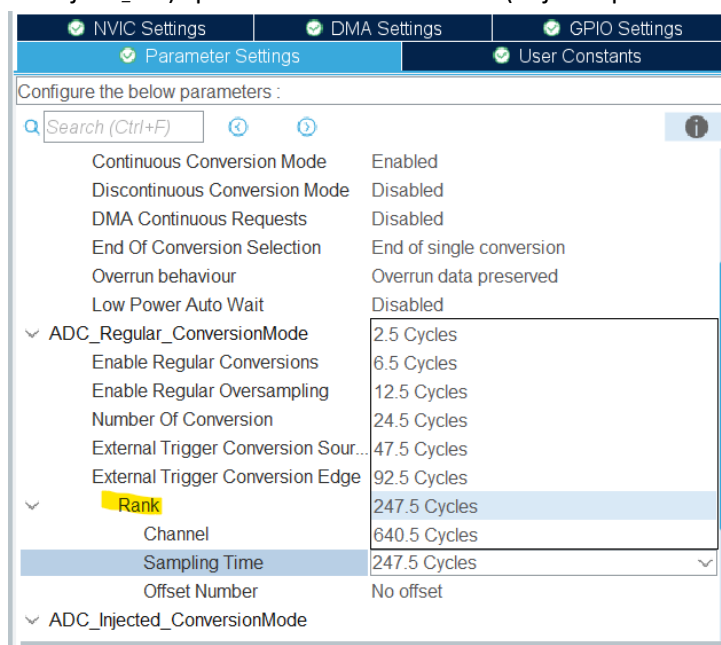


Vaja 2 – kontinuirana ADC pretvorba z Nucleo

- Cilj naloge:** S pomočjo programskega okolja **STM32CubeIDE** in HAL knjižnicami sprogramirajte mikroprocesor tako, da bo izvajal neprekinjene ADC pretvorbe z izbranim potenciometrom. Takšna pretvorba je primerna za hitro in nenehno branje vhodne vrednosti.
- Postopek inicializacije periferije.**
 - Zaženite **STM32CubeIDE** in ustvarite nov STM32 projekt (pod zavihkom *information Center*). V zavihku *Board selector* s pomočjo filtrov, Type in MCU/MPU Series Izberite ustrezno razvojno ploščo (v našem primeru Nucleo), kliknite *Next*, projekt poimenujte **vaja2_ADC_pretvorba_Sk_X** in kliknite *Next*, v oknu za knjižnice izberite *Add necessary library files as reference ..* ter gumb *Finish* (na možnosti opcije za inicializacije periferije izberite *Yes*, izbrana naj bo tudi opcija perspektive za STM32CubeMX).
 - V *Analog* razdelku levo od sheme mikroprocesorja kliknite *Analog*. Koliko je vseh ADC pretvornikov?
3
 - Izberite en ADC pretvornik in v njem izberite (prosti) kanal, kjer boste zajemali analogno (*single-ended*) meritev. Na oknu *Pinout View* se določen pin pobarva v zeleno – zapišite naslov dodeljenega pin-a PC0.
 - Kaj se izpiše poleg pina? ADC1_IN1. Potenciometer pravilno priključite na Nucleo razvojno ploščo preko *protoborda* (potenciometer ima 3 priključke: GND, out in +3,3 V).
 - V *Clock Configuration* spremenimo *APB1 peripheral clock (MHz)* na **16 MHz**. (potrdite z ENTER) Kaj opazite? Prescaler se prestavi na /4.
 - V *Configuration* kliknemo *ADC* gumb. V *Parameter settings* izberite ločljivost pretvorbe na **8-bitno**, torej je območje vrednosti od $0 \div 255$.
 - Clock Prescaler* nastavimo z deliteljem **4**. Kolikšna je sedaj preskalirana frekvenca takta $f_{preskalirana}$?
4 MHz
 - Omogočimo *Continuous Conversion Mode*.
 - Sampling time* (čas vzorčenja t_{vz_ciklih}) spremenite na **247.5** ciklov (najdete pod *Rank*).



Pravi čas vzorčenja se nato **poveča** še za **12** ciklov zaradi procesa samega. Koliko znaša pravi čas vzorčenja t_{vz} v **mikro sekundah**?

(enačba: $t_{vz} = t'_{vz_ciklih} / f_{preskalirana}$)? 64,875 us

- Sedaj generirajte kodo tako, da enostavno kliknete ikono *Save* in po potrebi še enkrat potrdimo generiranje kode.

Vaja 2 – kontinuirana ADC pretvorba z Nucleo

3. Programiranje v IDE:

a) V skrajno levem oknu *Project Explorer* poiščemo **main.c** datoteko pod *Core* → *Src* → *main.c* (dvokliknite na datoteko, odpre se tekstovni urejevalnik za main.c).

b) V *User code begin 0* deklariramo našo ADC spremenljivko kot pozitivno 32-bitno:

```
uint32_t adcVal;
```

c) V *User code begin 2* nastavimo ADC pretvornik z ukazom:

```
HAL_ADC_Start(&ADC_HANDLE_TYPE);
```

Ime tega parametra boste našli v vrstici pod *Private variables* ----. Del zgornje kode v rdečem torej zamenjajte z imenom te ročice (ADC_HandleTypeDef **XXXX**);).

d) V neskočni zanki programa *While* (1) zapišemo ukaz za zajem vrednosti:

```
adcVal = HAL_ADC_GetValue(&XXXX);
```

Izmerjena vrednost se bo shranila v spremenljivko *adcVal*.

4. Naložitev kode (Run)

a) Kodo preverite s tipko **Build** (ikona za kladivce - *Debug*). Ko je preverjanje končano lahko preverimo, če smo med pisanjem kode naredili kakšno napako sintakse, sicer se pod kodo v oknu *Console* izpiše 0 errors.

b) Priklopite STM32 Nucleo na vaš računalnik preko USB kabla.

c) S tipko **Run** (zelena ikona za play – *Run as STM...*) prenesete program na STM32 Nucleo.

d) V oknu *Edit Configuration* kliknemo OK. Nekaj sekund bo na ploščici STM izmenično utripala zelena in rdeča LED, ko je program naložen, sveti LED rdeče.

5. Branje vrednosti spremenljivke znotraj IDE (debug mode)

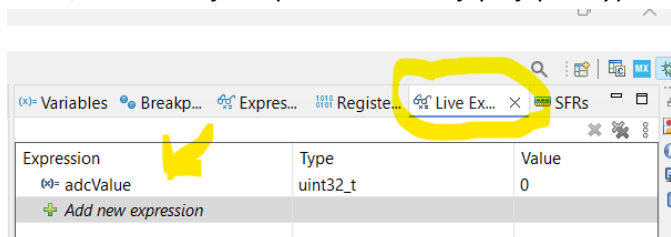
a) Kliknemo puščico poleg *Debug* ikone in izberemo iz spustnega seznama *Debug Configurations* ..

b) V zavihu *Debugger* omogočimo *Serial Wire Viewer (SWV)*, frekvenco jedra pustimo privzeto.

c) Še nižje preverimo, da je *Enable live expressions* omogočen.

d) Kliknemo na gumb *Apply*, nato še *Debug*. Sedaj se okolje spremeni v razhroščevalni (debug) način.

e) V iskalniku (prazno belo polje desno zgoraj) vpišemo **Live expressions (Debug)**, izberemo najdeno opcijo ter v tabelo pod *Expression* vpišemo ime naše spremenljivke *adcValue* in pritisnemo tabulator. V kolikor je zapis spremenljivke pravičen, se samodejno izpolnita naslednji polji pod *Type* in *Value*.



f) Sedaj lahko kliknemo na ikono *Resume* (lahko pritisnemo tudi tipko F8). V tabeli *Live Expressions* bi morali videti vrednost spremenljivke v realnem času.

g) Za ustavitev branja kliknemo gumb *Terminate* (ali CTRL + F2) in se tako vrnemo v pogledu programiranja.

h) Delovanje posnamite s telefonom.

6. Branje vrednosti s programom STM Studio (v kolikor debug mode ne deluje!)

a) Zaženite program STM Studio.

b) Kliknite na *Import variables from executable*. Kliknemo ... in poiščemo našo **.elf** datoteko (nahaja se v mapi vašega projekta).

c) V nastali tabeli izberite vaše spremenljivke *adcVal* in kliknite *Import*. Kliknite *Close*.

Vaja 2 – kontinuirana ADC pretvorba z Nucleo

- d) Na levi strani (okno *Display Variables*) desni klik na *vlotages* in jo pošljite na *VarViewer1*. V *Viewer Settings* namesto *Bar Graph* izberemo *Table*. Kliknite ikono za RUN (ne pozabite resetirati STM!).
- e) Sedaj lahko spreminjate pozicijo potenciometra in opazujete vrednost spremenljivke *adcValue*. Delovanje posnamite s telefonom.
7. **Vaš projekt** (datoteko *main.c* [15%], slikovni izrezek *Pinout* mikroporcesorja iz CubeMX [15%], kratek videposnetek delovanja [15%], fotografija vezave potenciometra na Nucleo ploščici [10%]) **naložite v Github** kot nov *Repository* (public!) z imenom **Vaja2-ADC-continuos-conversion-Nucleo** [5%]. V *readme* datoteki zapišite vse **odgovore** [30%] na vprašanja ter **komentar** [10%] na delovanje. **Oceno** pridobite iz Github dokumentacije!
8. Dodatna video pomoč z razlago: [Video Tutorial 4](#).
9. Pinout NUCLEO-L476RG:

