Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

*Zavod za automatiku i računalno inženjerstvo*

**Sveprisutno računarstvo**

**1. Laboratorijska vježba**

# 

[**Operacijski sustav FreeRTOS [1]**](#_rg947a707s4g) **2**

[**Pokretanje primjera blink**](#_jcuk3q9o4pyy) **3**

[**Klasa CLed**](#_g2shckx3dzce) **4**

[**Zadatak**](#_h0kur260qiry) **6**

[**Predaja**](#_jsxd4i93hhjc) **6**

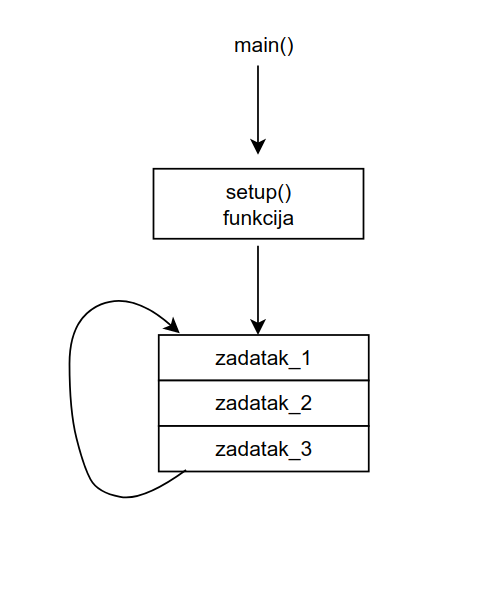
[**Reference**](#_mwicxc3het1b) **6**

# 

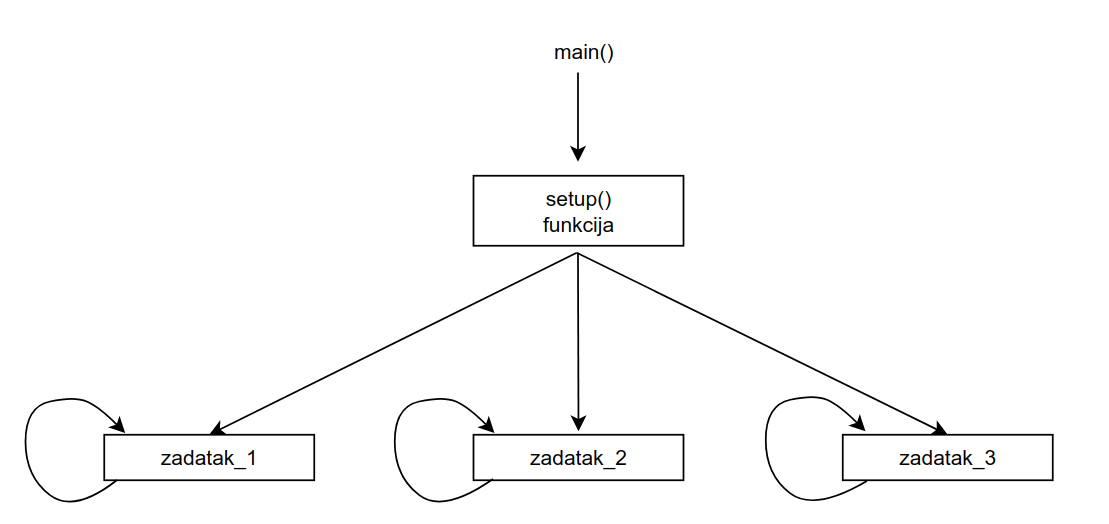
# Operacijski sustav FreeRTOS [1]

Upravljanje, komunikacija, prikupljanje i obrada podataka. Sve su to funkcije koje možemo obavljati pomoću ugradbenih računala. U početku su ugradbena računala imala dosta malu računalnu snagu i koristila se uglavnom za obavljanje jedne funkcije. Za svaki sustav pisala se posebna programska potpora prilagođena mogućnostima sklopovlja. Današnji sustavi imaju znatno veću računalnu snagu. Obavljaju više različitih funkcija, a sve to uz manje troškove izrade, manju potrošnju i mogućnost paralelnog izvođenje zadataka. Postoji cijeli niz operacijskih sustava posebno prilagođenih ugradbenim sustavima s podrškom za rad u stvarnom vremenu (engl. real-time operating system – RTOS).

Veliki dio Ugradbenih računalnih sustava bazirano na mikrokontrolerima izvodi programski kôd unutar takozvane “super petlje”. Nakon inicijalne funkcije postavljanja izvodi se velika beskonačna petlja tj. “super petlja” u kojoj je moguće čitati sa senzora, pisati u memoriju i izvršavati sve napisane naredbe pomoću algoritma kružnog dodjeljivanja (*eng. round robin*).



Operacijski sustavi za rad u stvarnom vremenu (RTOS) dodaju više funkcionalnosti “super petlji”, poput paralelnog izvođenja zadataka i veće kontrole nad raspoređivačem zadataka.



FreeRTOS je operacijski sustav namijenjen ugradbenim računalnim sustavima za rad u stvarnom vremenu i optimiran prema kriteriju malih zahtjeva prema sklopovlju (spremnički prostor i procesorska moć) te se može koristiti i na mikrokontrolerima poput ESP32. Korištenje FreeRTOSa zahtjeva detaljnije poznavanje njegova sučelja i načina rada. Više informacija o FreeRTOSu možete pronaći [ovdje](https://www.freertos.org/a00106.html) [2].

ESP32 ima prilagođenu verziju FreeRTOSa. Razlike između FreeRTOS i ESP-IDF implementacije FreeRTOSa možete pronaći [ovdje](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/freertos-smp.html) [3].

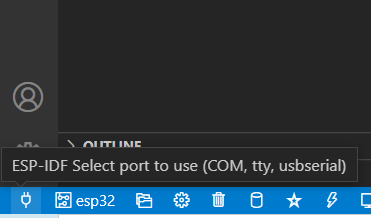
Prilikom izrade laboratorijskih vježbi potrebno je koristiti ESP32 Technical Reference Manual [4].

# Pokretanje primjera blink

* Kliknite na **View -> Command Palette…** i upišite “**ESP-IDF:Show examples Projects”**.
* Pod **get-started** otvorite primjer **blink.**
* U kôdu **blink\_example\_main.c** definirajte izlazni pin za LED pod

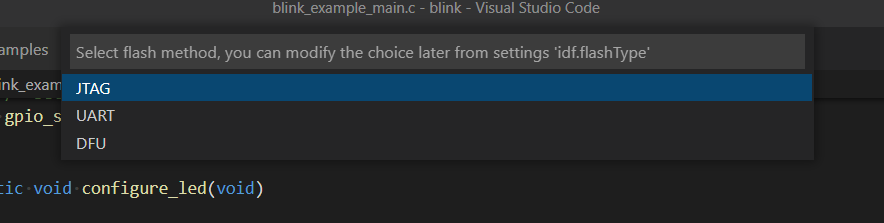
**#define BLINK\_GPIO**

* Spojite mikrokontroler sa računalom.
* U doljnjem lijevom kutu odaberite utor na koji ste spojili mikrokontroler.

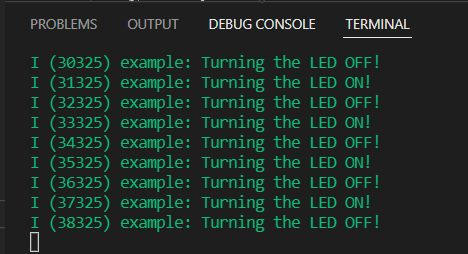


Odaberite pripadajući USB utor i repozitorij vašeg programa.

* Kliknite na **View -> Command Palette…** i upišite “**ESP-IDF: Build, Flash and start a monitor on your device”**.
* Nakon što se izvrši Build, odaberite UART



* Nakon učitavanja koda na mikrokontroler, Led dioda bi se trebala gasiti i paliti:



# Klasa CLed

U primjeru projekta imate klasu CLed koja vam može poslužiti kao primjer za izvedbu klase Cbutton.

# Zadatak

Vaš zadatak potrebno je napraviti klasu CButton kojoj prilikom instanciranja proslijedite GPIO pin na kojem je spojeno tipkalo. Klasa mora razlikovati više stanja tipkala: single-press, double-press i long-press. Ovisno o načinu pritiska tipke klasa mora pozivati vanjsku funkciju koju korisnik odabere zasebno piše.

U glavnom programu napravite instanciranje i postavite tri callback funkcije koje će se zvati u tim situacijama.

Predložak klase CButton nalazi se u primjeru sa predavanja.

# Predaja

Vježbe se predaju preko moodla sukladno uputama koje će biti na web-u.

# 

# Reference

[1] L. JELENKOVIĆ, „Sustavi za rad u stvarnom vremenu – skripta“, <http://www.zemris.fer.hr/~leonardo/srsv/skripta/SRSV-skripta.pdf>

[2] FreeRTOS API Reference,

<https://www.freertos.org/a00106.html>

[3] ESP-IDF FreeRTOS,

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/freertos-smp.html>

[4] ESP32 Technical Reference Manual,

<https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf>