

Ergänzung zum Beitrag in FA 1/16, S. 36 f. „Achtstellige Siebensegment-LED- Anzeige mit SPI-Schnittstelle (1)“

Ich bin ein Freund von Beispielen. Also erläutere ich hier kurz, wie die Firmware-basierte Version des Displaytreibers mit einem PIC16F628A verwendet werden kann und wie die durchzuführenden Einzelschritte aussehen.

Wir nehmen an, dass die ersten drei Bits des PortB für den Anschluss des Displaytreibers vorgesehen sind:

RB0 → LOAD

RB1 → CLK

RB2 → DIN

Schritt 1

Im Displaytreiber selbst (Datei *AMS_DDriver_SW_SPI_v1p09.inc*) sind zuerst die Definitionen der zum Displaymodul und somit zum AMS-Schaltkreis führenden Anschlüsse anzupassen.

```
; AMS to PIC Connection Definition
; connection of the AMS chip to PIC:
; AMS                      Microcontroller
; LOAD/CSN (pin 12)        RB0
; CLK (pin 13)             RB1
; DIN (pin 1)              RB2
; definition of the connections
#define AMS_LOAD    PORTB,D'000'
#define AMS_CLK     PORTB,D'001'
#define AMS_DIN     PORTB,D'002'
; definition of the corresponding
; TRISx / ANSELx registers
#define AMS_TRIS1    TRISB,D'000'
#define AMS_TRIS2    TRISB,D'001'
#define AMS_TRIS3    TRISB,D'002'
#define AMS_ANSEL1   ANSELA,D'002'
#define AMS_ANSEL2   ANSELA,D'001'
#define AMS_ANSEL3   ANSELA,D'000'
```

Da unser Mikrocontroller kein ANSEL-Register beinhaltet, können wir die drei ANSEL-Definitionen einfach ignorieren und im Originalzustand lassen.

Schritt 2

Jetzt ist zu prüfen, ob die Variablendefinition für den verwendeten Mikrocontroller passt – was hier der Fall ist. Der PIC16F628A verfügt unter anderem über adressierbare Register in der Bank 0 auf den Adressen 20h bis 7Fh. Die standardmäßige Definition der Treibervariablen auf den Adressen 40h bis 4Dh ist also ohne weiter verwendbar.

Genauso muss sichergestellt werden, dass das Hauptprogramm die vom Displaytreiber verwendeten Speicherplätze nicht selber nutzt. Das Demo-Programm tut es nicht, also ist alles in Ordnung. Auch an dieser Stelle sind keine Änderungen notwendig.

Schritt 3

Im nächsten Schritt ist die *Include*-Datei des Displaytreibers als vorletzte Zeile in das Hauptprogramm einzubinden.

```
; ...
INCLUDE "AMS_DDriver_SW2_SPI_v1p09.inc"
END
```

Und das war es auch schon. Das komplette Hauptprogramm *AMS_SW2_Demo2_v1p09.asm* dieses Beispiels ist zusammen mit dem eingangs genannten Displaytreiber *AMS_DDriver_SW2_SPI_v1p09.inc* auch im Download-Bereich auf www.funkamateur.de verfügbar. Es führt die bereits im Beitrag genannten Aktionen in einer Endlosschleife aus.

miroslav.cina@t-online.de

Tabelle 1: Code B Fonts des AS110x

Zeichen	Register Data ¹⁾ D3D2D1D0	Segmente ²⁾ a b c d e f g						
		a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	1	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0	1	1	1	1	1	1	1
9	1 0 0 1	1	1	1	1	0	1	1
--	1 0 1 0	0	0	0	0	0	0	1
E	1 0 1 1	1	0	0	1	1	1	1
H	1 1 0 0	0	1	1	0	1	1	1
L	1 1 0 1	0	0	0	1	1	1	0
P	1 1 1 0	1	1	0	0	1	1	1
leer	1 1 1 1	0	0	0	0	0	0	0

¹⁾ Bit D7 bis D4 nicht verwendet/signifikant

²⁾ Dezimalpunkt unberücksichtigt

**Tabelle 2: Zeichencodierung
im selbst erstellten Displaytreiber**

Code	Zeichen	Code	Zeichen
00h	0	,2'	2
01h	1	,3'	3
02h	2	,4'	4
03h	3	,5'	5
04h	4	,6'	6
05h	5	,7'	7
06h	6	,8'	8
07h	7	,9'	9
08h	8	,A'	A
09h	9	,b'	b
0Ah	A	,C'	C
0Bh	b	,d'	d
0Ch	C	,E'	E
0Dh	d	,F'	F
0Eh	E	,L'	L
0Fh	F	,U'	U
10h	¹⁾	,I'	I
11h	¹⁾	,-'	-
12h	¹⁾	, '2)	2)
13h	¹⁾	,P'	P
14h	¹⁾	,S'	S/5
15h	¹⁾	,i'	i
16h	¹⁾	,r'	r
,0'	0	,t'	t
,1'	1	,o'	o

¹⁾ Spezialzeichen

²⁾ Leerzeichen