|  |
| --- |
| Lötübung 7-Segment  Testprogramm |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Änderung | Name | Datum |
| 1.0 | Erstellung | Thomas Plüss | 26. Okt. 2016 |

Inhalt

[1 Aufgabenstellung 3](#_Toc461527044)

[2 Hardware-Struktur 3](#_Toc461527045)

[2.1 Systembeschreibung 3](#_Toc461527046)

[2.2 In- / Outputs 3](#_Toc461527047)

[2.2.1 Spezielle Hinweise 3](#_Toc461527048)

[2.2.2 Inputs 4](#_Toc461527049)

[2.2.3 Outputs 4](#_Toc461527050)

[3 Software-Analyse 4](#_Toc461527051)

[3.1 Datenstruktur 4](#_Toc461527052)

[3.2 Grafische Lösung (Struktogramme) 5](#_Toc461527053)

[4 Source-Code 5](#_Toc461527054)

[5 Testprotokoll 7](#_Toc461527055)

[5.1 Testablauf 7](#_Toc461527056)

[5.2 Testbericht mit Fazit und Massnahmen 7](#_Toc461527057)

# Aufgabenstellung

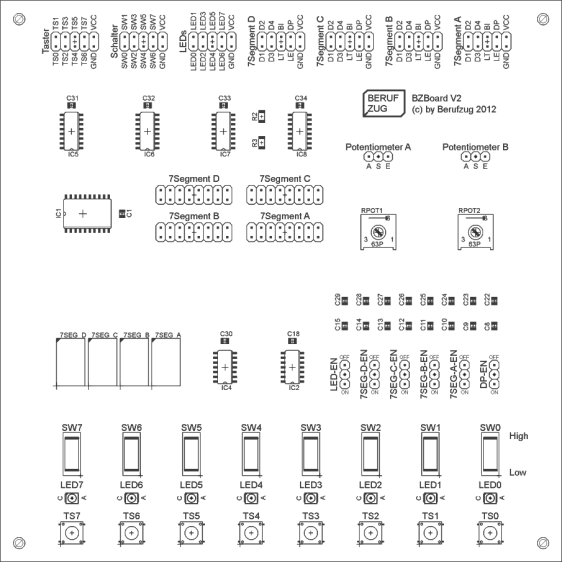
Das Programm soll einen einfachen aber möglichst vollständigen Funktionstest der Lötübung 7-Segment ermöglichen. Während des Testablaufs müssen alle Segmente inkl. Doppelpunkt und Dezimalpunkt mindestens einmal aufleuchten. Idealerweise erfolgt die Ansteuerung über den Decoder-IC (Ausnahmen Doppel- und Dezimalpunkt).

# Hardware-Struktur

## Systembeschreibung

* Mikrocontroller: Atmel ATmega2560
* Hardware: STK600 und BERUF ZUG Development Board
* Portverbindung: Die Ports werden mit ihrer Peripherie über Flachbandkabel verbunden. Der Anschluss des  
   7-Segment Displays erfolgt über einzelne Litzen individuell.

## In- / Outputs

### Spezielle Hinweise

Auf dem Developmentboard sind die uC-Ports über Flachbandkabel mit der Peripherie verbunden.  
Die Kabelverbindungen entsprechen den nachfolgenden Tabellen, in welchen auch die Zuordnungen  
zwischen der Beschreibung aus der Aufgabenstellung, dem Codebezeichner, dem uC-Anschluss und  
dem Bauteil definiert.

**Logik STK600**

Taster: 1= Nicht gedrückt 0= Gedrückt

LED: 1= Dunkel 0= Leuchtet

**Logik BZ-Board**

Taster: 1= Gedrückt 0= Nicht gedrückt

Schalter: 1= Schalterposition oben 0= Schalterposition unten

LED: 1= Leuchtet 0= Dunkel

### Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code-Bezeichner** | **Anschluss** | **Bauteil** | **Signalbeschreibung** | **Beschreibung** |
|  | PINA0 | TS0 | 1 = Gedrückt | Zufallszahl 0..9 generieren |
|  | PINB0 | SW0 | 1 = Ein | Zähler läuft |

### Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code-Bezeichner** | **Anschluss** | **Bauteil** | **Signalbeschreibung** | **Beschreibung** |
|  | PORTC | LED0..3 |  | Binär-Ausgabe (0..9) |
|  | PORTL 0..7 |  |  | Ansteuerung Decoder Verkabelung gemäss SourceCode |
|  | PORTG 0 |  |  | Ansteuerung Doppelpunkt |

# Software-Analyse

## Datenstruktur

Auf Grund der geringen Aufgabengrösse wird auf die Darstellung der Datenstruktur verzichtet. Detaillierte Informationen zu den verwendeten Variablen sind im Programmcode kommentiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Grösse** | **Code-Bezeichner** | **Wert(bereich)** | **Anwendung / Beschreibung** |
|  |  |  |  |

## Grafische Lösung (Struktogramme)

# Source-Code

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. Firma : Siemens
3. Autoren : Plüss Thomas / Bösch Dario
4. Projekt : Lötübung 7-Segment
5. Version : 1.0
6. Dateiname : LoetUebung 7-Segment\_10Pin.c
7. Erstelldatum : 15.09.16
8. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
9. Chip type : ATmega2560
10. Clock frequency : 8.0000 MHz
11. Memory model : small
12. Internal RAM size : 8 kBytes
13. External RAM size : --
14. Data Stack size : --
15. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
16. Datum Vers. Kommentar / Änderungsgrund
17. 15.09.16 1.0 Erstellt
18. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
19. // Verkabelung
20. /\*
21. STK-600 7-Segment-Anzeige
22. ------- -----------------
23. PORTL Bit 0 --- X(10-pol.) Pin 4
24. PORTL Bit 1 --- X(10-pol.) Pin 5
25. PORTL Bit 2 --- X(10-pol.) Pin 6
26. PORTL Bit 3 --- X(10-pol.) Pin 7
27. PORTL Bit 4 --- X(10-pol.) Pin 3
28. PORTL Bit 5 --- X(10-pol.) Pin 2
29. PORTL Bit 6 --- X(10-pol.) Pin 1
30. PORTL Bit 7 --- X(10-pol.) Pin 8
31. X(PortL) Pin 9 --- X(10-pol.) Pin 9
32. X(PortL) Pin 10 --- X(10-pol.) Pin 10
33. PORTG Bit 0 --- X(12-pol.) Pin 11
34. \*/
35. // Definitionen für die Delayfunktion
36. #ifndef *F\_CPU*
37. /\* Definiere F\_CPU, wenn F\_CPU nicht bereits vorher definiert
38. (z.&nbsp;B. durch Übergabe als Parameter zum Compiler innerhalb
39. des Makefiles). Zusätzlich Ausgabe einer Warnung, die auf die
40. "nachträgliche" Definition hinweist \*/
41. #warning "F\_CPU war noch nicht definiert, wird nun mit 3686400 definiert"
42. #define *F\_CPU* 1000000UL /\* Quarz mit 8 Mhz \*/
43. #endif
44. /\*------------------------------- Einbindungen -------------------------------\*/
45. #include <avr/io.h>
46. #include <stdint.h>
47. #include <inttypes.h>
48. #include <util/delay.h>
49. /\*---------------------------- Globale Variablen -----------------------------\*/
50. /\*--------------------- Konstanten & Definitionen ----------------------------\*/
51. // Makros
52. #define SETBIT(Adr,Bit) (Adr |= (1<<Bit))
53. #define CLEARBIT(Adr,Bit) (Adr &= ~(1<<Bit))
54. #define CHECKBIT(Adr,Bit) (Adr & (1<<Bit))
55. #define INVERTBIT(Adr,Bit) (Adr ^= (1<<Bit))
56. // I/O's definieren
57. #define TASTER PINA
58. #define SCHALTER PINB
59. #define LED PORTC
60. #define SEGA PORTL
61. #define DOPP PORTG
62. /\*------------------------------- Funktionen ---------------------------------\*/
63. void Seg7\_aus(volatile *uint8\_t* \*Segment)
64. {
65. \*Segment = 0x10; // Zahl mit 0x10 verodern (0bxxx1xxxx)
66. }
67. void Seg7(unsigned char Zahl, volatile *uint8\_t* \*Segment) {
68. Zahl |= 0x30; // Zahl mit 0x30 verodern (0bxx11xxxx)
69. \*Segment = Zahl; // Zahl an 7-Segment ausgeben
70. // \_delay\_us(500);
71. // CLEARBIT (SEGA,5);
72. // \_delay\_ms(1);
73. }
74. /\*------------------------------- Hauptprogramm --------------------------------\*/
75. int main(void) {
76. DDRA = 0x00; // PORTA ist Eingang (Taster)
77. DDRC = 0xFF; // PORTD ist Ausgang (LEDs)
78. DDRL = 0xFF; // PORTL ist Ausgang (7-Segment-Anzeige Platine)
79. DDRG = 0xFF; // PORTG ist Ausgang (Doppelpunkt)
81. unsigned char zahl = 0;
82. unsigned int i = 0;
84. while (1) {
85. while (TASTER == 0x01) { //Zufallszahl 0..9 generieren während Tastendruck T0
86. Seg7\_aus(&SEGA); //Display ausschalten
87. SETBIT(SEGA,7); //Dezimalpunkt einschalten
88. SETBIT (DOPP,0); //Doppelounkt einschalten
89. ++zahl;
90. if (zahl == 10) zahl = 0;
91. }
92. if (SCHALTER == 0x01) { //Durchzählen, wenn S0 ein
93. for (i=0;i<10;i++)
94. {
95. LED = i;
96. Seg7(i, &SEGA);
97. *\_delay\_ms*(400);
98. }
99. }
100. else { //Ausgabe Zufallszahl (ohne Doppel- und Dezimalpunkt
101. LED = zahl;
102. CLEARBIT(DOPP,0);
103. Seg7(zahl, &SEGA);
104. }
105. }
106. }

# Testprotokoll

## Testablauf

|  |  |  |  |
| --- | --- | --- | --- |
| **Testfall** | **Vorgang** | **Erwartung** | **Befund** |
| Reset/Programmstart | Power On oder Reset | S0 Ein: Zähler zählt 0..9 S0 Aus: Anzeige = 0 | o.k. |
| Zufallszahl (S0 = Aus) | Tastendruck TS0 | Doppelpunkt wird angezeigt, Dezimalpunkt wird angezeigt | o.k. |
|  | Taste TS0 loslassen | Zufallszahl 0..9 wird angezeigt, Doppel- und Dezimalpunkt wird nicht angezeigt | o.k. |
| Zähler | Schalter S0 = Ein | Zähler zählt (endlos) von 0..9 | o.k. |
|  | Schalter S0 = Aus | Zähler zählt fertig bis 9… | o.k. |

## Testbericht mit Fazit und Massnahmen

--

Ort, Datum und Unterschrift:

Zug, 27. Oktober 2016, Thomas Plüss