

Die folgende Aufgabe dient zur Einarbeitung in Programmierung mit C. Bitte wiederholen Sie zunächst die C Grundlagen aus dem ersten Semester.

Aufgabe „Buchstaben ausgeben“

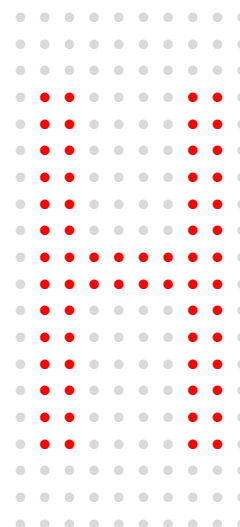
In dieser Aufgabe sollen Sie einen C Code schreiben, der einen einzelnen Buchstaben am Display ausgibt. Ein Buchstabe wird dabei aus Pixeln dargestellt, die Sie entweder ausmalen oder leer lassen müssen. Später werden wir diese Funktion im Praktikum verwenden, um Text auf dem LCD Display auszugeben.

Beschreibung

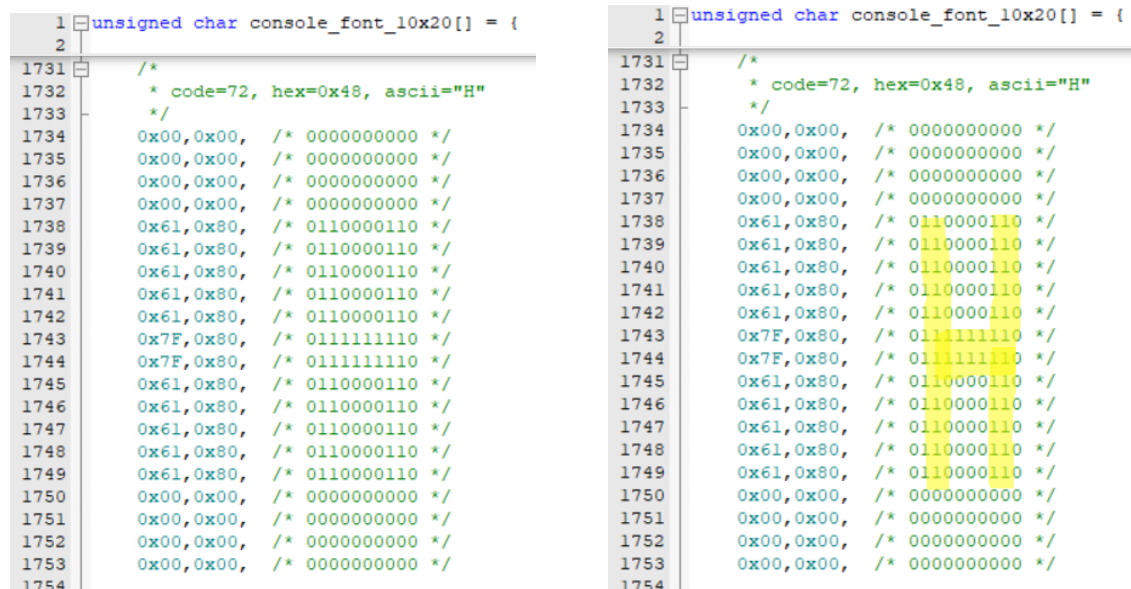
Für diese Aufgabe verwenden wir zur Ausgabe die Funktion `printf()`, mit der Sie entweder ein „*“ oder ein Blank ausgeben. Schreiben Sie also eine Funktion `LCD_WriteLetter()`, die einen Buchstaben als Parameter hat und dann ein Pixel-Muster ausgibt, dass diesem Buchstaben entspricht.

Hinweise

- Zum Testen / Entwickeln können jede beliebige C Entwicklungsumgebung verwenden, wie die VM aus dem ersten Semester, oder WSL. Im Praktikum nutzen wir die Keil Entwicklungsumgebung (<https://www.keil.com/demo/eval/arm.htm>); für die Einrichtung lesen Sie bitte die Praktikumsunterlagen durch. Oder Sie programmieren gleich in Moodle.
- Die Buchstabenmuster sind in einer eigenen Datei `fonts.h` hinterlegt. Diese Datei müssen Sie mit `#include` in Ihrem Source-Code integrieren. Die Datei enthält einen kompletten ASCII-Satz mit 256 verschiedenen Zeichen (manche davon leer). Der Font hat die Ausmaße 10x20, d.h. jedes Zeichen besteht aus 20 Zeilen mit jeweils 10 Pixel. Die Information ist bit-kodiert: jede Zeile wird durch zwei 8-Bit-Zahlen dargestellt (insgesamt 16 Bit pro Zeile, wobei die letzten 6 Bits ungenutzt sind).
- Die Ausgabe eines Buchstabens geschieht, indem jedes einzelne Pixel für den Buchstaben gezeichnet wird. Im praktischen Teil zeichnen wir für jeden Buchstaben ein Rechteck mit 10 Pixel Breite und 20 Pixel Höhe. In dieses Rechteck zeichnen jedes Pixel entweder als * oder als Leerzeichen. Wenn wir das richtige Muster zeichnen, so erscheint ein Buchstabe am Bildschirm, z.B. Buchstabe H:



- Woher wissen wir, welche Punkte wir als * oder als Blank zeichnen müssen? Dazu verwenden wir eine Fonts-Tabelle. Für jeden Buchstaben sind dort alle Pixel hinterlegt. Dazu wird ein Bitmuster verwendet: 0 bedeutet Hintergrund, 1 bedeutet Vordergrund. Da wir 10 Pixel pro Zeile benötigen, verwendet die Fonttabelle 2 Byte pro Zeile. Bei 20 Zeilen für einen Buchstaben werden also 40 Bytes benötigt. Dies ist im folgenden Bild dargestellt: links ein Auszug aus der font.h Datei, rechts sind zur Verdeutlichung die 1-Pixel nachgefärbt.



- Wie finden wir den richtigen Buchstaben in der Tabelle? Die Tabelle ist nach dem ASCII-Code angeordnet. Um nun die Pixelwerte für den Buchstaben H zu finden, müssen wir den ASCII-Code ermitteln (für H ist das 0x48 bzw. 72). Da jeder Buchstabe 40 Byte belegt, müssen wir noch mit 40 multiplizieren, und somit beginnt H im Feld console_font_10x20 an Position $72 * 40 = 2880$.

Wenn Sie ein eigenes Programm schreiben, verwenden Sie eine Endlosschleife im Hauptprogramm, in der Sie einen Buchstaben einlesen und dann auf der Konsole mittels WriteLetter() ausgeben. Zum Einlesen des Zeichens verwenden Sie:

- Microsoft Windows:** z.B. `_getch()` (in der Systemdatei `conio.h` definiert).
- Linux:** z.B. `getchar()`. Die Funktion ist jedoch zeilengepuffert, d.h. Sie müssen immer einen Buchstaben und dann ENTER drücken. Das Enter können Sie dann ignorieren. Ihr Code könnte so ausschauen:

```
#include <stdio.h>

char c = 0;
while( c != 27 ) {
    c = getchar();
    if( c != 10 ) {
        printf("ASCII-Code: %d \n", c);
        LCD_WriteLetter( c );
    }
}
```