# **Vaja 4** – Uporaba tipke kot prekinitev z STM32L1 Discovery

1. **Cilj naloge**: S pomočjo programskega okolja **STM32CubeIDE** in HAL knjižnicami sprogramirajte mikroprocesor tako, da bo ob pritisku modre tipke kot prekinitev na STM32L1 se prižgala/ugasnila zelena LED. Modra LED pa vedno utripa.
2. **Postopek inicializacije periferije**.
3. Zaženite **STM32CubeIDE** in ustvarite nov STM32 projekt (pod zavihkom *information Center*). V zavihku *Board selector* s pomočjo filtrov Vendor, Type in MCU/MPU izberite ustrezno razvojno ploščo, kliknite *Next,* projekt poimenujte **vaja4\_interupt\_tipka\_LED** in kliknite *Finish* (na možnosti opcije za prenastavitev periferije izberite *Yes*, izbrana naj bo tudi opcija perspektive za STM32CubeMX).
4. V *Pinout % Configuraton* pogledu glede na vašo razvojno ploščico ugotovite, ali lahko uporabite modro tipko za **digitalni vhod kot interrupt (GPIO\_EXT…).** Če da, kateri pin je to PA0   
   Če ni, uporabite drugi ustrezni pin \_\_\_\_\_\_\_, ki ga boste prožili s pomočjo Pull-UP vezja.
5. Zapišite naslov izhodnih pinov za **zeleno in modro LED**:

PC9 Green\_LED in PC8 Blue\_LED.

1. Pod *System Core* v razdelku *NVIC* pod NVIC *Interrupt table* omogočimo *EXTI line interrupt.*
2. Preverite, če je prekinitev omogočena pod *System Core 🡪 GPIO 🡪 NVIC* zavihek.
3. Sedaj generirajte kodo tako, da enostavno kliknete ikono *Save* in po potrebi še enkrat potrdimo generiranje kode.
4. **Programiranje v** **IDE:**
5. 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).
6. V *user code begin 4* zapišite funkcijo za klic callback funkcije prekinitve:

void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)

{

UNUSED(GPIO\_Pin);

}

1. Znotraj te funkcije zapišite ukaz za vklop/izklop **zelene** LED (pomagajte si z metodo *toggle*, glej **vaja0a**). HAL\_GPIO\_TogglePin(GPIOC, Green\_Led\_Pin);.
2. Znotraj iste funkcije dodajte še zakasnitev, ki je potrebna, ko uporabimo mehanska tipkala/stikala:

for(uint32\_t i=0; i<10000; i++);

Koliko znaša (v mili sekundah) zapisana zakasnitev, ki jo proizvede zanka **for**? 10000.

1. V *user code begin 3* - zanka *while(1)* - zapišite ukaz za utripanje modre LED (metoda *toggle*, glej **vaja0a**):

**if** (moja\_tipka == 1){

HAL\_GPIO\_WritePin(GPIOC, Blue\_Led\_Pin, *GPIO\_PIN\_SET*);

}

**else**{

HAL\_GPIO\_WritePin(GPIOC, Blue\_Led\_Pin, *GPIO\_PIN\_RESET*);

}

1. V to zanko dodajte ukaz za zakasnitev z funkcijo *Delay* iz knjižnice *HAL*, in sicer pol sekunde (glej vaja0a):

HAL\_Delay(500);.

1. **Naložitev kode (*Run*) in opazovanje delovanja na STM32L1.**
2. Kodo preverite s tipko ***Build*** (ikona za kladivce - *Debug*). Ko je preverjanje končano lahko preverimo, če smo med pisnjem kode naredili kakšno napako sintakse, sicer se pod kodo v oknu *Console* izpiše 0 errors.
3. Priklopite STM32L1 na vaš računalnik preko USB kabla.
4. S tipko ***Run***(zelena ikona za play – *Run as STM…*)prenesete program na STM32L1.
5. 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.
6. Opazujte delovanje (utripanje modre LED). Kaj se zgodi, ko pritisnemo na modro tipko na STM32L1?

Modra led preneha svetiti

Delovanje posnamite s telefonom.

1. Ali pritisk na modro tipko vpliva na utripanje modre LED in zakaj?

Da, preneha svetiti

1. **Vaš projekt** (datoteko *main.c* [25%], slikovni izrezek *Pinout* mikroporcesorjaiz CubeMX [15%]*,* kratek videposnetek delovanja [20%], **naložite v Github** kot nov *Repository* z imenom **Vaja4-interrupt-button-STM32L1** [5%]**.** V *readme* datoteki zapišite vse **odgovore** [30%] na vprašanja v *Readme* ter **komentar** [5%] na delovanje. **Oceno** pridobite iz Github dokumentacije!
2. Dodatna video pomoč z razlago: [Video Tutorial 12](https://www.youtube.com/watch?v=E39UA86ThYU).