

T170B417 Įterptinės sistemos

Inžinerinis projektas: užduotis ir instrukcijos

Turinys

1	Užduotis	2
2	Užduoties variantai	3
3	Vertinimo skalė (rubrika).....	4
4	Indikatoriai	5
4.1	Mokomoji medžiaga	5
4.2	LCD indikatoriaus su HD44780 valdikliu prijungimas prie STM32 mikroprocesorių	5
4.3	OLED indikatoriaus su SSD1306 valdikliu prijungimas prie STM32 mikroprocesorių.....	6
5	Duomenų perdavimas į kompiuterį.....	6
5.1	USART sąsajos panaudojimas.....	6
5.2	UART ryšys panaudojant STM32L053-DISCO	7
5.3	COM prievado aptikimas Windows operacijų sistemoje	8
5.4	UART ryšys su kompiuteriu Nucleo tipo maketuose	9
5.4.1	Nucleo-F401RE (Nucleo-64 plokštės tipas)	9
5.4.2	Nucleo-F433CP (Nucleo-64-P MB1319 tipas).....	10
5.4.3	Nucleo-L073 (Nucleo-64 plokštės tipas).....	11
5.4.4	Paklauskime ChatGPT	12
5.4.5	ChatGPT atsakymas.....	12
5.5	Terminalinės programos naudojimas.....	14
5.6	UART komunikacijų programavimas mikrovaldiklyje	15
5.6.1	USART sąsajos konfigūravimas	15
5.6.2	USART programavimas HAL funkcijų pagalba.....	17
5.7	Pavyzdžiai.....	23
5.8	Įrenginio paklaidos skaičiavimo pavyzdžiai.....	23
6	Bendrieji reikalavimai techniniams sprendimams	24
7	Ataskaitoje pateikti	24

1 Užduotis

Suprojektuoti ir realizuoti įterptinę sistemą, sudarytą iš:

- STM32xxx serijos mikrovaldiklio
- Nurodytų jutiklių
- Pasirinkto indikatoriaus
- Duomenų perdavimo sąsajos į personalinį kompiuterį (PK)

Suprojektuota įterptinė sistema matuojamus duomenis turi perduoti į PK atvaizdavimui ir išsaugojimui.

2 Užduoties variantai

Lentelė 1. Užduoties variantų lentelė

Var. Nr.	Matuojamas fizikinis dydis arba jutiklis	Kanalų skaičius	Vaizduojami ir perduodami į kompiuterį parametrai	Signalų pralaidumo juosta	Dydzio diapazonas, signalo amplitudė	Parodymų atnaujinimo indikatoriuje periodas, s	Perduodamų į kompiuterį parametrų diskretizavimo periodas, s
1.	Garsas (Elektretinis/MEMS mikrofonas)	1	Vidutinė kvadratinė vertė	(0.1-5.0) kHz	parinkti	2	0,5
2.	Vibracijos (Akselerometras)	1	Vidutinė kvadratinė vertė	(0-100) Hz	parinkti	2	0,2
3.	Temperatūra	2	Imties vertė, kanalų skirtumai	(0-1) Hz	-20 .. +40 C	5	2
4.	Barometrinis slėgis	1	Vidurkis	(0-10) Hz	(90-110)kPa	5	1
5.	Apšviestumas	2	Vidurkis, kanalų skirtumai	(0-100) Hz	parinkti	2	0,2
6.	Atstumas (ultragarsinis arba infraraudonųjų sp. jutiklis)	2	Vidurkis, kanalų skirtumai	(0-2) Hz	(1 – 30) cm	2	0,2
7.	Polinkio kampas (Giroskopas)	1	Vidurkis	(0-10) Hz	(0-180) laipsn.	1	0,2
8.	Oro drėgmė	2	Vidurkis, kanalų skirtumai	(0-0,1) Hz	(10-70) %	10	1
9.	Nuolatinė įtampa	3	Vidurkis	(0-0,1) Hz	(0.1 – 25) V	2	1
10.	Nuolatinė srovė	3	Vidurkis	(0-0,1) Hz	(0.1 – 100) mA	2	1
11.	Dažnis	1	Vertė	(0.01-10) kHz	(0.1-5) V	1	
12.	Kintama įtampa	1	Vidutinė kvadratinė vertė, amplitudė	(0.05-3) kHz	1 mV- 10 V	1	0,2
13.	Kintama srovė	1	Vidutinė kvadratinė vertė, amplitudė	(0.05-3) kHz	(0.1- 100) mA	1	0,2
14.	Varža	2	Vidurkis	(0-10) Hz	1 Ω – 100 kΩ	1	1
15.	Talpumas	1	Vidurkis	(0-10) Hz	1 nF – 10 μF	1	1
16.	Rato apskukų greičio matavimas (herkonas)	1	Vidurkis	-	(0-30km/val.)	1	0,2
17.	Garsas (Elektretinis/MEMS mikrofonas)	1	Vidutinė kvadratinė vertė	(0.1-5.0) kHz	parinkti	1	0,5
18.	Vibracijos (Akselerometras)	1	Vidutinė kvadratinė vertė	(0-400) Hz	parinkti	3	0,1
19.	Temperatūra	2	Imties vertė, kanalų diapazonai (maksimumas, minimumas)	(0-0,5) Hz	-0 .. +75 C	2	4

20.	Barometrinis slėgis	1	Diapazonas (maksimumas, minimumas)	(0-10) Hz	(90-110) kPa	3	2
21.	Apšviestumas	2	Vidurkis, diapazonas (maksimumas, minimumas)	(0-10) Hz	parinkti	2	0,2
22.	Atstumas (ultragarsinis arba infraraudonųjų sp. jutiklis)	2	Vidurkis, kanalų skirtumai	(0-2) Hz	(1 – 30) cm	2	0,2
23.	Polinkio kampas (Giroskopas)	1	Vidurkis	(0-50) Hz	(0-180) laipsn.	2	0,2
24.	Oro drėgmė	2	Vidurkis, kanalų skirtumai	(0-0,01) Hz	(30-70) %	10	1
25.	Nuolatinė įtampa	2	Vidurkis	(0-0,1) Hz	(0.1 – 25) V	2	1
26.	Nuolatinė srovė	3	Vidurkis	(0-0,1) Hz	(0.1 – 100) mA	2	1
27.	Dažnis	1	Vertė	(0.1-100) kHz	(0.1-0.5) V	1	
28.	Kintama įtampa	1	Vidutinė kvadratinė vertė, amplitudė	(0.05-1) kHz	1 mV- 10 V	0,5	0,2
29.	Kintama srovė	1	Vidutinė kvadratinė vertė, amplitudė	(0.05-5) kHz	(0.1- 500) mA	0,5	0,2
30.	Varža	2	Vidurkis	(0-1) Hz	0.1 Ω – 10 k Ω	0,5	1
31.	Talpumas	2	Vidurkis, kanalų skirtumas	(0-1) Hz	10 nF – 100 μ F	0,5	1
32.	Rato apskukų greičio matavimas (herkonas)	1	Vertė	-	(0-60 aps./min.)	0,5	0,2

Atliekant užduotį 2 studentams reikia realizuoti abiejų studentų užduoties variantus.

3 Vertinimo skalė (rubrika)

Realizuotos funkcijos	Skiriamas maksimalus balas
Sudaryta principinė schema ir pagrindinė programa	1
Jutiklių duomenys nuskaitomi	1
Požymio skaičiavimai realizuoti ir testuoti	1
Matavimo tikrinimas kraštinuose diapazono taškuose	1
Duomenų perdavimas į kompiuterį užduotu periodu	2
Atvaizdavimas indikatoriuje užduotu periodu	2
Tarpinės ataskaitos pateikimas laiku	1
Galutinė ataskaita pateikiama laiku	1
	10 (maksimalus balas)

4 Indikatoriai

4.1 Mokomoji medžiaga

LCD modulis arba OLED modulis

Mokomieji video:

1. [OLED display with STM32 || I2C || CubeMx || Keil u5](#)
2. [SSD1306 OLED and STM32 || 128x64 || SW4STM || CubeMX](#)
3. <https://controllerstech.com/stm32/>
4. <https://controllerstech.com/oled-display-using-i2c-stm32/>

4.2 LCD indikatoriaus su HD44780 valdikliu prijungimas prie STM32 mikroprocesorių

HD44780 (išorinio įrenginio valdiklis/protokolas) arba su juo suderinami protokolai Sintronix ST7066U, Samsung KS0065B

STM32F, STM32F4 architecture , microprocessor family

I2C arba „parallel 4 bit“ (fizinės sąsaja)

HAL - periferinių įrenginių tvarkyklės (driver libraries of microprocessor)

Paieškos frazė (google search): „**HD44780 STM32F4 I2C HAL** compatible library“:

1. STM32 LCD 16×2 Tutorial & Library | Alphanumeric LCD 16×2 Interfacing
<https://deepbluembedded.com/stm32-lcd-16x2-tutorial-library-alphanumeric-lcd-16x2-interfacing/>
(ECUAL.rar)
2. <https://github.com/4ilo/HD44780-Stm32HAL> (HD44780-Stm32HAL-master.zip)
3. <https://adastra-soft.com/hd44780-library-for-stm32/>
4. Interface LCD 16×2 via I2C with STM32 <https://controllerstech.com/i2c-lcd-in-stm32/> (I2C, STM32CUBEIDE)

Kuo gali skirtis internete rastos atviros prieigos bibliotekos:

1. Patikimumo lygiu (veikti arba neveikti arba veikti nestabiliai)
2. Projekto paruošimo kažkuriai projektavimo aplinkai (Keil, STM32CubeIDE, ...)
3. Dokumentacijos lygiu
4. Sprendimo efektyvumu (Greitaveika (panaudojanti HAL / LL bibliotekas arba tiesiogine kreiptimi į registrus), naudojant vėlinimus/pertrauktis arba tikrinant būsenos bitus)

4.3 OLED indikatoriaus su SSD1306 valdikliu prijungimas prie STM32 mikroprocesorių

1. OLED display using I2C with STM32 <https://controllerstech.com/oled-display-using-i2c-stm32/>
2. STM32 open source code-0.96 inch OLED display SPI
<https://www.programmersought.com/article/18983663074/>

5 Duomenų perdavimas į kompiuterį

5.1 USART sąsajos panaudojimas

Darbe panaudokite mikrovaldiklio UART sąsają duomenų perdavimui į kompiuterį. Priklausomai nuo naudojamo maketo mikrovaldiklio sąsają galima prijungti vienu iš būdų:

1. UART sąsajos signalai prijungti prie USB hub mikroschemos, kuri panaudota integruoto adapterio (on-board ST-Link) realizavimui. Šiuo būdu realizuojamas virtualus COM portas (Virtual COM Port).
 - 1.1. NUCLEO-L073RZ maketuose šis variantas yra paruoštas pagal nutylėjimą (UM1724 User manual STM32 Nucleo-64 boards (MB1136) 6.8 USART communication).

6.8 USART communication

The USART2 interface available on PA2 and PA3 of the STM32 microcontroller can be connected to ST-LINK MCU, ST morpho connector, or to ARDUINO® connector. The choice can be changed by setting the related solder bridges. By default, the USART2 communication between the target STM32 and ST-LINK MCU is enabled, in order to

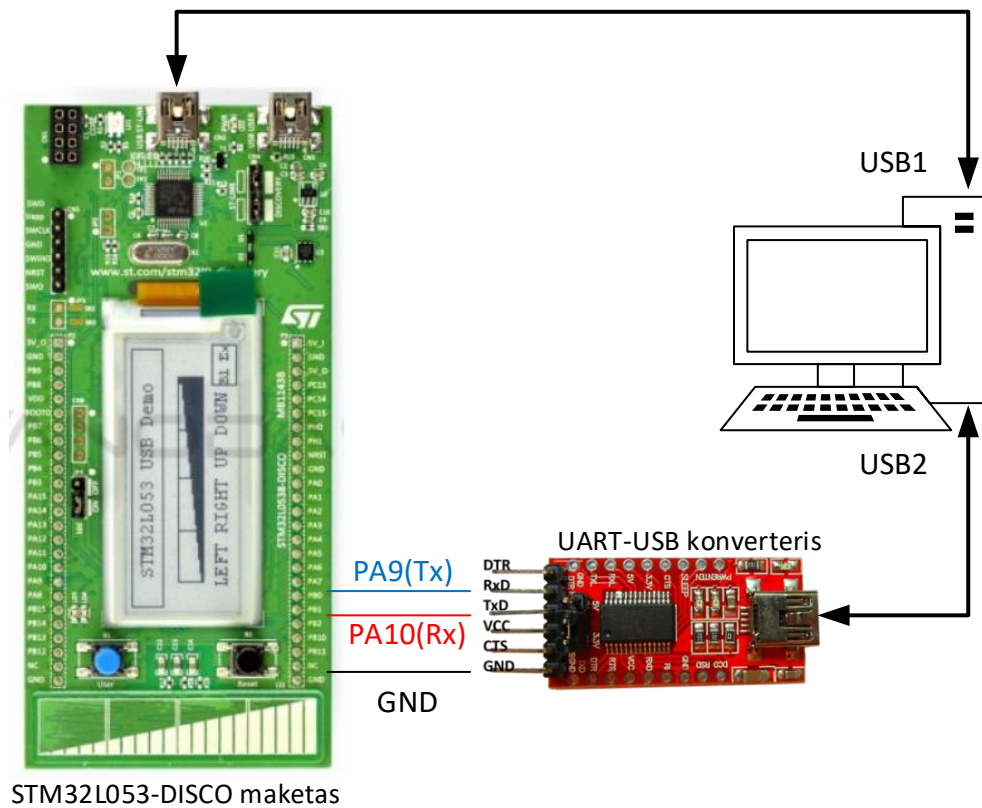
- 1.2. Toks variantas iš karto nėra paruoštas **STM32L053-DISCO** makete, bet yra galimas, užtrumpinus dvi kontaktines aikšteles, kaip aprašyta maketo vartotojo vadove (4.11 paragrafas *UM1775 User manual. Discovery kit for STM32L0 series with STM32L053C8 MCU*).
2. UART sąsajos signalus (USARTx_RX, USARTx_TX ir GND) per išorinį UART-USB keitiklį prijungti prie USB sąsajos. Toks variantas yra galimas **STM32L053-DISCO** arba **STM32F4DISCOVERY** maketuose.

UART-USB konverterių (keitiklių) pavyzdžiai (šie keitikliai dažniausiai realizuojami FTDI mikroschemų <https://ftdichip.com/> pagrindu):

1. <https://www.anodas.lt/ft232-ftdi-usb-ttl-5v-3-3v-konverteris>
2. <https://rcl.lt/products/akad012-konverteris-usb-rs232-ttl-pl2303hx-3-3v-ir-5v>

5.2 UART ryšys panaudojant STM32L053-DISCO

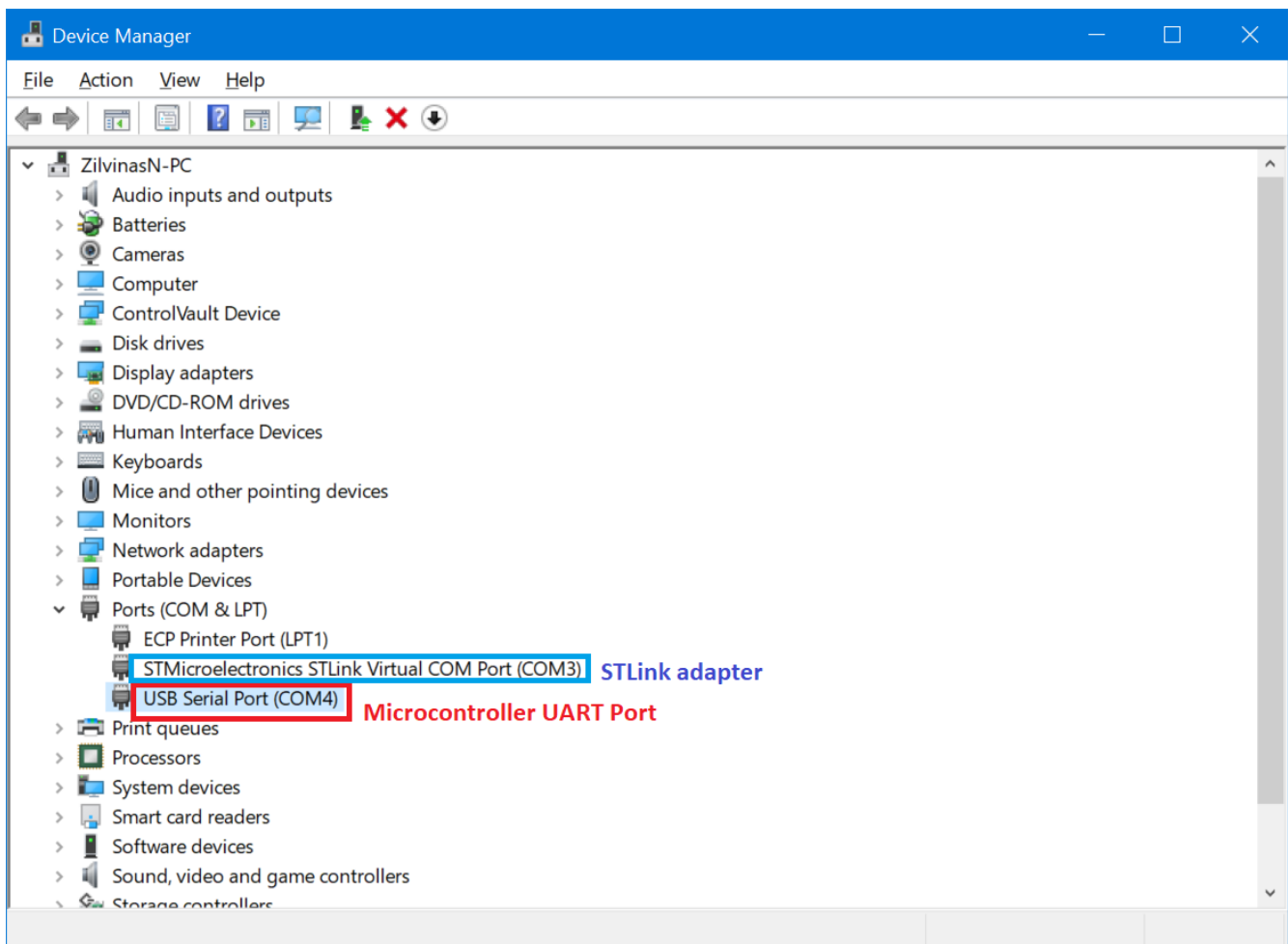




3 pav. STM32L053-DISCO maketo prijungimas prie kompiuterio USB per RS232-USB konverterį

5.3 COM prievado aptikimas Windows operacijų sistemoje

Prijungus maketą ir UART-USB konverterį **Windows Device Manager** programoje jie abu turi būti matomi **Port (COM & LPT)** skyrelyje. Čia mums svarbu įsidėmėti, kad vartotojo prievadui priskirtas numeris COM4, nes jį reikės nurodyti terminalinėje programoje.

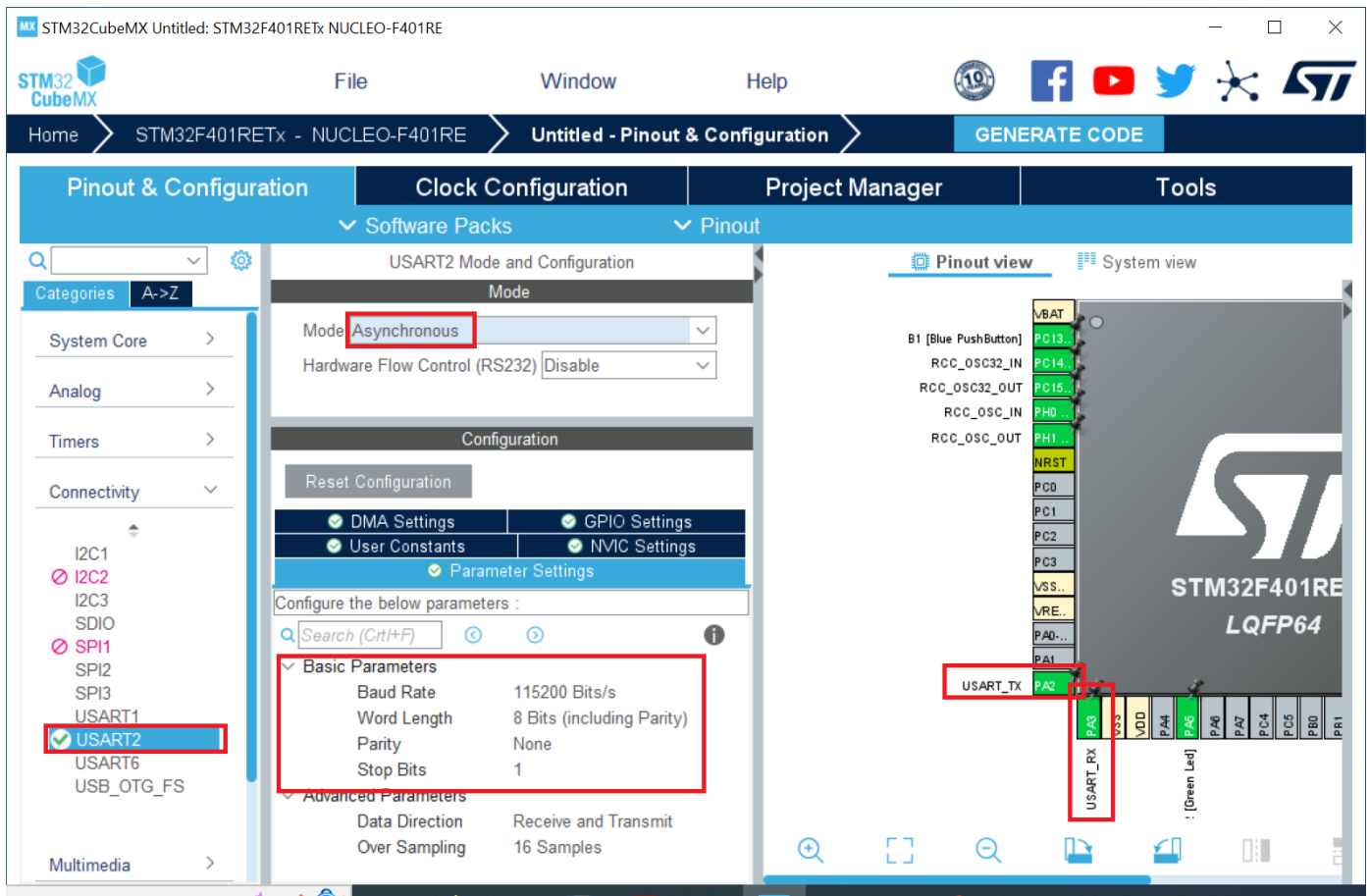


5.4 UART ryšys su kompiuteriu Nucleo tipo maketuose

5.4.1 Nucleo-F401RE (Nucleo-64 plokštės tipas)

Pagal nutylėjimą šio tipo plokštėse STM32 mikrovaldiklio USART2 sąsaja (prievadai PA2 ir PA3) yra prijungta prie ST-LINK adapterio virtualaus COM porto realizavimui.

Daugiau informacijos skaitykite **STM32 Nucleo-64 boards (MB1136) User manual** (6.8 paragrafas 25 psl.) https://www.st.com/resource/en/user_manual/um1724-stm32-nucleo64-boards-mb1136-stmicroelectronics.pdf



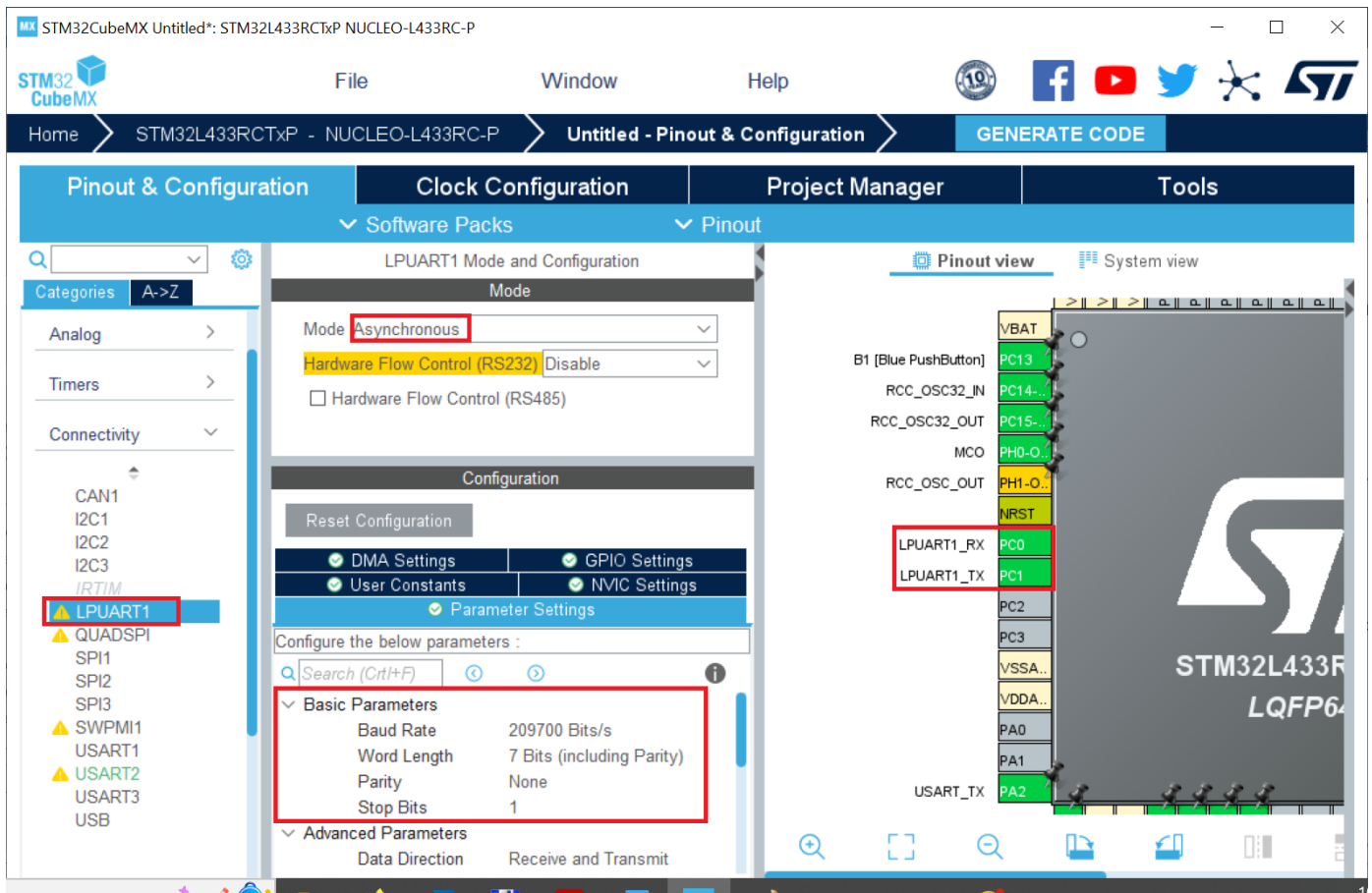
5.4.2 Nucleo-F433CP (Nucleo-64-P MB1319 tipas)

Šio tipo maketuose pagal nutylėjimą:

- STM32 mikrovaldiklio **LPUART1** yra prijungtas prie ST-LINK adapterio virtualaus COM (Virtual COM) realizavimui. Norint naudoti šį prievadą **UART-USB adapteris nereikalingas**, nes jungiame per tą pačią USB jungtį, kaip ir ST-LINK adapterį.
- STM32 mikrovaldiklio **USART1** yra prijungtas prie ARDUINO ir ST morpho jungčių. Norint per šią sąsają jungtis prie kompiuterio USB prievado yra **reikalingas UART-USB adapteris**.

Daugiau informacijos skaitykite **STM32 Nucleo-64-P boards (MB1319) User manual** (6.9 paragrafas 28 psl.)

https://www.st.com/resource/en/user_manual/um2206-stm32-nucleo64p-boards-mb1319-stmicroelectronics.pdf



5.4.3 Nucleo-L073 (Nucleo-64 plokštės tipas)

NUCLEO-L073RZ , STM32 Nucleo-64 development board with STM32L073RZ MCU:

<https://www.st.com/en/evaluation-tools/nucleo-l073rz.html>

6.8 USART communication

The USART2 interface available on PA2 and PA3 of the STM32 microcontroller can be connected to ST-LINK MCU, ST morpho connector, or to ARDUINO® connector. The choice can be changed by setting the related solder bridges. By default, the USART2 communication between the target STM32 and ST-LINK MCU is enabled, in order to support virtual COM port for Mbed™ (SB13 and SB14 ON, SB62 and SB63 OFF). If the communication between the target STM32 PA2 (D1) or PA3 (D0) and shield or extension board is required, SB62 and SB63 must be ON, while SB13 and SB14 must be OFF. In such a case, it is possible to connect another USART to ST-LINK MCU using flying wires between the ST morpho connector and CN3. For instance, on NUCLEO-F103RB it is possible to use USART3 available on PC10 (TX) and PC11 (RX). Two flying wires need to be connected as follow:

- PC10 (USART3_TX) available on CN7 pin 1 to CN3 pin RX
- PC11 (USART3_RX) available on CN7 pin 2 to CN3 pin TX

5.4.4 Paklauskime ChatGPT

Question: „How to use USART communication between NUCLEO-L073RZ and PC terminal?“

5.4.5 ChatGPT atsakymas

To establish USART communication between the **NUCLEO-L073RZ** board and a **PC terminal**, follow these steps:

1. Connect the NUCLEO-L073RZ to the PC

The **NUCLEO-L073RZ** features an **ST-LINK/V2-1** debugger, which provides a **virtual COM port (VCP)** over USB. This allows USART communication without additional hardware.

1. **Connect the board** to your PC via **USB** (use the micro-USB connector labeled ST-LINK).
2. The board should appear as a virtual COM port on your PC.

2. Identify the COM Port on the PC

1. Open **Device Manager** on Windows.
2. Expand **Ports (COM & LPT)**.
3. Look for "**STMicroelectronics STLink Virtual COM Port (COMx)**".
4. Note the **COM port number** (e.g., COM3).

3. Configure the USART in Software

You need to configure USART in your firmware using **STM32CubeIDE** or **Keil**.

Using STM32CubeIDE:

1. Open **STM32CubeMX** (integrated in STM32CubeIDE).
2. Select **NUCLEO-L073RZ** as the target board.
3. Go to **Connectivity** → **USART2 (Asynchronous Mode)**.
 - Enable **TX** and **RX**.
 - Set baud rate to **115200** (or any preferred rate).
 - Keep **8-N-1** format (**8 data bits, no parity, 1 stop bit**).
 - Select **Hardware Flow Control** as **None**.
4. Click **Project** → **Generate Code**.

Code Example (HAL Library)

Modify `main.c` to send and receive data:

```
#include "usart.h"

uint8_t rxData;
char txMessage[] = "Hello from NUCLEO-L073RZ\r\n";

int main(void) {
    HAL_Init();
    SystemClock_Config();
    MX_USART2_UART_Init();

    HAL_UART_Transmit(&huart2, (uint8_t*)txMessage, sizeof(txMessage) - 1,
    HAL_MAX_DELAY);

    while (1) {
        if (HAL_UART_Receive(&huart2, &rxData, 1, HAL_MAX_DELAY) == HAL_OK) {
            HAL_UART_Transmit(&huart2, &rxData, 1, HAL_MAX_DELAY); // Echo received
            character
        }
    }
}
```

4. Open a Serial Terminal on PC

Use a terminal emulator like **PuTTY**, **Tera Term**, or **Arduino Serial Monitor**.

1. Select the correct **COM port**.
2. Set **Baud rate: 115200** (match firmware setting).
3. Set **Data: 8 bits, Stop: 1 bit, Parity: None**.
4. Click **Open**.
5. Test Communication

- The NUCLEO board should send "Hello from NUCLEO-L073RZ".
- When you type in the terminal, the board should echo back the characters.

Optional: Use an External USB-to-Serial Adapter

If USART2 is unavailable, you can use **USART1 (PA9: TX, PA10: RX)** with an external **USB-to-Serial adapter** (e.g., CP2102, FTDI).

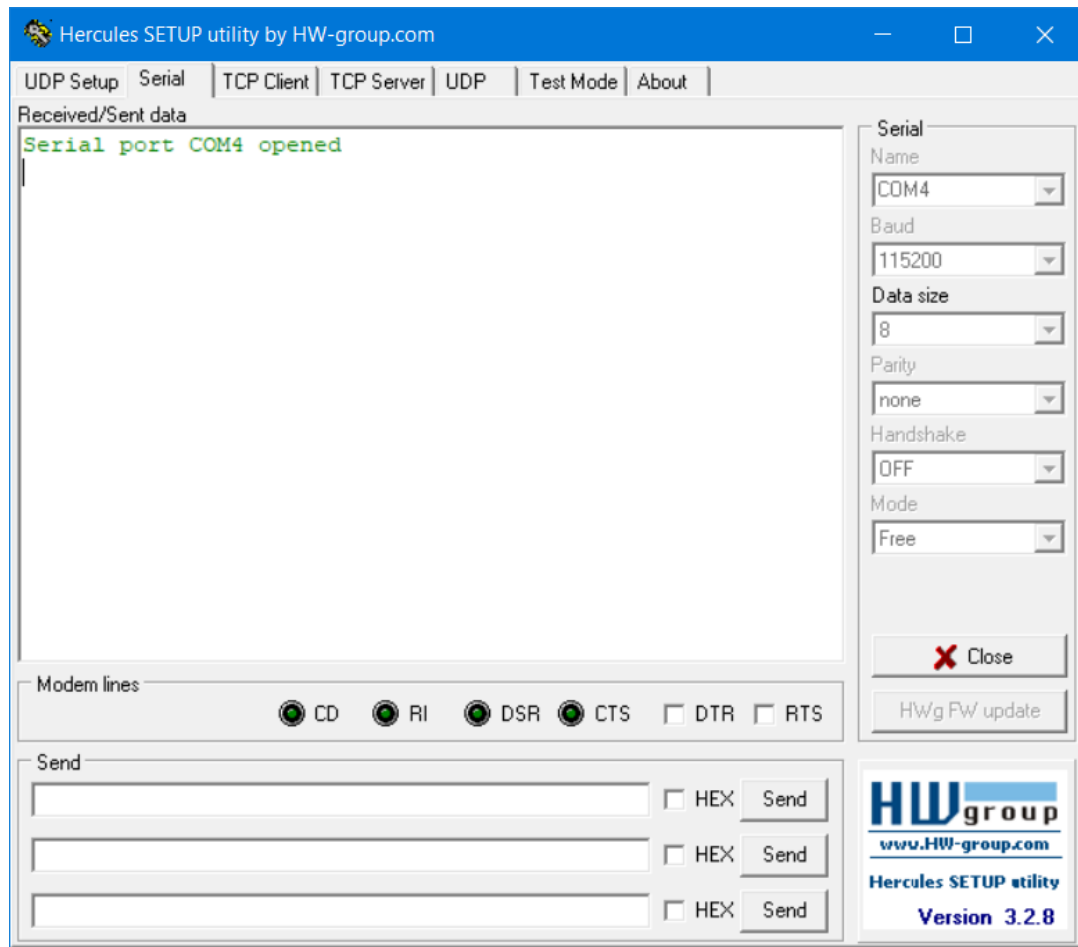
1. Connect:
 - **PA9 (TX) → USB-TTL RX**
 - **PA10 (RX) → USB-TTL TX**
 - **GND → GND**
2. Open the terminal and repeat the steps above.

5.5 Terminalinės programos naudojimas

Kompiuteryje naudojama terminalinė programa duomenų priėmimui ir išsaugojimui. Galimų panaudoti atvirojo kodo programų pavyzdžiai yra:

1. Tera Term https://download.cnet.com/Tera-Term/3000-2094_4-75766675.html
2. Hercules Setup Utility <https://www.hw-group.com/software/hercules-setup-utility>

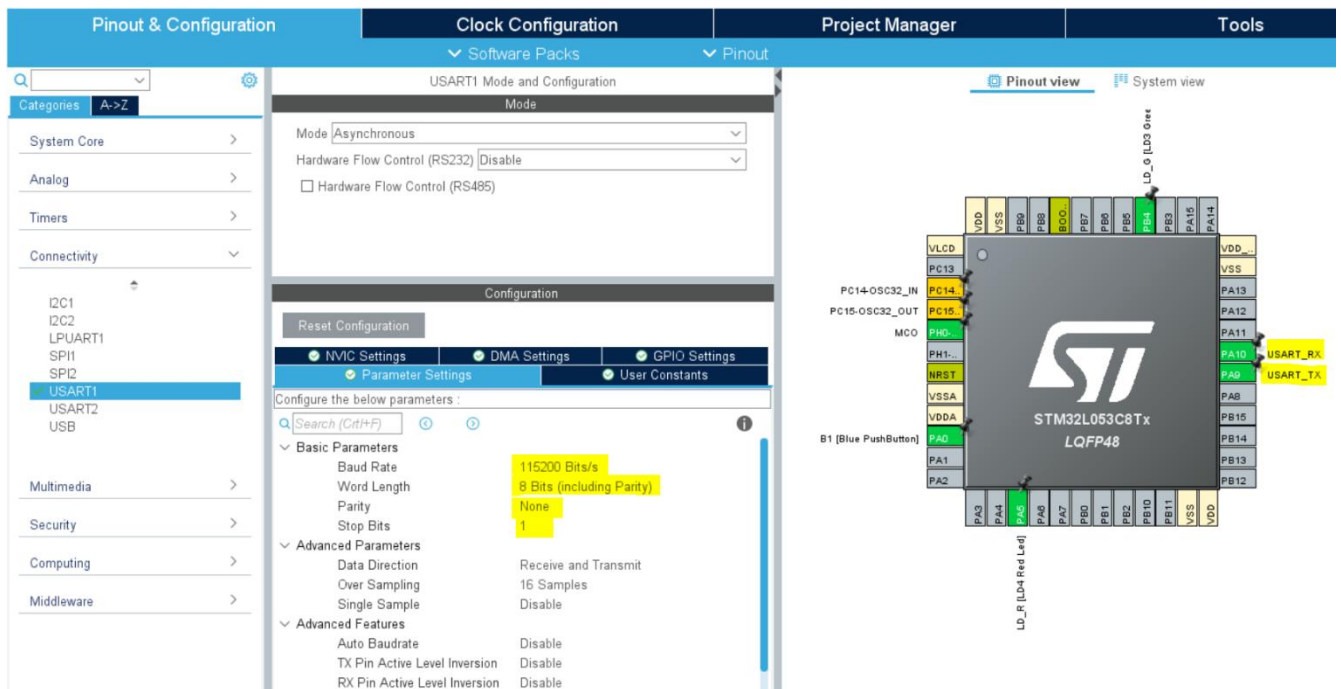
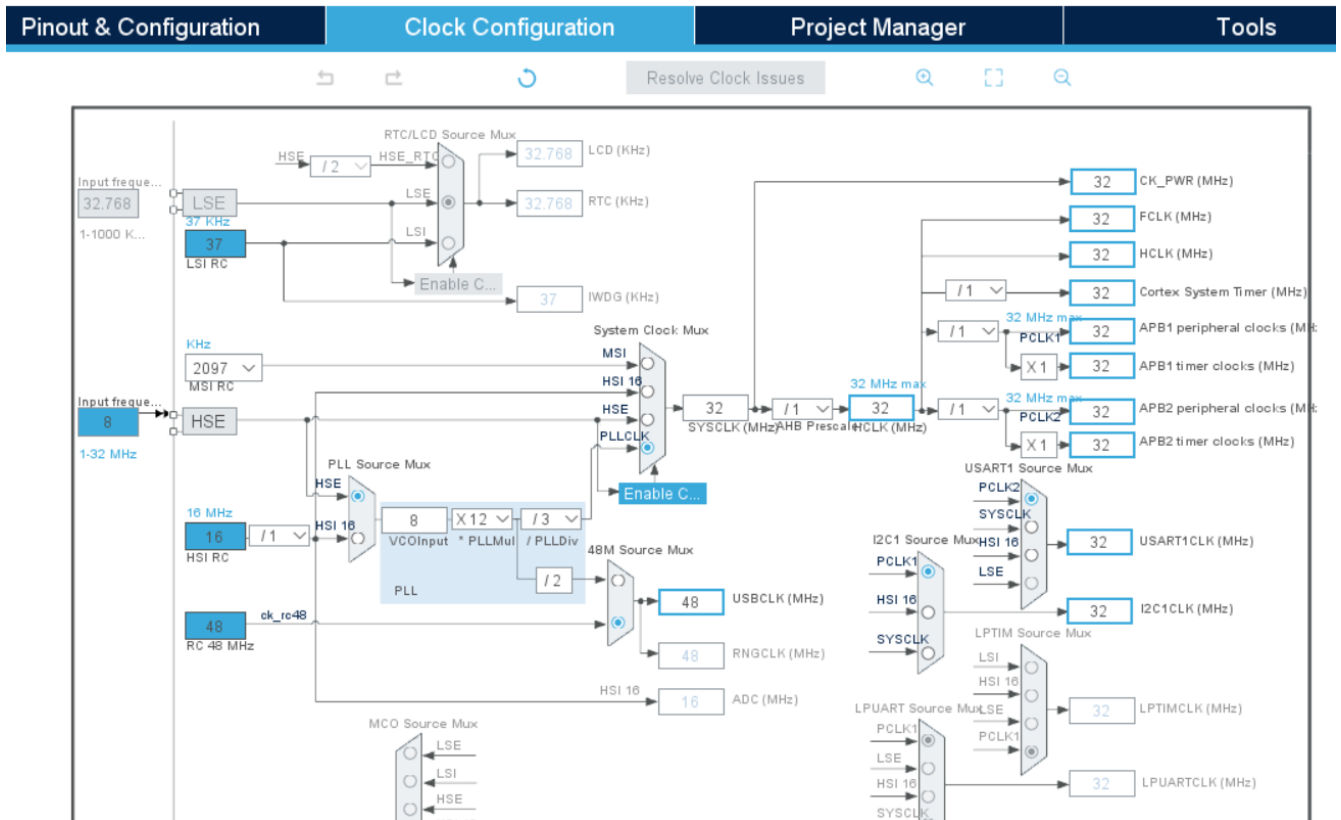
Žemiau parodytas **Hercules** terminalinės programos langas, pasiruošęs priimti duomenis per COM4 prievadą.



5.6 UART komunikacijų programavimas mikrovaldiklyje

5.6.1 USART sąsajos konfigūravimas

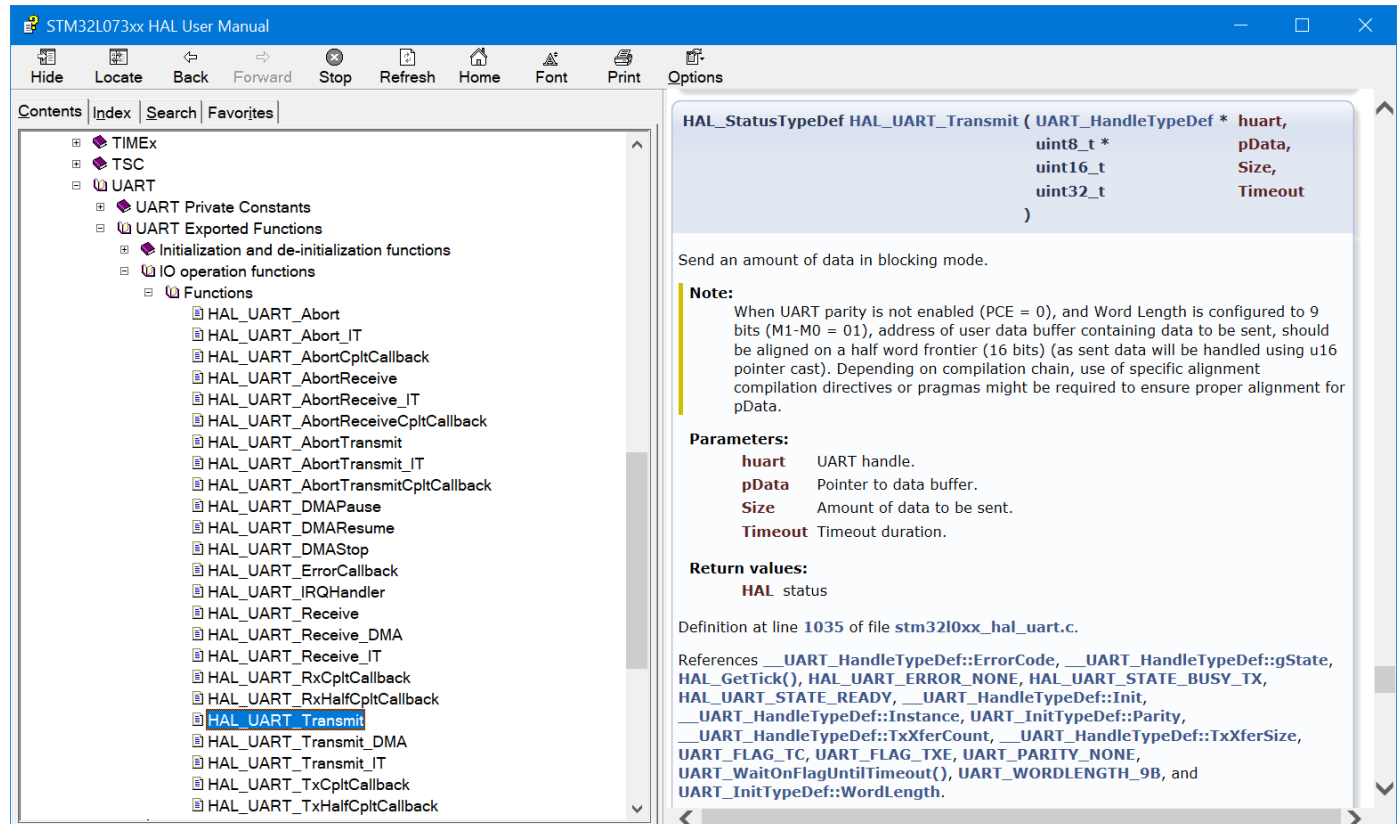
Naudokimės STM32CubeMX aplinka.



4 pav. STM32L053-DISCO UART1 sąsajos konfigūravimas

5.6.2 USART programavimas HAL funkcijų pagalba

5.6.2.1 HAL funkcijų UART skaitymui /rašymui santrauka



Išsiuntimo funkcijos:

- [HAL UART Transmit](#)
- [HAL UART Transmit IT](#)
- [HAL UART Transmit DMA](#)
- [HAL UART TxCpltCallback](#)

Priėmimo funkcijos:

- [HAL UART Receive](#)
- [HAL UART Receive IT](#)
- [HAL UART Receive DMA](#)
- [HAL UART RxCpltCallback](#)

5.6.2.2 Programos fragmentas su blokuojančiomis funkcijomis

```
/* USER CODE BEGIN Includes */
```

```

#include <stdio.h>
#include <string.h>
/* USER CODE END Includes */

/* Private user code -----*/
/* USER CODE BEGIN 0 */
uint8_t TxBuffer[20];
uint8_t RxBuffer[20];
int i;
void HandleError()
{
    uint32_t uart_err;
    uart_err=HAL_UART_GetError(&huart1);
}

/* USER CODE END 0 */

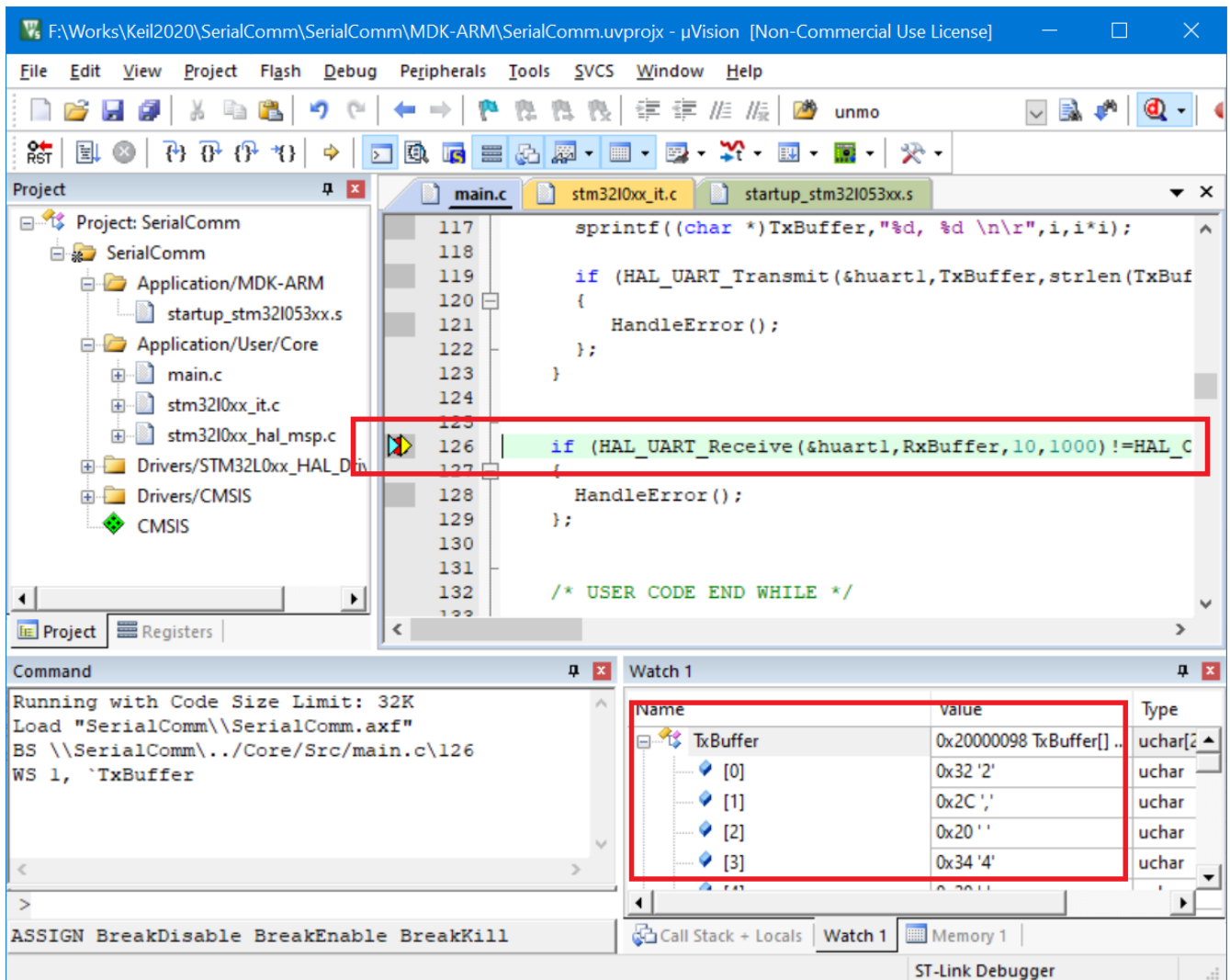
int main(void)
{
    sprintf((char *)TxBuffer,"Hello STM32 \n\r");
    if (HAL_UART_Transmit(&huart1,TxBuffer,strlen(TxBuffer),1000)!=HAL_OK)
    {
        HandleError();
    };

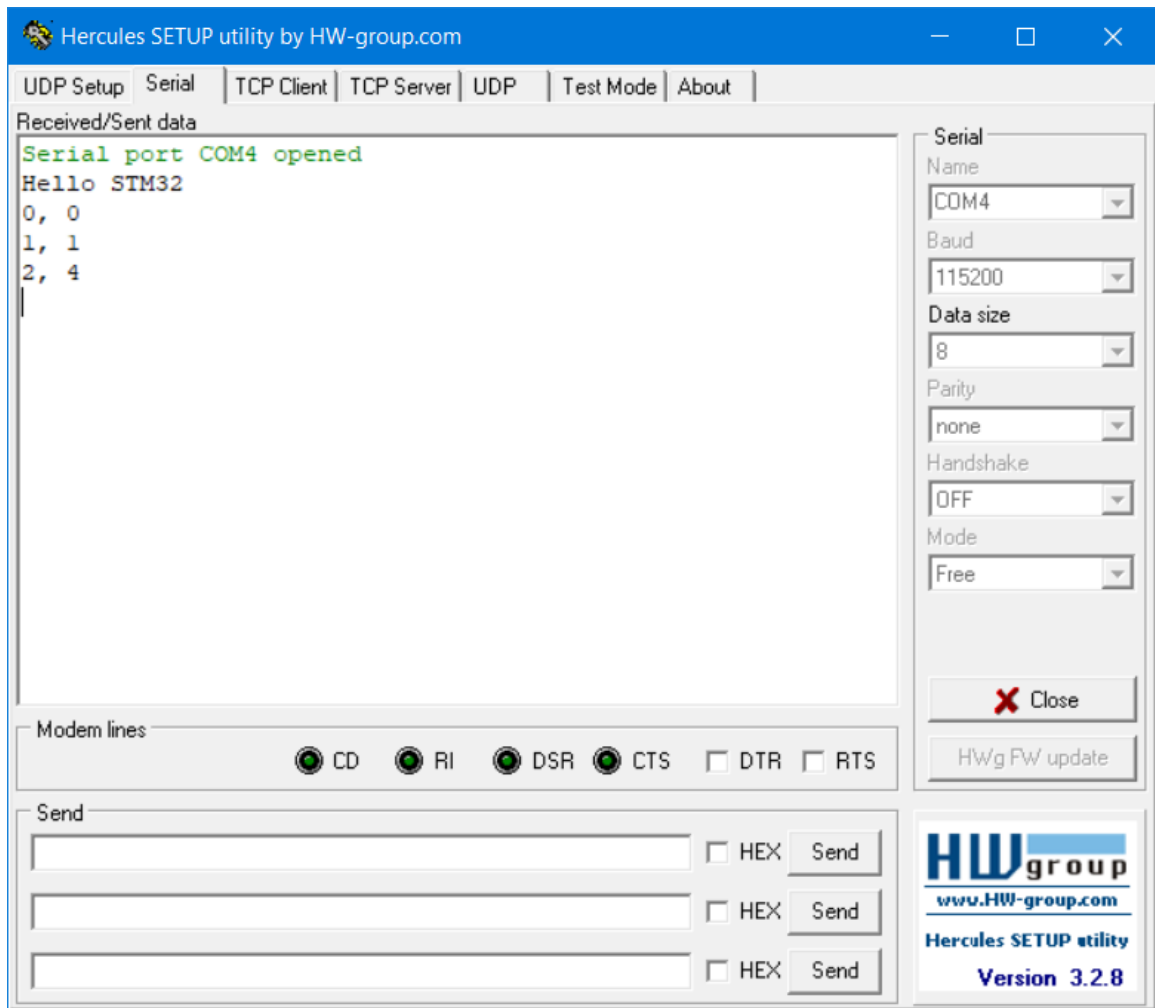
    while (1)
    {
        for (i=0;i<3;i++)
        {
            sprintf((char *)TxBuffer,"%d, %d \n\r",i,i*i);
            if (HAL_UART_Transmit(&huart1,TxBuffer,strlen(TxBuffer),1000)!=HAL_OK)
            {
                HandleError();
            };
        }

        if (HAL_UART_Receive(&huart1,RxBuffer,10,1000)!=HAL_OK)
        {
            HandleError();
        };
    }
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

```

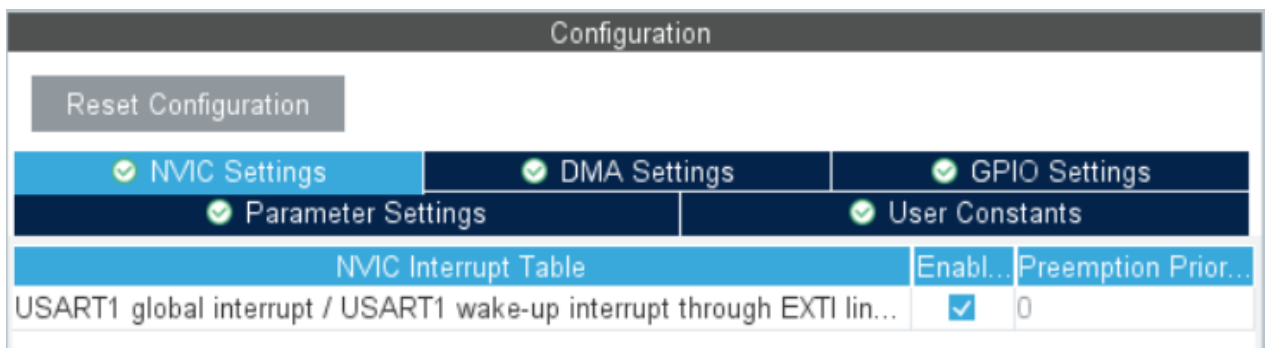
Programą įvykdžius iki eilutės, kurioje parodytas stabdymo taškas, terminalinėje programoje matome perduotus iš mikrovaldiklio duomenis:





5.6.2.3 Programos fragmentas su neblokuojančiomis funkcijomis

Reikia aktyvuoti pertraukties aptarnavimą ir realizuoti **HAL_UART_TxCpltCallback** funkciją.



```

/* USER CODE BEGIN Includes */
#include <stdio.h>
#include <string.h>
/* USER CODE END Includes */

/* USER CODE BEGIN 0 */
uint8_t TxBuffer[20];
uint8_t RxBuffer[20];
int i;
typedef enum {UART_READY,UART_BUSY,UART_FINISHED} UART_Status;
UART_Status Status=UART_READY;
long WorkCounter=0;

/*
 * Implement Transmit complete callback function
 */
void HAL_UART_TxCpltCallback(UART_HandleTypeDef *huart)
{
    if (Status==UART_BUSY)
        Status=UART_READY; // transmission finished, can send more if needed
}

void HandleError()
{
    uint32_t uart_err;
    uart_err=HAL_UART_GetError(&huart1);
}

int main(void)
{
    i=0;
    while (1)
    {
        if (Status==UART_READY)
        {
            sprintf((char *)TxBuffer,"%d, %d \n\r",i,i*i);
            if (HAL_UART_Transmit_IT(&huart1,TxBuffer,strlen(TxBuffer))!=HAL_OK)
            {
                HandleError();
            } else
            {
                Status=UART_BUSY; // sending data

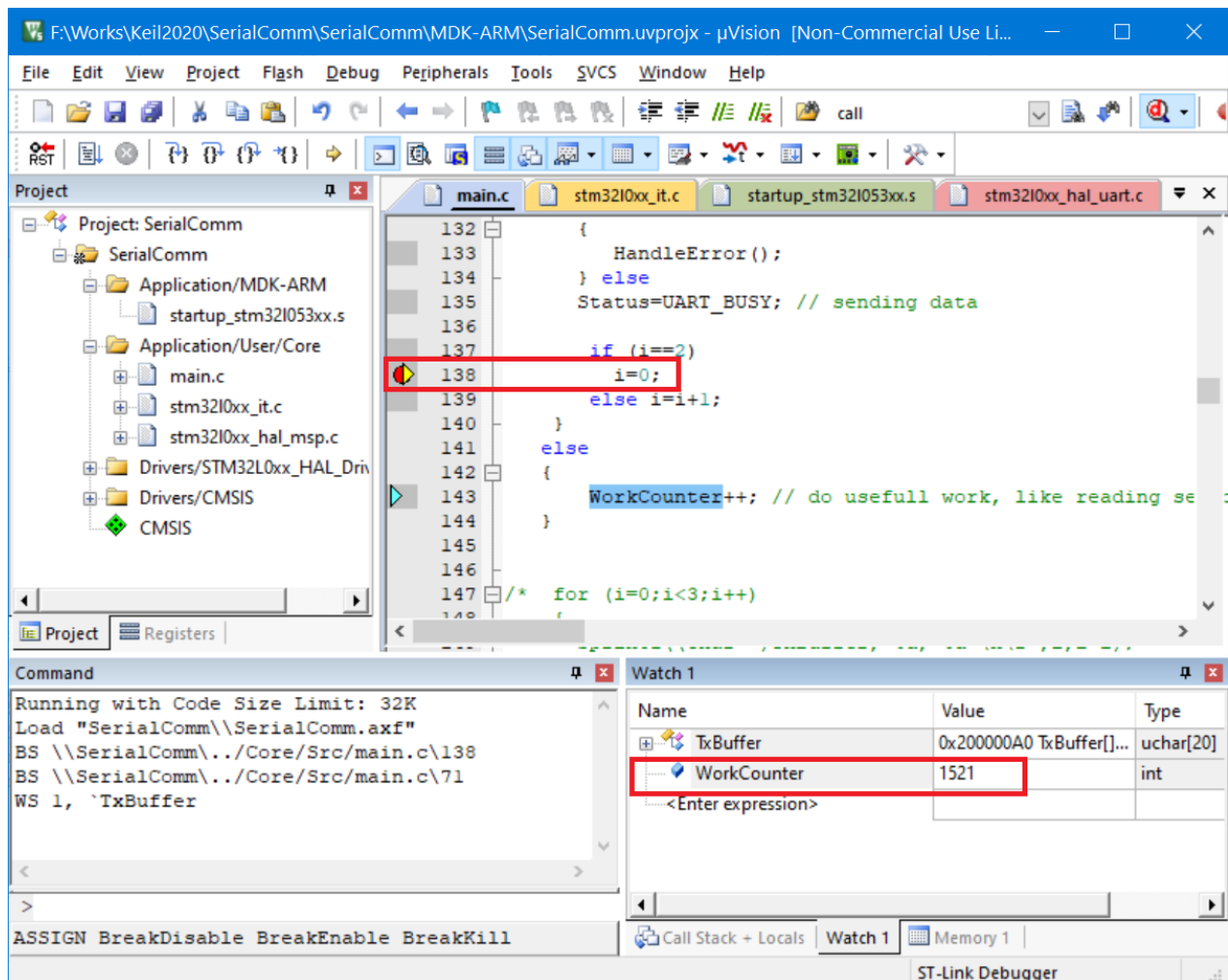
                if (i==2)
                {
                    i=0;
                    Status=UART_FINISHED; // stop sending
                }
                else i=i+1;
            }
        } else
        {
            WorkCounter++; // do usefull work, like reading sensors
        }
    }
}
/* USER CODE END WHILE */

```

```

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

```



WorkCounter pasiekta reikšmė rodo, kiek procesorius galėjo skirti resursų, kol UART periferinis įrenginys siuntė duomenis į kompiuterį.

5.6.2.4 Programos fragmentas naudojant DMA kanalą

5.7 Pavyzdžiai

Send and Receive data to PC without UART (STM32 USB COM) <https://controllerstech.com/send-and-receive-data-to-pc-without-uart-stm32-usb-com/>

Tutorial about data sending over USB: <https://www.youtube.com/watch?v=92A98iEFmaA>

5.8 Įrenginio paklaidos skaičiavimo pavyzdžiai

Netiesioginių matavimų pavyzdys

Tegul matuojame judėjimo greitį v , tiesiogiai matuodami rato diametrą d ir vieno apsisukimo laiką T .

$$v = \frac{\pi d}{T}$$

Greičio paklaida

$$\sigma v = \sqrt{\left(\frac{\sigma v}{\sigma d}\right)^2 (\sigma d)^2 + \left(\frac{\sigma v}{\sigma T}\right)^2 (\sigma T)^2},$$

čia

$$\frac{\sigma v}{\sigma d} = \frac{\pi}{T}$$

$$\frac{\sigma v}{\sigma T} = -\frac{\pi d}{T^2}.$$

Diametro išmatavimo standartinis nuokrypis

$$\sigma d = \frac{\Delta d}{\sqrt{12}},$$

kur Δd yra liniuotės, su kuria išmatuotas diametras, padalos vertė.

$$\sigma T = \frac{\Delta T}{\sqrt{12}}$$

kur ΔT yra laikmačio atraminio taktavimo generatoriaus periodas.

Tegul rato diametras yra $d=0,7 \text{ m}$, liniuotės padala $\Delta d = 1 \text{ cm}$, išmatuotas laikas su laikmačiu $T=0,2 \text{ sek.}$, o laikmačio $\Delta T = 0,01 \text{ ms}$. Tuomet greitis

$$v = \frac{3,14 \cdot 0,7 \text{ m}}{0,2 \text{ s}} = 11 \text{ m/s}$$

paklaidą apskaičiuosime

$$\sigma v = \sqrt{\left(\frac{\pi}{T}\right)^2 \left(\frac{\Delta d}{\sqrt{12}}\right)^2 + \left(-\frac{\pi d}{T^2}\right)^2 \left(\frac{\Delta T}{\sqrt{12}}\right)^2} = \sqrt{\left(\frac{3,14}{0,2s}\right)^2 \left(\frac{10^{-2}}{\sqrt{12}}\right)^2 + \left(-\frac{3,14 \cdot 0,7m}{(0,2s)^2}\right)^2 \left(\frac{10^{-5}}{\sqrt{12}}\right)^2} = 4.53 \text{ cm/s},$$

```
T=0.2; % [sek.]
D=0.7; % [m]
dD=0.01; % [m]
dT=0.01e-3; % [sek.]
stdv=sqrt((pi/T)^2*((dD/sqrt(12))^2)+(-pi*D/T^2)^2*((dT/sqrt(12))^2))
```

Paklaidą rekomenduojama paskaičiuoti prie minimalaus ir maksimalaus planuojamo matuoti greičių.

6 Bendrieji reikalavimai techniniams sprendimams

Projektuojamas įrenginys turi būti paruoštas konfigūravimui:

1. Po pagaminimo gamykloje,
2. Eksploatacijos metu pas vartotoją.

Konfigūravimo (parametrizavimo) duomenys, pvz., PIN kodas, laikas ir data, įrenginio komunikacijų protokolo adresas, matavimo kanalų kalibravimo koeficientai, keitiklių charakteristikų ištiesinimo koeficientai ir pan., turi būti įvedami ir keičiami vykdant programą (*run-time*), o ne ją kompiliuojant (*design-time*). Konfigūravimo duomenys turi būti išsaugomi energetiškai nepriklausomoje atmintyje. Konfigūravimo duomenų įvedimui turi būti numatyta sąsaja, pvz., klaviatūra, mygtukai, komunikacijų sąsaja UART (per USB ar *Bluetooth*) su protokolu, palaikančiu įvedimą iš kompiuterio ar išmanaus įrenginio.

7 Ataskaitoje pateikti

1. Įrenginio principinę schemą
2. Komponentų parametrų skaičiavimą ir pagrindimą
3. Programos algoritmo aprašą
4. Komunikacijų su kompiuteriu protokolo aprašą
5. Įrenginio testavimą ir rezultatus
6. Matavimo paklaidos įvertinimą (jeigu matavimo uždavinys)
7. Išvadas (kur toks įrenginys gali būti taikomas)
8. Nuoroda į github svetainę, kurioje yra įkeltas programinės įrangos projektas