การพัฒนาโปรแกรม ET-ARM STAMP LPC2138 ด้วย GCCARM ร่วมกับ Editor ของ Keil uVision3

ในการพัฒนาโปรแกรมของ ARM7 ด้วย GCCARM นั้น ตามปรกติแล้ว GCCARM จะเป็นเพียงโปรแกรม C-Complier อย่างเดียว ไม่มีส่วนของโปรแกรม Text Editor รวมอยู่ด้วย ดังนั้นในการสร้าง Source Code จะต้องใช้โปรแกรมจำพวก Text Editor ตัวอื่นๆในการสร้าง Source Code ที่เป็น Text File แล้วบันทึกเป็นไฟล์ภาษาซีไว้ จากนั้นจึงใช้ GCCARM ในการสั่งแปลคำสั่ง (Complier) อีกครั้งหนึ่ง โดยในการสั่ง Complier นั้นจะเป็นการสั่งงานแบบ Command Line โดยผู้ใช้จะต้องเข้าใจรูปแบบคำสั่งและ Option ต่างๆในการสั่ง Complier เป็นอย่างดีจึงจะสามารถสั่ง Complier ได้อย่างถูกต้องตามเงื่อนไขที่ต้องการ ซึ่งจะเห็นได้ว่าการใช้งานโปรแกรม GCCARM นั้นจะไม่ค่อยมีความสะดวกมากนัก เนื่องจากต้องสลับการทำงานของโปรแกรมจาก Text Editor ไปยัง Command Line เพื่อสั่งแปลโปรแกรม ซึ่งถ้าเกิดข้อผิดพลาดขึ้นก็จะต้องกลับเข้าไปแก้ไข Source Code นั้นใหม่ แล้วกลับไปสั่งแปลโปรแกรมใหม่อีก โดยอาจต้องวนเวียนอยู่เช่นนี้หลายๆรอบ จนกว่าโปรแกรมที่เขียนขึ้นนั้นจะเสร็จสมบูรณ์ แต่อย่างไรก็ดี GCCARM ก็มีข้อดี ที่ทุกคนปฏิเสธไม่ได้ คือ เป็นของฟรี ซึ่งสามารถ Download มาใช้ได้โดยไม่เสียค่าใช้จ่ายใดๆ ถึงแม้ว่าจะมีข้อจำกัดในเรื่องของความสะดวกสบายในการใช้งานอยู่บ้างก็ตาม

สำหรับในกรณีที่ผู้ใช้มีชุดโปรแกรม Keil-ARM รุ่น Demo สำหรับทดลองใช้งานอยู่แล้วนั้น สามารถ อาศัยโปแกรม Text Editor ของ Keil-ARM (Keil uVision3) สำหรับเขียน Source Code ภาษาซี แล้วกำหนดการเชื่อมโยง Command Line ให้ไปเรียกใช้โปรแกรม GCCARM ในการแปลคำสั่งแทนได้ ซึ่งจะทำให้ผู้ใช้งานสามารถทำการเขียนโปรแกรมภาษาซีและสั่งแปลโปรแกรมได้จากหน้าเมนูคำสั่งของโปรแกรม Keil uVision3 ได้เช่นเดียวกันการใช้ Keil-ARM ทุกประการ เพียงแต่ว่า ข้อกำหนดในการเขียนโปรแกรม Source Code ที่เป็นภาษาซีนั้น จะต้องยึดหลักการเขียนโปรแกรมตามรูปแบบและข้อกำหนดของ GCCARM แทนเท่านั้น ซึ่งข้อดีของการใช้ GCCARM ก็คือ ไม่มีข้อจำกัดในเรื่องขนาดของ Output Hex File เหมือนการใช้ Keil-ARM รุ่นทดลองใช้(Demo)

ข้อแนะนำในการใช้งานโปรแกรม

- โปรแกรม Keil-ARM รุ่น Demo สามารถ Download ได้จาก WWW.KEIL.COM โดยในการติดตั้งใช้งานนั้น เพื่อให้สะดวกต่อการกำหนดค่าตัวเลือกใช้งานต่างๆขอแนะนำให้ติดตั้งโปรแกรมใน Folder ชื่อ C:\keil ตามค่าตัวเลือกที่เป็นค่า Default ของโปรแกรมติดตั้งจะดีที่สุด

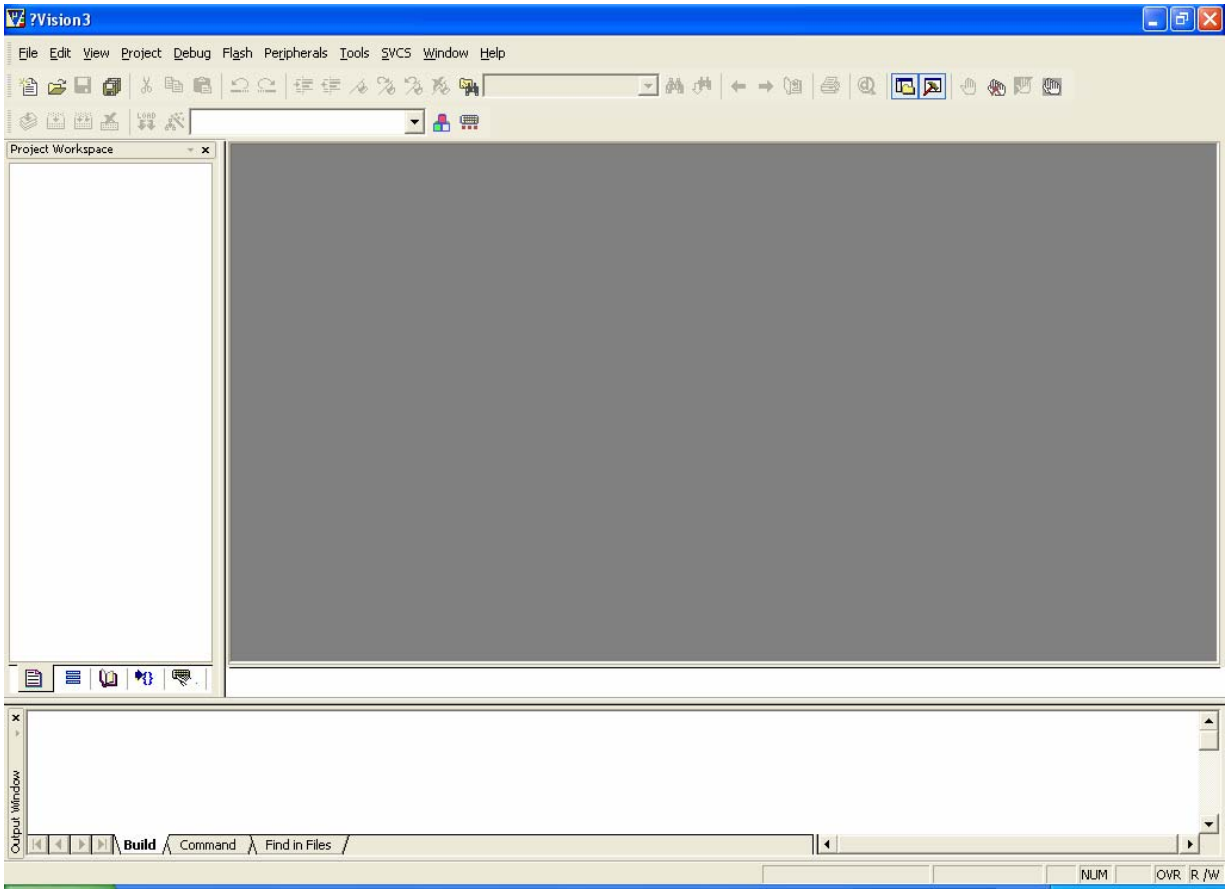
- โปรแกรม GCCARM สามารถ Download ได้จาก WWW.KEIL.COM เช่นเดียวกัน โดยในการติดตั้งใช้งานนั้นขอแนะนำให้ติดตั้งโปรแกรมไว้ใน Folder ชื่อ C:\Cygnus ตามค่าตัวเลือกที่เป็นค่า Default ของโปรแกรมติดตั้งจะดีที่สุด เพราะจะง่ายในการสั่งกำหนดค่า Option ต่างๆในการใช้งาน

เนื่องจากถ้าติดตั้งโปรแกรมใน Folder ที่แตกต่างไปจากนี้แล้ว ผู้ใช้ต้องทำการปรับแก้ตำแหน่ง Folder ในขั้นตอนของการกำหนดค่าตัวเลือกในโปรแกรมใหม่ตามที่ติดตั้งโปรแกรมไว้จริงด้วยจะไม่สามารถอ้างอิงตามคำแนะนำนี้ได้ทั้งหมด

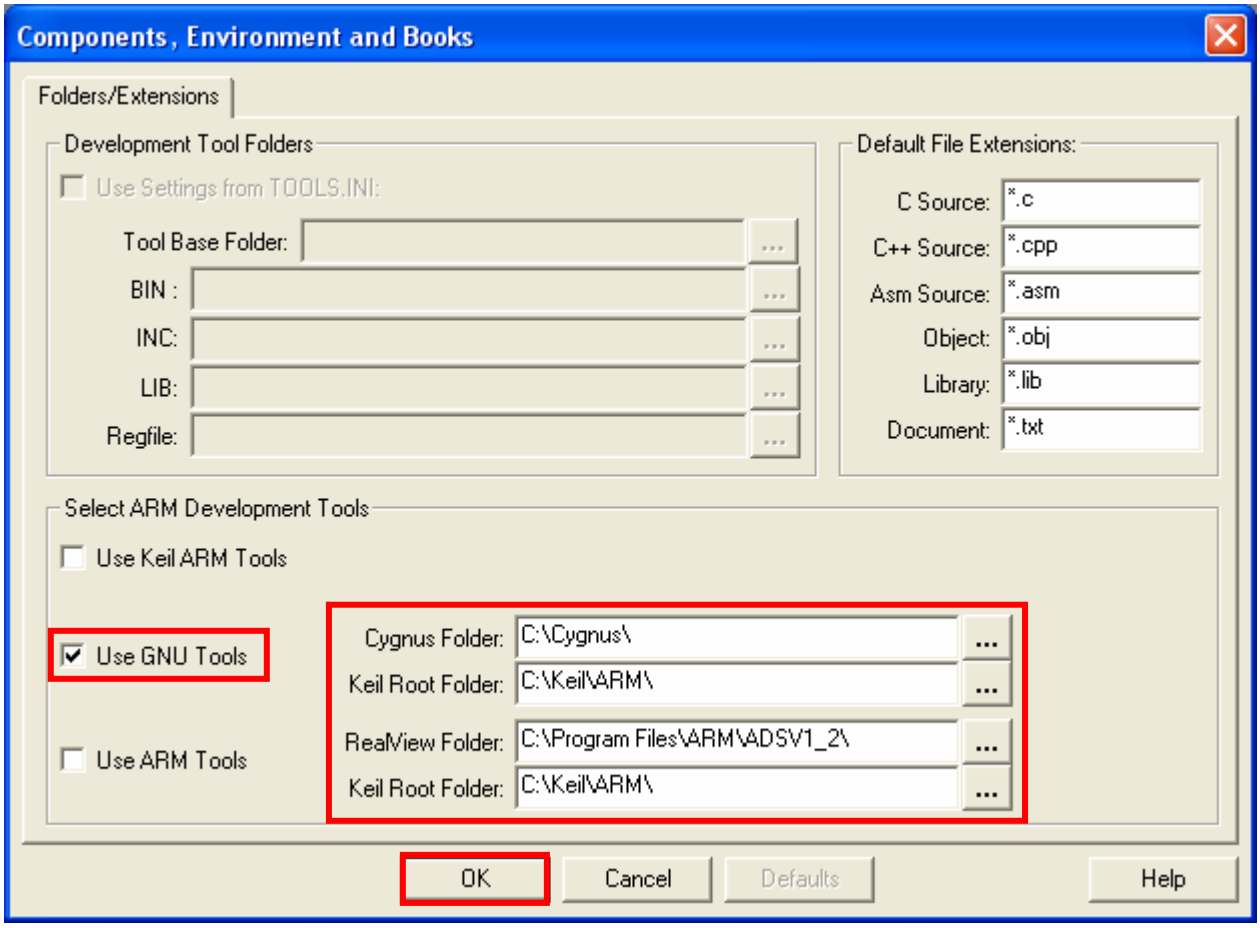
การใช้ Keil uVision3 ร่วมกับ GCCARM Complier

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

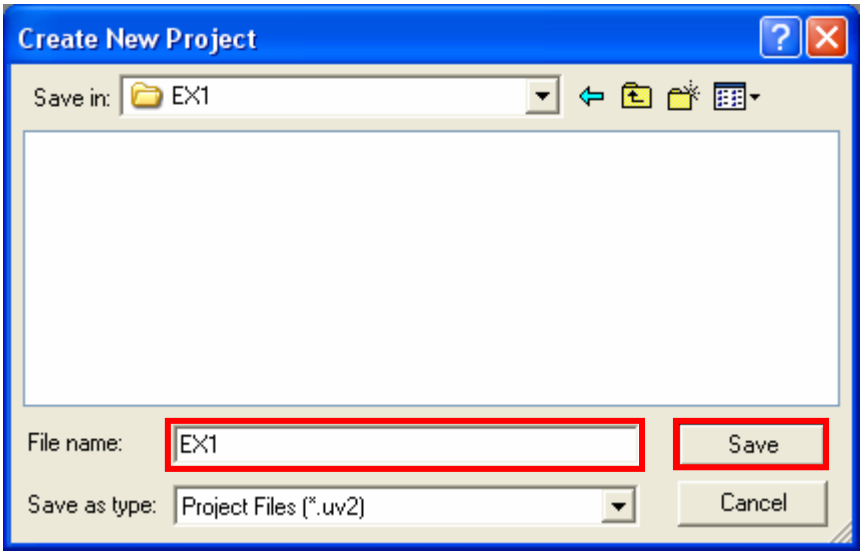
1. เปิดโปรแกรม Keil uVision3 ซึ่งเป็นโปรแกรม Text Editor ของ Keil-ARM ใช้สำหรับใช้ในการเขียนโปรแกรมที่เป็น Source Code ภาษาซี โดยจะมีลักษณะดังรูป



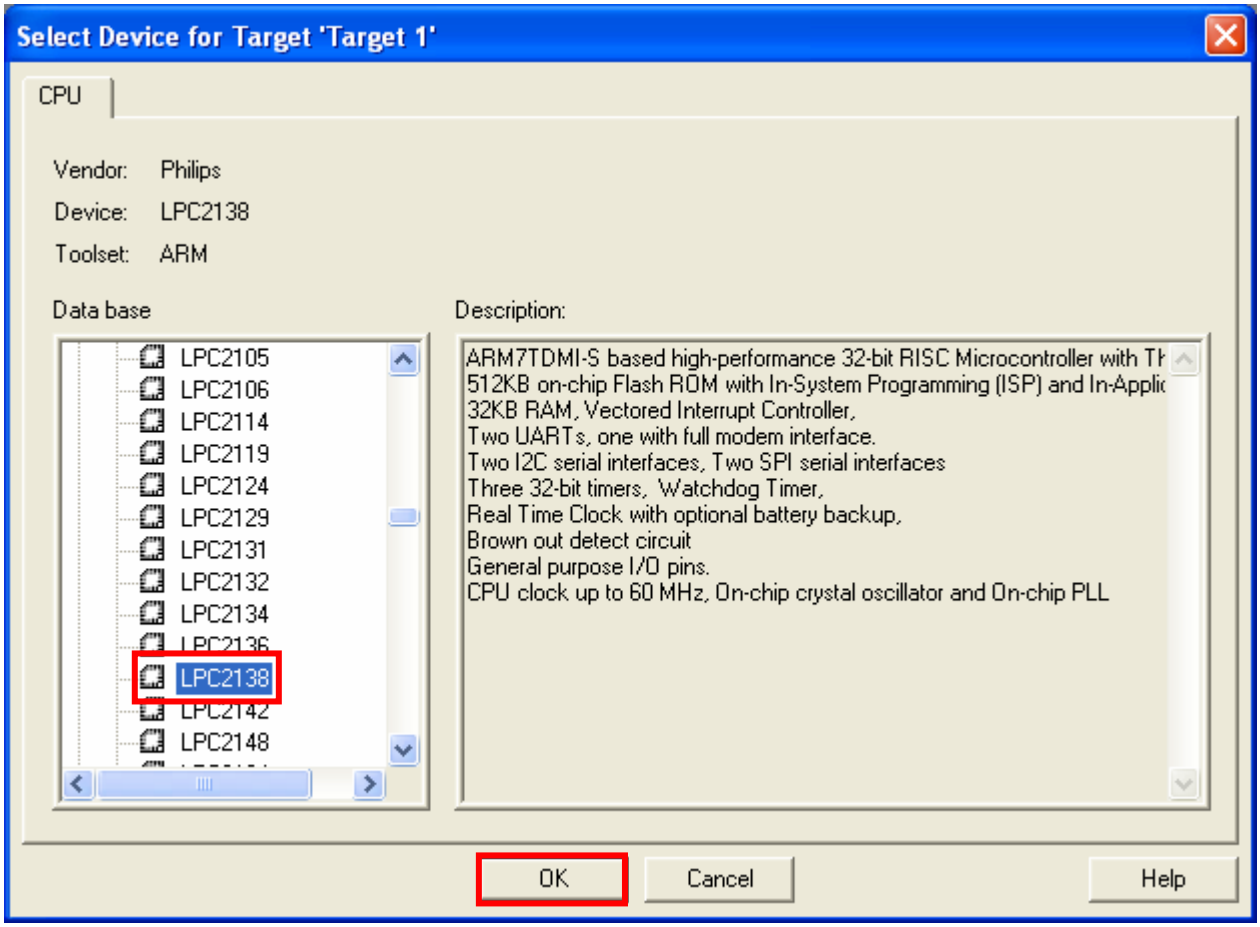
2. ทำการกำหนดค่าตัวเลือกในการแปลคำสั่งของ uVision3 ให้ใช้งานกับโปรแกรม Keil uVision3 และ GCCARM โดยให้เลือกคลิกเมาส์ที่เมนูคำสั่ง Project ⟶ Components ,Environment, Books… จากนั้นให้เลือกค่าตัวเลือกสำหรับกำหนดการใช้งาน Complier จากหัวข้อ Select ARM Development Tools ซึ่งจะมีค่าตัวเลือกอยู่ 3 แบบ คือ Use Keil ARM Tools ,Use GNU Tools และ Use ARM Tools โดยให้เลือกเป็น “Use GNU Tools” จากนั้นให้เลือกกำหนดตำแหน่ง Folder ที่ได้ทำการติดตั้งโปรแกรมไว้ ซึ่งตามปรกติแล้วถ้าติดตั้งโปรแกรม GCCARM ตามค่า Default จะอยู่ที่ C:\Cygnus… ซึ่งจะได้ดังรูป (ถ้าติดตั้งโปรแกรมต่างไปจากนี้ให้แก้ไขตามความเป็นจริงด้วย)



3. ทำการสร้าง Project File ขึ้นมาใหม่ โดยเรียกเมนูคำสั่ง Project ⟶ New Project จากนั้นให้เลือกกำหนดหรือสร้างตำแหน่ง Folder ที่จะบันทึก Project File พร้อมกับกำหนดชื่อ Project File ตามต้องการ เช่น ถ้าต้องการสร้าง Project File ชื่อ EX1 โดยเก็บไว้ใน Folder ชื่อ EX1 ก็สามารถกำหนดตำแหน่ง Folder และชื่อ Project File ได้เอง โดยเมื่อกำหนดชื่อในช่อง File name แล้วให้เลือก Save เพื่อบันทึก Project File ไว้ ดังรูป

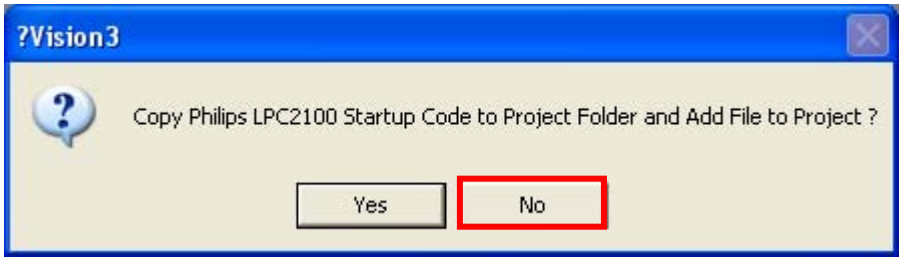


หลังจากกำหนดชื่อและสั่ง Save Project File แล้ว โปรแกรมจะรอให้ผู้ใช้ทำการกำหนด เบอร์ MCU ที่จะใช้งานใน Project ที่สั่ง Save นั้น ซึ่งในกรณีที่ใช้งานกับบอร์ด ET-ARM STAMP LPC2138 นั้น ให้เลือกกำหนดเบอร์ของ MCU เป็น LPC2138 ของ Philips แล้วเลือก OK ดังรูป



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

แต่เนื่องจากไฟล์ Startup ของ Keil-ARM นั้น เป็นไฟล์ภาษาแอสแซมบลี้ ซึ่งทั้ง Keil-ARM และ GCCARM นั้น จะใช้โปรแกรมแอสแซมเบอร์ต่างกัน ดังนั้นข้อกำหนดและรูปแบบไวยากรณ์ทางด้านการกำหนดค่าบางอย่างจะมีความแตกต่างกันอยู่ไม่สามารถใช้งานไฟล์ Startup ร่วมกันได้ ดังนั้นในการที่จะใช้โปรแกรม GCCARM ในการแปลคำสั่งให้นั้น ผู้ใช้จะต้องสร้างไฟล์ Startup ใหม่โดยต้องกำหนดรูปแบบให้ถูกต้องกับที่ GCCARM กำหนดไว้ ดังนั้นในที่นี้จะต้องเลือก “No” เพื่อไม่ให้ Keil uVision3 ทำการ Copy ไฟล์ Startup ของ Keil-ARM มาใช้ใน Project นี้ด้วย เนื่องจากรูปแบบของคำสั่งแตกต่างกันและใช้งานร่วมกันไม่ได้



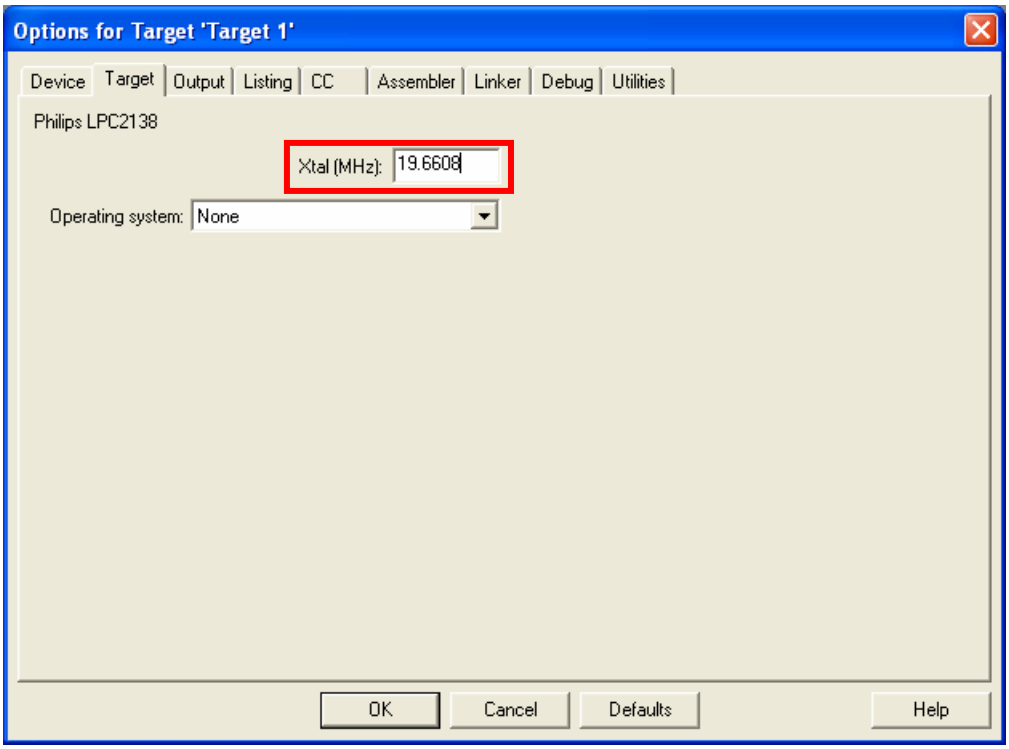
4. ให้ทำการ Copy File ชื่อ “Flash.ld” และ “Startup.S” ที่ทาง อีทีที จัดเตรียมไว้ใน CD-ROM ซึ่งเก็บไว้ใน Folder ชื่อ “SOURCE_GCC” มาไว้ในตำแหน่ง Folder เดียวกันกับ Project File ที่สร้างขึ้นมาใหม่นี้

โดยไฟล์ “Flash.ld” จะเป็น Script File ซึ่งใช้เก็บค่าตัวเลือกสำหรับกำหนดค่าตำแหน่งเริ่มต้นและสิ้นสุด รวมทั้งขนาดของหน่วยความจำ Flash และ RAM ที่มีอยู่จริงภายในตัวของ MCU เบอร์ LPC2138 ของ Philips ซึ่งเป็น MCU ที่ใช้กับบอร์ด ET-ARM STAMP LPC2138 โดยค่าต่างๆที่กำหนดไว้ใน Script File ตัวนี้จะใช้เพื่อบอกให้โปรแกรม Keil uVision3 ทราบว่าจะต้องกำหนดเงื่อนไขการแปลคำสั่งให้กับ GCCARM อย่างไรบ้าง ซึ่งถ้ากำหนดเงื่อนไขไม่ถูกต้องแล้ว HEX File ที่ได้จากการแปลคำสั่งก็จะเกิดความผิดพลาดตามไปด้วย

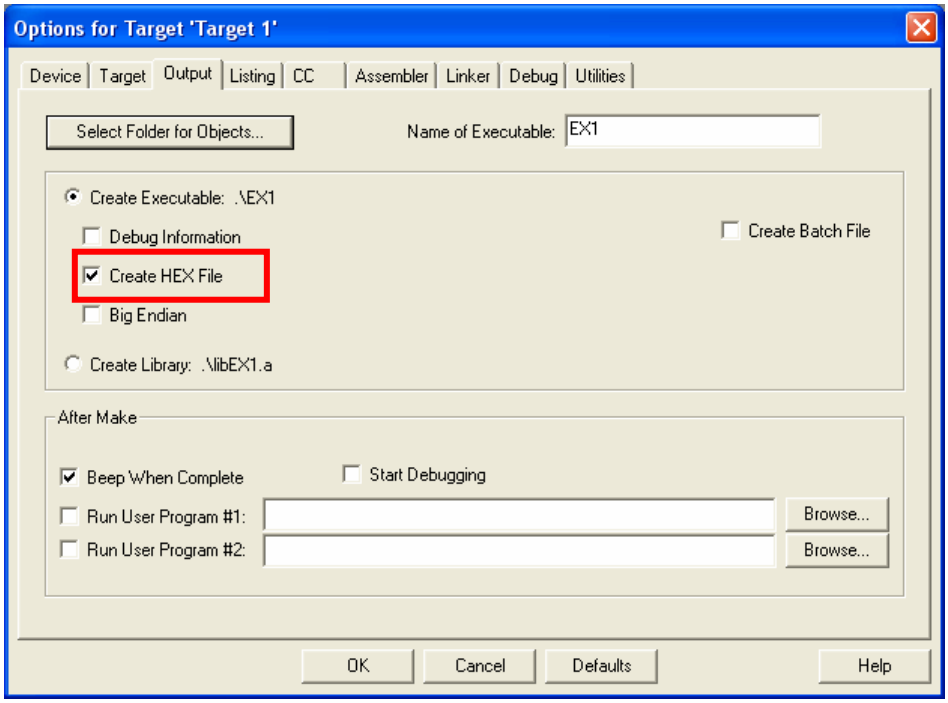
ส่วนไฟล์ “Startup.S” จะเป็นไฟล์ซึ่งบรรจุคำสั่งภาษาแอสแซมบลี้ของ ARM7 สำหรับทำหน้าที่กำหนดค่าเริ่มต้นการทำงานที่จำเป็นให้กับ MCU ซึ่งได้แก่การ กำหนดค่า Stack ใหกับ MCU การ Initial Phase-Lock-Loop การกำหนดค่าให้กับ MAM Function และการกำหนดตำแหน่ง Vector ต่างๆของ MCU

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

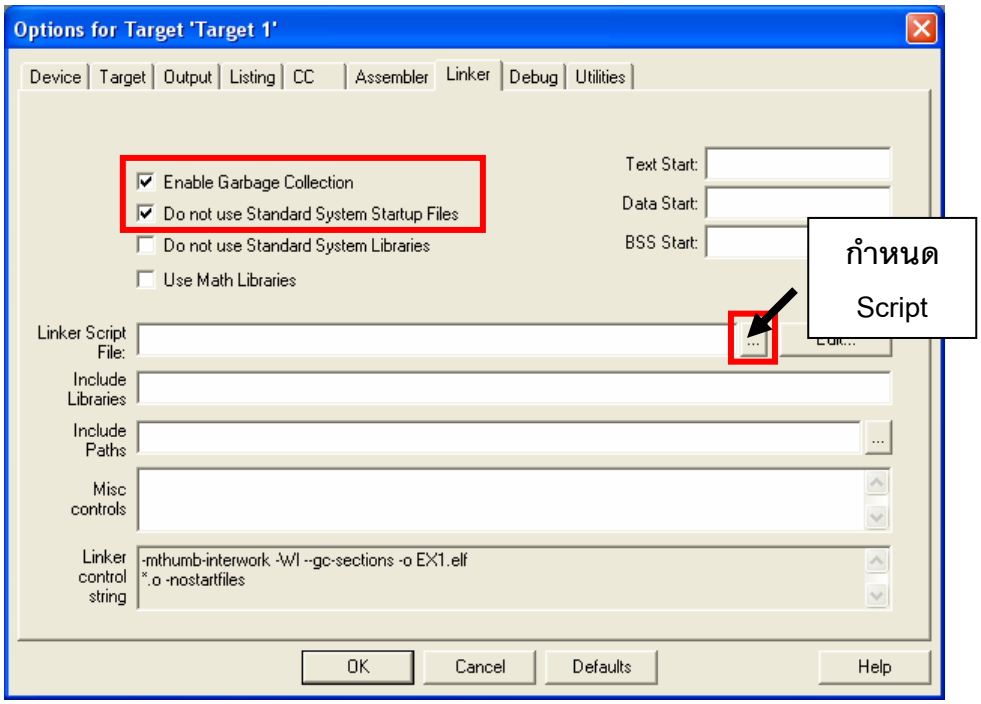
5.1 X-TAL ให้กำหนดเป็น 19.6608 MHz

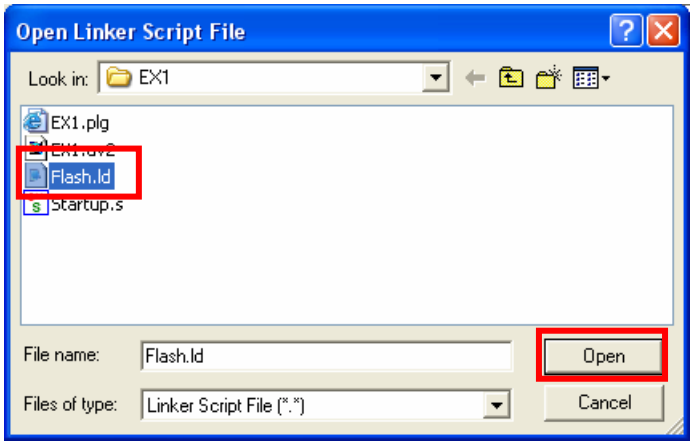


5.2 Output ให้เลือกคลิกเมาส์ที่ค่าตัวเลือก Create HEX File

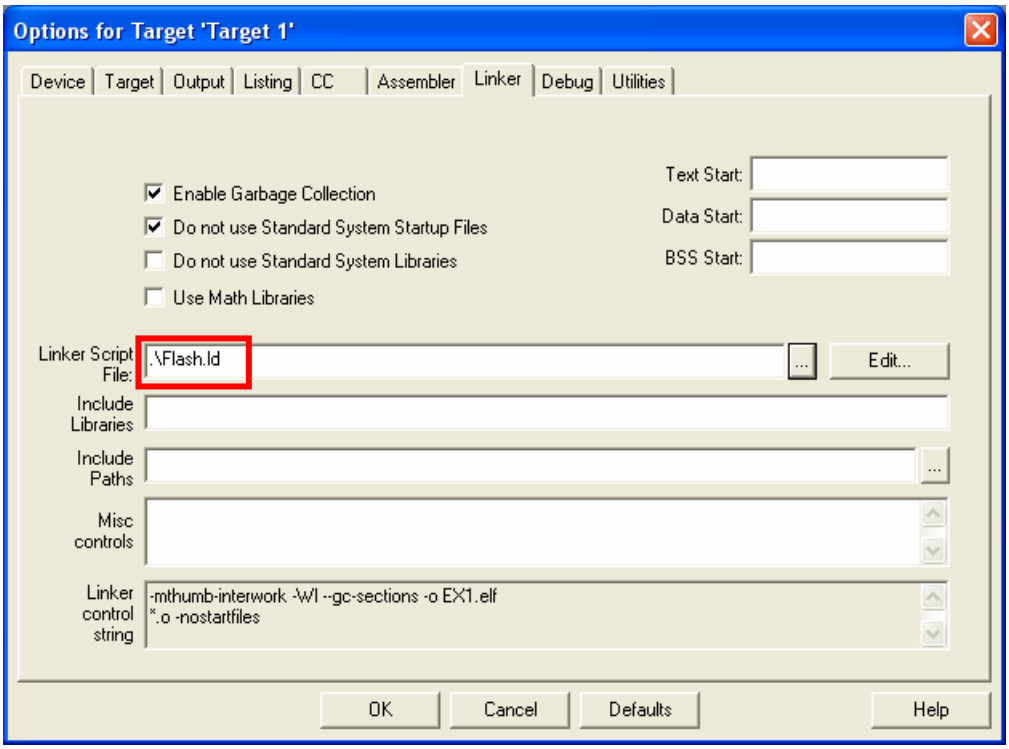


5.3 Linker ให้เลือกคลิกเมาส์ที่ตัวเลือก Enable Garbage Collection และ Do not use Standard System Startup Files ส่วน Linker Script File นั้นให้เลือกกำหนดเป็น “Flash.ld” โดยคลิกเมาส์ที่ปุ่มคำสั่ง [...] ที่อยู่ระหว่างช่องแสดงชื่อ Script File และปุ่มคำสั่ง “Edit...” จากนั้นก็ให้คลิกเมาส์ที่ไอคอนของไฟล์ “Flash.ld” แล้วเลือก Open ซึ่งจะปรากฏชื่อของ Script File ดังกล่าวในช่องแสดงชื่อของ Script File ทันที





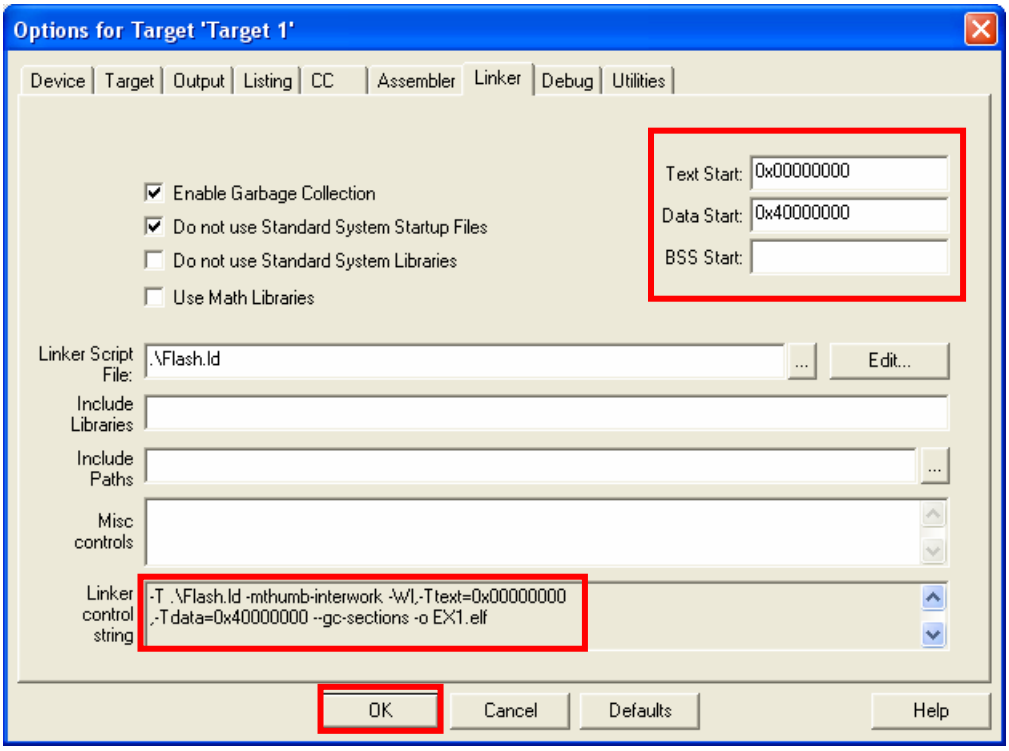
↓



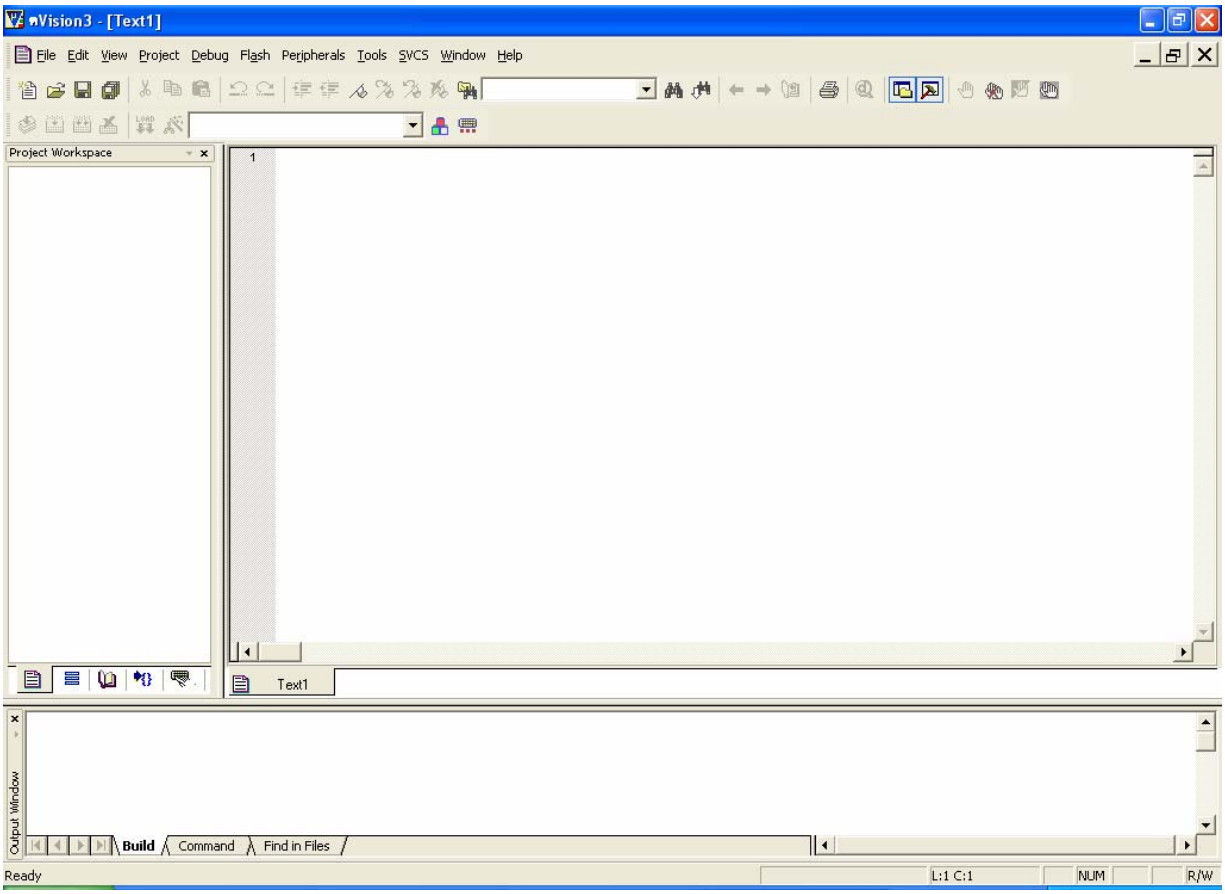
หลังการทำการกำหนด Script File ให้กับ Project เป็นที่เรียบร้อยแล้ว ในขั้นตอนถัดไปให้ทำการกำหนดค่าตำแหน่งแอดเดรสเริ่มต้นของหน่วยความจำที่จะใช้ในการเก็บโปรแกรมและข้อมูล โดยในกรณีของ LPC2138 ให้ทำการกำหนดค่าเป็นดังนี้

- Text Start ให้กำหนดเป็น 0x0000 0000
- Data Start ให้กำหนดเป็น 0x4000 0000

ซึ่งจะสังเกตเห็นว่าโปรแกรม Keil uVision3 จะแสดง Option ในการแปลคำสั่งให้เราทราบด้วยทางช่องแสดงค่าของ Linker Control String โดยค่าจะเปลี่ยนแปลงไปตามการกำหนดเงื่อนไขของเราดังรูป



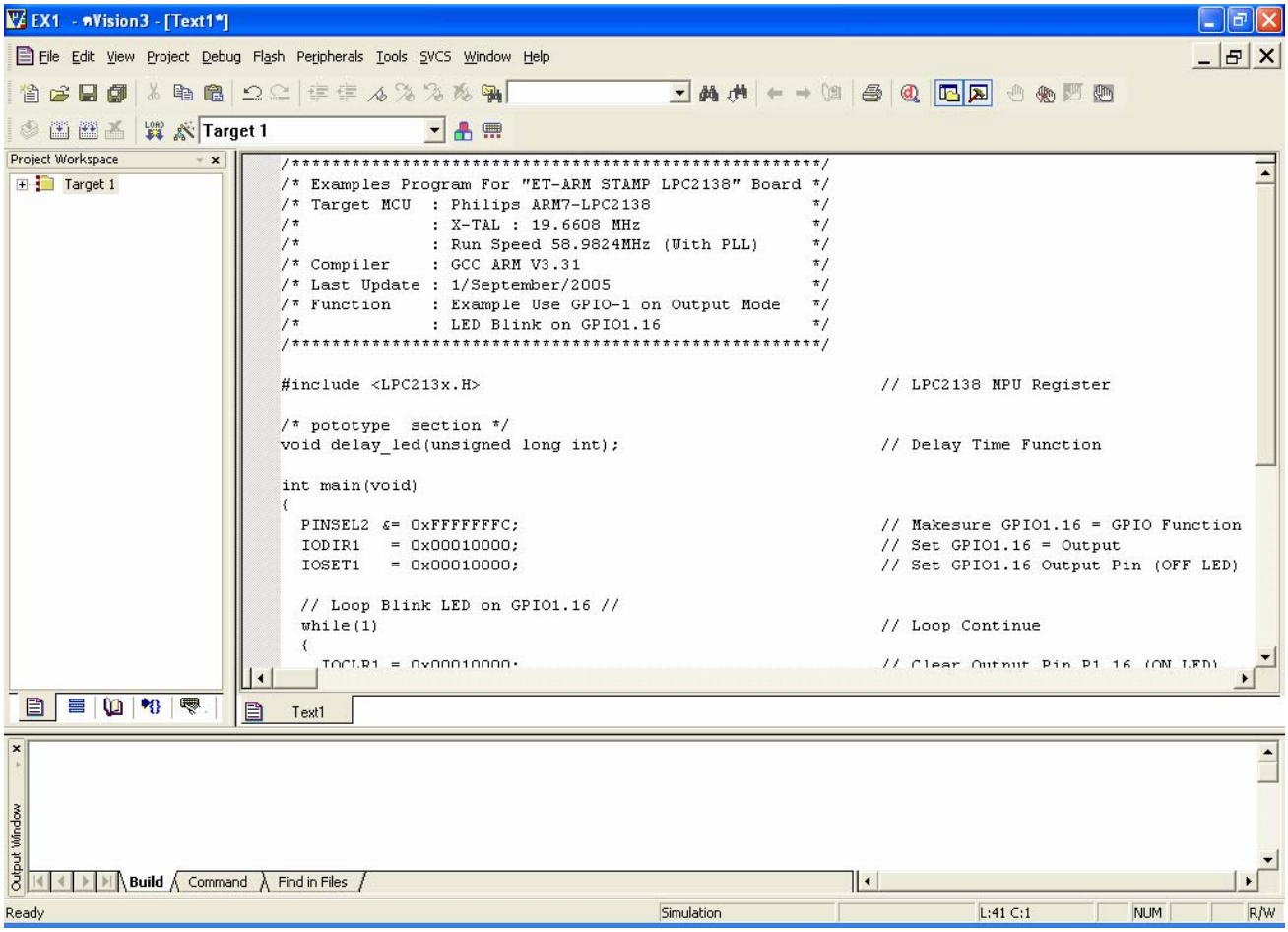
6. เริ่มต้นเขียน Source Code ภาษาซี โดยให้เลือกคลิกเมาส์ที่เมนูคำสั่ง File ⟶ New… ซึ่งจะได้พื้นที่ในการเขียน Text File เกิดขึ้นมา โดยในครั้งแรกจะกำหนดชื่อตามค่า Default เป็น “Text1” ดังรูป



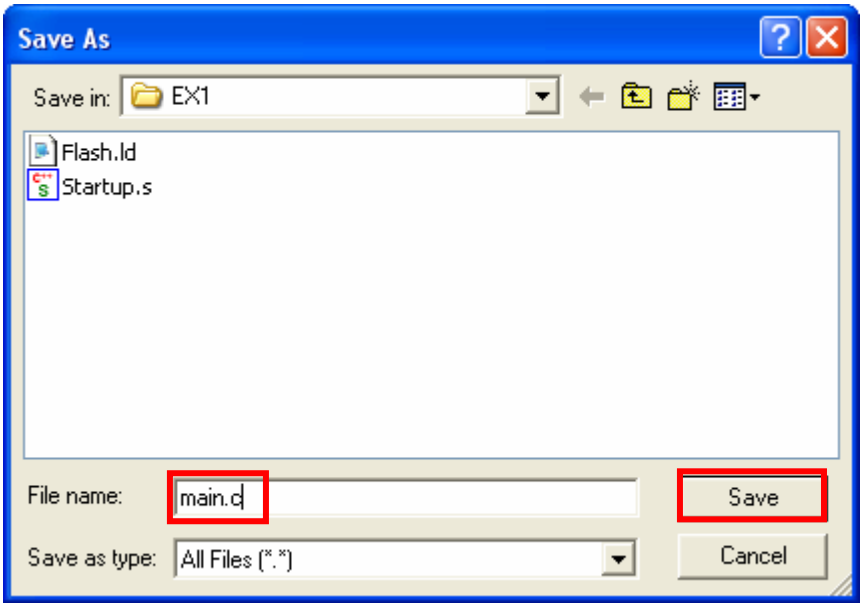
ในขั้นตอนนี้ให้ทำการพิมพ์ Source Code ภาษาซี ตามข้อกำหนดของ GCCARM ในพื้นที่เขียนโปรแกรมตามต้องต้องการดังตัวอย่าง

```
/*****************************************************/
/* Examples Program For "ET-ARM STAMP LPC2138" Board */
/* Target MCU  : Philips ARM7-LPC2138                */
/*           : X-TAL : 19.6608 MHz                   */
/*             : Run Speed 58.9824MHz (With PLL)     */
/* Compiler    : GCC ARM V3.31                       */
/* Last Update : 1/September/2005                    */
/* Function    : Example Use GPIO-1on Output Mode    */
/*             : LED Blink on GPIO1.16               */
/*****************************************************/

#include <LPC213x.H>                // LPC2138 MPU Register

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

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

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

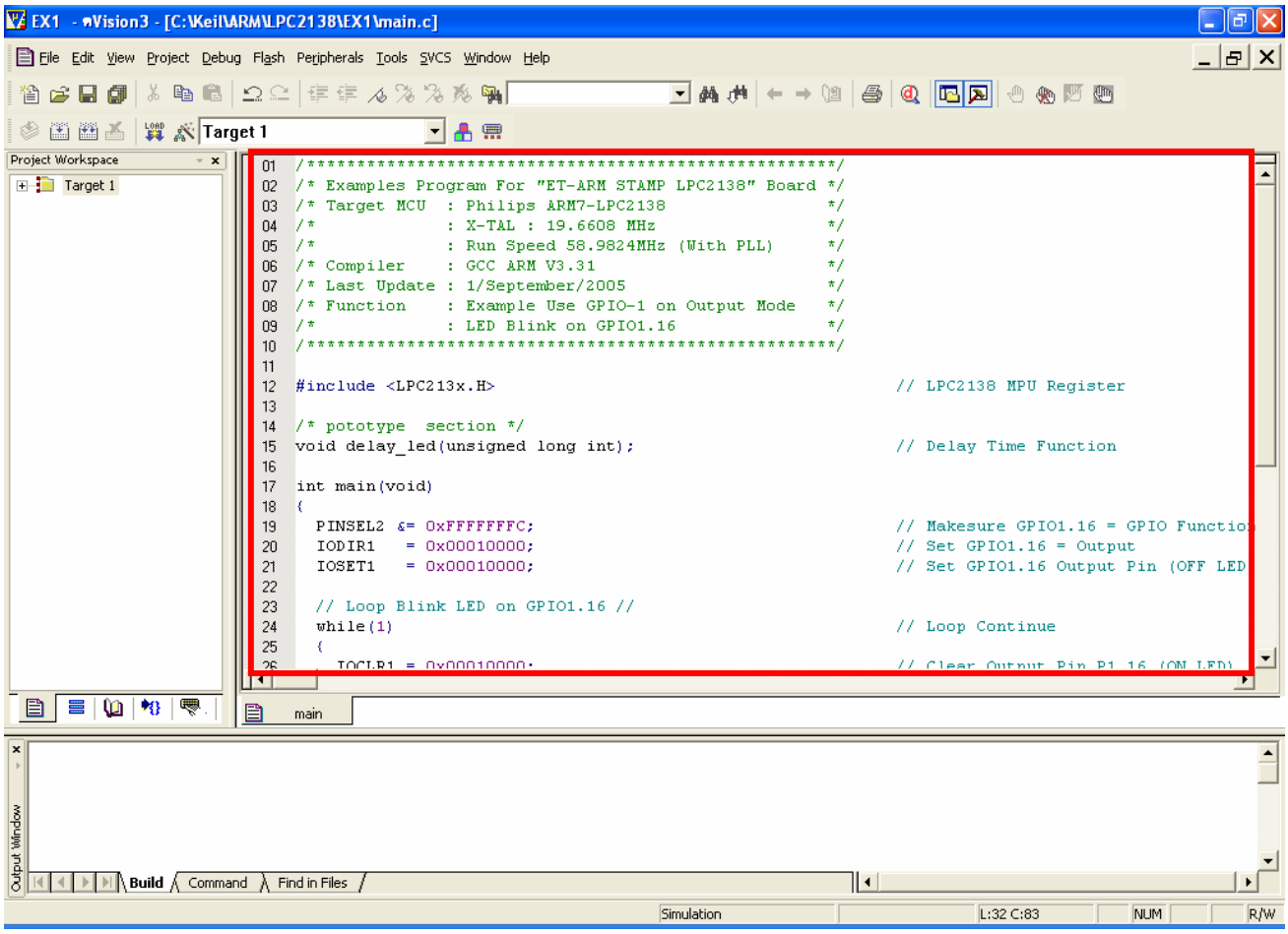
ตัวอย่างโปรแกรมไฟกระพริบ 1 ดวง ที่ GPIO1.16



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

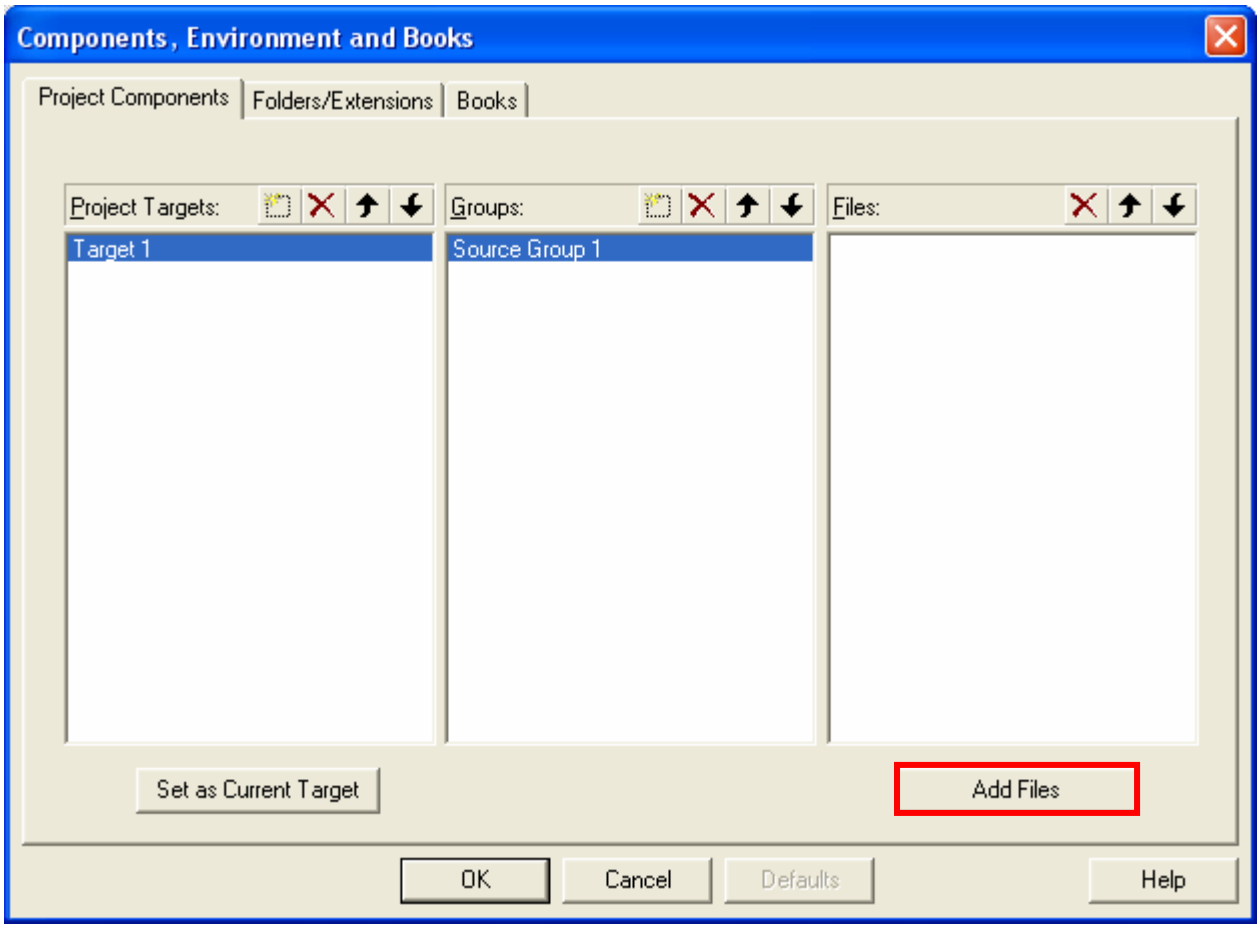


7. ทำการสั่ง Add File ต่างๆเข้ากับ Project File โดยให้เลือกคลิกเมาส์ที่คำสั่ง Project ⟶ Components, Environment, Books… จากนั้นให้เลือกที่ Tab Project Components แล้วเลือกที่ Add File ที่ต้องการจะเพิ่มเข้าไปใช้งานร่วมกับ Project File

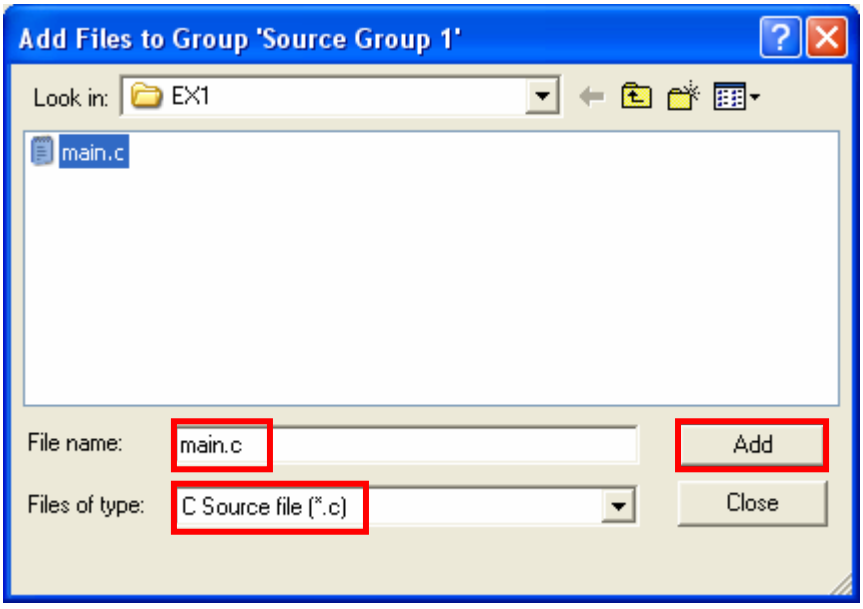
โดยในครั้งแรกให้เลือก Files of type เป็น “C Source files(*.c)” ซึ่งจะปรากฏชื่อไฟล์ต่างๆที่เป็น Source Code ภาษาซีให้เห็น โดยในที่นี้ให้เลือกคลิกเมาส์ที่ไอคอนของไฟล์ชื่อ “main.c” แล้วเลือก Add เพื่อสั่งเพิ่มไฟล์ชื่อ “Startup.s” เข้าไปรวมกับ Project Files ที่เราสร้างไว้

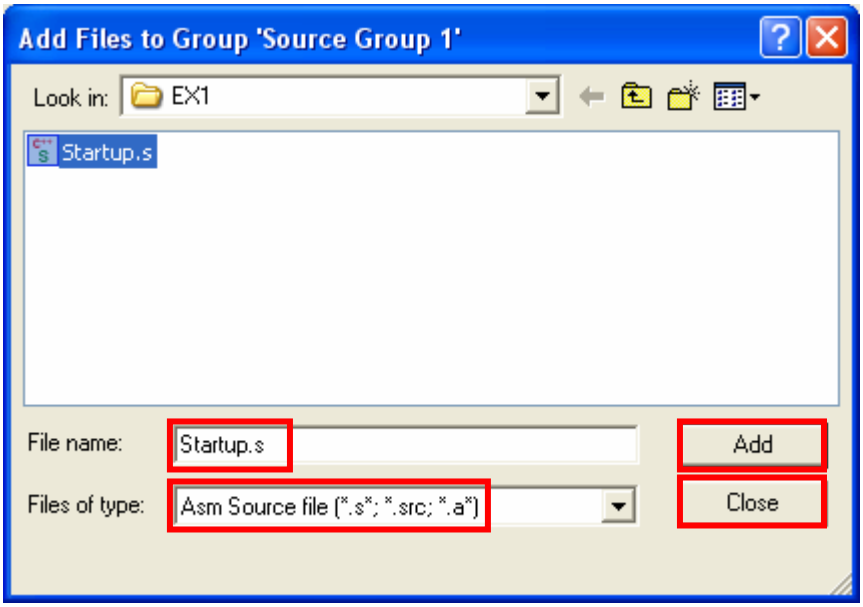
จากนั้นให้เลือกกำหนด File of type ใหม่เป็น “ASM Source files(*.s*;*.src;*.a*) ซึ่งจะปรากฏชื่อของไฟล์ Startup.s ให้เห็นในช่องแสดงชื่อไฟล์ ให้ทำการคลิกเมาส์ที่ไอคอนของไฟล์ “Startup.s” แล้วเลือก Add เพื่อสั่งเพิ่มไฟล์ชื่อ “Startup.s” เข้าไปรวมกับ Project Files ที่เราสร้างไว้

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

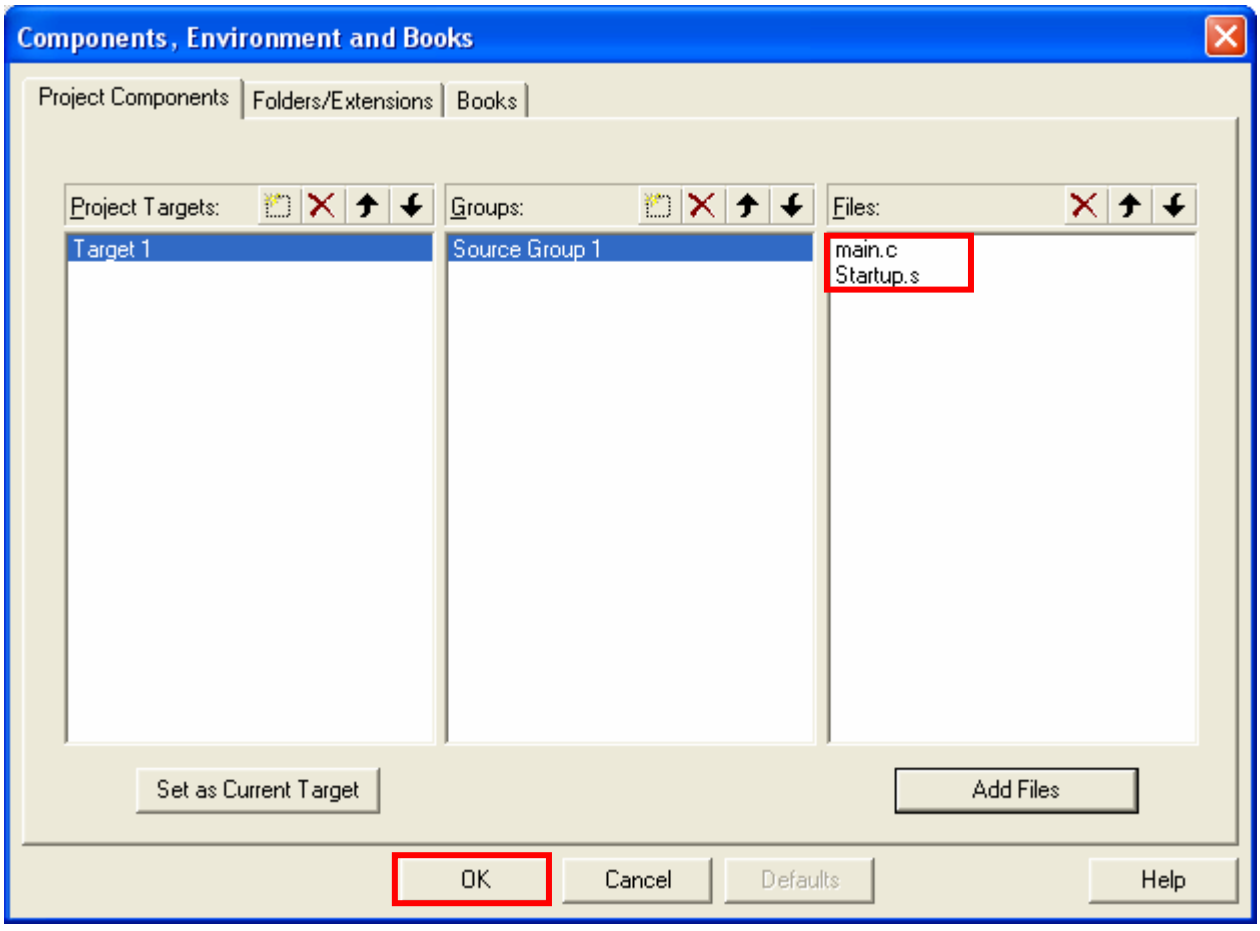


↓



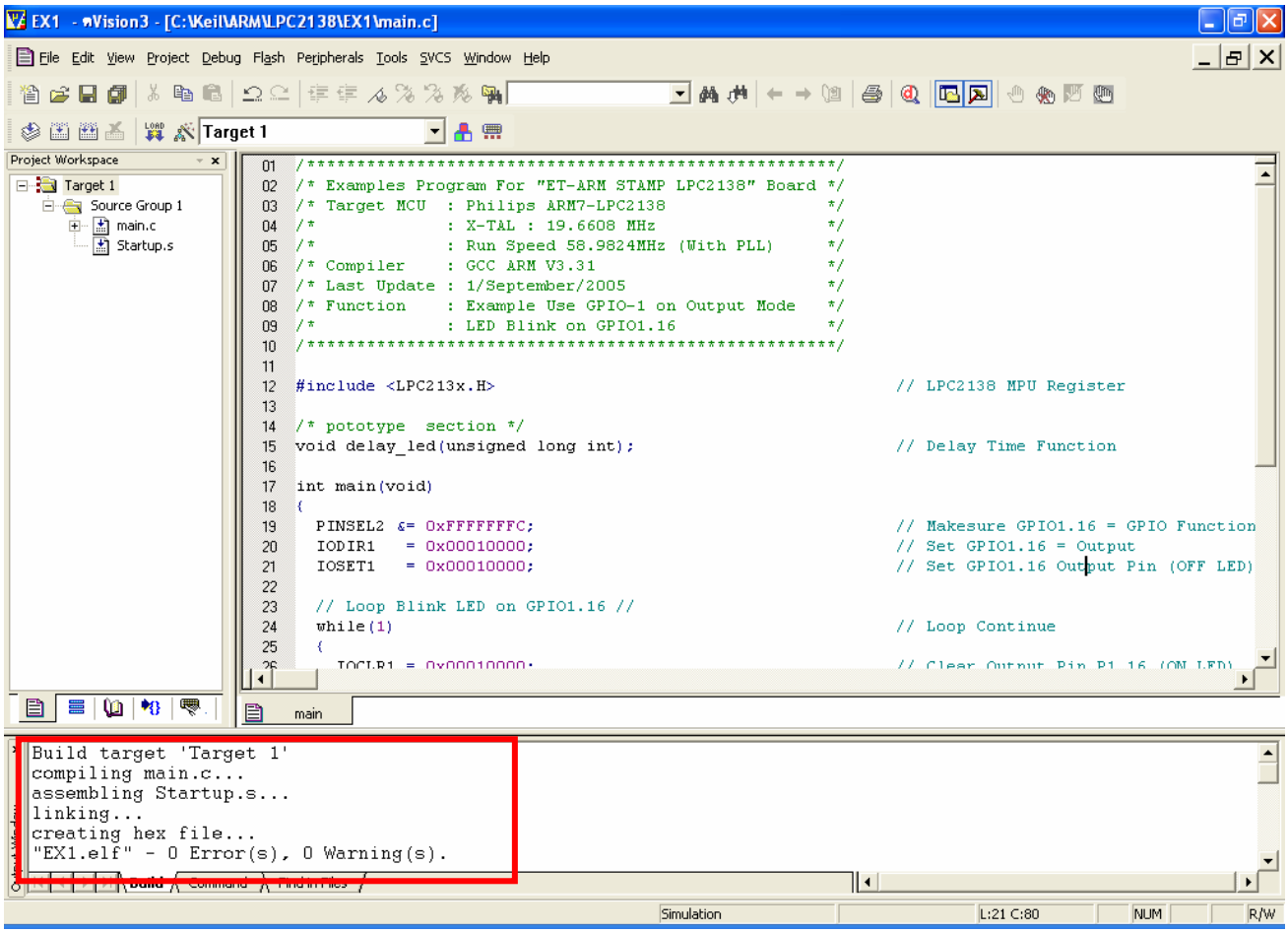


↓



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 GCCARM 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

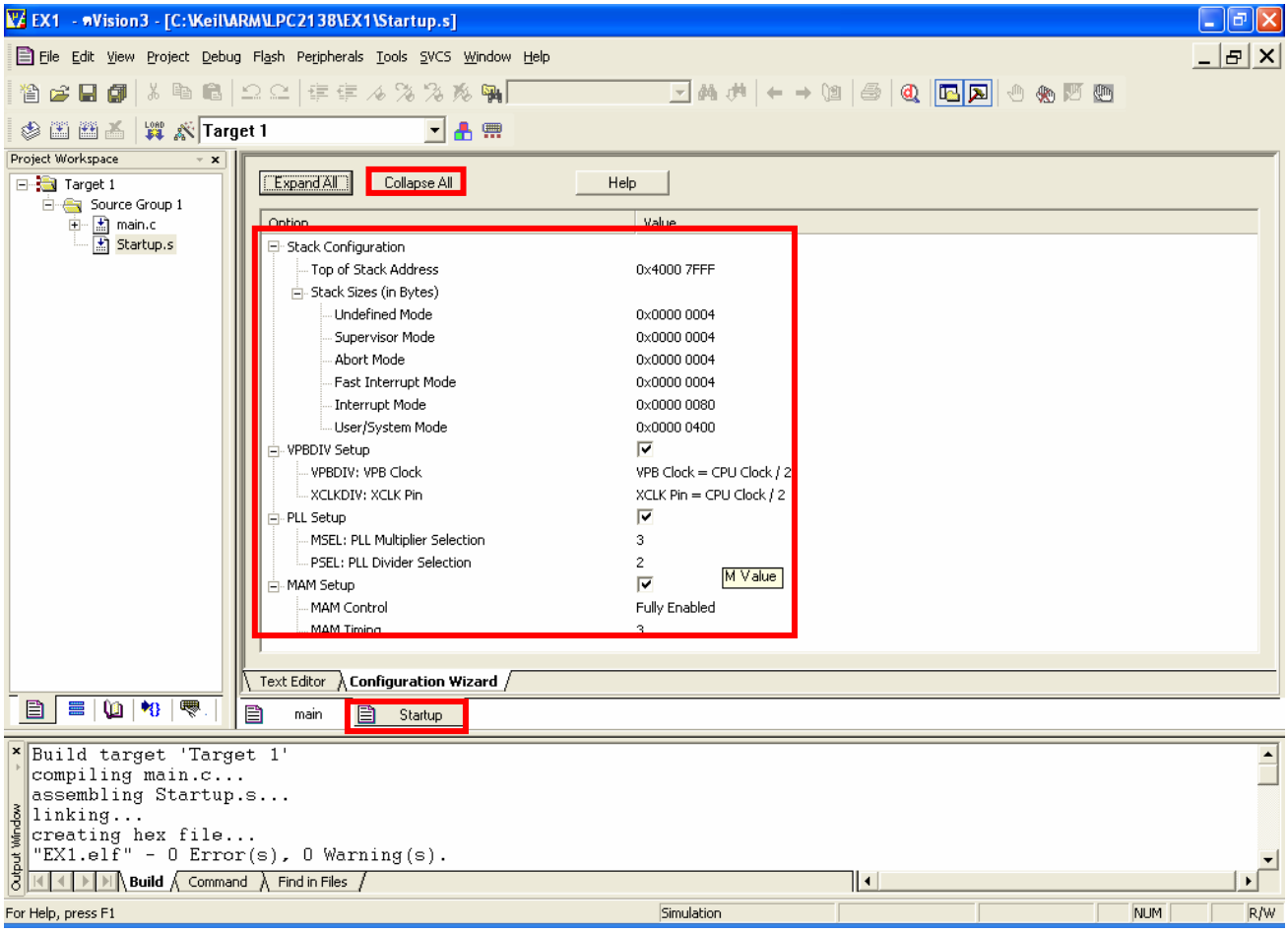
เพื่อให้การทำงานของ MCU มีประสิทธิภาพสูงสุด ทั้งในด้านของความเร็วในการประมวลผลคำสั่ง และการทำงานควรกำหนดค่าต่างๆให้กับ MCU ดังต่อไปนี้

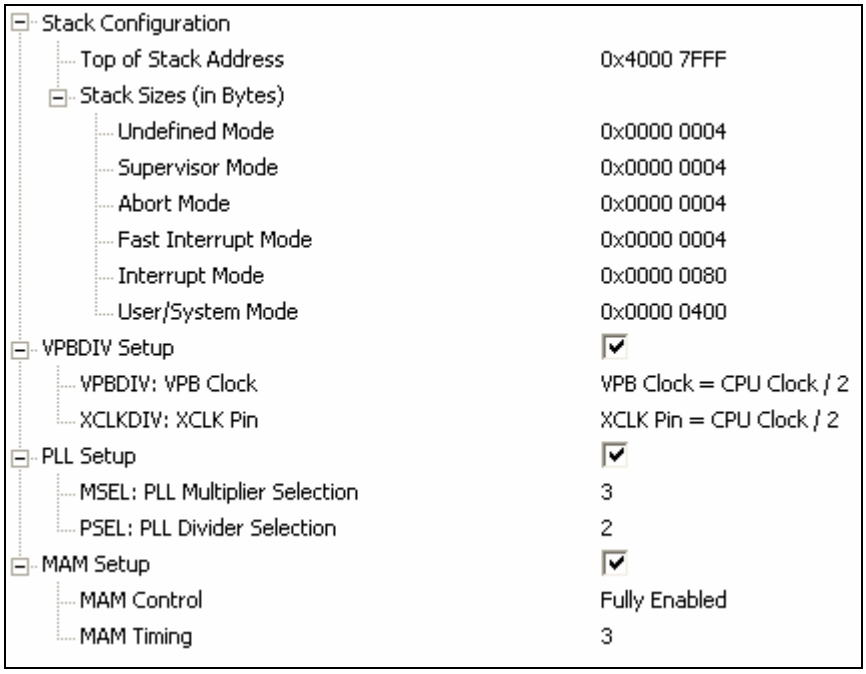
- ควรกำหนดค่า PLL ให้ได้ค่า Processor Clock (cclk) = 58.9824MHz โดยในกรณีที่ใช้ค่า XTAL เป็น 19.6608MHz ต้องให้ค่า M(Multiply)=3 และ P(Divide)=2 และ FFCO =235.9296MHz
- ควรกำหนดค่า VPB Clock (pclk) ให้มีค่า เป็นครึ่งหนึ่งของ cclk หรือ 29.4912MHz
- ควรกำหนดค่า MAM Timing ให้มีค่า 3 Cycle ของ cclk (MAMTIM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

โดยการกำหนดค่าต่างๆดังกล่าวข้างต้นนั้น สามารถทำได้ 2 วิธีคือ การเขียน Code คำสั่งในโปรแกรมที่เราเขียนขึ้นเองทั้งหมด หรือใช้การ Copy ไฟล์ที่เป็น Startup File และ Script File ที่สร้างไว้เรียบร้อยแล้วมาใช้งาน จากนั้นจึงสั่ง Add Startup File และ Script File เข้ามาใช้งานใน Project ที่เราสร้างขึ้น ซึ่งการตรวจสอบและแก้ไขค่าของ Startup File และ Script File นั้น สามารถทำได้ 2 แบบ คือ เข้าไปแก้ไข Code คำสั่งในไฟล์ตามความต้องการ หรือ ใช้การกำหนดค่า Startup File และ Script 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

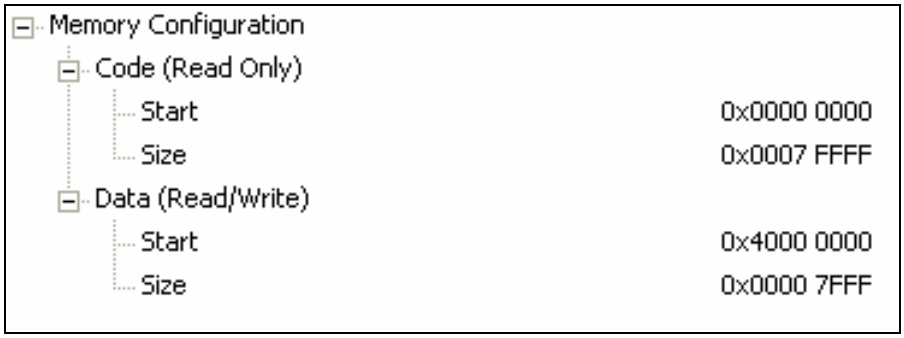




แสดงการกำหนดค่า Startup File สำหรับ LPC2138

การตรวจสอบ Script File

สำหรับหน้าที่ของ Script File นั้นจะใช้เก็บค่าที่กำหนดตำแหน่งและขนาดของหน่วยความจำใน MCU หรือ Target บอร์ดที่จะใช้กับโปรแกรมที่เราเขียนขึ้น โดยในกรณีที่ใช้ LPC2138 นั้น จะมีขนาดของหน่วยความจำ Code Flash ขนาด 512K โดยมีตำแหน่งเริ่มต้นอยู่ที่ 00000000H-0007FFFFH ส่วนหน่วยความจำ Data RAM จะมีขนาด 32KByte โดยมีตำแหน่งอยู่ที่ 40000000H-40007FFFH ซึ่งใน Script File ชื่อ “Flash.ld” ที่จัดทำไว้ในตัวอย่างภาษาซี ของ GCCARM โดยอีทีที นั้นจะกำหนดค่า Configuration ของหน่วยความจำไว้อย่างถูกต้องสำหรับ LPC2138 อยู่แล้ว แต่ถ้ามีการเข้าไปแก้ไขหรือเปลี่ยนแปลงค่าเป็นอย่างอื่นอาจทำให้โปรแกรมที่ได้จากการแปลคำสั่งของ GCCARM ไม่สามารถทำงานได้อย่างถูกต้อง โดยการตรวจสอบค่าความถูกต้องของ Configuration ใน Script File สามารถทำได้โดยการสั่ง Open ไฟล์ ชื่อ “Flash.ld” ซึ่งอยู่ใน Folder เดียวกันกับ Project File ด้วยโปรแกรม Keil uVision3 หรือถ้ามีการสั่ง Open File ไว้อยู่แล้วให้เลือก Tab ของไฟล์ไปยัง “Flash.ld” แล้วเลือก”Expand All” จากนั้นให้ตรวจสอบค่าต่างๆดูว่าเป็นไปตามที่กำหนดไว้หรือไม่ ซึ่งค่าที่ถูกต้องจะต้องเป็นดังรูป



แสดงการกำหนดค่า Script File สำหรับ LPC2138

ตัวอย่าง Code ภาษาซีของ GCCARM สำหรับ Initial การทำงานของ LPC2138

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

```
// Initial PLL & VPB Clock For ET-ARM7 STAMP LPC2138
// 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 LPC2138 ก่อนเริ่มต้นทำงานใน Main โปรแกรม