

# Projektna naloga 2 – FreeRTOS Queue in Task Notification

Cilj druge projektne naloge je nadgradnja prejšnje naloge, kjer ste si pripravili opravila in naredili shemo delovanja celotnega sistema. V tej nalogi boste spoznali/implementirali dve komponenti sistema:

- komunikacija med opravili: povedano drugače, pošiljanje podatkov med opravili. V FreeRTOS obstaja več načinov komunikacije med opravili. Verjetno najbolj preprost mehanizem je uporaba globalnih spremenljivk, kar pa ni ravno najboljša praksa. Če že, jih lahko postavimo v kritično sekcijo s tem, da jih ogradimo s sinhronizacijskimi sredstvi (semafor, mutexi, ...), kar pa še vedno ne reši vseh težav (npr. inverzija prioritete). Zato je boljše uporabiti mehanizme, ki že obstajajo in že vsebujejo rešitve za te težave. Primer mehanizmov so: Queues ([LINK](#)), Stream Buffers ([LINK](#)) in Message Buffers ([LINK](#)) (tudi za multi-core sisteme).
- obvestila med opravili (angl. Task Notifications ([povezava](#))): uporabljamo jih za različne namene. **Prva možnost uporabe** je kot mehanizem, s katerim eno opravilo (ali pa recimo prekinitve) signalizira drugemu opravilu, da se je zgodil nek dogodek, ki ga je drugo opravilo čakalo, preden se nadaljuje. Ko dobi signal, se nadaljuje z izvajanjem. To funkcionalnost je možno doseči tudi z uporabo semaforjev (recimo binarnih), ki pa so precej bolj požrešni ([povezava](#)). Običajno želimo to funkcionalnost takrat, ko nočemo, da bi se neka stvar obdelala znotraj prekinitve, ampak želimo, da jo obdelava opravilo z visoko prioriteto. **Druga možnost uporabe** je pravzaprav nadgradnja prve možnosti uporabe in sicer se mehanizem uporablja kot preprost poštni nabiralnik (angl. mailbox) [preprosta vrsta] za pošiljanje podatkov. Gre se za zelo omejeno vrsto, saj je mogoče poslati samo en 32-bitni podatek. Ampak dosti krat je tudi to dovolj. Drugi načini uporabi so še **Event group** ([link](#)) in **števeni semafor** ([link](#)).

# 1 Naloge

V okviru naloge je potrebno dodati oba mehanizma predstavljena zgoraj v vaš projekt. Točke se seštevajo.

## 1.1 Pošiljanje podatkov med opravili [4T]

V tej nalogi je potrebno dodati komunikacijo med opravili. Če s projektom še niste tako daleč, da bi pridobivali/pošiljali dejanske podatke, lahko za zdaj pošiljajte *umetne* (mock) podatke, ki pa naj bodo enakih tipov, kot jih boste pošiljali v projektu (recimo, če boste v projektu iz senzorja pridobivali tri 32-bitna števila, ki jih boste združili v strukturo, si lahko to strukturo pripravite že sedaj in v njo shranite tri poljubna cela števila).

## 1.2 Opozorila med opravili [3T]

Cilj naloge je vključitev predstavljenega mehanizma v projekt. Preučite, na katerih mestih bi ga bilo primerno dodati in ga tam dodajte [oziroma zamenjajte obstoječo rešitev] (recimo, ob proženju prekinitve ali če imate komunikacijo med samo dvema opravilom v eno smer in ne pošiljate *veliko* podatkov). Obvezno ga dodajte vsaj na enem mestu.