INTEGRATING SDCC 8051 TOOLS INTO THE SILICON LABS IDE

1. Introduction

This application note describes how to integrate the SDCC 8051 Tools into the Silicon Laboratories IDE (Integrated Development Environment). It applies to Version 2.0 of the Silicon Laboratories IDE. Integrating SDCC 8051 Tools into the Silicon Laboratories IDE provides an efficient development environment with compose, edit, build, download, and debug operations integrated into the same program.

2. Key Points

- The Intel OMF-51 absolute object file generated by the SDCC 8051 tools enables source-level debug from the Silicon Laboratories IDE.
- Once SDCC Tools are integrated into the IDE, they are called by simply pressing the "Assemble/ Compile Current File" button or the "Build/Make Project" button.
- See the included software, AN198SW, for an example using the SDCC tools.

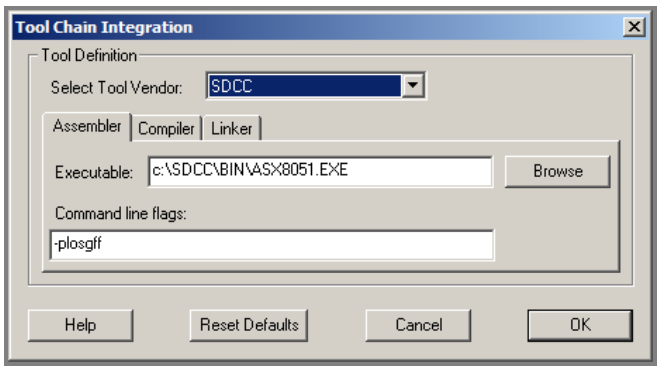
3. Create a Project in the Silicon Laboratories IDE

A project is necessary in order to link assembly files created by the compiler and build an absolute "OMF-51" output file. Follow these steps to create a project:

1. Under the "Project" menu, select "Add Files to Project...". Select the "C" source files that you want to add and click "Open". Continue adding files until all project files have been added.
2. To add files to the build process, right-click on the file name in the "Project Window" and select "Add *filename* to build".
3. Under the "Project" menu, select "Save Project As...". Enter a project workspace name and click "Save".

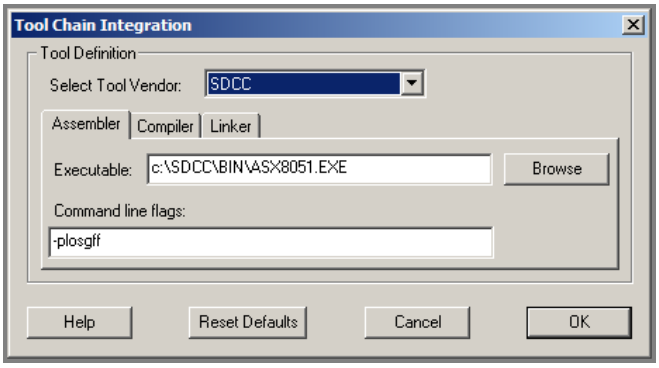
4. Configure the Tool Chain Integration Dialog

Under the "Project" menu, select "Tool Chain Integration". Select "SDCC" from the "Select Tool Vendor" drop-down list to bring up the dialog box shown below. Next, define the SDCC assembler, compiler, and linker as shown in the following sections.



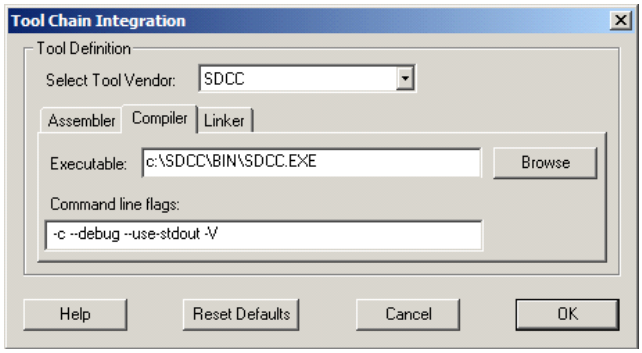
4.1. Assembler Definition

1. Under the "Assembler" tab, if the assembler executable is not already defined, click the browse button next to the "Executable:" text box, and locate the assembler executable. The default location for the SDCC assembler is “C:\SDCC\BIN\ASX8051.EXE”
2. Enter any additional command line flags directly in the "Command Line Flags" box.
3. See the following figure for the "Assembler" tab with the default SDCC settings.



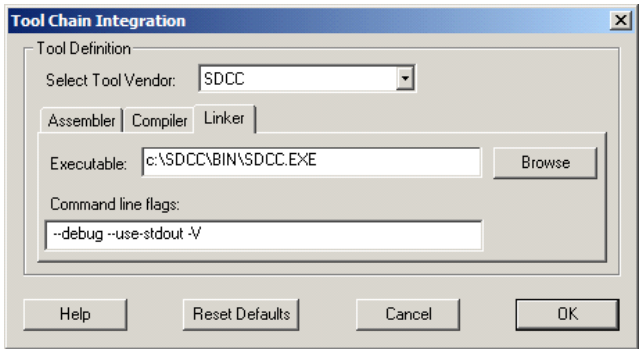
4.2. Compiler Definition

1. Under the "Compiler" tab, if the compiler executable is not already defined, click the browse button next to the "Executable:" text box, and locate the compiler executable. The default location for the SDCC compiler is “C:\SDCC\BIN\SDCC.EXE”.
2. Enter any additional command line flags directly in the "Command Line Flags" box.
3. See the following figure for the "Compiler" tab with the default SDCC settings.



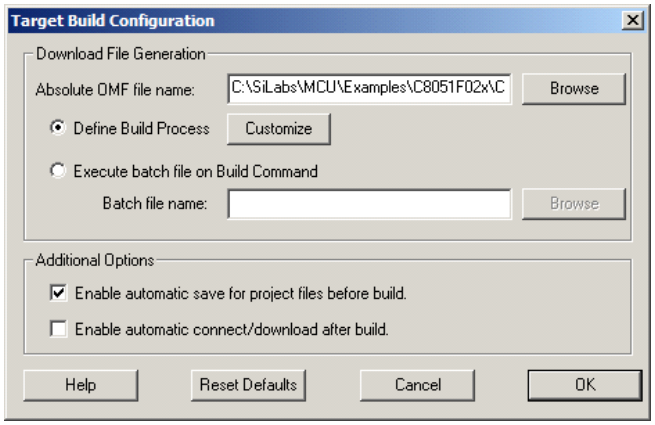
4.3. Linker Definition

1. Under the "Linker" tab, if the linker executable is not already defined, click the browse button next to the "Executable:" text box, and locate the linker executable. The default location for the SDCC linker is “C:\SDCC\BIN\SDCC.EXE”.
2. Enter any additional command line flags directly in the "Command line flags" box.
3. See the following figure for the "Linker" tab with the default SDCC settings.



5. Target Build Configuration

Under the "Project" menu select "Target Build Configuration" to bring up the following dialog box.

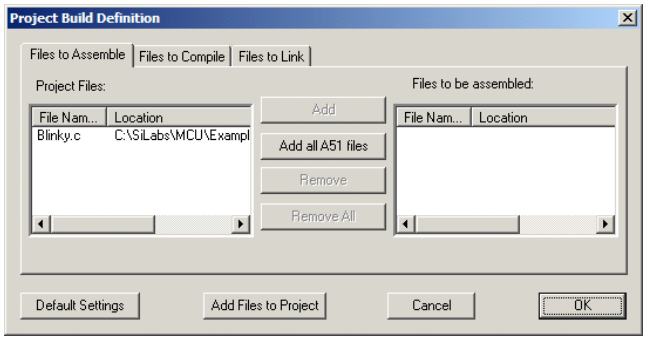


5.1. Output Filename

To customize a default filename or to create a new filename, click the browse button next to the "Absolute OMF file name:" edit box. Select a path and enter an output filename with no extension (ex. blinky).

5.2. Project Build Definition

Click the "Customize" button to bring up the "Project Build Definition" window shown below. This window allows selection of the files to be included in the build process. Although default assemble and compile selections will be made, ensure that all files have been correctly included in the build process. Under each tab, add files to assemble or compile by selecting the desired file and clicking the "Add" button. Files are removed in the same manner.



5.3. Additional Options

- If the "Enable automatic save for project files before build." box is checked, all files included in the project will be automatically saved when the "Build/Make project" button is pressed.
- If the "Enable automatic connect/download after build." box is checked, the project will be automatically downloaded to the target board when the "Build/Make project" button is pressed.

6. Building the Project

See the included software, AN198SW, for an example file (blinky.c) created for use with the SDCC compiler.

1. After saving all files that have been edited, the previous revisions will be saved in backup files. Backups are saved as the name of the file with the extension #1, #2, #3, and so on up to the number of backups (N) created and available. "#1" being the most recent and "N" being the least recent.
2. Click the "Assemble/Compile current file" button to compile only the current file.
3. Click the "Build/Make project" button to compile and link all the files in the project.
4. Review the errors and warnings generated during the build process located in the "Build" tab of the Output window (typically found at the bottom of the screen). Double-clicking on an error that is associated with a line number will automatically move the cursor to the proper line number in the source file that generated the error.

7. SDCC Considerations

This section outlines specific considerations that need to be taken into account when using the SDCC tools.

7.1. Compiler Considerations

- To enable the large memory model, add the command line parameter—large to the compiler and linker command line.

7.2. General Considerations

- SDCC does not have a default implementation of the putchar() function. This must be added to every project that uses putchar() or printf().
- SDCC does not have optimization levels, just directives to prevent certain optimizations. Some of these optimizations could cause confusion during debugging, or even incorrect behavior. If there are problems, see Section 3.2.7 of the SDCC manual for compiler directives to configure these optimizations.

7.3. SDCC Header Files

Header files for most Silicon Laboratories devices are available in the sdcc\include directory. If the header file for your device is not available, you can create the header file.

- The latest SDCC release includes a perl script, keil2sdcc.pl, that can be used to convert the Keil compatible header files to SDCC compatible header files. The perl script does not convert sbit declarations correctly. See below for the correct method of defining sfr16 and sbit variables.
- Convert sfr16 definitions to volatile unsigned data definitions as follows:

  ```
  sfr16 ADC0 = 0xBE;
  ```

  becomes:

  ```
  volatile unsigned data at 0xBE ADC0;
  ```

- Convert sbit definitions as follows:

  ```
  sbit LED = P1^6;
  ```

  becomes:

  ```
  sbit at 0x96 LED;
  ```

  The value of the sbit can be determined by looking at the Port I/O section of the MCU device datasheet. Each port has an SFR address. This determines the most-significant nibble of the sbit. The least-significant nibble is the bit position.

8. Source File Example

```
//-----------------------------------------------------------------------------
// Blinky.c
//-----------------------------------------------------------------------------
//
// This program flashes the green LED on the C8051F020 target board about five times
// a second using the interrupt handler for Timer3.
//
// Target: C8051F02x
//
// Date: 23 MAY 05
//
// Tool chain: SDCC 'c'
//
// Release 1.1
//
// Change Log:
//
// Revisions from 1.0 to 1.1:
// Changed LED = ~LED; to LED = !LED; because using the ~ operator causes SDCC
// to upcast to INT which leads to an incorrect result (always 1)

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------
#include <F020.h>                       // SFR declarations


//-----------------------------------------------------------------------------
// Global CONSTANTS
//-----------------------------------------------------------------------------

#define SYSCLK 2000000                  // approximate SYSCLK frequency in Hz

sbit at 0x96 LED;                       // green LED: '1' = ON; '0' = OFF

//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------
void PORT_Init (void);
void Timer3_Init (int counts);
void Timer3_ISR (void) interrupt 14;


//-----------------------------------------------------------------------------
// MAIN Routine
//-----------------------------------------------------------------------------
void main (void) {

   // disable watchdog timer
   WDTCN = 0xde;
   WDTCN = 0xad;

   PORT_Init ();
   Timer3_Init (SYSCLK / 12 / 10);      // Init Timer3 to generate interrupts

   EA = 1;                              // enable global interrupts
```

```
   while (1) {                          // spin forever
   }
}


//-----------------------------------------------------------------------------
// PORT_Init
//-----------------------------------------------------------------------------
//
// Configure the Crossbar and GPIO ports
//
void PORT_Init (void)
{
   XBR2    = 0x40;                  // Enable crossbar and weak pull-ups
   P1MDOUT |= 0x40;                 // enable P1.6 (LED) as push-pull output
}


//-----------------------------------------------------------------------------
// Timer3_Init
//-----------------------------------------------------------------------------
//
// Configure Timer3 to auto-reload and generate an interrupt at interval
// specified by <counts> using SYSCLK/12 as its time base.
//
void Timer3_Init (int counts)
{

   TMR3CN = 0x00;                   // Stop Timer3; Clear TF3;
                                    // use SYSCLK/12 as timebase
   TMR3RLH  = (-counts) >> 8;       // Init reload values
   TMR3RLL  = (-counts);            // Init reload values
   TMR3H    = 0xff;                 // set to reload immediately
   TMR3L    = 0xff;                 // set to reload immediately
   EIE2   |= 0x01;                  // enable Timer3 interrupts
   TMR3CN |= 0x04;                  // start Timer3
}


//-----------------------------------------------------------------------------
// Interrupt Service Routines
//-----------------------------------------------------------------------------

//-----------------------------------------------------------------------------
// Timer3_ISR
//-----------------------------------------------------------------------------
// This routine changes the state of the LED whenever Timer3 overflows.
//
void Timer3_ISR (void) interrupt 14
{
   TMR3CN &= ~(0x80);               // clear TF3
   LED = !LED;                      // change state of LED
}
```

9. Include File Example

```
//-----------------------------------------------------------------------------
// C8051F020.H
//-----------------------------------------------------------------------------
//
// Description: Register/bit definitions for the C8051F02x product family.
// Target: C8051F02x
// DATE: 7 JUN 04
// Tool chain: SDCC 'c'
// Revision: 1.0
//

/*  BYTE Registers  */
sfr at 0x80 P0      ;   /* PORT 0                                              */
sfr at 0x81 SP      ;   /* STACK POINTER                                       */
sfr at 0x82 DPL     ;   /* DATA POINTER - LOW BYTE                             */
sfr at 0x83 DPH     ;   /* DATA POINTER - HIGH BYTE                            */
sfr at 0x84 P4      ;   /* PORT 4                                              */
sfr at 0x85 P5      ;   /* PORT 5                                              */
sfr at 0x86 P6      ;   /* PORT 6                                              */
sfr at 0x87 PCON    ;   /* POWER CONTROL                                       */
sfr at 0x88 TCON    ;   /* TIMER CONTROL                                       */
sfr at 0x89 TMOD    ;   /* TIMER MODE                                          */
sfr at 0x8A TL0     ;   /* TIMER 0 - LOW BYTE                                  */
sfr at 0x8B TL1     ;   /* TIMER 1 - LOW BYTE                                  */
sfr at 0x8C TH0     ;   /* TIMER 0 - HIGH BYTE                                 */
sfr at 0x8D TH1     ;   /* TIMER 1 - HIGH BYTE                                 */
sfr at 0x8E CKCON   ;   /* CLOCK CONTROL                                       */
sfr at 0x8F PSCTL   ;   /* PROGRAM STORE R/W CONTROL                           */
sfr at 0x90 P1      ;   /* PORT 1                                              */
sfr at 0x91 TMR3CN  ;   /* TIMER 3 CONTROL                                     */
sfr at 0x92 TMR3RLL ;   /* TIMER 3 RELOAD REGISTER - LOW BYTE                  */
sfr at 0x93 TMR3RLH ;   /* TIMER 3 RELOAD REGISTER - HIGH BYTE                 */
sfr at 0x94 TMR3L   ;   /* TIMER 3 - LOW BYTE                                  */
sfr at 0x95 TMR3H   ;   /* TIMER 3 - HIGH BYTE                                 */
sfr at 0x96 P7      ;   /* PORT 7                                              */
sfr at 0x98 SCON0   ;   /* SERIAL PORT 0 CONTROL                               */
sfr at 0x99 SBUF0   ;   /* SERIAL PORT 0 BUFFER                                */
sfr at 0x9A SPI0CFG ;   /* SERIAL PERIPHERAL INTERFACE 0 CONFIGURATION         */
sfr at 0x9B SPI0DAT ;   /* SERIAL PERIPHERAL INTERFACE 0 DATA                  */
sfr at 0x9C ADC1    ;   /* ADC 1 DATA                                          */
sfr at 0x9D SPI0CKR ;   /* SERIAL PERIPHERAL INTERFACE 0 CLOCK RATE CONTROL    */
sfr at 0x9E CPT0CN  ;   /* COMPARATOR 0 CONTROL                                */
sfr at 0x9F CPT1CN  ;   /* COMPARATOR 1 CONTROL                                */
sfr at 0xA0 P2      ;   /* PORT 2                                              */
sfr at 0xA1 EMI0TC  ;   /* EMIF TIMING CONTROL                                 */
sfr at 0xA3 EMI0CF  ;   /* EXTERNAL MEMORY INTERFACE (EMIF) CONFIGURATION      */
sfr at 0xA4 P0MDOUT ;   /* PORT 0 OUTPUT MODE CONFIGURATION                    */
sfr at 0xA5 P1MDOUT ;   /* PORT 1 OUTPUT MODE CONFIGURATION                    */
sfr at 0xA6 P2MDOUT ;   /* PORT 2 OUTPUT MODE CONFIGURATION                    */
sfr at 0xA7 P3MDOUT ;   /* PORT 3 OUTPUT MODE CONFIGURATION                    */
sfr at 0xA8 IE      ;   /* INTERRUPT ENABLE                                    */
sfr at 0xA9 SADDR0  ;   /* SERIAL PORT 0 SLAVE ADDRESS                         */
sfr at 0xAA ADC1CN  ;   /* ADC 1 CONTROL                                       */
sfr at 0xAB ADC1CF  ;   /* ADC 1 ANALOG MUX CONFIGURATION                      */
sfr at 0xAC AMX1SL  ;   /* ADC 1 ANALOG MUX CHANNEL SELECT                     */
sfr at 0xAD P3IF    ;   /* PORT 3 EXTERNAL INTERRUPT FLAGS                     */
```

```
sfr at 0xAE SADEN1  ;   /* SERIAL PORT 1 SLAVE ADDRESS MASK                          */
sfr at 0xAF EMI0CN  ;   /* EXTERNAL MEMORY INTERFACE CONTROL                         */
sfr at 0xB0 P3      ;   /* PORT 3                                                    */
sfr at 0xB1 OSCXCN  ;   /* EXTERNAL OSCILLATOR CONTROL                               */
sfr at 0xB2 OSCICN  ;   /* INTERNAL OSCILLATOR CONTROL                               */
sfr at 0xB5 P74OUT  ;   /* PORTS 4 - 7 OUTPUT MODE                                   */
sfr at 0xB6 FLSCL   ;   /* FLASH MEMORY TIMING PRESCALER                             */
sfr at 0xB7 FLACL   ;   /* FLASH ACESS LIMIT                                         */
sfr at 0xB8 IP      ;   /* INTERRUPT PRIORITY                                        */
sfr at 0xB9 SADEN0  ;   /* SERIAL PORT 0 SLAVE ADDRESS MASK                          */
sfr at 0xBA AMX0CF  ;   /* ADC 0 MUX CONFIGURATION                                   */
sfr at 0xBB AMX0SL  ;   /* ADC 0 MUX CHANNEL SELECTION                               */
sfr at 0xBC ADC0CF  ;   /* ADC 0 CONFIGURATION                                       */
sfr at 0xBD P1MDIN  ;   /* PORT 1 INPUT MODE                                         */
sfr at 0xBE ADC0L   ;   /* ADC 0 DATA - LOW BYTE                                     */
sfr at 0xBF ADC0H   ;   /* ADC 0 DATA - HIGH BYTE                                    */
sfr at 0xC0 SMB0CN  ;   /* SMBUS 0 CONTROL                                           */
sfr at 0xC1 SMB0STA ;   /* SMBUS 0 STATUS                                            */
sfr at 0xC2 SMB0DAT ;   /* SMBUS 0 DATA                                              */
sfr at 0xC3 SMB0ADR ;   /* SMBUS 0 SLAVE ADDRESS                                     */
sfr at 0xC4 ADC0GTL ;   /* ADC 0 GREATER-THAN REGISTER - LOW BYTE                    */
sfr at 0xC5 ADC0GTH ;   /* ADC 0 GREATER-THAN REGISTER - HIGH BYTE                   */
sfr at 0xC6 ADC0LTL ;   /* ADC 0 LESS-THAN REGISTER - LOW BYTE                       */
sfr at 0xC7 ADC0LTH ;   /* ADC 0 LESS-THAN REGISTER - HIGH BYTE                      */
sfr at 0xC8 T2CON   ;   /* TIMER 2 CONTROL                                           */
sfr at 0xC9 T4CON   ;   /* TIMER 4 CONTROL                                           */
sfr at 0xCA RCAP2L  ;   /* TIMER 2 CAPTURE REGISTER - LOW BYTE                       */
sfr at 0xCB RCAP2H  ;   /* TIMER 2 CAPTURE REGISTER - HIGH BYTE                      */
sfr at 0xCC TL2     ;   /* TIMER 2 - LOW BYTE                                        */
sfr at 0xCD TH2     ;   /* TIMER 2 - HIGH BYTE                                       */
sfr at 0xCF SMB0CR  ;   /* SMBUS 0 CLOCK RATE                                        */
sfr at 0xD0 PSW     ;   /* PROGRAM STATUS WORD                                       */
sfr at 0xD1 REF0CN  ;   /* VOLTAGE REFERENCE 0 CONTROL                               */
sfr at 0xD2 DAC0L   ;   /* DAC 0 REGISTER - LOW BYTE                                 */
sfr at 0xD3 DAC0H   ;   /* DAC 0 REGISTER - HIGH BYTE                                */
sfr at 0xD4 DAC0CN  ;   /* DAC 0 CONTROL                                             */
sfr at 0xD5 DAC1L   ;   /* DAC 1 REGISTER - LOW BYTE                                 */
sfr at 0xD6 DAC1H   ;   /* DAC 1 REGISTER - HIGH BYTE                                */
sfr at 0xD7 DAC1CN  ;   /* DAC 1 CONTROL                                             */
sfr at 0xD8 PCA0CN  ;   /* PCA 0 COUNTER CONTROL                                     */
sfr at 0xD9 PCA0MD  ;   /* PCA 0 COUNTER MODE                                        */
sfr at 0xDA PCA0CPM0;   /* CONTROL REGISTER FOR PCA 0 MODULE 0                       */
sfr at 0xDB PCA0CPM1;   /* CONTROL REGISTER FOR PCA 0 MODULE 1                       */
sfr at 0xDC PCA0CPM2;   /* CONTROL REGISTER FOR PCA 0 MODULE 2                       */
sfr at 0xDD PCA0CPM3;   /* CONTROL REGISTER FOR PCA 0 MODULE 3                       */
sfr at 0xDE PCA0CPM4;   /* CONTROL REGISTER FOR PCA 0 MODULE 4                       */
sfr at 0xE0 ACC     ;   /* ACCUMULATOR                                               */
sfr at 0xE1 XBR0    ;   /* DIGITAL CROSSBAR CONFIGURATION REGISTER 0                 */
sfr at 0xE2 XBR1    ;   /* DIGITAL CROSSBAR CONFIGURATION REGISTER 1                 */
sfr at 0xE3 XBR2    ;   /* DIGITAL CROSSBAR CONFIGURATION REGISTER 2                 */
sfr at 0xE4 RCAP4L  ;   /* TIMER 4 CAPTURE REGISTER - LOW BYTE                       */
sfr at 0xE5 RCAP4H  ;   /* TIMER 4 CAPTURE REGISTER - HIGH BYTE                      */
sfr at 0xE6 EIE1    ;   /* EXTERNAL INTERRUPT ENABLE 1                               */
sfr at 0xE7 EIE2    ;   /* EXTERNAL INTERRUPT ENABLE 2                               */
sfr at 0xE8 ADC0CN  ;   /* ADC 0 CONTROL                                             */
sfr at 0xE9 PCA0L   ;   /* PCA 0 TIMER - LOW BYTE                                    */
sfr at 0xEA PCA0CPL0;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 0 - LOW BYTE    */
sfr at 0xEB PCA0CPL1;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 1 - LOW BYTE    */
sfr at 0xEC PCA0CPL2;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 2 - LOW BYTE    */
```

```
sfr at 0xED PCA0CPL3;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 3 - LOW BYTE  */
sfr at 0xEE PCA0CPL4;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 4 - LOW BYTE  */
sfr at 0xEF RSTSRC  ;   /* RESET SOURCE                                            */
sfr at 0xF0 B       ;   /* B REGISTER                                              */
sfr at 0xF1 SCON1   ;   /* SERIAL PORT 1 CONTROL                                   */
sfr at 0xF2 SBUF1   ;   /* SERAIL PORT 1 DATA                                      */
sfr at 0xF3 SADDR1  ;   /* SERAIL PORT 1                                           */
sfr at 0xF4 TL4     ;   /* TIMER 4 DATA - LOW BYTE                                 */
sfr at 0xF5 TH4     ;   /* TIMER 4 DATA - HIGH BYTE                                */
sfr at 0xF6 EIP1    ;   /* EXTERNAL INTERRUPT PRIORITY REGISTER 1                  */
sfr at 0xF7 EIP2    ;   /* EXTERNAL INTERRUPT PRIORITY REGISTER 2                  */
sfr at 0xF8 SPI0CN  ;   /* SERIAL PERIPHERAL INTERFACE 0 CONTROL                   */
sfr at 0xF9 PCA0H   ;   /* PCA 0 TIMER - HIGH BYTE                                 */
sfr at 0xFA PCA0CPH0;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 0 - HIGH BYTE */
sfr at 0xFB PCA0CPH1;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 1 - HIGH BYTE */
sfr at 0xFC PCA0CPH2;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 2 - HIGH BYTE */
sfr at 0xFD PCA0CPH3;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 3 - HIGH BYTE */
sfr at 0xFE PCA0CPH4;   /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 4 - HIGH BYTE */
sfr at 0xFF WDTCN   ;   /* WATCHDOG TIMER CONTROL                                  */


/*  BIT Registers  */

/*  TCON  0x88 */
sbit at 0x88 IT0;             /* EXT. INTERRUPT 0 TYPE      */
sbit at 0x89 IE0;             /* EXT. INTERRUPT 0 EDGE FLAG */
sbit at 0x8A IT1;             /* EXT. INTERRUPT 1 TYPE      */
sbit at 0x8B IE1;             /* EXT. INTERRUPT 1 EDGE FLAG */
sbit at 0x8C TR0;             /* TIMER 0 ON/OFF CONTROL     */
sbit at 0x8D TF0;             /* TIMER 0 OVERFLOW FLAG      */
sbit at 0x8E TR1;             /* TIMER 1 ON/OFF CONTROL     */
sbit at 0x8F TF1;             /* TIMER 1 OVERFLOW FLAG      */

/*  SCON0  0x98 */
sbit at 0x98 RI0 ;            /* RECEIVE INTERRUPT FLAG              */
sbit at 0x99 TI0 ;            /* TRANSMIT INTERRUPT FLAG             */
sbit at 0x9A RB80;            /* RECEIVE BIT 8                       */
sbit at 0x9B TB80;            /* TRANSMIT BIT 8                      */
sbit at 0x9C REN0;            /* RECEIVE ENABLE                      */
sbit at 0x9D SM20;            /* MULTIPROCESSOR COMMUNICATION ENABLE */
sbit at 0x9E SM10;            /* SERIAL MODE CONTROL BIT 1           */
sbit at 0x9F SM00;            /* SERIAL MODE CONTROL BIT 0           */

/*  IE  0xA8 */
sbit at 0xA8 EX0;             /* EXTERNAL INTERRUPT 0 ENABLE  */
sbit at 0xA9 ET0;             /* TIMER 0 INTERRUPT ENABLE     */
sbit at 0xAA EX1;             /* EXTERNAL INTERRUPT 1 ENABLE  */
sbit at 0xAB ET1;             /* TIMER 1 INTERRUPT ENABLE     */
sbit at 0xAC ES0;             /* UART0 INTERRUPT ENABLE       */
sbit at 0xAD ET2;             /* TIMER 2 INTERRUPT ENABLE     */
sbit at 0xAF EA ;             /* GLOBAL INTERRUPT ENABLE      */

/*  IP  0xB8 */
sbit at 0xB8 PX0;             /* EXTERNAL INTERRUPT 0 PRIORITY */
sbit at 0xB9 PT0;             /* TIMER 0 PRIORITY              */
sbit at 0xBA PX1;             /* EXTERNAL INTERRUPT 1 PRIORITY */
sbit at 0xBB PT1;             /* TIMER 1 PRIORITY              */
sbit at 0xBC PS0;             /* SERIAL PORT PRIORITY          */
sbit at 0xBD PT2;             /* TIMER 2 PRIORITY              */
```

```
/* SMB0CN 0xC0 */
sbit at 0xC0 SMBTOE;          /* SMBUS 0 TIMEOUT ENABLE          */
sbit at 0xC1 SMBFTE;          /* SMBUS 0 FREE TIMER ENABLE       */
sbit at 0xC2 AA    ;          /* SMBUS 0 ASSERT/ACKNOWLEDGE FLAG */
sbit at 0xC3 SI    ;          /* SMBUS 0 INTERRUPT PENDING FLAG  */
sbit at 0xC4 STO   ;          /* SMBUS 0 STOP FLAG               */
sbit at 0xC5 STA   ;          /* SMBUS 0 START FLAG              */
sbit at 0xC6 ENSMB ;          /* SMBUS 0 ENABLE                  */
sbit at 0xC7 BUSY  ;          /* SMBUS 0 BUSY                    */

/*  T2CON  0xC8 */
sbit at 0xC8 CPRL2 ;          /* CAPTURE OR RELOAD SELECT     */
sbit at 0xC9 CT2   ;          /* TIMER OR COUNTER SELECT      */
sbit at 0xCA TR2   ;          /* TIMER 2 ON/OFF CONTROL       */
sbit at 0xCB EXEN2 ;          /* TIMER 2 EXTERNAL ENABLE FLAG */
sbit at 0xCC TCLK0 ;          /* UART0 TX CLOCK SOURCE        */
sbit at 0xCD RCLK0 ;          /* UART0 RX CLOCK SOURCE        */
sbit at 0xCE EXF2  ;          /* EXTERNAL FLAG                */
sbit at 0xCF TF2   ;          /* TIMER 2 OVERFLOW FLAG        */

/*  PSW 0xD0 */
sbit at 0xD0 P     ;          /* ACCUMULATOR PARITY FLAG */
sbit at 0xD1 F1    ;          /* USER FLAG 1             */
sbit at 0xD2 OV    ;          /* OVERFLOW FLAG           */
sbit at 0xD3 RS0   ;          /* REGISTER BANK SELECT 0  */
sbit at 0xD4 RS1   ;          /* REGISTER BANK SELECT 1  */
sbit at 0xD5 F0    ;          /* USER FLAG 0             */
sbit at 0xD6 AC    ;          /* AUXILIARY CARRY FLAG    */
sbit at 0xD7 CY    ;          /* CARRY FLAG              */

/* PCA0CN D8H */
sbit at 0xD8 CCF0  ;          /* PCA 0 MODULE 0 INTERRUPT FLAG */
sbit at 0xD9 CCF1  ;          /* PCA 0 MODULE 1 INTERRUPT FLAG */
sbit at 0xDA CCF2  ;          /* PCA 0 MODULE 2 INTERRUPT FLAG */
sbit at 0xDB CCF3  ;          /* PCA 0 MODULE 3 INTERRUPT FLAG */
sbit at 0xDC CCF4  ;          /* PCA 0 MODULE 4 INTERRUPT FLAG */
sbit at 0xDE CR    ;          /* PCA 0 COUNTER RUN CONTROL BIT */
sbit at 0xDF CF    ;          /* PCA 0 COUNTER OVERFLOW FLAG   */

/* ADC0CN E8H */
sbit at 0xE8 AD0LJST;         /* ADC 0 RIGHT JUSTIFY DATA BIT              */
sbit at 0xE9 AD0WINT;         /* ADC 0 WINDOW COMPARE INTERRUPT FLAG       */
sbit at 0xEA AD0CM0 ;         /* ADC 0 START OF CONVERSION MODE BIT 0      */
sbit at 0xEB AD0CM1 ;         /* ADC 0 START OF CONVERSION MODE BIT 1      */
sbit at 0xEC AD0BUSY;         /* ADC 0 BUSY FLAG                           */
sbit at 0xED AD0INT ;         /* ADC 0 CONVERISION COMPLETE INTERRUPT FLAG */
sbit at 0xEE AD0TM  ;         /* ADC 0 TRACK MODE                          */
sbit at 0xEF AD0EN  ;         /* ADC 0 ENABLE                              */

/* SPI0CN F8H */
sbit at 0xF8 SPIEN  ;         /* SPI 0 SPI ENABLE          */
sbit at 0xF9 MSTEN  ;         /* SPI 0 MASTER ENABLE       */
sbit at 0xFA SLVSEL ;         /* SPI 0 SLAVE SELECT        */
sbit at 0xFB TXBSY  ;         /* SPI 0 TX BUSY FLAG        */
sbit at 0xFC RXOVRN ;         /* SPI 0 RX OVERRUN FLAG     */
sbit at 0xFD MODF   ;         /* SPI 0 MODE FAULT FLAG     */
sbit at 0xFE WCOL   ;         /* SPI 0 WRITE COLLISION FLAG */
sbit at 0xFF SPIF   ;         /* SPI 0 INTERRUPT FLAG      */
```

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- Updated Section 7 to reflect the release of SDCC 2.5.0.
- Updated the default assembler command line flags.
- Updated the example program.

NOTES:

CONTACT INFORMATION

Silicon Laboratories Inc.
4635 Boston Lane
Austin, TX 78735
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Email: MCUinfo@silabs.com
Internet: www.silabs.com

SILICON LABORATORIES