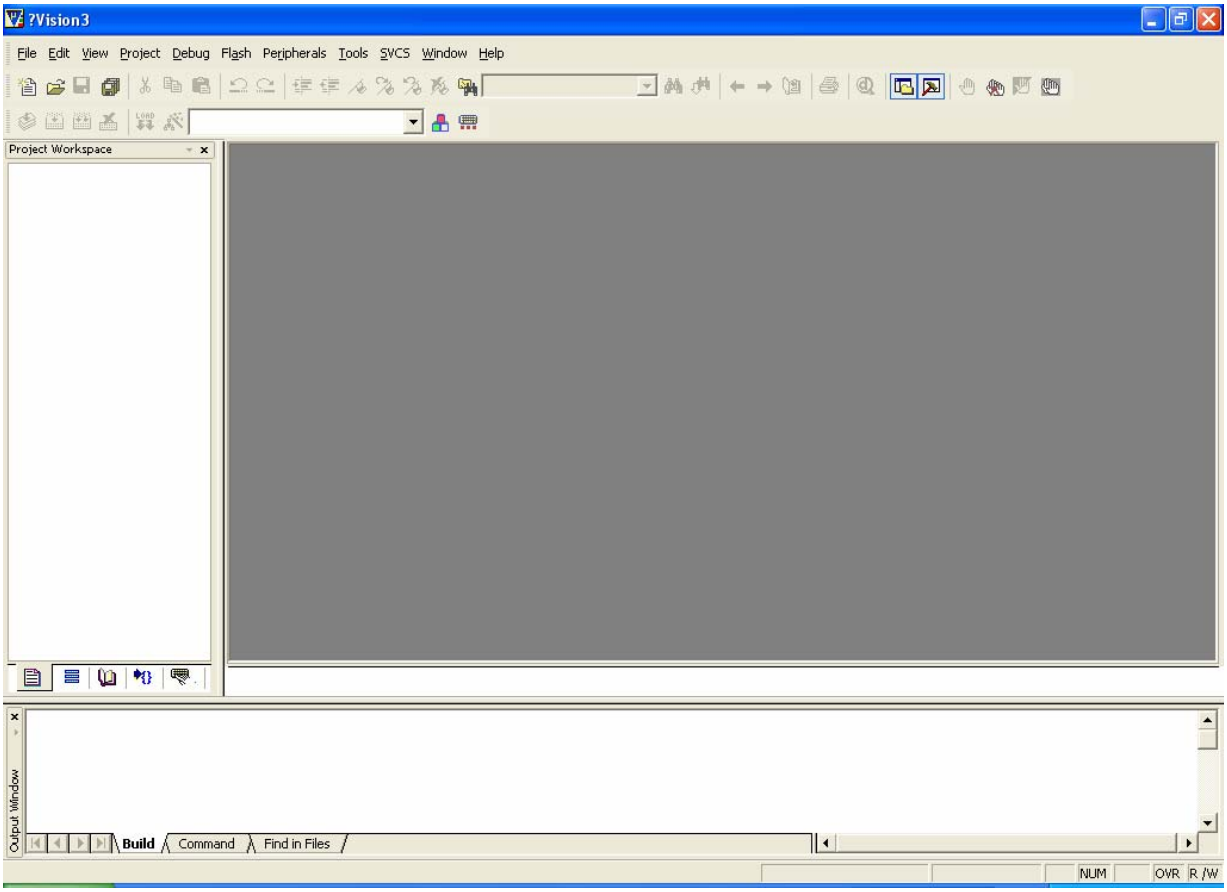
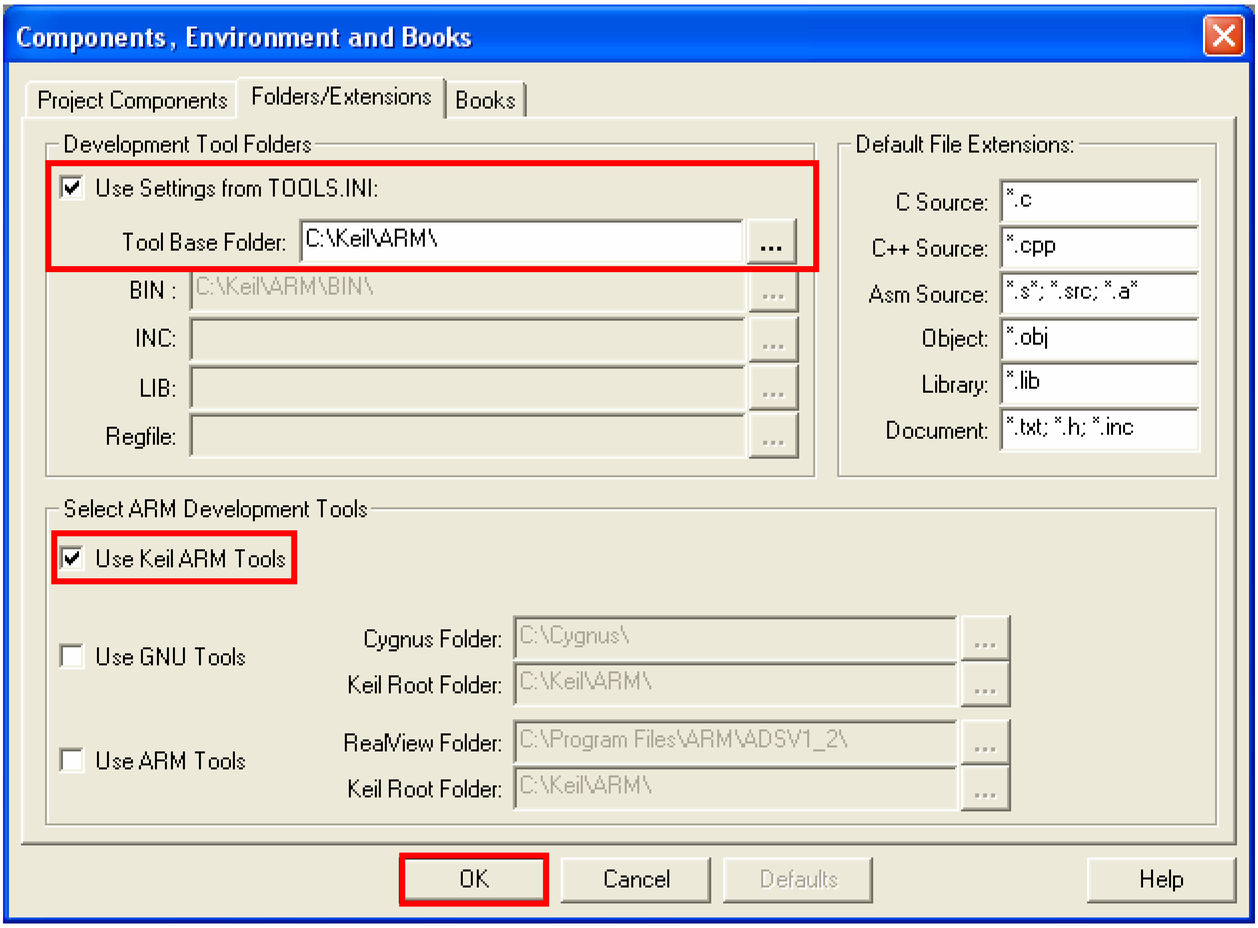
## ตัวอย่างการใช้ Keil uVision3 ในการสร้าง Project File ของ Keil ARM

ในที่นี้จะขอแสดงแนวทางการเขียนโปรแกรมภาษาซี โดยใช้ Keil ARM ในการแปลคำสั่ง ภายใต้โปรแกรม Text Editor ของ Keil (Keil uVision3) โดยจะขออธิบายถึงเฉพาะวิธีการกำหนดค่า Option สำหรับเชื่อมโยงคำสั่งในการสั่งแปลโปรแกรมด้วย Keil ARM ผ่านทาง Keil uVision3 เท่านั้น ส่วนรายละเอียดคำสั่งและการใช้งานฟังก์ชั่นต่างๆในการเขียนโปรแกรมของ Keil ARM นั้นขอให้ผู้ใช้ศึกษาจากคู่มือคำสั่งของ Keil RAM เอง โดยวิธีการกำหนดค่าตัวเลือกของ Keil uVision3 ให้ใช้งานกับ Keil ARM นั้นมีขั้นตอนพอสังเขปดังนี้คือ

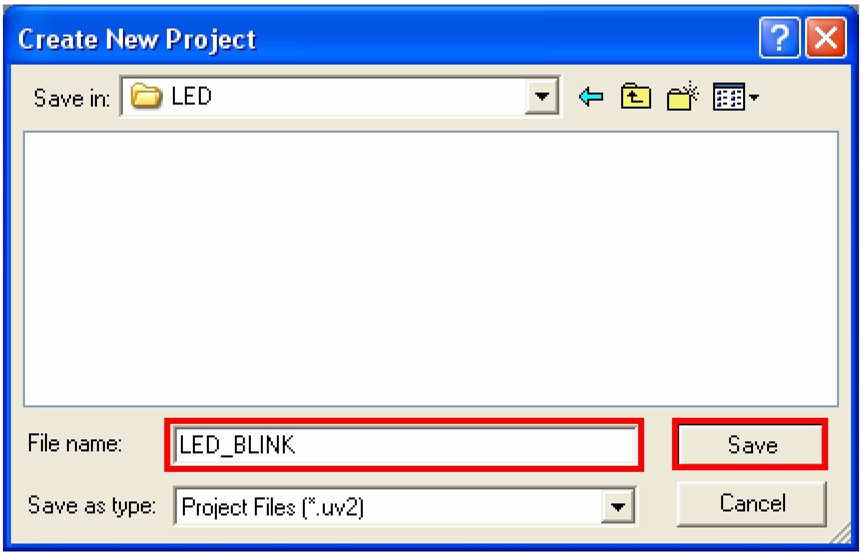
1. เปิดโปรแกรม Keil uVision3 ซึ่งเป็นโปรแกรม Text Editor ของ Keil-ARM ใช้สำหรับใช้ในการเขียนโปรแกรมที่เป็น Source Code ภาษาซี โดยจะมีลักษณะดังรูป



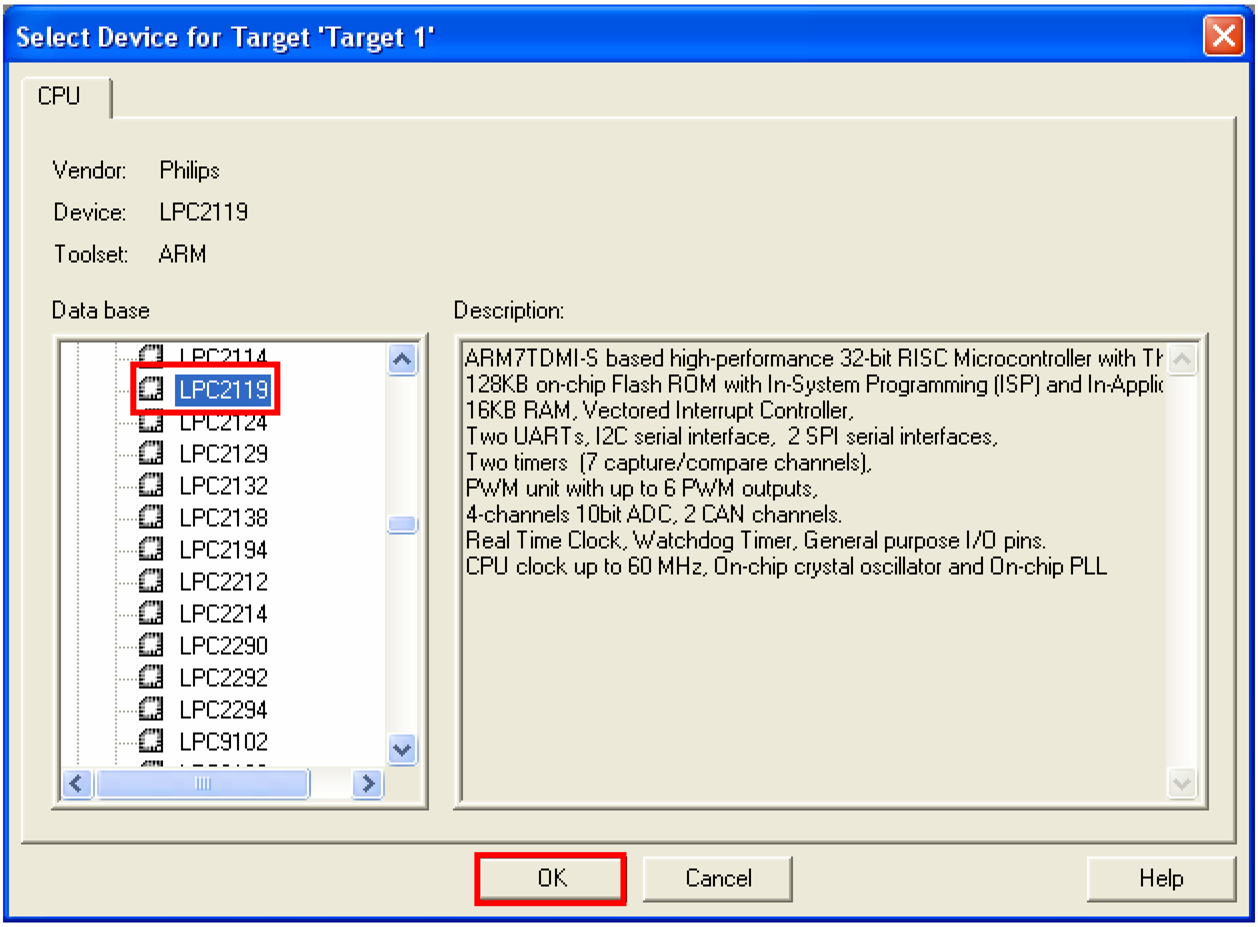

2. ทำการกำหนดค่าตัวเลือกในการแปลคำสั่งของ uVision3 ให้ใช้งานกับโปรแกรม Keil uVision3 และ Keil ARM โดยให้เลือกคลิกเมาส์ที่เมนูคำสั่ง Project ⟶ Components ,Environment, Books… จากนั้นให้เลือกค่าตัวเลือกสำหรับกำหนดการใช้งาน Complier จากหัวข้อ Select ARM Development Tools ซึ่งจะมีค่าตัวเลือกอยู่ 3 แบบ คือ Use Keil ARM Tools ,Use GNU Tools และ Use ARM Tools โดยให้เลือกเป็น “Use Keil ARM Tools” จากนั้นให้ทำการกำหนดตำแหน่ง Folder สำหรับเก็บค่าตัวเลือกการทำงานของโปรแกรม Keil ARM ซึ่งตามปรกติแล้วจะเป็น “C:\Keil\ARM\” แต่ถ้าติดตั้ง Keil ไว้ที่อื่นก็ต้องปรับเปลี่ยนให้ถูกต้องตามความเป็นจริงด้วยดังรูป




3. ทำการสร้าง Project File ขึ้นมาใหม่ โดยเรียกเมนูคำสั่ง Project ⟶ New Project จากนั้นให้เลือกกำหนดหรือสร้างตำแหน่ง Folder ที่จะบันทึก Project File พร้อมกับกำหนดชื่อ Project File ตามต้องการ เช่น ถ้าต้องการสร้าง Project File ชื่อ LED_BLINK โดยเก็บไว้ใน Folder ชื่อ LED ก็สามารถกำหนดตำแหน่ง Folder และชื่อ Project File ได้เอง โดยเมื่อกำหนดชื่อในช่อง File name แล้วให้เลือก Save เพื่อบันทึก Project File ไว้ ดังรูป

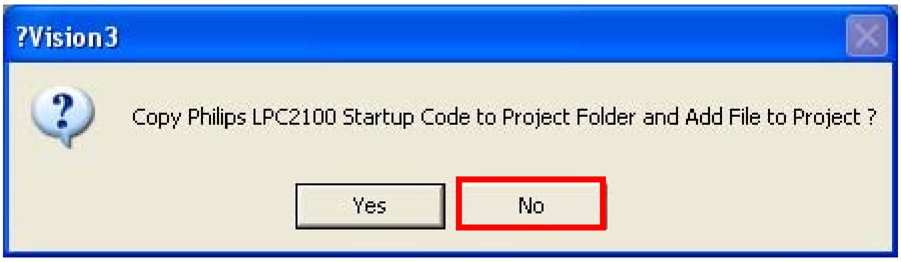


หลังจากกำหนดชื่อและสั่ง Save Project File แล้ว โปรแกรมจะรอให้ผู้ใช้ทำการกำหนด เบอร์ MCU ที่จะใช้งานใน Project ที่สั่ง Save นั้น ซึ่งในกรณีที่ใช้งานกับบอร์ด ET-ARM STAMP LPC2119 นั้น ให้เลือกกำหนดเบอร์ของ MCU เป็น LPC2119 ของ Philips แล้วเลือก OK ดังรูป




หลังจากเลือกกำหนดเบอร์ของ MCU เป็นที่เรียบร้อยแล้ว ในขั้นตอนนี้โปรแกรมจะรอให้ผู้ใช้ยืนยันว่าต้องการจะทำการ Copy ไฟล์ Startup ของ Keil เพื่อใช้งานกับ MCU ของ Philips มาใช้ใน Project ด้วยหรือไม่ โดย Startup ไฟล์จะเป็นส่วนของการกำหนดค่าเริ่มต้นการทำงานให้กับ MCU เช่น การกำหนดค่า Stack และการกำหนดค่าการทำงานให้กับ Phase-Lock-Loop ต่างๆ ก่อนที่จะเริ่มต้นทำงานตามโปรแกรมที่เราเขียนขึ้น ไม่เช่นนั้นแล้วโปรแกรมที่เราเขียนขึ้นมานั้นจะต้องเพิ่มคำสั่งในการเตรียมการทำงานส่วนเหล่านี้ให้ MCU เองทั้งหมด

แต่เนื่องจากไฟล์ Startup ของ Keil-ARM นั้น เป็นไฟล์ภาษาแอสแซมบลี้ ซึ่งกำหนดค่าการทำงานไว้กับชุดพัฒนาของ Keil เอง ดังนั้นข้อกำหนดและการกำหนดค่าบางอย่างจะมีความแตกต่างกันอยู่กับค่าที่ต้องการสำหรับบอร์ด “ET-ARM STAMP LPC2119” ไม่สามารถใช้งานไฟล์ Startup ได้ทันที ต้องมีการแก้ไขค่าตัวเลือกใหม่ ดังนั้นก่อนที่จะใช้โปรแกรม Keil ARM ในการแปลคำสั่งให้นั้น ผู้ใช้จะต้องเข้าไปแก้ไขไฟล์ Startup ใหม่โดยต้องกำหนดรูปแบบให้ถูกต้องตรงกับความต้องการของบอร์ด “ET-ARM STAMP LPC2119” ด้วย ดังนั้นในที่นี้ขอแนะนำให้เลือก “No” เพื่อไม่ให้ Keil uVision3 ทำการ Copy ไฟล์ Startup ของ Keil-ARM มาใช้ใน Project นี้ด้วย

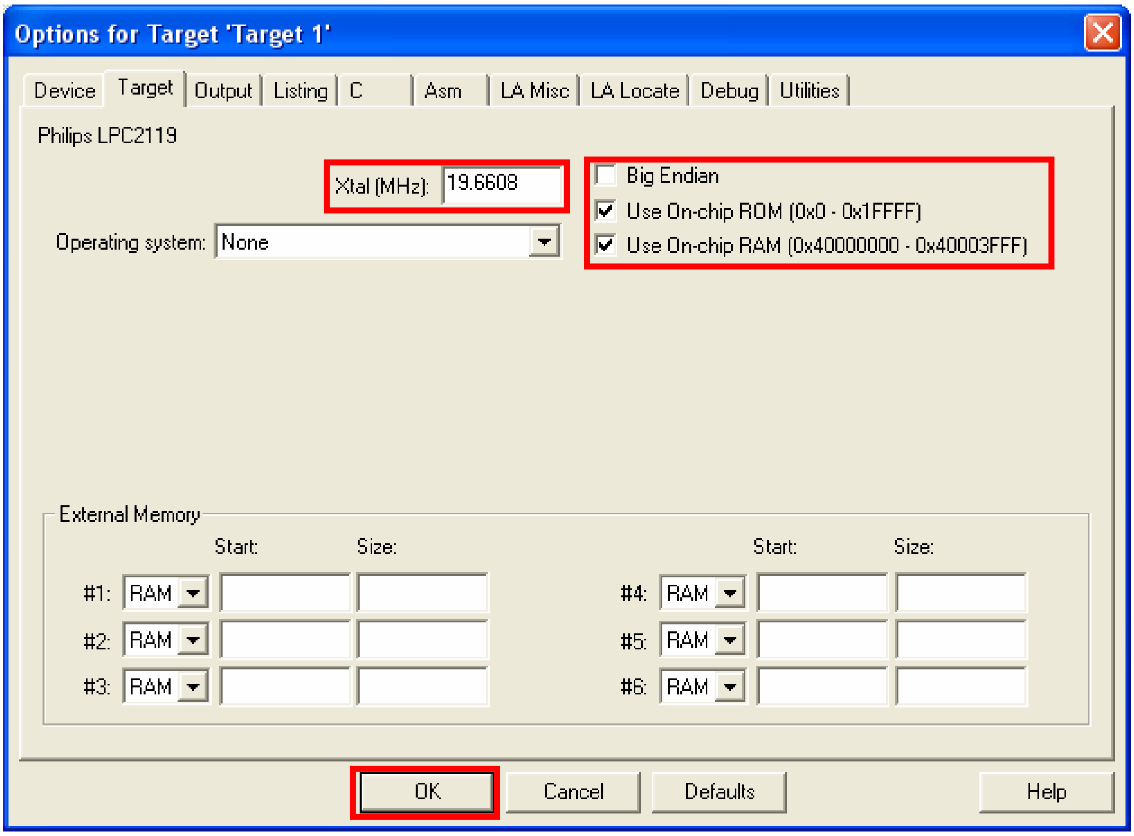



4. ให้ทำการ Copy File ชื่อ “Startup.s” ที่ทาง อีทีที จัดเตรียมไว้ใน CD-ROM ซึ่งเก็บไว้ใน Folder ชื่อ “SOURCE_KEIL” มาไว้ในตำแหน่ง Folder เดียวกันกับ Project File ที่สร้างขึ้นมาใหม่นี้

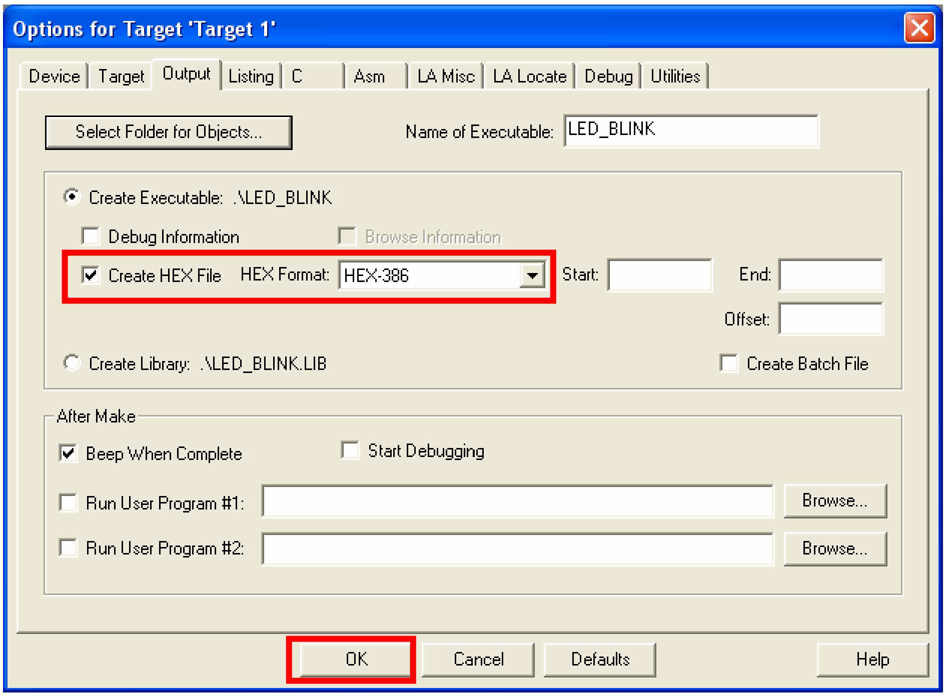
โดยไฟล์ “Startup.s” จะเป็นไฟล์ซึ่งบรรจุคำสั่งภาษาแอสแซมบลี้ของ ARM7 สำหรับทำหน้าที่กำหนดค่าเริ่มต้นการทำงานที่จำเป็นให้กับ MCU ซึ่งได้แก่การ กำหนดค่า Stack ให้กับ MCU การ Initial Phase-Lock-Loop การกำหนดค่าให้กับ MAM Function และการกำหนดตำแหน่ง Vector ต่างๆของ MCU สำหรับใช้งานร่วมกับบอร์ด “ET-ARM STAMP LPC2119” ซึ่งถ้าสั่ง Add ไฟล์ “Startup.s” จาก Keil หรือ Copy ไฟล์ดังกล่าวมาจากแหล่งอื่นๆ อาจมีการทำงานของโปรแกรมใน Startup ไม่เหมือนกัน ซึ่งจะส่งผลต่อการทำงานของโปรแกรมที่จะเขียนขึ้นด้วย

5. ให้ทำการกำหนดค่า Option ของ Project File โดยเลือกเมนูคำสั่ง Project ⟶ Option for Target ’Target 1’ จากนั้นเลือกที่ Tab ของ Target เพื่อกำหนดค่าของ MCU Target โดยให้กำหนดดังนี้

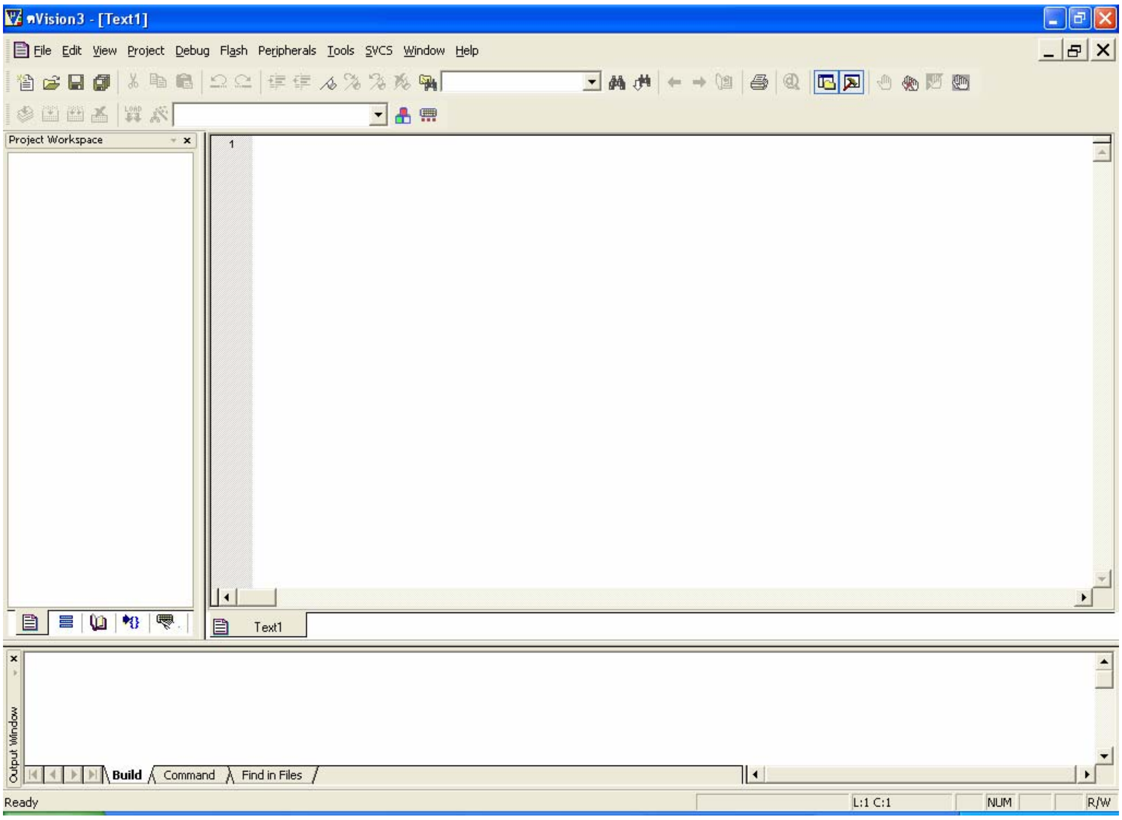
5.1 X-TAL ให้กำหนดเป็น 19.6608 MHz พร้อมกับเลือกกำหนดให้ใช้หน่วยความจำที่มีอยู่ใน MCU เป็นเงื่อนไขในการแปลโปรแกรมของ Keil ARM ด้วย ดังรูป



5.2 Output ให้เลือกคลิกเมาส์ที่ค่าตัวเลือก Create HEX File พร้อมกับเลือกกำหนดรูปแบบของ Hex ให้เป็นแบบ HEX-386 แล้วเลือก OK ดังรูป



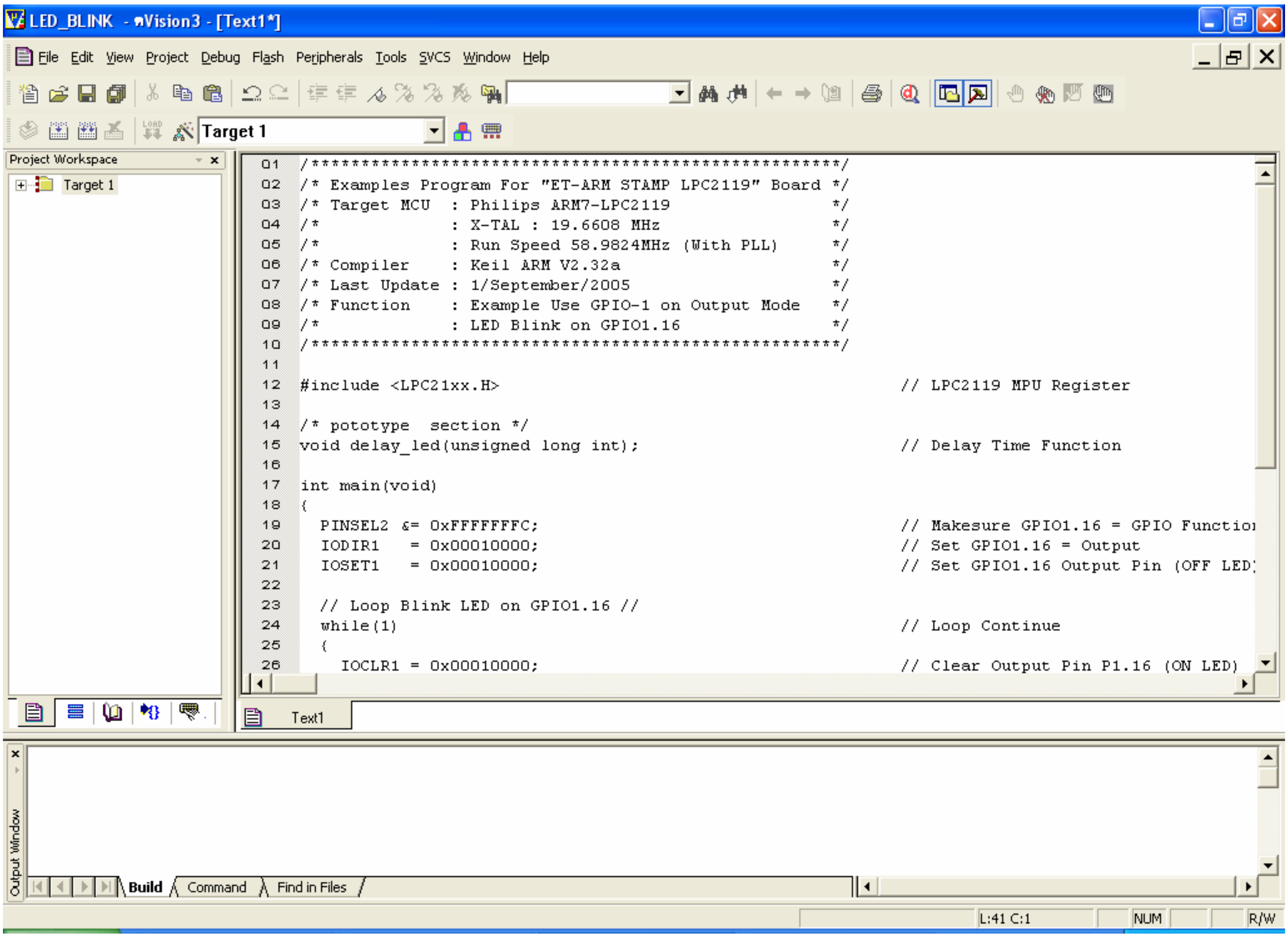

6. เริ่มต้นเขียน Source Code ภาษาซี โดยให้เลือกคลิกเมาส์ที่เมนูคำสั่ง File ⟶ New… ซึ่งจะได้พื้นที่ในการเขียน Text File เกิดขึ้นมา โดยในครั้งแรกจะกำหนดชื่อตามค่า Default เป็น “Text1” ดังรูป



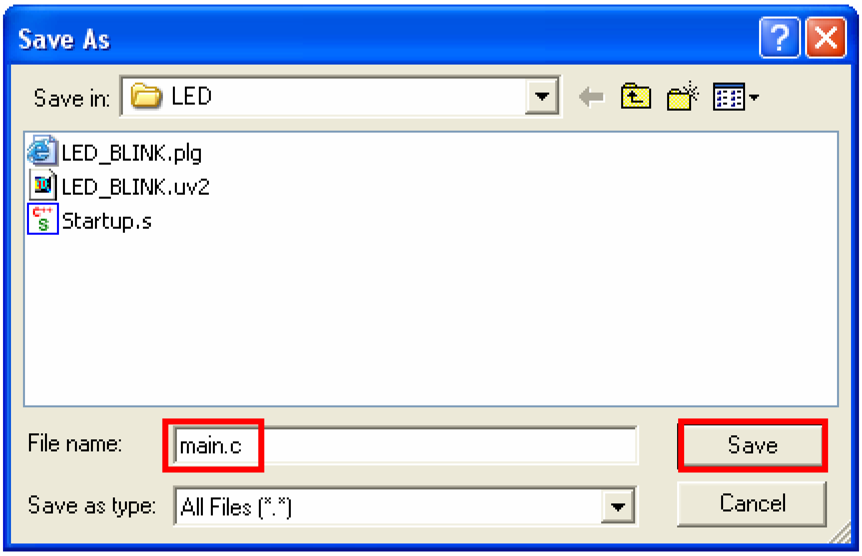
ในขั้นตอนนี้ให้ทำการพิมพ์ Source Code ภาษาซี ตามข้อกำหนดของ Keil ARM ในพื้นที่เขียนโปรแกรม ตามต้องต้องการดังตัวอย่าง

```
/*****************************************************/
/* 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 &= 0xFFFFFFFC;           // 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
}
```

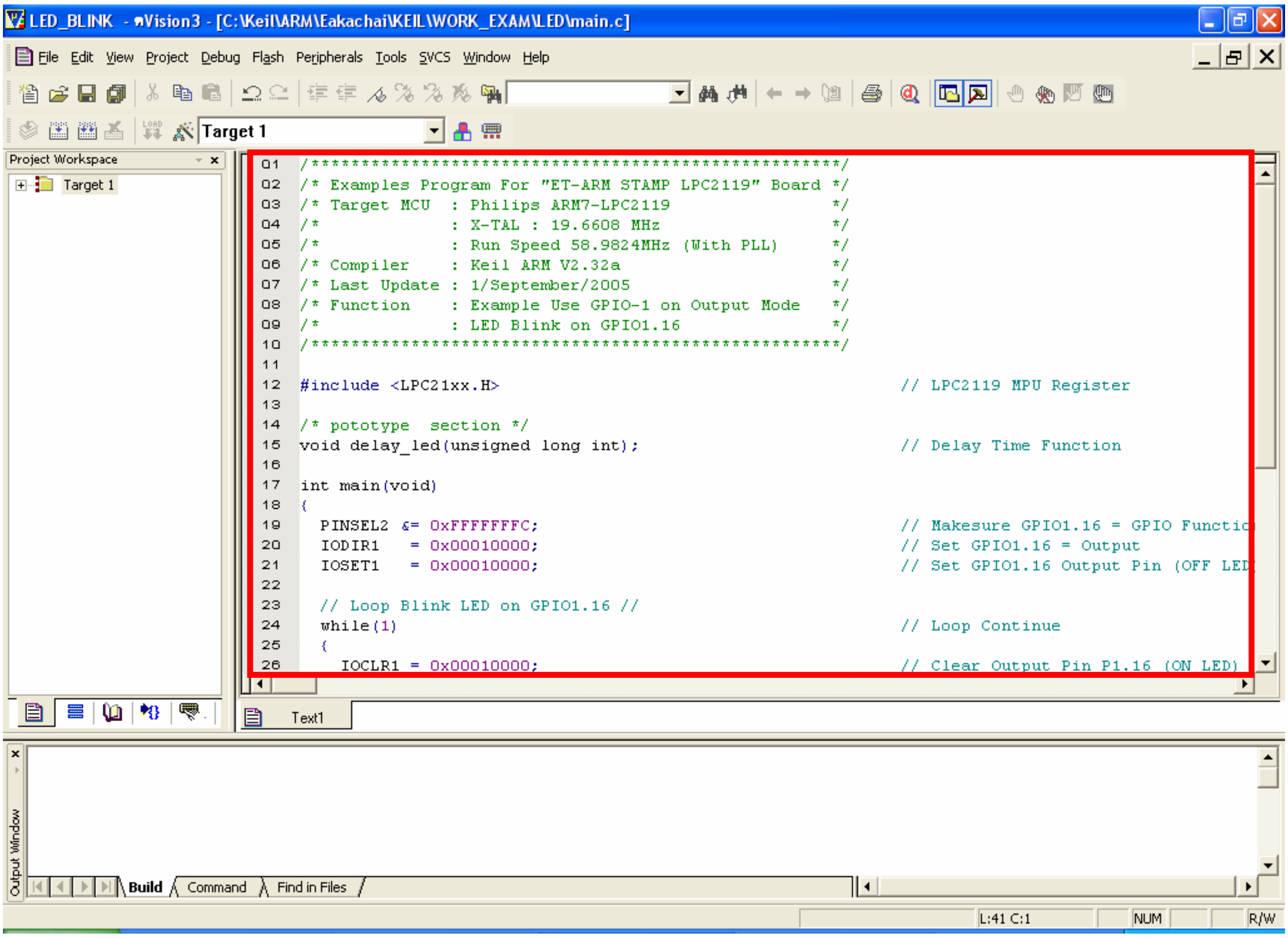
**ตัวอย่างโปรแกรมไฟกระพริบ 1 ดวง ที่** GPIO1.16



หลังจากพิมพ์คำสั่งภาษาซีเสร็จเรียบร้อยตามต้องการแล้ว ให้สั่ง Save ไฟล์ดังกล่าว โดยต้องกำหนดเป็นไฟล์ที่มีนามสกุลเป็น ".C" ในที่นี้ขอแนะนำให้สั่ง Save โดยใช้คำสั่ง File ⟶ Save As... แล้วกำหนดชื่อและนามสกุลของไฟล์เป็น .main.c" ดังรูป



ซึ่งหลังจากที่สั่ง Save ไฟล์เป็น “main.c” แล้วจะเห็นว่าลักษณะสีของตัวอักขระต่างๆในโปรแกรมจะเกิดการเปลี่ยนแปลงไปตามหน้าที่ เช่น Comment, ตัวแปร และ คำสั่ง เป็นต้น ซึ่งส่วนนี้เป็นข้อดีของ Keil uVision3 ซึ่งสามารถแยกและแสดงตัวอักขระได้อย่างเป็นหมวดหมู่ ทำให้ง่ายต่อการอ่านโปรแกรมด้วย ดังรูป

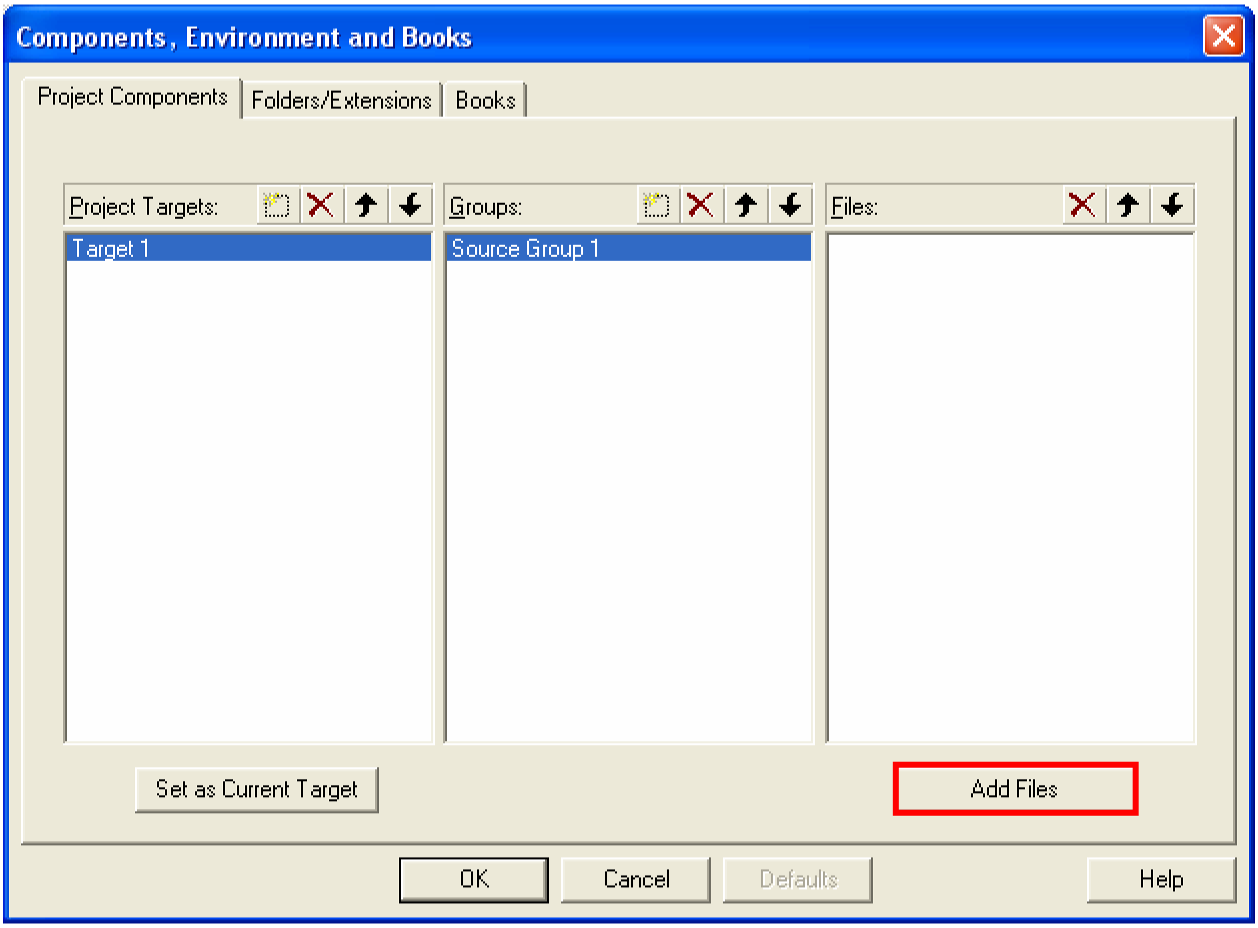



7. ทำการสั่ง Add File ต่างๆเข้ากับ Project File โดยให้เลือกคลิกเมาส์ที่คำสั่ง Project ⟶ Components, Environment, Books… จากนั้นให้เลือกที่ Tab Project Components แล้วเลือกที่ Add File ที่ต้องการจะเพิ่มเข้าไปใช้งานร่วมกับ Project File

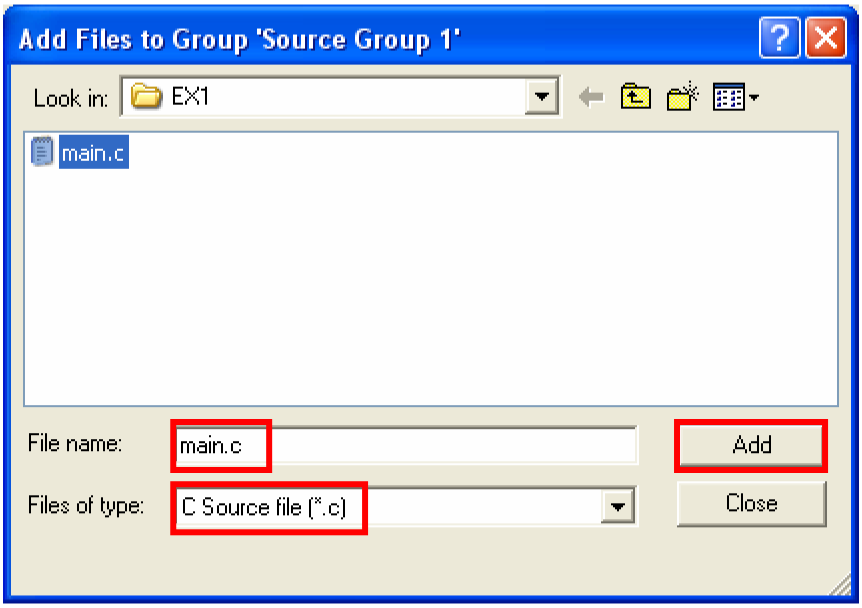
โดยในครั้งแรกให้เลือก Files of type เป็น “C Source files(*.c)” ซึ่งจะปรากฏชื่อไฟล์ต่างๆที่เป็น Source Code ภาษาซีให้เห็น โดยในที่นี้ให้เลือกคลิกเมาส์ที่ไอคอนของไฟล์ชื่อ “main.c” แล้วเลือก Add เพื่อสั่งเพิ่มไฟล์ชื่อ “Startup.s” เข้าไปรวมกับ Project Files ที่เราสร้างไว้

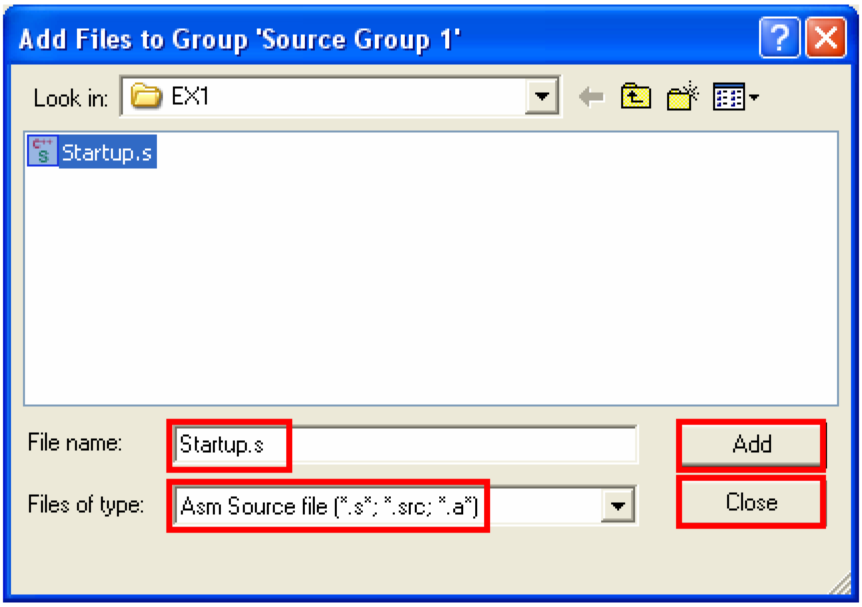
จากนั้นให้เลือกกำหนด File of type ใหม่เป็น “ASM Source files(*.s*;*.src;*.a*) ซึ่งจะปรากฏชื่อของไฟล์ Startup.s ให้เห็นในช่องแสดงชื่อไฟล์ ให้ทำการคลิกเมาส์ที่ไอคอนของไฟล์ “Startup.s” แล้วเลือก Add เพื่อสั่งเพิ่มไฟล์ชื่อ “Startup.s” เข้าไปรวมกับ Project Files ที่เราสร้างไว้

เมื่อทำการสั่ง Add ไฟล์ชื่อ “main.c’ และ “Startup.s” ให้กับ Project File เป็นที่เรียบร้อยแล้วให้เลือกที่ Close เพื่อเป็นการสิ้นสุดการสั่ง Add File ซึ่งจะได้ดังรูป

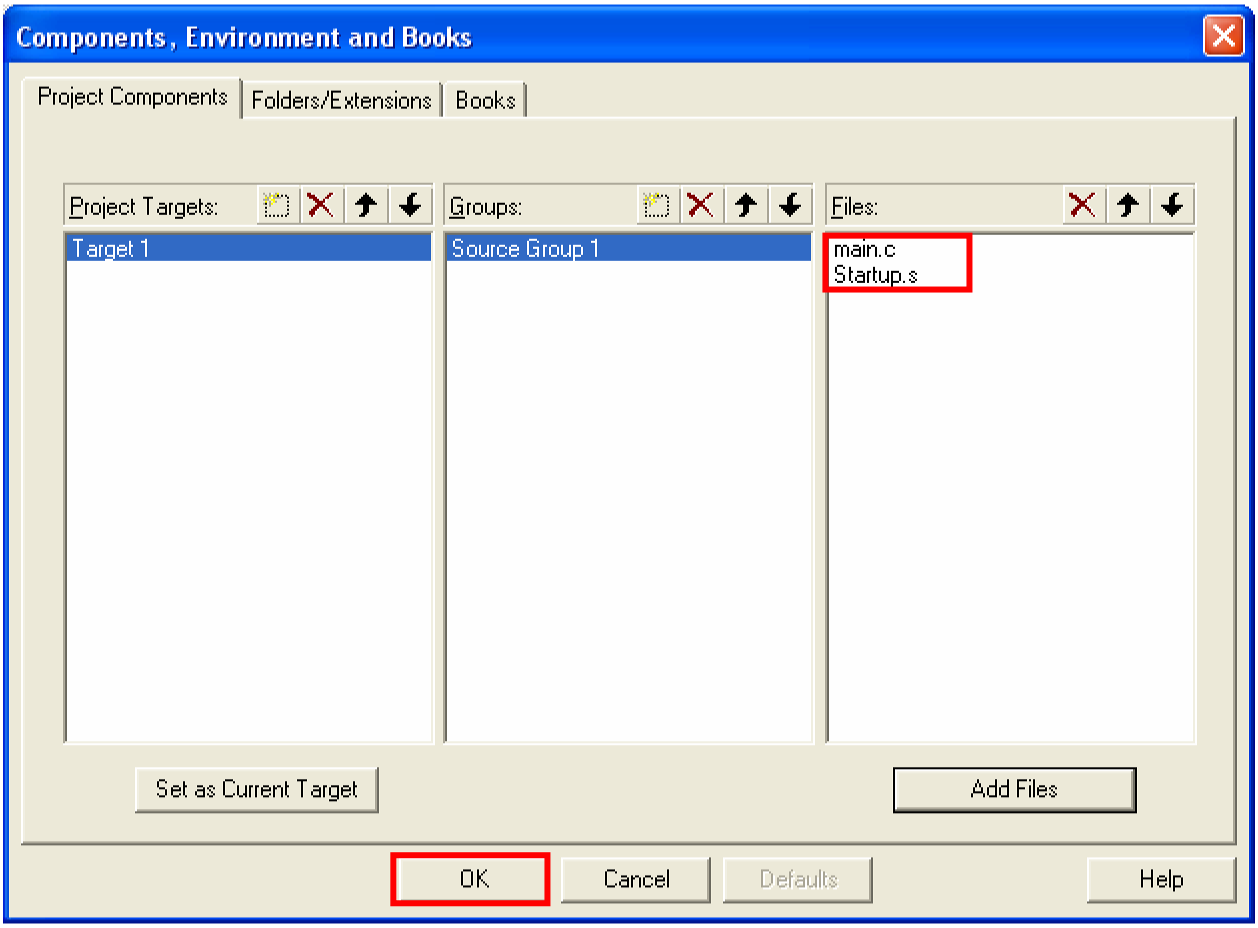



↓



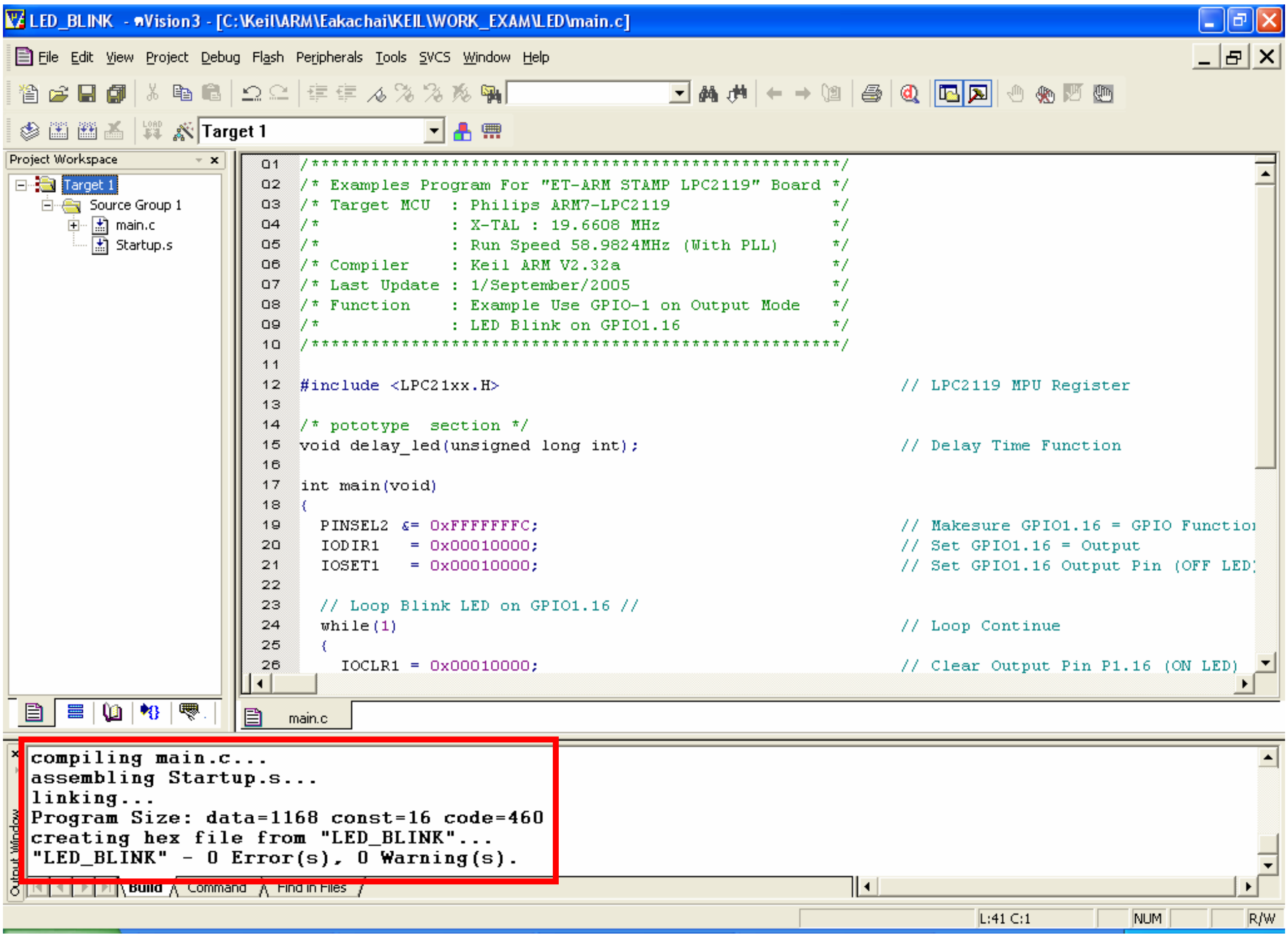



↓




ซึ่งหลังจากทำการสั่ง Add File ทั้ง “main.c” และ “Startup.s” ให้กับ Project File เป็นที่เรียบร้อยแล้ว จะสังเกตเห็นที่ช่อง Tab ของ File ปรากฏชื่อของ File ทั้ง 2 ให้เห็นด้วย

8. ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้นเรียบร้อยแล้ว โดยให้คลิกเมาส์ที่เมนูคำสั่ง Projects ⟶ Rebuild all target files ซึ่งโปรแกรม Keil uVision3 จะทำการสั่งให้โปรแกรม Keil ARM ทำการแปลคำสั่งให้ทันที




ซึ่งหลังจากสั่งแปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใดๆขึ้น (0 Error และ 0 Warning) ก็ได้ Hex File ซึ่งมีชื่อเหมือนกันกับชื่อของ Project File ที่สร้างไว้ ซึ่งผู้ใช้สามารถนำ Hex File ดังกล่าวไปทำการ Download ให้กับ MCU ได้ทันที

## ข้อแนะนำในการ Initial MCU ก่อนเริ่มต้นการทำงานของโปรแกรมหลัก

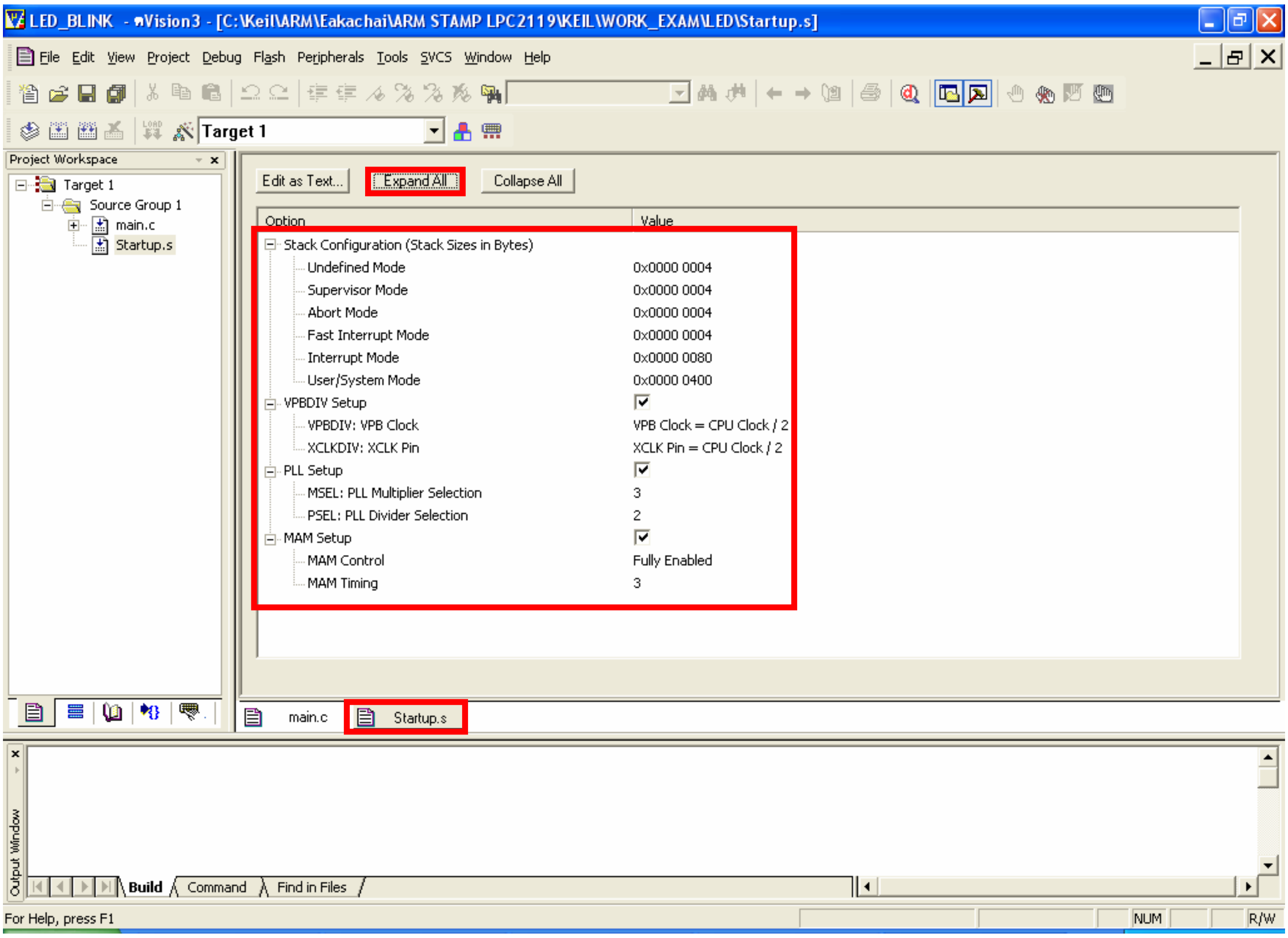
เพื่อให้การทำงานของ MCU มีประสิทธิภาพสูงสุด ทั้งในด้านของความเร็วในการประมวลผลคำสั่ง และการทำงานควรกำหนดค่าต่างๆให้กับ MCU ดังต่อไปนี้

- ควรกำหนดค่า PLL ให้ได้ค่า Processor Clock (cclk) = 58.9824MHz โดยในกรณีที่ใช้ค่า XTAL เป็น 19.6608MHz ต้องให้ค่า M(Multiply)=3 และ P(Divide)=2 และ FFCO =235.9296MHz
- ควรกำหนดค่า VPB Clock (pclk) ให้มีค่า เป็นครึ่งหนึ่งของ cclk หรือ 29.4912MHz
- ควรกำหนดค่า MAM Timing ให้มีค่า 3 Cycle ของ cclk (MAMTIM = 0x03)
- ควรกำหนดค่า MAM Mode เป็น Full Enable (MAMCR = 0x02)

โดยการกำหนดค่าต่างๆดังกล่าวข้างต้นนั้น สามารถทำได้ 2 วิธีคือ การเขียน Code คำสั่งในโปรแกรมที่เราเขียนขึ้นเองทั้งหมด หรือใช้การ Copy ไฟล์ที่เป็น Startup File ที่สร้างไว้เรียบร้อยแล้วมาใช้งาน จากนั้นจึงสั่ง Add Startup File เข้ามาใช้งานใน Project ที่เราสร้างขึ้น ซึ่งการตรวจสอบและแก้ไขค่าของ Startup File นั้น สามารถทำได้ 2 แบบ คือ เข้าไปแก้ไข Code คำสั่งในไฟล์ตามความต้องการ หรือ ใช้การกำหนดค่า Startup File จากหน้าต่างโปรแกรมของ Keil uVision3 เอง ซึ่งเพื่อให้เกิดความสะดวกต่อการใช้งานขอแนะนำให้ใช้วิธีการแก้ไขเปลี่ยนแปลงค่าจาก Keil uVision3 จะดีที่สุด

## การตรวจสอบค่า Startup File

หน้าที่ของ Startup File คือ บรรจุคำสั่งของโปรแกรมเริ่มต้นการทำงานของ MCU ก่อนที่จะเริ่มต้นมาทำงานตามคำสั่งที่เราเขียนขึ้น โดยโปรแกรมที่อยู่ใน Startup ไฟล์จะทำหน้าที่ Initial การทำงานของ MCU ในส่วนที่จำเป็นก่อน จากนั้นจึงจะกระโดดไปทำงานตามคำสั่งในโปรแกรมภาษาซีที่เราเขียนขึ้น ซึ่งการเข้าไปตรวจสอบค่าของ Startup ไฟล์สามารถทำได้โดย ให้เลือกคลิกที่ Tab ของไฟล์ Startup แล้วเลือก “Expand All” ซึ่งจะเห็นค่าที่กำหนดไว้ใน Startup File ดังรูป



| | |
|---|---|
| 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 | ☑ |
| VPBDIV: VPB Clock | VPB Clock = CPU Clock / 2 |
| XCLKDIV: XCLK Pin | XCLK Pin = CPU Clock / 2 |
| PLL Setup | ☑ |
| MSEL: PLL Multiplier Selection | 3 |
| PSEL: PLL Divider Selection | 2 |
| MAM Setup | ☑ |
| MAM Control | Fully Enabled |
| MAM Timing | 3 |

แสดงการกำหนดค่า Startup File สำหรับ LPC2119

**ตัวอย่าง Code ภาษาซีของ Keil ARM สำหรับ Initial การทำงานของ LPC2119**

สำหรับในกรณีที่ต้องการเขียนโปรแกรม Initial การทำงานของ MCU เอง ก็สามารถทำได้ โดยการเพิ่ม Code คำสั่งเข้าไปในส่วนเริ่มต้นการทำงานของ Main Program ดังตัวอย่าง

```
// 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

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 |= 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
.
```

**แสดงตัวอย่าง Code สำหรับ Initial การทำงานของ LPC2119 ก่อนเริ่มต้นทำงานใน Main โปรแกรม**