

FreeRTOS 2/3

Wstęp

Do wykonania poniższych zadań wykorzystany zostanie wyświetlacz 7-segmentowy. Dokładny schemat połączeń znajduje się w materiałach dodatkowych. Za kontrolę konkretnego segmentu (linii) odpowiedzialne są piny wyjściowe oznaczone literami od A do G. W celu aktywacji konkretnej pozycji (cyfry) należy ustawić stan niski na jednym z wyjść oznaczanych jako COM. W celu wyświetlenia zróżnicowanej sekwencji na wielu cyfrach należy wykorzystać multipleksowanie.

Przykładowo, wyświetlenie liczby 11 wymaga ustawienia:

- stanu wysokiego na segmentach A i B
- stanu niskiego na segmentach C, D, E, F, G
- stanu niskiego na pozycjach COM1, COM2
- stanu wysokiego na pozycjach COM3, COM4

W tym ćwiczeniu nie umieszczamy żadnych instrukcji w pętli głównej programu.

Do obsługi wyświetlacza wykorzystaj przygotowane materiały dostępne na stronie prowadzącego.

Przebieg ćwiczenia

1. Wstęp

Do projektu należy dołączyć obsługę FreeRTOS w wersji V1.

2. Sygnał zegarowy

System FreeRTOS do pracy wykorzystuje licznik sprzętowy o nazwie SysTick. Domyślnie jest on wykorzystywany przez bibliotekę HAL do odmierzenia czasu.

W zakładce SYS należy przypisać inny licznik dla biblioteki HAL np. TIM14.

3. FirstTask

Zmień nazwę domyślnego zadania na *FirstTask*, zostaw domyślny priorytet.

W pętli głównej zadania *FirstTask* ustaw naprzemiennie na wyświetlaczu liczby 1111 i 2222 co 1 sekundę.

Wykorzystaj do tego zadania funkcje opóźniające *osDelay* oraz *HAL_Delay*.

Porównaj i wyjaśnij różnice w działaniu podczas wykorzystania funkcji opóźniających *osDelay* oraz *HAL_Delay*.

4. SecondTask

Dodaj zadanie *SecondTask* z priorytetem normalnym.

Ustaw parametr *TICK_RATE_HZ* na 10 (FreeRTOS->Config parameteres)

W pętli głównej zadania *FirstTask* ustaw wyświetlacz na 1111 co 1 sekundę.

W pętli głównej zadania *SecondTask* ustaw wyświetlacz na 2222 co 1 sekundę.

Uruchom każdy z projektów w trybie *Debug*, umieść *breakpointy* w liniach zmieniających stan wyświetlacza.

Monitoruj stan zmiennej *uwTick* (czas pracy mikrokontrolera w ms) w zakładce *Live Expression*.

Porównaj i wyjaśnij różnice w działaniu podczas wykorzystania funkcji opóźniających *osDelay* oraz *HAL_Delay*.

5. Priorytety

Ustaw priorytet zadania *SecondTask* na *osPriorityAboveNormal*.

Porównaj i wyjaśnij różnice w działaniu podczas wykorzystania funkcji opóźniających *osDelay* oraz *HAL_Delay*.

6. IdleHook

Ustaw parametr USE_IDLE_HOOK na *Enable* (FreeRTOS->Config parameteres).

W funkcji *vApplicationIdleHook* ustaw wyświetlacz na 0000.

Zadania *FirstTask* i *SecondTask* pozostaw bez zmian.

Porównaj i wyjaśnij różnice w działaniu podczas wykorzystania funkcji opóźniających *osDelay* oraz *HAL_Delay*.

7. TickHook

Ustaw parametr USE_IDLE_HOOK na *Disable* (FreeRTOS->Config parameteres).

Ustaw parametr USE_TICK_HOOK na *Enable* (FreeRTOS->Config parameteres).

W funkcji *vApplicationTickHook* ustaw wyświetlacz na 0000.

Zadania *FirstTask* i *SecondTask* pozostaw bez zmian.

Porównaj i wyjaśnij różnice w działaniu podczas wykorzystania funkcji opóźniających *osDelay* oraz *HAL_Delay*.

8. Multipleksowanie

Ustaw parametr USE_IDLE_HOOK na *Disable* (FreeRTOS->Config parameteres).

Ustaw parametr USE_TICK_HOOK na *Disable* (FreeRTOS->Config parameteres).

Usuń zadania *FirstTask* i *SecondTask*.

Ustaw parametr TICK_RATE_HZ na 1000 (FreeRTOS->Config parameteres)

Dodaj zadanie *displayTask* którego celem będzie wyświetlanie dowolnej liczby na wyświetlaczu 7-segmentowego za pomocą multipleksowania. Na wyświetlaczu przedstaw licznik czasu pracy mikrokontrolera w ms. Wypróbuj różne częstotliwości odświeżania ekranu.

Funkcje

HAL

- void **HAL_Delay** (uint32_t Delay)
- uint32_t **HAL_GetTick** (void)

FreeRTOS

- osStatus **osDelay** (uint32_t millisec)

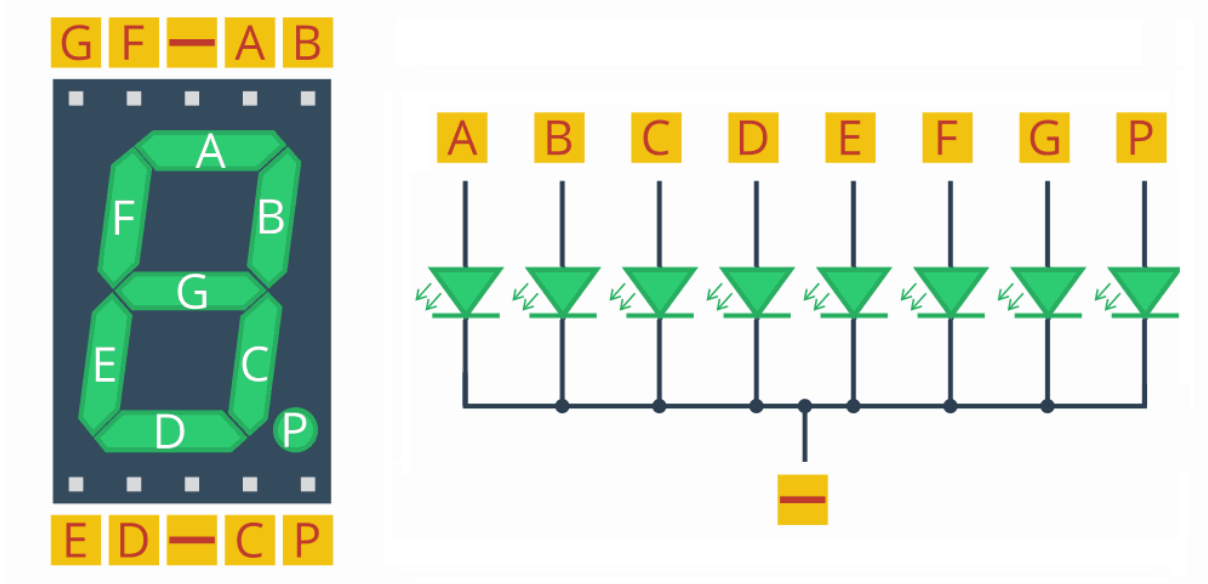
Display

- void **DISP_SetSegments**(enum dispSeg seg)
- void **DISP_SetPositions**(enum dispPos pos)
- void **DISP_SetDigit**(uint8_t digit)

Materiały dodatkowe

- [Description of STM32F4 HAL and low-layer drivers - User manual](#)
- [FreeRTOS API categories](#)
- [KA-Nucleo-Multisensor \(PL\) \(kamamilabs.com\)](#)

Wyświetlacz 7-segmentowy



STM32CubeMX

