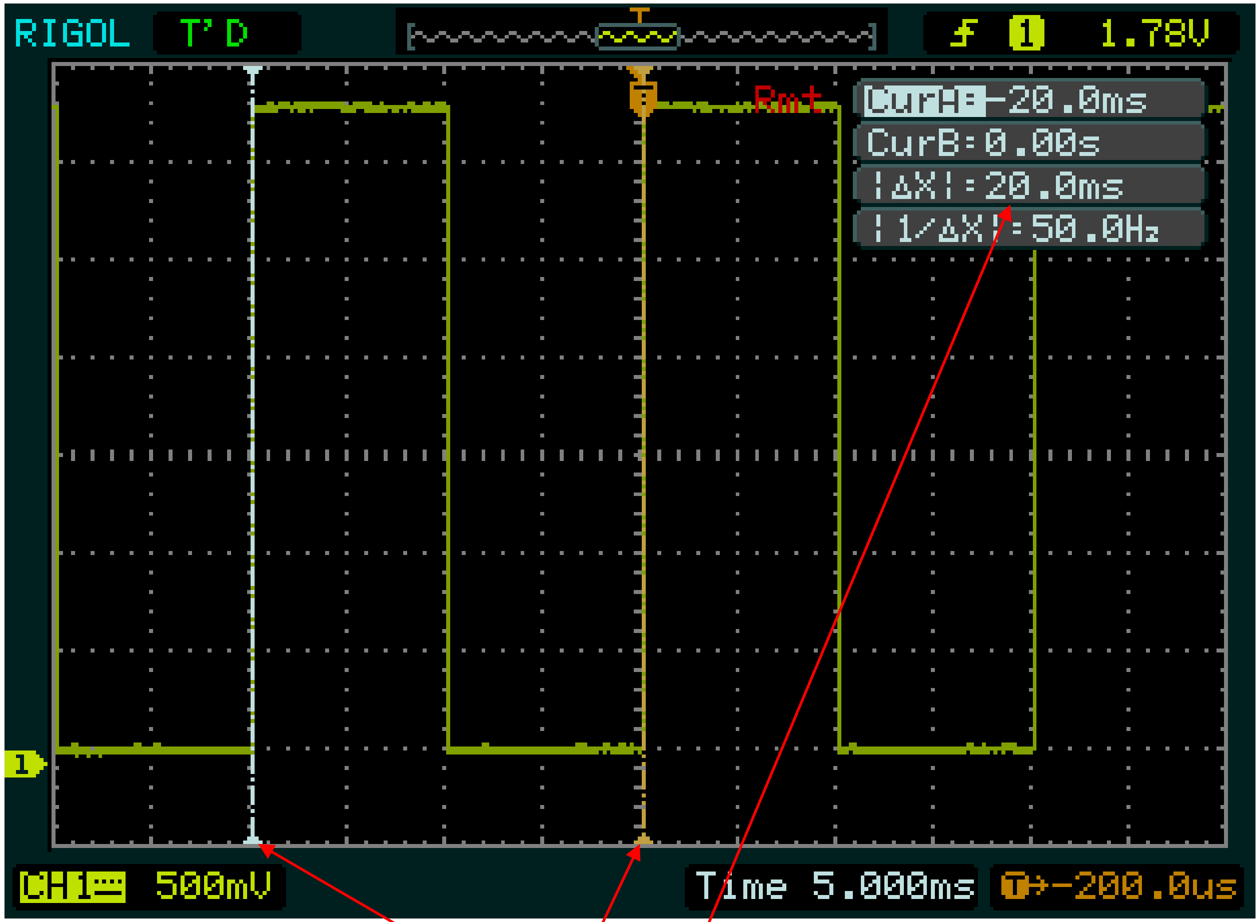
https://rigolshop.eu/dho914.html

Spoznavanje merilne opreme…

Meritev testnega signala



Testni signal – meritev periode, frekvence

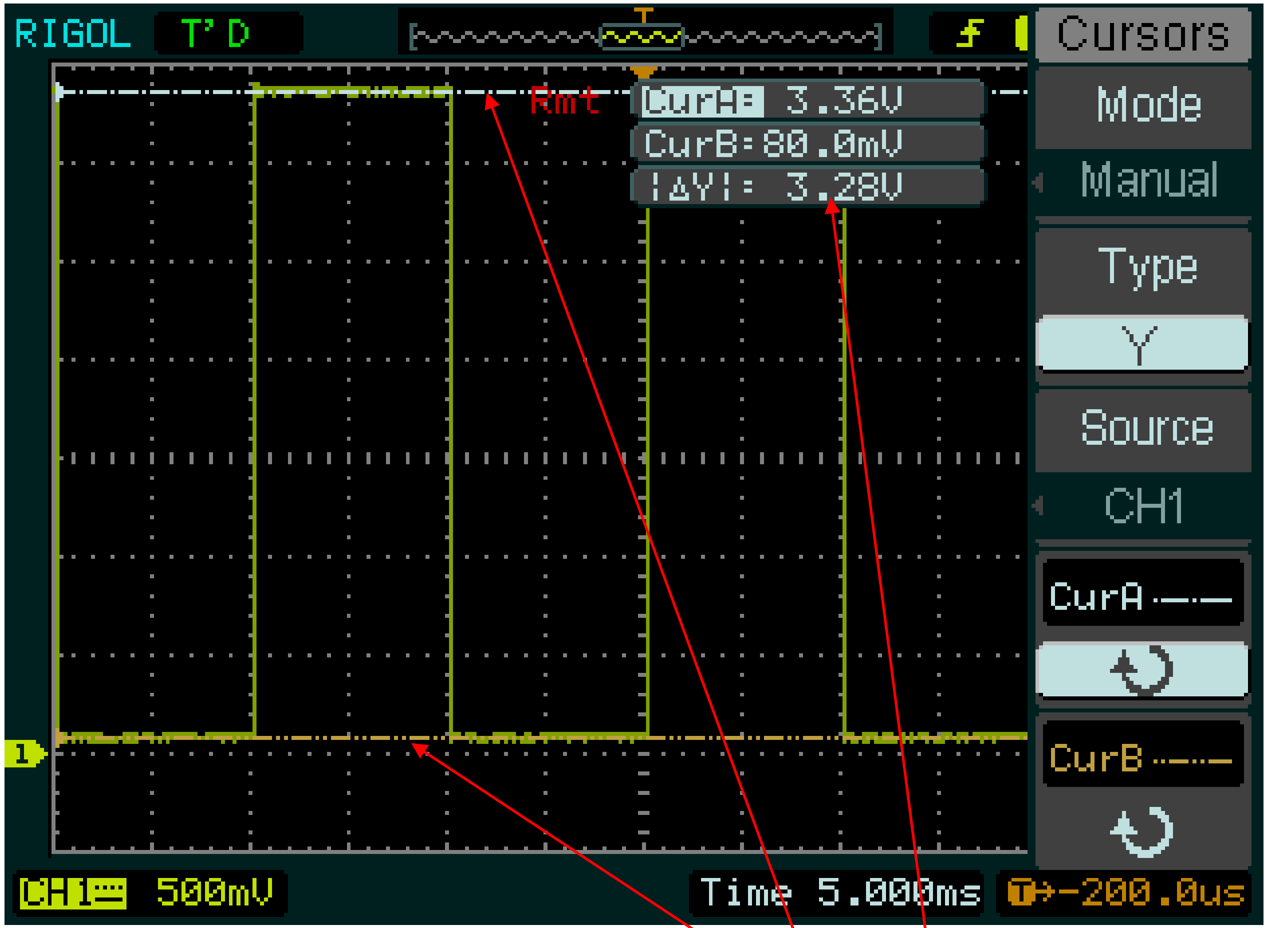


CurA: -20.0ms
CurB: 0.00s
|ΔX|: 20.0ms
|1/ΔX|: 50.0Hz

Meritev periode/frekvence signala:

- **? ms, ? Hz**

Testni signal – meritev amplitude



Meritev amplitude signala:

- **? V**

VP 6: STM32H7 I2C primeri, STM32F4 SPI, pospeškomer, „Air USB mouse“

- Diskusija, vprašanja ?