

snprintf functie.

© Harry Broeders. Laatste wijziging: 3 maart 2007

Deze pagina is bestemd voor studenten van de Haagse Hogeschool - TH Rijswijk/Academie voor Engineering.

De functie `snprintf` kan gebruikt om data geformatteerd naar een C-string te sturen. Het prototype van deze functie is gegeven in `<stdio.h>`:

```
int snprintf (char* s, size_t n, const char* fmt, ...);
```

Deze functie formateert de na `fmt` meegegeven parameters op de manier zoals in de format string `fmt` is opgegeven. Het resultaat wordt weggeschreven op de plaats die door de `char* s` wordt aangewezen en er worden maximaal `n` karakters weggeschreven (het laatste weggeschreven karakter is altijd het `'\0'` karakter). De functie geeft het aantal geformatteerde karakters (ook inclusief `'\0'`) terug. Als de retronwaarde groter is dan `n` dan was er dus niet voldoende ruimte.

Deze functie kan dus gebruikt worden om data te formateren en naar een `char` array te sturen op dezelfde manier als de bekende `printf` functie gebruikt kan worden om data te formateren en naar het beeldscherm te sturen. Bij het programmeren van microcontrollers is er meestal geen "normaal" beeldscherm (`stdout`) aanwezig. De functie `snprintf` kan dan gebruikt worden om data te formateren in een C-string. Deze string kan dan vervolgens op een eenvoudig LCD display weergegeven worden of via een seriële verbinding naar een PC of andere microcontroller verstuurd worden.

De `avr-libc` library bevat diverse varianten van de `printf` functie waaronder `snprintf`. Onderaan pagina 85 en op pagina 86 en 87 van de [avr-libc-user-manual-1.4.4.pdf](#)  staat beschreven hoe de format string `fmt` gebruikt kan worden.

Voorbeeld met een int.

Hieronder is een AVR gcc programma gegeven dat een `int` formateert in een C-string als decimaal getal en rechts uitlijnt op een veldbreedte van 6 karakters met de format string `"%6d"`. De C-string wordt vervolgens op een LCD display weergegeven met behulp van de [LCD functies](#).

```
#include <stdio.h>
#include "lcd.h"

int main() {
    int i=0;
    char buffer[11];
    lcd_init();
    lcd_cursor(false, false);
    while (1) {
        lcd_home();
        snprintf(buffer, sizeof buffer, "i = %6d", i++);
        lcd_puts(buffer);
    }
    return 0;
}
```

De operator `sizeof` bepaalt de grootte van de variabele `buffer` in bytes. Dit is dus het maximale aantal karakters wat `snprintf` mag wegschrijven omdat anders voorbij het einde van de variabele `buffer` wordt geschreven.

Voorbeeld met een float.

Hieronder is een AVR gcc programma gegeven dat een `float` formateert in een C-string als floatingpoint getal en rechts

uitlijnt op een veldbreedte van 8 karakters met 6 cijfers achter de decimalpoint met de format string "%8.6f". De C-string wordt vervolgens op een LCD display weergegeven met behulp van de [LCD functies](#).

```
#include <stdio.h>
#include <math.h>
#include "lcd.h"

int main() {
    char buffer[14];
    lcd_init();
    lcd_cursor(false, false);
    snprintf(buffer, sizeof buffer, "pi = %8.6f", M_PI);
    lcd_puts(buffer);
    while(1);
    return 0;
}
```

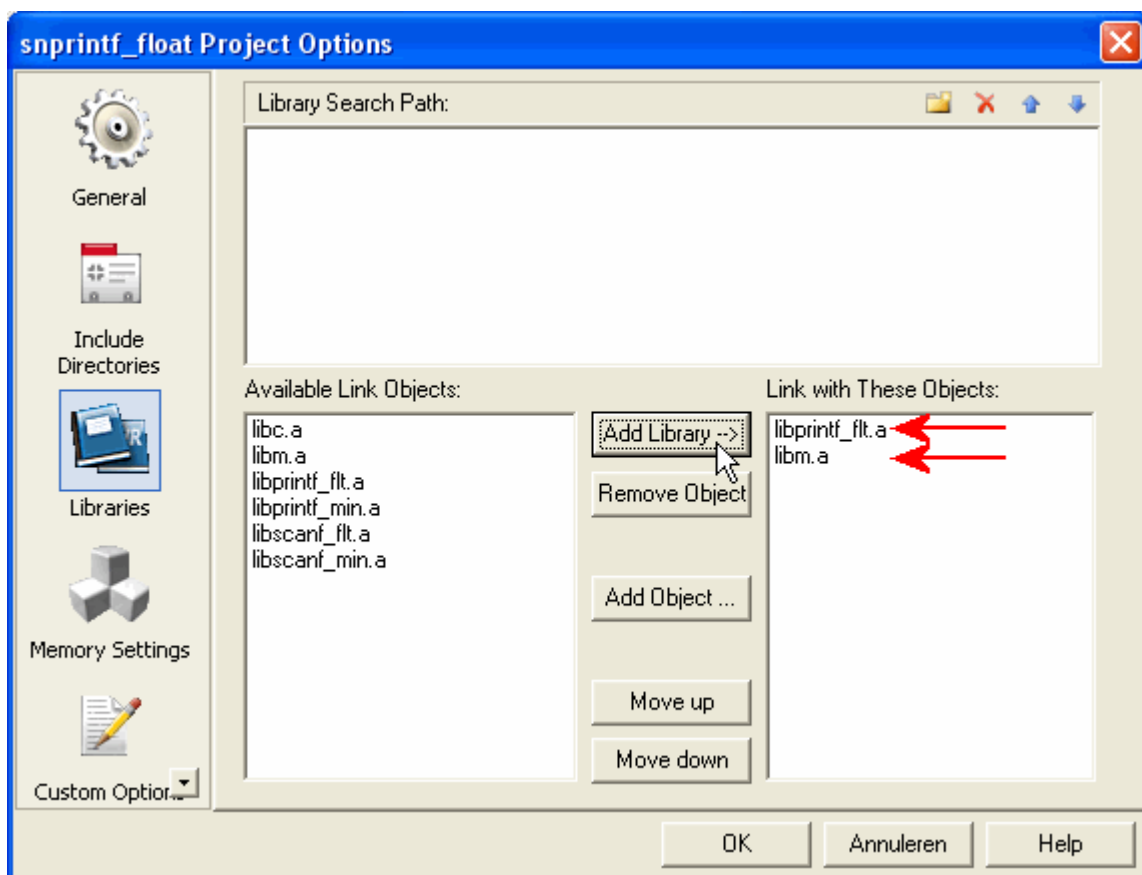
De constante M_PI is gedefinieerd in <math.h>.

De uitvoer van dit programma is:

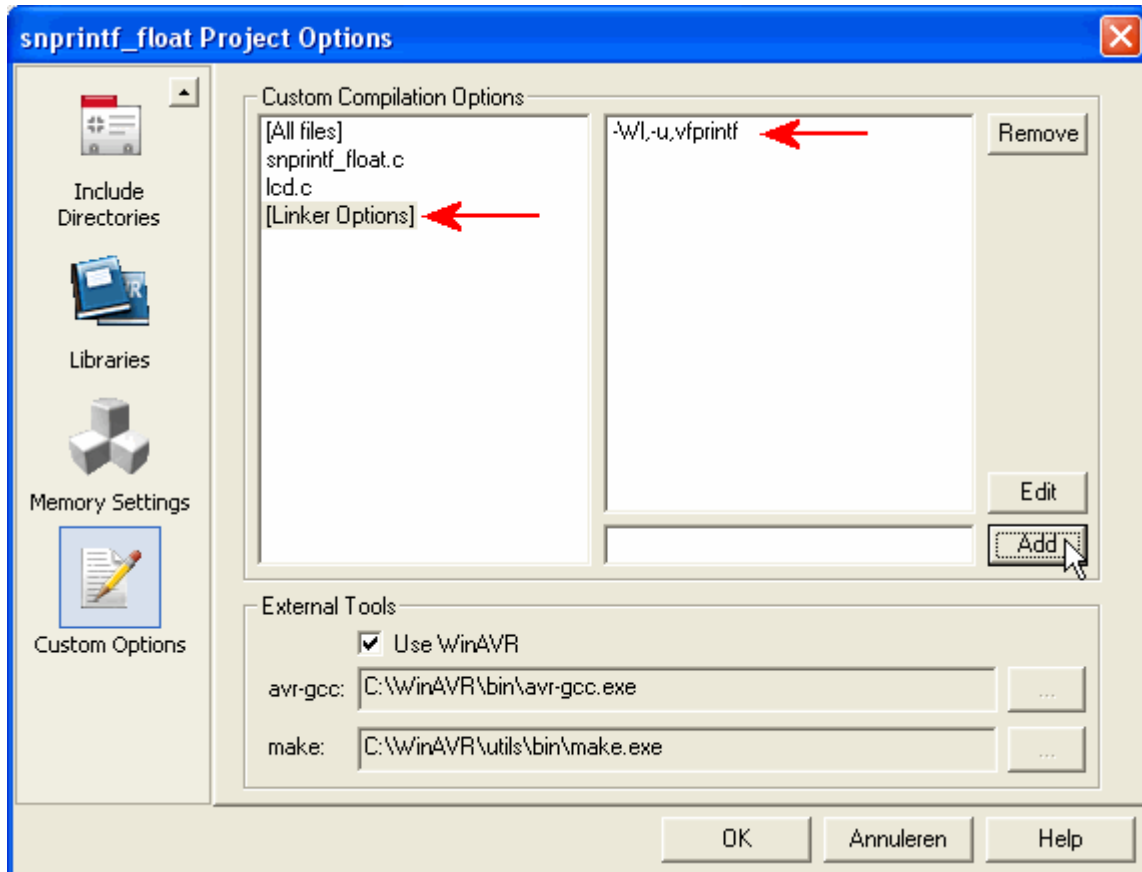
pi = ?

Om floats te kunnen gebruiken moet je het volgende doen:

- Klik in de Project Configuration Options op Libraries en voeg de libraries `libprintf_float.a` en `libm.a` toe. **Let op:** de **volgorde** is belangrijk!



- Klik in de Project Configuration Options op Custom Options, klik vervolgens op Linker Options en voeg de optie -wl, -u, vfprintf toe.



De uitvoer van het programma is nu wel correct:

pi = 3.141593

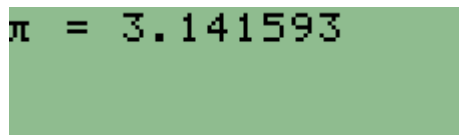
Gebruik van het griekse karakter pi.

De LCD wordt aangestuurd met een een [Hitachi HD44780 controller](#). Op pagina 17 van dit document kun je de karakterpatronen vinden. Je ziet dat de griekse letter pi kan worden weergegeven met karaktercode 0xF7. In het volgende programma maken we hier gebruik van:

```
#include <stdio.h>
#include <math.h>
#include "lcd.h"

int main() {
    char buffer[13];
    lcd_init();
    lcd_cursor(false, false);
    snprintf(buffer, sizeof buffer, "%c = %8.6f", 0xF7, M_PI);
    lcd_puts(buffer);
    while(1);
    return 0;
}
```

De uitvoer van dit programma is:

A green rectangular area representing an LCD screen. Inside, the text "π = 3.141593" is displayed in a black, monospaced font. The pi symbol is a simple Greek letter.

Helaas geeft de LCD simulatie van de AVE AVR Plugins het teken pi niet weer :-(