Osnove mikroprocesorske elektronike

Vaja 8: printf

Naloge:

- Napišite funkcije za pisanje na LCD prikazovalnik z ukazom »printf«. Ko pride tekst do konca vrstice, naj se nadaljuje v naslednji vrstici. Ko pride do konca zadnje vrstice, naj nadaljuje na začetku prve vrstice. Če je znak '\n', naj skoči v naslednjo vrstico in če je znak '\r', naj skoči na začetek trenutne vrstice.

Navodila:

Priprava podatkovnega toka:

Če želimo za pisanje na poljuben medij (LCD, serijski vmesnik,…) uporabiti funkcijo `printf` ali `fprintf`, moramo vključiti deklaracijsko datoteko `<stdio.h>`. Poleg tega moramo ustvariti podatkovni tok (data stream), ki je spremenljivka tipa `FILE`. Inicializiramo ga z makrojem `FDEV_SETUP_STREAM`. Ta makro sprejme tri parametre:

- naslov funkcije, ki napiše en znak,
- naslov funkcije, ki prebere en znak in
- tip podatkovnega toka (za branje, za pisanje ali oboje)

Če ene od obeh funkcij naprava ne podpira, lahko namesto naslova ustrezne funkcije podamo ničelni kazalec `NULL`.

```
FILE mystr = FDEV_SETUP_STREAM(myputc, NULL, _FDEV_SETUP_WRITE);
```

Funkcija za pisanje mora imeti sledečo deklaracijo (ime je seveda lahko poljubno):

```
int myputc (char c, FILE *stream);
```

Če izpis znaka uspe, mora vrniti vrednost 0, sicer -1. Parameter »`c`« je znak, ki ga je treba napisati. Parameter »`*stream`« kaže na podatkovni tok in ga lahko ignoriramo.

Funkcija `int fprintf(FILE *stream, char *fmt, …)` sprejme dva ali več parametrov. Prvi je naslov podatkovnega toka, na katerega želimo pisati, drugi je znakovni niz, ki ga želimo izpisati, sledijo morebitne spremenljivke, ki jih želimo izpisati. Primer klica:

```
fprintf( &mystr, "Hello world!");
```

Kazalec na standardni izhodni podatkovni tok - stdout

Funkcija `int printf(char *fmt, …)` opravlja enako funkcijo kot `fprintf`, le da znakovni niz izpiše na podatkovni tok, katerega naslov je podan v kazalcu `FILE *stdout`. Klic funkcije

```
printf("Hello world!");
```

v osnovi izvede funkcijo

```
fprintf(stdout,"Hello world!");
```

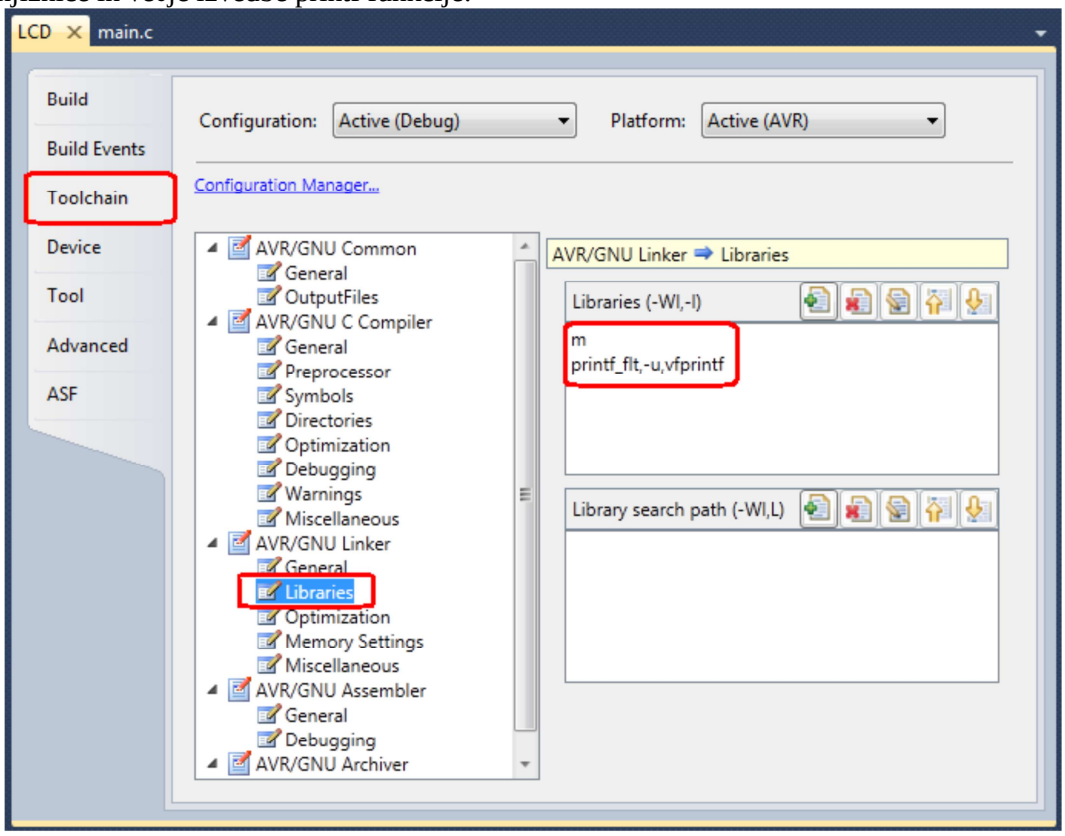
Kazalec `stdout` je deklariran v `<stdio.h>` in kaže na podatkovni tok brez funkcij za branje in pisanje. Če želimo uporabljati funkcijo `printf`, moramo kazalec `stdout` nastaviti, da kaže na naš podatkovni tok:

```
stdout=&mystr;
```

To običajno naredimo na začetku funkcije main, takoj po inicializaciji naprave, na katero želimo izpisovati (v našem primeru LCD prikazovalnika).

Izpis številk z decimalno piko

Celotna izvedba funkcije printf (in njenih sorodnih funkcij) je zelo velika. Pri majhnih mikrokrmilnik je to lahko velik problem, hkrati pa velikokrat ne potrebujemo vseh možnosti, ki jih celotna izvedba nudi. AVR libc vsebuje tri izvedbe: minimalna, privzeta in maksimalna. Le maksimalna podpira izpis sprememnljivk s plavajočo vejico. Če želimo izpisovati tudi števila z decimalno piko, moramo v nastavitvah povezovalnika (ang.: linker) nastaviti uporabo matematične knjižnice in večje izvedbe printf funkcije:



Izpis znakovnih nizov neposredno iz programskega spomina (manjša poraba RAMa)

Vsi znakovni nizi se pred klicem funkcije main naložijo v podatkovni spomin mikrokrmilnika. Ker je ta pri mikrokrmilnikih običajno precej majhen, to lahko predstavlja problem. Znakovne nize, ki se med delovanjem programa ne spreminjajo, lahko deklariramo in izpisujemo s posebej za to pripravljenimi makroji in funkcijami. Če jih želimo uporabljati, moramo vključiti deklaracijsko datoteko `<avr/pgmspace.h>`. Primer izpisa konstantnega niza:

```
printf_P(PSTR("Hello world!"));
```