
LED-Dot-Matrix

Wenn man eine etwas andere Anzeige für die Attiny-Platine sucht, sollte man in Erwägung ziehen, eine oder mehrere LED-Dot-Matrix-Elemente anzuschaffen. Versehen mit einer geeigneten Elektronik (MAX 7219) lassen sie sich leicht über SPI ansteuern und sind - zumindest als Bausatz - recht preiswert. Module mit 8 mal 8 LEDs sind so schon für unter 5 Euro zu erwerben (Abb. 1).

Die wenigen Bauteile sind rasch auf die Platine gelötet - auch ohne Anleitung. Lediglich beim Einstecken der LED-Matrix in die Buchsenleisten ist die Orientierung nicht aus der Platinenbeschriftung zu erkennen. Recherchen im Internet veranlassten mich, die LED-Matrix mit der beschrifteten Seite nach vorne zu montieren. Und damit funktioniert das Modul auch einwandfrei.

Die LEDs der Matrix werden hier nicht direkt vom Mikrocontroller angesteuert. Vielmehr geschieht dies über einen speziellen IC, den MAX 7219. Dieser Baustein ist speziell zum Betrieb von Anzeigeeinheiten wie z. B. mehrstellige 7-Segment-Anzeigen oder LED-Dot-Matrix-Elementen konzipiert. Gesteuert wird dieser IC über 2-Byte-Befehle. Diese Bytes werden über SPI übertragen.

Folgende Verdrahtung ist erforderlich, damit die weiter unten angegebenen Programmteile funktionieren können: Neben der elektrischen Versorgung muss der Eingang DIN des Moduls mit PB6, der Taktsignaleingang CLK mit PB7 und der Chip-Select-Eingang CS mit PB0 verbunden werden. Der CS-Eingang könnte auch mit anderen Ports verbunden werden; will man auf die im Attiny2313 fest verdrahteten USI-Funktionen zurückgreifen, dann muss die SPI-Kommunikation über PB6 und PB7 laufen.

Eine geeignete SPI-Funktion, die einzelne Bytes an einen Slave sendet, besteht nur aus wenigen Zeilen:

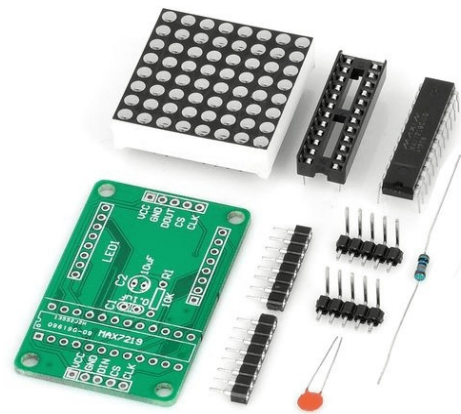


Abbildung 1

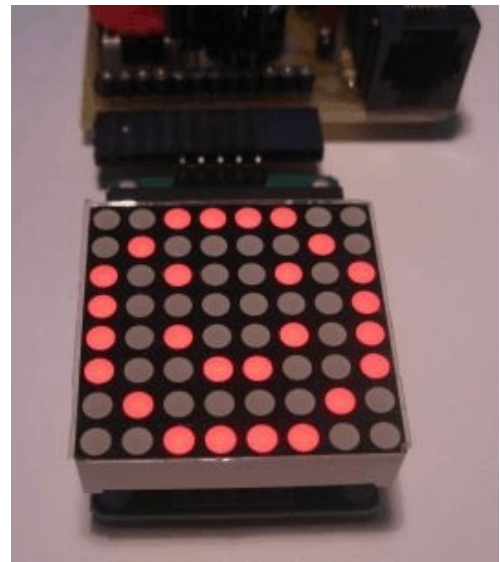


Abbildung 2

LED-Dot-Matrix

```
Function Spi(s As Byte) As Byte      'Hardware-SPI mit USI
  Ddrb = &B11011111
  Usidr = S
  Usisr = &B01000000                'USIOIF und Zähler löschen
  Usicr = &B00011010                'USIWM0, USICS1,USICLK auf 1
  Do
    Usicr.usitc = 1                  'Takten durch Setzen von usitc
  Loop Until Usisr.usioif = 1        '16 mal toggeln
  Spi = Usidr
End Function
```

Diese Funktion empfängt auch gleichzeitig Daten über PB5; diese Option wird im Folgenden aber nicht genutzt. Ausführlichere Erläuterungen zur SPI-Kommunikation mit dem Attiny2313 findet man unter <http://www.forum.g-heinrichs.de/viewtopic.php?f=12&t=49>.

Wie sehen nun die 2-Byte-Kommandos aus, mit denen das Dot-Matrix-Modul gesteuert wird? Im Wesentlichen bestehen sie aus einer Registeradresse und einem Datenbyte. Die Register stehen für verschiedene Funktionen. Diese sind in der folgenden Tabelle zusammengefasst.

Adresse (dezimal)	Bedeutung
0	No Operation
1	Digit 0 (Spalte 1)
2	Digit 1 (Spalte 2)
3	Digit 2 (Spalte 3)
4	Digit 3 (Spalte 4)
5	Digit 4 (Spalte 5)
6	Digit 5 (Spalte 6)
7	Digit 6 (Spalte 7)
8	Digit 7 (Spalte 8)
9	Decode Mode
10	Intensity
11	Scan Limit
12	Shut Down
15	Display Test

Wir wollen z. B. in der 3. Spalte der Matrix (von rechts gesehen) die 3., 5. und 6. LED (von unten gezählt) zum Leuchten bringen. Diese Situation ist in Abb. 2 dargestellt.

LED-Dot-Matrix

Dazu deuten wir das LED-Muster in Abb. 2 als Bitmuster einer Zahl. Der zugehörige dezimale Wert ist

$$4 + 16 + 32 = 52.$$

Die 3. Spalte wird durch das Digit-2-Register mit der Adresse 3 kontrolliert. Deswegen muss dieser Wert 52 nun in das Register 3 des MAX-Bausteins übertragen werden. Dazu werden folgende Schritte ausgeführt:

1. MAX-Baustein aktivieren (CS auf Low)
2. Den Wert 3 mit der SPI-Funktion an den MAX-Baustein übertragen (Kommando bzw. Registeradresse)
3. Den Wert 52 mit der SPI-Funktion an den MAX-Baustein übertragen (Parameterwert)
4. MAX-Baustein deaktivieren (CS auf High)

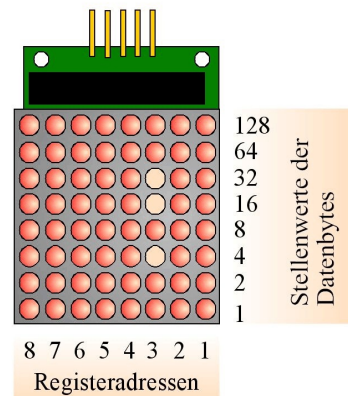


Abbildung 3

Diese Schrittfolge fast man sinnvollerweise zu einer Prozedur zusammen:

```
Sub Matrix_command(b As Byte , D As Byte)
  Portb.0 = 0
  Waitus 1
  Dummy = Spi(b)
  Dummy = Spi(d)
  Portb.0 = 1
  Waitus 1
End Sub
```

Das LED-Muster aus der Abb. 2 kann damit auf einfache Weise erzeugt werden:

```
befehl = 3
wert = 52
call matrix_command(befehl, wert)
```

Wir verzichten hier übrigens auf eine call-by-value-Referenzierung, weil dies den SRAM-Speicher des Attinys belastet und gegebenenfalls zu einem Stapelüberlauf und damit zu merkwürdigen Effekten führen könnte.

Die restlichen Spalten werden natürlich in analoger Weise übertragen. Es ist klar: Die LEDs lassen sich immer nur spaltenweise ansprechen.

Um das Dot-Matrix-Modul erfolgreich betreiben zu können, muss man nicht alle Befehle des MAX 7219 kennen. Die folgenden Befehle sind aber wichtig:

LED-Dot-Matrix

1. Shut down

Wenn das Modul eingeschaltet worden ist, befindet es sich zunächst im Shut-Down-Modus. Um das Modul aufzuwecken, muss der Wert 1 in das Shut-Down-Register geschrieben werden.

2. Scan Limit

Über dieses Register stellen wir die Anzahl der Spalten ein, die von der Matrix maximal angezeigt werden sollen. In unserem Fall müssen wir den Wert 8 in das Scan-Limit-Register schreiben.

3. Display Test

Mit diesem Register können wir die LEDs der Matrix testen: Schreiben wir den Wert 1 in dieses Register, gehen alle LEDs an. Schreiben wir den Wert 0 in das Register, wird das Muster angezeigt, welches zuvor bestanden hatte, angezeigt. Die Werte in den Registern 1 bis 8 werden durch den Test also nicht gelöscht und durch das Beenden des Test wieder zur Anzeige gebracht.

4. Intensity

Mithilfe dieses Registers kann die Helligkeit der LEDs eingestellt werden. Zulässig sind Werte von 0 (minimale Helligkeit) bis 15 (maximale Helligkeit). Die Helligkeit lässt sich nicht für jede LED gesondert einstellen. Vielmehr bezieht sich der Helligkeitswert immer auf alle eingeschalteten LEDs.

Das LED-Dot-Matrix-Modul kann ggf. Den Upload-Vorgang stören. Es empfiehlt sich deswegen, vor dem Upload die Verbindung zu Vcc zu unterbrechen und nach dem Upload die Attiny-Platine auszuschalten, die Verbindung zu Vcc wieder herzustellen und zuletzt die Attiny-Platine wieder einzuschalten.

Test-Programme und ein Video findet man unter:

<http://www.forum.g-heinrichs.de/download/file.php?id=104>