

Frei programmierbare
4-8 Digit / Bargraph LED-Anzeige
für μ Controlleranwendungen

Rainer Rick
Tholeyer Straße 28
66606 St. Wendel
Deutschland
Email: IB_Rick@yahoo.de

Inhalt

Einleitung	2
Datenformat	2
Erklärung der einzelnen Funktionen (Adressen)	3
No-Operation (Adresse 00)	3
Digit 0 - Digit 7 (Adressen 01 - 08)	3
Dekodiermodus (Adresse 09)	3
Helligkeit (Adresse 0A)	4
Anzahl der Digits (Adresse 0B)	4
Stromsparfunktion (Adresse 0C)	4
Selbsttest (Adresse 0F)	4
Erläuterungen zum Programm	5
Erläuterungen zum Schaltplan	5
Nachbau	5
Rechtliches	5
Quellcode in Batronix-Assembler V4.22	6

Einleitung

Viele Anwendungen verlangen eine (auch im Dunkeln) gut ablesbare Anzeige, die sich leicht aus 7-Segmentanzeigen realisieren lässt.

Hat man genügend freie Portleitungen zur Verfügung, ist die preisgünstigste Möglichkeit sicher, die Segmentleitungen parallel und den gemeinsamen Anschluss im Multiplexbetrieb anzusteuern. Ich verwende in meinen Schaltungen die integrierte Lösung, den MAX 7219 von MAXIM. Dieser ermöglicht die Ansteuerung von bis zu acht Digits oder 64 Einzel-LEDs.

Auch ein gemischter Betrieb ist problemlos möglich, zum Beispiel eine Meßgeräteanzeige mit 5 Digits Auflösung und zusätzlich 24 LEDs Bargraph-Anzeige.

Datenformat

Kontrolliert wird der Chip über eine SPI-kompatible serielle Schnittstelle, das heißt, man benötigt nur drei Portleitungen (ChipSelect, DATA, CLK). Das Timing der Leitungen ist im Datenblatt nachzulesen. Der Baustein erwartet jeweils eine 16 Bit breite Information, wobei die ersten acht Bit das entsprechende Register im Chip adressieren und das zweite Byte den eigentlichen Befehl darstellt:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	Adresse				MSB	Befehl						LSB

Auch wenn ein Blick ins Datenblatt immer anzuraten ist, möchte ich an dieser Stelle dem "Nicht-Englisch Sprechenden" das Verständnis erleichtern und die einzelnen Adressen und Befehle vorstellen:

Der MAX 7219 hat insgesamt 13 Adressregister:

Register	HEX-Code der Adresse (Bits D8-D15)
No-Operation	00
Digit 0	01
Digit 1	02
Digit 2	03
Digit 3	04
Digit 4	05
Digit 5	06
Digit 6	07
Digit 7	08
Dekodiermodus (Decode mode)	09
Helligkeit (Intensity)	0A
Anzahl der Digits (Scan limit)	0B
Stromsparfunktion (Shutdown)	0C
Selbsttest (Display Test)	0F

Anm.: Die obersten vier Bits der Adresse (D12-D15) sind nicht relevant und hier mit '0' belegt.

Erklärung der einzelnen Funktionen (Adressen)

No-Operation (Adresse 00)

Wird nur benötigt, wenn mehrere gleiche Chips kaskadiert werden. Dazu werden die ChipSelects und CLKs zusammengeschaltet und der DIN-Pin des nachfolgenden ICs mit dem DOUT des vorgeschalteten verbunden. Da ja nun bei x-ICs ein x*16 Bit langer Programmiercode erforderlich ist, werden alle Chips, die NICHT reagieren sollen, mit einer No-Operation-Adresse bedacht.

Digit 0 - Digit 7 (Adressen 01 - 08)

Über diese Adressen werden die einzelnen Digits angesprochen.

Dekodiermodus (Adresse 09)

Der Dekodiermodus entscheidet, ob der Inhalt des Datenbytes eines Digits als BCD-Zahl (0-9, E,H,L,P und '-') dekodiert wird, also ob zum Beispiel das Datenbyte '7' auch als '7' auf der Anzeige erscheint. Werden statt der 7-Segment Anzeigen einzelne LEDs angeschlossen (zum Beispiel bei Bargraph), so darf nicht dekodiert werden !

Die Dekodierung ist für jedes Digit einzeln einzustellen, dabei steht eine '1' für eingeschaltete Dekodierung:

Dekodiermodus	Datenbyte								HEX
	Bit D7	Bit D6	Bit D5	Bit D4	Bit D3	Bit D2	Bit D1	Bit D0	
Keine Dekodierung für Digits 7-0	0	0	0	0	0	0	0	0	00
Dekodierung für Digit 0 Keine Dekodierung für Digits 7-1	0	0	0	0	0	0	0	1	01
Dekodierung für Digit 0, 2 Keine Dekodierung für Digits 7-3, 1	0	0	0	0	0	1	0	1	05
Dekodierung für Digits 3-0 Keine Dekodierung für Digits 7-4	0	0	0	0	1	1	1	1	0F
Dekodierung für alle Digits	1	1	1	1	1	1	1	1	FF

Eng mit dem Dekodiermodus hängt nun das Ansprechen der entsprechenden Segmente der 7-Segment Anzeige abhängig vom Inhalt des Datenbytes zusammen. Das Bit D7 wird benutzt, um den Dezimalpunkt (DP) ein- oder auszuschalten ('1' = an). Es gilt die folgende Zuordnung:

7-Segmentanzeige	Dekodiermodus	Datenbyte							
		Bit D7	Bit D6	Bit D5	Bit D4	Bit D3	Bit D2	Bit D1	Bit D0
0	an	X	0	0	0	0	0	0	0
1	an	X	0	0	0	0	0	0	1
2	an	X	0	0	0	0	0	1	0
3	an	X	0	0	0	0	0	1	1
4	an	X	0	0	0	0	1	0	0
5	an	X	0	0	0	0	1	0	1
6	an	X	0	0	0	0	1	1	0

7	an	X	0	0	0	0	1	1	1
8	an	X	0	0	0	1	0	0	0
9	an	X	0	0	0	1	0	0	1
-	an	X	0	0	0	1	0	1	0
E	an	X	0	0	0	1	0	1	1
H	an	X	0	0	0	1	1	0	0
L	an	X	0	0	0	1	1	0	1
P	an	X	0	0	0	1	1	1	0
blank (kein Segment an)	an	X	0	0	0	1	1	1	1

Möchte man jedoch einen Buchstaben darstellen, der nicht im BCD Code (0-9, E,H,L,P,'-') liegt, muss man ihn sich selbst zusammenstellen, beispielsweise wird oft ein 'U' für Spannung gewünscht. Die Bezeichnung der einzelnen Segmente ist mit A-G festgelegt, ein U bedeutet somit, dass die Segmente B,C,D,E und F eingeschaltet werden müssen. Dieses Beispiel ist in der letzten Zeile der Tabelle angegeben. Gleiche Überlegungen muss man sich bei der Planung einer Bargraph-Anzeige bzw. der Einzel-LED-Ansteuerung machen:

	Dekodiermodus	Daten							
		Bit D7	Bit D6	Bit D5	Bit D4	Bit D3	Bit D2	Bit D1	Bit D0
Segment (LED)	aus	DP	A	B	C	D	E	F	G
z.B. U (siehe Text)	aus	0	0	1	1	1	1	1	0

Helligkeit (Adresse 0A)

Die Helligkeit ist in 15 Stufen einstellbar, eine '00' im Datenbyte entspricht der niedrigsten, eine '0F' entspricht der höchsten Helligkeit. Dabei wird der Strom moduliert, der mit dem Widerstand am Pin ISET vorgegeben ist.

Anzahl der Digits (Adresse 0B)

Damit der interne Multiplexer korrekt arbeitet, muss dem Chip die Anzahl der angeschlossenen Digits oder LEDs bekannt sein. Nur ein angeschlossenes Digit entspricht dann einer '00' im Datenbyte, alle acht möglichen entsprechen einer '07". Man sollte jedoch beachten, dass der Chip mit mindestens vier Segmenten betrieben wird, ansonsten müssen wesentliche Veränderungen im Schaltungsaufbau vorgenommen werden (siehe dazu unbedingt das Datenblatt !).

Stromsparfunktion (Adresse 0C)

Eine '01' im Datenbyte lässt den Chip normal arbeiten, eine '00' leitet den Shutdown ein.

Selbsttest (Adresse 0F)

Eine '00' im Datenbyte lässt den Chip normal arbeiten, eine '01' leitet den Selbsttest ein, das heißt alle LEDs werden eingeschaltet, unabhängig vom Inhalt der Digit-Datenregister.

Erläuterungen zum Programm

Das angefügte Beispielprogramm enthält die Routinen LED_init, LED_clr und LED_out.

Die LED_init Routine bedient sich der zuvor beschriebenen Funktionen, um

- die Anzahl der Digits auf fünf einzustellen,
- Dekodierung für alle Digits einzuschalten,
- und normale Operation bei geringster Helligkeit einzuschalten.

In der LED_clr Routine wird allen Digits ein '0F' übermittelt, der Wert für 'blank'.

Die LED_out Routine schließlich übermittelt die 16 Bits an den Chip, wobei die Adresse (also das erste Byte) im Register R1 und das Datenbyte im Register R0 stehen muss.

Erläuterungen zum Schaltplan

Der Schaltplan ist sehr allgemein gehalten und sollte ohne Schwierigkeiten auf die eigene Anwendung angepasst werden können. Wie erwähnt, können bis zu acht 7-Segment anzeigen (gemeinsame Kathode) mit dem Chip angesteuert werden. Der maximale Segmentstrom wird durch den Widerstand an ISET bestimmt, der sich wie im Datenblatt beschrieben, berechnen lässt. Für 4-5 Digits sind die hier verwendeten 10kOhm ein guter Wert.

Der verwendete Port ist durchgeschleift, so dass sich weitere Anwendungen direkt wieder einstecken lassen. Sollen weitere MAX 7219 kaskadiert werden, so kann der DOUT Pin über den Schalter kontaktiert werden.

Nachbau

Ich wünsche Ihnen viel Spaß und gutes Gelingen bei der Integration der vorgestellten Schaltung und deren Programmierung. Der Chip ist zwar mit ca. DM 20,- nicht gerade eine günstige Alternative, jedoch macht die vielfältige Einsetzbarkeit gerade bei vielen Digits oder Einzel-LEDs den Schaltkreis interessant. MAXIM ist bekannt für seinen schnellen und kostenlosen Musterversand - Für die ersten Versuche sicherlich ausreichend !

Sollten noch Fragen offen sein, so werde ich gerne versuchen, Ihnen diese zu beantworten - Schreiben Sie mir einfach eine Email ! Die Softwareeroutinen sind nur als 'Einstiegswerkzeug' zu verstehen, zur Integration in eigene Programme muss man sie meistens anpassen.

Rechtliches

Die dargestellten Schaltungen und Routinen sind nach bestem Wissen getestet worden - Eine Garantie - gleich welcher Art - kann und werde ich dafür nicht übernehmen. Alle Informationen entstammen dem Datenblatt von MAXIM, das dem Interessenten auf der Internet-Site www.maxim-ic.com zum Download angeboten wird.

St. Wendel, März 2001

Quellcode in Batronix-Assembler V4.22

```

; *****
; ***          MAX7219 Displaytreiber-Ansteuerung          ***
; *** Die Programmierung des Bausteins erfolgt seriell lt. Datenblatt ***
; ***          (c) by Rainer Rick 3/2001          ***
; *****

```

; Annahmen:

; Quartzfrequenz: 12 MHz

; Benutzt werden die Portleitungen (siehe 'Konstanten'):

; P1.0 = serial Clock (HEX 90)

; P1.1 = ChipSelect (HEX 91)

; P1.2 = serial Data (HEX 92)

; Es werden die Register R0 und R1, sowie die RAM-Adressen 20h und

; 21h benutzt, dabei steht in R1 später die im MAX7219 anzusprechende

; Adresse und in R0 das entsprechende Datenbyte.

Port_Init:

MOV Port1,#FF

LCALL LED_init

```

; *****
; ***          Hauptprogramm          ***
; *****
; *** Zum Test des Displays werden alle Digits nacheinander hoch- ***
; *** gezählt, hier bis 99999, um dann wieder bei 00000 zu beginnen. ***
; *****

```

Programm:

LCALL LED_clr

MOV R1,#1

For R0 = #0 To #9

LCALL LED_out

LCALL Pause

Next R0

MOV R1,#2

For R0 = #0 To #9

LCALL LED_out

LCALL Pause

Next R0

MOV R1,#3

For R0 = #0 To #9

LCALL LED_out

LCALL Pause

Next R0

MOV R1,#4

For R0 = #0 To #9

```

    LCALL LED_out
    LCALL Pause
Next R0

```

```

MOV R1,#5
For R0 = #0 To #9
    LCALL LED_out
    LCALL Pause
Next R0

```

```

LCALL Pause

```

```

LJMP Programm          ;Endlosschleife
;Ende Hauptprogramm

```

```

; *****
; ***                Unterprogramm LED_clr                ***
; *****
; ***      Jedes Digit wird definiert zurückgesetzt (blank)      ***
; *****

```

```

LED_clr:

```

```

MOV R1,#01          ;Digit 1
MOV R0,#0F
LCALL LED_out

```

```

MOV R1,#02          ;Digit 2
MOV R0,#0F
LCALL LED_out

```

```

MOV R1,#03          ;Digit 3
MOV R0,#0F
LCALL LED_out

```

```

MOV R1,#04          ;Digit 4
MOV R0,#0F
LCALL LED_out

```

```

MOV R1,#05          ;Digit 5
MOV R0,#0F
LCALL LED_out
RET

```

```

; ****
; ***
; ****
; **** Der Baustein MAX7219 wird konfiguriert (s.a. Datenblatt): ****
; ****   Adr.:  0B: Anzahl der verwendeten Digits          ****
; ****           09: Art der Dekodierung                  ****
; ****           0A: Helligkeit der Anzeige               ****
; ****           0C: Operation (Normalbetrieb, Sleep, Self-Test) ****
; ****

```

LED_init:

```

MOV R1,#0B      ;Scan Limit = 5 Digits
MOV R0,#04
LCALL LED_out

```

```

MOV R1,#09      ;Decode BCD Code
MOV R0,#1F
LCALL LED_out

```

```

MOV R1,#0A      ;Intensity
MOV R0,#01
LCALL LED_out

```

```

MOV R1,#0C      ;Normal Operation
MOV R0,#01
LCALL LED_out
RET

```

```

; ****
; ***
; ****
; **** Adresse und Befehl werden seriell ausgegeben ****
; ****

```

LED_out:

```

CLR ChipSelect ;active LOW !
MOV A,#80
MOV R2,A
;HighByte ausgeben
For Zähler = #0 to #07

```

```

    ANL A,R1
    JNZ DHigh
    CLR Data
    LJMP weiter

```

```

DHigh:
    SETB Data

```

```

weiter:
    SETB Clock
    CLR Clock
    MOV A,R2
    RR A

```

```

MOV R2,A
Next Zähler

```

```

MOV A,#80
MOV R2,A

```

```

;LowByte ausgeben

```

```

For Zähler = #0 to #07
  ANL A,R0
  JNZ DHigh2
  CLR Data
  LJMP weiter2
DHigh2:
  SETB Data
weiter2:
  SETB Clock
  CLR Clock
  MOV A,R2
  RR A
  MOV R2,A

```

```

Next Zähler
SETB ChipSelect
RET

```

```

; *****
; ***               Unterprogramm Pause               ***
; *****
; ***               Beliebige Pausenroutine           ***
; *****

```

```

Pause:

```

```

For Zähler = #0 To #8F
  For Zähler2 = #0 To #FF
    NOP
  Next Zähler2
Next Zähler
RET

```