VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

ELEKTRONIKOS FAKULTETAS

ELEKTRONINIŲ SISTEMŲ KATEDRA

LABORATORINIS DARBAS 2

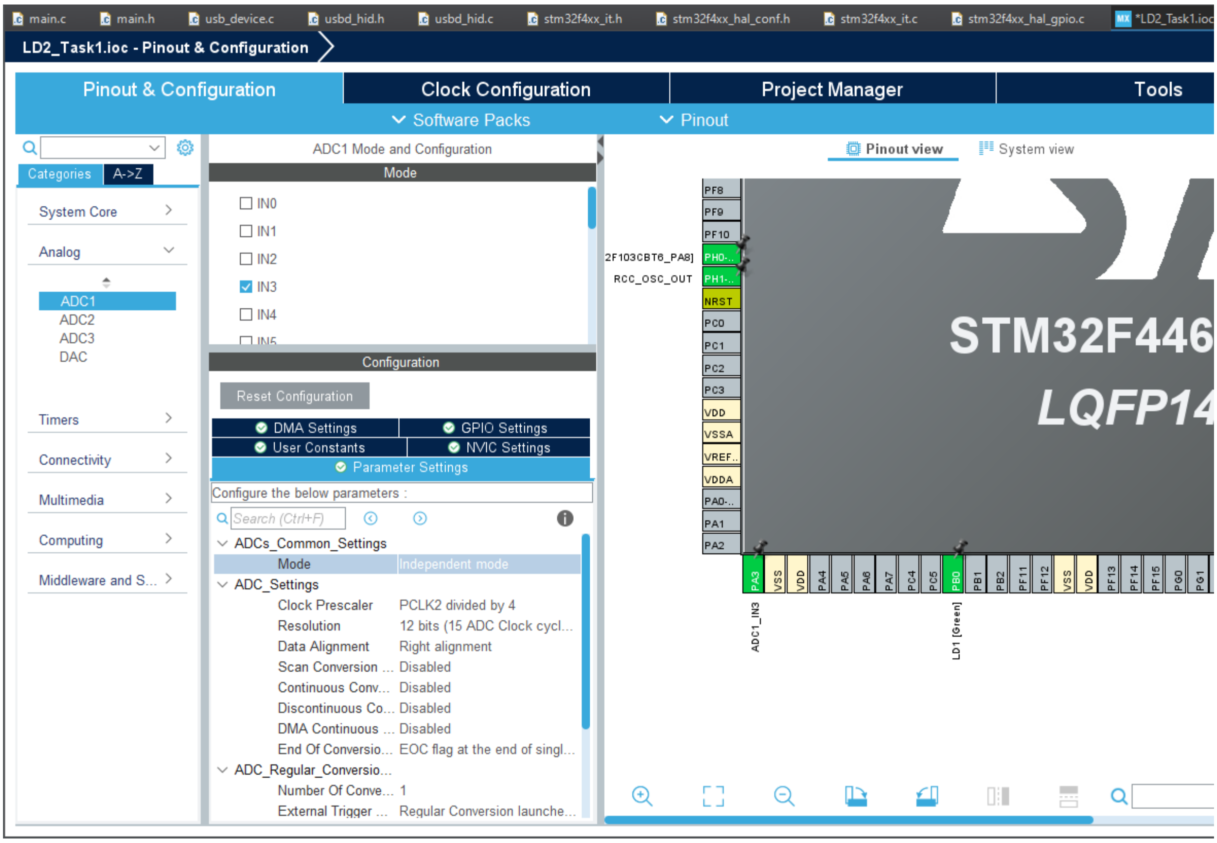
Įterptinių sistemų inžinerija

Atliko: EKSfm-24 gr. Ignas Malinauskas

Tikrino: dr. Eldar Šabanovič

VILNIUS 2025

1. LD_Tas1 projekto konfigūracija



ADC1 Mode and Configuration

Mode

☐ IN0
☐ IN1
☐ IN2
☑ IN3
☐ IN4
☐ IN5

Configuration

Reset Configuration

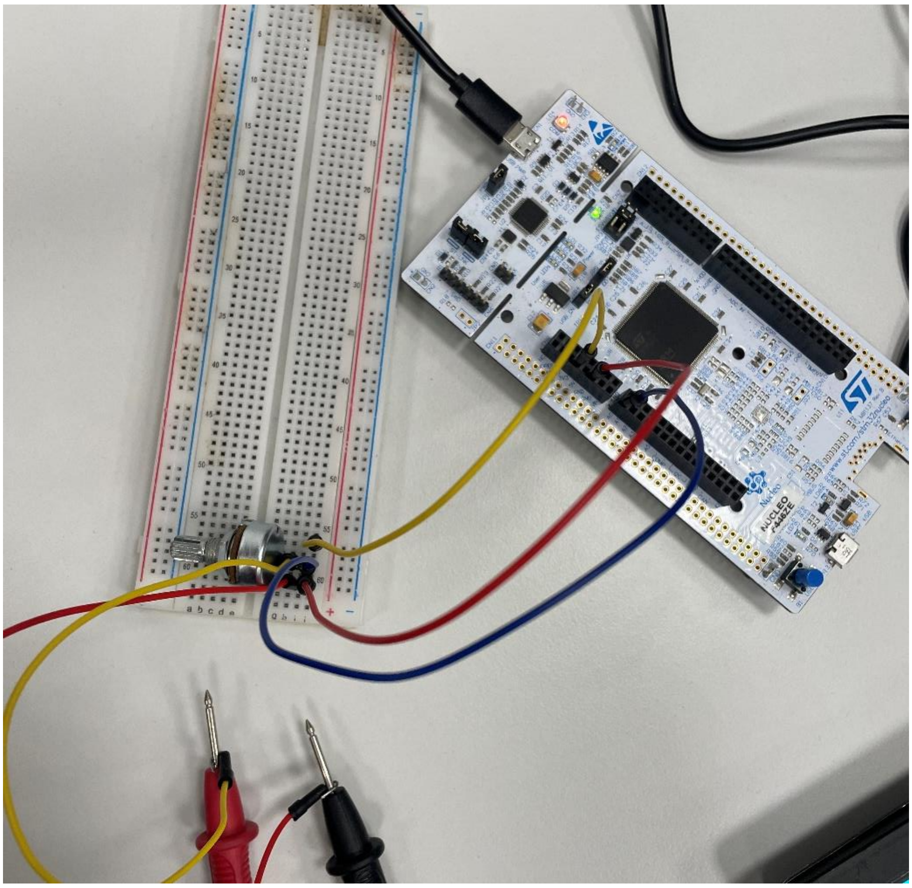
DMA Settings | GPIO Settings | User Constants | NVIC Settings | Parameter Settings

Configure the below parameters :

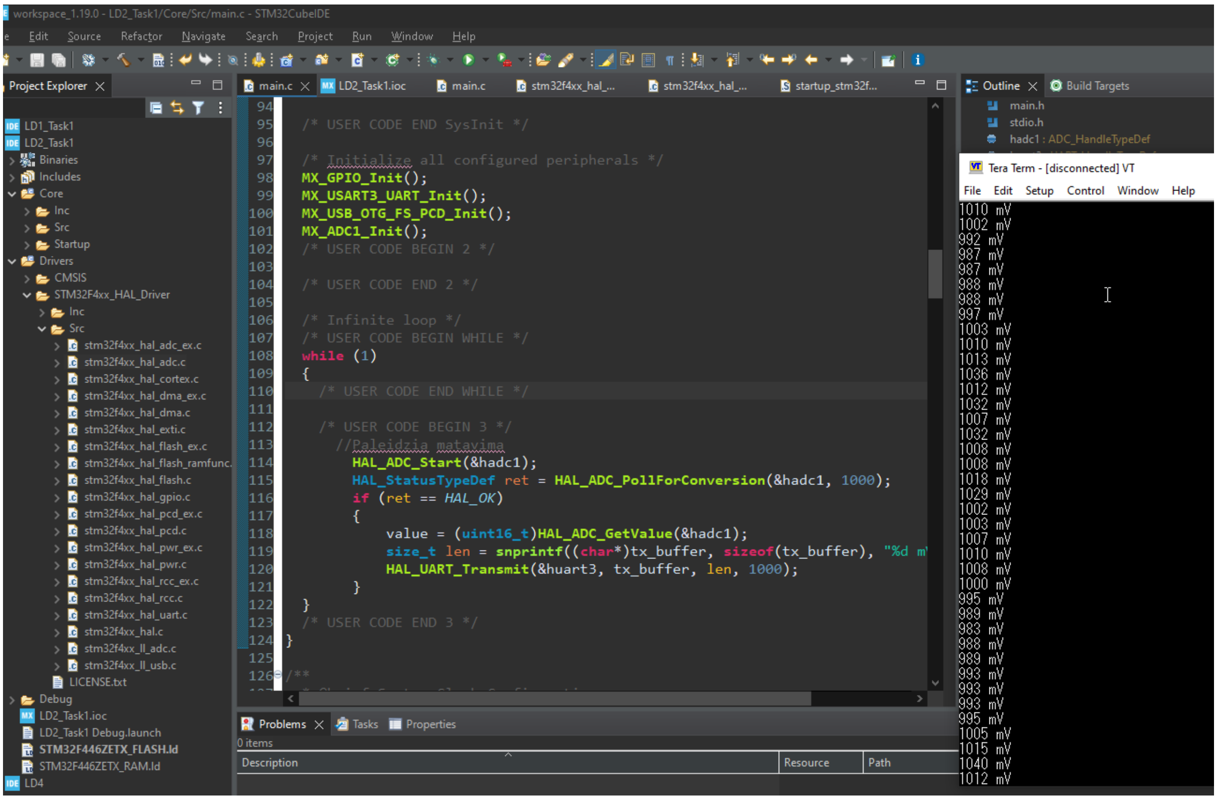
Search (Ctrl+F)

```
ADCs_Common_Settings
    Mode                        Independent mode
ADC_Settings
    Clock Prescaler             PCLK2 divided by 4
    Resolution                  12 bits (15 ADC Clock cycl...
    Data Alignment              Right alignment
    Scan Conversion ...         Disabled
    Continuous Conv...          Disabled
    Discontinuous Co...         Disabled
    DMA Continuous ...          Disabled
    End Of Conversio...         EOC flag at the end of singl...
ADC_Regular_Conversio...
    Number Of Conve...          1
    External Trigger ...        Regular Conversion launche...
```

2. Sujungimas



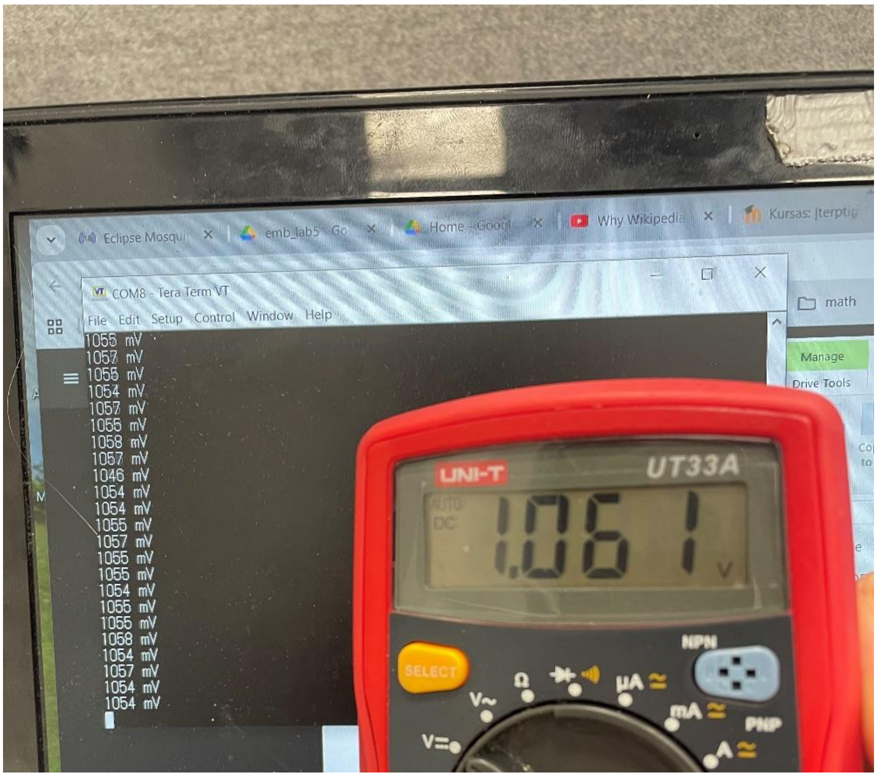
3. ADC (A0) kodo patikrinimas be potenciometro per Tera Term



```c
  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_ADC1_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

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

    /* USER CODE BEGIN 3 */
      //Paleidzia matavima
        HAL_ADC_Start(&hadc1);
        HAL_StatusTypeDef ret = HAL_ADC_PollForConversion(&hadc1, 1000);
        if (ret == HAL_OK)
        {
            value = (uint16_t)HAL_ADC_GetValue(&hadc1);
            size_t len = snprintf((char*)tx_buffer, sizeof(tx_buffer), "%d m
            HAL_UART_Transmit(&huart3, tx_buffer, len, 1000);
        }
  }
  /* USER CODE END 3 */
}

/**
```

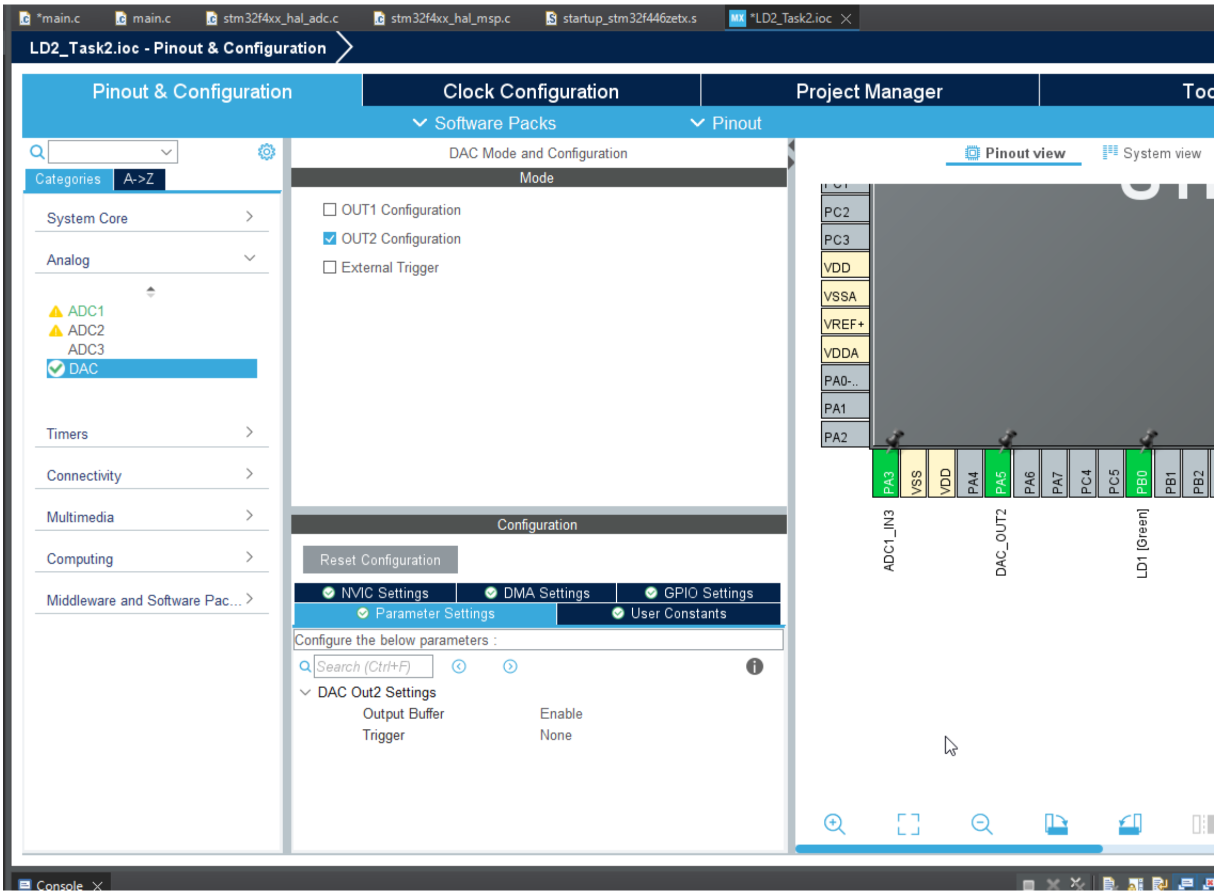
```
1010 mV
1002 mV
992 mV
987 mV
987 mV
988 mV
988 mV
997 mV
1003 mV
1010 mV
1013 mV
1036 mV
1012 mV
1032 mV
1007 mV
1032 mV
1008 mV
1008 mV
1018 mV
1029 mV
1002 mV
1003 mV
1007 mV
1010 mV
1008 mV
1000 mV
995 mV
989 mV
983 mV
988 mV
989 mV
993 mV
993 mV
993 mV
995 mV
1005 mV
1015 mV
1040 mV
1012 mV
```

4. ADC nuskaitytos vertės iš potenciometro parodomos per UART ir multimetrą



```
1055 mV
1057 mV
1055 mV
1054 mV
1057 mV
1055 mV
1058 mV
1057 mV
1046 mV
1054 mV
1054 mV
1055 mV
1057 mV
1055 mV
1055 mV
1054 mV
1055 mV
1055 mV
1058 mV
1054 mV
1057 mV
1054 mV
1054 mV
```

5. LD2_Task2 projekto konfigūracija



DAC Mode and Configuration

Mode

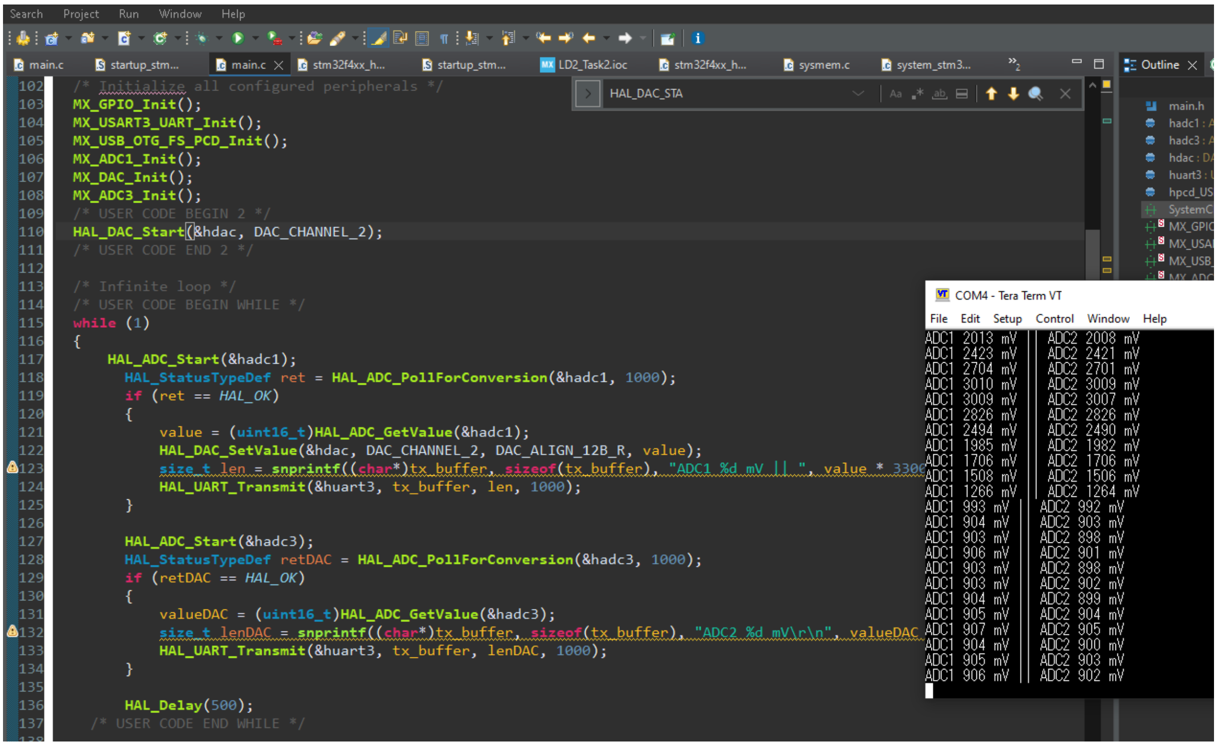
☐ OUT1 Configuration
☑ OUT2 Configuration
☐ External Trigger

Configuration

Configure the below parameters :

DAC Out2 Settings
Output Buffer Enable
Trigger None

6. DAC (D13 (PA_5)) išvesties kodas ir patikrinimas su ADC (A3(PF3)

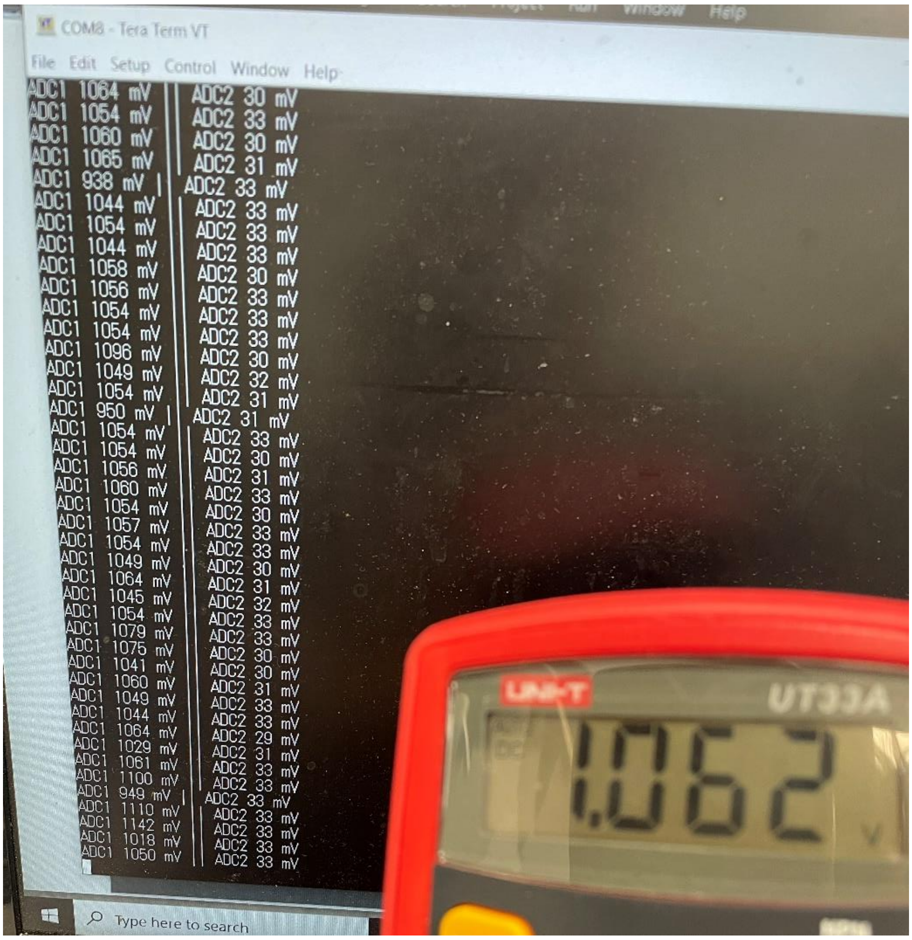


```c
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_ADC1_Init();
  MX_DAC_Init();
  MX_ADC3_Init();
  /* USER CODE BEGIN 2 */
  HAL_DAC_Start(&hdac, DAC_CHANNEL_2);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
      HAL_ADC_Start(&hadc1);
        HAL_StatusTypeDef ret = HAL_ADC_PollForConversion(&hadc1, 1000);
        if (ret == HAL_OK)
        {
            value = (uint16_t)HAL_ADC_GetValue(&hadc1);
            HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B_R, value);
            size_t len = snprintf((char*)tx_buffer, sizeof(tx_buffer), "ADC1 %d mV || ", value * 3300
            HAL_UART_Transmit(&huart3, tx_buffer, len, 1000);
        }

        HAL_ADC_Start(&hadc3);
        HAL_StatusTypeDef retDAC = HAL_ADC_PollForConversion(&hadc3, 1000);
        if (retDAC == HAL_OK)
        {
            valueDAC = (uint16_t)HAL_ADC_GetValue(&hadc3);
            size_t lenDAC = snprintf((char*)tx_buffer, sizeof(tx_buffer), "ADC2 %d mV\r\n", valueDAC
            HAL_UART_Transmit(&huart3, tx_buffer, lenDAC, 1000);
        }

        HAL_Delay(500);
    /* USER CODE END WHILE */
```

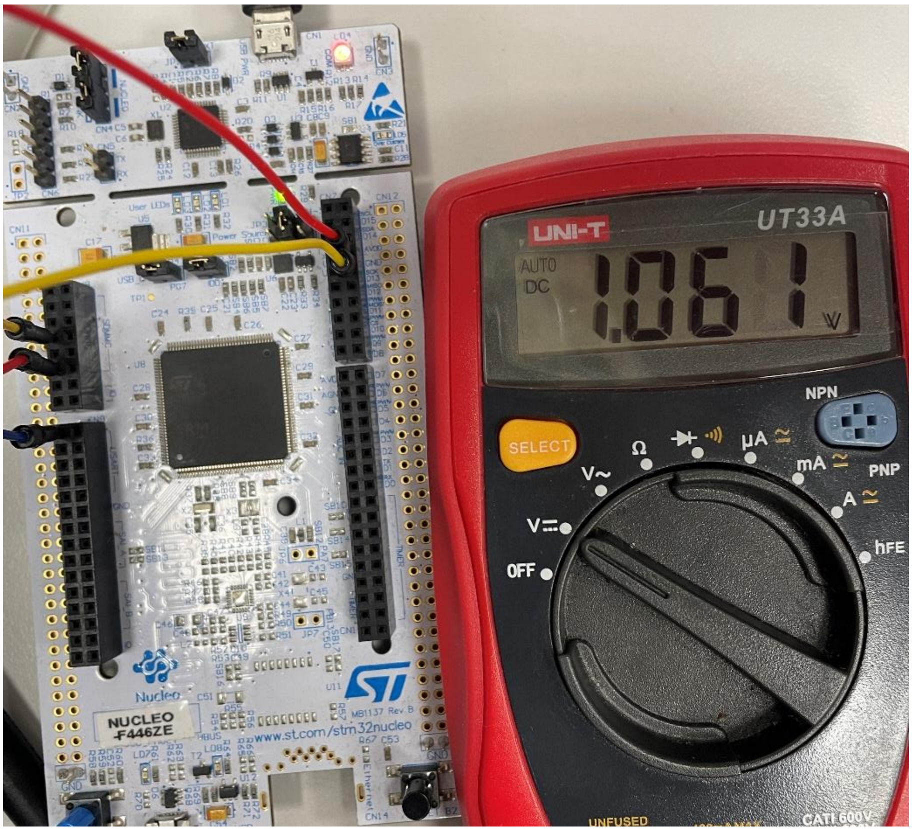
COM4 - Tera Term VT

```
ADC1 2013 mV || ADC2 2008 mV
ADC1 2423 mV || ADC2 2421 mV
ADC1 2704 mV || ADC2 2701 mV
ADC1 3010 mV || ADC2 3009 mV
ADC1 3009 mV || ADC2 3007 mV
ADC1 2826 mV || ADC2 2826 mV
ADC1 2494 mV || ADC2 2490 mV
ADC1 1985 mV || ADC2 1982 mV
ADC1 1706 mV || ADC2 1706 mV
ADC1 1508 mV || ADC2 1506 mV
ADC1 1266 mV || ADC2 1264 mV
ADC1 993 mV || ADC2 992 mV
ADC1 904 mV || ADC2 903 mV
ADC1 903 mV || ADC2 898 mV
ADC1 906 mV || ADC2 901 mV
ADC1 903 mV || ADC2 898 mV
ADC1 903 mV || ADC2 902 mV
ADC1 904 mV || ADC2 899 mV
ADC1 905 mV || ADC2 904 mV
ADC1 907 mV || ADC2 905 mV
ADC1 904 mV || ADC2 900 mV
ADC1 905 mV || ADC2 903 mV
ADC1 906 mV || ADC2 902 mV
```

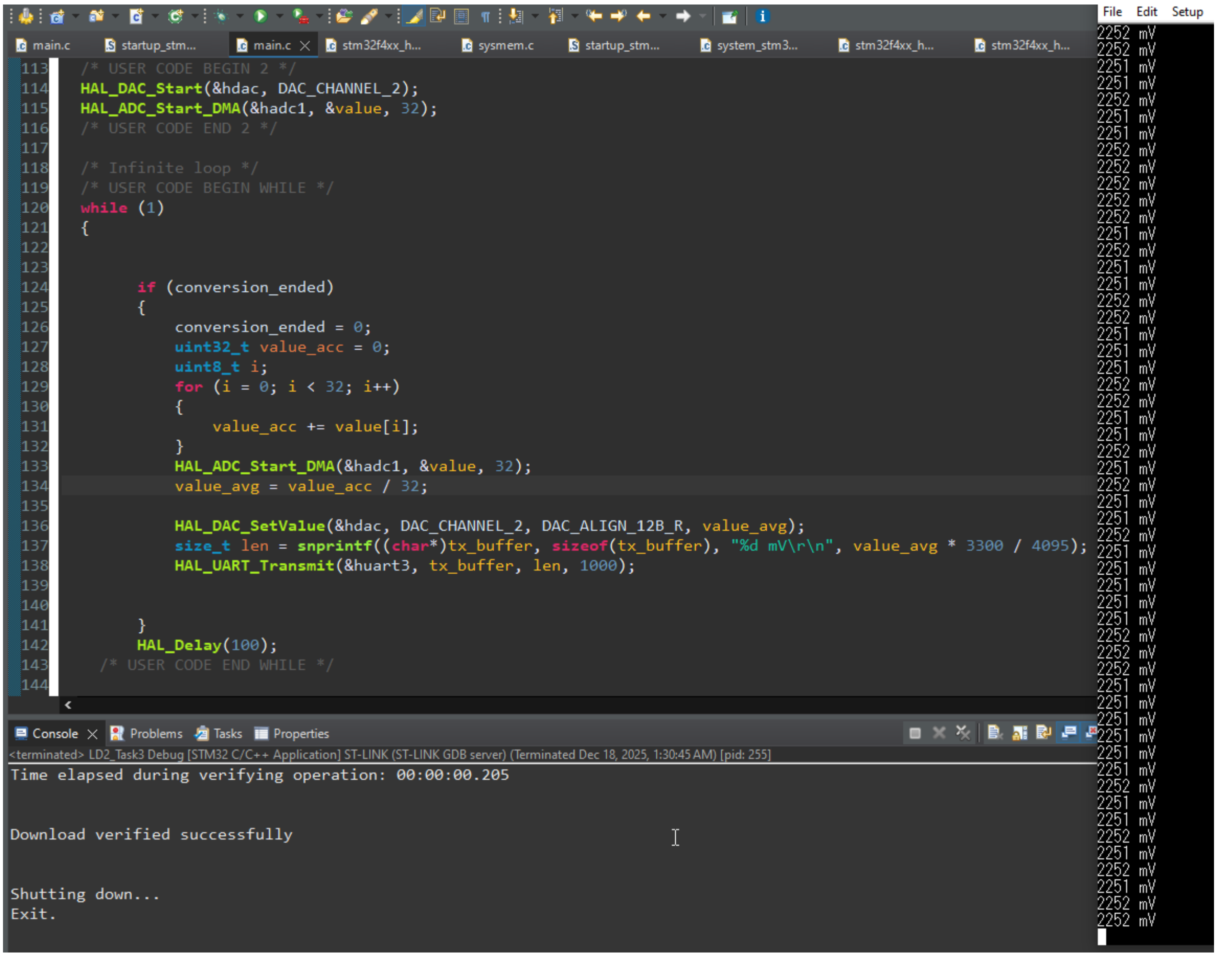
7. DAC reikšmės patikrinimas multimetru (įtampa tokia kaip 4 punkte)



```
ADC1 1064 mV || ADC2 30 mV
ADC1 1054 mV || ADC2 33 mV
ADC1 1060 mV || ADC2 30 mV
ADC1 1065 mV || ADC2 31 mV
ADC1 938 mV || ADC2 33 mV
ADC1 1044 mV || ADC2 33 mV
ADC1 1054 mV || ADC2 33 mV
ADC1 1044 mV || ADC2 33 mV
ADC1 1058 mV || ADC2 30 mV
ADC1 1056 mV || ADC2 33 mV
ADC1 1054 mV || ADC2 33 mV
ADC1 1054 mV || ADC2 33 mV
ADC1 1096 mV || ADC2 30 mV
ADC1 1049 mV || ADC2 32 mV
ADC1 1054 mV || ADC2 31 mV
ADC1 950 mV || ADC2 31 mV
ADC1 1054 mV || ADC2 33 mV
ADC1 1054 mV || ADC2 30 mV
ADC1 1056 mV || ADC2 31 mV
ADC1 1060 mV || ADC2 33 mV
ADC1 1054 mV || ADC2 30 mV
ADC1 1057 mV || ADC2 33 mV
ADC1 1054 mV || ADC2 33 mV
ADC1 1049 mV || ADC2 30 mV
ADC1 1064 mV || ADC2 31 mV
ADC1 1045 mV || ADC2 32 mV
ADC1 1054 mV || ADC2 33 mV
ADC1 1079 mV || ADC2 33 mV
ADC1 1075 mV || ADC2 30 mV
ADC1 1041 mV || ADC2 30 mV
ADC1 1060 mV || ADC2 31 mV
ADC1 1049 mV || ADC2 33 mV
ADC1 1044 mV || ADC2 33 mV
ADC1 1064 mV || ADC2 29 mV
ADC1 1029 mV || ADC2 31 mV
ADC1 1061 mV || ADC2 33 mV
ADC1 1100 mV || ADC2 33 mV
ADC1 949 mV || ADC2 33 mV
ADC1 1110 mV || ADC2 33 mV
ADC1 1142 mV || ADC2 33 mV
ADC1 1018 mV || ADC2 33 mV
ADC1 1050 mV || ADC2 33 mV
```



8.) ADC DMA su pertrauktim kodas ir rezultatai per UART

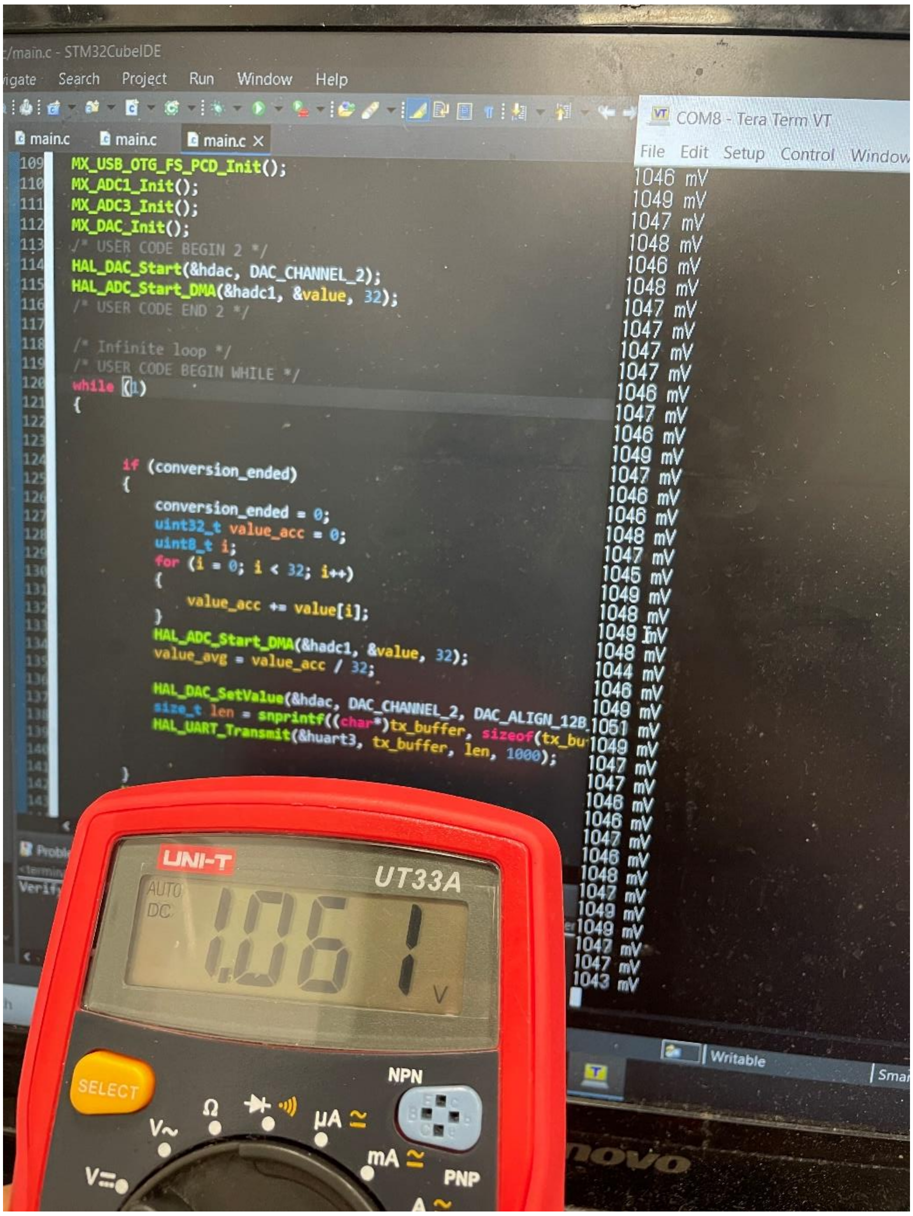


```c
/* USER CODE BEGIN 2 */
HAL_DAC_Start(&hdac, DAC_CHANNEL_2);
HAL_ADC_Start_DMA(&hadc1, &value, 32);
/* USER CODE END 2 */

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


      if (conversion_ended)
      {
          conversion_ended = 0;
          uint32_t value_acc = 0;
          uint8_t i;
          for (i = 0; i < 32; i++)
          {
              value_acc += value[i];
          }
          HAL_ADC_Start_DMA(&hadc1, &value, 32);
          value_avg = value_acc / 32;

          HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B_R, value_avg);
          size_t len = snprintf((char*)tx_buffer, sizeof(tx_buffer), "%d mV\r\n", value_avg * 3300 / 4095);
          HAL_UART_Transmit(&huart3, tx_buffer, len, 1000);


      }
      HAL_Delay(100);
  /* USER CODE END WHILE */

```

```
<terminated> LD2_Task3 Debug [STM32 C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Dec 18, 2025, 1:30:45 AM) [pid: 255]
Time elapsed during verifying operation: 00:00:00.205


Download verified successfully


Shutting down...
Exit.
```

```
2252 mV
2252 mV
2251 mV
2251 mV
2252 mV
2251 mV
2251 mV
2252 mV
2252 mV
2252 mV
2252 mV
2252 mV
2251 mV
2252 mV
2251 mV
2251 mV
2252 mV
2252 mV
2251 mV
2251 mV
2251 mV
2252 mV
2252 mV
2251 mV
2251 mV
2252 mV
2251 mV
2252 mV
2251 mV
2251 mV
2252 mV
2251 mV
2251 mV
2251 mV
2251 mV
2251 mV
2252 mV
2252 mV
2252 mV
2251 mV
2251 mV
2251 mV
2251 mV
2251 mV
2251 mV
2252 mV
2251 mV
2251 mV
2252 mV
2251 mV
2252 mV
2251 mV
2252 mV
2252 mV
```

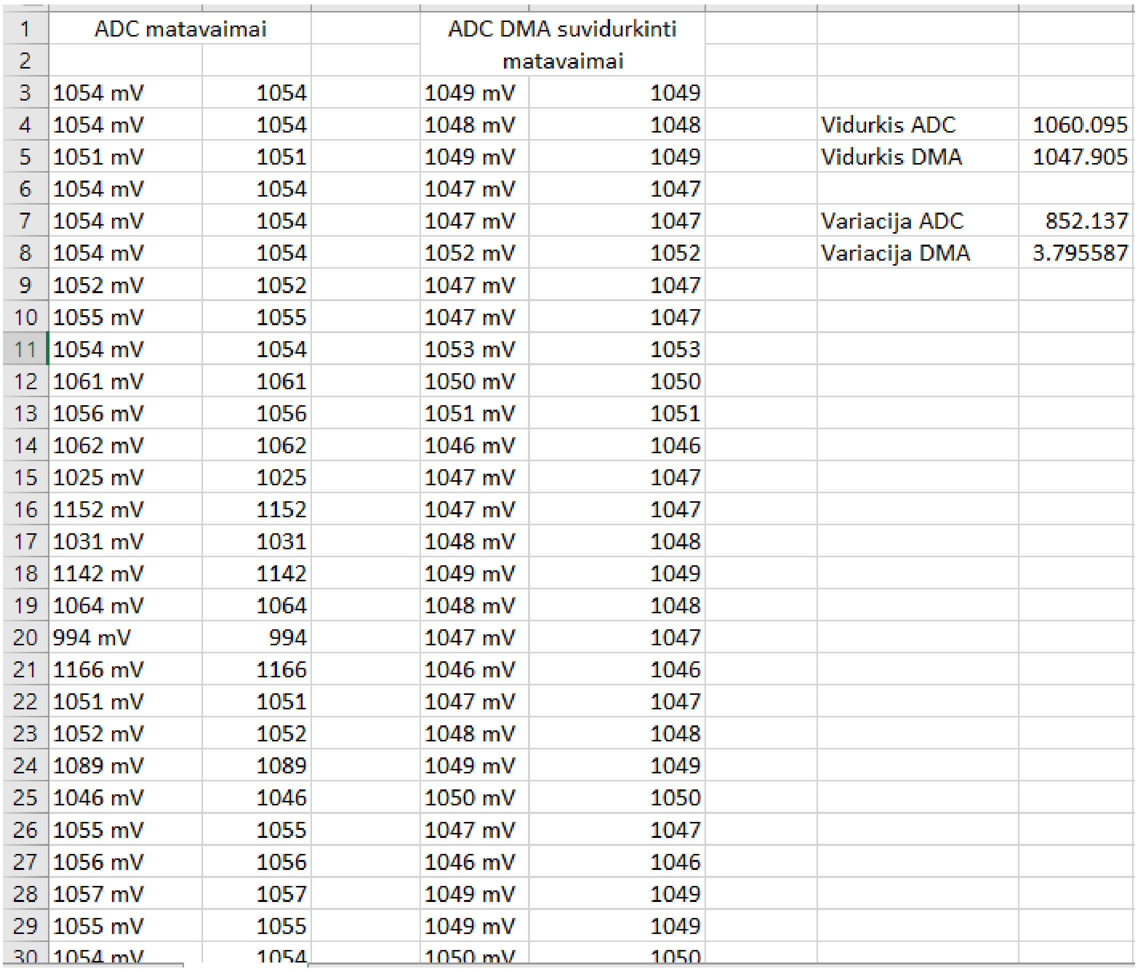
9. ADC DMA rezultatų tikrinimas multimetru (pagal 4 punkto įtampą)



```
MX_USB_OTG_FS_PCD_Init();
MX_ADC1_Init();
MX_ADC3_Init();
MX_DAC_Init();
/* USER CODE BEGIN 2 */
HAL_DAC_Start(&hdac, DAC_CHANNEL_2);
HAL_ADC_Start_DMA(&hadc1, &value, 32);
/* USER CODE END 2 */

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


    if (conversion_ended)
    {
        conversion_ended = 0;
        uint32_t value_acc = 0;
        uint8_t i;
        for (i = 0; i < 32; i++)
        {
            value_acc += value[i];
        }
        HAL_ADC_Start_DMA(&hadc1, &value, 32);
        value_avg = value_acc / 32;

        HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B
        size_t len = snprintf((char*)tx_buffer, sizeof(tx_bu
        HAL_UART_Transmit(&huart3, tx_buffer, len, 1000);

    }
```

```
1046 mV
1049 mV
1047 mV
1048 mV
1046 mV
1048 mV
1047 mV
1047 mV
1047 mV
1047 mV
1046 mV
1047 mV
1046 mV
1049 mV
1047 mV
1046 mV
1046 mV
1048 mV
1047 mV
1045 mV
1049 mV
1048 mV
1049 mV
1048 mV
1044 mV
1046 mV
1049 mV
1051 mV
1049 mV
1047 mV
1047 mV
1048 mV
1046 mV
1047 mV
1048 mV
1048 mV
1047 mV
1049 mV
1049 mV
1047 mV
1047 mV
1043 mV
```

10. Skaičiavimai nekeičiant potenciometro pozicijos



1	ADC matavaimai			ADC DMA suvidurkinti matavaimai				
2								
3	1054 mV	1054		1049 mV	1049			
4	1054 mV	1054		1048 mV	1048		Vidurkis ADC	1060.095
5	1051 mV	1051		1049 mV	1049		Vidurkis DMA	1047.905
6	1054 mV	1054		1047 mV	1047			
7	1054 mV	1054		1047 mV	1047		Variacija ADC	852.137
8	1054 mV	1054		1052 mV	1052		Variacija DMA	3.795587
9	1052 mV	1052		1047 mV	1047			
10	1055 mV	1055		1047 mV	1047			
11	1054 mV	1054		1053 mV	1053			
12	1061 mV	1061		1050 mV	1050			
13	1056 mV	1056		1051 mV	1051			
14	1062 mV	1062		1046 mV	1046			
15	1025 mV	1025		1047 mV	1047			
16	1152 mV	1152		1047 mV	1047			
17	1031 mV	1031		1048 mV	1048			
18	1142 mV	1142		1049 mV	1049			
19	1064 mV	1064		1048 mV	1048			
20	994 mV	994		1047 mV	1047			
21	1166 mV	1166		1046 mV	1046			
22	1051 mV	1051		1047 mV	1047			
23	1052 mV	1052		1048 mV	1048			
24	1089 mV	1089		1049 mV	1049			
25	1046 mV	1046		1050 mV	1050			
26	1055 mV	1055		1047 mV	1047			
27	1056 mV	1056		1046 mV	1046			
28	1057 mV	1057		1049 mV	1049			
29	1055 mV	1055		1049 mV	1049			
30	1054 mV	1054		1050 mV	1050			

DMA apskaičiuotas reikšmės turi mažiau triukšmo ir mažiau svyruoja aplink vidurkį, tačiau ADC nesuvidurkintos reiškmės yra arčiau tikrosios paduodamos įtampos (1.061V). Tai galėtų būti dėl per greito ADC DMA diskretizavimo laiko.