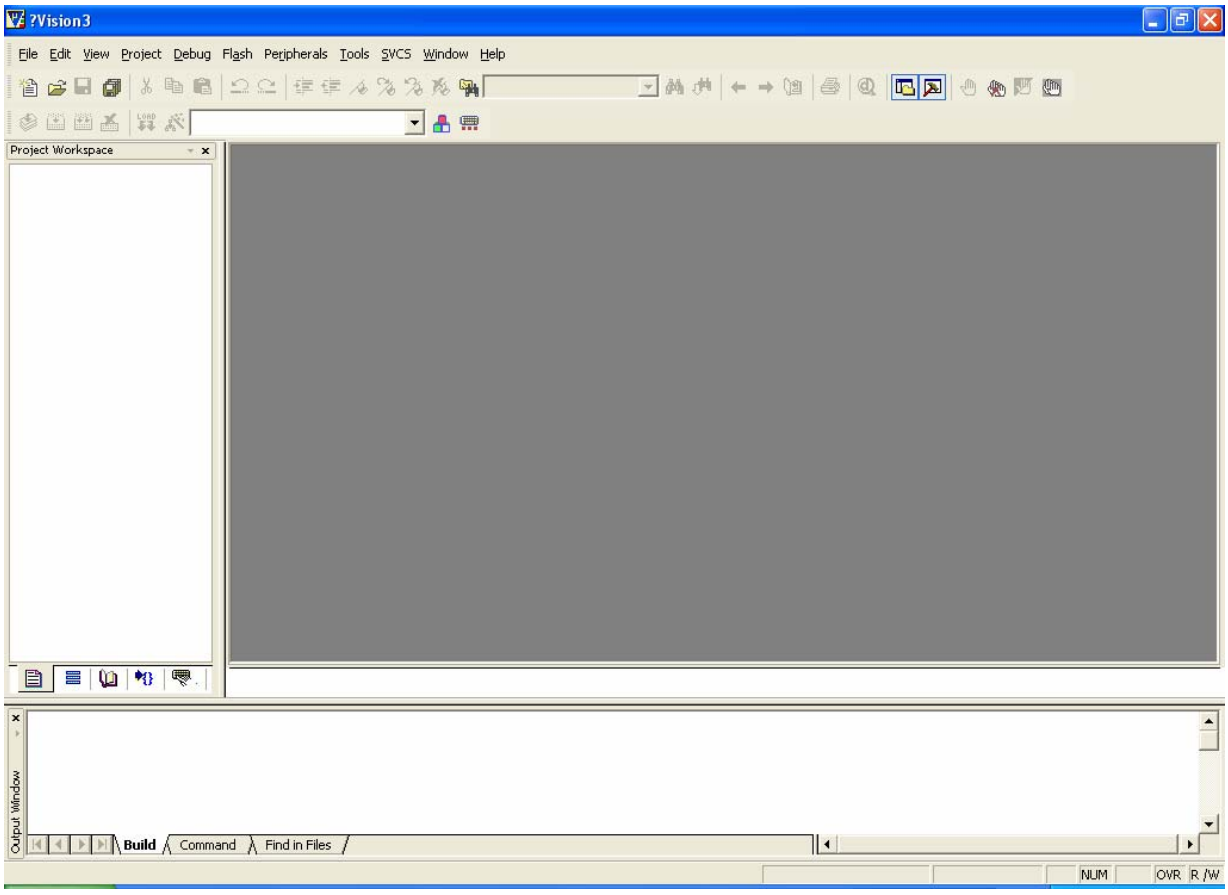
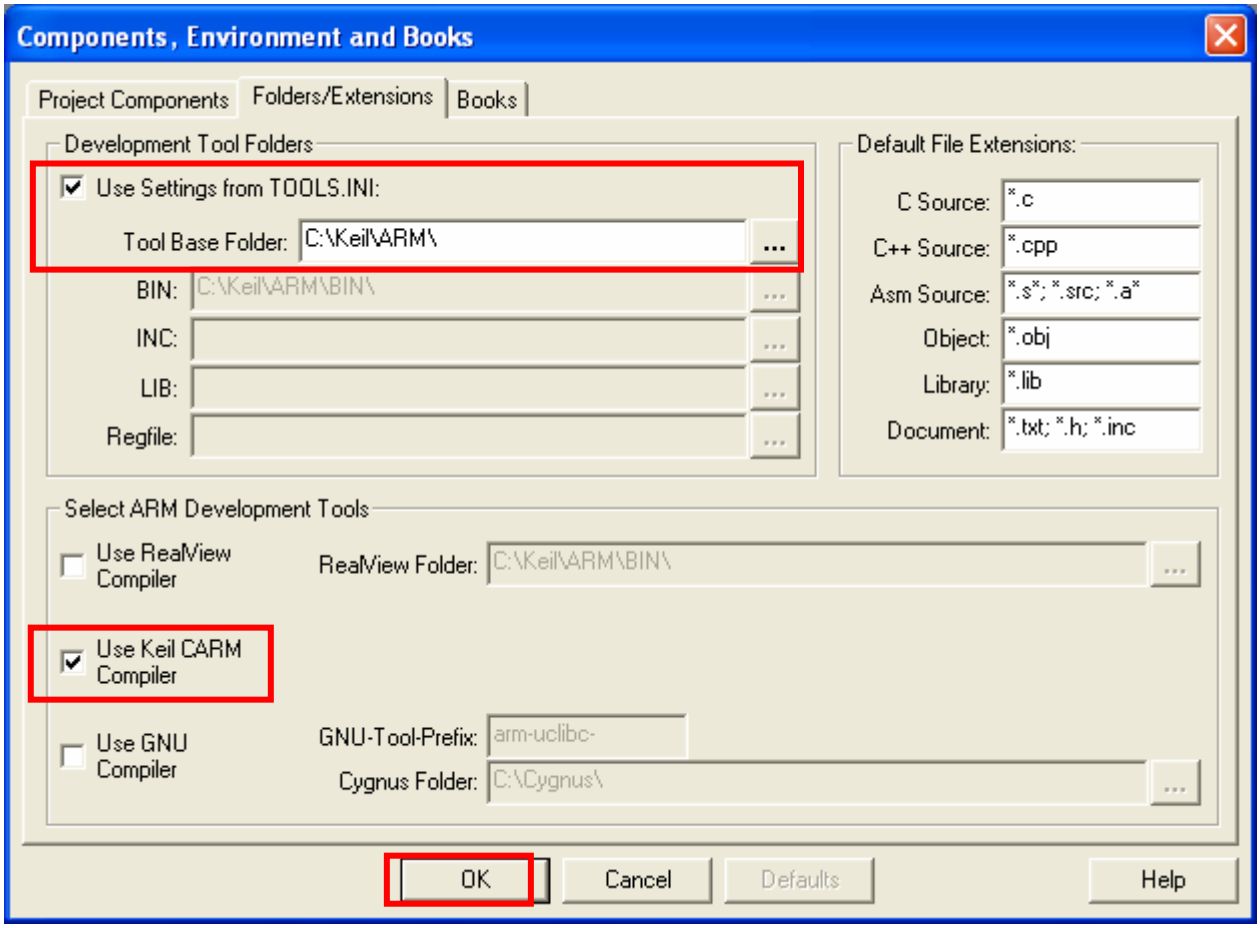
An Example of using Keil uVision3 to create Project File of Keil ARM

In this case, we will mention about the proceeding to write program by using C Language Program that is Keil-CARM. It is used to interpret command under Program Text Editor of Keil (Keil uVision3). We only mention about the proceeding to configure Option value for connection commands of interpretation program together by using Keil-CARM through Keil uVision3. For more detailed commands and functions usage for writing program by Keil-CARM, user can learn them by self from User's Manual command of Keil-CARM. We can summarize the proceeding to configure default values of Keil uVision3 for using with Keil-CARM as follows;

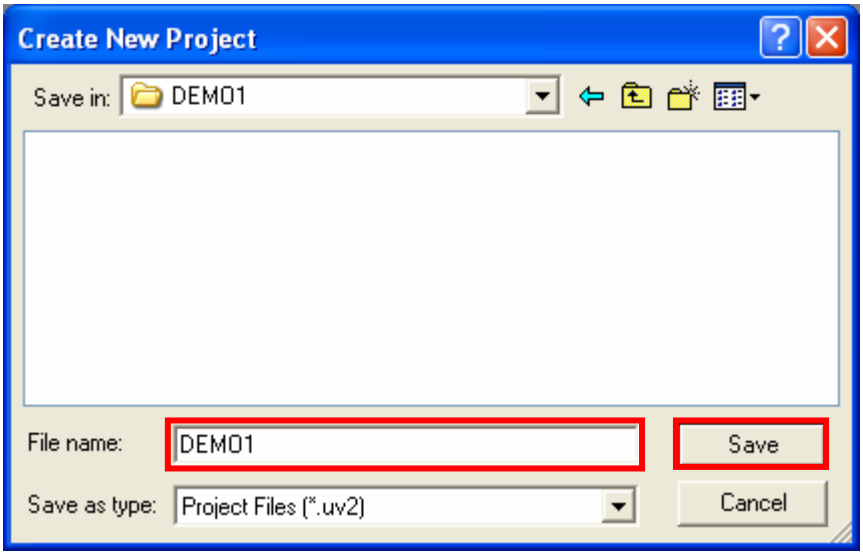
1. Open program Keil uVision3 that is a program Text Editor of Keil-CARM, it is used to write C Language Source Code program and the feature of this program is look like in the picture below.



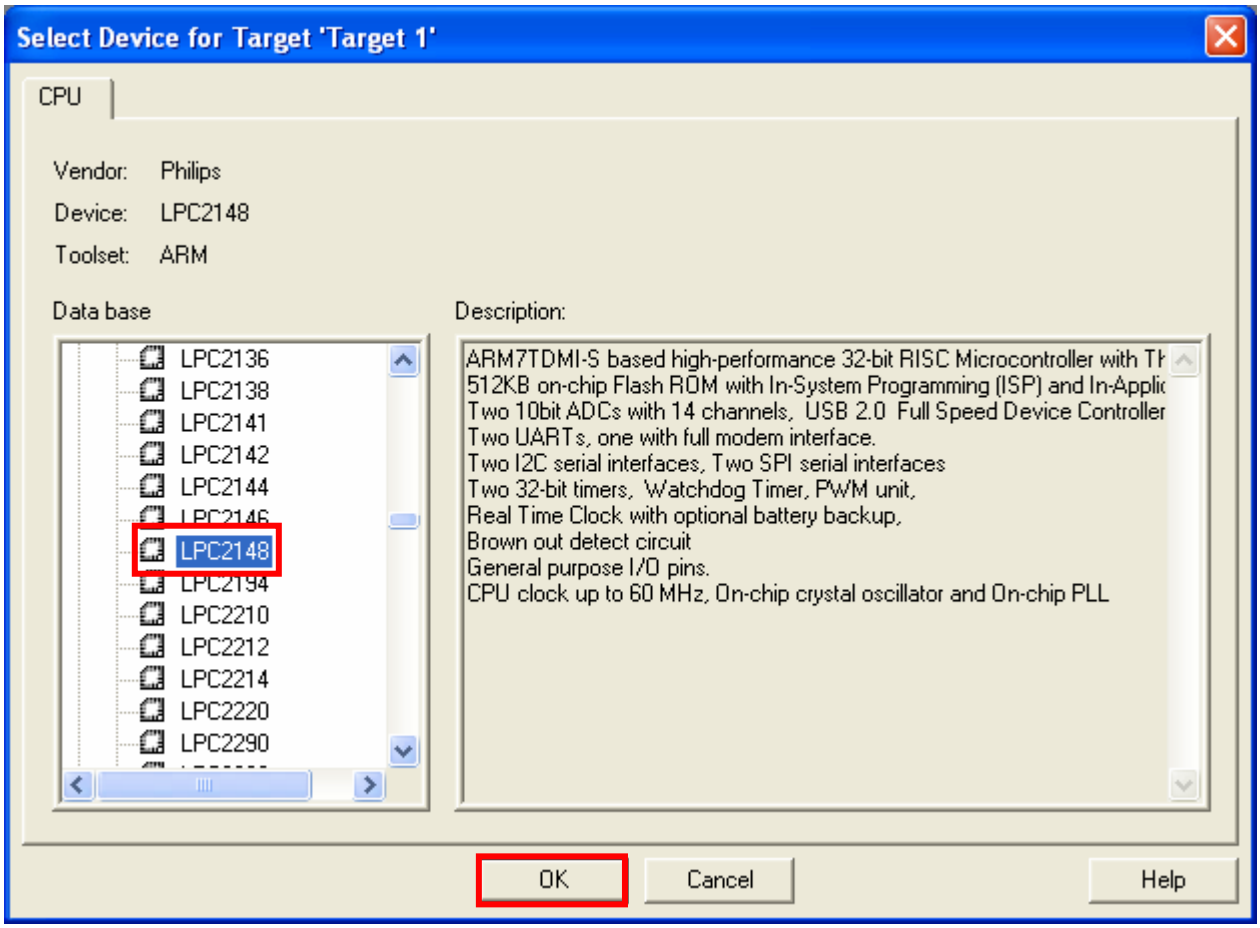
2. Configure default values to interpret commands of uVision3 and can be used with Program Keil uVision3 and Keil-CARM. Click **Project → Components, Environment, Books…** and then select default value for Compiler from the title **Select ARM Development Tools** that has 3 modes; **Use Keil-CARM Tools, Use GNU Tools** and **Use ARM Tools.** In this case, we must select **"Use Keil ARM Tools"**, and then we must configure position of folder to store default values of program Keil ARM. Generally, it is in **"C:\Keil\ARM\"** but if we install Keil in other folder, we must change the format of it suitably and corresponding with truly usage as in the picture below.



3. Create new Project File by using command **Project → New Project** and then configure or create position Folder that we want to save new Project File with preferred Project File name. For example, if we create new Project File named **DEMO1** and wants to save it into Folder named **DEMO1**; we can configure position of Folder and Project File name by self. After we configure Project File name in the blank of File Name successfully, click **Save** to save new Project File as in the picture below.

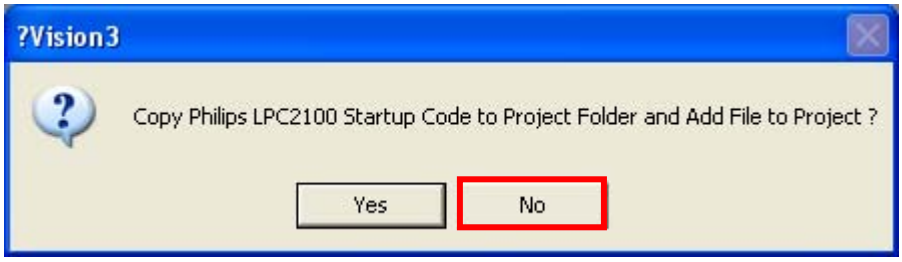


After we configure new Project File name and save it completely, Program will wait for user to configure MCU number that is used in the saved Project File. If using with Board "CP-JR ARM7 USB-LPC2148", we must configure MCU number to be LPC2148 from Philips and then select **OK** as in the picture below.



After we configure MCU number successfully, in this step, program will wait for user to confirm copy File Startup of Keil and wants to use it with MCU of Philips in new Project File or not. Startup File is the part to configure the default value of operation for MCU; for example, to configure Stack value and to configure value to run for Phase-Lock-Loop before start running follow by our written program. Otherwise, our written program must totally be added these commands into operation of MCU by self.

File Startup of Keil-ARM is Assembly Language File that is configured operation values with development set of Keil, so some configurations and default values are different and it makes Board "CP-JR ARM7 USB-LPC2148" can not be used with File Startup instantly. Therefore, we must modify some default value before using with program Keil-CARM. For interpretation commands, we must modify new File Startup and must set new format that is corresponding with the board's need. In this case, we will recommend selecting "No" to protect Keil uVision3 not copy File Startup of Keil-CARM to use in Project.

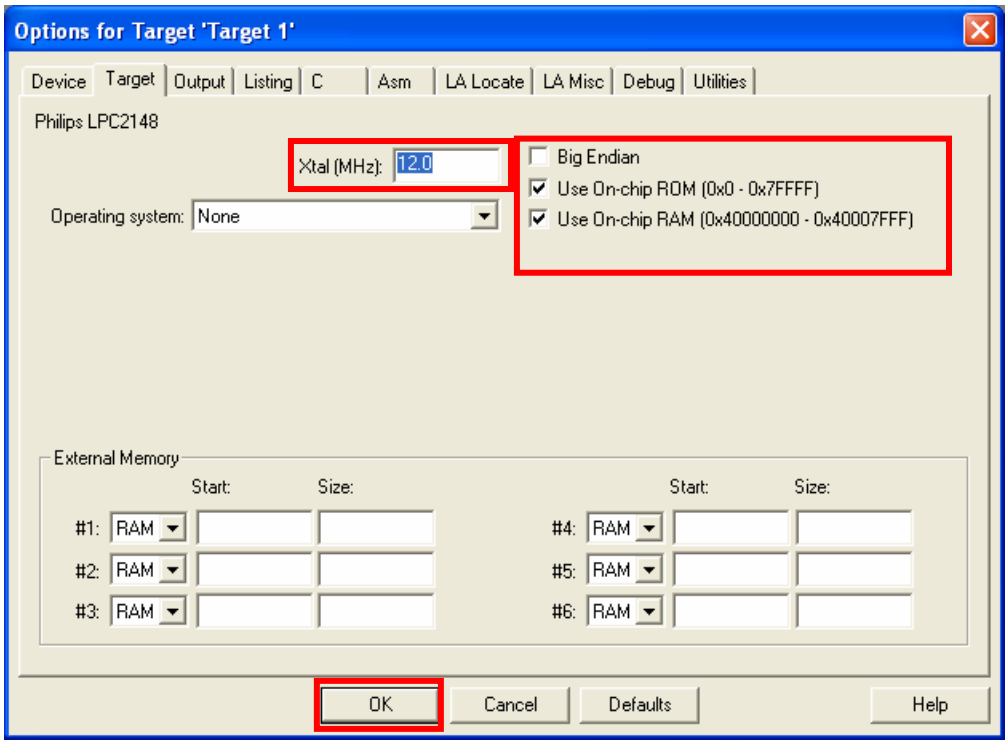


4. Copy File named **"Startup.s"** that ETT has already provided in CD-ROM and is saved in Example named **"Startup.s"**, then to place it in the same position folder of new Project File that we created completely.

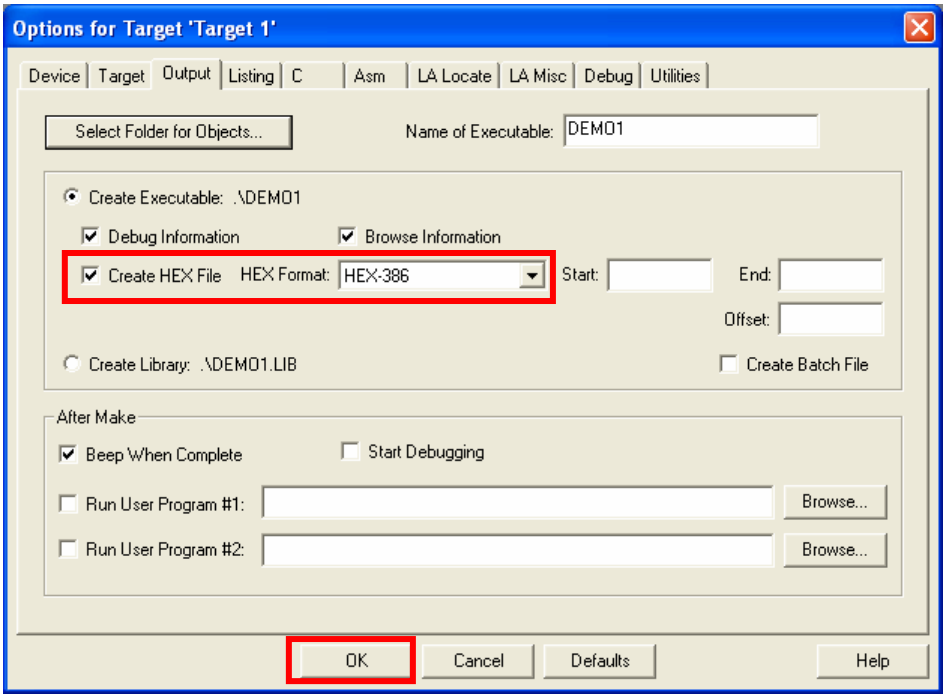
File **"Startup.s"** is a file that contains Assembly Language Commands of ARM7 to configure the necessary default value for MCU; for example, to configure Stack value into MCU, to configure Initial Phase-Lock-Loop, to configure value into MAM Function and to configure position Vectors of MCU. For using with Board "CP-JR ARM7 USB-LPC2148", if we Add File **"Startup.s"** from Keil or Copy this File from other positions, it will be effected on the operation of program in Startup because some operations are different.

5. Configure Option value of Project File by using command **Project → Option for Target 'Target 1'** and then select Tab of Target to configure value of MCU Target as follows.

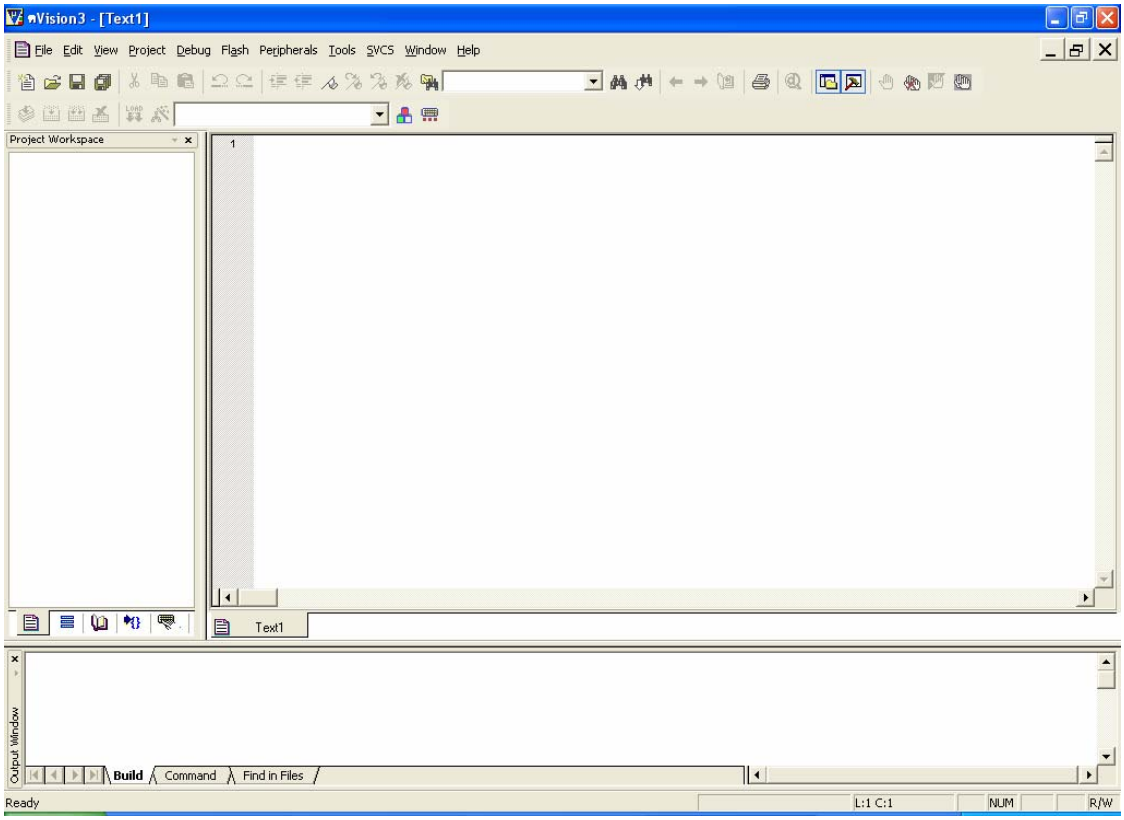
5.1 Configure X-TAL to be 12 MHz and then configure Memory internal MCU to be condition of interpretation program of Keil-CARM as in the picture below.



5.2 Output: we must click default values of Create HEX File, configure format of Hex to be HEX-386 and then select OK as in the picture below.



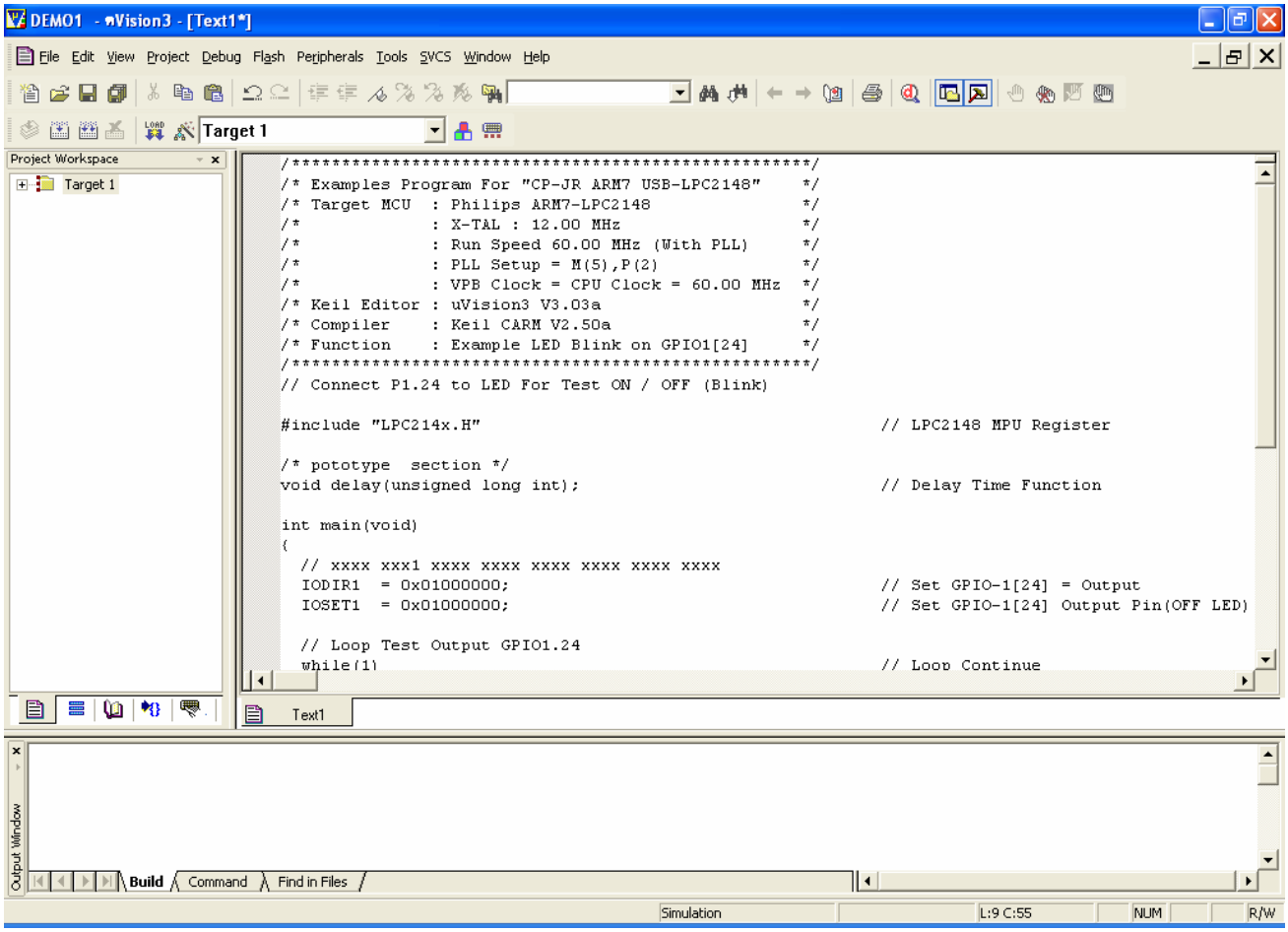
6. Start writing C Language Source Code, click command **File → New…** and we will get the available are to write Text File. In the first time, we must configure File name to be "Text" follow by the Default as in the picture below.



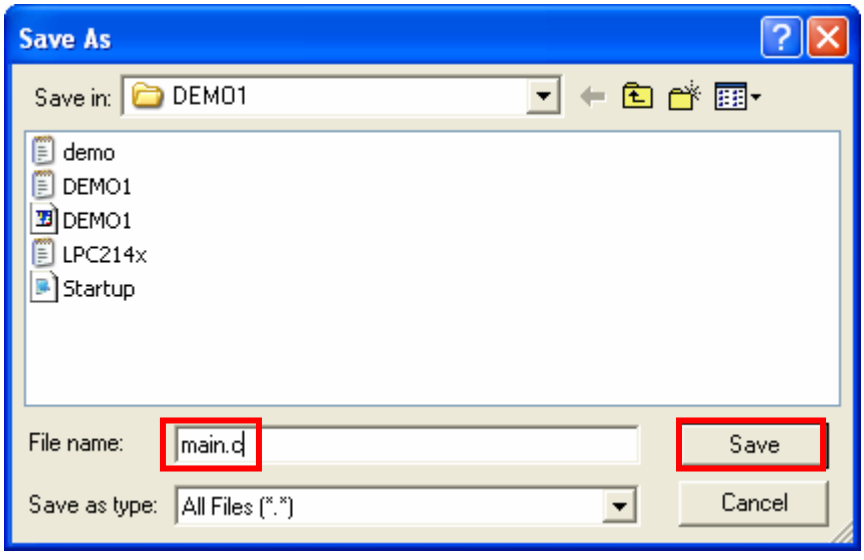
In this step, it is typing C Language Source Code in the available area under configurations of Keil-CARM and we can write program preferably as in the picture below.

```
/*************************************************/
/* Examples Program For "CP-JR ARM7 USB-LPC2148"  */
/* Target MCU  : Philips ARM7-LPC2148             */
/*           : X-TAL : 12.00 MHz                  */
/*             : Run Speed 60.00 MHz (With PLL)   */
/*             : PLL Setup = M(5),P(2)            */
/*             : VPB Clock = CPU Clock = 60 MHz   */
/* Keil Editor : uVision3 V3.03a                  */
/* Compiler    : Keil CARM V2.50a                 */
/* Function    : Example LED Blink on GPIO1[24]   */
/*************************************************/
// Connect P1.24 to LED For Test ON / OFF (Blink)

#include "LPC214x.H"               // LPC2148 MPU Register

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

int main(void)
{
  IODIR1  = 0x01000000;            // Set GPIO-1[24] = Output
  IOSET1  = 0x01000000;            // Set GPIO-1[24] Output Pin

  // Loop Test Output GPIO1.24
  while(1)                         // Loop Continue
  {
    IOCLR1 = 0x01000000;           // Clear Output GPIO1[24]
    delay(1000000);                // Display Delay

    IOSET1 = 0x01000000;           // Set Output GPIO1[24]
    delay(1000000);                // Display Delay
  }
}

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

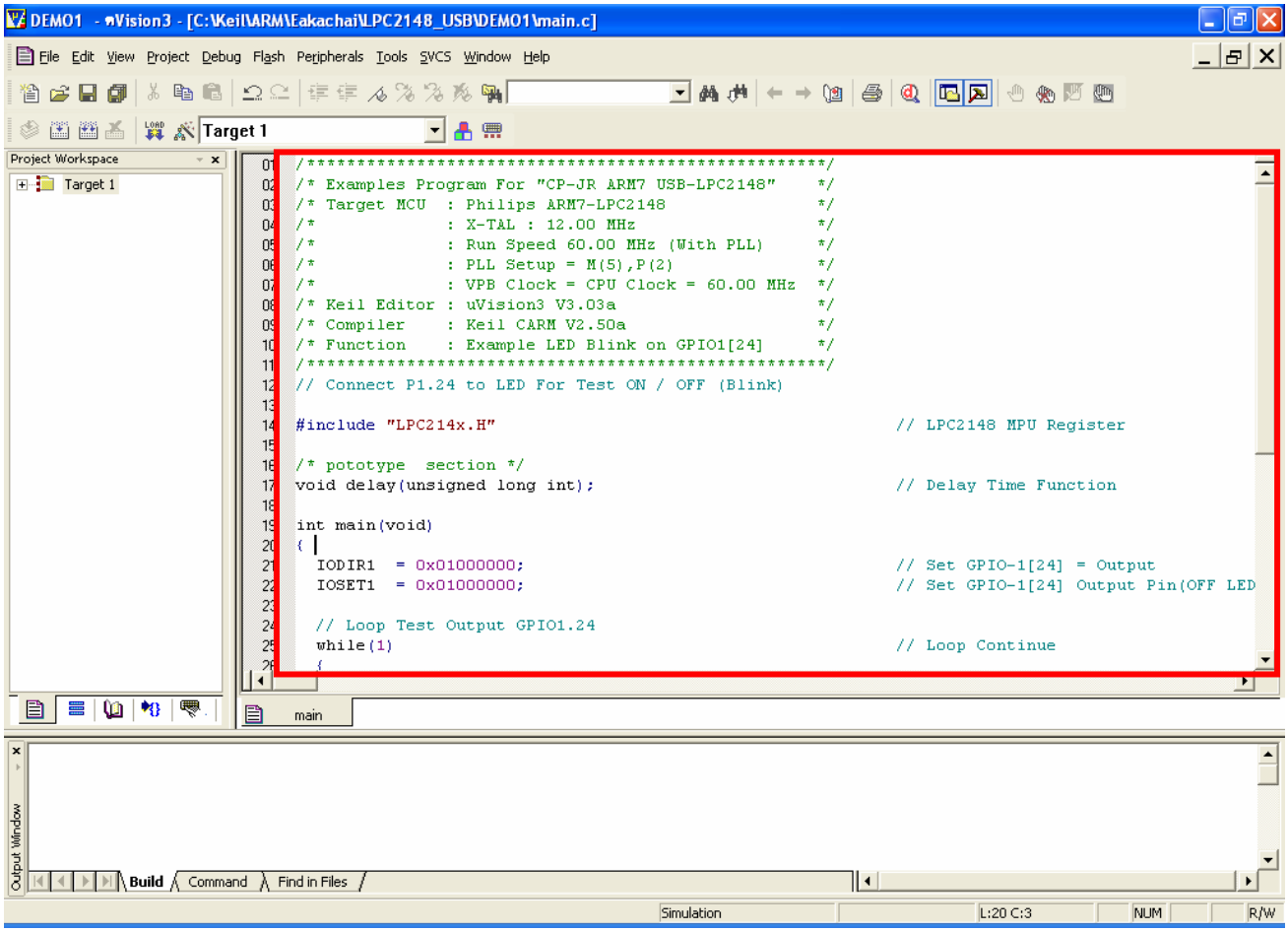
It is an example program of 1 blinking light at GPIO1.24.



After typing C Language commands completely, we must save this File and must configure File surname to be ".C". In this case, we recommend to save by using command **File → Save As**... and then configure File name and File surname as ".main.c" as in the picture below.



After save File as ".main.c" completely, we can see color of characters in program are changed follow by the functions such as Comment, Variable and Command. It is an advantage of Keil uVision3 that can extract and display characters follow by their functions, it makes user understand program and read program easily as in the picture below.

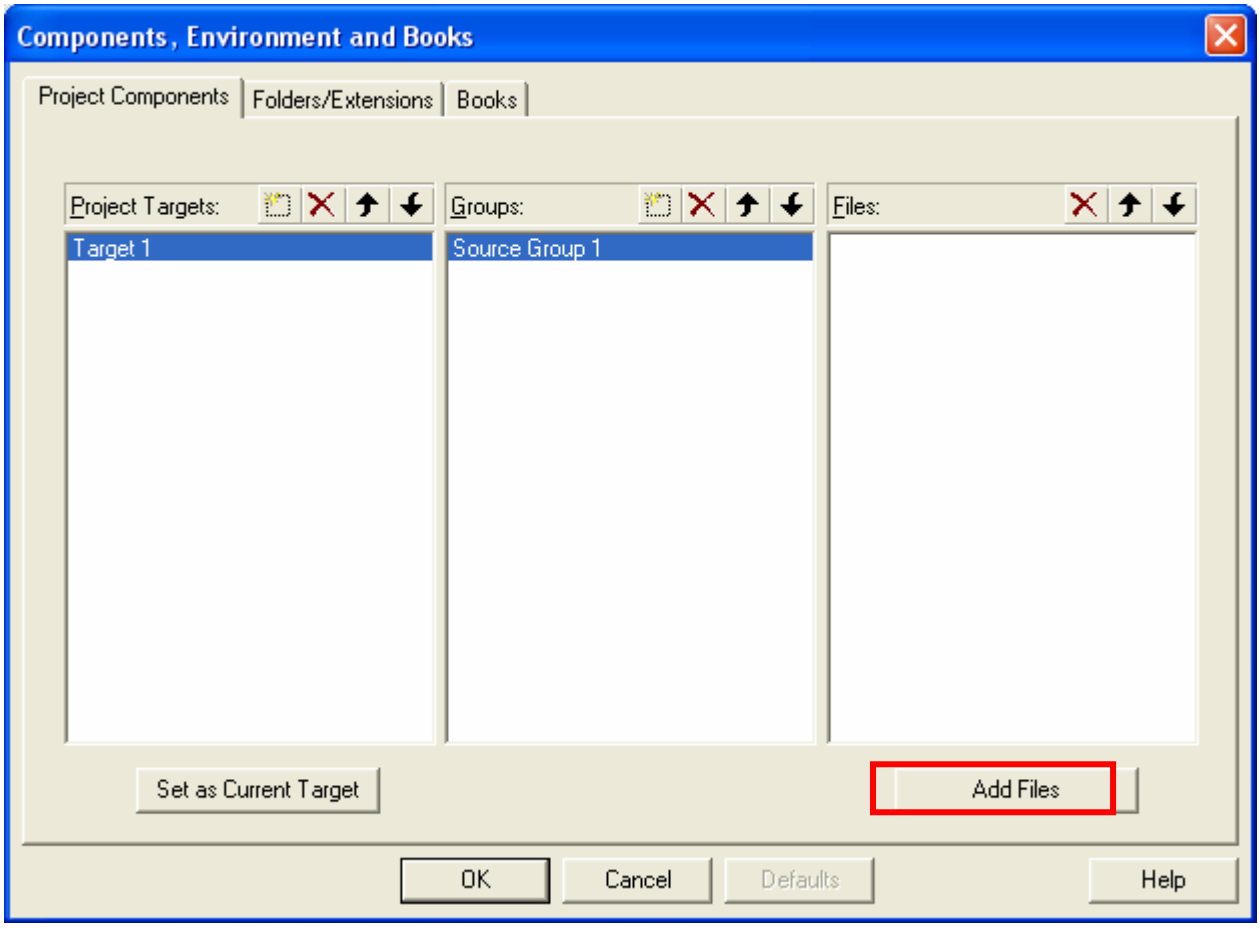


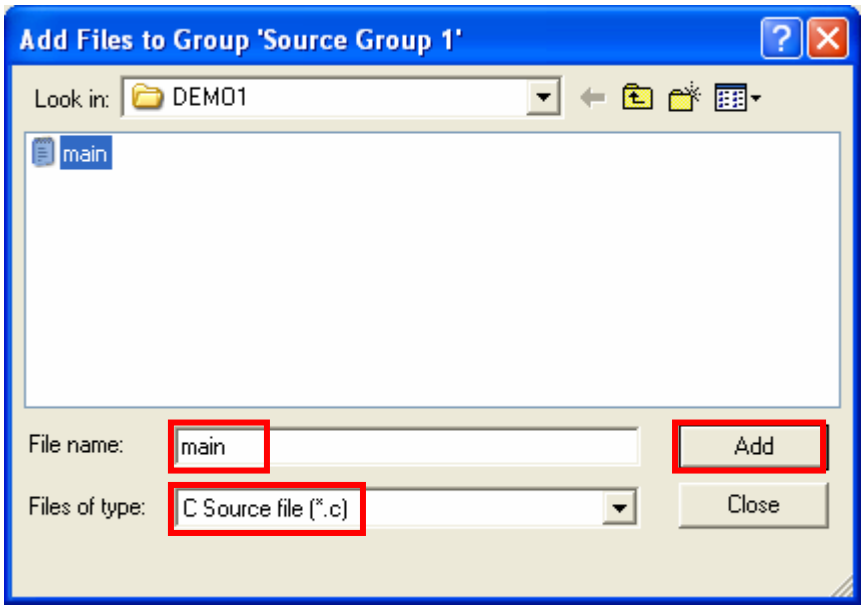
7. Add Files into Project File, click command **Project → Components, Environment, Books…**, select **Tab Project Components** and then select desired **Add File** to add into Project File.

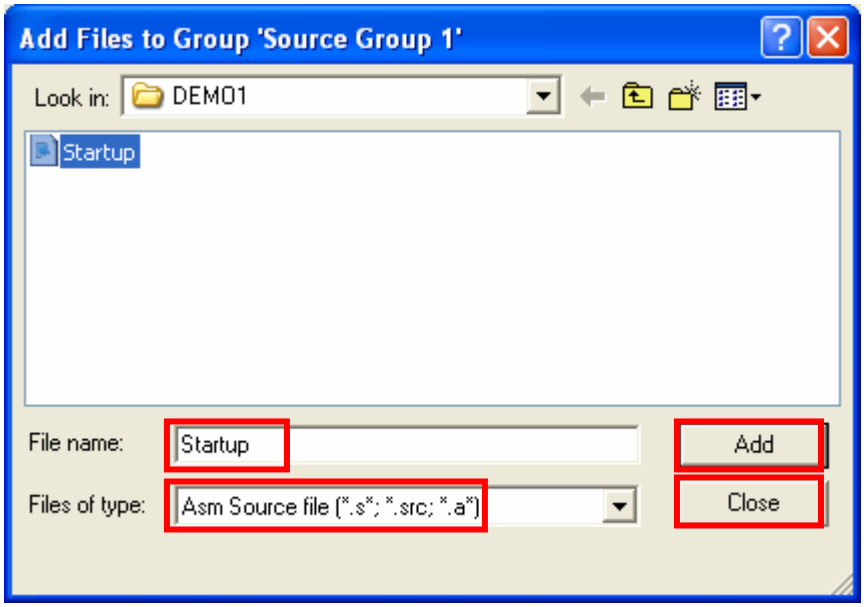
In the first time, we must select **Files of type** to be **"C Source files (*.c)"** and it will display Files name that is C Language Source Code. Click icon of File named **"main.c"** and then select **Add** File named **"Startup.s"** into Project Files that we created.

Then we must configure new **File of type** to be **"ASM Source files(*.s*;*.src;*.a*)**, it will display File name **Startup.s** in the blank of File name, so click icon of File **"Startup.s"** and then select **Add** File named **"Startup.s"** into Project Files that we created.

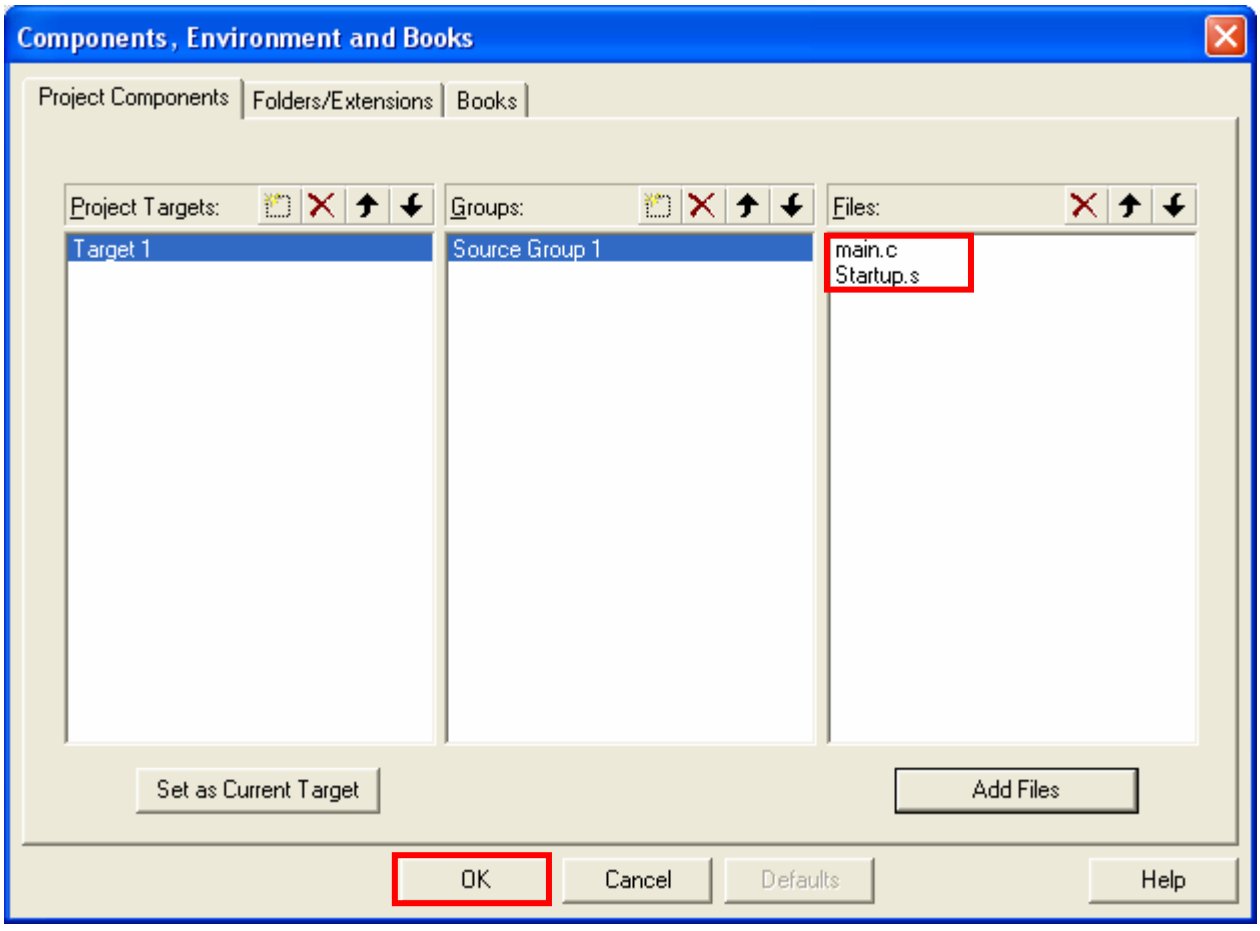
When we command **ADD** both File name **"main.c"** and **"Startup.s"** into Project File successfully, we must select **Close** to end the command **Add File** and it will display result of operation as in the picture below.





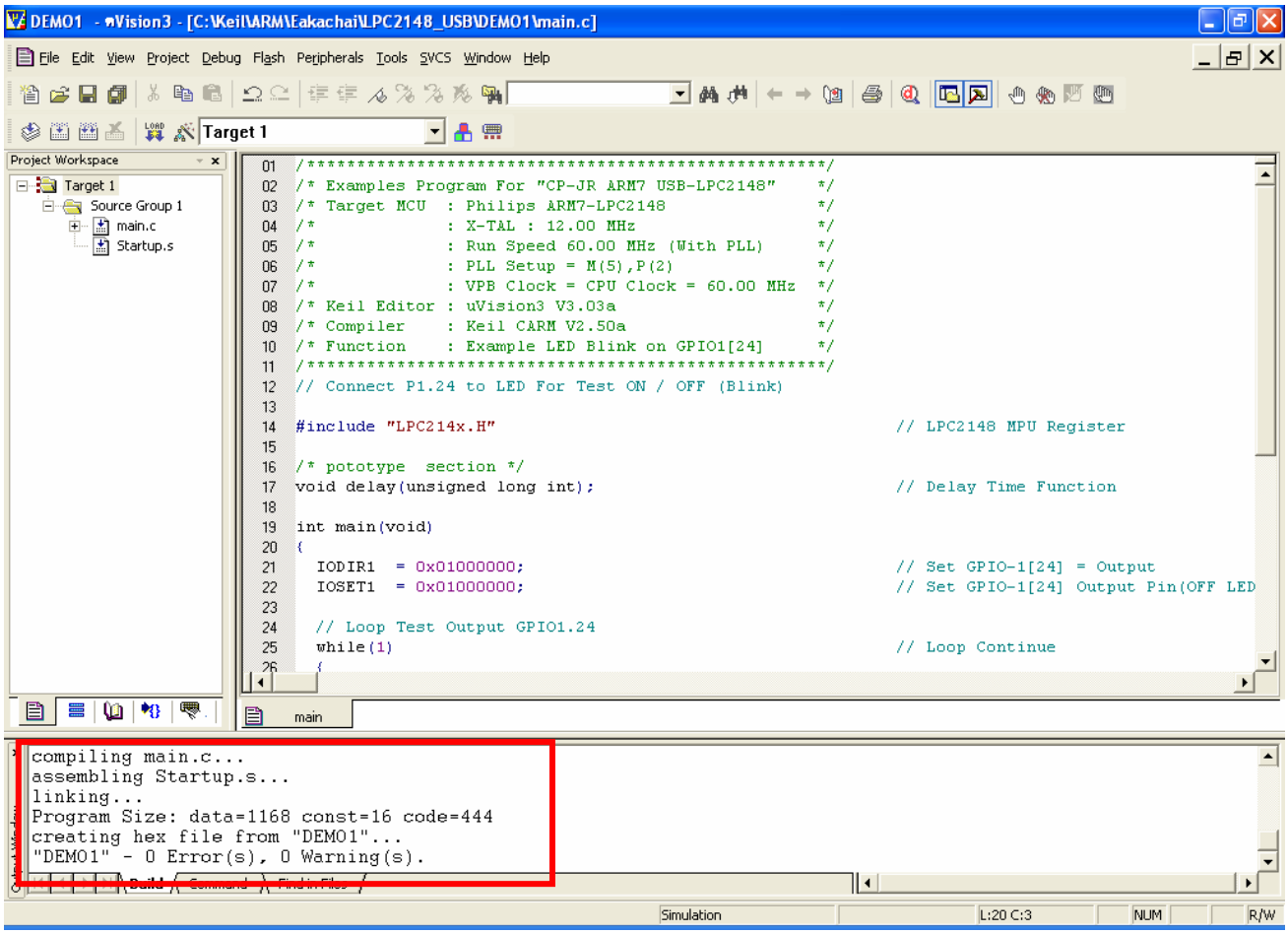

Add Files to Group 'Source Group 1'
Look in:
DEMO1
Startup
File name:
Startup
Files of type:
Asm Source file (*.s*; *.src; *.a*)
Add
Close

↓


Components, Environment and Books
Project Components
Folders/Extensions
Books
Project Targets:
Groups:
Files:
Target 1
Source Group 1
main.c
Startup.s
Set as Current Target
Add Files
OK
Cancel
Defaults
Help

After command **Add File** both **"main.c"** and **"Startup.s"** into Project File successfully, we can see both Files name are displayed in the blank of Tab of File.

8. Command to interpret the written program, click command **Projects → Rebuild all target files** and program Keil uVision3 will command program Keil-CARM to interpret commands instantly.



After we command to interpret program successfully and everything is correct without any error (0 Error and 0 Warning), we will get Hex File that has names to be the same as the created Project File name and we can use this Hex File to download into MCU instantly.

An advice to Initial MCU before operation of main program is started

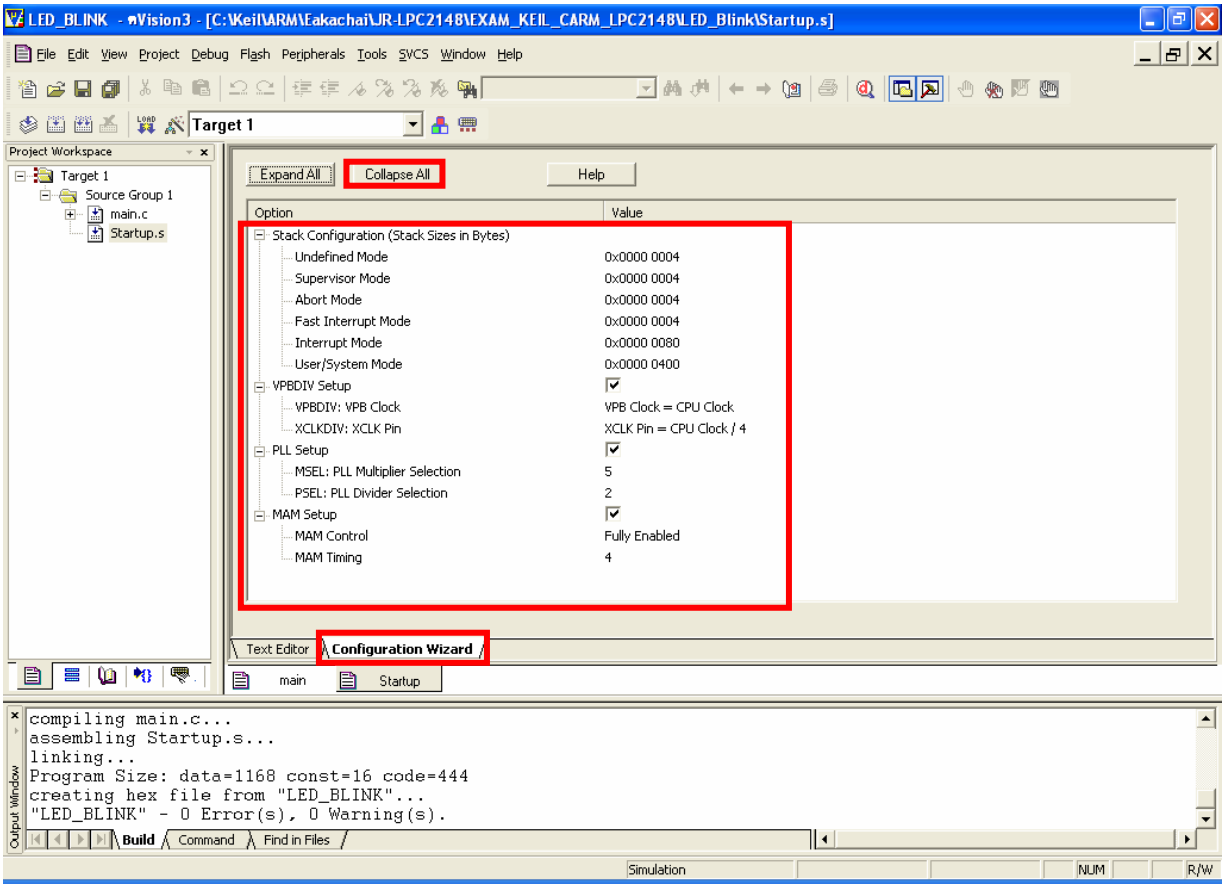
If we want operation of MCU is the most efficient both high speed to collect data of command and operations, we should configure default values into MCU as follows;

- Should configure **PLL** value to be **Processor Clock (cclk) = 60 MHz**, in case of using XTAL value to be 12 MHz, must configure value of M(Multiply) = 5 and P(Divide) = 2 and FFCO = 240 MHz
- Should configure **VPB Clock (pclk)**value to equal **cclk** or **60 MHz**
- Should configure **MAM Timing** value to be **4 Cycle of cclk (MAMTIM = 0x04)**
- Should configure **MAM Mode** value to be **Full Enable (MAMCR = 0x02)**

There are 2 methods to configure default values above; firstly, writing all Code commands in written program by self and secondly, copies file Startup that has already written, in this case, we can command to Add Startup File into our new created Project file. Additionally, there are 2 methods to check and modify file Startup value; firstly, can modify Code command in file as desired and secondly, configure Startup value from window program of Keil uVision3 by self. In this case, we recommend modifying value from Keil uVision3 because it is quit convenient.

To check Startup File value

Functions of Startup File are containing commands of program for starting operation of MCU before running follow by our written program. Function of Program in Startup file is Initial operation of MCU in the important port first and then jump to run follow by the command in our written C Language Program. If we want to check the Startup value, click Tab of File Startup and then select **"Expand All"** and we can see the default values that are configured in Startup file as in the picture below.



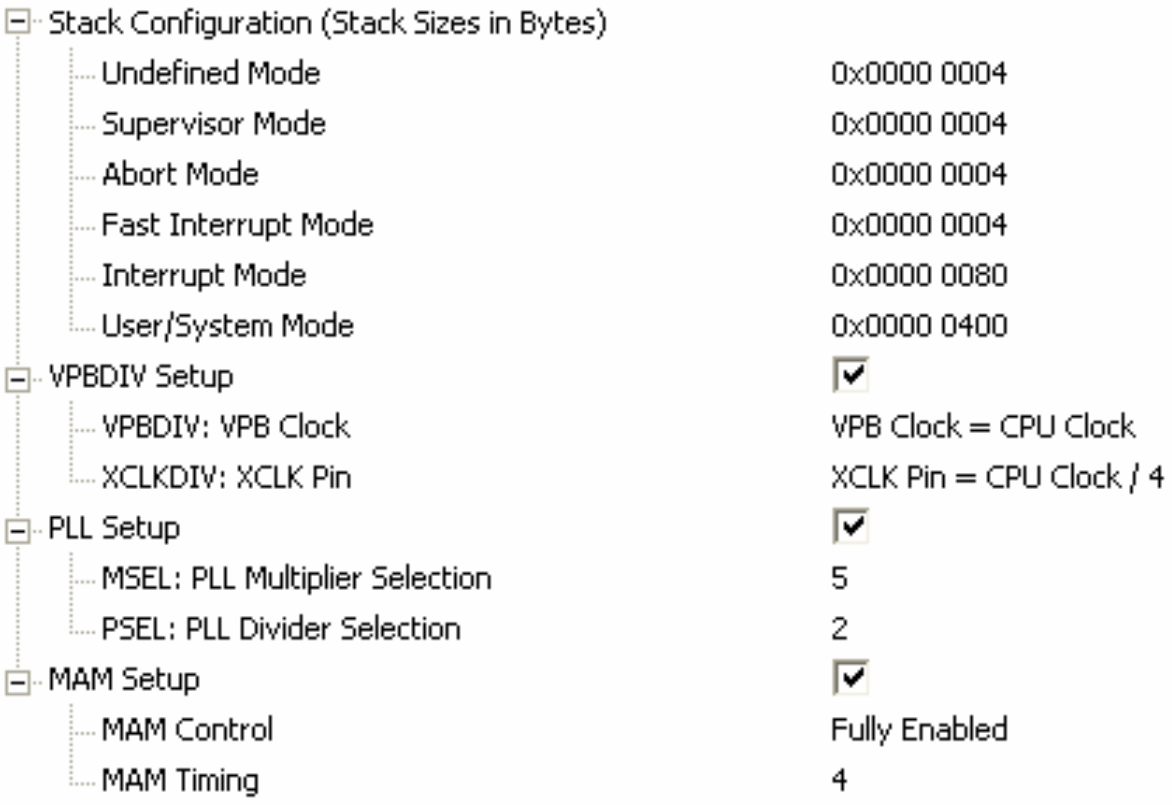


Figure Show the method to configure Startup File for LPC2148.

An example of C Code of Keil-CARM for Initial operation of LPC2148

If we want to write program Initial operation MCU by self, we only add code command into the starting point of main Program as in the picture below.

```
// Initial PLL & VPB Clock For CP-JR ARM7 USB-LPC2148
// Start of Initial PLL for Generate Processor Clock
// PLL Configuration Setup
// X-TAL = 12 MHz
// M(Multiply) = 5
// P(Divide) = 2
// Processor Clock(cclk) = M x OSC
//                       = 5 x 12 MHz
//                       = 60 MHz
// FCCO = cclk x 2 x P
//      = 60 x 2 x P
//      = 240 MHz
// VPB Clock(pclk) = 60 MHz
// Start of Initial PLL for Generate Processor Clock
PLLCFG &= 0xE0;                     // Reset MSEL0:4
PLLCFG |= 0x05;                     // MSEL(PLL Multiply) = 5
PLLCFG &= 0x9F;                     // Reset PSEL0:1
PLLCFG |= 0x20;                     // PSEL(PLL Devide) = 2

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

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

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

VPBDIV &= 0xFC;                     // Reset VPBDIV
VPBDIV |= 0x01;                     // VPB Clock(pclk) = cclk
// End of Initial PLL for Generate Processor Clock

// Start of Initial MAM Function
MAMCR = 0x00;                       // Disable MAM Function
MAMTIM = 0x04;                      // MAM Timing = 4 Cycle of cclk
MAMCR = 0x02;                       // Enable MAM = Full Function
// End of Initial MAM Function

// Start of Main Function Here
.
```

Figure Show the sample Code for initial operation of LPC2148