

Einführung MPLAB X

Erstes ASM-Programm

1

Erstes Programm

🔗 Verwendete HW

- 🔗 Demoboard mit PICKit3
- 🔗 Prozessor 16F886

🔗 I. Neues Projekt erstellen

- 🔗 MPLAB-X öffnen
- 🔗 Projekt anlegen

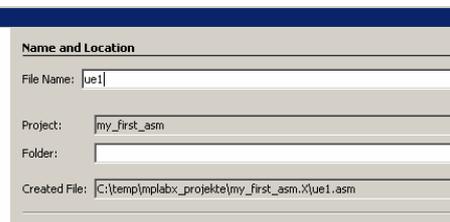
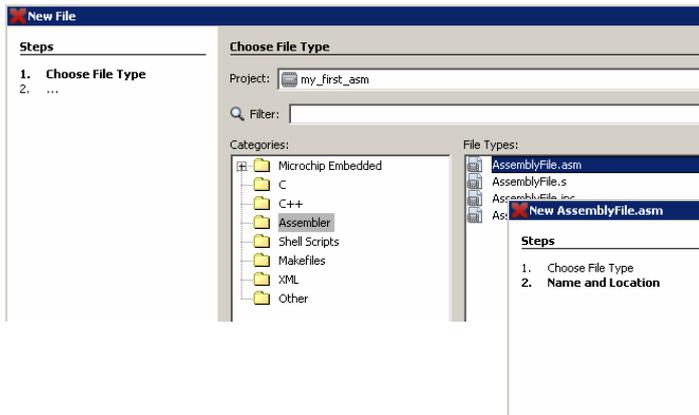
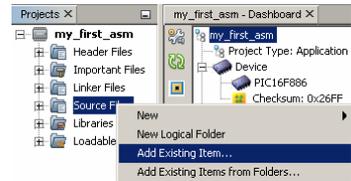
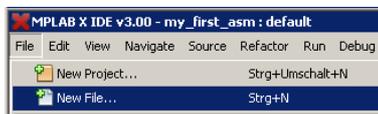
The image displays four sequential screenshots of the MPLAB X IDE's 'New Project' wizard, connected by red dashed arrows. The first screenshot shows the 'New Project' dialog with 'New Project...' selected. The second shows the 'Select Compiler' dialog where 'mpasm (v5.62)' is chosen. The third shows the 'Select Device' dialog with 'PIC16F886' selected. The fourth shows the 'Select Project Name and Folder' dialog with 'my_first_asm' as the project name and a folder path in 'C:\temp\mplabx_projekte\my_first_asm\2'.

2

Erstes Programm

II. ASM File erstellen

- a) altes Programm modifizieren
- b) oder Neues erstellen



Erstes Programm (ASM File erstellen)

II. ASM File erstellen

- a) altes Programm modifizieren
- b) oder Neues erstellen

; ... Kommentare

Header dient nur der Definition des Druckerausgabeformates

#include <p16f886> Definition aller Prozessorregister (Namen und Adresszuweisung)

Prozessor-Konfiguration

ORG 0008 Direktive, die folgenden Befehle werden ab Adresse 0008 geschrieben

clrf PORTA
löscht Register PORTA

movlw b'00001111' (Format: 0x0f/ .15 / 'y')

ins „working“ wird die Konstante Zahl geschrieben

bsf STATUS,RP0 (setzt 1 Bit, hier Bankwechsel)

movwf TRISA
working ins Register TRISA schreiben

end (zeigt das Ende des Programm an !!)

```

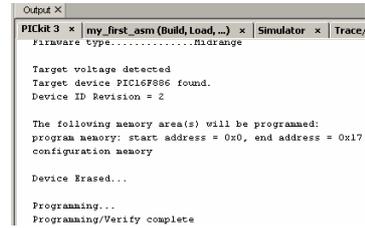
1  LIST           "UE1.asm"
2  LIST          C=135      ;number of columns
3  LIST          N=65      ;number of lines
4  LIST         p=16F886
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```

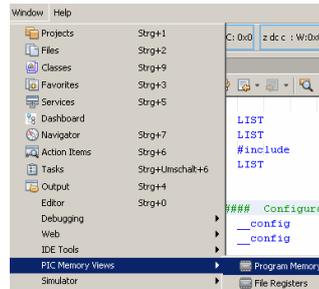
Erstes Programm (Programmierung)

III. HEX-Code erstellen

- Make and Program Device
- Kontrolle Output
- Ansicht Programm- Memory



Line	Address	Opcode	Label	DisAssey
1	0000	2808		GOTO 0x8
2	0001	3400		RETLW 0x0
3	0002	3400		RETLW 0x0
4	0003	3FFF		ADDLW 0xFF
5	0004	3FFF		ADDLW 0xFF
6	0005	3FFF		ADDLW 0xFF
7	0006	3FFF		ADDLW 0xFF
8	0007	3FFF		ADDLW 0xFF
9	0008	0185		CLRF PORTA
10	0009	301F		MOVLW 0x1F
11	000A	1683		BSF STATUS, 0x5
12	000B	0085		MOVWF PORTA
13	000C	1283		BCF STATUS, 0x5
14	000D	3080		MOVLW 0x80
15	000E	0085		MOVWF PORTA
16	000F	0000		NOP
17	0010	280F		GOTO 0xF
18	0011	3FFF		ADDLW 0xFF
19	0012	3FFF		ADDLW 0xFF
20	0013	3FFF		ADDLW 0xFF



Erstes Programm (Verbesserung der Lesbarkeit)

```

8 ;***** Configurationsbits Einstellung
9
10     __config      _CONFIG1,0x23E4
11     __config      _CONFIG2,0x3E4F
12 ;*****
13
14 ;*****
15 ;*****
16 ;*****
17 ;## 7 Segmentanzeige
18 #define          A_SEG  PORTB,0
19 #define          B_SEG  PORTB,1
20 #define          C_SEG  PORTB,2
21 #define          D_SEG  PORTB,3
22 #define          E_SEG  PORTA,4
23 #define          F_SEG  PORTA,5
24 #define          G_SEG  PORTA,6
25 #define          DP_SEG PORTA,7
26
27 #define          SEG0   PORTB,4
28 #define          SEG1   PORTC,3
29 #define          SEG2   PORTC,4
30 #define          SEG3   PORTC,5
31
32 ;## Leds
33 #define          LED_GRUEN  PORTC,0
34 #define          LED_ROT    PORTC,2
35 #define          LED_GELB   PORTC,1
36
37 ;## Analogpins
38 #define          POT1   PORTA,0
39 #define          TEMP   PORTA,1
40 #define          LDR    PORTA,2
41
42 ;## Taster
43 #define          S1     PORTB,5
44 #define          S2     PORTA,3
45
46 ;## RS232
47 #define          TX     PORTC,6
48 #define          RX     PORTC,7
49 ;*****

```

#define (zur leichteren Lesbarkeit des Codes)
HW-Eigendefinition

Erstes Programm (Verbesserung der Lesbarkeit)

- Macros zur Programmvereinfachung

Befehl:

bsf STATUS,RP0

(setzt FLAG RP0 im STATUS Register)

bcf STATUS,RP0

(löscht FLAG RP0 im STATUS)

```

50 ;Makros
51 Bank0 MACRO ;macro to select data RAM bank 0
52     bcf STATUS,RP0
53     bcf STATUS,RP1
54     ENDM
55
56 Bank1 MACRO ;macro to select data RAM bank 1
57     bcf STATUS,RP0
58     bcf STATUS,RP1
59     ENDM
60 Bank2 MACRO ;macro to select data RAM bank 2
61     bcf STATUS,RP0
62     bcf STATUS,RP1
63     ENDM
64
65 Bank3 MACRO ;macro to select data RAM bank 3
66     bcf STATUS,RP0
67     bcf STATUS,RP1
68     ENDM
69
    
```

7

Erstes Programm (Verbesserung der Lesbarkeit)

- **call** init_pic

Aufruf von Subroutinen

```

87 ;*****
88 ; Beginn des Hauptprogrammes
89 ;*****
90
91 ;*****
92 ; MAINROUTINE:  main
93 ;*****
94 main    call    init_pic    ;Aufruf Subroutinen
95
96        bcf    LED_ROT    ;led aktivieren
97
98        endlos
99        goto   endlos
100
    
```

- Funktionale Subroutinen erstellen
z.B. init_pic
zur Initialisierung des µ-Controllers

```

107 ;*****
108 ; SUBROUTINE:  initial PIC
109 ;*****
110
111 init_pic Bank0 ;select Bank 0
112         clrfs PORTA ;initialice porta, portb and portc
113         clrfs PORTB
114         clrfs PORTC
115
116 ;register configure BANK 1
117 Bank1   ;select Bank 1
118         ;Internen Oszillator einstellen 4MHz
119         movlw b'01100001'
120         movwf OSCCON
121
122         ;option_reg Control-Register???
123         movlw b'10000111' ;pullup disable,
124         movwf OPTION_REG ;option_reg register (tm0 1:256 etc.)
125
126         ;Port Control-Register???
127         movlw b'00001111' ;OUT/IN definieren
128         movwf TRISA
129         movlw b'11100000' ;OUT/IN definieren
130         movwf TRISB
131         movlw b'10000000' ;OUT/IN definieren
132         movwf TRISC
133
134         ;Interrupt Control-Register???
135         movlw b'00000000' ;all Interrupts disable
136         movwf INTCON
137
138 ;register configure BANK 3
139 Bank3   ;select Bank 3
140         movlw b'00000111'
141         movwf ANSEL ;AN0,1,2 = Analog
142         clrfs ANSELH ;Rest digital
143
144 ;register configure BANK 0
145 Bank0   ;select Bank 0
146         clrfs ADCON0 ;AD Converter = off
147
148         retlw 00h
149 ;*****
    
```

8

Übung

🔗 Übungsaufgabe (Abgabe nicht erforderlich)

- 🔗 a) Schreiben sie ein Programm das die Zahl x an der Stelle 0 der 7-Segmentanzeige zum Leuchten bringt (x letzte Ziffer ihrer Matrikelnummer)
- 🔗 b) Ergänzen sie das Programm so, dass die Zahl an der Stelle 3 ausgegeben wird, wenn die Taste S1 gedrückt ist.
- 🔗 Hinweis:
Benutzen sie folgende Befehle
movlw, movwf, bsf, bcf (für die Ausgabe)
für die Tastenabfrage stehen folgende Befehle zur Verfügung
btfs...bit test file skip if set (skip=nächster Befehl wird übersprungen)
btfsc...bit test file skip if clear

9

Einführung MPLAB X

SW-Debugging

10

Übung: Erstellen einer Wartezeit (ohne Interrupt)

```
*****
;
;   SUBROUTINE:   Wartezeit mit Timer0
;
;   Timer0 = 8 bit Timer bei 4MHz --> 1Takt =1us --> 1:4 --> 1Zählpuls=4us
;   --> 250*4us = 1ms
*****
wait_lms   movlw   .9           ;dezimal 256-250=6  (=9 Korrektur Befehlszeiten)
           movwf  TMR0
           bcf    INTCON,TOIF   ;TOIF löschen
           btfsz  INTCON,TOIF   ;Timer0 Interrupt?
           goto   $-1
           nop
           return

wait_xms   movwf   counter      ;counter=w
           call   wait_lms      ;Aufruf der lms Wartezeit
           decfsz counter,f     ;counter=counter-1 falls 0 Sprung
           goto   $-2
           return

wait_ls    movlw   .250         ;tatsächlich gemessen 1,003s
           call   wait_xms
           movlw  .250
           call   wait_xms
           movlw  .250
           call   wait_xms
           movlw  .250
           call   wait_xms
           return
```

z.B. Wartezeit mit Timer0

Timer wird so konfiguriert, dass ein Überlauf nach definierten Zeit passiert

Überprüfung mit Software-Debugger

13

Übung: Erstellen einer Wartezeit (ohne Interrupt)

```
*****
;
;   SUBROUTINE:   Wartezeiten
;   @ 4MHz
;   1Takt=lusec   exakte wartezeit=1028us
*****
wait_lms   clrf    wait_tmp1
           nop
           decfsz wait_tmp1,f
           goto   $-2
           return

wait_xms   movwf   wait_tmp2
           call   wait_lms
           decfsz wait_tmp2,f
           goto   $-2
           return

end
```

z.B. Wartezeit ohne Timer

Es werden die Befehlslaufzeiten genutzt

14

Übung1 (Blinkende Ziffer)

• Übungsaufgabe

- Verändern sie die UE1 so das bei Tastendruck die Ziffer blinkt
Blinkfrequenz abhängig von der Matrikelnummer
(Bsp. 2020274005 → $1/(5+1) \approx 0.167$ Hz)