



## ARDUINO, vežbe 2

Namenski Računarski sistemi

# Task management

- Zadaci (tasks) su funkcije koje mogu da se automatski startuju u određenim vremenskim intervalima i urade neki zadatak
- Zaglavlje funkcije je oblika:  
`void imeZadatka(int id, void * tptr)`
  - `id` predstavlja identifikator taska
  - `tptr` predstavlja pokazivač na parametre task-a
- Zadaci se startuju funkcijom `createTask()` oblika:  
`int createTask(taskFunc task, unsigned long period, unsigned short stat, void * var)`
  - `task` predstavlja pokazivač na funkciju koja predstavlja zadatak
  - `period` predstavlja vreme u ms, tj. vreme posle kojeg se zadatak ponavlja
  - `stat` predstavlja stanje zadatka po njegovom kreiranju (`TASK_ENABLE`, `TASK_DISABLE`, `TASK_RUN_ONCE` ili broj ponavljanja zadatka)
  - `var` predstavlja početno stanje parametara task-a
- Vraća redni broj task-a ukoliko je on uspešno kreiran, odnosno -1 ako task nije uspešno kreiran

# Task management

- Ostale korisne funkcije task management-a su:
  - `destroyTask(int id)` – uništava task sa rednim brojem id
  - `getTaskId(taskFunc task)` – vraća id prvog taska za neku funkciju
  - `getTaskNextExec(int id)` – vraća vreme sledećeg izvršavanja zadatka sa datim id
  - `getTaskPeriod(int id)` - vraća period izvršavanja zadatka sa datim id
  - `getTaskState(int id)` - vraća stanje zadatka sa datim id
  - `getTaskVar(int id)` - vraća parametar zadatka sa datim id
  - `setTaskPeriod(int id, unsigned long period)` – postavlja period za task sa datim id-em
  - `setTaskState(int id, unsigned short state)` - postavlja stanje za task sa datim id-em
  - `setTaskVar(int id, void * var)` – postavlja vrednosti parametara za task sa datim id-em
  - `startTaskAt(int id, unsigned long start, unsigned short state)` – postavlja vreme startovanja za task sa određenim id-em, kao i početno stanje

# Task management

- Korisna funkcija System management-a:
  - `executeSoftReset(uint32_t options)` - resetuje arduino platformu sa određenim opcijama (RUN\_SKETCH\_ON\_BOOT, ENTER\_BOOTLOADER\_ON\_BOOT)
- Funkcija `random()`:
  - `long random(long howbig)` - vraća broj manji od howbig
  - `long random(long howsmall, long howbig)` – vraća broj iz intervala [howsmall, howbig]
- Funkcija `millis()` vraća broj milisekundi koji je protekao od startovanja Arduino platforme

# Zadaci

- Zadatak 1: Napisati program koji pali i gasi led diode LD8 i LD2 na pinovima 33 i 27 na svakih 500, odnosno 25 milisekundi. Kreirati dva task-a blink\_task1 i blink\_task2, za svaku lampicu po jedan. Nakon toga, kreirati task blink\_task3 koja omogućuje/onemogućuje task blink\_task2 u zavisnosti od stanja prekidača SW2
- Zadatak 2: Kreirati program koji koristeći task blink\_task1 pali i gasi lampicu LD8 na pinu 33 u zavisnosti od dugmeta BTN4 na pinu 37.
  - Pratiti stanje dugmeta BTN4 i otkriti da li je stisnut duže od 2 sekunde. Po uočavanju ovakvog događaja, na silaznoj ivici programski resetovati ploču uc32
  - Koristiti primitive *millis* i *executeSoftReset*
  - Promenu stanja dugmeta pratiti na svakih 50 ms i ukoliko je stanje različito od prethodnog, prepoznati promenu stanja

# Zadaci za samostalan rad

- Zadatak 3: Napisati program koji koristeći task `blink_task1` pali i gasi led diodu LD8 na pinu 33 na svakih 100 milisekundi. Kreirati i task `blink_task2`, koji na svakih 25 milisekundi proverava stanje prekidača na pinovima 2, 7, 8, i 35, broji koliko ima upaljenih prekidača i u zavisnosti od toga, postavlja period blinkanja lampice LD8 na 100, 200, 300, 400 ili 500ms. Period blinkanja menjati periodom na koji se startuje `blink_task1`
- Zadatak 4: Napisati program koji koristeći task `blink_task1` proverava stanje na prekidaču i ukoliko je prekidač upaljen, upaliti lampicu, a ukoliko je prekidač ugašen, ugasiti lampicu. Redni broj prekidača i lampice primiti kao parametar. Koristeći dati task, kontrolisati lampice na pinovima 26, 27, 28 i 29 u zavisnosti od stanja na prikidačima na pinovima 2, 7, 8 i 35.