



Integrating Dunfield 8051 Tools into the Cygnal IDE

1. Introduction

This application note describes how to integrate the Dunfield 8051 Tools into the Cygnal IDE (Integrated Development Environment). It applies to Version 1.71 of the Cygnal IDE. Integrating Dunfield 8051 Tools into the Cygnal 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 Dunfield 8051 tools enables source-level debug from the Cygnal IDE.
- Once Dunfield 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, AN032SW, for an example using the Dunfield tools.

3. Create a Project in the Cygnal IDE

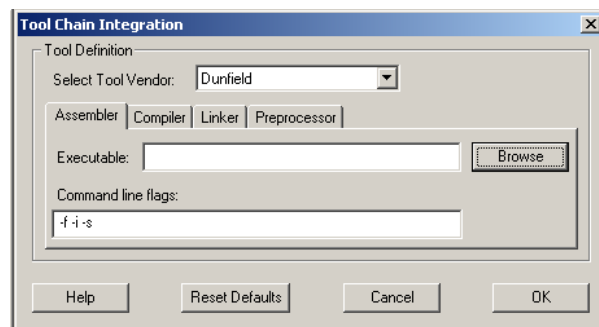
A project is necessary in order to link assembly files created by the compiler and build an absolute ‘OMF-51’ output file. Follow these steps to create a project:

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’ to bring up the dialog box shown below. First, select ‘Dunfield’ from the ‘Select Tool Vendor’ drop down list. Next, define the Dunfield assembler, compiler, and linker as shown in the following sections.

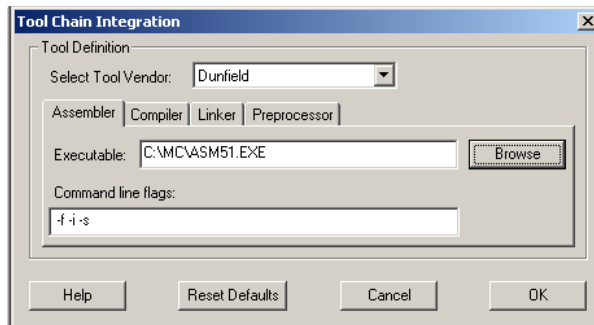


4.1. Assembler Definition

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 Dunfield assembler is “C:\MC\ASM51.exe”.
2. Enter any additional command line flags directly in the ‘Command Line Flags’ box.

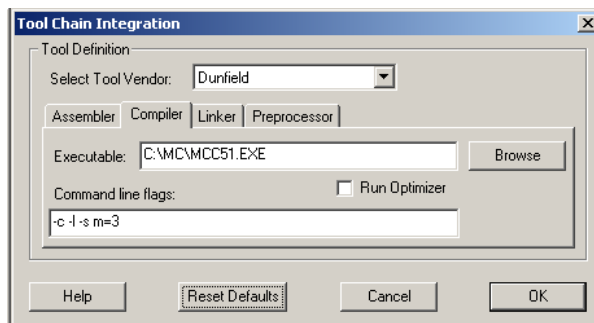


3. See the following figure for the ‘Assembler’ tab with the default Dunfield settings.



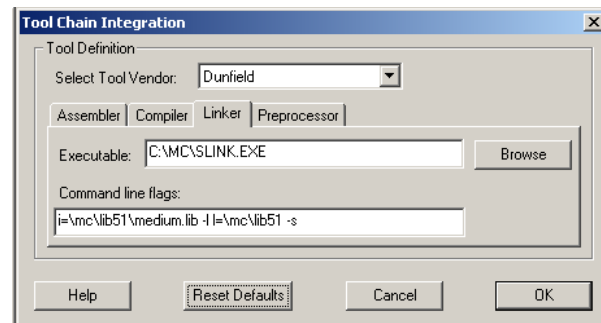
4.2. Compiler Definition

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 Dunfield compiler is “C:\MCMCC51.EXE”.
2. Enter any additional command line flags directly in the ‘Command Line Flags’ box.
3. Check the ‘Run Optimizer’ box if optimization is desired.
4. See the following figure for the ‘Compiler’ tab with the default Dunfield settings.



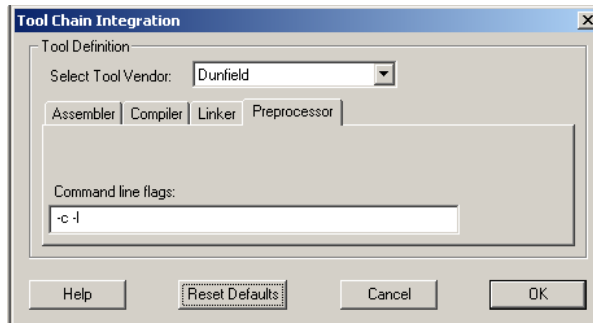
4.3. Linker Definition

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 Dunfield linker is “C:\MCVSLINK.EXE”.
2. Enter any additional command line flags directly in the ‘Command line flags’ box. If the command line becomes too long and some of the files are library files, add the additional libraries directly to ‘medium.lib’ or ‘small.lib’ according to the memory model used.
3. If the Dunfield tools weren’t installed in the default directory, change the default command line to reflect the new library path information.
4. See the following figure for the ‘Linker’ tab with the default Dunfield settings.



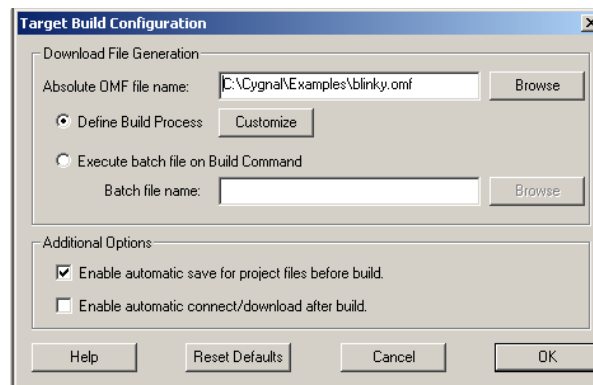
4.4. Preprocessor Definition

1. Enter any additional command line flags directly in the 'Command Line Flags' box.
2. See the following figure for the 'Preprocessor' tab with the default Dunfield settings.



5. Target Build Configuration

Under the 'Project' menu select 'Target Build Configuration' to bring up the dialog box shown below.

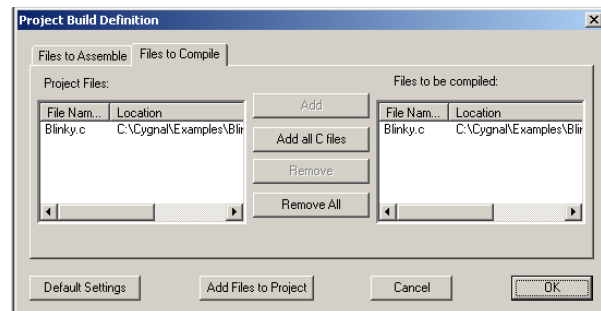


5.1. Output Filename

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 a '.omf' extension (ex. blinky.omf).

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

1. If the 'Enable automatic save for project files before build.' box is checked, then all files included in the project will be automatically saved when the 'Build/Make project' button is pressed.
2. If the 'Enable automatic connect/download after build.' box is checked, then 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, AN032SW, for several example files created for use with the Dunfield compiler, including blinky.c and include files (.h) for each of the Cygnal parts.

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. The following edits need to be made to the library file 8051RLPM.ASM before compiling. By default, this file is located at "C:\MC\LIB51\8051RLPM.ASM".

- a. This edit reserves enough contiguous space in program memory for the highest number of interrupts used. Change the value written to reflect the interrupt vector space needed for each project. See the device datasheet for the addresses corresponding to the interrupts being used. The following example reserves space for all interrupt vectors of a C8051F020.

Original Lines:

```
AJMP  +$0032  Skip interrupt vectors
DS    $0032-2 Reserve space for
        interrupt vectors
```

Edited Lines:

```
AJMP  *+$00B3 Skip interrupt vectors
DS    $00B3-2 Reserve space for ALL
        interrupt vectors
```

- b. This edit changes the CODE start address.

Original Line:

```
ORG  $0800  CODE Starts here
                (normally in ROM)
```

Edited Line:

```
ORG  $0000  CODE Starts here
                (normally in ROM)
```

3. Click the 'Assemble/Compile current file' button to compile just the current file.

4. Click the 'Build/Make project' button to compile and link all the files in the project.
5. 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). Carefully viewing the entire length of the build report is recommended, due to the fact that the Dunfield linker will continue to link even when there are compiler errors. 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. Dunfield Considerations

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

7.1. Compiler Considerations

- See the included software, AN032SW, for Cygnal include files (.h) created for use with the Dunfield compiler.
- Because the Dunfield tools do not allow bit operations in 'C', setbit(), clrbit(), and cplbit() macros are provided in the include files.
- The INTERRUPT(*vector_address*) macro is also provided in the include files. This macro allows you to use a 'C' function to handle an interrupt.
- When using the Dunfield tools, an assignment to a SFR generates 2 instructions:


```
ADC0 = 0xFE;    // SFR assignment in C
mov  A, 0FEh   ; Equivalent assembly
mov  ADC0, A   ; code generated
```
- The default settings for the Dunfield compiler use the medium memory model, and use medium memory model libraries.



AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE

- The Dunfield compiler does not create relocatable objects.
- The Dunfield compiler generates an assembly language object file.
- The following diagram illustrates the steps taken with the Dunfield tools to produce an OMF-51 file for downloading to the Cygnal MCU. These steps are automated when using the Cygnal IDE.

7.2. Assembler Considerations

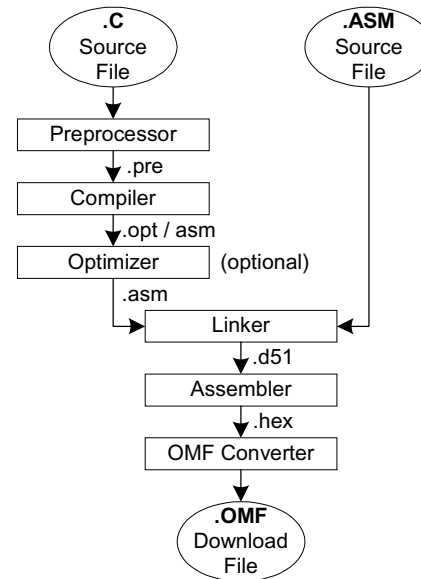
- Due to the fact that the Dunfield assembler can't resolve external references, running the assembler on a file that has external references will generate errors.
- The Dunfield assembler does not create relocatable objects.
- The Dunfield assembler generates an absolute object file (eg. Intel hex).

7.3. Linker Considerations

- The Dunfield linker only accepts assembly files.
- The Dunfield linker generates an absolute assembly language object file.

7.4. Other Considerations

- The Dunfield tools are DOS based executables. Because of this, there cannot be any command lines greater than 128 characters and all file names and folder names must conform to the 8.3 standard.
- The following extensions are created during the Dunfield compile/assemble process:
 - “.pre” - Output of the preprocessor.
 - “.opt” - Output of the compiler when “Run Optimizer” is selected.
 - “.asm” - Output of the compiler when “Run Optimizer” is not selected. Output of the optimizer when “Run Optimizer” is selected.
 - “.d51” - assembly language output by the Linker
 - “.hex” - output of the Assembler
 - “.omf” - absolute OMF file output by the OMF mapping utility





8. Source File Example

```
//-----  
// Blinky.c  
//-----  
// Copyright 2001 Cygnal Integrated Products, Inc.  
//  
// AUTH: BW  
// DATE: 25 SEP 02  
//  
// This program flashes the green LED on the C8051F020 target board about five times  
// a second using the interrupt handler for Timer3.  
// Target: C8051F02x  
//  
// Tool chain: Dunfield 'C' Compiler  
//  
  
//-----  
// Includes  
//-----  
#include "8051f020.h"                // SFR declarations  
  
  
//-----  
// Global CONSTANTS  
//-----  
#define SYSCLK 2000                  // approximate SYSCLK frequency in kHz  
  
  
//-----  
// Function PROTOTYPES  
//-----  
void PORT_Init (void);  
void Timer3_Init (unsigned int counts);  
void Timer3_ISR (void);  
  
  
//-----  
// MAIN Routine  
//-----  
void main (void)  
{  
    DISABLE_WDTCN;                   // disable watchdog timer  
  
    PORT_Init ();  
    Timer3_Init (SYSCLK / 12 * 1000 / 10); // Init Timer3 to generate interrupts  
                                         // at a ~10Hz rate.  
    setbit(EA);                       // enable global interrupts  
  
    while (1);                         // spin forever  
}
```



AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE

```
//-----  
// 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 (unsigned int counts)  
{  
    TMR3CN = 0x00;          // Stop Timer3; Clear TF3;  
                          // use SYSCLK/12 as timebase  
    TMR3RLL = (unsigned char)(-counts);    // Init reload values  
    TMR3RLH = (unsigned char)(-counts >> 8); // Init reload values  
  
    TMR3L  = 0xff;        // set to reload immediately  
    TMR3H  = 0xff;  
    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.  
//  
INTERRUPT(073h) void Timer3_ISR(void)  
{  
    TMR3CN &= ~(0x80);    // clear TF3  
    cplbit(P1.6);        // change state of LED  
}
```



9. Include File Example

```

/*-----
; Copyright (C) 2001 CYGNAL INTEGRATED PRODUCTS, INC.
; All rights reserved.
;
;
; FILE NAME : C8051F020.H
; TARGET MCUs: C8051F020, 'F021, 'F022, 'F023
; DESCRIPTION: Register/bit definitions for the C8051F02x product family
;               for use with the Dunfield Development Systems products.
;
; REVISION 1.0
;
;-----*/

/* BYTE Registers */
asm {
;P0      EQU  $80  /* PORT 0                      */
;SP      EQU  $81  /* STACK POINTER                */
;DPL     EQU  $82  /* DATA POINTER - LOW BYTE     */
;DPH     EQU  $83  /* DATA POINTER - HIGH BYTE    */
P4       EQU  $84  /* PORT 4                        */
P5       EQU  $85  /* PORT 5                        */
P6       EQU  $86  /* PORT 6                        */
;PCON    EQU  $87  /* POWER CONTROL                 */
;TCON    EQU  $88  /* TIMER CONTROL                 */
;TMOD    EQU  $89  /* TIMER MODE                    */
;TL0     EQU  $8A  /* TIMER 0 - LOW BYTE           */
;TL1     EQU  $8B  /* TIMER 1 - LOW BYTE           */
;TH0     EQU  $8C  /* TIMER 0 - HIGH BYTE          */
;TH1     EQU  $8D  /* TIMER 1 - HIGH BYTE          */
CKCON    EQU  $8E  /* CLOCK CONTROL                 */
PSCCTL   EQU  $8F  /* PROGRAM STORE R/W CONTROL    */
;P1      EQU  $90  /* PORT 1                        */
TMR3CN   EQU  $91  /* TIMER 3 CONTROL              */
TMR3RLL  EQU  $92  /* TIMER 3 RELOAD REGISTER - LOW BYTE
TMR3RLH  EQU  $93  /* TIMER 3 RELOAD REGISTER - HIGH BYTE
TMR3L    EQU  $94  /* TIMER 3 - LOW BYTE           */
TMR3H    EQU  $95  /* TIMER 3 - HIGH BYTE          */
P7       EQU  $96  /* PORT 7                        */
SCON0    EQU  $98  /* SERIAL PORT 0 CONTROL        */
SBUF0    EQU  $99  /* SERIAL PORT 0 BUFFER         */
SPIOCFG  EQU  $9A  /* SERIAL PERIPHERAL INTERFACE 0 CONFIGURATION
SPIODAT  EQU  $9B  /* SERIAL PERIPHERAL INTERFACE 0 DATA
ADC1     EQU  $9C  /* ADC 1 DATA                   */
SPIOCKR  EQU  $9D  /* SERIAL PERIPHERAL INTERFACE 0 CLOCK RATE CONTROL
CPT0CN   EQU  $9E  /* COMPARATOR 0 CONTROL         */
CPT1CN   EQU  $9F  /* COMPARATOR 1 CONTROL         */
;P2      EQU  $A0  /* PORT 2                        */
EMI0TC   EQU  $A1  /* EMIF TIMING CONTROL           */
EMI0CF   EQU  $A3  /* EXTERNAL MEMORY INTERFACE (EMIF) CONFIGURATION
P0MDOUT  EQU  $A4  /* PORT 0 OUTPUT MODE CONFIGURATION
P1MDOUT  EQU  $A5  /* PORT 1 OUTPUT MODE CONFIGURATION
P2MDOUT  EQU  $A6  /* PORT 2 OUTPUT MODE CONFIGURATION
P3MDOUT  EQU  $A7  /* PORT 3 OUTPUT MODE CONFIGURATION

```



AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE

```
;IE      EQU  $A8  /* INTERRUPT ENABLE */
SADDR0   EQU  $A9  /* SERIAL PORT 0 SLAVE ADDRESS */
ADC1CN   EQU  $AA  /* ADC 1 CONTROL */
ADC1CF   EQU  $AB  /* ADC 1 ANALOG MUX CONFIGURATION */
AMX1SL   EQU  $AC  /* ADC 1 ANALOG MUX CHANNEL SELECT */
P3IF     EQU  $AD  /* PORT 3 EXTERNAL INTERRUPT FLAGS */
SADEN1   EQU  $AE  /* SERIAL PORT 1 SLAVE ADDRESS MASK */
EMIOCN   EQU  $AF  /* EXTERNAL MEMORY INTERFACE CONTROL */
;P3      EQU  $B0  /* PORT 3 */
OSXCEN   EQU  $B1  /* EXTERNAL OSCILLATOR CONTROL */
OSCCEN   EQU  $B2  /* INTERNAL OSCILLATOR CONTROL */
P74OUT   EQU  $B5  /* PORTS 4 - 7 OUTPUT MODE */
FLSCL    EQU  $B6  /* FLASH MEMORY TIMING PRESCALER */
FLACL    EQU  $B7  /* FLASH ACCESS LIMIT */
;IP      EQU  $B8  /* INTERRUPT PRIORITY */
SADENO   EQU  $B9  /* SERIAL PORT 0 SLAVE ADDRESS MASK */
AMX0CF   EQU  $BA  /* ADC 0 MUX CONFIGURATION */
AMX0SL   EQU  $BB  /* ADC 0 MUX CHANNEL SELECTION */
ADC0CF   EQU  $BC  /* ADC 0 CONFIGURATION */
P1MDIN   EQU  $BD  /* PORT 1 INPUT MODE */
ADC0L    EQU  $BE  /* ADC 0 DATA - LOW BYTE */
ADC0H    EQU  $BF  /* ADC 0 DATA - HIGH BYTE */
SMB0CN   EQU  $C0  /* SMBUS 0 CONTROL */
SMB0STA  EQU  $C1  /* SMBUS 0 STATUS */
SMB0DAT  EQU  $C2  /* SMBUS 0 DATA */
SMB0ADR  EQU  $C3  /* SMBUS 0 SLAVE ADDRESS */
ADC0GTL  EQU  $C4  /* ADC 0 GREATER-THAN REGISTER - LOW BYTE */
ADC0GTH  EQU  $C5  /* ADC 0 GREATER-THAN REGISTER - HIGH BYTE */
ADC0LTL  EQU  $C6  /* ADC 0 LESS-THAN REGISTER - LOW BYTE */
ADC0LTH  EQU  $C7  /* ADC 0 LESS-THAN REGISTER - HIGH BYTE */
;T2CON   EQU  $C8  /* TIMER 2 CONTROL */
T4CON    EQU  $C9  /* TIMER 4 CONTROL */
;RCAP2L  EQU  $CA  /* TIMER 2 CAPTURE REGISTER - LOW BYTE */
;RCAP2H  EQU  $CB  /* TIMER 2 CAPTURE REGISTER - HIGH BYTE */
;TL2     EQU  $CC  /* TIMER 2 - LOW BYTE */
;TH2     EQU  $CD  /* TIMER 2 - HIGH BYTE */
SMB0CR   EQU  $CF  /* SMBUS 0 CLOCK RATE */
;PSW     EQU  $D0  /* PROGRAM STATUS WORD */
REF0CN   EQU  $D1  /* VOLTAGE REFERENCE 0 CONTROL */
DAC0L    EQU  $D2  /* DAC 0 REGISTER - LOW BYTE */
DAC0H    EQU  $D3  /* DAC 0 REGISTER - HIGH BYTE */
DAC0CN   EQU  $D4  /* DAC 0 CONTROL */
DAC1L    EQU  $D5  /* DAC 1 REGISTER - LOW BYTE */
DAC1H    EQU  $D6  /* DAC 1 REGISTER - HIGH BYTE */
DAC1CN   EQU  $D7  /* DAC 1 CONTROL */
PCA0CN   EQU  $D8  /* PCA 0 COUNTER CONTROL */
PCA0MD   EQU  $D9  /* PCA 0 COUNTER MODE */
PCA0CPM0 EQU  $DA  /* CONTROL REGISTER FOR PCA 0 MODULE 0 */
PCA0CPM1 EQU  $DB  /* CONTROL REGISTER FOR PCA 0 MODULE 1 */
PCA0CPM2 EQU  $DC  /* CONTROL REGISTER FOR PCA 0 MODULE 2 */
PCA0CPM3 EQU  $DD  /* CONTROL REGISTER FOR PCA 0 MODULE 3 */
PCA0CPM4 EQU  $DE  /* CONTROL REGISTER FOR PCA 0 MODULE 4 */
;ACC     EQU  $E0  /* ACCUMULATOR */
XBR0     EQU  $E1  /* DIGITAL CROSSBAR CONFIGURATION REGISTER 0 */
XBR1     EQU  $E2  /* DIGITAL CROSSBAR CONFIGURATION REGISTER 1 */
XBR2     EQU  $E3  /* DIGITAL CROSSBAR CONFIGURATION REGISTER 2 */
RCAP4L   EQU  $E4  /* TIMER 4 CAPTURE REGISTER - LOW BYTE */
```

AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE



```
RCAP4H    EQU    $E5    /* TIMER 4 CAPTURE REGISTER - HIGH BYTE    */
EIE1      EQU    $E6    /* EXTERNAL INTERRUPT ENABLE 1                */
EIE2      EQU    $E7    /* EXTERNAL INTERRUPT ENABLE 2                */
ADC0CN    EQU    $E8    /* ADC 0 CONTROL                               */
PCA0L     EQU    $E9    /* PCA 0 TIMER - LOW BYTE                     */
PCA0CPL0  EQU    $EA    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 0 - LOW BYTE */
PCA0CPL1  EQU    $EB    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 1 - LOW BYTE */
PCA0CPL2  EQU    $EC    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 2 - LOW BYTE */
PCA0CPL3  EQU    $ED    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 3 - LOW BYTE */
PCA0CPL4  EQU    $EE    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 4 - LOW BYTE */
RSTSRC    EQU    $EF    /* RESET SOURCE                               */
;B        EQU    $F0    /* B REGISTER                                 */
SCON1     EQU    $F1    /* SERIAL PORT 1 CONTROL                     */
SBUF1     EQU    $F2    /* SERIAL PORT 1 DATA                       */
SADDR1    EQU    $F3    /* SERIAL PORT 1                             */
TL4       EQU    $F4    /* TIMER 4 DATA - LOW BYTE                  */
TH4       EQU    $F5    /* TIMER 4 DATA - HIGH BYTE                 */
EIP1      EQU    $F6    /* EXTERNAL INTERRUPT PRIORITY REGISTER 1    */
EIP2      EQU    $F7    /* EXTERNAL INTERRUPT PRIORITY REGISTER 2    */
SPI0CN    EQU    $F8    /* SERIAL PERIPHERAL INTERFACE 0 CONTROL     */
PCA0H     EQU    $F9    /* PCA 0 TIMER - HIGH BYTE                   */
PCA0CPH0  EQU    $FA    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 0 - HIGH BYTE */
PCA0CPH1  EQU    $FB    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 1 - HIGH BYTE */
PCA0CPH2  EQU    $FC    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 2 - HIGH BYTE */
PCA0CPH3  EQU    $FD    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 3 - HIGH BYTE */
PCA0CPH4  EQU    $FE    /* CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 4 - HIGH BYTE */
WDTCN     EQU    $FF    /* WATCHDOG TIMER CONTROL                    */
}
```

```
extern register char\
```

```
    P0,
    SP,
    DPL,
    DPH,
    P4,
    P5,
    P6,
    PCON,
    TCON,
    TMOD,
    TL0,
    TL1,
    TH0,
    TH1,
    CKCON,
    PSCTL,
    P1,
    TMR3CN,
    TMR3RLl,
    TMR3RLH,
    TMR3L,
    TMR3H,
    P7,
    SCON0,
    SBUF0,
    SPI0CFG,
    SPI0DAT,
```



AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE

ADC1,
SPI0CKR,
CPT0CN,
CPT1CN,
P2,
EMI0TC,
EMI0CF,
P0MDOUT,
P1MDOUT,
P2MDOUT,
P3MDOUT,
IE,
SADDR0,
ADC1CN,
ADC1CF,
AMX1SL,
P3IF,
SADEN1,
EMI0CN,
P3,
OSCXCN,
OSICN,
P74OUT,
FLSCL,
FLACL,
IP,
SADEN0,
AMX0CF,
AMX0SL,
ADC0CF,
P1MDIN,
ADC0L,
ADC0H,
SMB0CN,
SMB0STA,
SMB0DAT,
SMB0ADR,
ADC0GTL,
ADC0GTH,
ADC0LTL,
ADC0LTH,
T2CON,
T4CON,
RCAP2L,
RCAP2H,
TL2,
TH2,
SMB0CR,
PSW,
REF0CN,
DAC0L,
DAC0H,
DAC0CN,
DAC1L,
DAC1H,
DAC1CN,
PCA0CN,



```
PCA0MD,
PCA0CPM0,
PCA0CPM1,
PCA0CPM2,
PCA0CPM3,
PCA0CPM4,
ACC,
XBR0,
XBR1,
XBR2,
RCAP4L,
RCAP4H,
EIE1,
EIE2,
ADC0CN,
PCA0L,
PCA0CPL0,
PCA0CPL1,
PCA0CPL2,
PCA0CPL3,
PCA0CPL4,
RSTSRC,
B,
SCON1,
SBUF1,
SADDR1,
TL4,
TH4,
EIP1,
EIP2,
SPI0CN,
PCA0H,
PCA0CPH0,
PCA0CPH1,
PCA0CPH2,
PCA0CPH3,
PCA0CPH4,
WDTCN;

/* BIT Addressable Registers */

/* TCON 0x88 */
#define TF1 TCON.7 /* TIMER 1 OVERFLOW FLAG */
#define TR1 TCON.6 /* TIMER 1 ON/OFF CONTROL */
#define TF0 TCON.5 /* TIMER 0 OVERFLOW FLAG */
#define TR0 TCON.4 /* TIMER 0 ON/OFF CONTROL */
#define IE1 TCON.3 /* EXT. INTERRUPT 1 EDGE FLAG */
#define IT1 TCON.2 /* EXT. INTERRUPT 1 TYPE */
#define IE0 TCON.1 /* EXT. INTERRUPT 0 EDGE FLAG */
#define IT0 TCON.0 /* EXT. INTERRUPT 0 TYPE */

/* SCON0 0x98 */
#define SM00 SCON0.7 /* SERIAL MODE CONTROL BIT 0 */
#define SM10 SCON0.6 /* SERIAL MODE CONTROL BIT 1 */
#define SM20 SCON0.5 /* MULTIPROCESSOR COMMUNICATION ENABLE */
#define RENO SCON0.4 /* RECEIVE ENABLE */
```



AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE

```
#define TB80    SCON0.3          /* TRANSMIT BIT 8          */
#define RB80    SCON0.2          /* RECEIVE BIT 8           */
#define TI0     SCON0.1          /* TRANSMIT INTERRUPT FLAG */
#define RI0     SCON0.0          /* RECEIVE INTERRUPT FLAG  */

/* IE 0xA8 */
#define EA      IE.7             /* GLOBAL INTERRUPT ENABLE */
#define ET2     IE.5             /* TIMER 2 INTERRUPT ENABLE */
#define ES0     IE.4             /* UART0 INTERRUPT ENABLE  */
#define ET1     IE.3             /* TIMER 1 INTERRUPT ENABLE */
#define EX1     IE.2             /* EXTERNAL INTERRUPT 1 ENABLE */
#define ET0     IE.1             /* TIMER 0 INTERRUPT ENABLE */
#define EX0     IE.0             /* EXTERNAL INTERRUPT 0 ENABLE */

/* IP 0xB8 */
#define PT2     IP.5             /* TIMER 2 PRIORITY        */
#define PS      IP.4             /* SERIAL PORT PRIORITY    */
#define PT1     IP.3             /* TIMER 1 PRIORITY        */
#define PX1     IP.2             /* EXTERNAL INTERRUPT 1 PRIORITY */
#define PT0     IP.1             /* TIMER 0 PRIORITY        */
#define PX0     IP.0             /* EXTERNAL INTERRUPT 0 PRIORITY */

/* SMB0CN 0xC0 */
#define BUSY     SMB0CN.7        /* SMBUS 0 BUSY            */
#define ENSMB    SMB0CN.6        /* SMBUS 0 ENABLE          */
#define STA      SMB0CN.5        /* SMBUS 0 START FLAG      */
#define STO      SMB0CN.4        /* SMBUS 0 STOP FLAG       */
#define SI       SMB0CN.3        /* SMBUS 0 INTERRUPT PENDING FLAG */
#define AA       SMB0CN.2        /* SMBUS 0 ASSERT/ACKNOWLEDGE FLAG */
#define SMBFTE   SMB0CN.1        /* SMBUS 0 FREE TIMER ENABLE */
#define SMBTOE   SMB0CN.0        /* SMBUS 0 TIMEOUT ENABLE  */

/* T2CON 0xC8 */
#define TF2      T2CON.7         /* TIMER 2 OVERFLOW FLAG   */
#define EXF2     T2CON.6         /* EXTERNAL FLAG           */
#define RCLK     T2CON.5         /* RECEIVE CLOCK FLAG      */
#define TCLK     T2CON.4         /* TRANSMIT CLOCK FLAG     */
#define EXEN2    T2CON.3         /* TIMER 2 EXTERNAL ENABLE FLAG */
#define TR2      T2CON.2         /* TIMER 2 ON/OFF CONTROL  */
#define CT2      T2CON.1         /* TIMER OR COUNTER SELECT */
#define CPRL2    T2CON.0         /* CAPTURE OR RELOAD SELECT */

/* PSW */
#define CY       PSW.7           /* CARRY FLAG              */
#define AC       PSW.6           /* AUXILIARY CARRY FLAG    */
#define F0       PSW.5           /* USER FLAG 0            */
#define RS1     PSW.4           /* REGISTER BANK SELECT 1  */
#define RS0     PSW.3           /* REGISTER BANK SELECT 0  */
#define OV       PSW.2           /* OVERFLOW FLAG           */
#define F1       PSW.1           /* USER FLAG 1            */
#define P        PSW.0           /* ACCUMULATOR PARITY FLAG */

/* PCA0CN D8H */
#define CF       PCA0CN.7        /* PCA 0 COUNTER OVERFLOW FLAG */
#define CR       PCA0CN.6        /* PCA 0 COUNTER RUN CONTROL BIT */
#define CCF4     PCA0CN.4        /* PCA 0 MODULE 4 INTERRUPT FLAG */
#define CCF3     PCA0CN.3        /* PCA 0 MODULE 3 INTERRUPT FLAG */
```

AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE



```
#define CCF2 PCA0CN.2          /* PCA 0 MODULE 2 INTERRUPT FLAG */
#define CCF1 PCA0CN.1          /* PCA 0 MODULE 1 INTERRUPT FLAG */
#define CCF0 PCA0CN.0          /* PCA 0 MODULE 0 INTERRUPT FLAG */

/* ADC0CN E8H */
#define AD0EN ADC0CN.7         /* ADC 0 ENABLE */
#define AD0TM ADC0CN.6         /* ADC 0 TRACK MODE */
#define AD0INT ADC0CN.5        /* ADC 0 CONVERSION COMPLETE INTERRUPT FLAG */
#define AD0BUSY ADC0CN.4       /* ADC 0 BUSY FLAG */
#define AD0CM1 ADC0CN.3        /* ADC 0 START OF CONVERSION MODE BIT 1 */
#define AD0CM0 ADC0CN.2        /* ADC 0 START OF CONVERSION MODE BIT 0 */
#define AD0WINT ADC0CN.1       /* ADC 0 WINDOW COMPARE INTERRUPT FLAG */
#define AD0LJST ADC0CN.0       /* ADC 0 RIGHT JUSTIFY DATA BIT */

/* SPI0CN F8H */
#define SPIF SPI0CN.7          /* SPI 0 INTERRUPT FLAG */
#define WCOL SPI0CN.6          /* SPI 0 WRITE COLLISION FLAG */
#define MODF SPI0CN.5          /* SPI 0 MODE FAULT FLAG */
#define RXOVRN SPI0CN.4        /* SPI 0 RX OVERRUN FLAG */
#define TXBSY SPI0CN.3         /* SPI 0 TX BUSY FLAG */
#define SLVSEL SPI0CN.2        /* SPI 0 SLAVE SELECT */
#define MSTEN SPI0CN.1         /* SPI 0 MASTER ENABLE */
#define SPIEN SPI0CN.0         /* SPI 0 SPI ENABLE */

/* Cygnal MACROS */
#define DISABLE_WDTN asm{\
    mov 0ffh,#0deh\
    mov 0ffh,#0adh\
}

/* DUNFIELD MACROS */

/* The following Macros have been excerpted from Dunfield Development Systems
 * header files, 8051bit.h and 8051int.h, distributed with Dunfield Development
 * Systems toolset for 8051 microcontrollers. 8051bit.h, 8051int.h
 */

/*
 * Macros to allow direct access to the I/O bits of the 8051
 * internal registers from DDS MICRO-C/51.
 *
 * This file **REQUIRES** the extended pre-processor (MCP).
 * (CC51 ... -P, or "Preprocess" step enabled in DDSIDE)
 *
 * The 'setbit', 'clrbit' and 'cplbit' macros use arguemnts in the standard
 * 8051 assembly language syntax of: address.bit
 *
 * In the case of the 'tstbit' macro, the optimizer will attempt to
 * convert references to: P0-P3, SCON, PSW, IP, IE, TCON and T2CON ANded
 * with the single-bit mask generated by (1<<bit) into a single JB or JNB
 * instruction. To accomodate user defined "bit" variables, this optimization
 * will also be applied to any global "register char" variable which has a
 * leading '_' in it's name.
 *
 * Also note, that misapplying this optimization to a variable which is NOT
 * in the bit addressable ($20-$2F + $80-$FF even multiples of 8) address
 * ranges will result in an "Out of range" assembly error. For this reason,
```



AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE

```
* you should NOT use "register" variable names beginning with '_' unless
* the variables reside in one of these bit addressable locations.
*
* Copyright 1991-2002 Dave Dunfield
* All rights reserved.
*/

#define setbit(bit) asm { /* Macro to set a I/O bit */\
    SETB bit\
}

#define clrbit(bit) asm { /* Macro to clear an I/O bit */\
    CLR bit\
}

#define cplbit(bit) asm { /* Macro to complement an I/O bit */\
    CPL bit\
}

#define testbit(address,bit) (address&(1<<bit)) /* Test an I/O bit */
//03180 12/6/2002

/*
* This macro allows you to use a 'C' function to handle an 8051 interrupt.
*
* The only parameter is the DECIMAL address of the interrupt vector.
* Defines are provided for the common values.
*
* The macro MUST be placed IMMEDIATELY preceeding the definition of the
* function which is to be the interrupt handler. Example:
*
*     INTERRUPT(_SER_) serial_int_handler()
*     {
*         /* ... Function code to handle serial interrupt ... */
*     }
*
* BEFORE USING THIS MACRO... Make sure that the startup code leaves
* room for the interrupt vectors ($0003 - $0032), Use something like
* this at the beginning of the "prefix" file:
*
*             AJMP    *+$0032    Jump around vectors
*             DS      $0032-2    Reserve space for vectors
*
* Unfortunately, the code generated by the compiler and in the runtime
* library sometimes has to reference registers by address. This means
* that we cannot simply "switch" the register bank during the interrupt,
* but have to "push/pop" all of the registers instead.
*
***NOTE 1: If you are using the -Z (2K addressing) option, which translates
* LCALLs into ACALLs, you MUST change the offset in the LCALL instruction
* below to 51 (instead of 52 which is required when not using -Z).
*
***NOTE 2: If you are using a memory model with an EXTERNAL stack, and have
* interrupt handlers written in C that use the external stack (ie: have local
* variables), you must modify the runtime library functions which manipulate
* the external stack pointer to disable interrupts during those manipulations
* (look for references to ?stack in 8051RLP?.ASM, the main problem occurs in
```

AN032 - Integrating Dunfield 8051 Tools into the Cygnal IDE



```
* the ?adjex1/?adjex2 routine if the interrupt happens after the low byte of
* the stack pointer has been adjusted, but before the upper byte is updated).
*
* This file **REQUIRES** the extended pre-processor (MCP).
* (CC51 ... -P, or "Preprocess" step enabled in DDSIDE)
*
* Copyright 1991-2002 Dave Dunfield
* All rights reserved.
*/

/*
* The following symbol must be set to the address of the interrupt vector
* table. For ROM based system, this will be 0, which is the start of the
* ROM. When running under a debugger (such as MON51 or MONICA), this must
* be changed to the address to which the debugger re-vectors interrupts.
*/
#define INTBASE 0

#define INTERRUPT(vec) asm {\
$SE:0                                /* Insure output in code segment      */\
_/**/vec /* This label will be the address of the "prefix" stub */\
    ORG     INTBASE+vec /* Position to interrupt vector                */\
    PUSH   PSW          /* Save Processor Status Word          */\
    PUSH   A            /* Save accumulator (2 bytes)         */\
    LJMP   _/**/vec     /* Don't overwrite next vector (+3 = 7) */\
    ORG    _/**/vec     /* Position back to original PC       */\
    PUSH   B            /* Save B register (+2 = 4)           */\
    PUSH   DPH          /* Save Data Pointer (HIGH)           */\
    PUSH   DPL          /* Save Data Pointer (LOW)            */\
    PUSH   0            /* Save R0                             */\
    PUSH   1            /* Save R1                             */\
    PUSH   2            /* Save R2                             */\
    PUSH   3            /* Save R3                             */\
    PUSH   4            /* Save R4                             */\
    PUSH   5            /* Save R5                             */\
    PUSH   6            /* Save R6                             */\
    PUSH   7            /* Save R7                             */\
    LCALL  _/**/vec+52 /* Invoke 'C' function. *SEE NOTE ABOVE */\
    POP    7            /* Restore R7                          */\
    POP    6            /* Restore R6                          */\
    POP    5            /* Restore R5                          */\
    POP    4            /* Restore R4                          */\
    POP    3            /* Restore R3                          */\
    POP    2            /* Restore R2                          */\
    POP    1            /* Restore R1                          */\
    POP    0            /* Restore R0                          */\
    POP    DPL          /* Restore Data Pointer (LOW)          */\
    POP    DPH          /* Restore Data Pointer (HIGH)         */\
    POP    B            /* Restore B register                   */\
    POP    A            /* Restore A accumulator                */\
    POP    PSW          /* Restore Processor Status Word       */\
    RETI              /* End interrupt                       */\
}
#undef INTBASE
```