

Serijska komunikacija z UART v STM32F7

Zastavice

Stanje naprave lahko spremljamo preko zastavic. Spodaj je zapisan seznam vseh pomembnejših zastavic s kratkimi opisi njihovega pomena. Stanje zastavice preverjamo s funkcijo `_HAL_UART_GET_FLAG()`, brišemo pa jih s funkcijo `_HAL_UART_CLEAR_FLAG()`. Zastavica `RXNE` (prejeli smo nove podatke) se pobriše samodejno, ko preberemo prispeli podatek, tako da nam te zastavice ni potrebno brisati ročno.

```
1 // uporabljen pri HW Flow Control. oznacuje da naprava lahko
   // poslje podatek
2 UART.FLAG.CTS
3 // lahko zacnemo naslednji prenos, funkcija Transmit zastavico
   // preveri sama
4 UART.FLAG.TXE
5 // zadnji prenos se je zakljucil
6 UART.FLAG.TC
7 // prejeli smo nov podatke
8 UART.FLAG.RXNE
9 // napaka paritete
10 UART.FLAG.PE
```

Prekinitve UART naprav

Vse zastavice lahko uporabimo tudi za proženje prekinitiev. Imena prekinitiev se ujemaajo z imeni zastavic, le da `_FLAG_` zamenjamo z `_IT_`. Za vklop prekinitve uporabimo funkcijo `_HAL_UART_ENABLE_IT()`, podobno kot ste to storili pri časovnikih. Ne pozabite, da je še vedno potrebno vklopiti prekinitve za izbrano UART napravo na prekinitvenem krmilniku (NVIC) ter jim določiti prioriteto. Spodaj je podana tabela prekinitvenih oznak ter imen prekinitveno-servisnih programov za vse UART naprave. V primeru realizacije branja s prekinitvami lahko znotraj PSP-ja uporabite funkcijo `HAL_UART_Receive`, le da v tem primeru vedno preberete samo en podatek (tisti, ki je bil ravnokar prejet), timeout pa obvezno nastavite na 0! V naslednjem podpoglavju bomo sicer spoznali enostavnejšo rešitev za pošiljanje in branje s prekinitvami.

UART	Prekinitvene oznake	Imena PSP-jev
USART1	USART1_IRQn	USART1_IRQHandler
USART2	USART2_IRQn	USART2_IRQHandler
USART3	USART3_IRQn	USART3_IRQHandler
UART4	UART4_IRQn	UART4_IRQHandler
UART5	UART5_IRQn	UART5_IRQHandler
USART6	USART6_IRQn	USART6_IRQHandler
UART7	UART7_IRQn	UART7_IRQHandler
UART8	UART8_IRQn	UART8_IRQHandler

Ne-blokirajoče pošiljanje in sprejemanje

Poleg blokirajočih funkcij `HAL_UART_Transmit()` in `HAL_UART_Receive()`, ki smo jih spoznali prejšnji teden, obstajata še funkciji, ki omogočata, da program nadaljuje z izvajanjem, branje in pošiljanje pa se spremlja in izvaja preko prekinitev. Za ta namen sta na voljo funkciji `HAL_UART_Transmit_IT()` ter `HAL_UART_Receive_IT()`. Omenjeni funkciji vklopita vse potrebne prekinitve USART naprave, ki se ob zaključku prenosa tudi izklopijo.

Pri uporabi ne-blokirajočih UART funkcij je potrebno v PSP dodati funkcijo `HAL_UART_IRQHandler(UART_HandleTypeDef*)`. Ta funkcija namesto vas poskrbi, da se v primeru sprejemanja, prebere prejeti podatek in shrani na pravilno mesto v podani spremenljivki. Funkcija v PSP-ju prav tako pobriše vse zastavice in v primeru, da so bili prejeti vsi pričakovani bajti izklopi prekinitve. V primeru pošiljanja funkcija v PSP-ju poskrbi, da se začne pošiljati naslednji zeleni znak, oziroma zaključi pošiljanje, če so bili prenešeni vsi podatki. Prekinitveno servisni program za napravo USART2 morajo torej izgledati kot je prikazano spodaj.

```

1 void USART2_IRQHandler()
2 {
3     HAL_UART_IRQHandler(&uart);
4 }

```

Če želimo, lahko ob zaključku pošiljanja `HAL_UART_IRQHandler()` pokliče tudi tako imenovano callback funkcijo. V primeru končanega pošiljanja se pokliče funkcija `HAL_UART_TxCpltCallback(UART_HandleTypeDef*)`, v primeru branja pa `HAL_UART_RxCpltCallback(UART_HandleTypeDef*)`. Ti dve funkciji lahko definirate v poljubnem delu projekta. Bodite pa pozorni na

to, da se funkciji kličeta iz prekinitveno-servisnega programa. To pomeni, da si tudi tu želimo, da se zaključita čimprej.