**Speicherbereich auf LEDs**

Inhaltsverzeichnis

[1 Produktanforderungen 2](#_Toc495696328)

[2 Softwaredesign 3](#_Toc495696329)

[3 Speicher / Registerbelegung 4](#_Toc495696330)

[4 Berechnung der verwendeten Zeitschleife 4](#_Toc495696331)

[5 Programmlisting 5](#_Toc495696332)

[6 Testplan 6](#_Toc495696333)

[7 Probleme 6](#_Toc495696334)

[8 Erkenntnisse 7](#_Toc495696335)

[9 Zeitaufwand 7](#_Toc495696336)

10 Lastenheft

# Produktanforderungen

Es ist ein mit Unterprogrammen strukturiertes Programm für den Zilog Z80 Mikroprozessor zu schreiben:  
Auf der Adresse 1901 beginnt ein Speicherbereich, in dem abwechselnd auszugebende Bitmuster (Diese Werte werden vor Programmstart dort händisch eingegeben)  
Die Länge des Speicherbereiches (Anzahl der nun folgenden Bytepaare) steht in 1900h.  
Dauer Code: 1 – 8, d.h. das 1 bis 8 – fache der Grunddauer.  
Grunddauer = 0,333 sec  
Die Bitmuster sollen der jeweiligen Zeit entsprechend immer wieder an den LEDs angezeigt werden.  
Das Programm soll als Endlosprogramm realisiert werden.

**Schalter – Belegung:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S7** | **S6** | **S5** | **S4** | **S3** | **S2** | **S1** | **S0** |
| Hauptschalter | - | - | - | - | - | - | - |

**LED – Belegung:**

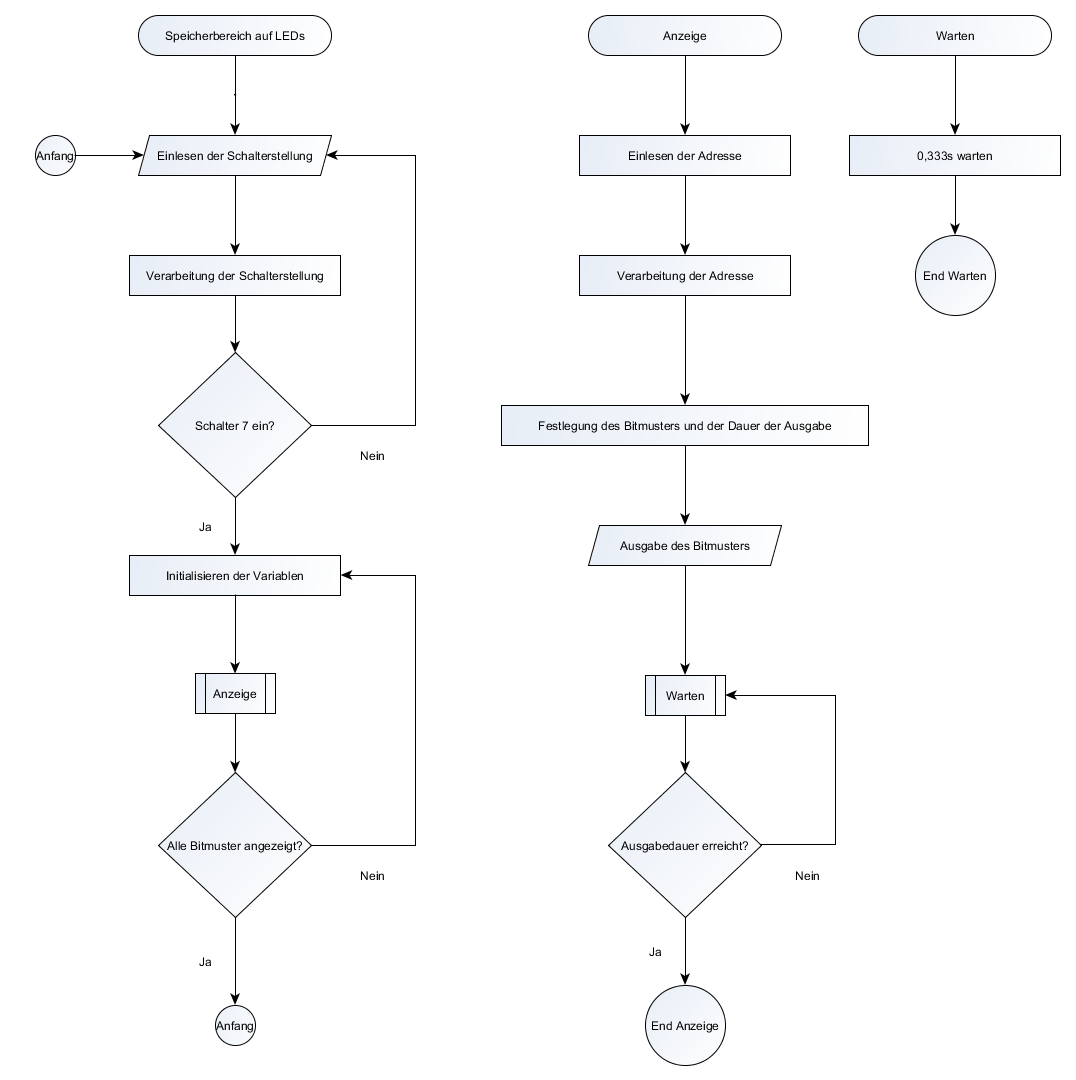
Es gibt keine LED – Belegung, da ein Bitmuster an die LEDs angezeigt wird.

Nach dem Einschalten des Hauptschalters soll die im Speicher 1901H beginnenden Bitmuster mit der jeweiligen Dauer ausgegeben werden.

Falls der Hauptschalter während dem Programmablauf ausgeschaltet wird, soll das Programm den derzeitigen Ablauf fertigstellen und dann den Ablauf stoppen.

Dabei ist nur der Hauptschalter wichtig, alle anderen werden nicht berücksichtigt.

# Softwaredesign



# Speicher / Registerbelegung

Der RAM Speicher wurde für den Programmcode und als Zwischenspeicher für B und C verwendet, da die Register für die Warteschleife benötigt wurden.  
(B auf der Adresse 18FF, C auf der Adresse 18FE)

Register:

|  |  |
| --- | --- |
| A | Ausgeben und Umladen von Registern |
| B | Anzahl von auszugebenden Bitmustern |
| C | Dauer der Ausgabe der Bitmuster |
| D | Doppelregister (2 Byte): Zähler |
| E |
| H | Doppelregister (2 Byte): Adressnummer |
| L |

I/O Einheiten:

LED / Schalter: Die LEDs bzw. Schalter werden mit der Adresse C0H angesprochen.

# Berechnung der verwendeten Zeitschleife

|  |  |  |  |
| --- | --- | --- | --- |
|  | Programmcode | Taktzyklen | Anzahl Durchläufe |
| Wait | … |  |  |
|  | … |  |  |
|  | LD B, #05 | 7 | 1 |
| loopB | LD A, #21 | 7 | 5 |
| loopA | LD C, #00 | 7 | 5\*33 |
| loopC | DEC C | 4 | 5\*33\*256 |
|  | JP NZ, loopC | 10 | 5\*33\*256 |
|  | DEC A | 4 | 5\*33 |
|  | JP NZ, loopA | 10 | 5\*33 |
|  | DEC B | 4 | 5 |
|  | JP NZ, loopB | 10 | 5 |
|  | … |  |  |
|  | … |  |  |
|  | RET | 10 | 1 |

1 Taktzyklus des Systems dauert 1 / 1,79MHz = 0,56µs

Dauer Warteschleife:   
0,56µs \* [7+(7\*5)+(7\*5\*33)+(4\*5\*33\*256)+(10\*5\*33\*256)+(4\*5\*33)+(10\*5\*33)+(4\*5)+(10\*5)+10]

= **333ms**

# Programmlisting

|  |  |
| --- | --- |
| Speicherbereich auf LEDs | |
| Simulation des Z80 | |
| Schalter: | Lampen: |
| 7: Hauptschalter | 7-0: Bitmuster |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Adresse** | **Op-Code (hex)** | **Symbol (Label)** | **Mnemonic** | **Kommentar** |
| Hauptprogramm | | | | |
| 1800 | DB C0 | Anfang | IN A, (C0) | Einlesen der Schalterstellung (C0) |
| 1802 | E6 80 |  | AND #80 | Bits gezielt auf 0 setzen |
| 1804 | CA 00 18 |  | JP Z, Anfang | S7 nicht gesetzt dann JP auf Anfang |
| 1807 | 11 00 00 |  | LD DE, #0000 | Zähler DE auf 0 setzen |
| 180A | 21 00 19 |  | LD HL, #1900 | HL auf die Adr. Num. 1900 setzen |
| 180D | 3A 00 19 |  | LD A, (1900) | Inhalt von 1900 in A speichern |
| 1810 | 47 |  | LD B, A | A umladen in B |
| 1811 | 13 | loop1 | INC DE | Zähler DE um 1 erhöhen |
| 1812 | CD 2A 18 |  | CALL Leucht | UP - Aufruf Leucht |
| 1815 | 05 |  | DEC B | B um 1 erniedrigen |
| 1816 | C2 12 18 |  | JP NZ, loop1 | B nicht 0 dann JP auf loop1 |
| 1819 | C3 00 18 |  | JP Anfang | unbedingt auf Anfang springen |
| Adresslücke (181A - 1829) | | | |  |
|
| Leucht | | | | |
| 182A | 19 | Leucht | ADD HL, DE | Aktuelle Adresse um DE erhöhen |
| 182B | 7E |  | LD A, (HL) | Inhalt des Ergebnisses in A speichern |
| 182C | 23 |  | INC HL | Aktuelle Adresse um 1 erhöhen |
| 182D | 4E |  | LD C, (HL) | Inhalt der Adresse in C speichern |
| 182E | D3 C0 |  | OUT (C0, A) | Ausgabe von A auf Anzeige C0 |
| 1830 | 78 |  | LD A, B | B umladen in A |
| 1831 | 32 FF 18 |  | LD (18FF), A | A auf Adresse zwischenspeichern |
| 1834 | CD 5A 18 | loop2 | CALL Wait | UP - Aufruf Wait |
| 1837 | 0D |  | DEC C | Ausgabedauer um 1 erniedrigen |
| 1838 | C2 34 18 |  | JP NZ, loop2 | C nicht 0, dann JP auf loop2 |
| 183B | 3A FF 18 |  | LD A, (18FF) | A aus dem Zwischenspeicher holen |
| 183E | 47 |  | LD B, A | A umladen in B |
| 183F | C9 |  | RET | Zurück ins HP springen |
| Adresslücke (1840 - 1859) | | | |  |
|
| Warteschleife | | | | |
| 185A | 79 | Wait | LD A, C | Ausgabedauer C in A umladen |
| 185B | 32 FE 18 |  | LD (18FE), A | A auf Adresse zwischenspeichern |
| 185E | 06 05 |  | LD B, #05 | B auf 5 setzen |
| 1860 | 3E 21 | loopB | LD A, #21 | A auf 33 setzen |
| 1862 | 0E 00 | loopA | LD C, #00 | C auf 0 setzen |
| 1864 | 0D | loopC | DEC C | C um 1 erniedrigen |
| 1865 | C2 64 18 |  | JP NZ, loopC | C nicht 0, dann JP auf loopC |
| 1868 | 3D |  | DEC A | A um 1 erniedrigen |
| 1869 | C2 62 18 |  | JP NZ, loopA | A nicht 0, dann JP auf loopA |
| 186C | 05 |  | DEC B | B um 1 erniedrigen |
| 186D | C2 60 18 |  | JP NZ, loopB | B nicht 0, dann JP auf loopB |
| 1870 | 3A FE 18 |  | LD A, (18FE) | A aus dem Zwischenspeicher holen |
| 1873 | 4F |  | LD C, A | A in Ausgabedauer C umladen |
| 1874 | C9 |  | RET | Zurück ins UP Leucht |

# Testplan

|  |  |  |
| --- | --- | --- |
| **Schalter / Dauer – Code** | **Wirkung** | **Anmerkung** |
| Hauptschalter (S7) ein | Bitmuster werden ausgegeben | Je nach angegebenen Dauer – Code |
| Hauptschalter (S7) aus | Ablauf der Ausgabe der Bitmuster wird gestoppt | Nach Ablauf der aktuellen Bitmusterreihe, hält die Anzeige beim letzten Bitmuster |
| Beliebiger Schalter an oder aus | Keine Wirkung | Schalterstellung wird nicht berücksichtigt |
| Dauer – Code: 0 | 0 \* 0,333 = 0 | Keine Wartezeit |
| Dauer – Code: 1 | 1 \* 0,333 = 0,333 | Wartezeit = 0,333s |
| Dauer – Code: 2 | 2 \* 0,333 = 0,666 | Wartezeit = 0,666s |
| Dauer – Code: 3 | 3 \* 0,333 = 0,999 | Wartezeit = 0,999s |
| Dauer – Code: 4 | 4 \* 0,333 = 1,332 | Wartezeit = 1,332s |
| Dauer – Code: 5 | 5 \* 0,333 = 1,665 | Wartezeit = 1,665s |
| Dauer – Code: 6 | 6 \* 0,333 = 1,998 | Wartezeit = 1,998s |
| Dauer – Code: 7 | 7 \* 0,333 = 2,331 | Wartezeit = 2,331s |
| Dauer – Code: 8 | 8 \* 0,333 = 2,664 | Wartezeit = 2,664s |
| Dauer – Code: Faktor k > 8 | k \* 0,333 = k \* 0,333 | Wartezeit = k \* 0,333s |