# 5x7-Dot-Matrix-Ansteuerung mit dem MAX7219

(mit PIC-Mikrocontroller)

Autor: Buchgeher Stefan

Letzte Bearbeitung: 25. September 2008

# Inhaltsverzeichnis

# 1. Einleitung

Der integrierte Baustein MAX7219 (der Fa. Maxim) wurde für die Ansteuerung mehrerer 7-Segment-Anzeigen entwickelt. Er eignet sich aber auch sehr gut für die Ansteuerung einer 5x7-Dot-Matrix bzw. einer 8x8-Dot-Matrix. Ich beschränke mich hier aber auf die 5x7-Dot-Matrix.

Durch Hintereinanderschalten (kaskadieren) mehrer MAX7219 können auch größerer Anzeigen realisiert werden.

Als Programmiersprachen für die PIC-Software wurden Assembler und C (mit dem mikroC-Compiler[1]) verwendet.

# 2. Hardware

Bild 2.1. zeigt das Prinzipschaltbild für die Ansteuerung einer Dot-Matrix-Anzeige mit Hilfe des MAX7219.

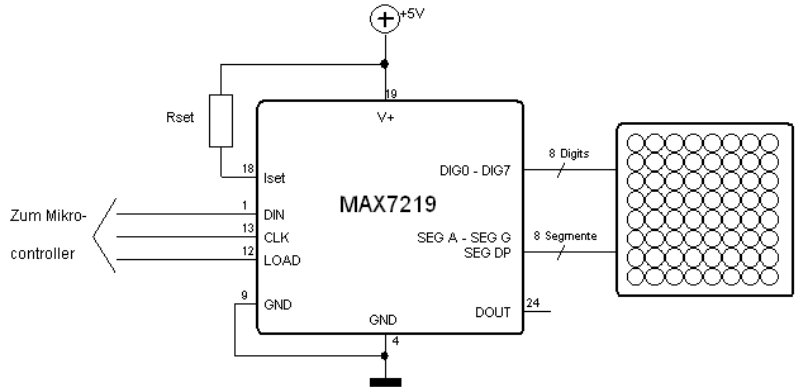



Bild 2.1.: Prinzipschaltbild für eine Dot-Matrix-Anzeige am MAX7219

Die Dot-Matrix-Anzeige ist mit dem MAX7219 über die Digits (DIG0 bis DIG7) und Segment (SEG A bis SEG G und SEG DP) verbunden, wobei die Digits mit den Spalten der Dot-Matrix-Anzeige, und die Segmente mit den Zeilen der Dot-Matrix-Anzeige verbunden werden. (siehe auch Demonstrationsbeispiel 1 im Abschnitt 5.)

Die Kommunikation des MAX7219 mit einem Mikrocontroller erfolgt mit den 3 Portleitungen DIN (Daten, Pin 1), CLK (Takt, Pin 13) und LOAD (Übernehmen, Pin 12). Eine genauere Beschreibung erfolgt im Abschnitt 3 (Ansteuerung des MAX7219).

Am Ausgang DOUT (Pin 24) kann der Eingang DIN (Pin 1) eines weiteren MAX7219 angeschlossen werden. Auf diese Weise können mehrere MAX7219 kaskadiert (also hintereinander geschaltet) werden[2].

Der Widerstand Rset bestimmt die maximale Helligkeit der Anzeige.

[1] Der mikroC-Compiler wurde von der Fa. mikroElektronika (www.mikroe.com/) entwickelt und dient zum Programmieren der PIC16Fxx-Familie in der Hochsprache C. Die freie Demoversion ist nur auf eine Programmspeichergröße von 2k begrenzt, ansonst voll kompatibel zur Vollversion.

[2] In diesem Fall weicht das Übertragungsprotokoll, welches auf den nächsten Seiten beschrieben ist, ab, sodass die hier beschriebenen Unterprogramme (Abschnitt 4) dann nicht funktionieren. Die Abschnitte 7 bis 9 (Demonstrationsbeispiel 3) zeigen, wie man mehrere MAX7219 hintereinander schaltet (kaskadiert).

# 3. Ansteuerung des MAX7219

Die Konfiguration des MAX7219 und die Ansteuerung der einzelnen Digits (hier der einzelnen Leuchtdioden der 5x7-Dot-Matrix) erfolgt mit 16 Registern. Diese 16 Register sind nach folgendem Schema aufgebaut:

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | ADDRESS | | | | MSB | | | DATA | | | | LSB |

Tabelle 3.1.: Registerformat (16-Bit)

Jedes Register besteht aus einer 8-Bit-Adresse (wobei nur die Bits D8 bis D11 von Bedeutung sind) und aus 8-Bit-Daten.

Die Adresse gibt sozusagen an, welches Register angesprochen werden soll. Die folgende Tabelle zeigt die Bedeutung der einzelnen Register.

| REGISTER | ADDRESS | | | | | HEX CODE |
|---|---|---|---|---|---|---|
| | D15–D12 | D11 | D10 | D9 | D8 | |
| No-Op | X | 0 | 0 | 0 | 0 | 0xX0 |
| Digit 0 | X | 0 | 0 | 0 | 1 | 0xX1 |
| Digit 1 | X | 0 | 0 | 1 | 0 | 0xX2 |
| Digit 2 | X | 0 | 0 | 1 | 1 | 0xX3 |
| Digit 3 | X | 0 | 1 | 0 | 0 | 0xX4 |
| Digit 4 | X | 0 | 1 | 0 | 1 | 0xX5 |
| Digit 5 | X | 0 | 1 | 1 | 0 | 0xX6 |
| Digit 6 | X | 0 | 1 | 1 | 1 | 0xX7 |
| Digit 7 | X | 1 | 0 | 0 | 0 | 0xX8 |
| Decode Mode | X | 1 | 0 | 0 | 1 | 0xX9 |
| Intensity | X | 1 | 0 | 1 | 0 | 0xXA |
| Scan Limit | X | 1 | 0 | 1 | 1 | 0xXB |
| Shutdown | X | 1 | 1 | 0 | 0 | 0xXC |
| Display Test | X | 1 | 1 | 1 | 1 | 0xXF |

Tabelle 3.2.: Registerzuordnung

Register **No-Op** (Adresse: 00h):
Dieses Register wird benötigt, wenn mehrere MAX7219 kaskadiert (also hintereinander geschaltet) werden. Mehr dazu ab Abschnitt 7.

Register **Digit 0** bis **Digit 7** (Adressen: 01h bis 08h):
Ist eine 5x7-Dot-Matrix-Anzeige z.B. nach Bild 3.1. angeschlossen und es soll der Buchstabe „**A**“ angezeigt werden, so müssen die Register 01h (Digit 0) bis 05h (Digit 4) mit den folgenden Werten geladen werden:

- Digit 0 (1. Spalte): 2 + 4 + 8 + 16 + 32 + 64 = 126
- Digit 1 (2. Spalte): 1 + 16 = 17
- Digit 2 (3. Spalte): 1 + 16 = 17
- Digit 3 (4. Spalte): 1 + 16 = 17
- Digit 4 (5. Spalte): 2 + 4 + 8 + 16 + 32 + 64 = 126

Anmerkungen:

- Diese Werte ergeben sich, da die Wertigkeit der Zeilen von oben nach unten zunimmt. Zeile 1 (Segment G) besitzt die Wertigkeit 1 (= $2^0$), Zeile 2 (Segment F) die Wertigkeit 2 (=$2^1$), Zeile 3 (Segment E) die Wertigkeit 4 (=$2^2$) usw. bis zur 7ten Zeile (Segment A) mit der Wertigkeit 64 (=$2^7$). Für jede Spalte müssen nun die Wertigkeiten jeder Zeilen welche aufleuchten sollen addiert werden.
- Die Digits 5 bis 7 werden hier nicht verwendet. Es ist daher auch nicht notwendig diese Register mit Werten zu laden.

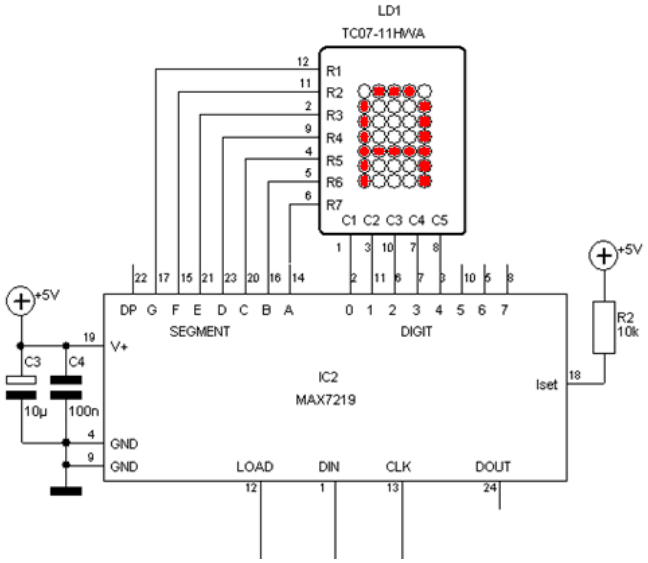



Bild 3.1.: Anschlussbeispiel für eine 5x7-Dot-Matrix-Anzeige am MAX7219

Register **Decode Mode** (Adresse: 09h):
Für die Ansteuerung einer Dot-Matrix-Anzeige sind die Datenbits (D0 bis D7) dieses Registers mit 0 zu laden (gemäß Tabelle 3.3.)

| DECODE MODE | REGISTER DATA | | | | | | | | HEX CODE |
|---|---|---|---|---|---|---|---|---|---|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| No decode for digits 7–0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x00 |
| Code B decode for digit 0<br>No decode for digits 7–1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0x01 |
| Code B decode for digits 3–0<br>No decode for digits 7–4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0x0F |
| Code B decode for digits 7–0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0xFF |

Tabelle 3.3.: Register 09h (Decode Mode)

Register **Intensity** (Adresse: 0Ah):
Dieses Register steuert die Helligkeit der Anzeige softwaremäßig in 16 Stufen. Tabelle 3.4. zeigt welche Datenbits für welche Helligkeitsstufe gesetzt werden müssen.

Anmerkung: Die Helligkeit (Intensity) kann auch hardwaremäßig per Widerstand am Eingangspin $I_{SET}$ (Pin 18) eingestellt werden. (z.B. mit einem Fotowiderstand für ein automatische Anpassung an die Umgebungshelligkeit)

| DUTY CYCLE | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | HEX CODE |
|---|---|---|---|---|---|---|---|---|---|---|
| MAX7219 | MAX7221 | | | | | | | | | |
| 1/32 (min on) | 1/16 (min on) | X | X | X | X | 0 | 0 | 0 | 0 | 0xX0 |
| 3/32 | 2/16 | X | X | X | X | 0 | 0 | 0 | 1 | 0xX1 |
| 5/32 | 3/16 | X | X | X | X | 0 | 0 | 1 | 0 | 0xX2 |
| 7/32 | 4/16 | X | X | X | X | 0 | 0 | 1 | 1 | 0xX3 |
| 9/32 | 5/16 | X | X | X | X | 0 | 1 | 0 | 0 | 0xX4 |
| 11/32 | 6/16 | X | X | X | X | 0 | 1 | 0 | 1 | 0xX5 |
| 13/32 | 7/16 | X | X | X | X | 0 | 1 | 1 | 0 | 0xX6 |
| 15/32 | 8/16 | X | X | X | X | 0 | 1 | 1 | 1 | 0xX7 |
| 17/32 | 9/16 | X | X | X | X | 1 | 0 | 0 | 0 | 0xX8 |
| 19/32 | 10/16 | X | X | X | X | 1 | 0 | 0 | 1 | 0xX9 |
| 21/32 | 11/16 | X | X | X | X | 1 | 0 | 1 | 0 | 0xXA |
| 23/32 | 12/16 | X | X | X | X | 1 | 0 | 1 | 1 | 0xXB |
| 25/32 | 13/16 | X | X | X | X | 1 | 1 | 0 | 0 | 0xXC |
| 27/32 | 14/16 | X | X | X | X | 1 | 1 | 0 | 1 | 0xXD |
| 29/32 | 15/16 | X | X | X | X | 1 | 1 | 1 | 0 | 0xXE |
| 31/32 | 15/16 (max on) | X | X | X | X | 1 | 1 | 1 | 1 | 0xXF |

Tabelle 3.4.: Register 0Ah (Intensity Mode)

Register **Scan Limit** (Adresse: 0Bh):
Mit diesem Register wird dem MAX7219 mitgeteilt wie viele Spalten die angeschlossene Dot-Matrix-Anzeige besitzt. Diese Angabe ist notwendig, damit der MAX7219 nicht unnötig Daten an die nicht vorhandenen Spalten ausgibt. Tabelle 3.5. zeigt wie die Datenbits je nach Anzahl der vorhanden Spalten (oder Digits bei 7-Segment-Anzeigen) gesetzt werden müssen. Eine falsche Einstellung beeinträchtigt auch die Helligkeit der Anzeige!

| SCAN LIMIT | REGISTER DATA | | | | | | | | HEX CODE |
|---|---|---|---|---|---|---|---|---|---|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| Display digit 0 only* | X | X | X | X | X | 0 | 0 | 0 | 0xX0 |
| Display digits 0 & 1* | X | X | X | X | X | 0 | 0 | 1 | 0xX1 |
| Display digits 0 1 2* | X | X | X | X | X | 0 | 1 | 0 | 0xX2 |
| Display digits 0 1 2 3 | X | X | X | X | X | 0 | 1 | 1 | 0xX3 |
| Display digits 0 1 2 3 4 | X | X | X | X | X | 1 | 0 | 0 | 0xX4 |
| Display digits 0 1 2 3 4 5 | X | X | X | X | X | 1 | 0 | 1 | 0xX5 |
| Display digits 0 1 2 3 4 5 6 | X | X | X | X | X | 1 | 1 | 0 | 0xX6 |
| Display digits 0 1 2 3 4 5 6 7 | X | X | X | X | X | 1 | 1 | 1 | 0xX7 |

Tabelle 3.5.: Register 0Bh (Scan Limit)

Register **Shutdown** (Adresse: 0Ch):
Mit diesem Register kann der MAX7219 ausgeschaltet werden. Dass bedeutet natürlich auch dass **keine** angeschlossenen Segmente leuchten. Dazu muss nur das Bit D0 (im Register 0Ch gelöscht werden. (siehe Tabelle 3.6).

| MODE | ADDRESS CODE (HEX) | REGISTER DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Shutdown Mode | 0xXC | X | X | X | X | X | X | X | 0 |
| Normal Operation | 0xXC | X | X | X | X | X | X | X | 1 |

Tabelle 3.6.: Register 0Ch (Shutdown)

Register **Display Test** (Adresse: 0Fh):
Mit diesem Register kann ein Selbsttest des MAX7219 erfolgen, d.h. alle Segmente leuchten mit der maximalen Helligkeit. Dazu muss nur das Bit D0 (im Register 0Fh) gesetzt werden. (siehe Tabelle 3.7).

| MODE | REGISTER DATA | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Normal Operation | X | X | X | X | X | X | X | 0 |
| Display Test Mode | X | X | X | X | X | X | X | 1 |

Tabelle 3.7.: Register 0Fh (Display Test)

Das beschreiben der soeben besprochenen Register erfolgt üblicherweise mit einem Mikrocontroller (siehe Bild 2.1) und benötigt 3 Portpins. Bild 3.2. zeigt das Protokoll und Tabelle 3.8. die einzuhaltenden Zeiten.

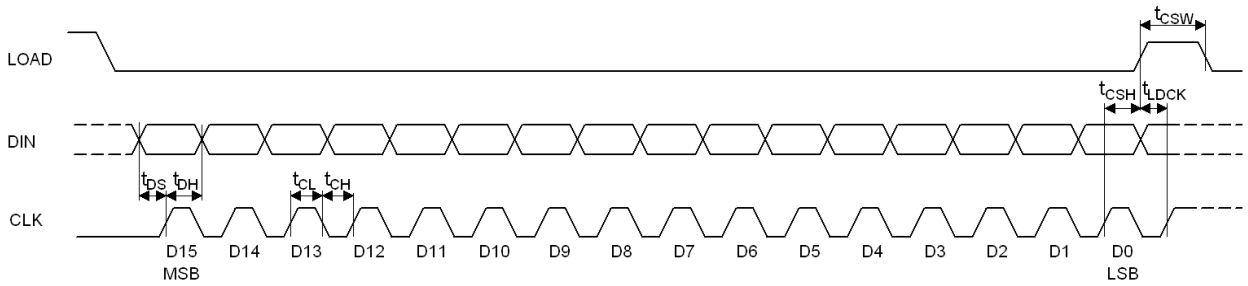



Bild 3.2.: Protokoll

| | Symbol | Min | Typ | Max | Einheit |
|---|---|---|---|---|---|
| CLK Pulse Width Low | $t_{CL}$ | 50 | — | — | ns |
| CLK Pulse Width High | $t_{CH}$ | 50 | — | — | ns |
| DIN Setup Time | $t_{DS}$ | 25 | — | — | ns |
| DIN Hold Time | $t_{DH}$ | 0 | — | — | ns |
| CLK Rise Time to LOAD Rise Hold Time | $t_{CSH}$ | 0 | — | — | ns |
| LOAD-Rising Edge to Next CLK-Rising Edge | $t_{LDCK}$ | 50 | — | — | ns |
| Minimum LOAD Pulse High | $t_{CSH}$ | 50 | — | — | ns |

Tabelle 3.8: Zeiten

Wie aus Bild 3.2. ersichtlich ist die Kommunikation mit dem MAX7219 sehr einfach:

- Zuerst muss LOAD zurückgesetzt werden.
- Danach werden die 16 Datenbits nacheinander, beginnend mit D15, in den MAX7219, getaktet. D.h. DIN muss je nach Inhalt entweder gesetzt oder gelöscht werden, und an CLK erfolgt eine positive Taktflanke (setzen und anschließendes rücksetzen).
- Wurden alle 16 Datenbits an den MAX7219 übertragen, LOAD wieder setzen.

Betrachtet man die Zeiten, so erkennt man dass keine Verzögerungsschleifen in der Software notwendig sind. Die Abarbeitungszeit einer Assembleranweisung benötigt im Durchschnitt 1µs bei einem typischen Takt von 4 MHz (Bei einer maximalen Taktfrequenz von 20 MHz benötigt eine Assembleranweisung 250ns, was auch noch viel größer ist, als die Werte in der Tabelle).

# 4. Software (in Assembler)

Die Aufgabe der Software ist das im Abschnitt 3 beschriebene Protokoll umzusetzen.

## 4.1. Benötigte Register und Portdefinition

**Register:**
Für die Realisierung des Softwareprotokolls werden neben einigen PIC internen Registern (SFR, **S**pezielle **F**unktions-**R**egister) noch zwei Übergaberegister (`TEMP1` und `TEMP2`) für den Aufruf von Unterprogrammen und ein Hilfsregister (`TEMP3`) benötigt. Diese drei Register (`TEMP1` bis `TEMP3`) sind so genannte temporäre Register. D.h. Sie werden nur kurzzeitig verwendet und können daher auch in beliebigen anderen Unterprogrammen verwendet werden.

**Portdefinition:**
Im Allgemeinen wird bei jeder Anwendung der MAX7219 an anderen Portpins angeschlossen. Damit dies in der Software nur an einer Stelle berücksichtigt werden muss befindet sich in der Software eine Portdefinition für den MAX7219. Dieser besteht aus den folgenden 5 Parametern:

- `MAX7219PORT:` Dieser Parameter gibt den Port an (z.B. Port A, Port B,…)
- `MAX7219TRIS:` Dieser Parameter ist für die Initialisierung des verwendeten Ports zuständig. Für die Ansteuerung des MAX7219 müssen die verwendeten Portpins als Ausgang definiert werden. Achtung: Wird für den Parameter `MAX7219PORT` der PORTA verwendet, so muss der Parameter `MAX7219TRIS` den Parameter TRISA beinhalten!
- `MAX7219DIN:` Dieser Parameter gibt den Portpin der Steuerleitung *DIN (Data Input)* an.
- `MAX7219CLK:` Dieser Parameter gibt den Portpin der Steuerleitung *CLK (Takt)* an.
- `MAX7219LOAD:` Dieser Parameter gibt den Portpin der Steuerleitung *LOAD* an.

Eine mögliche Portdefinition für einen PIC-Mikrocontroller aus der PIC16Fxx-Familie ist:

```
MAX7219PORT       equ   PORTB
MAX7219TRIS       equ   TRISB

MAX7219DIN        equ   5
MAX7219CLK        equ   4
MAX7219LOAD       equ   2
```

Bei Verwendung eines PIC-Mikrocontroller aus der PIC12Fxx-Familie lautet diese Portdefinition wie folgt:

```
MAX7219PORT       equ   GPIO
MAX7219TRIS       equ   TRISIO

MAX7219DIN        equ   5
MAX7219CLK        equ   4
MAX7219LOAD       equ   2
```

# 4.2. Initialisierung (Unterprogramm `INIT`)

Dieses Unterprogramm dient zur Initialisierung des Mikrocontrollers. Bei diesem Beispiel ist nur die Umschaltung der Comparatoreingänge auf digitale Ein- oder Ausgänge und die Definition der verwendeten Portpins (für die Ansteuerung des MAX7219) als Ausgang notwendig.

Der folgende Programmausschnitt zeigt eine mögliche Initialisierungsroutine für den PIC12F629.

```
INIT            clrf  GPIO
                movlw 0x07              ;Alle Comparatoreingaenge
                movwf CMCON             ;  auf digital I/O umschalten

                bsf   STAT,RP0          ;Registerseite 1
                clrf  TRISIO            ;Port als Ausgang definieren
                bcf   STAT,RP0          ;Registerseite 0

                clrf  GPIO              ;Port loeschen

                return
```

# 4.3. Unterprogramme für die Ansteuerung des MAX7219

Zur Ansteuerung des MAX7219 sind 2 Unterprogramme notwendig:

- Zuerst muss der MAX7219 initialisiert werden. Das Unterprogramm `MAX7219_INIT` übernimmt diese Aufgabe.
- Mit dem Unterprogramm `MAX7219_KOMMANDO` werden die Kommandos (also auch die Daten) an den MAX7219 übermittelt.

## 4.3.1. Unterprogramm `MAX7219_INIT`

**Aufgabe:**
MAX7219 initialisieren

**Vorgehensweise:**
Übergibt nacheinander die Befehle zur Initialisierung an den MAX7219:
Z.B.

- Adresse 0Bh: Scan Limit (hier: 04h da nur 5 Digits verwendet werden)
- Adresse 0Ch: Shutdown Format (hier: 01h für Normal Operation)
- Adresse 09h: Decode-Mode (hier: 00h für no decode)
- Adresse 0Ah: Intensity (hier: 0Fh für maximale Helligkeit)

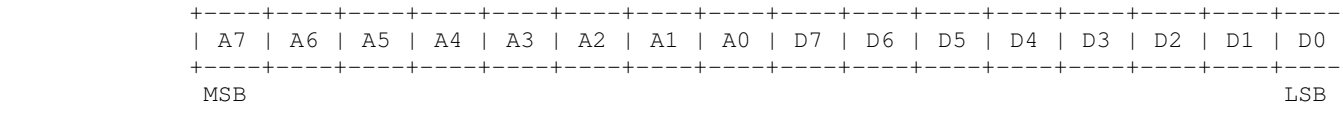
**Hier das Unterprogramm:**

```
MAX7219_INIT
            ; Addresse 0B
            movlw 0x0b        ;Adresse 0bh in Register TEMP1 kopieren
            movwf TEMP1
            movlw 0x04        ;Wert 04h in Register TEMP2 kopieren
            movwf TEMP2
            call MAX7219_KOMMANDO ;Adresse und Wert an den MAX7219 uebergeben

            ; Addresse 0C
            movlw 0x0c        ;Adresse 0ch in Register TEMP1 kopieren
            movwf TEMP1
            movlw 0x01        ;Wert 01h in Register TEMP2 kopieren
            movwf TEMP2
            call MAX7219_KOMMANDO ;Adresse und Wert an den MAX7219 uebergeben

            ; Addresse 09
            movlw 0x09        ;Adresse 09h in Register TEMP1 kopieren
            movwf TEMP1
            movlw 0x00;Wert 00h in Register TEMP2 kopieren
            movwf TEMP2
            call MAX7219_KOMMANDO ;Adresse und Wert an den MAX7219 uebergeben

            ; Addresse 0A
            movlw 0x0a       ;Adresse 0ah in Register TEMP1 kopieren
            movwf TEMP1
            movlw 0x0f       ;Wert 0fh in Register TEMP2 kopieren
            movwf TEMP2
            call MAX7219_KOMMANDO ;Adresse und Wert an den MAX7219 uebergeben

            return
```

**Anmerkung:**
Die Werte (Daten) der einzelnen Register hängen von der jeweiligen Anwendung ab. Im Demobeispiel 1 (Abschnitt 5) wird zum Beispiel das Scan Limit (Register 0Bh) mit 04h geladen, während es im Demobeispiel 2 (Abschnitt 6) mit 05h geladen wird.

## 4.3.2. Unterprogramm `MAX7219_KOMMANDO`

**Aufgabe:**
Dieses Unterprogramm sendet ein Kommando an den MAX7219. (gemäß Bild 3.2.) Ein Kommando besteht dabei aus acht Adressbits (A7 - A0; befinden sich im Übergaberegister `TEMP1`) und aus acht Datenbits (D7 - D0; befinden sich Übergaberegister `TEMP2`):




**Vorgehensweise:**
- Steuerleitung LOAD löschen
- Die Adresse (`TEMP1`) an den MAX7219 senden
  - Schleifenzähler (`TEMP3`) mit dem Wert 8 laden. (die folgende Schleife wird demnach achtmal durchlaufen
  - Schleifenbeginn

    - Alle Bits im Übergaberegister (hier: `TEMP1`) auf die nächst höhere Stelle schieben
    - Datenleitung DIN = Carry (= Flag im Statusregister): Für diese Anweisung ist jedoch kein Assemblerbefehl vorhanden! Es ist daher folgender Umweg notwendig: Zuerst prüfen, ob das Carry-Flag gesetzt ist. Ist es gesetzt, die Datenleitung (DIN) ebenfalls setzen. Andernfalls die Datenleitung (DIN) löschen. Ist das soeben eingelesene Bit (Bit 0 vom Register `TEMP3`) gesetzt, Bit 0 des Übergaberegister `TEMP1` setzen
    - Takt erzeugen: Dazu zunächst die Taktleitung (CLK) setzen, und wieder zurücksetzen
  - Schleifenende
- Data (`TEMP2`) auf die gleiche Weise wie die Adresse (`TEMP1`) an den MAX7219 senden
- Steuerleitung LOAD wieder setzen, und eine positive Taktflanke erzeugen

**Übergabeparameter:**

- `TEMP1`: Adresse
- `TEMP2`: Data

**Hier das Unterprogramm:**

```
MAX7219_KOMMANDO
                  ;Schritt 1: Steuerleitung LOAD loeschen
                  bcf    MAX7219PORT,MAX7219LOAD ;Steuerleitung LOAD loeschen

                  ;Schritt 2: Die Adresse (TEMP1) an den MAX7219 senden
                  movlw  .8         ;Schleifenzaehler (TEMP3) mit dem Wert 8
                  movwf  TEMP3      ; laden

MAX7219_SCHL1     rlf    TEMP1,f    ;Alle Bits im Uebergaberegister TEMP1 auf
                                    ;  die naechst hoehere Stelle schieben
                  btfss  STAT,C     ;ist das soeben ins Carry geschobene Bit
                                    ;  1?
                  goto   MAX7219_WEITER1
                  bsf    MAX7219PORT,MAX7219DIN;ja: Datenleitung DIN = 1
                  goto   MAX7219_WEITER2
MAX7219_WEITER1   bcf    MAX7219PORT,MAX7219DIN  ;nein: Datenleitung DIN = 0
MAX7219_WEITER2   bsf    MAX7219PORT,MAX7219CLK  ;Takt (positive und negative
                  bcf    MAX7219PORT,MAX7219CLK  ;  Flanke) erzeugen

                  decfsz TEMP3,f    ;Diesen Vorgang 8mal durchfuehren
                  goto   MAX7219_SCHL1

                  ;Schritt 3: Data (TEMP2) auf die gleiche Weise wie die
                  ;  Adresse (TEMP1) an den MAX7219 senden
                  movlw  .8         ;Schleifenzaehler (TEMP3) mit dem Wert 8
                  movwf  TEMP3      ;  laden

MAX7219_SCHL2     rlf    TEMP2,f   ;Bits im Uebergaberegister TEMP2 auf die
                                    ;  naechst hoehere Stelle schieben
                  btfss  STAT,C    ;ist das soeben ins Carry geschobene Bit
                                    ;  1?
                  goto   MAX7219_WEITER3
                  bsf    MAX7219PORT,MAX7219DIN  ;ja: Datenleitung DIN = 1
                  goto   MAX7219_WEITER4
MAX7219_WEITER3   bcf    MAX7219PORT,MAX7219DIN  ;nein: Datenleitung DIN = 0
MAX7219_WEITER4   bsf    MAX7219PORT,MAX7219CLK  ;Takt (positive und negative
                  bcf    MAX7219PORT,MAX7219CLK  ;  Flanke) erzeugen
```

```
        decfsz TEMP3,f   ;Diesen Vorgang 8mal durchfuehren
        goto   MAX7219_SCHL2

        ;Schritt 4: Die Steuerleitung LOAD wieder setzen, und eine
        ;  positive Taktflanke erzeugen
        bsf    MAX7219PORT,MAX7219LOAD  ;Steuerleitung LOAD wieder
        bsf    MAX7219PORT,MAX7219CLK   ;  setzen und eine positive
                                        ;  Taktflanke erzeugen
        return
```

**Anmerkungen:**

- Die temporären Register `TEMP1` bis `TEMP3` dienen hier nur als Übergabe- oder Hilfsregister. Diese Register können daher auch in anderen Unterprogrammen verwendet werden.
- Das Hauptprogramm oder ein anderes Unterprogramm welches dieses Unterprogramm aufruft muss die beiden Übergaberegister (`TEMP1` und `TEMP2`) mit den entsprechenden Werten laden!

# 5. Demonstrationsbeispiel 1

Das folgende Beispiel dient nur zur Demonstration. Es zeigt eine mögliche Einbindung der oben beschriebenen Unterprogramme.
In diesem Beispiel soll eine 5x7-Dot-Matrix so angesteuert werden, dass ein Buchstabe mit 5 Spalten und 7 Zeilen angezeigt wird.

## 5.1. Hardware

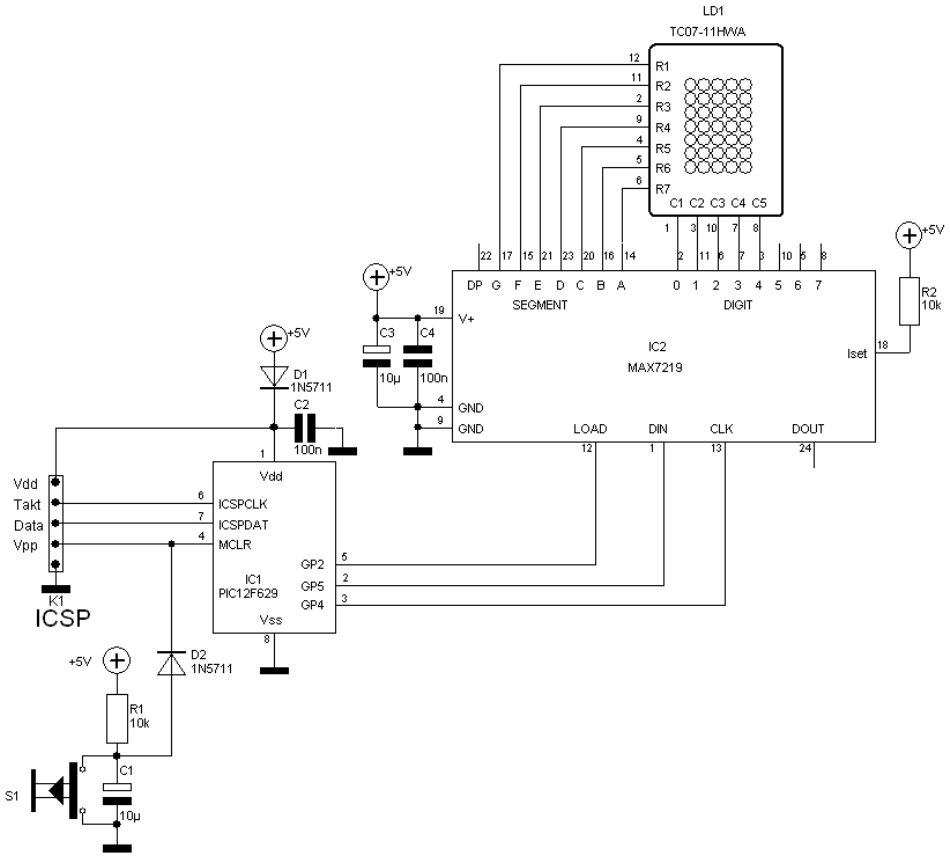



Bild 5.1: Schaltung zum Demonstrationsbeispiel 1

Als Dot-Matrix-Anzeige (LD1) wird der Typ TC07-11HWA verwendet. Die 5 Spalten der Dot-Matrix-Anzeige (C1 bis C5) werden mit den DIGIT-Ausgängen des MAX7219 (IC2) verbunden und die 7 Zeilen der Dot-Matrix-Anzeige (R1 bis R7) werden mit den SEGMENT-Ausgängen A bis G des MAX7219 verbunden.

Als Mikrocontroller (IC1) dient hier ein PIC12F629. Auffällig beim Mikrocontroller ist die fehlende Oszillatorschaltung. Diese ist hier auch nicht notwendig, da diese Anwendung nicht zeitkritisch ist und daher der interne Oszillator verwendet werden kann.

Zur Erzeugung des Reset (für den Mikrocontroller) wurde eine einfache Standardlösung bestehend aus einem Widerstand (R1) und einem Elektrolyt-Kondensator (C1) gewählt. Zusätzlich kann auch ein Reset mit dem Taster S1 erfolgen. Da der Reseteingang des Mikrocontrollers (MCLR, Pin 4) auch gleichzeitig die Programmierspannung (ca. 13V) bei der ICSP-Programmierung ist, darf während einer Programmierung kein Reset ausgelöst werden. Durch die Schottky-Diode D2 ist ein Reset durch das RC-Glied (R1 und C1) während einer Programmierung via ICSP nicht möglich.

Die Schottky-Diode D1 ist notwendig, damit beim Programmieren des Mikrocontroller (IC1) das Programmiergerät nicht die gesamte Hardware versorgt, sondern nur den Mikrocontroller.

Die Kommunikation des Mikrocontroller mit dem MAX7219 erfolgt über die drei Portpins GP5 (DIN), GP4 (CLK) und GP2 (LOAD).

Der Widerstand R2 bestimmt die maximale Helligkeit der Anzeige.

Der Kondensator C2 dient zur Entkoppelung der Betriebsspannung für den Mikrocontroller. Für diesen Koppelkondensator sollte ein Keramiktyp verwendet werden. Dieser muss möglichst nahe an diesen IC angebracht werden.

Die ICSP-Schnittstelle (K1) dient zur Programmierung des Mikrocontrollers (IC1), wobei bei dieser Methode der Mikrocontroller nicht aus der Schaltung entfernt werden muss.

Diese ICSP-Schnittstelle beinhaltet folgende Leitungen:

- Betriebsspannung (Vdd, Pin 1): Damit beim Programmieren das Programmiergerät nicht die gesamte Hardware versorgen muss (könnte das Programmiergerät überlasten) wird mit Hilfe der Diode D1 nur der Mikrocontroller IC1 mit dem Programmiergerät versorgt.
- Taktleitung (ICSPCLK, Pin 6): Hier wird dieser Pin nur für die ICSP-Schnittstelle benötigt.
- Datenleitung (ICSPDAT, Pin 7): Hier wird dieser Pin nur für die ICSP-Schnittstelle benötigt.
- Programmierspannung (MCLR/Vpp, Pin 4): Dieser Pin hat eine Doppelfunktion, er ist auch der Reset-Eingang, und muss deshalb mit einer Diode (D2) vor einem Reset durch das RC-Glied bestehend aus R1 und C1 geschützt werden.
- Masse (Vss, Pin 8)

Die Kondensatoren C3 und C4 dienen zur Entkoppelung der Betriebsspannung für den MAX7219. Für C4 sollte ein Keramiktyp verwendet werden. Dieser muss möglichst nahe an diesen IC angebracht werden.

Als Betriebsspannung muss eine stabile 5-V-Spannungsquelle verwendet werden.

# 5.2. Software

```
;*****************************************************************************************
;** Demonstrationsbeispiel 1 zum MAX7219 in Assembler                                    **
;**                                                                                      **
;** Entwickler: Buchgeher Stefan                                                         **
;** Entwicklungsbeginn der Software: 19. Februar 2006                                    **
;** Funktionsfaehig seit: 19. Februar 2006                                               **
;** Letzte Bearbeitung: 30. Maerz 2006                                                   **
;*****************************************************************************************

               List p=PIC12F629


;******************************** Register (in Registerseite 0) ******************************
STAT                    equ     3               ;Statusregister
GPIO                    equ     5               ;Port-Register
CMCON                   equ     1F              ;Komparator-Register


;******************************** Register (in Registerseite 1) ******************************
TRISIO                  equ     5               ;Port-Richtungsregister


;******************************** Eigene Register (in Registerbank 0) ************************
TEMP1                   equ     20              ;allgemeines Hilfsregister 1
TEMP2                   equ     21              ;allgemeines Hilfsregister 2
TEMP3                   equ     22              ;allgemeines Hilfsregister 3


;******************************** Bits in Registern ******************************************
C                       equ     0               ;Carrybit
RP0                     equ     5               ;Seitenauswahlbit 1 im Statuswort-Register


;******************************** Portbelegung ***********************************************
MAX7219PORT             equ     GPIO
MAX7219TRIS             equ     TRISIO

MAX7219DIN              equ     5
MAX7219CLK              equ     4
MAX7219LOAD             equ     2


;******************************** Konstanten *************************************************
; keine Konstanten


;******************************** Ziele der Registeroperationen ******************************
w                       equ     0
f                       equ     1


;******************************** Konfigurations-Bits ****************************************
       __config b'00000110100100'
;                  ++--------------- Bit 13-12 (BG1:BG0): Bandgap-Calibration
;                    ||||||||||||         0  0  : Lowest bandgap voltage
;                    ||||||||||||      ->  1  1  : Highest bandgap voltage
;                    +++------------ Bit 11-9: Reserve
;                       +----------- Bit  8 (CPD): Data Code Protection (Data Memory)
;                        ||||||||         0 : CPD on     (= enabled)
;                        ||||||||      ->  1 : CPD off    (= disabled)
;                        +---------- Bit 7 (CP): Code Protection (Programm Memory)
;                         |||||||         0 : CP on      (= enabled)
;                         |||||||      ->  1 : CP off     (= disabled)
;                         +--------- Bit 6 (BODEN): Brown-Out Detection
;                          ||||||      ->  0 : BODEN on   (= disabled)
;                          ||||||          1 : BODEN off  (= enabled)
;                          +-------- Bit 5 (MCLRE)
;                           |||||          0 : GP3 = I/O
;                           |||||      ->  1 : GP3 = MCLR
;                           +------- Bit 4 (PWRTE): Power Up Timer
;                            ||||     ->   0 : PWRT on    (= enabled)
;                            ||||          1 : PWRT off   (= disabled)
;                            +------ Bit 3 (WDTE): Watchdog Timer
;                             |||     ->   0 : WDT off    (= disabled)
;                             |||          1 : WDT on     (= enabled))
;                             +++--- Bit 2-0 (FOSC2:FOSC0): Oszillator Selectio
```

```
;                                       000 : LP Oszillator     (GP4=CLKOUT, GP5=CLKIN)
;                                       001 : XT Oszillator     (GP4=CLKOUT, GP5=CLKIN)
;                                       010 : HS Oszillator     (GP4=CLKOUT, GP5=CLKIN)
;                                       011 : EC Oszillator     (GP4=I/O, GP5=CLKIN)
;                                    ->  100 : INTOSC Oszillator (GP4=I/O, GP5=I/O)
;                                       101 : INTOSC Oszillator (GP4=CLKOUT, GP5=I/O)
;                                       110 : RC Oszillator     (GP4=I/O, GP5=RC)
;                                       111 : RC Oszillator     (GP4=CLKOUT, GP5=RC)


;******************************** Tabellen *********************************************
; keine Tabellen


;******************************** Einsprungadressen ************************************
               ORG     0x000
               goto    BEGINN
               ORG     0x004
               goto    ISR


;******************************** ISR **************************************************
ISR            retfie                      ;keine ISR vorhanden


;******************************** Unterprogramme ***************************************

;***************************************************************************************
;** Initialisierung des Prozessor:                                                    **
;**     + Comparatoreingaenge auf digitale I/O-Pins umschalten                        **
;**     + Port: als Ausgaenge definieren und loeschen                                 **
;***************************************************************************************
INIT           clrf    GPIO
               movlw   0x07                ;Alle Comparatoreingaenge
               movwf   CMCON               ;  auf digital I/O umschalten

               bsf     STAT,RP0            ;Registerseite 1
               clrf    TRISIO              ;Port als Ausgang definieren
               bcf     STAT,RP0            ;Registerseite 0

               clrf    GPIO                ;Port loeschen

               return


;***************************************************************************************
;** AUSGABE                                                                           **
;**                                                                                   **
;** Aufgabe:                                                                          **
;**     Den Buchstaben "Q" an den MAX7219 senden und am 5x7-Dot-Matrix ausgeben       **
;**                                                                                   **
;** Vorgehensweise:                                                                   **
;**     Dazu muessen nacheinander die Daten fuer die 5 Spalten an den MAX7219 mit Hilfe des  **
;**     Unterprogramms MAX7219_KOMMANDO uebergeben werden.                            **
;**              1  XXX                                                               **
;**              2 X   X                                                              **
;**              4 X   X                                                              **
;**              8 X   X                                                              **
;**             16 X X X                                                              **
;**             32 X  X                                                               **
;**             64  XX X                                                              **
;**                |||||                                                              **
;**                +------------------------------  Adresse 1: 62 (= 2 + 4 + 8 + 16 + 32)     **
;**                 +-----------------------------  Adresse 2: 65 (= 1 + 64)                  **
;**                  +----------------------------  Adresse 3: 81 (= 1 + 16 + 64)             **
;**                   +---------------------------  Adresse 4: 33 (= 1 + 32)                  **
;**                    +--------------------------  Adresse 5: 94 (= 2 + 4 + 8 + 16 + 64)     **
;**                                                                                   **
;** Uebergabeparameter (an das Unterprogramm MAX7219_KOMMANDO):                       **
;**     TEMP1: Adresse (hier die Nummer der Spalte)                                   **
;**     TEMP2: Date (hier der Wert der Spalte)                                        **
;**                                                                                   **
;** Anmerkung:                                                                        **
;**     Die beiden (temporaeren) Register TEMP1 und TEMP2 dienen hier nur als Uebergabe-     **
;**     register. Sie koennen auch in anderen Unterprogrammen verwendet werden.       **
;***************************************************************************************
AUSGABE        ; Adresse 01
               movlw   0x01                ;Adresse 01h in Register TEMP1 kopieren
               movwf   TEMP1
```

```
                movlw   .62                     ;Wert 62 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 02
                movlw   0x02                    ;Adresse 02h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .65                     ;Wert 65 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 03
                movlw   0x03                    ;Adresse 03h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .81                     ;Wert 81 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 04
                movlw   0x04                    ;Adresse 04h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .33                     ;Wert 33 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 05
                movlw   0x05                    ;Adresse 05h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .94                     ;Wert 94 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                return

;******************************** MAX7219 Routinen ***************************************

;*************************************************************************************************
;** MAX7219_INIT                                                                                **
;**                                                                                             **
;** Aufgabe:                                                                                    **
;**     MAX7219 initialisieren                                                                  **
;**                                                                                             **
;** Vorgehensweise:                                                                             **
;**     Uebergibt nacheinander die Befehle zur Initialisierung an den MAX7219:                  **
;**       Adresse 0Bh: Scan Limit (hier: 04h da nur 5 Digits verwendet werden)                  **
;**       Adresse 0Ch: Shutdown Format (hier: 01h für Normal Operation)                         **
;**       Adresse 09h: Decode-Mode (hier: 00h für no decode)                                    **
;**       Adresse 0Ah: Intensity (hier: 0Fh für maximale Helligkeit)                            **
;*************************************************************************************************
MAX7219_INIT    ; Adresse 0B (Scan Limit)
                movlw   0x0b                    ;Adresse 0bh in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x04                    ;Wert 04h in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 0C (Shutdown Format)
                movlw   0x0c                    ;Adresse 0ch in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x01                    ;Wert 01h in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 09 (Decode-Mode)
                movlw   0x09                    ;Adresse 09h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x00                    ;Wert 00h in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 0A (Intensity)
                movlw   0x0a                    ;Adresse 0ah in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x0f                    ;Wert 0fh in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                return
```

```
;*****************************************************************************************
;** MAX7219_KOMMANDO                                                                    **
;**                                                                                     **
;** Aufgabe:                                                                            **
;**     Dieses Unterprogramm sendet ein Kommando an den MAX7219. Ein Kommando besteht dabei  **
;**     aus acht Adressbits (A7 - A0; befinden sich im Uebergaberegister TEMP1) und aus acht **
;**     Datenbits (D7 - D0; befinden sich Uebergaberegister TEMP2):                     **
;**       +----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+  **
;**       | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |  **
;**       +----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+  **
;**        MSB                                                                   LSB   **
;**                                                                                     **
;** Vorgehensweise:                                                                     **
;**     + Steuerleitung LOAD loeschen                                                   **
;**     + Die Adresse (TEMP1) an den MAX7219 senden                                     **
;**         + Schleifenzaehler (TEMP3) mit dem Wert 8 lasen. (Die folgende Schleife wird     **
;**           demnach achtmal durchlaufen)                                              **
;**         + Schleifenbeginn                                                           **
;**            + Alle Bits im Uebergaberegister (hier: TEMP1) auf die naechst hoehere Stelle **
;**              schieben                                                               **
;**            + Datenleitung DIN = Carry (= Flag im Statusregister): Fuer diese Anweisung   **
;**              ist jedoch kein Assemblerbefehl vorhanden! Es ist daher folgender Umweg    **
;**              notwendig: Zuerst pruefen, ob das Carry-Flag gesetzt ist. Ist es gesetzt,  **
;**              die Datenleitung (DIN) ebenfalls setzen. Andernfalls die Datenleitung (DIN) **
;**              loeschen.                                                              **
;**            + Takt erzeugen: Dazu zunaechst die Taktleitung (CLK) setzen, und wieder     **
;**              zuruecksetzen                                                          **
;**         + Schleifenende                                                             **
;**     + Data (TEMP2) auf die gleiche Weise wie die Adresse (TEMP1) an den MAX7219 senden  **
;**     + Steuerleitung LOAD wieder setzen, und eine positive Taktflanke erzeugen       **
;**                                                                                     **
;** Uebergabeparameter:                                                                 **
;**     TEMP1:  Adresse                                                                 **
;**     TEMP2:  Data                                                                    **
;**                                                                                     **
;** Anmerkungen:                                                                        **
;**     + Die temporaeren Register TEMP1 bis TEMP3 dienen hier nur als Uebergabe- oder      **
;**       Hilfsregister. Diese Register koennen daher auch in anderen Unterprogrammen       **
;**       verwendet werden.                                                             **
;**     + Das Hauptprogramm oder eine anderes Unterprogramm welches dieses Unterprogramm    **
;**       aufruft muss die beiden Uebergaberegister (TEMP1 und TEMP2) mit den entsprechenden **
;**       Werten laden!                                                                 **
;*****************************************************************************************
MAX7219_KOMMANDO
               ;Schritt 1: Steuerleitung LOAD loeschen
               bcf MAX7219PORT,MAX7219LOAD     ;Steuerleitung LOAD loeschen

               ;Schritt 2: Die Adresse (TEMP1) an den MAX7219 senden
               movlw   .8                      ;Schleifenzaehler (TEMP3) mit dem Wert 8 laden
               movwf   TEMP3

MAX7219_SCHL1  rlf     TEMP1,f                 ;Alle Bits im Uebergaberegister TEMP1 auf die
                                               ;  naechst hoehere Stelle schieben
               btfss   STAT,C                  ;ist das soeben ins Carry geschobene Bit 1?
               goto    MAX7219_WEITER1
               bsf     MAX7219PORT,MAX7219DIN  ;ja: Datenleitung DIN = 1
               goto    MAX7219_WEITER2
MAX7219_WEITER1 bcf    MAX7219PORT,MAX7219DIN  ;nein: Datenleitung DIN = 0
MAX7219_WEITER2 bsf    MAX7219PORT,MAX7219CLK  ;Takt (positive und negative Flanke) erzeugen
               bcf     MAX7219PORT,MAX7219CLK

               decfsz  TEMP3,f                 ;Diesen Vorgang 8mal durchfuehren
               goto    MAX7219_SCHL1

               ;Schritt 3: Data (TEMP2) auf die gleiche Weise wie die Adresse (TEMP1)
               ;  an den MAX7219 senden
               movlw   .8                      ;Schleifenzaehler (TEMP3) mit dem Wert 8 laden
               movwf   TEMP3

MAX7219_SCHL2  rlf     TEMP2,f                 ;Bits im Uebergaberegister TEMP2 auf die
                                               ;  naechst hoehere Stelle schieben
               btfss   STAT,C                  ;ist das soeben ins Carry geschobene Bit 1?
               goto    MAX7219_WEITER3
               bsf     MAX7219PORT,MAX7219DIN  ;ja: Datenleitung DIN = 1
               goto    MAX7219_WEITER4
MAX7219_WEITER3 bcf    MAX7219PORT,MAX7219DIN  ;nein: Datenleitung DIN = 0
MAX7219_WEITER4 bsf    MAX7219PORT,MAX7219CLK  ;Takt (positive und negative Flanke) erzeugen
```

```
                bcf     MAX7219PORT,MAX7219CLK

                decfsz  TEMP3,f                 ;Diesen Vorgang 8mal durchfuehren
                goto    MAX7219_SCHL2

                ;Schritt 4: Die Steuerleitung LOAD wieder setzen, und eine positive
                ; Taktflanke erzeugen
                bsf     MAX7219PORT,MAX7219LOAD ;Steuerleitung LOAD wieder setzen
                bsf     MAX7219PORT,MAX7219CLK  ;und eine positive Taktflanke erzeugen
                return


;******************************** Hauptprogramm *************************************************

;************************************************************************************************
;** Aufgaben des Hauptprogramms:                                                               **
;** + Controller initialisieren (Unterprogramm INIT)                                           **
;** + MAX7219 initialisieren (Unterprogramm MAX7219_INIT)                                      **
;** + Ein Zeichen zur Demonstration ausgeben (Unterprogramm AUSGABE)                           **
;** + Endlosschleife                                                                           **
;************************************************************************************************
BEGINN          call    INIT                    ;Controller initialisieren
                Call    MAX7219_INIT            ;MAX7219 initialisieren

                call    AUSGABE                 ;Ein Zeichen zur Demonstration ausgeben

SCHLEIFE        goto    SCHLEIFE                ;Endlosschleife

                end
```

# 5.3. Anmerkungen zur Software

Die Software besteht im Wesentlichen aus einem kurzen Hauptprogramm, die im Abschnitt 4. besprochenen Unterprogramme zur Erzeugung des Protokolls für den MAX7219, und einem Unterprogramm `AUSGABE`.

Das Hauptprogramm besteht nach der Initialisierung des Mikrocontrollers (Unterprogramm `INIT`) und der Initialisierung des MAX7219 (Unterprogramm `MAX7219_INIT`) nur mehr aus dem Unterprogramm `AUSGABE` und einer Endlosschleife.

Das Unterprogramm `INIT` dient zur Initialisierung des Controllers. Hier werden die Ports konfiguriert (Port dient hier als Ausgang) und die Comparatoreingänge auf digitale Ausgänge umgestellt. Dieses Unterprogramm ist vom Controllertyp abhängig und je nach Anwendung mehr oder weniger umfangreich. Siehe auch Abschnitt 4.2 (Initialisierung)

Das Unterprogramm `AUSGABE` soll hier noch näher betrachtet werden. Dieses Unterprogramm hat hier die Aufgabe das Zeichen, welches am MAX7219 dargestellt werden soll, hier der Buchstabe „Q“, in seine Spalten zu zerteilen und diese Bytes nacheinander in die Digit-Register 0 bis 4 zu schreiben (siehe Listing).

Wichtig im Unterprogramm `MAX7219_INIT` ist, dass hier das Register 0Bh (Scan Limit des MAX7219) mit dem Wert 04h geladen wird.

# 6. Demonstrationsbeispiel 2

Bei diesem Beispiel soll eine 5x7-Dot-Matrix so angesteuert werden, dass zwei Ziffern mit je 3 Spalten und 5 Zeilen angezeigt werden. (z. B. die Zahl 12)

## 6.1. Hardware

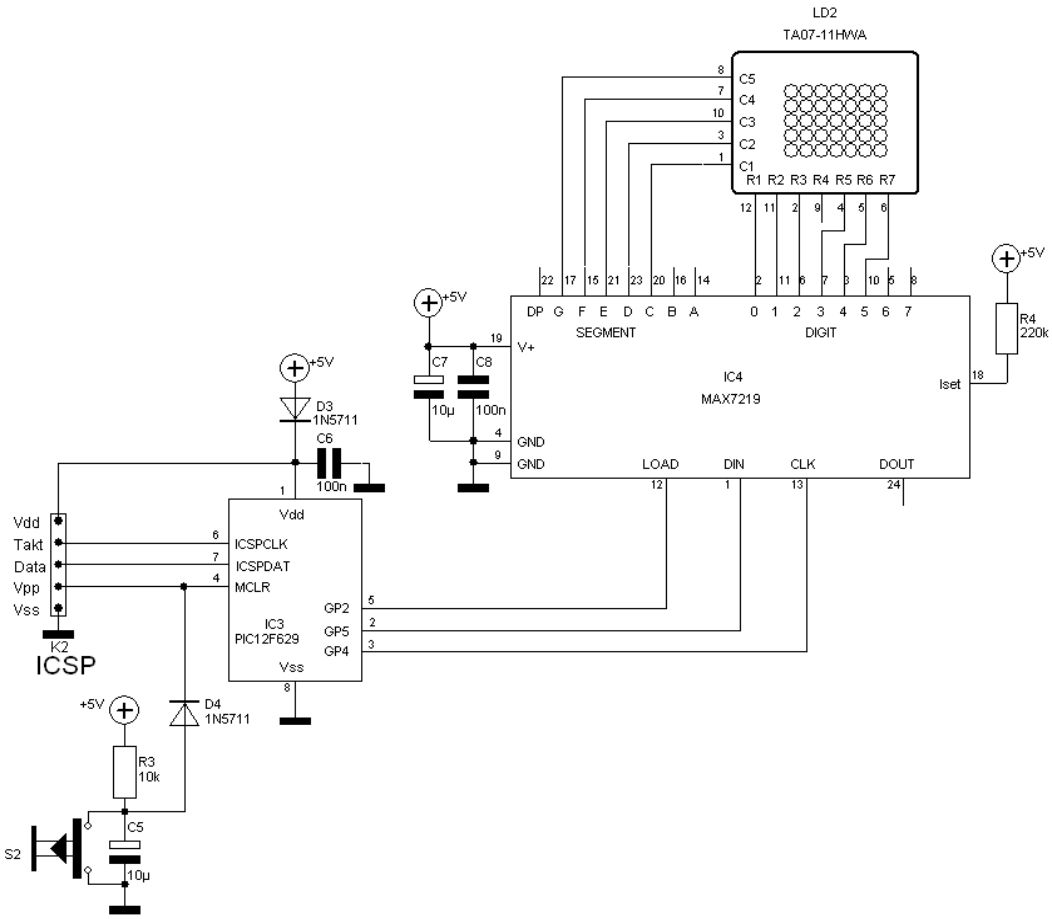



Bild 6.1: Schaltung zum Demonstrationsbeispiel 2

Im Gegensatz zum Demonstrationsbeispiel 1 wird hier für die Dot-Matrix-Anzeige (LD2) der Typ TA07-11HWA verwendet. Dies ist notwendig da hier diese Anzeige nicht „stehend“ sondern „liegend“ verwendet wird und daher die Spalten und Zeilen miteinander vertauscht sind. Die 5 Spalten der Dot-Matrix-Anzeige (C1 bis C5), die bei dieser Anwendung eigentlich Zeilen sind, werden mit den SEGMENT-Ausgängen C bis G des MAX7219 (IC4) verbunden und die 7 Zeilen der Dot-Matrix-Anzeige (R1 bis R7), die bei dieser Anwendung eigentlich Spalten sind, werden mit den DIGIT-Ausgängen 1 bis 5 des MAX7219 verbunden, wobei die „Spalte“ R4 nicht benötigt wird, und somit als Trennung zwischen den beiden Ziffern dient.

Als Mikrocontroller (IC3) dient auch hier ein PIC12F629. Auch hier ist die Anwendung nicht zeitkritisch, so dass der interne Oszillator verwendet werden kann.

Bis auf den Widerstand R4 (hier 220k) zur Bestimmung der maximalen Helligkeit sind keine weiteren Änderungen im Vergleich zum Demonstrationsbeispiel 1 vorhanden. Dieser Widerstand wurde verändert, um zu zeigen, dass die Helligkeit tatsächlich von diesem externen Widerstand abhängt.

# 6.2. Software

```
;*****************************************************************************************
;** Demonstrationsbeispiel 2 zum MAX7219 in Assembler                                    **
;**                                                                                      **
;** Entwickler: Buchgeher Stefan                                                         **
;** Entwicklungsbeginn der Software: 28. Maerz 2006                                      **
;** Funktionsfaehig seit: 28. Maerz 2006                                                 **
;** Letzte Bearbeitung: 9. Maerz 2006                                                    **
;*****************************************************************************************

               List p=PIC12F629


;******************************** Register (in Registerseite 0) ******************************
STAT                   equ     3              ;Statusregister
GPIO                   equ     5              ;Port-Register
CMCON                  equ     1F             ;Komparator-Register


;******************************** Register (in Registerseite 1) ******************************
TRISIO                 equ     5              ;Port-Richtungsregister


;******************************** Eigene Register (in Registerbank 0) ************************
TEMP1                  equ     20             ;allgemeines Hilfsregister 1
TEMP2                  equ     21             ;allgemeines Hilfsregister 2
TEMP3                  equ     22             ;allgemeines Hilfsregister 3


;******************************** Bits in Registern ******************************************
C                      equ     0              ;Carrybit
RP0                    equ     5              ;Seitenauswahlbit 1 im Statuswort-Register


;******************************** Portbelegung ***********************************************
MAX7219PORT            equ     GPIO
MAX7219TRIS            equ     TRISIO

MAX7219DIN             equ     5
MAX7219CLK             equ     4
MAX7219LOAD            equ     2


;******************************** Konstanten *************************************************
; keine Konstanten


;******************************** Ziele der Registeroperationen ******************************
w                      equ     0
f                      equ     1


;******************************** Konfigurations-Bits ****************************************
       __config b'00000110100100'
;                  ++--------------- Bit 13-12 (BG1:BG0): Bandgap-Calibration
;                    ||||||||||||         0  0  : Lowest bandgap voltage
;                    ||||||||||||      ->  1  1  : Highest bandgap voltage
;                    +++------------ Bit 11-9: Reserve
;                       +----------- Bit  8 (CPD): Data Code Protection (Data Memory)
;                        ||||||||          0 : CPD on     (= enabled)
;                        ||||||||       ->  1 : CPD off    (= disabled)
;                        +---------- Bit 7 (CP): Code Protection (Programm Memory)
;                         |||||||          0 : CP on      (= enabled)
;                         |||||||       ->  1 : CP off     (= disabled)
;                         +--------- Bit 6 (BODEN): Brown-Out Detection
;                          ||||||       ->  0 : BODEN on   (= disabled)
;                          ||||||           1 : BODEN off  (= enabled)
;                          +-------- Bit 5 (MCLRE)
;                           |||||           0 : GP3 = I/O
;                           |||||        ->  1 : GP3 = MCLR
;                           +------- Bit 4 (PWRTE): Power Up Timer
;                            ||||     ->    0 : PWRT on    (= enabled)
;                            ||||           1 : PWRT off   (= disabled)
;                            +------ Bit 3 (WDTE): Watchdog Timer
;                             |||     ->    0 : WDT off    (= disabled)
;                             |||           1 : WDT on     (= enabled))
;                             +++--- Bit 2-0 (FOSC2:FOSC0): Oszillator Selectio
```

```
;                                       000 : LP Oszillator     (GP4=CLKOUT, GP5=CLKIN)
;                                       001 : XT Oszillator     (GP4=CLKOUT, GP5=CLKIN)
;                                       010 : HS Oszillator     (GP4=CLKOUT, GP5=CLKIN)
;                                       011 : EC Oszillator     (GP4=I/O, GP5=CLKIN)
;                                    ->  100 : INTOSC Oszillator (GP4=I/O, GP5=I/O)
;                                       101 : INTOSC Oszillator (GP4=CLKOUT, GP5=I/O)
;                                       110 : RC Oszillator     (GP4=I/O, GP5=RC)
;                                       111 : RC Oszillator     (GP4=CLKOUT, GP5=RC)


;******************************** Tabellen *********************************************
; keine Tabellen


;******************************** Einsprungadressen ************************************
                ORG     0x000
                goto    BEGINN
                ORG     0x004
                goto    ISR


;******************************** ISR **************************************************
ISR             retfie                          ;keine ISR vorhanden


;******************************** Unterprogramme ***************************************

;***************************************************************************************
;** Initialisierung des Prozessor:                                                    **
;**     + Comparatoreingaenge auf digitale I/O-Pins umschalten                        **
;**     + Port: als Ausgaenge definieren und loeschen                                 **
;***************************************************************************************
INIT            clrf    GPIO
                movlw   0x07                    ;Alle Comparatoreingaenge
                movwf   CMCON                   ;  auf digital I/O umschalten

                bsf     STAT,RP0                ;Registerseite 1
                clrf    TRISIO                  ;Port als Ausgang definieren
                bcf     STAT,RP0                ;Registerseite 0

                clrf    GPIO                    ;Port loeschen

                return


;***************************************************************************************
;** AUSGABE                                                                           **
;**                                                                                   **
;** Aufgabe:                                                                          **
;**     Die Zahl "12" an den MAX7219 senden und am 5x7-Dot-Matrix ausgeben            **
;**                                                                                   **
;** Vorgehensweise:                                                                   **
;**     Dazu muessen nacheinander die Daten fuer die 6 Spalten an den MAX7219 mit Hilfe des  **
;**     Unterprogramms MAX7219_KOMMANDO uebergeben werden.                            **
;**              1  X  XX                                                             **
;**              2 XX    X                                                            **
;**              4  X   X                                                             **
;**              8  X  X                                                              **
;**             16  X  XXX                                                            **
;**                |||  |||                                                          **
;**                +------------------------------  Adresse 1:  2 (= 2)              **
;**                 +-----------------------------  Adresse 2: 31 (= 1 + 2 + 4 + 8 + 16)  **
;**                  +----------------------------  Adresse 3:  0                     **
;**                    |||                                                          **
;**                    +--------------------------  Adresse 4: 25 (= 1 + 8 + 16)     **
;**                     +-------------------------  Adresse 5: 21 (= 1 + 4 + 16)     **
;**                      +------------------------  Adresse 6: 18 (= 2 + 16)         **
;**                                                                                   **
;** Uebergabeparameter (an das Unterprogramm MAX7219_KOMMANDO):                       **
;**     TEMP1: Adresse (hier die Nummer der Spalte)                                   **
;**     TEMP2: Date (hier der Wert der Spalte)                                        **
;**                                                                                   **
;** Anmerkungen:                                                                      **
;**     Die beiden (temporaeren) Register TEMP1 und TEMP2 dienen hier nur als Uebergabe-  **
;**     register. Sie koennen auch in anderen Unterprogrammen verwendet werden.       **
;***************************************************************************************
AUSGABE         ; Adresse 01
                movlw   0x01                    ;Adresse 01h in Register TEMP1 kopieren
                movwf   TEMP1
```

```
                movlw   .2                      ;Wert 2 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 02
                movlw   0x02                    ;Adresse 02h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .31                     ;Wert 31 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 03
                movlw   0x03                    ;Adresse 03h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .0                      ;Wert 0 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 04
                movlw   0x04                    ;Adresse 04h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .25                     ;Wert 25 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 05
                movlw   0x05                    ;Adresse 05h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .21                     ;Wert 21 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 06
                movlw   0x06                    ;Adresse 06h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   .18                     ;Wert 18 in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                return

;******************************** MAX7219 Routinen *****************************************

;***************************************************************************************************
;** MAX7219_INIT                                                                                  **
;**                                                                                               **
;** Aufgabe:                                                                                      **
;**     MAX7219 initialisieren                                                                    **
;**                                                                                               **
;** Vorgehensweise:                                                                               **
;**     Uebergibt nacheinander die Befehle zur Initialisierung an den MAX7219:                    **
;**       Adresse 0Bh: Scan Limit (hier: 05h da nur 6 Digits verwendet werden)                    **
;**       Adresse 0Ch: Shutdown Format (hier: 01h für Normal Operation)                           **
;**       Adresse 09h: Decode-Mode (hier: 00h für no decode)                                      **
;**       Adresse 0Ah: Intensity (hier: 0Fh für maximale Helligkeit)                              **
;***************************************************************************************************
MAX7219_INIT    ; Adresse 0B (Scan Limit)
                movlw   0x0b                    ;Adresse 0bh in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x05                    ;Wert 05h in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 0C (Shutdown Format)
                movlw   0x0c                    ;Adresse 0ch in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x01                    ;Wert 01h in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 09 (Decode-Mode)
                movlw   0x09                    ;Adresse 09h in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x00                    ;Wert 00h in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                ; Adresse 0A (Intensity)
```

```
                movlw   0x0a                    ;Adresse 0ah in Register TEMP1 kopieren
                movwf   TEMP1
                movlw   0x0f                    ;Wert 0fh in Register TEMP2 kopieren
                movwf   TEMP2
                call    MAX7219_KOMMANDO        ;Adresse und Wert an den MAX7219 uebergeben

                return


;*****************************************************************************************
;** MAX7219_KOMMANDO                                                                    **
;**                                                                                     **
;** Aufgabe:                                                                            **
;**     Dieses Unterprogramm sendet ein Kommando an den MAX7219. Ein Kommando besteht dabei  **
;**     aus acht Adressbits (A7 - A0; befinden sich im Uebergaberegister TEMP1) und aus acht **
;**     Datenbits (D7 - D0; befinden sich Uebergaberegister TEMP2):                     **
;**       +----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+  **
;**       | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |  **
;**       +----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+  **
;**        MSB                                                                   LSB   **
;**                                                                                     **
;** Vorgehensweise:                                                                     **
;**     + Steuerleitung LOAD loeschen                                                   **
;**     + Die Adresse (TEMP1) an den MAX7219 senden                                     **
;**         + Schleifenzaehler (TEMP3) mit dem Wert 8 lasen. (Die folgende Schleife wird     **
;**           demnach achtmal durchlaufen)                                              **
;**         + Schleifenbeginn                                                           **
;**            + Alle Bits im Uebergaberegister (hier: TEMP1) auf die naechst hoehere Stelle **
;**              schieben                                                               **
;**            + Datenleitung DIN = Carry (= Flag im Statusregister): Fuer diese Anweisung   **
;**              ist jedoch kein Assemblerbefehl vorhanden! Es ist daher folgender Umweg     **
;**              notwendig: Zuerst pruefen, ob das Carry-Flag gesetzt ist. Ist es gesetzt,   **
;**              die Datenleitung (DIN) ebenfalls setzen. Andernfalls die Datenleitung (DIN) **
;**              loeschen.                                                              **
;**            + Takt erzeugen: Dazu zunaechst die Taktleitung (CLK) setzen, und wieder      **
;**              zuruecksetzen                                                          **
;**         + Schleifenende                                                             **
;**     + Data (TEMP2) auf die gleiche Weise wie die Adresse (TEMP1) an den MAX7219 senden   **
;**     + Steuerleitung LOAD wieder setzen, und eine positive Taktflanke erzeugen       **
;**                                                                                     **
;** Uebergabeparameter:                                                                 **
;**     TEMP1:  Adresse                                                                 **
;**     TEMP2:  Data                                                                    **
;**                                                                                     **
;** Anmerkungen:                                                                        **
;**     + Die temporaeren Register TEMP1 bis TEMP3 dienen hier nur als Uebergabe- oder       **
;**       Hilfsregister. Diese Register koennen daher auch in anderen Unterprogrammen       **
;**       verwendet werden.                                                             **
;**     + Das Hauptprogramm oder eine anderes Unterprogramm welches dieses Unterprogramm     **
;**       aufruft muss die beiden Uebergaberegister (TEMP1 und TEMP2) mit den entsprechenden **
;**       Werten laden!                                                                 **
;*****************************************************************************************
MAX7219_KOMMANDO
                ;Schritt 1: Steuerleitung LOAD loeschen
                bcf MAX7219PORT,MAX7219LOAD     ;Steuerleitung LOAD loeschen

                ;Schritt 2: Die Adresse (TEMP1) an den MAX7219 senden
                movlw   .8                      ;Schleifenzaehler (TEMP3) mit dem Wert 8 laden
                movwf   TEMP3

MAX7219_SCHL1   rlf     TEMP1,f                 ;Alle Bits im Uebergaberegister TEMP1 auf die
                                                ;  naechst hoehere Stelle schieben
                btfss   STAT,C                  ;ist das soeben ins Carry geschobene Bit 1?
                goto    MAX7219_WEITER1
                bsf     MAX7219PORT,MAX7219DIN  ;ja: Datenleitung DIN = 1
                goto    MAX7219_WEITER2
MAX7219_WEITER1 bcf     MAX7219PORT,MAX7219DIN  ;nein: Datenleitung DIN = 0
MAX7219_WEITER2 bsf     MAX7219PORT,MAX7219CLK  ;Takt (positive und negative Flanke) erzeugen
                bcf     MAX7219PORT,MAX7219CLK

                decfsz  TEMP3,f                 ;Diesen Vorgang 8mal durchfuehren
                goto    MAX7219_SCHL1

                ;Schritt 3: Data (TEMP2) auf die gleiche Weise wie die Adresse (TEMP1)
                ;  an den MAX7219 senden
                movlw   .8                      ;Schleifenzaehler (TEMP3) mit dem Wert 8 laden
                movwf   TEMP3

MAX7219_SCHL2   rlf     TEMP2,f                 ;Bits im Uebergaberegister TEMP2 auf die
```

```
                                        ;  naechst hoehere Stelle schieben
                btfss   STAT,C          ;ist das soeben ins Carry geschobene Bit 1?
                goto    MAX7219_WEITER3
                bsf     MAX7219PORT,MAX7219DIN  ;ja: Datenleitung DIN = 1
                goto    MAX7219_WEITER4
MAX7219_WEITER3 bcf     MAX7219PORT,MAX7219DIN  ;nein: Datenleitung DIN = 0
MAX7219_WEITER4 bsf     MAX7219PORT,MAX7219CLK  ;Takt (positive und negative Flanke) erzeugen
                bcf     MAX7219PORT,MAX7219CLK

                decfsz  TEMP3,f         ;Diesen Vorgang 8mal durchfuehren
                goto    MAX7219_SCHL2

                ;Schritt 4: Die Steuerleitung LOAD wieder setzen, und eine positive
                ; Taktflanke erzeugen
                bsf     MAX7219PORT,MAX7219LOAD ;Steuerleitung LOAD wieder setzen
                bsf     MAX7219PORT,MAX7219CLK  ;und eine positive Taktflanke erzeugen
                return


;******************************** Hauptprogramm *********************************************

;********************************************************************************************
;** Aufgaben des Hauptprogramms:                                                           **
;** + Controller initialisieren (Unterprogramm INIT)                                       **
;** + MAX7219 initialisieren (Unterprogramm MAX7219_INIT)                                  **
;** + Ein Zeichen zur Demonstration ausgeben (Unterprogramm AUSGABE)                       **
;** + Endlosschleife                                                                       **
;********************************************************************************************
BEGINN          call    INIT            ;Controller initialisieren
                Call    MAX7219_INIT    ;MAX7219 initialisieren

                call    AUSGABE         ;Ein Zeichen zur Demonstration ausgeben

SCHLEIFE        goto    SCHLEIFE        ;Endlosschleife

                end
```

## 6.3. Anmerkungen zur Software

Hier gilt im Wesentlichen das gleiche wie beim Demonstrationsbeispiel 1, geändert wurden nur die folgenden 2 Punkte:

- Unterprogramm `AUSGABE`:
- Unterprogramm `MAX7219_INIT`: Hier muss das Register 0Bh (Scan Limit des MAX7219) mit dem Wert 05h geladen wird.

# 7. Kaskadieren mehrerer MAX7219

Bild 7.1. zeigt das Prinzipschaltbild zum Hintereinanderschalten (kaskadieren) mehrerer MAX7219.

Für jede Dot-Matrix-Anzeige wird nun ein eigener MAX7219 verwendet. Neu ist nun, dass der Ausgang DOUT mit dem Eingang DIN des darauffolgenden MAX7219 verbunden werden muss. Alle LOAD-Eingänge müssen miteinander verbunden werden, und auch alle CLK-Eingänge werden miteinander verbunden.

Für den zu steuernden Mikrocontroller ist also kein zusätzlicher Portpin notwendig.

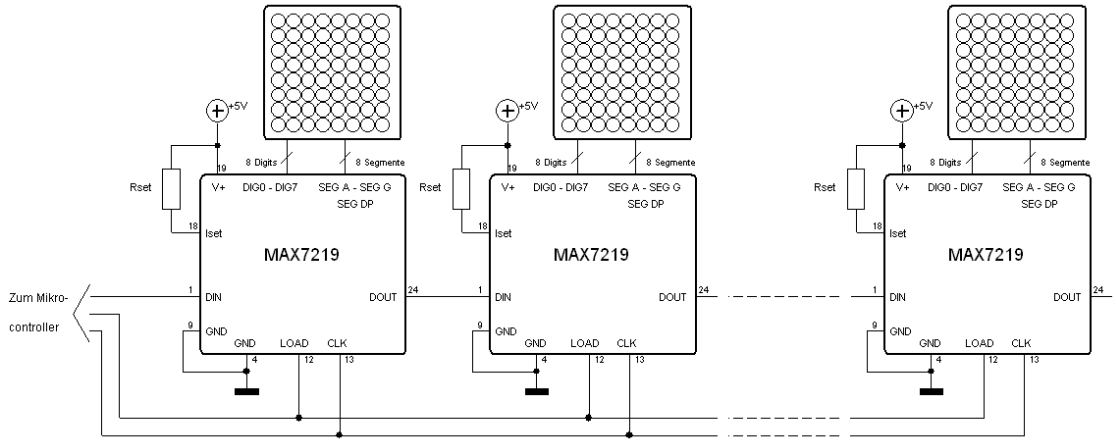



Bild 7.1.: Prinzipschaltbild zum Kaskadieren mehrerer Dot-Matrix-Anzeige mit mehreren MAX7219

Natürlich muss auch die Software wie folgt angepasst werden. Bild 7.2. zeigt dabei das Protokoll für zwei hintereinandergeschaltete MAX7219.

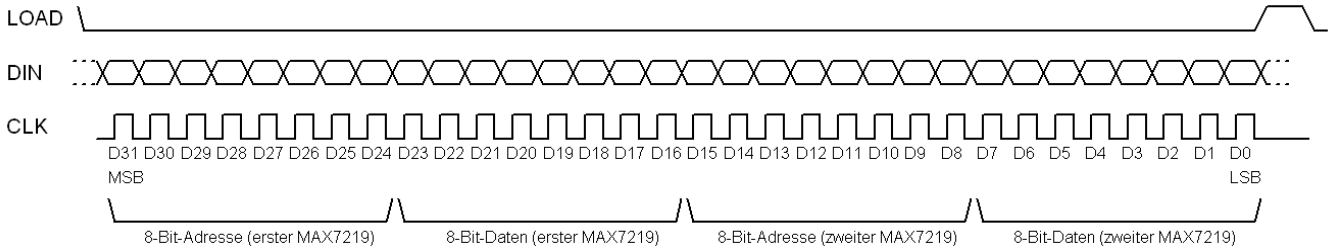



Bild 7.2.: Protokoll für zwei MAX7219

Wie aus Bild 7.2. ersichtlich ist die Kommunikation mit dem MAX7219 auch in diesem Fall sehr einfach und unterscheidet sich nur in der Anzahl der Takte (CLK) zum Protokoll für nur eine MAX7219 (siehe Bild 3.2):

- Zuerst muss LOAD zurückgesetzt werden.
- Danach werden die 16 Datenbits für den ersten MAX7219[3] nacheinander, hier beginnend mit D31 bis D16, in den MAX7219, getaktet. D.h. DIN muss je nach Inhalt entweder gesetzt oder gelöscht werden, und an CLK erfolgt eine positive Taktflanke (setzen und anschließendes rücksetzen).
- Nun folgen die 16 Datenbits für den zweiten MAX7219 (Hier D15 bis D0).
- Wurden alle 32 Datenbits übertragen, LOAD wieder setzen.

Anmerkung:
Werden nur einem MAX7219 Daten gesendet, so muss dennoch das Protokoll nach Bild 7.2. gesendet werden. Dem anderen MAX7219 muss in diesem Fall die Adresse 00h (= No Operation, siehe Seite 6) gesendet werden.

[3] Achtung: Der erste MAX7219 ist jener, der vom Mikrocontroller am weitesten „entfernt“ ist!

# 8. Software (in C mit mikroC)

Die hier beschriebenen Unterprogramme orientieren sich sehr stark an Abschnitt 4, werden aber diesmal in der Programmiersprache C für den mikroC-Compiler ausgeführt. Diese Unterprogramme können natürlich auch in Assembler realisiert werden, und sollten auch sehr einfach in andere C-Compiler übergeführt werden können.

Für die Portdefinition und für das Initialisieren des Mikrocontrollers gilt das gleiche wie schon in Abschnitt 4.1. und 4.2. beschrieben, nur die C-Syntax für die Portdefinition unterscheidet sich zu Assembler:

Beim mikroC-Compiler lautet die Portdefinition für einen PIC-Mikrocontroller aus der PIC16Fxx-Familie:

```
#define MAX7219DIN          PORTB.F4
#define MAX7219LOAD         PORTB.F2
#define MAX7219CLK          PORTB.F5
```

Bei Verwendung eines PIC-Mikrocontrollers aus der PIC12Fxx-Familie lautet diese Portdefinition wie folgt:

```
#define MAX7219DIN          GPIO.F4
#define MAX7219LOAD         GPIO.F2
#define MAX7219CLK          GPIO.F5
```

Die Ansteuerung von mehreren (kaskadierten) erfolgt mit den Unterprogrammen `MAX7219_INIT2()`, `MAX7219_KOMMANDO2()` und dem Hilfsunterprogramm `MAX7219_SENDBYTE()`. Die Ziffer 2 bedeutet hier, dass nur zwei MAX7219 kaskadiert sind, werden drei oder mehr MAX7219 hintereinandergeschaltet, so empfiehlt es sich für diese Fälle eigene Unterprogramme zu schreiben, die entsprechend erweitert sind.

In C ist es üblich zusammengehörende Unterprogramme in einer eigenen Datei mit der Endung `.C` zu schreiben. Die nun folgenden drei Unterprogramme befinden sich daher in einer Datei mit dem Namen `MAX7219.C`. (siehe Demonstrationsbeispiel 3, Abschnitt 9).

## 8.1. Unterprogramm `MAX7219_INIT2()`

**Aufgabe:**
Die beiden kaskadierten MAX7219 initialisieren

**Vorgehensweise:**
Übergibt nacheinander die Befehle zur Initialisierung an beide MAX7219 (mit Hilfe des Unterprogramms `MAX7219_KOMMANDO2`)

- Adresse 0Bh: Scan Limit
- Adresse 0Ch: Shutdown Format
- Adresse 09h: Decode-Mode
- Adresse 0Ah: Intensity

**Übergabeparameter:**

- `Scanlimit`
- `ShutdownFormat`
- `DecodeMode`
- `Intensity`

**Rückgabeparameter:**
keiner

**Hier das Unterprogramm:**

```
void MAX7219_INIT2(char Scanlimit, char ShutdownFormat, char DecodeMode, char
Intensity)
{
    // Scanlimit an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x0b, Scanlimit, 0x0b, Scanlimit);

    // Shutdown-Format an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x0c, ShutdownFormat, 0x0c, ShutdownFormat);

    // Decode-Mode an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x09, DecodeMode, 0x09, DecodeMode);

    // Intensity an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x0a, Intensity, 0x0a, Intensity);
}
```

# 8.2. Unterprogramm `MAX7219_KOMMANDO2()`

**Aufgabe:**
Dieses Programm sendet nacheinander zwei Kommandos an die beiden MAX7219. (Für jeden MAX7219 ein Kommando). Ein Kommando besteht dabei aus acht Adressbits (A7 - A0) und aus acht Datenbits (D7 - D0).

```
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
| A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
 MSB                                                                       LSB
```

**Vorgehensweise:**

- Steuerleitung LOAD löschen
- Die Adresse des ersten MAX7219 (Übergabeparameter `Adr1`) mit Hilfe eine Hilfsunterprogramms (`MAX7219_SENDBYTE`) an den MAX7219 senden.
- Die Daten des ersten MAX7219 (Übergabeparameter `Data1`) mit dem gleichen Hilfsunterprogramm (`MAX7219_SENDBYTE`) an den MAX7219 senden.
- Die Adresse des zweiten MAX7219 (Übergabeparameter `Adr2`) mit dem gleichen Hilfsunterprogramm (`MAX7219_SENDBYTE`) an den MAX7219 senden.
- Die Daten des zweiten MAX7219 (Übergabeparameter `Data2`) mit dem gleichen Hilfsunterprogramm (`MAX7219_SENDBYTE`) an den MAX7219 senden.
- Steuerleitung LOAD wieder setzen, und eine positive Taktflanke erzeugen

**Übergabeparameter:**

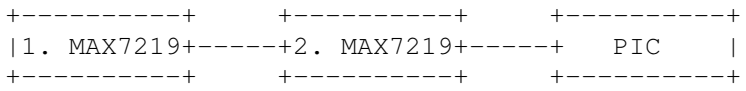
- Adresse des 1. MAX7219 (`Adr1`)
- Daten des 1. MAX7219 (`Data1`)
- Adresse des 2. MAX7219 (`Adr2`)
- Daten des 2. MAX7219 (`Data2`)

**Rückgabeparameter:**
keiner

**Hier das Unterprogramm:**

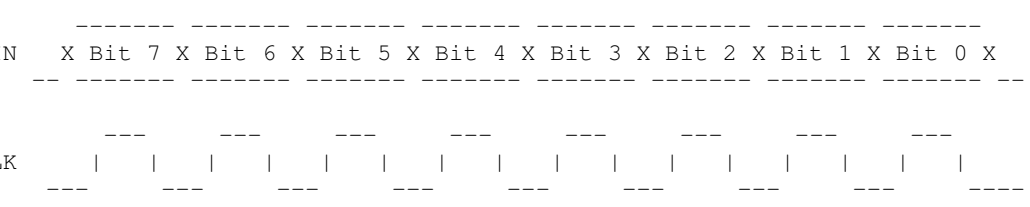
```
void MAX7219_KOMMANDO2(char Adr1, char Data1, char Adr2, char Data2)
{
    // Schritt 1: Steuerleitung LOAD loeschen
    MAX7219LOAD = 0;

    // Schritt 2: Die Adresse des ersten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Adr1);

    // Schritt 3: Die Daten des ersten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Data1);

    // Schritt 4: Die Adresse des zweiten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Adr2);

    // Schritt 5: Die Daten des zweiten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Data2);

    // Schritt 6: Steuerleitung LOAD setzen und ein positive Taktflanke erzeugen
    MAX7219LOAD = 1;
    MAX7219CLK = 1;
}
```

**Anmerkung:**
Der 1. MAX7219 ist jener, der vom PIC (Mikrocontroller) am weitesten entfernt ist. Die Adresse und die Daten für den 1. MAX7219 werden also durch den 2. MAX7219 geschoben.




# 8.3. Unterprogramm `MAX7219_SENDBYTE()`

**Aufgabe:**
Ein Byte seriell (also Bit für Bit) an den MAX7219 senden. Beginnend mit dem MSB (also Bit 7) und dazwischen je einen Takt erzeugen.




**Übergabeparameter**:
Das zu übertragende Byte (`Byte`)

**Rückgabeparameter:**
keiner

**Hier das Unterprogramm:**

```
void MAX7219_SENDBYTE(char Byte)
{
    char i;


    for (i=8; i>0; i--)
    {
        // MSB ausgeben
        MAX7219DIN = Byte.F7;

        // Takt erzeugen
        MAX7219CLK = 1;
        MAX7219CLK = 0;

        // Bits verschieben (Bit 6 -> Bit 7, Bit 5 -> Bit 6 usw.)
        Byte = Byte << 1;
    }
}
```

# 9. Demonstrationsbeispiel 3

Das folgende Beispiel dient nur zur Demonstration, und zeigt eine mögliche Einbindung der im vorigen Abschnitt beschriebenen Unterprogramme.

In diesem Beispiel soll eine einfache Uhr mit Minuten und Sekunden erzeugt werden. Zusätzlich soll ein Sekundentakt mit Hilfe von zwei Leuchtdioden (LEDs) erzeugt werden, und zur Demonstration soll auch die Helligkeit mit einem Taster verändert werden.

## 9.1. Hardware

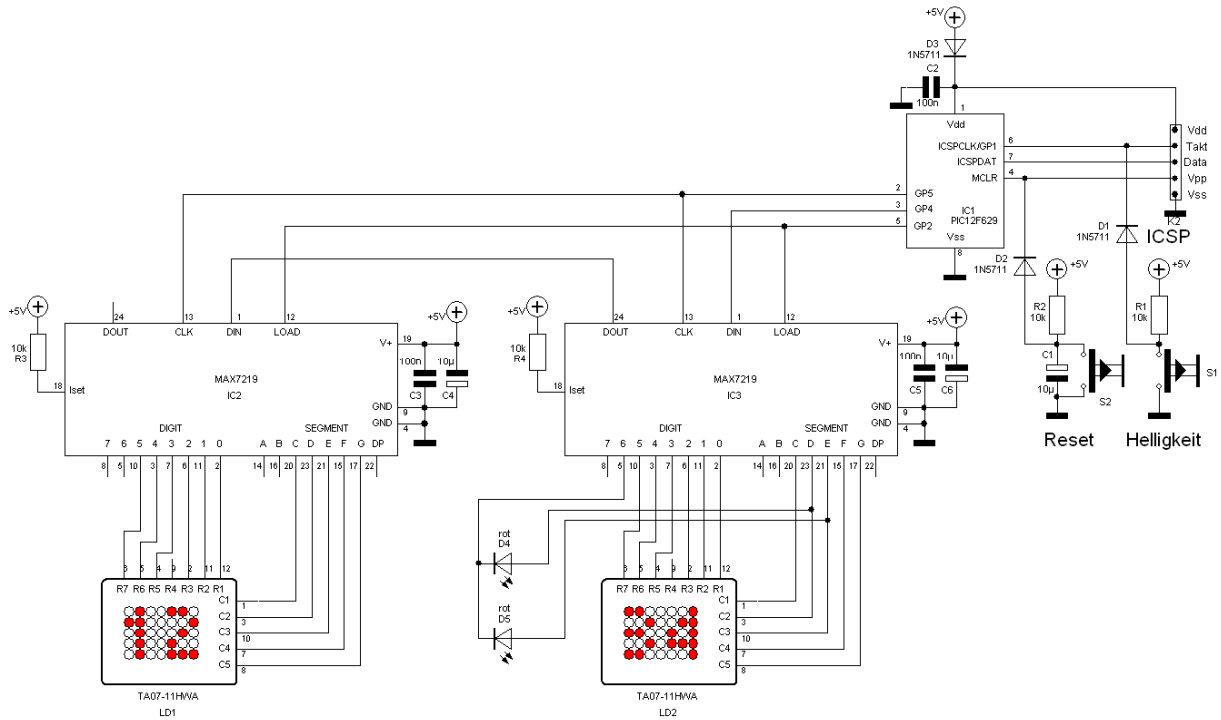



Bild 9.1: Schaltung zum Demonstrationsbeispiel 3

Bei diesem Demonstrationsbeispiel werden die beiden Dot-Matrix-Anzeigen (LD1 und LD2) „liegend“ verwendet. Daher muss für beide unbedingt der Typ TA07-11HWA verwendet werden. (Siehe auch Demonstrationsbeispiel 2, Abschnitt 6)

Neu sind bei diesem Demo die beiden Leuchtdioden D4 und D5. Diese sind an Pin 5 (Digit 6) von IC3 angeschlossen, da dieser Digit für die Dot-Matrix-Anzeigen nicht benötigt wird.

Neu ist auch der Taster S1 mit R1 als Pull-Up-Widerstand. Dieser Taster dient hier zur Veränderung der Helligkeit der Anzeigen. Als externer Widerstand für die Helligkeit dient wieder ein 10k-Widerstand (R3 und R4), wie schon beim Demonstrationsbeispiel 1 (siehe Abschnitt 5)

Als Mikrocontroller (IC1) dient auch hier ein PIC12F629. Auch hier ist die Anwendung nicht zeitkritisch, so dass der interne Oszillator verwendet werden kann.

Achtung:
Die Portbelegung zur Ansteuerung der beiden MAX7219 unterscheidet sich zu den Demonstrationsbeispielen 1 und 2, und muss auch in der Software (siehe Abschnitt 9.2.) berücksichtigt werden. (Die Steuerleitungen DIN und CLK wurden beim layouten vertauscht)

# 9.2. Software

## 9.2.1. Datei `max7219_demo3.c`

```
/*******************************************************************************/
/* Demonstrationsbeispiel 3 zum MAX7219 in C                                   */
/*                                                                             */
/* interner Takt (4 MHz)                                                       */
/*                                                                             */
/* Compiler: mikroC                                                            */
/*                                                                             */
/* Entwickler: Buchgeher Stefan                                                */
/* Entwicklungsbeginn der Software: 5. September 2008                          */
/* Funktionsfaehig seit: 11. September 2008                                    */
/* Letzte Bearbeitung: 14. September 2008                                      */
/*******************************************************************************/

/****************** Include-Dateien ********************************************/
#include "MAX7219.H"


/****************** Strukturen *************************************************/
/* keine Strukturen verwendet */


/****************** Externe Register *******************************************/
/* keine externen Register */


/****************** Bits in den externen Registern *****************************/
/* keine externen Register */


/****************** Globale Register *******************************************/
char    FLAGISRHP;
char    ZAEHLERZEITBASIS10MS;
char    ZAEHLERZEITBASIS1SEK;
char    HELLIGKEIT;


/****************** Bits in den globalen Registern *****************************/
```

```
/* Register FLAGSISRHP */
#define FLAGZEITBASIS10MS       FLAGISRHP.F0
#define FLAGZEITBASIS1SEK       FLAGISRHP.F1


/****************** Portbelegung ************************************************/
/* Siehe MAX7219.H */
/* Siehe Unterprogramm TASTENROUTINE */


/****************** Konstanten **************************************************/
// Konstanten fuer die Zeitbasen
#define KONSTZEITBASIS10MS      5
#define KONSTZEITBASIS1SEK      100


/****************** Tabellen ****************************************************/
// Tabellen fuer die Ziffern 0 bis 9 (Zeichensatz)

//    SEG
//  1  G   X     X    XX    XX      X   XXX    X    XX     X     X
//  2  F  X X   XX      X     X    XX   X     X       X   X X   X X
//  4  E  X X    X      X    XX    X X   XX    XX      X    X     XX
//  8  D  X X    X     X       X   XXX     X   X X    X    X X     X
// 16  C   X    XXX   XXX   XX      X   XX     X     X     X     X

const char TabZeichensatz3x5_sp1[10] = {14,18,25,17,12,23,14,1,10,2};
const char TabZeichensatz3x5_sp2[10] = {17,31,21,21,10,21,21,25,21,21};
const char TabZeichensatz3x5_sp3[10] = {14,16,18,10,31,9,8,6,10,14};


/****************** Funktionsprototypen *****************************************/
void INIT(void);
void TASTENROUTINE(void);
void UHR(void);


/****************** ISR - Timer0 ************************************************/

/********************************************************************************/
/* Interrupt Service Routine:                                                   */
/*                                                                              */
/* Aufruf:                                                                      */
/*      alle 2 ms (ungefaehr)                                                   */
/*                                                                              */
/* Aufgaben:                                                                    */
/*      + Zeitbasen fuer 10 Millisekunde und 1 Sekunde erzeugen                 */
/*      + Das Timer-Interrupt-Flag T0IF wieder loeschen                         */
/********************************************************************************/
void interrupt (void)                   // Interruptroutine
{
    ZAEHLERZEITBASIS10MS--;
    if (ZAEHLERZEITBASIS10MS == 0)      // Botschaftsflag fuer 10ms-Zeitbasis setzen
    {
        FLAGZEITBASIS10MS = 1;
        ZAEHLERZEITBASIS10MS = KONSTZEITBASIS10MS; // Zaehlregister fuer 10-ms-
                                        // Zeitbasis neu laden
        ZAEHLERZEITBASIS1SEK--;
        if (ZAEHLERZEITBASIS1SEK == 0)
        {
            FLAGZEITBASIS1SEK = 1;     // Botschaftsflag fuer 1sek-Zeitbasis setzen

            ZAEHLERZEITBASIS1SEK = KONSTZEITBASIS1SEK; // Zaehlregister fuer 1-sek-
        }                               // Zeitbasis neu laden
    }
    INTCON.T0IF = 0;                    // Timer-Interrupt-Flag T0IF wieder loeschen
}


/****************** Unterprogramme und Funktionen *******************************/

/********************************************************************************/
/* INIT:                                                                        */
/*                                                                              */
/* Aufgabe:                                                                     */
/*   Initialisierung des Prozessor:                                             */
/*      + Timer 0 (TMR0) loeschen                                               */
/*      + Timer 0-ISR soll ca. alle 2ms aufgerufen werden, daher den Vorteiler mit 1:8 */
/*        laden (bei einem internen 4-MHz-Takt)                                 */
```

```
/*      + Port konfigurieren                                                    */
/*      + Comparator konfigurieren (hier ausschalten)                           */
/*      + diverse Register initialisieren                                       */
/*          + Zaehlregister fuer Zeitbasen initialisieren                       */
/*          + Register fuer die Helligkeit initialisieren                       */
/*      + Interrupt freigeben (Timer0 freigeben durch Setzen von GIE und T0IE im*/
/*        INTCON-Register)                                                      */
/*                                                                              */
/* Uebergabeparameter:                                                          */
/*   keiner                                                                     */
/*                                                                              */
/* Rueckgabeparameter:                                                          */
/*   keiner                                                                     */
/********************************************************************************/
void INIT(void)
{
    // Timer-0-Interrupt konfigurieren
    TMR0 = 0;                       // Timer 0 auf 0 voreinstellen
    OPTION_REG = 0b10000010;        // Option-Register (Bank 1)
/*                  +---------------- Bit 7 (nGPPU): Pull-Up-Widerstaende
                    |||||||               0 : Pull-Up aktiviert
                    |||||||           ->  1 : Pull-Up deaktiviert
                    +---------------- Bit 6 (INTEDG): Interrupt Edge Select Bit
                     ||||||           ->  0 : Interrupt on falling edge of GP2/INT pin
                     ||||||               1 : Interrupt on rising edge of GP2/INT pin
                     +---------------- Bit 5 (T0CS): Timer 0 Clock source Select Bit
                      |||||           ->  0 : Transition on GP2/T0CKI pin
                      |||||               1 : Internal instruction cycle clock (CLKOUT)
                      +---------------- Bit 4 (T0SE): TMR0 Source Edge Select bit
                       ||||           ->  0 : Increment on low-to-high tranition
                       ||||               1 : Increment on high-to-low tranition
                       +---------------- Bit 3 (PSA): Prescaler Assignment bit
                        |||           ->  0 : Prescaler is assigned to the TIMER0 module
                        |||               1 : Prescaler is assigned to the WDT
                        +++-------------- Bit 2-0 (PS2:PS0): Prescaler Rate Select bits
                                            Bit Value        TMR0 Rate         WDT Rate
                                           000 :             1:2               1:1
                                           001 :             1:4               1:2
                                       ->  010 :             1:8               1:4
                                           011 :             1:16              1:8
                                           100 :             1:32              1:16
                                           101 :             1:64              1:32
                                           110 :             1:128             1:64
                                           111 :             1:256             1:128
*/

    // Port konfigurieren
    TRISIO = 0b00000010;            // Richtungsregister Port (0: Ausgang, 1: Eingang)
/*              ++------------------ Bit 7-6 Reserve
                  +----------------- Bit 5 Port GP5: hier Ausgang (Takt CLK)
                   +---------------- Bit 4 Port GP4: hier Ausgang (Data DIN)
                    +--------------- Bit 3 Port GP3: hier unbenutzt
                     +-------------- Bit 2 Port GP2: hier Ausgang (LOAD)
                      +------------- Bit 1 Port GP1: hier Eingang (Taster Helligkeit)
                       +------------ Bit 0 Port GP0: hier unbenutzt
*/

    // Comparator konfigurieren
    CMCON = 0b00000111;             // Comparator Control-Register (Bank 0)
/*             +-+------------------- Bit 7,5 : Reserve
                +-------------------- Bit 6 (COUT): Comparator Output bit
                  |||||                   When CINV = 0:
                  |||||               ->    0 : Vin+ < Vin-
                  |||||                     1 : Vin+ > Vin-
                  |||||                 When CINV = 1:
                  |||||                   0 : Vin+ > Vin-
                  |||||                   1 : Vin+ < Vin-
                  +------------------ Bit 4 (CINV): ComparatorOutput Inversion Bit
                   ||||               ->  0 : Output not inverted
                   ||||                   1 : Output inverted
                   +----------------- Bit 3 (CIS): Comparator Input Switch Bit
                    |||                   When CM2:CM0 = 110 or 101:
                    |||               ->  0 : Vin- connects to CIN-
                    |||                   1 : Vin- connects to CIN+
                    +++--------------- Bit 2-0 (CM2:CM0): Comp. Mode bits (Datenbl. S.37)
                                       000 : Comparator Reset
                                       001 : Comparator with Output
                                       010 : Comparator without Output
                                       011 : Comp. with Output and Internal Ref.
```

```
                                        100 : Comp. w/o Output and Internal Ref.
                                        101 : Multiplexed Input with Int. Ref and Outp.
                                        110 : Multiplexed Input with Int. Ref
                                     ->  111 : Off
*/

    // Zaehlregister fuer Zeitbasen initialisieren
    ZAEHLERZEITBASIS10MS = KONSTZEITBASIS10MS; // Zaehlregister fuer 100-ms-Zeitbasis
                                               // initialisieren
    ZAEHLERZEITBASIS1SEK = KONSTZEITBASIS1SEK; // Zaehlregister fuer 1-sek-Zeitbasis
                                               // initialisieren

    // Register fuer die Helligkeit initialisieren
    HELLIGKEIT = 8;                 //Mittlere Helligkeit

    // Interrupt freigeben
    INTCON = 0b10100000;            // Interrupt-Control-Register (Bank 1)
/*              +-------------------- Bit 7 (GIE): Global Interrupt Enable bit
                ||||||||                 0 : Disables all interrupts
                ||||||||              ->   1 : Enables all unmasked interrupts
                +-------------------- Bit 6 (PEIE): Peripheral Interrupt Enable bit
                 ||||||               ->   0 : Disables all peripheral interrupts
                 ||||||                    1 : Enables all unmasked peripheral interrupts
                 +------------------- Bit 5 (T0IE): Timer 0 Overflow Interrupt Enable Bit
                  |||||                    0 : Disabled
                  |||||               ->   1 : Enabled
                  +------------------ Bit 4 (INTE): GP2/INT External Interrupt Enable Bit
                   ||||               ->   0 : Disabled
                   ||||                    1 : Enabled
                   +----------------- Bit 3 (RPIE): Port Change Interrupt Enable Bit
                    |||               ->   0 : Disabled
                    |||                    1 : Enabled
                    +---------------- Bit 2 (T0IF): Timer 0 Overflow Interrupt Flag bit
                     ||               ->   0 : TMR0 register did not overflow
                     ||                    1 : TMR0 register has overflowed
                     +--------------- Bit 1 (INTF): GP2/INT External Interrupt Flag bit
                      |               ->   0 : GP2/INT external interrupt did not occur
                      |                    1 : GP2/INT external interrupt occurred
                      +-------------- Bit 0 (RBIF): Port Change Interrupt Flag bit
                                      ->   0 : None of GP5:GP0 pins have changed state
                                           1 : One of the GP5:GP0 pins changed state
*/
}


/*****************************************************************************************/
/* TASTENROUTINE:                                                                        */
/*                                                                                       */
/* Aufruf:                                                                               */
/*   alle 10 ms (vom Hauptprogramm)                                                      */
/*                                                                                       */
/* Aufgabe:                                                                              */
/*   Ermitteln, ob die Taste gedrueckt wurde, und das Register HELLIGKEIT erhoehen,      */
/*   oder loeschen, wenn das Register HELLIGKEIT den Wert 16 (oder mehr) besitzt.        */
/*                                                                                       */
/* Portbelegung (Taster):                                                                */
/*   GP1: Helligkeit                                                                     */
/*                                                                                       */
/* Vorgehensweise:                                                                       */
/*   + Port einlesen und im Register temp1 sichern                                       */
/*   + dieses Register invertieren und in temp2 sichern (temp1 wird spaeter noch         */
/*     einmal benoetigt und darf daher nicht ueberschrieben werden!)                     */
/*   + Das Register TASTENALT gibt beim Aufruf (dieses Unterprogramms) den Zustand der   */
/*     Taste an, welcher beim vorhergehenden Aufruf dieses Unterprogramms am Port        */
/*     herrschte. Also den Zustand am Port vor 10 ms.                                    */
/*   + Mit Hilfe der beiden Register temp2 und TASTENALT laesst sich somit ermitteln,    */
/*     ob eine bestimmte Taste zwischen dem vorhergehenden Aufruf dieses Unter-          */
/*     programms und den gerade bearbeitenden Aufruf gedrueckt wurde. Dies erfolgt mit   */
/*     einer bitweisen logischen-UND-Verknuepfung. Das Ergebnis dieser UND-Verknuepfung  */
/*     im Register TASTENSTATUS sichern.                                                 */
/*     Mit diesem Verfahren wird auch gleichzeitig das sogenannte Tastenprellen ueber-   */
/*     brueckt, da dieses Tastenprellen in der Regel weit weniger als 10ms dauert.       */
/*   + Je nachdem welche Taste nun gedrueckt wurde, das entsprechende Bit im Register    */
/*     TASTENSTATUS ist gesetzt, diese Kerze ein bzw. ausschalten (also invertieren).    */
/*   + Register TASTENALT mit dem Inhalt von temp1 fuer den naechsten Unterprogramm-     */
/*     aufruf laden.                                                                     */
/*                                                                                       */
/* Uebergabeparameter:                                                                   */
/*   keiner                                                                              */
```

```
/*                                                                                    */
/* Rueckgabeparameter:                                                                */
/*   keiner                                                                           */
/*                                                                                    */
/* Anmerkungen:                                                                       */
/*   + Die Taste ist am Port mit einem Pull-up-Widerstand angeschlossen. Im Ruhe-     */
/*     zustand (Taste nicht gedrueckt) liegt an dem Portpin ein High-Pegel. Durch     */
/*     Druecken der Taste wird dieser Portpin auf Masse gelegt (Low-Pegel). Dieses    */
/*     Verhalten wird auch als low-aktiv bezeichnet.                                  */
/*   + Das so genannte Tastenprellen wird hier umgangen (Die Prellzeit ist kleiner als */
/*     das Aufrufintervall von 10ms)                                                  */
/**************************************************************************************/
void TASTENROUTINE(void)
{
    static char TASTEALT = 0xFF;
    static char TASTESTATUS = 0;
    char temp1;
    char temp2;


    temp1 = GPIO;

    temp2 = ~temp1;                 // Alle Bits invertieren

    TASTESTATUS = TASTEALT & temp2;

    if (TASTESTATUS.F1)             // Taste Helligkeit gedrueckt
    {
        HELLIGKEIT++;
        if (HELLIGKEIT > 15)
        {
            HELLIGKEIT = 0;
        }

        // Helligkeit an beide MAX7219 senden
        MAX7219_KOMMANDO2(0x0a, HELLIGKEIT, 0x0a, HELLIGKEIT);
    }
    TASTEALT = temp1;
}


/**************************************************************************************/
/* UHR:                                                                               */
/*                                                                                    */
/* Aufruf:                                                                            */
/*   jede Sekunde (vom Hauptprogramm)                                                 */
/*                                                                                    */
/* Aufgabe:                                                                           */
/*   erzeugt eine freilaufende Uhr (nur Minuten und Sekunden) und gibt diese mit Hilfe */
/*   von zwei (kaskadierten) MAX7219 an zwei 5x7-Dot-Matrix-Anzeigen aus. Zusaetzlich  */
/*   wird ein Sekundentakt mit Hilfe von zwei Leuchtdioden angezeigt.                 */
/*                                                                                    */
/* Vorgehensweise:                                                                    */
/*   + Uhrzeit ermitteln                                                              */
/*   + Einer und Zehnerstellen fuer Minuten und Sekunden ermitteln                    */
/*   + Uhrzeit an die beiden MAX7219 senden und an den beiden 5x7-Dot-Matrix-Anzeigen */
/*     ausgeben.                                                                      */
/*   + Sekundentakt aus dem LSB des Registers fuer die Sekunde erzeugen und mit den   */
/*     beiden Leuchtdioden anzeigen (via MAX7219)                                     */
/*                                                                                    */
/* Anmerkungen:                                                                       */
/*   + Ist der Wert der Minute <10 die fuehrende Null (Zehnerstelle) nicht anzeigen   */
/*   + Achtung: Die Datenbits sind gemaess dem Datenblatt zum MAX7219 wie folgt den   */
/*     Segmenten zugeordnet:                                                          */
/*                            Registerdaten:  D7  D6  D5  D4  D3  D2  D1  D0          */
/*                            Segment:        DP  A   B   C   D   E   F   G           */
/*                                                                                    */
/*                          SEG                                                       */
/*                         1  G      X     XX            XX       X                  */
/*                         2  F     XX       X             X     XX                  */
/*                         4  E      X      X      X      XX    X X                  */
/*                         8  D      X     X       X       X    XXX                  */
/*                        16  C     XXX    XXX           XX       X                  */
/*                                                                                    */
/*                         Digit    012    345     6     012    345                  */
/*                                  \_____________/     \________/                  */
/*                                     LD1 / IC2          LD2 / IC3                  */
/*                                                                                    */
/*   + Die Dekodierung der Ziffern in die Registerdaten erfolgt mit Hilfe von drei    */
```

```
/*     Tabellen, fuer jede Ziffern-Spalte eine eigene Tabelle.                    */
/*********************************************************************************/
void UHR(void)
{
    static char UHR_SEKUNDEN = 0;
    static char UHR_MINUTEN = 0;
    char    UHR_SEK_EINER;
    char    UHR_SEK_ZEHNER;
    char    UHR_MIN_EINER;
    char    UHR_MIN_ZEHNER;
    char    hlpAUSGABE_SEK;
    char    hlpAUSGABE_MIN;


    // Schritt 1: Uhrzeit ermitteln
    UHR_SEKUNDEN++;
    if (UHR_SEKUNDEN > 59)
    {
        UHR_SEKUNDEN = 0;
        UHR_MINUTEN++;
        if (UHR_MINUTEN > 59)
        {
            UHR_MINUTEN = 0;
        }
    }

    // Schritt 2: Einer und Zehnerstellen fuer Minuten und Sekunden ermitteln
    UHR_MIN_ZEHNER = (char)(UHR_MINUTEN / 10);
    UHR_MIN_EINER = UHR_MINUTEN % 10;

    UHR_SEK_ZEHNER = (char)(UHR_SEKUNDEN / 10);
    UHR_SEK_EINER = UHR_SEKUNDEN % 10;

    // Schritt 3: Uhrzeit ausgeben
    // Schritt 3a: Zehnerstellen
    if (UHR_MIN_ZEHNER == 0)
    {
        // Digit 0
        hlpAUSGABE_SEK = TabZeichensatz3x5_sp1[UHR_SEK_ZEHNER];
        MAX7219_KOMMANDO2(0x01, 0, 0x01, hlpAUSGABE_SEK);
        // Digit 1
        hlpAUSGABE_SEK = TabZeichensatz3x5_sp2[UHR_SEK_ZEHNER];
        MAX7219_KOMMANDO2(0x02, 0, 0x02, hlpAUSGABE_SEK);
        // Digit 2
        hlpAUSGABE_SEK = TabZeichensatz3x5_sp3[UHR_SEK_ZEHNER];
        MAX7219_KOMMANDO2(0x03, 0, 0x03, hlpAUSGABE_SEK);
    }
    else
    {
        // Digit 0
        hlpAUSGABE_MIN = TabZeichensatz3x5_sp1[UHR_MIN_ZEHNER];
        hlpAUSGABE_SEK = TabZeichensatz3x5_sp1[UHR_SEK_ZEHNER];
        MAX7219_KOMMANDO2(0x01, hlpAUSGABE_MIN, 0x01, hlpAUSGABE_SEK);
        // Digit 1
        hlpAUSGABE_MIN = TabZeichensatz3x5_sp2[UHR_MIN_ZEHNER];
        hlpAUSGABE_SEK = TabZeichensatz3x5_sp2[UHR_SEK_ZEHNER];
        MAX7219_KOMMANDO2(0x02, hlpAUSGABE_MIN, 0x02, hlpAUSGABE_SEK);
        // Digit 2
        hlpAUSGABE_MIN = TabZeichensatz3x5_sp3[UHR_MIN_ZEHNER];
        hlpAUSGABE_SEK = TabZeichensatz3x5_sp3[UHR_SEK_ZEHNER];
        MAX7219_KOMMANDO2(0x03, hlpAUSGABE_MIN, 0x03, hlpAUSGABE_SEK);
    }

    // Schritt 3b: Einerstellen
    // Digit 3
    hlpAUSGABE_MIN = TabZeichensatz3x5_sp1[UHR_MIN_EINER];
    hlpAUSGABE_SEK = TabZeichensatz3x5_sp1[UHR_SEK_EINER];
    MAX7219_KOMMANDO2(0x04, hlpAUSGABE_MIN, 0x04, hlpAUSGABE_SEK);
    // Digit 4
    hlpAUSGABE_MIN = TabZeichensatz3x5_sp2[UHR_MIN_EINER];
    hlpAUSGABE_SEK = TabZeichensatz3x5_sp2[UHR_SEK_EINER];
    MAX7219_KOMMANDO2(0x05, hlpAUSGABE_MIN, 0x05, hlpAUSGABE_SEK);
    // Digit 5
    hlpAUSGABE_MIN = TabZeichensatz3x5_sp3[UHR_MIN_EINER];
    hlpAUSGABE_SEK = TabZeichensatz3x5_sp3[UHR_SEK_EINER];
    MAX7219_KOMMANDO2(0x06, hlpAUSGABE_MIN, 0x06, hlpAUSGABE_SEK);

    // Schritt 4: Sekundentakt ausgeben (Digit 6)
    if (UHR_SEKUNDEN.F0)
```

```
    {
        MAX7219_KOMMANDO2(0x07, 0, 0x07, 12);
    }
    else
    {
        MAX7219_KOMMANDO2(0x07, 0, 0x07, 0);
    }
}


/****************** Hauptprogramm ************************************************/

/*****************************************************************************/
/* Aufgaben des Hauptprogramms:                                              */
/*   + Controller initialisieren (Unterprogramm INIT)                        */
/*   + Beide MAX7219 initialisieren (Unterprogramm MAX7219_INIT2)            */
/*       + Scanlimit: 06h da nur 7 Digits verwendet werden                   */
/*       + Shutdown-Format: 01h fuer Normal Operation                        */
/*       + Decode-Mode: 00h fuer No Decode                                   */
/*       + Intensity: Register HELLIGKEIT (hier 8, also mittlere Helligkeit) */
/*   + Taetigkeiten/Unterprogramme, die alle zyklisch durchgefuehrt werden muessen:  */
/*     + alle 10 ms: Taste abfragen (Unterprogramm TASTENROUTINE)            */
/*     + jede Sekunde: Sekunde erhoehen, und neue Uhrzeit ermitteln (Unterprogramm  */
/*       UHR)                                                                */
/*****************************************************************************/
void main(void)
{
    INIT();                             // Controller initialisieren

    MAX7219_INIT2(0x06, 0x01, 0x00, HELLIGKEIT); // MAX7219 initialisieren
/*                +------------------------ Scanlimit: 7 Digits
                        +------------------- Shutdown-Format: Normal Operation
                              +------------- Decode-Mode: No Decode
                                    +------ Intensity: Register HELLIGKEIT
*/

    while(1)                            // Endlosschleife
    {
        if (FLAGZEITBASIS10MS)
        {
            TASTENROUTINE();
            FLAGZEITBASIS10MS = 0;      // Botschaftsflag (fuer 10ms-Zeitbasis)
        }                               // zuruecksetzen

        if (FLAGZEITBASIS1SEK)
        {
            UHR();
            FLAGZEITBASIS1SEK = 0;      // Botschaftsflag (fuer 1sek-Zeitbasis)
        }                               // zuruecksetzen
    }
}
```

## 9.2.2. Datei MAX7219.C

```
/*****************************************************************************/
/* Unterprogramme zur Kommunikation mit dem MAX7219                          */
/*                                                                           */
/* Compiler: mikroC                                                          */
/*                                                                           */
/* Entwickler: Buchgeher Stefan                                              */
/* Entwicklungsbeginn der Software: 5. September 2008                        */
/* Funktionsfaehig seit: 7. September 2008                                   */
/* Letzte Bearbeitung: 14. September 2008                                    */
/*****************************************************************************/

/****************** Include-Dateien ******************************************/
#include "MAX7219.H"


/****************** Unterprogramme fuer zwei MAX7219 (Kaskadiert) *************/

/*****************************************************************************/
/* MAX7219_INIT2:                                                            */
/*                                                                           */
/* Aufgabe:                                                                  */
/*   Die beiden kaskadierten MAX7219 initialisieren:                         */
```

```
/*                                                                                    */
/* Vorgehensweise:                                                                    */
/*   Uebergibt nacheinander die Befehle zur Initialisierung an beide MAX7219 (mit     */
/*   Hilfe des Unterprogramms MAX7219_KOMMANDO2()                                     */
/*     + Adresse 0Bh: Scan Limit                                                      */
/*     + Adresse 0Ch: Shutdown Format                                                 */
/*     + Adresse 09h: Decode Mode                                                     */
/*     + Adresse 0Ah: Intensity                                                       */
/*                                                                                    */
/* Uebergabeparameter:                                                                */
/*   + Scanlimit                                                                      */
/*   + ShutdownFormat                                                                 */
/*   + DecodeMode                                                                     */
/*   + Intensity                                                                      */
/*                                                                                    */
/* Rueckgabeparameter:                                                                */
/*   keiner                                                                           */
/************************************************************************************/
void MAX7219_INIT2(char Scanlimit, char ShutdownFormat, char DecodeMode, char Intensity)
{
    // Scanlimit an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x0b, Scanlimit, 0x0b, Scanlimit);

    // Shutdown-Format an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x0c, ShutdownFormat, 0x0c, ShutdownFormat);

    // Decode-Mode an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x09, DecodeMode, 0x09, DecodeMode);

    // Intensity an beide MAX7219 senden
    MAX7219_KOMMANDO2(0x0a, Intensity, 0x0a, Intensity);
}


/************************************************************************************/
/* MAX7219_KOMMANDO2:                                                               */
/*                                                                                  */
/* Aufgabe:                                                                         */
/*   Dieses Programm sendet nacheinander zwei Kommandos an die beiden MAX7219. (Fuer */
/*   jeden MAX7219 ein Kommando). Ein Kommando besteht dabei aus acht Adressbits    */
/*   (A7 - A0) und aus acht Datenbits (D7 - D0)                                     */
/*   +----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+ */
/*   | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | */
/*   +----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+ */
/*    MSB                                                                   LSB       */
/*                                                                                    */
/* Vorgehensweise:                                                                  */
/*   + Steuerleitung LOAD loeschen                                                  */
/*   + Die Adresse des ersten MAX7219 (Uebergabeparameter Adr1) mit Hilfe eine Hilfs- */
/*     unterprogramms (MAX7219_SENDBYTE) an den MAX7219 senden                      */
/*   + Die Daten des ersten MAX7219 (Uebergabeparameter Data1) mit dem gleichen Hilfs- */
/*     unterprogramm (MAX7219_SENDBYTE) an den MAX7219 senden                       */
/*   + Die Adresse des zweiten MAX7219 (Uebergabeparameter Adr2) mit dem gleichen   */
/*     Hilfsunterprogramm (MAX7219_SENDBYTE) an den MAX7219 senden                  */
/*   + Die Daten des zweiten MAX7219 (Uebergabeparameter Data2) mit dem gleichen Hilfs-*/
/*     unterprogramm (MAX7219_SENDBYTE) an den MAX7219 senden                       */
/*   + Steuerleitung LOAD wieder setzen, und eine positive Taktflanke erzeugen      */
/*                                                                                  */
/* Uebergabeparameter:                                                              */
/*   + Adresse des 1. MAX7219 (Adr1)                                                */
/*   + Daten des 1. MAX7219 (Data1)                                                 */
/*   + Adresse des 2. MAX7219 (Adr2)                                                */
/*   + Daten des 2. MAX7219 (Data2)                                                 */
/*                                                                                  */
/* Rueckgabeparameter:                                                              */
/*   keiner                                                                         */
/*                                                                                  */
/* Anmerkung:                                                                       */
/*   Der 1. MAX7219 ist jener, der vom PIC (Mikrocontroller am weitesten entfernt ist. */
/*   Die Adresse und die Daten fuer den 1. MAX7219 werden also durch den 2. MAX7219 */
/*   geschoben.                                                                     */
/*           +----------+     +----------+     +----------+                       */
/*           |1. MAX7219+-----+2. MAX7219+-----+   PIC    |                       */
/*           +----------+     +----------+     +----------+                       */
/************************************************************************************/
void MAX7219_KOMMANDO2(char Adr1, char Data1, char Adr2, char Data2)
{
    // Schritt 1: Steuerleitung LOAD loeschen
    MAX7219LOAD = 0;
```

```
    // Schritt 2: Die Adresse des ersten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Adr1);

    // Schritt 3: Die Daten des ersten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Data1);

    // Schritt 4: Die Adresse des zweiten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Adr2);

    // Schritt 5: Die Daten des zweiten MAX7219 bitweise senden
    MAX7219_SENDBYTE(Data2);

    // Schritt 6: Steuerleitung LOAD setzen und ein positive Taktflanke erzeugen
    MAX7219LOAD = 1;
    MAX7219CLK = 1;
}


/****************** Hilfsunterprogramm *************************************************/

/***********************************************************************************/
/* MAX7219_SENDBYTE:                                                               */
/*                                                                                 */
/* Aufgabe:                                                                        */
/*   Ein Byte seriell (also Bit fuer Bit) an den MAX7219 senden. Beginnend mit dem MSB */
/*   (also Bit 7) und dazwischen je einen Takt erzeugen.                           */
/*                 _______ _______ _______ _______ _______ _______ _______ _______ */
/*   MAX7219DIN   X Bit 7 X Bit 6 X Bit 5 X Bit 4 X Bit 3 X Bit 2 X Bit 1 X Bit 0 X */
/*              -- _______ _______ _______ _______ _______ _______ _______ _______ -- */
/*                                                                                 */
/*                   ___     ___     ___     ___     ___     ___     ___     ___    */
/*   MAX7219CLK     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |      */
/*                ___     ___     ___     ___     ___     ___     ___     ___     ____ */
/*                                                                                 */
/* Uebergabeparameter:                                                             */
/*   Das zu uebertragende Byte (Byte)                                              */
/*                                                                                 */
/* Rueckgabeparameter:                                                             */
/*   keiner                                                                        */
/***********************************************************************************/
void MAX7219_SENDBYTE(char Byte)
{
    char i;


    for (i=8; i>0; i--)
    {
        // MSB ausgeben
        MAX7219DIN = Byte.F7;

        // Takt erzeugen
        MAX7219CLK = 1;
        MAX7219CLK = 0;

        // Bits verschieben (Bit 6 -> Bit 7, Bit 5 -> Bit 6 usw.)
        Byte = Byte << 1;
    }
}
```

## 9.2.3. Datei `MAX7219.H`

```
/***********************************************************************************/
/* Header fuer die Unterprogramme zur Kommunikation mit dem MAX7219                */
/*                                                                                 */
/* Compiler: mikroC                                                                */
/*                                                                                 */
/* Entwickler: Buchgeher Stefan                                                    */
/* Entwicklungsbeginn der Software: 5. September 2008                              */
/* Funktionsfaehig seit: 7. September 2008                                         */
/* Letzte Bearbeitung: 14. September 2008                                          */
/***********************************************************************************/

/****************** Portbelegung ***************************************************/
#define MAX7219DIN              GPIO.F4
#define MAX7219LOAD             GPIO.F2
```

```
#define MAX7219CLK              GPIO.F5


/***************** Funktionsprototypen ************************************************/
// Zwei MAX7219 hintereinander (kaskadiert)
void MAX7219_INIT2(char Scanlimit, char ShutdownFormat, char DecodeMode, char Intensity);
void MAX7219_KOMMANDO2(char Adr1, char Data1, char Adr2, char Data2);

// Hilfsunterprogramm
void MAX7219_SENDBYTE(char Byte);
```

# 9.3. Anmerkungen zur Software

Die Software besteht bei diesem Demonstrationsbeispiel aus drei Dateien. Die in Abschnitt 8 besprochenen Unterprogramme zur Kommunikation mit dem MAX7219 befinden sich in der Datei `MAX7219.C` (Abschnitt 9.2.2.) und werden hier daher nicht mehr näher erläutert. Die Datei `MAX7219.H` (Abschnitt 9.2.3.) beinhaltet die dazugehörigen Funktionsprototypen und die Portdefinitionen für den MAX7219.

Die Anwendung, also das „eigentliche“ Demonstrationsbeispiel befindet sich in der Datei `max7219_demo3.c` (Abschnitt 9.2.1.) und hat folgende Aufgaben:

1) Den Mikrocontroller (hier den PIC12F629) initialisieren
2) Jede Sekunde das Sekundenregister erhöhen und daraus die Uhrzeit (Minuten und Sekunden) ermitteln. Diese Uhrzeit mit Hilfe von Tabellen so an die beiden MAX7219 senden, dass aus der Uhrzeit lesbare Ziffern entstehen, die die Uhrzeit wiedergeben. Zusätzlich noch einen Sekundentakt an den beiden Leuchtdioden (D4 und D5) ausgeben.
3) In regelmäßigen Abständen (hier ca. alle 10 ms) überprüfen, ob der Taster (S1) gedrückt wurde, und wenn ja die Helligkeit der Anzeigen erhöhen. (Also das Register Intensity, des MA7219, siehe Abschnitt 3) um eins erhöhen und an die beiden MAX7219 senden. Nach der maximalen Helligkeit, wieder mit der minimalsten Helligkeit beginnen.

Die Software besteht daher aus den folgenden Programmteilen, die im Quellcode ausreichend kommentiert sind:

- kurzes Hauptprogramm `main()`
- eine kurze Interrupt-Service-Routine (kurz ISR) `interrupt()`
- Unterprogramm `INIT()` zur Initialisierung des Mikrocontrollers
- Unterprogramm `UHR()` für den obigen Punkt 2
- Unterprogramm `TASTENROUTINE()` für den obigen Punkt 3

Ein kleiner Nachteil beim mikroC-Compiler ist, dass die Konfigurationsbits in der IDE gesetzt werden müssen (und leider nicht im Quellcode). Hier sind die notwendigen Einstellungen für dieses Demonstrationsbeispiel:

- ☑ _CPD_OFF
- ☑ _CP_OFF
- ☑ _BODEN_ON
- ☑ _MCLRE_ON
- ☑ _PWRTE_ON
- ☑ _WDT_OFF
- ☑ _INTRC_OSC_NOCLKOUT

# 10. Quellen


- Datenblatt des 8-Digit LED Display Driver MAX7219