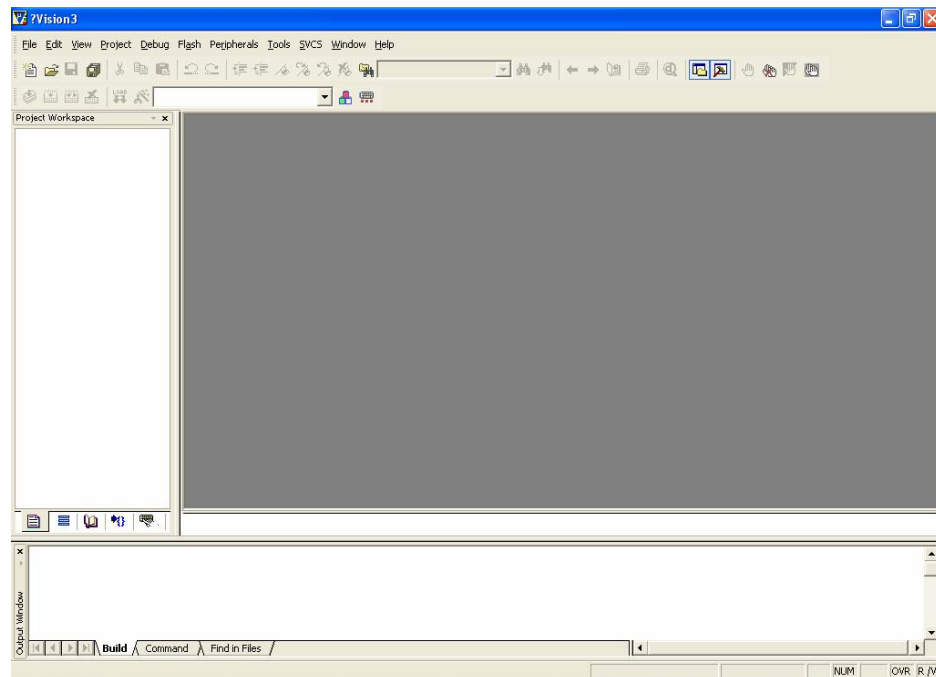


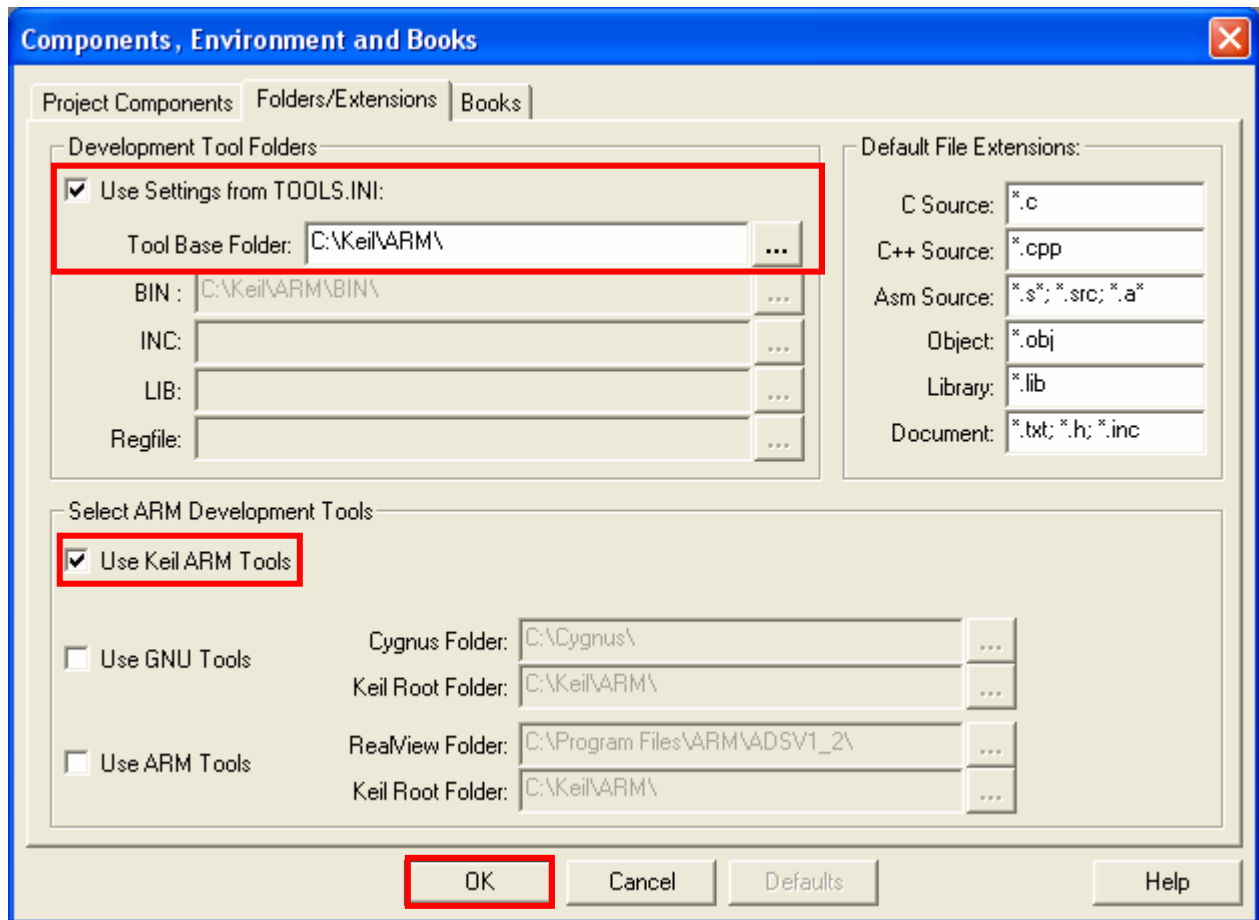
## An Example of using Keil uVision3 for creating Keil ARM's Project File

In this chapter, represent how to write C Language Program via Keil ARM for translating orders under Text Editor Program of Keil (Keil uVision3). We would explain the method of setting Option for connecting orders to translate Program with Keil ARM through Keil uVision3 only. Other detailed instruction and how to write Keil ARM, users can learn from user manual of Keil ARM. Method of setting default value of Keil uVision3 to use with Keil ARM is;

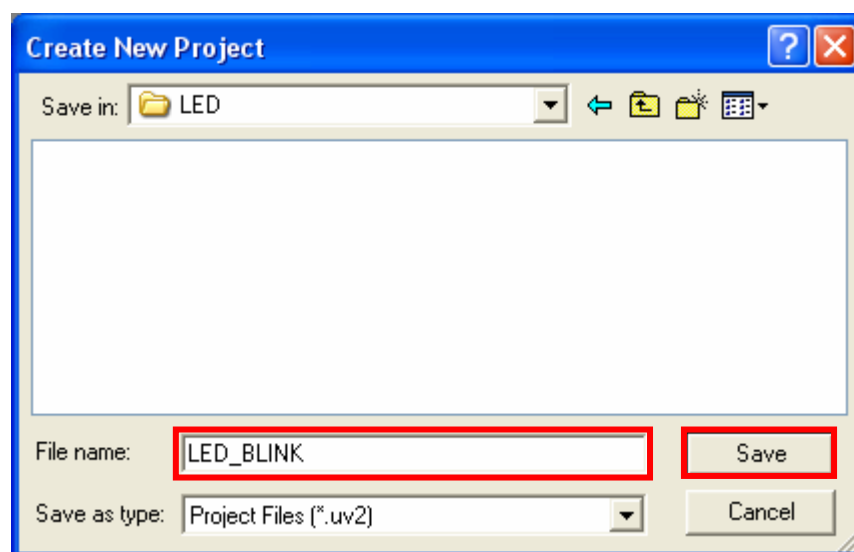
1. Open Keil uVision Program which is Text Editor of Keil-ARM uses for writing C Language Source Code Program as in the picture



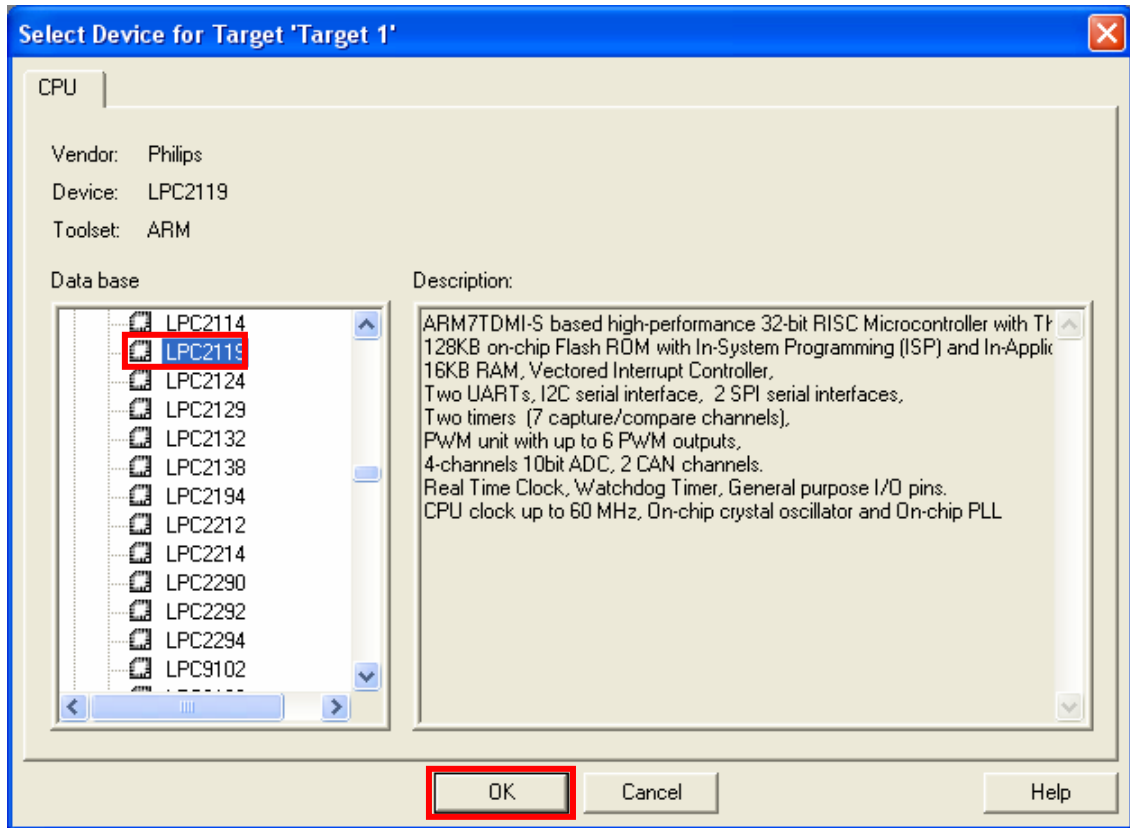
2. Set default value to translate uVision3 Code to use with Keil uVision3 Program and Keil ARM. Click **Project** → **Components, Environment, Books...** then select default value to use Compiler titled **Select ARM Development Tools**. There's 3 default values of them -**Use Keil ARM Tools**, **Use GNU Tools**, and **Use ARM Tools**- in this case, select "**Use Keil ARM Tools**" then assign position of folder to save default value of Keil ARM Program. Generally, it is "C:\Keil\ARM\", but if you install Keil in other position, you need to change correctly as in the picture



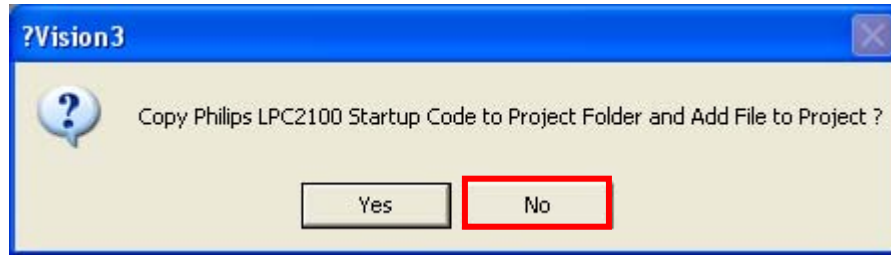
3. Create new Project File. Click **Project** → **New Project** then set the position of folder that you want to save with its title. For example, if you want to create Project File named LED\_BLINK and save in folder named LED, you can set them by yourself. After you fill its name in File name blank, click **Save** project File as in the picture



Program proceeding waits for users to set No. of MCU after you saved Project File. No.of Philips' MCU in ET-ARM STAMP LPC2119 Board is LPC2119 then click **OK** as in the picture



Next, users need to confirm to copy File Startup of Keil to use with Philips' MCU or not because Startup File is the beginning value of MCU such as Stack values and default value to use with Phase-Lock-Loop. You should set values of them before starting to run your written program, otherwise MCU is added orders to run by the written program. Startup File of Keil-ARM is Assembly Language and is set to work with Keil Development. So, there's some different specifications to use. You can't use Startup File with ET-ARM STAMP LPC2119 Board immediately, you need to change some default value. Using Keil ARM Program to translate orders, users need to correct Startup File to set value as well as ET-ARM STAMP LPC2119 Board form. In this case, select **"No"** to protect Keil uVision3 copy Startup File of Keil-ARM to use in this Project

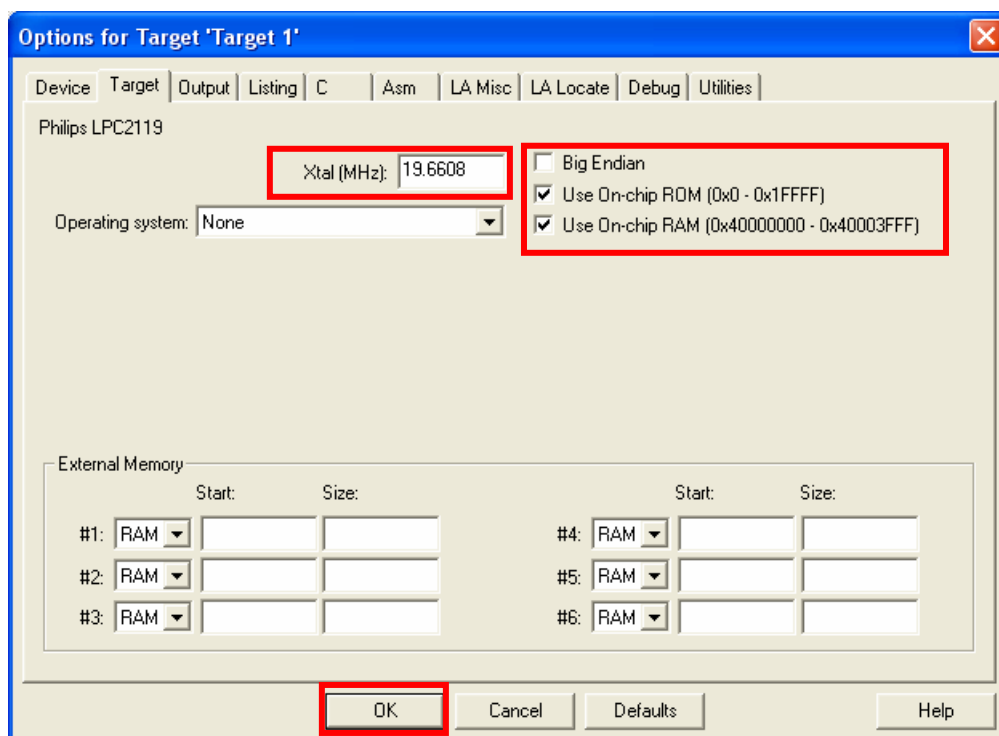


4. Copy File named "Startup.s" that is in ETT's CD-ROM in folder named "SOURCE\_KEIL" into the same position of New Project File

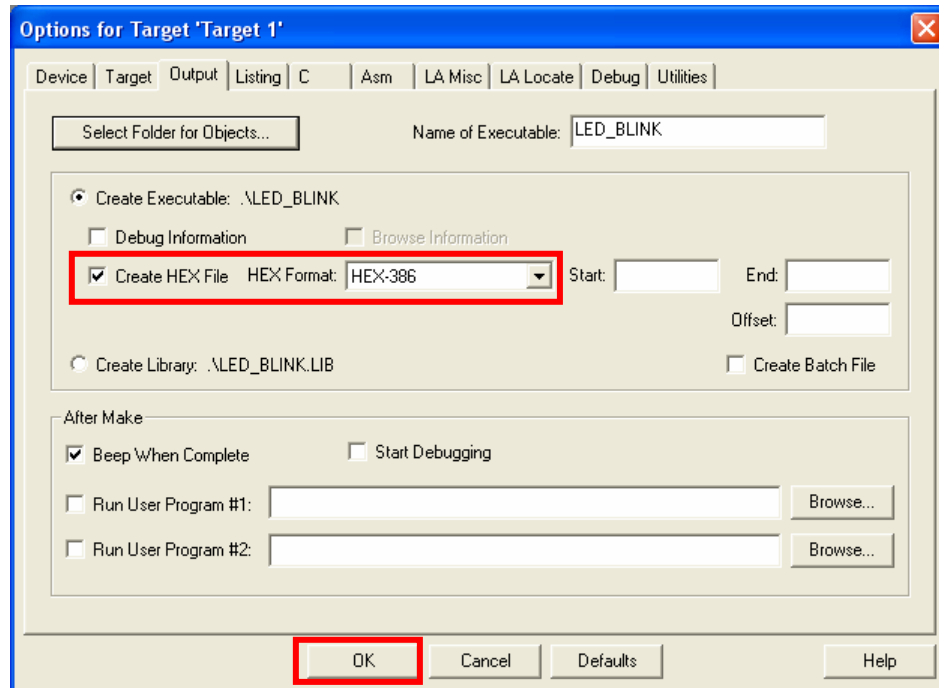
File named "Startup.S" which is contained orders of ARM7 in Assembly Language uses to set the necessary default value of MCU such as setting Stack value, Initial Phase-Lock-Loop, MAM Function, and MCU's Vector. If you add "Startup.s File" from Keil or copy file from other source to use with "ET-ARM STAMP LPC2119 Board", its program in Startup is different, and it will effect on written program

5. Select Option of Project File. Click **Project** → **Option for target 'target 1'** then select Tab of Target for setting MCU Target value. The proceeding is;

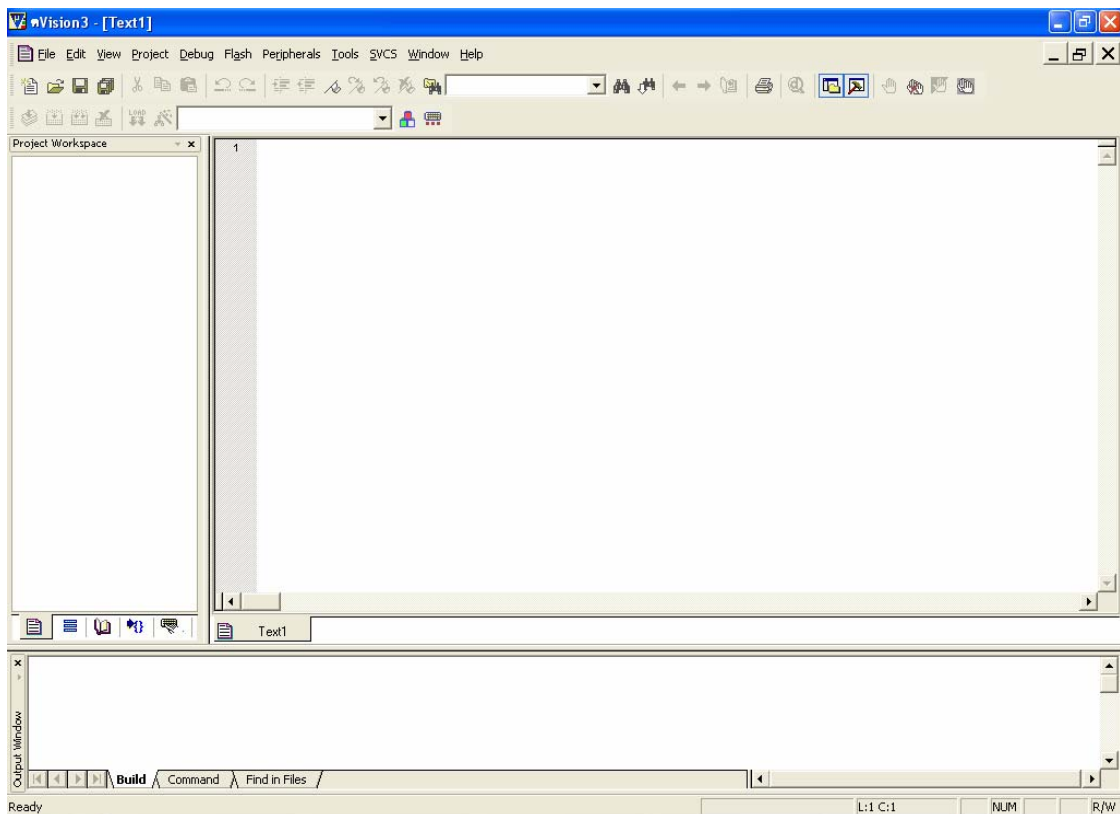
5.1 X-TAL is 19.6608 MHz and set the MCU Memory is condition to translate Keil ARM Program as in the picture



5.2 **Output**, click the default value of Create HEX File and set HEX Form as HEX-386 then click OK as in the picture



6. Start to write Source Code with C Language by click **File** → **New...** The Text File is displayed. In the first time, you should set its name as "Text1" as in the picture



In this step, you should write Source Code with C Language in the Keil ARM's form as in the example

```

/*****/
/* Examples Program For "ET-ARM STAMP LPC2119" Board */
/* Target MCU   : Philips ARM7-LPC2119          */
/*              : X-TAL : 19.6608 MHz          */
/*              : Run Speed 58.9824MHz (With PLL) */
/* Compiler     : Keil ARM V2.32a             */
/* Last Update  : 1/September/2005           */
/* Function     : Example Use GPIO-1on Output Mode */
/*              : LED Blink on GPIO1.16       */
/*****/

#include <LPC21xx.H>                // LPC2119 MPU Register

/* pototype section */
void delay_led(unsigned long int); // Delay Time Function

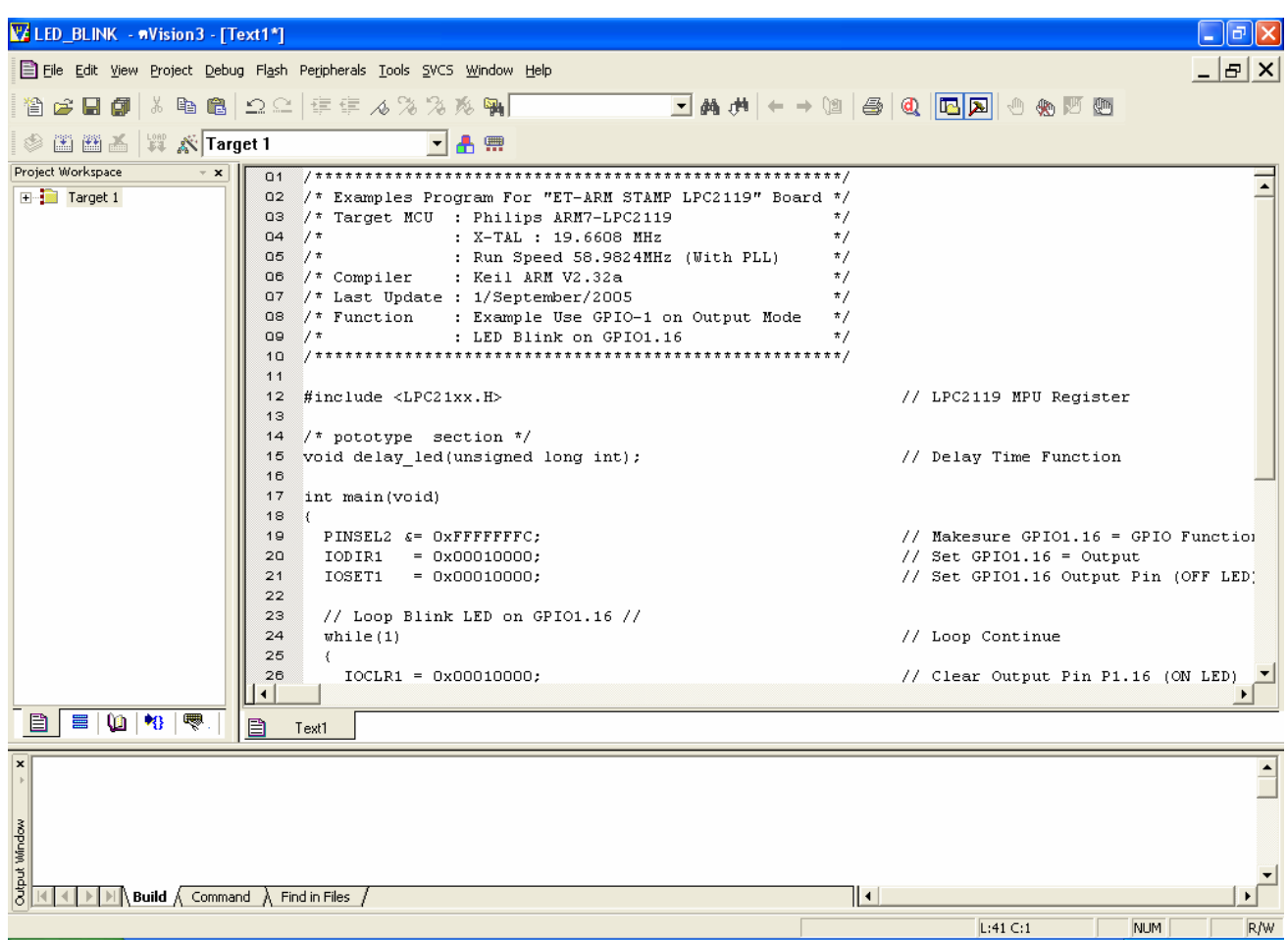
int main(void)
{
    PINSEL2  &= 0xFFFFFFF0;        // Set GPIO1.16 = GPIO Function
    IODIR1   = 0x00010000;        // Set GPIO1.16 = Output
    IOSET1   = 0x00010000;        // Set GPIO1.16 Pin (OFF LED)

    // Loop Blink LED on GPIO1.16 //
    while(1)                        // Loop Continue
    {
        IOCLR1 = 0x00010000;        // Clear Pin P1.16 (ON LED)
        delay_led(1500000);        // Display LED Delay
        IOSET1 = 0x00010000;        // Set Pin P1.16 (OFF LED)
        delay_led(1500000);        // Display LED Delay
    }
}

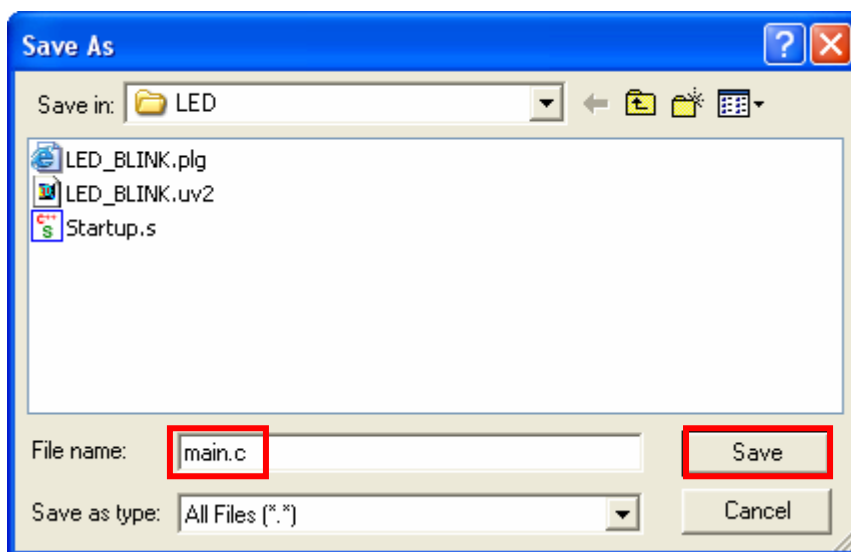
/*****/
/* Delay Time Function */
/*   1-4294967296      */
/*****/
void delay_led(unsigned long int count1)
{
    while(count1 > 0) {count1--;}    // Loop Decrease Counter
}

```

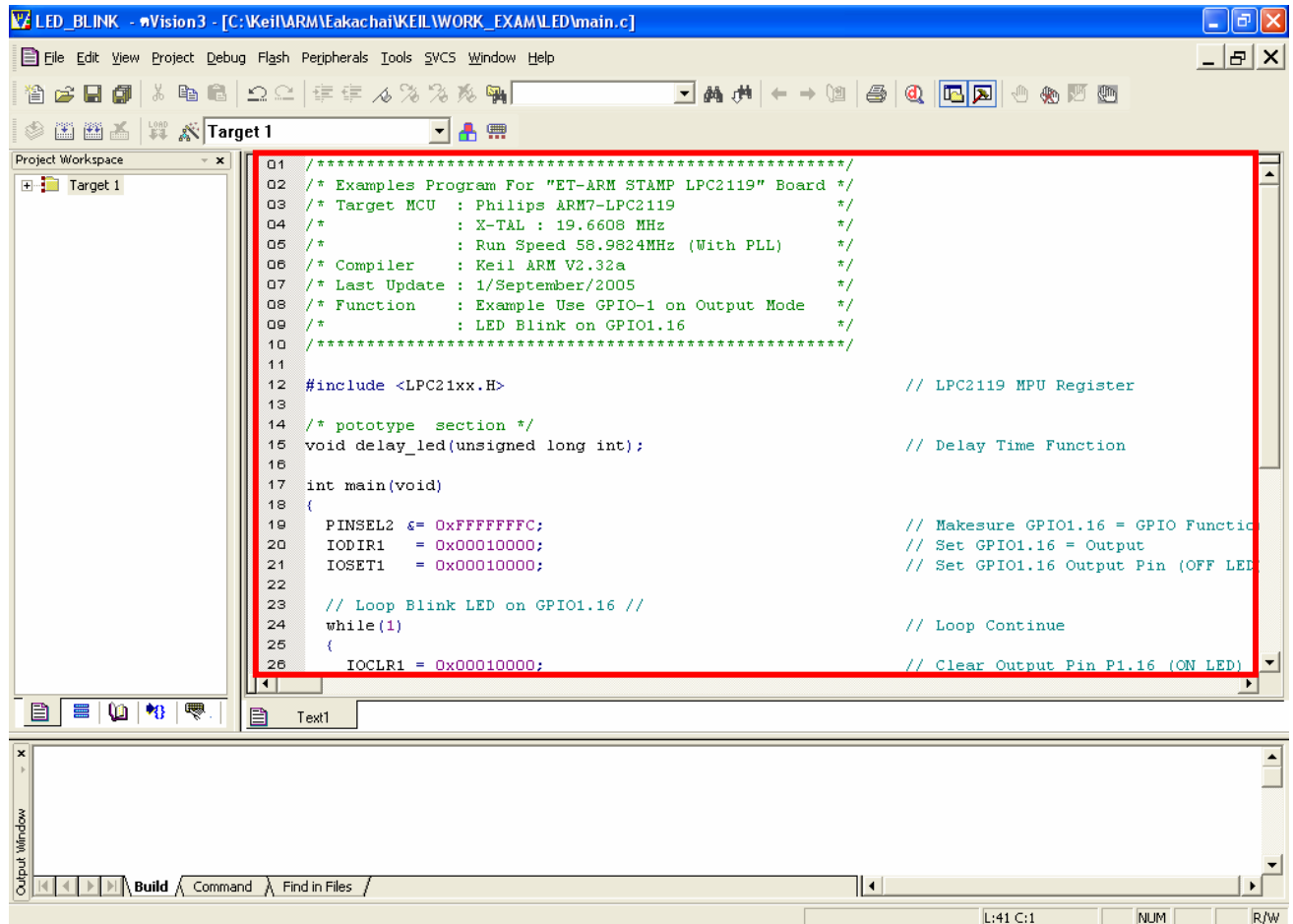
**An example of a blinker Program at GPIO1.16**



After finished to write program in C Language, you click save this file with its surname as ".C". For example, **File** → **Save As...** and then set its file name as ".main.c" as in the picture



After saved file, colour of letters in this program are changed with its function such as Comment, Variable, and Order. It is the good point of Keil uVision3 program and users use it easily as in the picture



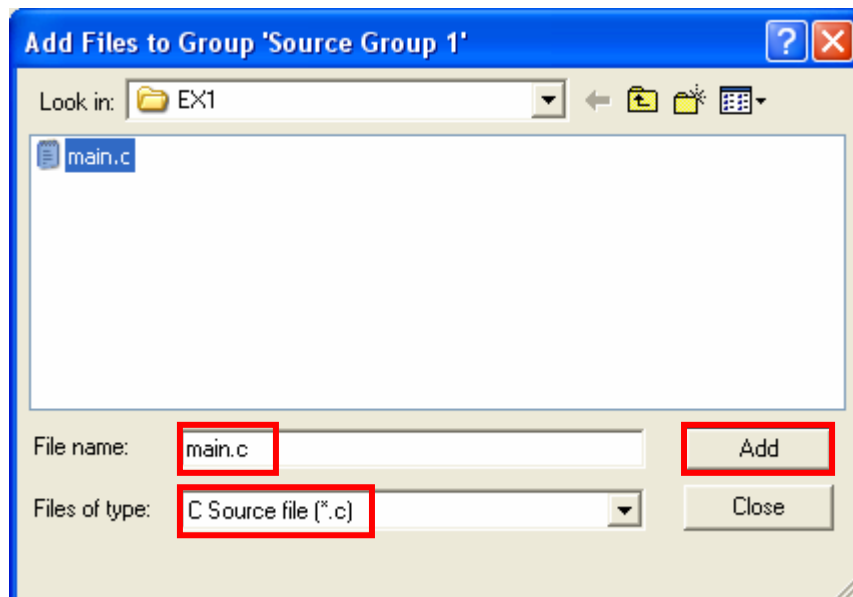
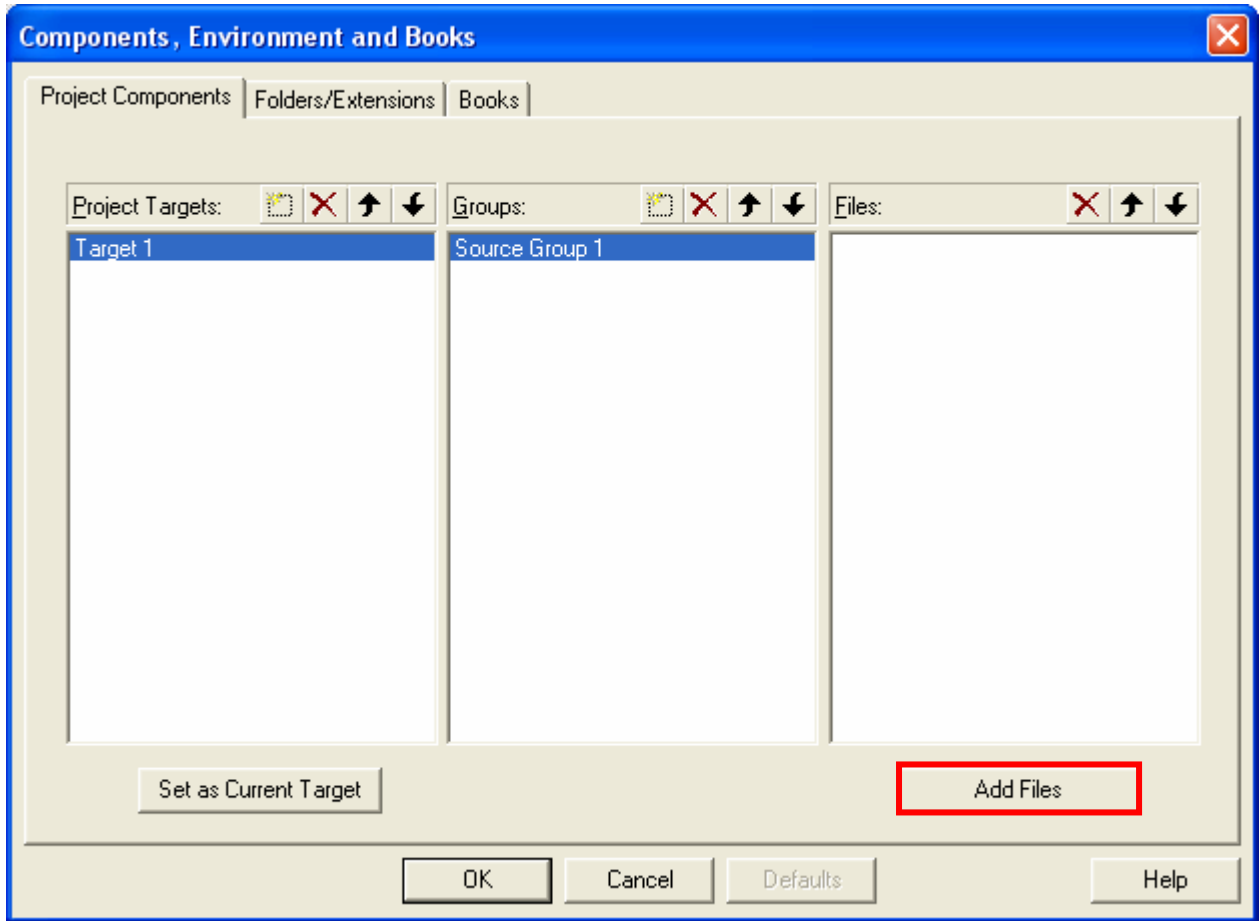
7. Add File with Project File by click **Project** → **Components, Environment, BOOKs...** then select Tab **Project Components** and then select **Add File**

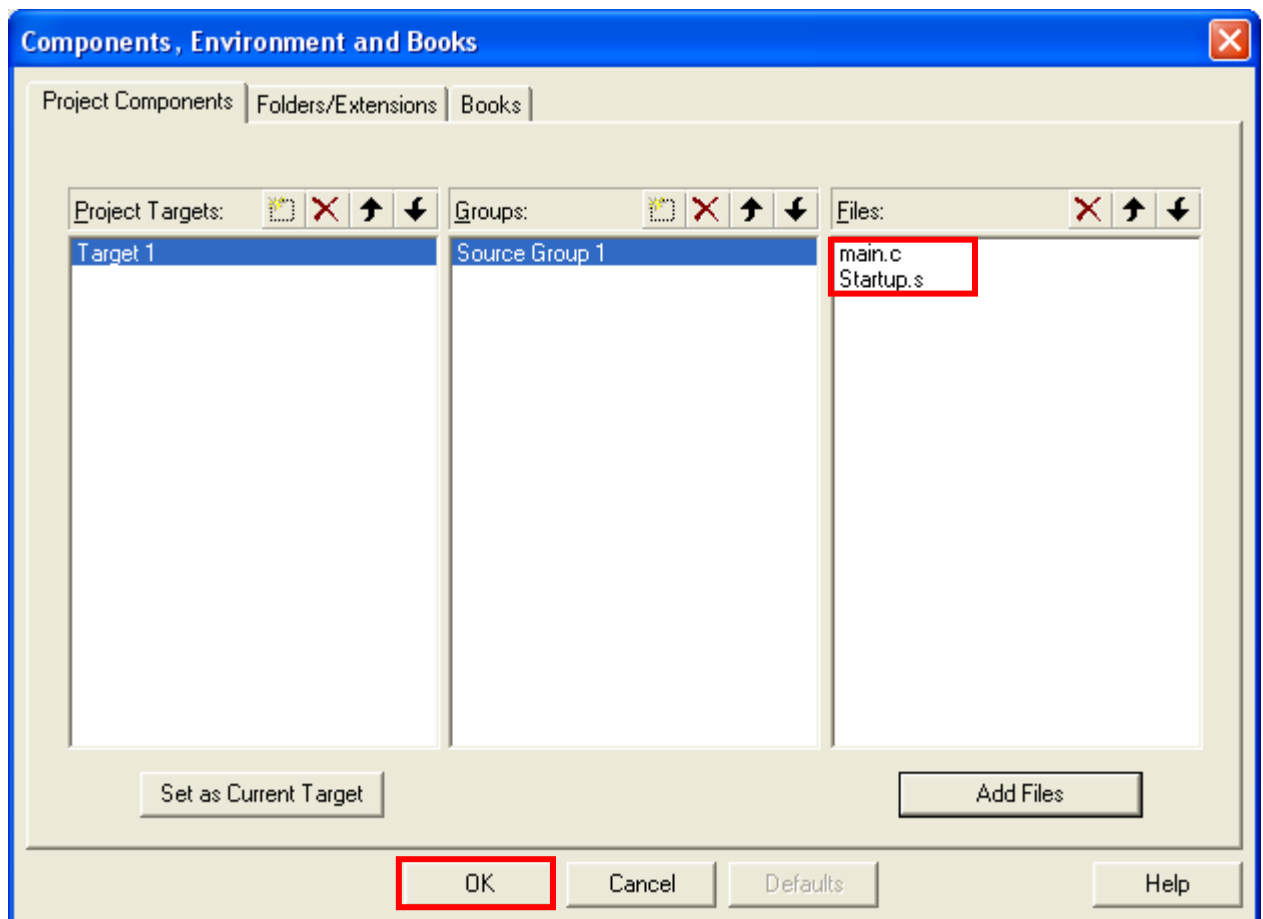
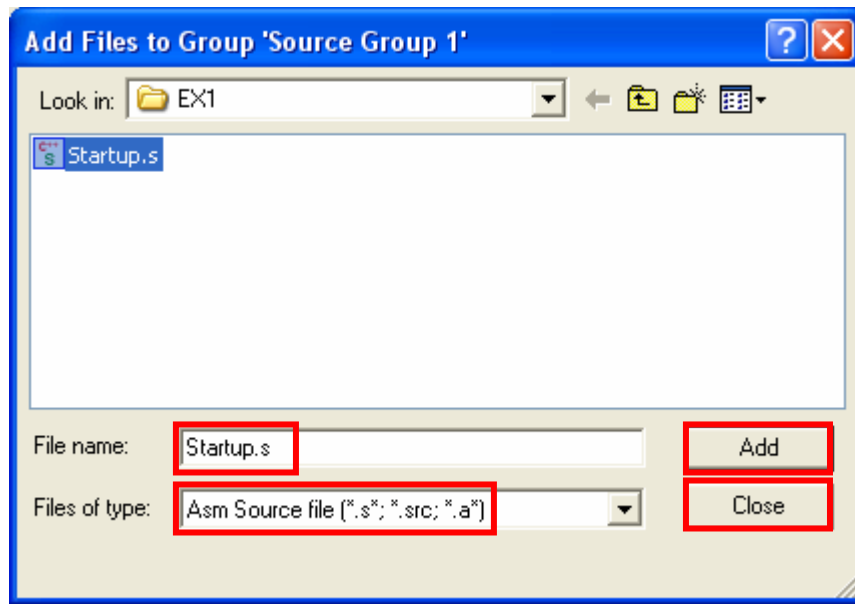
In the first time, you should select Files of type as "C Source files (\*.c)" because it shows other files name in C Source Code. Click icon file named "main.c" then select **Add** file named "Startup.s" to Project File

Set new File of type as "ASM Source files (\*.s\*;\*.src;\*.a\*)". Its file named Startup.s is shown in file name blank, click icon file "Startup.s", and then select **Add**



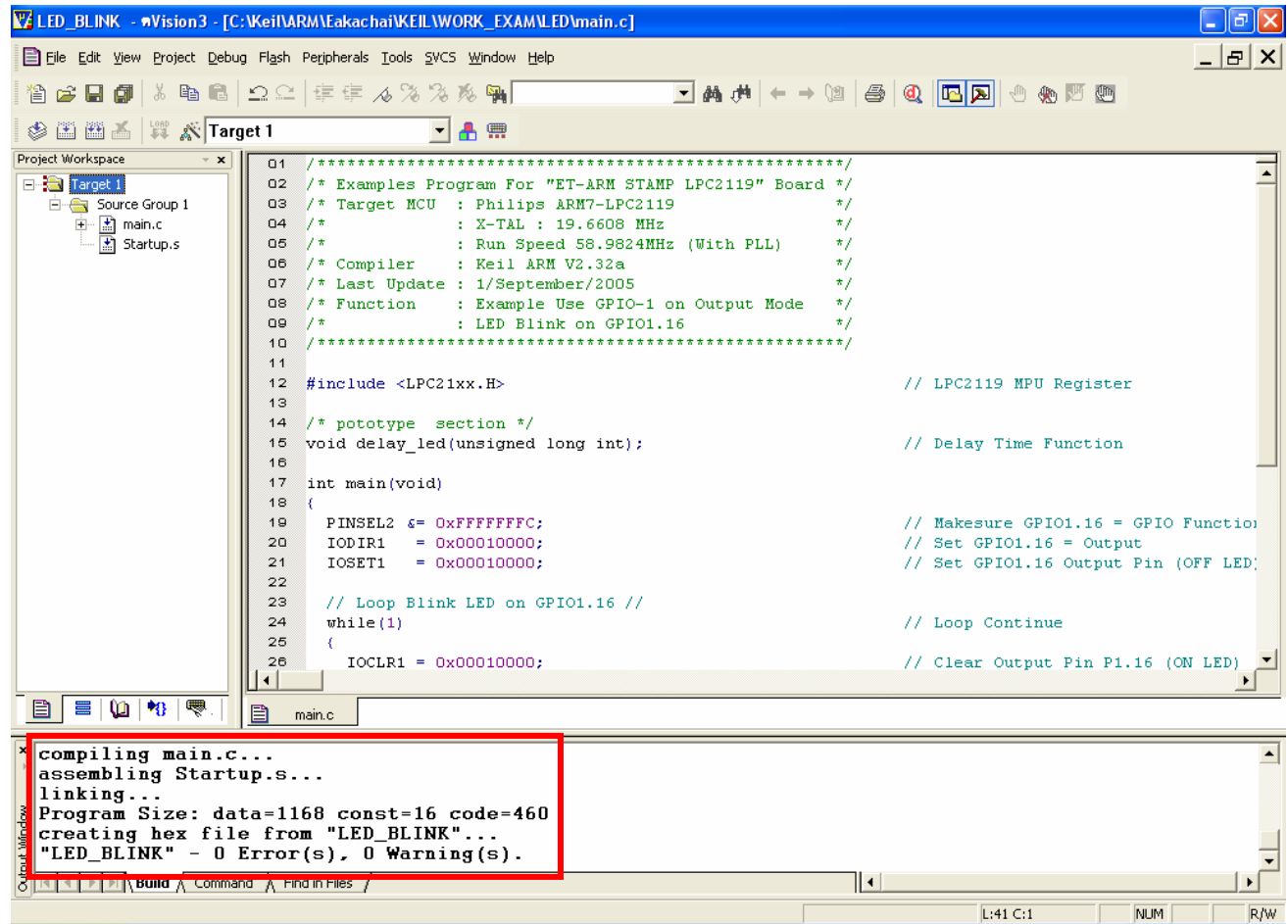
After Add file named "main.c" and "Startup.s" to Project File completely, select **Close** as in the picture





After both of file named "main.c" and "Startup.s" are added to Project File , in the Tab of File displays theirs names

8. Translate program by click **Projects** → **Rebuild all target files**. Keil uVision3 Program can order Keil ARM Program to translate program immediately



```

01 /*****
02 /* Examples Program For "ET-ARM STAMP LPC2119" Board */
03 /* Target MCU : Philips ARM7-LPC2119 */
04 /*      : X-TAL : 19.6608 MHz */
05 /*      : Run Speed 58.9824MHz (With PLL) */
06 /* Compiler : Keil ARM V2.32a */
07 /* Last Update : 1/September/2005 */
08 /* Function : Example Use GPIO-1 on Output Mode */
09 /*      : LED Blink on GPIO1.16 */
10 *****/
11
12 #include <LPC21xx.H> // LPC2119 MPU Register
13
14 /* pototype section */
15 void delay_led(unsigned long int); // Delay Time Function
16
17 int main(void)
18 {
19     PINSEL2 &= 0xFFFFF0C; // Makesure GPIO1.16 = GPIO Function
20     IODIR1 = 0x00010000; // Set GPIO1.16 = Output
21     IOSET1 = 0x00010000; // Set GPIO1.16 Output Pin (OFF LED)
22
23     // Loop Blink LED on GPIO1.16 //
24     while(1) // Loop Continue
25     {
26         IOCLR1 = 0x00010000; // Clear Output Pin P1.16 (ON LED)

```

```

compiling main.c...
assembling Startup.s...
linking...
Program Size: data=1168 const=16 code=460
creating hex file from "LED_BLINK"...
"LED_BLINK" - 0 Error(s), 0 Warning(s).

```

After translated program correctly without any mistake (0 Error and 0 Warning), you will get the HEX File name as same as the Project File name. Users can download HEX File to MCU immediately

### The Instructions of Initial MCU before Main Program Starting to Work

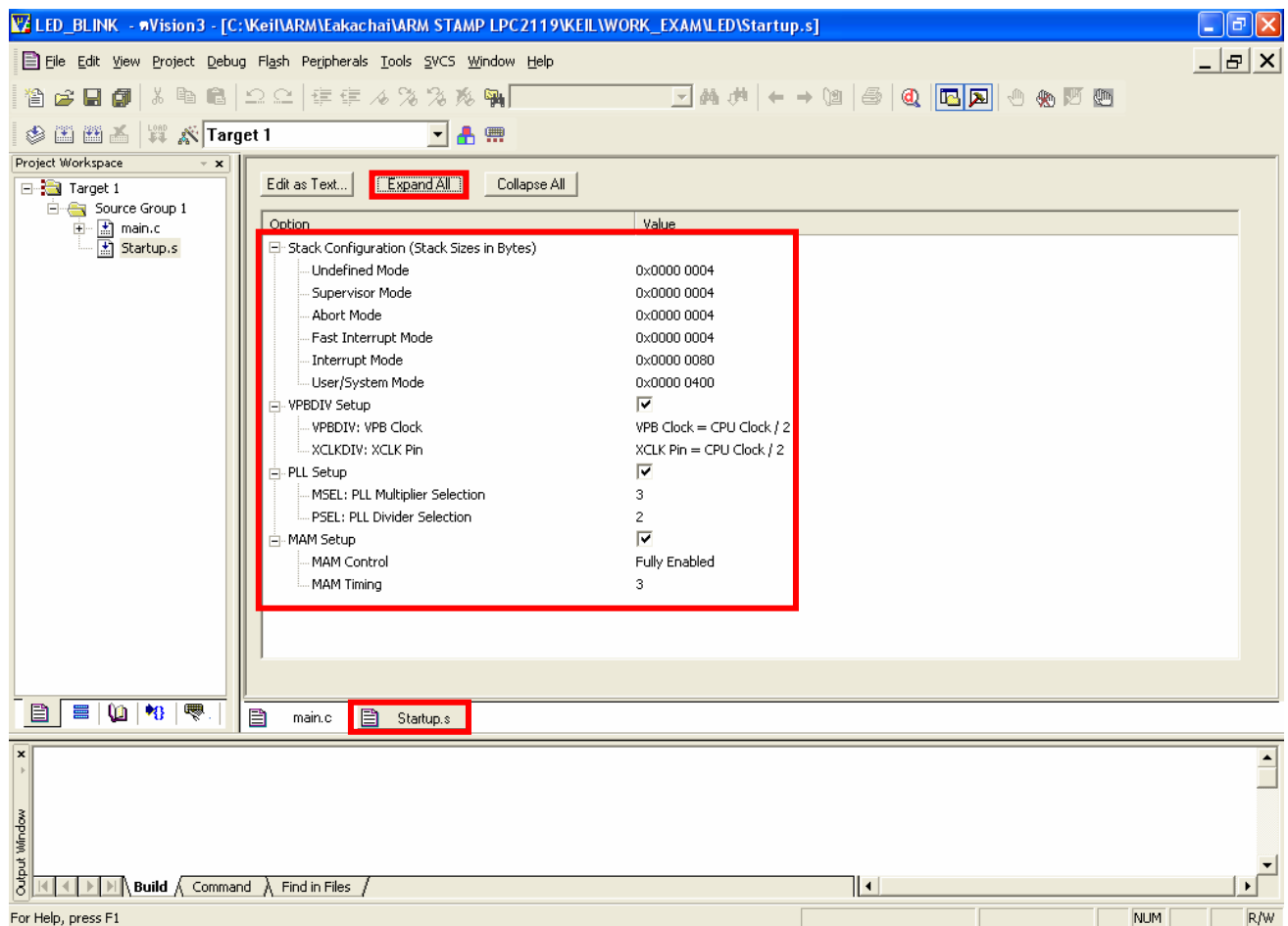
Your MCU will be the most efficient in high speed to collect data and running program, if you set default values as;

- Default value of PLL is Processor Clock (cclk)=58.9824MHz, in case of, using XTAL=19.6608MHz, default value is M(Multiply)=3 and P(Divide)=2 and FFCO=235.9296MHz
- Default value of VPB Clock(pclk) is a half of cclk(29.4912MHz)
- Default value of MAM Timing is 3 cycle of cclk (MAMTIM=0x03)
- Default value of MAM Mode is Full Enable (MAMCR=0x02)

There's two methods to set default value as in the example. Firstly, writing Code in all written program by yourself and secondly, copying Startup File into your Project File. There's two ways to verify the default values of Startup File. First, correct Code directly and second, set default value of Startup File from Keil uVision3 Program. In this case, correcting default value of them by Keil uVision3 Program is easier

### Verification of Startup File

Function of Startup File is contained beginning orders of MCU before start to run with written program by yourself. Function of Program in Startup File is Initial MCU in the necessary part and then jump to run in written Program in C Language. The method to verify Startup is clicking Tab of Startup File then selecting "Expand All". The correct default value of Startup File is like in the picture



[-] Stack Configuration (Stack Sizes in Bytes)	
Undefined Mode	0x0000 0004
Supervisor Mode	0x0000 0004
Abort Mode	0x0000 0004
Fast Interrupt Mode	0x0000 0004
Interrupt Mode	0x0000 0080
User/System Mode	0x0000 0400
[-] VPBDIV Setup	<input checked="" type="checkbox"/>
VPBDIV: VPB Clock	VPB Clock = CPU Clock / 2
XCLKDIV: XCLK Pin	XCLK Pin = CPU Clock / 2
[-] PLL Setup	<input checked="" type="checkbox"/>
MSEL: PLL Multiplier Selection	3
PSEL: PLL Divider Selection	2
[-] MAM Setup	<input checked="" type="checkbox"/>
MAM Control	Fully Enabled
MAM Timing	3

*Picture shows setting the default value of Startup File in LPC2119*

## An example of Keil ARM's C Language Code for Initial LPC2119

If you want to write Initial Program for running MCU by yourself, you should add orders in the beginning of Main Program as in the example

```
// Initial PLL & VPB Clock For ET-ARM7 STAMP LPC2119
// Start of Initial PLL for Generate Processor Clock
// PLL Configuration Setup
// X-TAL = 19.6608MHz
// M(Multiply) = 3
// P(Divide) = 2
// Processor Clock(cclk) = M x OSC
//                               = 3 x 19.6608MHz
//                               = 58.9824MHz
// FCCO = cclk x 2 x P
//       = 58.9824 x 2 x P
//       = 235.9296 MHz
// VPB Clock(pclk) = 29.4912MHz
// Start of Initial PLL for Generate Processor Clock
PLLCFG &= 0xE0;           // Reset MSEL0:4
PLLCFG |= 0x02;          // MSEL(PLL Multiply) = 3
PLLCFG &= 0x9F;          // Reset PSEL0:1
PLLCFG |= 0x20;          // PSEL(PLL Devide) = 2

PLLCON &= 0xFC;          // Reset PLLC,PLLE
PLLCON |= 0x01;          // PLLE = 1 = Enable PLL

PLLFEEED = 0xAA;         // Start Update PLL Config
PLLFEEED = 0x55;
while (!(PLLSTAT & 0x00000400)); // Wait PLL Lock bit

PLLCON |= 0x02;          // PLLC = 1 (Connect PLL Clock)
PLLFEEED = 0xAA;         // Start Update PLL Config
PLLFEEED = 0x55;

VPBDIV &= 0xFC;          // Reset VPBDIV
VPBDIV |= 0x02;          // VPB Clock(pclk) = cclk / 2
// End of Initial PLL for Generate Processor Clock

// Start of Initial MAM Function
MAMCR = 0x00;            // Disable MAM Function
MAMTIM = 0x03;           // MAM Timing = 3 Cycle of cclk
MAMCR = 0x02;            // Enable MAM = Full Function
// End of Initial MAM Function

// Start of Main Function Here
.
```

*An example of Code for Initial LPC2119 before Main Program start to work*