VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

ELEKTRONIKOS FAKULTETAS

ELEKTRONINIŲ SISTEMŲ KATEDRA

LABORATORINIS DARBAS 3

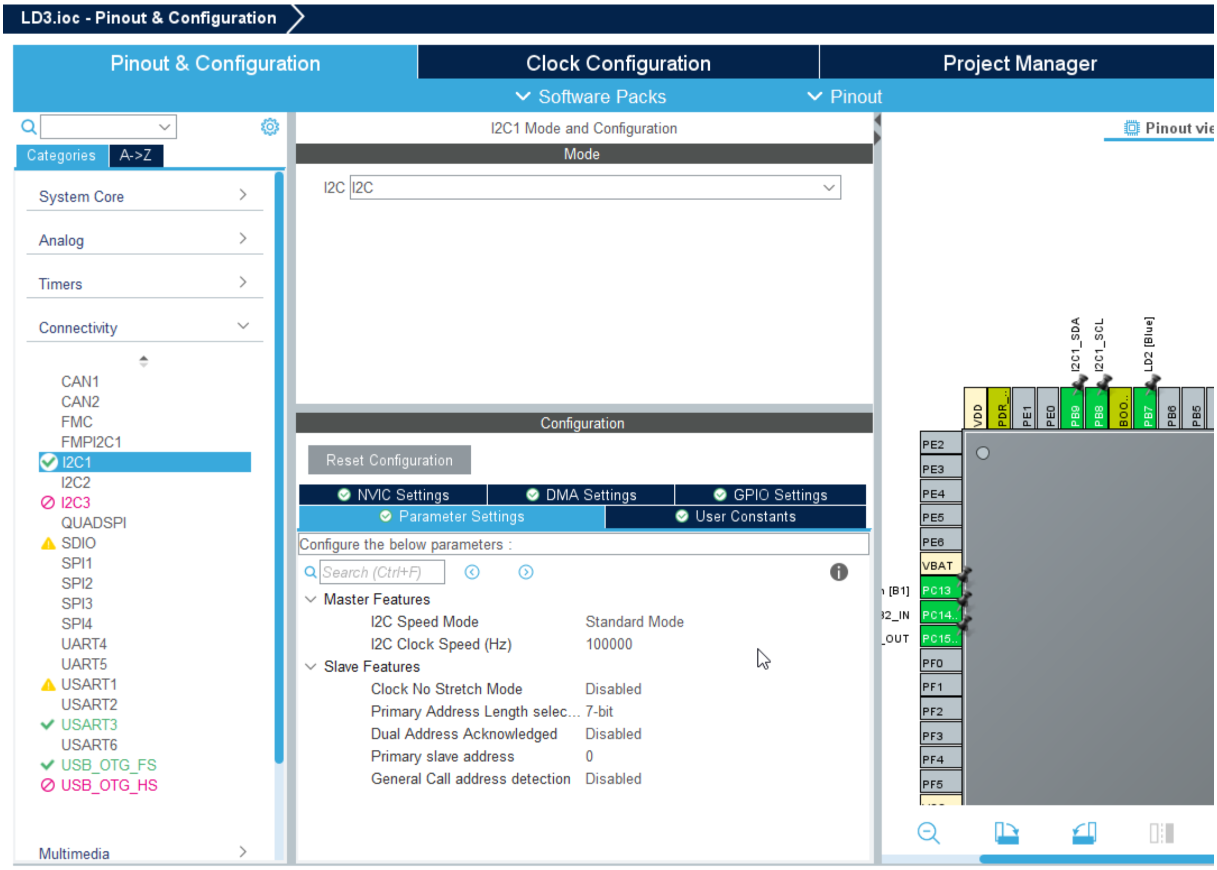
Įterptinių sistemų inžinerija

Atliko: EKSfm-24 gr. Ignas Malinauskas

Tikrino: dr. Eldar Šabanovič

VILNIUS 2025

1. Projekto konfigūracija (be pull-up)



I2C: I2C

Configure the below parameters :

Master Features	
I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000
Slave Features	
Clock No Stretch Mode	Disabled
Primary Address Length selec...	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

2. I^{2}C užduotis su MCP4725

Prie STM32F446ZE I^{2}C kaiščių D15(PB8) SCL ir D14(PB9) SDA prijungta 12-bit DAC MCP4725. Programoje paduota 2048 reikšmę (~1.6V) Fast-Mode formatu, kas yra 2 baitų komanda. Duomenys perduodami per tx_buffer masyvą, kur pirmam elemente iš 12 duomenų baitų paliekami 4 ir nustumiami į dešinę, paliekant viršų (MSB) komandų bitams. Likę 8 duomenų bitai įrašomi į 2 tx_buffer elementą.

1 baitas

- bit7,6 = 0, 0; reiškia greito rėžimo formatą
- bit5,4 - PD1, PD0 = 00; nurodo išvesties rėžimą į normalų
- bit3-0 – D11-D8 = duomenų bitai

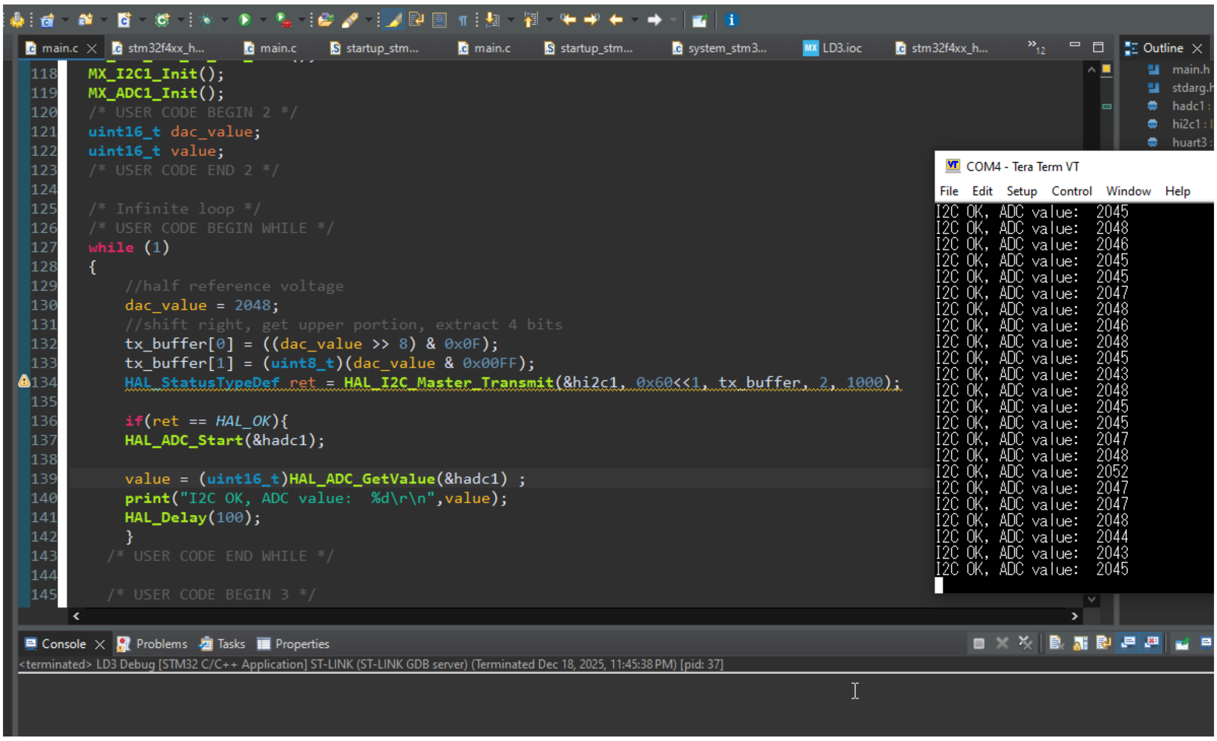
2 baitas

- bit7-0 – D7-D0 = duomenų bitai

Eilutėje HAL_I2C_Master_Transmit(&hi2c1, 0x60<<1, tx_buffer, 2, 1000); 0x60 bitai nurodo DAC adresą, kur nurodo kuris įrenginys dabar turės klausyti.

3. Programos kodas it Tera Term rezultatai

DAC išvedamą reikšmę nuskaito ADC ir rezultatas išvedamas per UART.

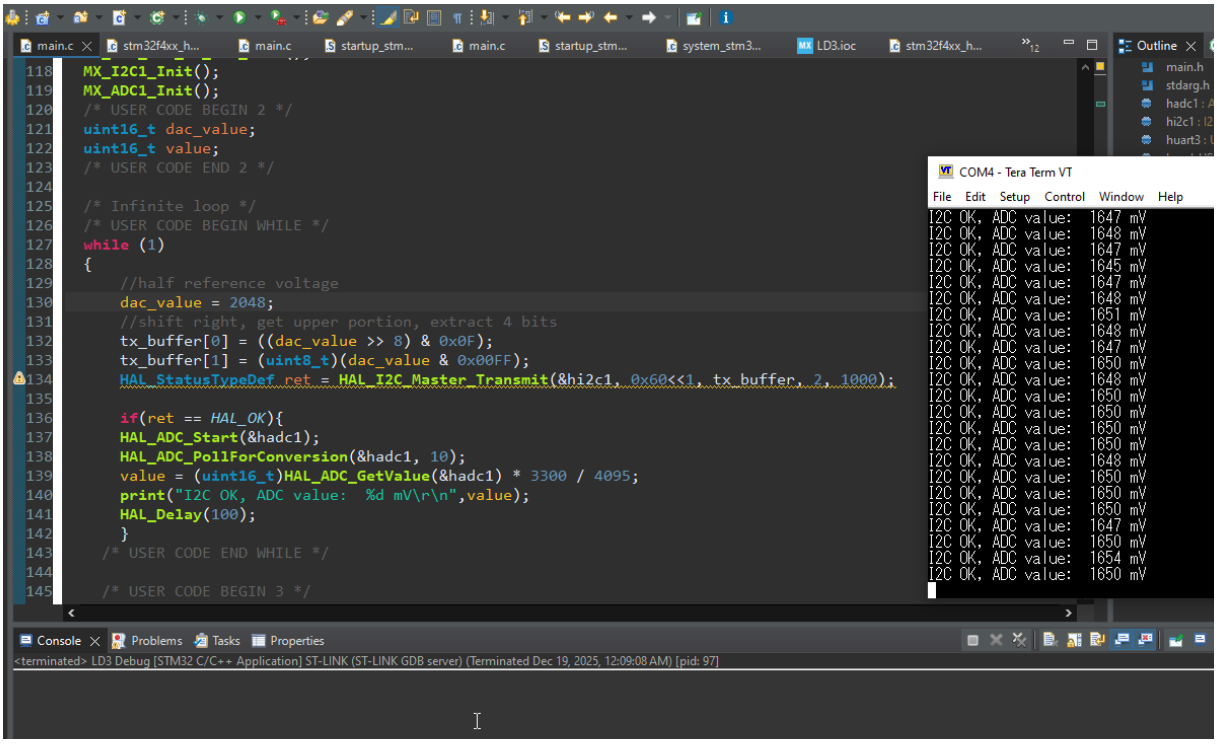


```c
MX_I2C1_Init();
MX_ADC1_Init();
/* USER CODE BEGIN 2 */
uint16_t dac_value;
uint16_t value;
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    //half reference voltage
    dac_value = 2048;
    //shift right, get upper portion, extract 4 bits
    tx_buffer[0] = ((dac_value >> 8) & 0x0F);
    tx_buffer[1] = (uint8_t)(dac_value & 0x00FF);
    HAL_StatusTypeDef ret = HAL_I2C_Master_Transmit(&hi2c1, 0x60<<1, tx_buffer, 2, 1000);

    if(ret == HAL_OK){
    HAL_ADC_Start(&hadc1);

    value = (uint16_t)HAL_ADC_GetValue(&hadc1) ;
    print("I2C OK, ADC value:  %d\r\n",value);
    HAL_Delay(100);
    }
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
```

COM4 - Tera Term VT

```
I2C OK, ADC value:  2045
I2C OK, ADC value:  2048
I2C OK, ADC value:  2046
I2C OK, ADC value:  2045
I2C OK, ADC value:  2045
I2C OK, ADC value:  2047
I2C OK, ADC value:  2048
I2C OK, ADC value:  2046
I2C OK, ADC value:  2048
I2C OK, ADC value:  2045
I2C OK, ADC value:  2043
I2C OK, ADC value:  2048
I2C OK, ADC value:  2045
I2C OK, ADC value:  2045
I2C OK, ADC value:  2047
I2C OK, ADC value:  2048
I2C OK, ADC value:  2052
I2C OK, ADC value:  2047
I2C OK, ADC value:  2047
I2C OK, ADC value:  2048
I2C OK, ADC value:  2044
I2C OK, ADC value:  2043
I2C OK, ADC value:  2045
```

```
<terminated> LD3 Debug [STM32 C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Dec 18, 2025, 11:45:38 PM) [pid: 37]
```



```c
MX_I2C1_Init();
MX_ADC1_Init();
/* USER CODE BEGIN 2 */
uint16_t dac_value;
uint16_t value;
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    //half reference voltage
    dac_value = 2048;
    //shift right, get upper portion, extract 4 bits
    tx_buffer[0] = ((dac_value >> 8) & 0x0F);
    tx_buffer[1] = (uint8_t)(dac_value & 0x00FF);
    HAL_StatusTypeDef ret = HAL_I2C_Master_Transmit(&hi2c1, 0x60<<1, tx_buffer, 2, 1000);

    if(ret == HAL_OK){
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, 10);
    value = (uint16_t)HAL_ADC_GetValue(&hadc1) * 3300 / 4095;
    print("I2C OK, ADC value:  %d mV\r\n",value);
    HAL_Delay(100);
    }
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
```

COM4 - Tera Term VT

```
I2C OK, ADC value:  1647 mV
I2C OK, ADC value:  1648 mV
I2C OK, ADC value:  1647 mV
I2C OK, ADC value:  1645 mV
I2C OK, ADC value:  1647 mV
I2C OK, ADC value:  1648 mV
I2C OK, ADC value:  1651 mV
I2C OK, ADC value:  1648 mV
I2C OK, ADC value:  1647 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1648 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1648 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1647 mV
I2C OK, ADC value:  1650 mV
I2C OK, ADC value:  1654 mV
I2C OK, ADC value:  1650 mV
```

```
<terminated> LD3 Debug [STM32 C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Dec 19, 2025, 12:09:08 AM) [pid: 97]
```

4. Sujungimas