

Analogno-digitalna konverzija



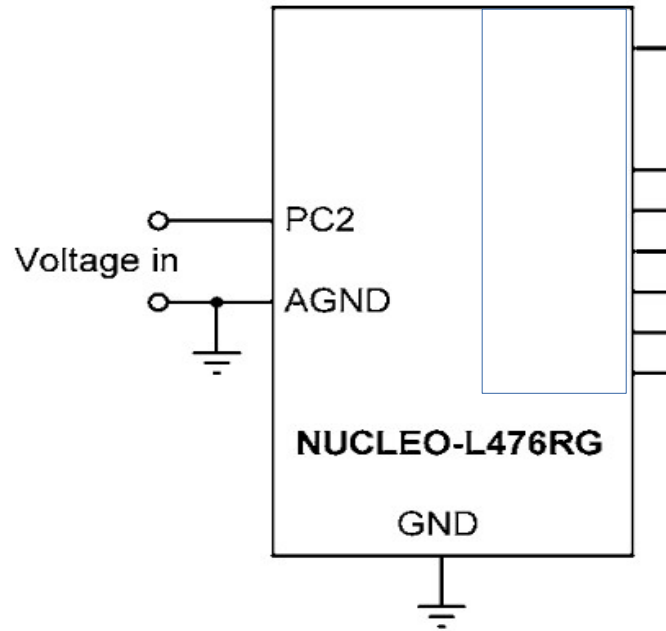
Analogno-digitalna konverzija



Povezati promenljivi otpornik na proto ploču!
Uzeti 3.3 V sa Nucleo ploče!
Unimerom pratiti promene napona na klizacu!

Single channel ADC

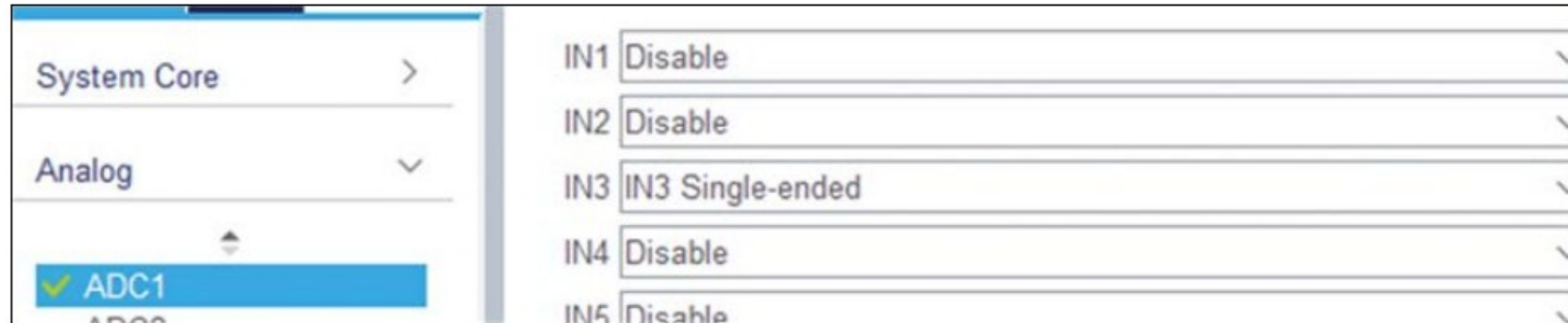
U projektu koristiti User Led
ili neku eksternu led!



Single channel ADC

Klik na karticu Analog i izaberi ADC1, a zatim uključi kanal IN3 kao Single-ended da bi se omogućio ADC!

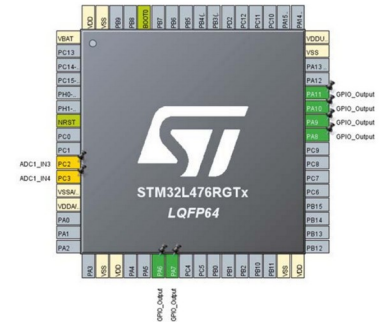
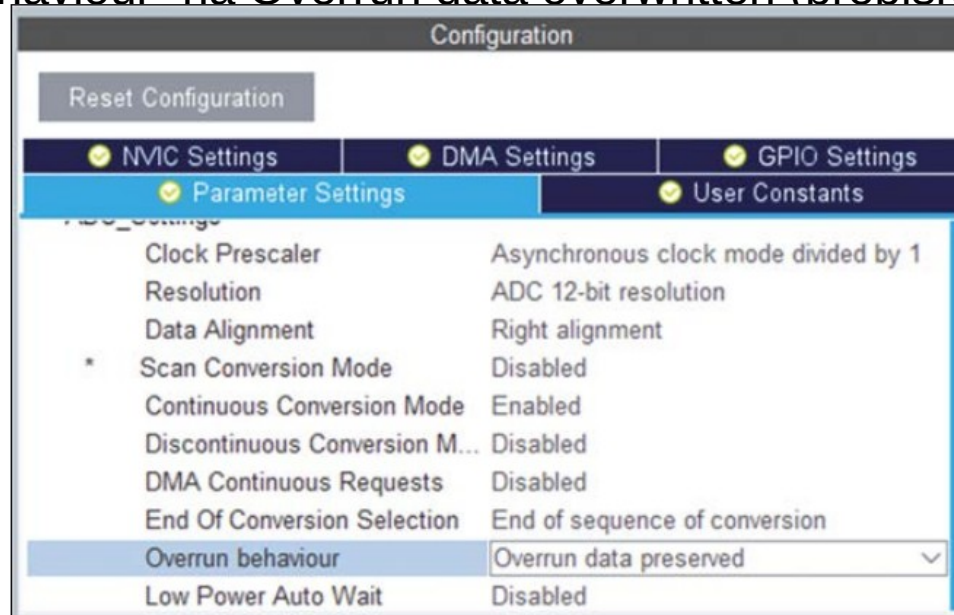
Proveriti pin na kome je ADC1!



Single channel ADC

U okviru “Parameter Settings”, klik na

- Resolution i izaberi ADC 12-bitnu rezoluciju.
- continuous conversion mode (režim neprekidnog konvertovanja).
- Podesiti “End of Conversion Selection” na End of sequence of conversions (kraj sekvence konverzija).
- Podesiti “Overrun behaviour” na Overrun data overwritten (prepisivanje podataka pri prekoračenju).



Single channel ADC

Klik na Clock Configuration i provera da su sistemski takt i ADC takt podešeni kako je potrebno.

- Generisti Code!
- Klik na Core, zatim Src, i dupli klik na main.c da se otvori glavni program..

```
#include «main.h»
#include «LCD.h»
#include «stdio.h»
uint32_t adcResult = 0;
ADC_HandleTypeDef hadc1;
.....
.....
.....
```

Pogledati:
static void MX_ADC1_Init(void)

```
.....
.....
.....
HAL_ADC_Start(&hadc1);
while (1)
{
    HAL_ADC_PollForConversion(&hadc1, 100);
    adcResult = HAL_ADC_GetValue(&hadc1);
    HAL_Delay(1000);
}
```

Single channel ADC

Okretati pazljivo klizac promenljivog otpornika I u Debug mode pratiti promenu promenljive adcResult!

```
#include «main.h»
#include «LCD.h»
#include «stdio.h»
uint32_t adcResult = 0;
ADC_HandleTypeDef hadc1;
.....
.....
.....
.....
.....
HAL_ADC_Start(&hadc1);
while (1)
{
    HAL_ADC_PollForConversion(&hadc1, 100);
    adcResult = HAL_ADC_GetValue(&hadc1);
    HAL_Delay(1000);
}
```

Single channel ADC

Okretati pazljivo klizac promenljivog otpornika!

Ako je vrednost adcResult iznad zadatog praga, ukljuciti user led ili neku drugu led po izboru!

```
.....  
.....  
.....  
HAL_ADC_Start(&hadc1);  
while (1)  
{  
    HAL_ADC_PollForConversion(&hadc1, 100);  
    adcResult = HAL_ADC_GetValue(&hadc1);  
    //USER zadatak  
    //ako je vrednost konverzije iznad zadatog praga  
    //ukljuci user LED!  
    HAL_Delay(1000);  
}
```

Single channel ADC

Sledeće HAL funkcije se koriste u ovom programu:

HAL_Init(): ova funkcija inicijalizuje ADC.

HAL_ADC_Start(&hadc1): ova funkcija pokreće ADC konverziju.

HAL_ADC_PollForConversion(&hadc1, 100): ova funkcija čeka da se ADC konverzija završi. Broj 100 u ovom primeru predstavlja timeout u milisekundama.

adcResult = HAL_ADC_GetValue(&hadc1): ova funkcija čita konvertovane ADC podatke u promenljivu adcResult.

Single channel ADC

Neke druge korisne HAL ADC funkcije su:

HAL_ADC_Stop(handle): ova funkcija zaustavlja ADC konverziju.

HAL_ADC_Start_IT(handle): ova funkcija pokreće ADC konverziju regularne grupe sa prekidima (interruption).

HAL_ADC_Stop_IT(handle): ova funkcija zaustavlja ADC konverziju kada su prekidi omogućeni.

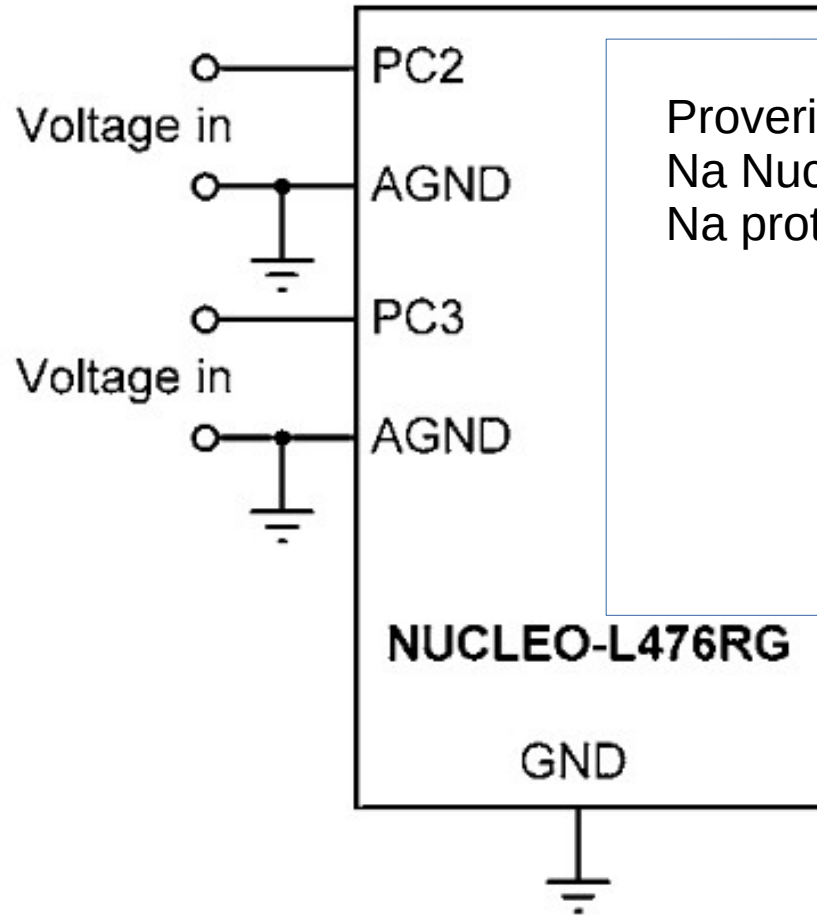
HAL_ADC_Start_DMA(handle, data, length): ova funkcija omogućava ADC, pokreće konverziju regularne grupe i prenosi rezultat putem DMA.

HAL_ADC_Stop_DMA(handle): ova funkcija zaustavlja ADC konverziju regularne grupe i onemogućava DMA prenose ADC-a.

HAL_ADC_IRQHandler(handle): ova funkcija obrađuje ADC interrupt zahteve.

HAL_ADC_ConvCpltCallback(handle): ovo je callback funkcija koja se poziva kada konverzija bude završena u non-blocking modu.

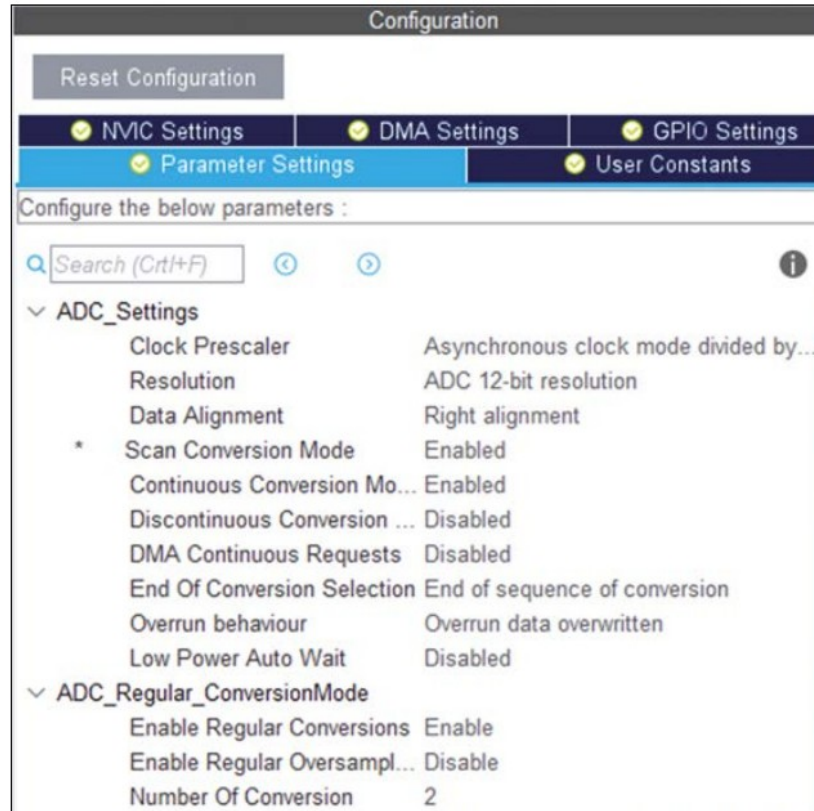
Multiple Inputs (polling ADC)



Proveriti ulaze ADC1 I ADC2
Na Nucleo ploci na kojoj se radi projekat!
Na proto ploci dodati drugi prom. otpornik!

Multiple Inputs (polling ADC)

- Set **Number of Conversions** to **2**.
- Set **Rank 1 Channel** to **Channel 3**. Also, set **Rank 2 Channel** to **Channel 4** (Figure 8.9).



Multiple Inputs (polling ADC)

U glavnom programu dodati:

```
uint32_t adcResult1, adcResult2;
```

Pogledati:

```
static void MX_ADC1_Init(void)
```

```
while (1)
```

```
{
```

```
    HAL_ADC_Start(&hadc1); // Start ADC
```

```
    HAL_ADC_PollForConversion(&hadc1, 100); // Poll ADC
```

```
    adcResult1 = HAL_ADC_GetValue(&hadc1); // Get the result
```

```
    HAL_ADC_PollForConversion(&hadc1, 100); // Poll the ADC
```

```
    adcResult2 = HAL_ADC_GetValue(&hadc1); // Get next result
```

```
    HAL_ADC_Stop(&hadc1);
```