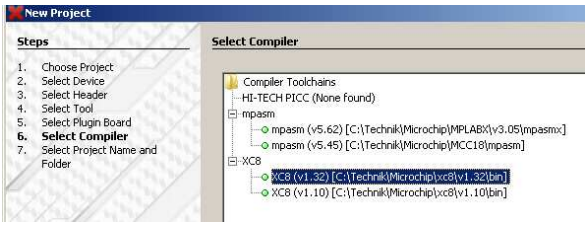
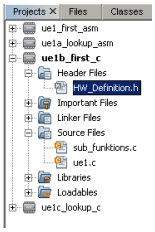
Einführung MPLAB-X

Erstes C-Programm

Erstes Programm

- **Verwendete HW**
  - Demoboard mit PICKit3
  - Prozessor 16F886
- **I Neues Projekt mit Wizard erstellen**
  - MPLAB-X öffnen
  - File-> New Project -> Menü folgen
    - Device PIC16F886
    - Microchip XC8 Toolsuite
    - Projekt Pfad (keine Sonderzeichen und Leerzeichen)
    - Dateien inkludieren (falls vorhanden)
  - Projektstruktur





Erstes Programm (C File erstellen)

II. C File erstellen

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

```
X:\Referenten\Hlavacek\VO_UE\u-Controller\SW\_SW_WS13\UE03_UE1_FirstProgr_C\ue1b_
1   /*****************************************
2   * Programm
3   * written by G.Hlavacek
4   *
5   * Datum: 05.08.2014
6   *
7   * CPU runs with 4MHz intern
8   *****************************************/
9
10  #include <xc.h>
11  //#include <pic16f866.h> nicht notwendig ??
12
13  // CONFIG1
14  //#pragma config    CONFIG1 = 0x03f4
15  #pragma config FOSC = INTRC_NOCLKOUT
16  #pragma config WDTE = OFF
17  #pragma config PWRTE = OFF
18  #pragma config MCLRE = ON
19  #pragma config CP = OFF
20  #pragma config CPD = OFF
21  #pragma config BOREN = ON
22  #pragma config IESO = OFF
23  #pragma config FCMEN = OFF
24  #pragma config LVP = OFF
25
26  // CONFIG2
27  #pragma config BOR4V = BOR21V
28  #pragma config WRT = OFF
29
30  #include "HW_Definition.h"      //Hardware Definitionen
31
32  extern void init_pic(void);
33
34
35  //*********************************************************************
36  //## Main                                                            ##
37  //*********************************************************************
38  int main(void) //@ 0x010
39  {
40      init_pic();
41
42      PORTB = 0x0f | PORTB; //alle Segment off (negative Logik)
43
```

- /* */ … Kommentare
- #include <xc.h> Definition aller Prozessorregister (Namen und Adresszuweisung)
- Prozessor-Konfiguration
- Headerdatei (eigene Definition der vorhandenen Hardware, im selben Verzeichnis)
- Deklaration der Funktionen (liegt in diesen Fall extern, d.h. in einem anderen C-File)
- Main Funktion
- Aufruf der Funktion init_pic();
- Modifikation von PORTB

Erstes Programm (C File erstellen)

II. Files erstellen

sub_funktions.c

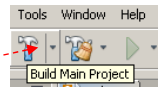
```
X:\Referenten\Hlavacek\VO_UE\BMMS_u-Controller\FW\FW_WS17_mplabx\ue1b_first_c.X\sub_funktions.c
Source  History
4   *
5   *
6   * CPU runs with 4MHz intern
7   *****************************************/
8   #include <xc.h>
9   #include "HW_Definition.h"
10
11  //*************************************************
12  //## Init Pic                                    ##
13  //*************************************************
14  void init_pic(void)
15  {
16      //NOP();                //Bank0             ;select Bank 0
17      PORTA=0x00;             //clrf    PORTA     ;initialice porta, portb and p
18      PORTB=0x00;             //clrf  PORTB
19      PORTC=0x00;             //clrf  PORTC
20
21                              //;register configure BANK 1
22                              //Bank1             ;select Bank 1
23                              //;Internen Oszilator einstellen 4MHz
24                              //movlw b'01100001'
25      OSCCON=0b01100001;      //movwf OSCCON
26                              //;option_reg  Control-Register???
27                              //movlw   b'10000111'     ;pullup disable,
28      OPTION_REG=0b10000110;  //movwf   OPTION_REG      ;option_reg register (tm
29                              //;Port  Control-Register???
30                              //movlw   b'00001111'     ;OUT/IN definieren
31      TRISA=0b00001111;       //movwf   TRISA
32                              //movlw   b'11100000'     ;OUT/IN definieren
33      TRISB=0b11100000;       //movwf   TRISB
34                              //movlw   b'10000000'     ;OUT/IN definieren
35      TRISC=0b10000000;       //movwf   TRISC
36                              //;Interrupt  Control-Register???
37                              //movlw   b'00000000'     ;all Interrupts disable
38      INTCON=0x00;            //movwf   INTCON
39                              //;register configure BANK 3
40                              //Bank3
41                              //movlw b'00000111'
42      ANSEL=0b00000111;       //movwf ANSEL           ;AN0,1,2 = Analog
43      ANSELH=0x00;            //clrf  ANSELH          ;Rest digital
44                              //;register configure BANK 0
45                              //Bank0                 ;select Bank 0
46      ADCON0=0;               //clrf  ADCON0          ;AD Converter = off
47                              //retlw 00h
48
49
```

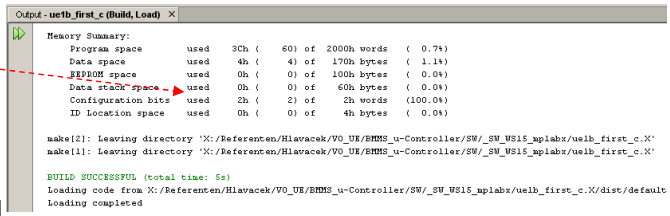
HW_Definition.h

```
X:\Referenten\Hlavacek\VO_UE\BMMS_u-Controller\FW\FW_WS17_mplabx\ue1b_first_c.X\HW_Definition.h
Source  History
1   /* ******************************************************
2       Definitionen der Hardware Demoboard V3.4
3       siehe Schaltplan
4   ******************************************************/
5
6   // #### Definitionen ###
7   #define on      1
8   #define off     0
9
10  /*****************************
11  ####  7 Segmentanzeige ####
12  *****************************/
13  #define     A_SEG   PORTBbits.RB0
14  #define     B_SEG   PORTBbits.RB1
15  #define     C_SEG   PORTBbits.RB2
16  #define     D_SEG   PORTBbits.RB3
17  #define     E_SEG   PORTAbits.RA4
18  #define     F_SEG   PORTAbits.RA5
19  #define     G_SEG   PORTAbits.RA6
20  #define     DP_SEG  PORTAbits.RA7
21
22  #define     SEG0    PORTBbits.RB4
23  #define     SEG1    PORTCbits.RC3
24  #define     SEG2    PORTCbits.RC4
25  #define     SEG3    PORTCbits.RC5
26
27  /*****************************
28  ####  LEDs ####
29  *****************************/
30  #define     LED_GRUEN   PORTCbits.RC0
31  #define     LED_ROT     PORTCbits.RC2
32  #define     LED_GELB    PORTCbits.RC1
33  #define     PWM_OUT     PORTCbits.RC2
34
35  /*****************************
36  ####  Analogpins ####
37  *****************************/
38  #define     POTI    PORTAbits.RA0
39  #define     TEMP    PORTAbits.RA1
40  #define     LDR     PORTAbits.RA2
41
42  /*****************************
43  ####  Taster ####
44  *****************************/
45  #define     S1      PORTBbits.RB5
46  #define     S2      PORTAbits.RA3
47
```

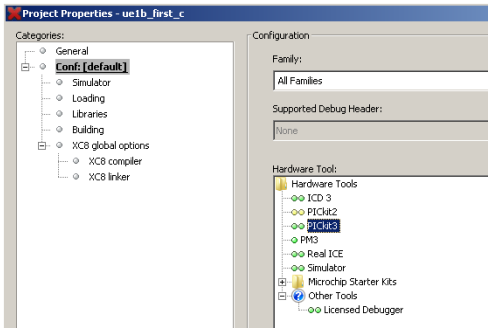
Erstes Programm (Programmierung)

- **III. HEX-Code erstellen**
  - Projekt -> Build
  - Kontrolle Output





- **IV. Chip Programmierung**
  - Select Programmer
  - Programm Device



Einführung MPLAB-X

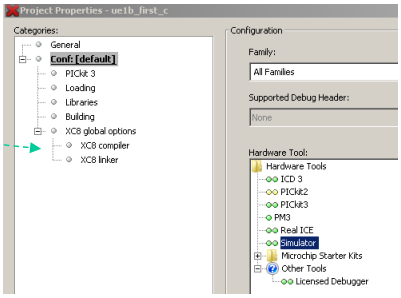
SW-Debugging

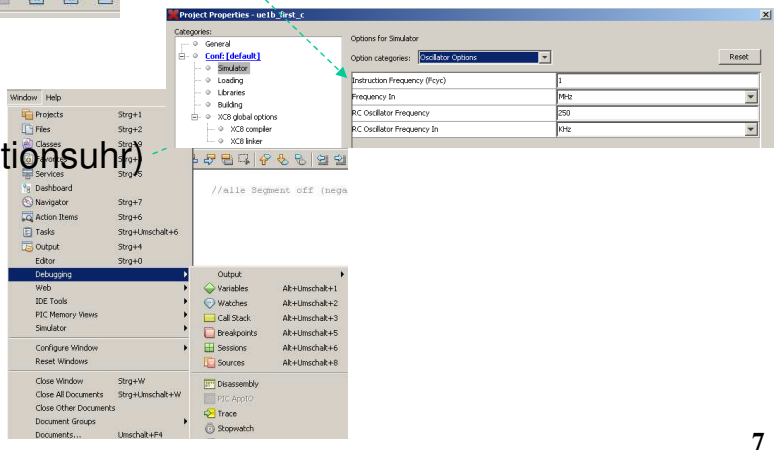
(C Programm)

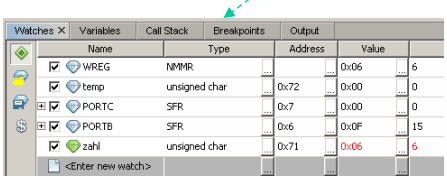
Übung: SW-Debugging, Debuggen von Übung 1

- **Fehlersuche (SW-Debugging)**
  - HardwareTools „Simulator“ wählen
  - Settings kontrollieren
    - richtige Prozessorfrequenz !
  - Navigation mit Debuggerleiste

  - Hilfsmittel
    - Breakpoints
    - StopWatch (Simulationsuhr)
    - Watchwindow



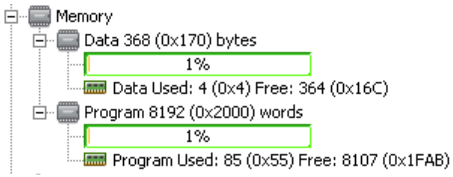




Speicherplatzvergleich ASM ←→ C

- ASM → 85 Programwords (14Bit)
- C → 205 Programwords (14Bit)
- Obwohl das ASM-Programm (UE1) und das C-Programm den gleichen Funktionsumfang haben ist deutlich erkennbar, dass das ASM Programm nicht mal die Hälfte an Speicherplatz benötigt.
- Bessere Compiler kommen zwar in die Nähe des ASM-Programmes, aber die Maschinennahe ASM Programmierung benötigt den geringsten Platzbedarf

ASM



C