

Izveštaj V1

25. Stefan Kangrga-Mikulić 0526/2017

- **Zadatak projekta**

Startuje akvizicija sa kanala A0, A1 na svakih 1000 ms (AB=01) pomoću taska xTaskTimer koji koristi vTaskDelayUntil (C=1). Potrebno je implementirati odloženu obradu prekida (*deferred interrupt processing*) AD konvertora, tako što se rezultat konverzije u prekidnoj rutini upisuje u red sa porukama (*Queue*) i obaveštava se task xTask1 o prispeću nove poruke putem direktne notifikacije taskova (*Direct-to-task notification*) u vidu grupe događaja (D=0). Poruka treba da sadrži informaciju o kanalu koji je očitana i gornjih 9 bita rezultata AD konverzije. Task xTask1 čuva poslednju očitanu vrednost za svaki kanal. Task xTask2 ispituje tastere S1-S4 i na pritisak odgovarajućeg tastera (E=0) obaveštava task xTask1 putem direktne notifikacije taskova (*Direct-to-task notification*) u vidu grupe događaja (D=0) o kanalu čije očitane vrednosti rezultata konverzije treba da šalje tasku xTask3. Svaki put kada stigne nova vrednost sa AD konvertora task xTask1 smešta odgovarajući podatak u red sa porukama na kojem čeka task xTask3. Task xTask3 računa razliku između uzastopnih vrednosti očitano kanala i prikazuje na UART-u (F=1).

1. Detekcija pritiska tastera - Ovu logičku celinu implementirano je koristeći prekidne rutine portova i „xTask2“ task. U okviru prekidnih rutina portova detektuje se da se desio potencijalni pritisak tastera dok će odloženu obradu nastaviti u okviru „xTask2“ taska (koji šalje notifikaciju „xTask1“ tasku setovanjem maske `mainBUTTON_S1` ili `mainBUTTON_S4`).
2. Upravljanje AD konvertorom - Start konverzije implementirano je u okviru „xTaskTimer“ taska koji periodično započinje konverziju dok ćemo rezultat konverzije čitati iz prekidne rutine i smeštati rezultat u „ADC task“ tasku (koji kasnije prosleđuje rezultat konverzije „xTask1“ tasku samo što setuje bit 12 ili bit 13 na „1“ i na taj način daje informaciju o kanalu koji je očitana i šalje mu notifikaciju setovanjem maske `mainNEW_MESSAGE`). Promenu aktivnog kanala implementirano je u okviru „xTask1“ taska.
3. Prikaz rezultata preko UART interfejsa - Slanje stringa implementirano je u okviru „xTask3“ taska.

ADC task je dodat u projekat radi spajanja rezultata konverzije i informacije o kanalu sa kojeg je očitana (setovanjem bita 12 ili 13 na "1") i kako bi tako prosledio "xTask1" tasku.

Napomena: Informacija iz kog kanala je dobijen rezultat konverzije je ubačen setovanjem bita 12 ili 13 na "1" na rezultat konverzije i to je implementirano u "ADC task" tasku. Isti ti biti su uklonjeni u "xTask1" tasku pošto nije bilo dalje definisano njihovo korišćenje u tekstu zadatka projekta.

Prilikom konfigurisanja hardvera, inicijalizuje se dugme S4 i S1(koji potiče iz porta 2 za razliku od S4 i zato imamo dve prekidne rutine koji se poziva kao posledica pritiska na jedan od dva tastera dok će se odložena obrada nastaviti u okviru "xTask2" taska), interfejs UART-a i AD konvertor (na početku programa kreće konverzija iz ADC12 ulaznog kanala 0).

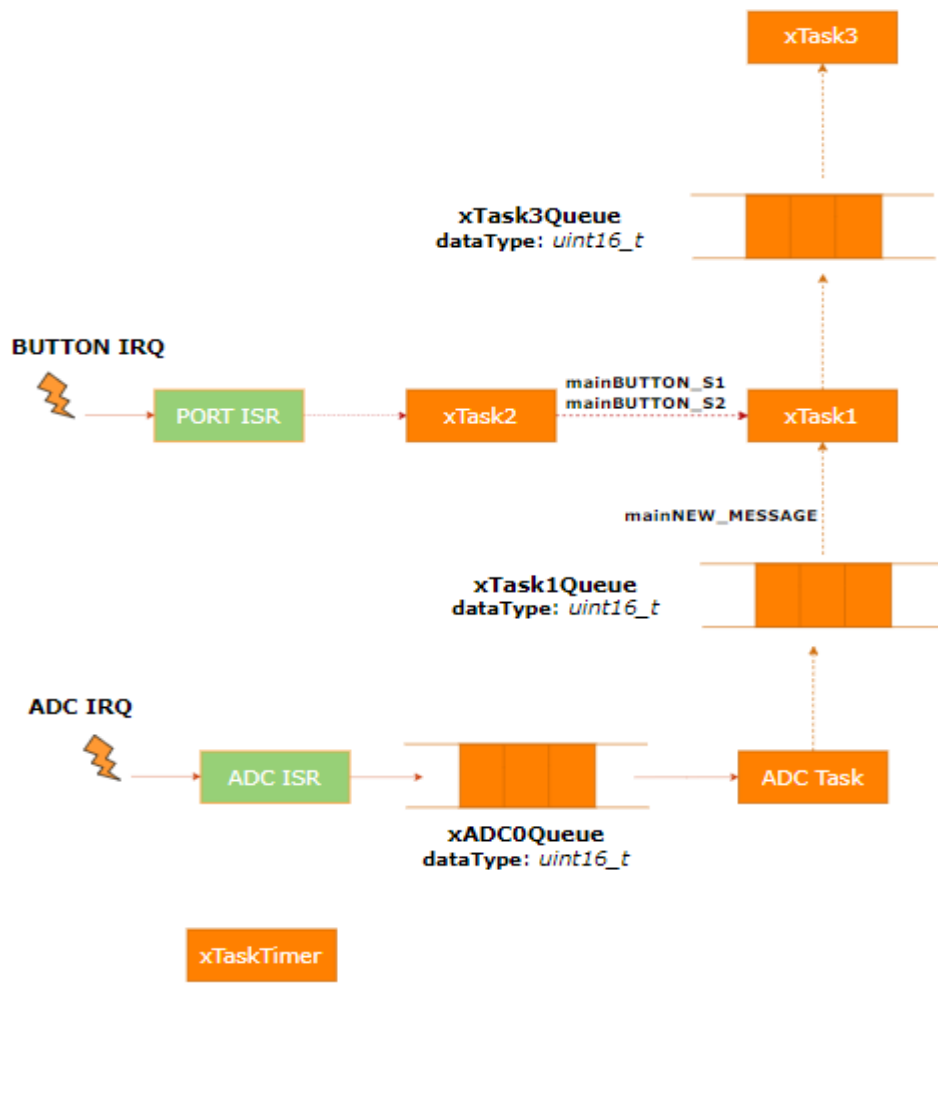
Slično za detekciju dugmeta, i ADC konverzija se odvija kao odložena obrada prekida samo što se obrada nastaviti u okviru "ADC task" taska i već u prekidnoj rutini se skalira vrednost sa 12 bita na 9 bita.

"ADC task" čita rezultat iz prekidne rutine zaduženu za ADC konverziju i setuje bit 12 na 1 ako je rezultat dat iz kanala A0 ili setuje bit 13 na 1 ako je rezultat dat iz kanala A1. Rezultat obrade šalje dalje u "xTask1" i šalje notifikaciju za setovanjem maske `mainNEW_MESSAGE`. Aktivan kanal konverzije je poznat na osnovu vrednosti promenljive `buttonCommand`.

"xTask3" ispituje da li je došao novi broj zatim ispituje da li je veći ili manji od prethodnog da bi rezultat razlike koji se šalje na UART bio pozitivan bez obzira na smer okretanja potencijometra. Zatim se šalje odgovarajuća poruka koji ukazuje na aktivan kanal konverzije(npr. „Kanal 0: "...), onda se šalje rezultat tako što se konvertuje u niz znakova koji preklapaju znakove iz maske promenljive `string_mask`. I onda se šalje na UART znakovi za novi red na početak.

Prioriteti taskova su tako postavljeni da “xTask3” radi sve dok ne dodje do prekida(ili uzrokovanom pritiskom tastera ili početak konverzije koji pokreće “xTimerTask”).

- **Blok dijagram**



- **Blok aktivnosti**

