ETT

www.ett.co.th

KEIL SOFTWARE

ARM7

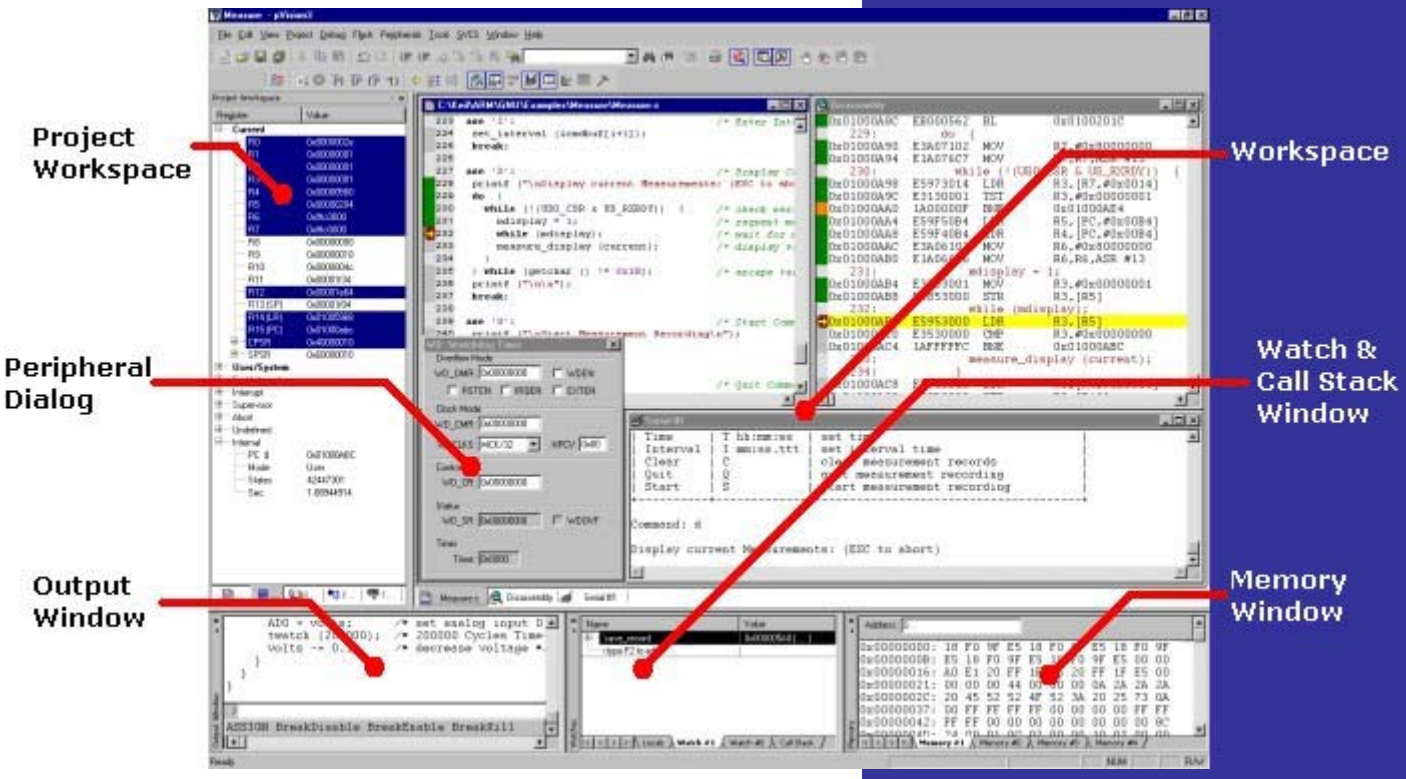
ARM Development Guide

µVision3 จาก Keil Software เป็น Tools ซึ่งมีความสามารถในการจัดการ Project แก้ไข Source Code การ Debug โปรแกรม และ Flash Program ซึ่งรวมอยู่ในโปรแกรมเดียว ซึ่งยอมให้ผู้พัฒนาใช้ Compiler ที่คุ้นเคยเช่น Keil, GNU, หรือ ARM กับ CPU ARM7-TDMI จาก Analog Devices, Atmel, Philips, และ Samsung

สำหรับ µVision3 ได้เพิ่มความสามารถใหม่เช่น Source Outline, Function Navigation, Editor Templates, Incremental Search, 2 Configuration Wizard, Logic Analyzer, CAN และ I^2C Simulation, Flash Programming, และ JTAG Debugger สามารถ Download ได้ที่ www.keil.com/demo/eval/arm.htm โดยถ้าเป็นรุ่นทดลองใช้ให้ ดูข้อจำกัดการใช้งานที่ Referents

Quick Start Guide นี้จะแนะนำคร่าวๆเกี่ยวกับการสร้าง Project ที่ใช้ µVision3 IDE (Keil CARM) ซึ่งมีรายละเอียดดังนี้

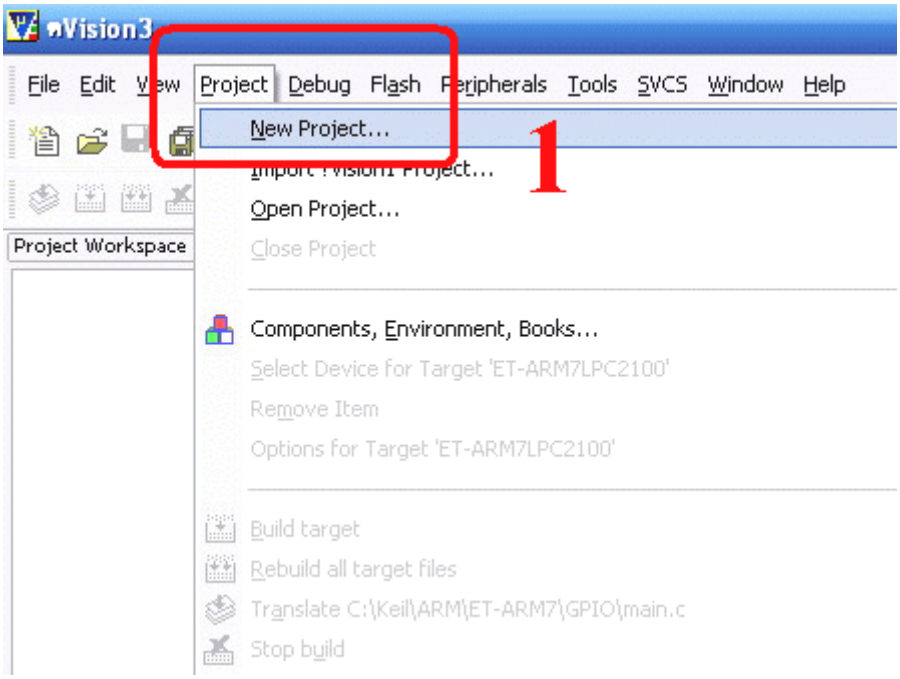
- การจัดการ Project Files, ตั้งค่าอุปกรณ์
- วิธีการ Simulator
- วิธีการ Upload ด้วย LPC2000 Flash Utility จาก Philips



รูปที่ 1 โครงสร้างของ µVision3 IDE

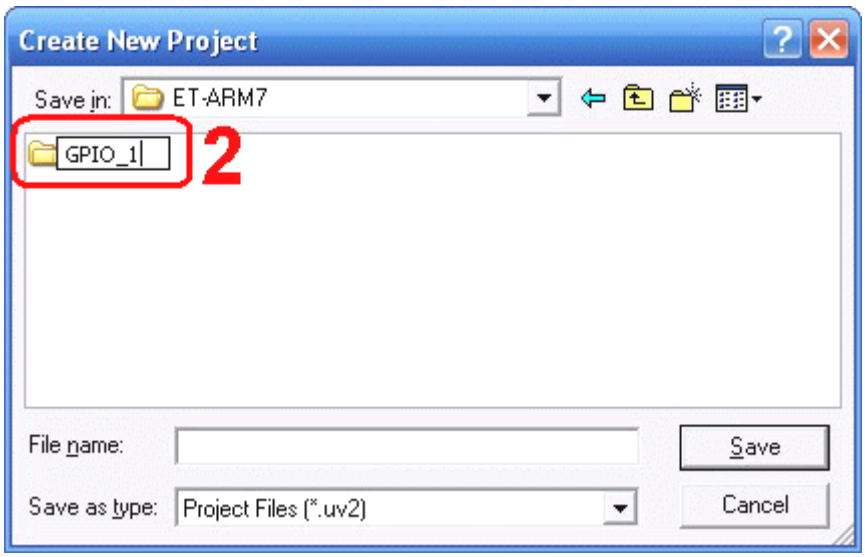
การสร้าง Project ใหม่

เปิด Keil µVision3 ขึ้นมา แล้วไปที่ Project → New Project… (หมายเลข 1) ดังแสดงในรูปที่ 2



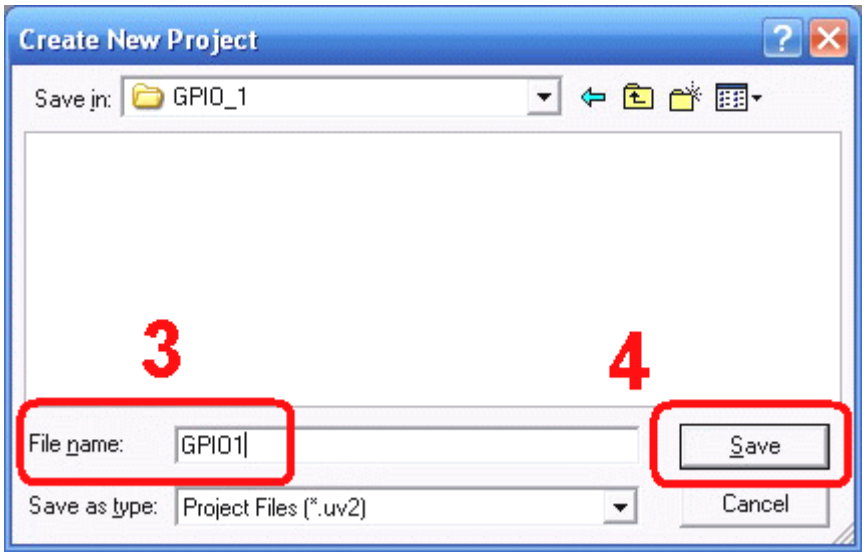
รูปที่ 2 แสดงการสร้าง New Project

จากนั้นสร้างโฟลเดอร์สำหรับเก็บรวบรวมไฟล์ต่างๆของ Project ขึ้นมา โดยแนะนำให้กำหนดชื่อให้สอดคล้องกับงานที่ทำและเพื่อความเป็นระเบียบ ในตัวอย่างนี้ตั้งเป็น GPIO_1 (หมายเลข 2) ดังแสดงในรูปที่ 3



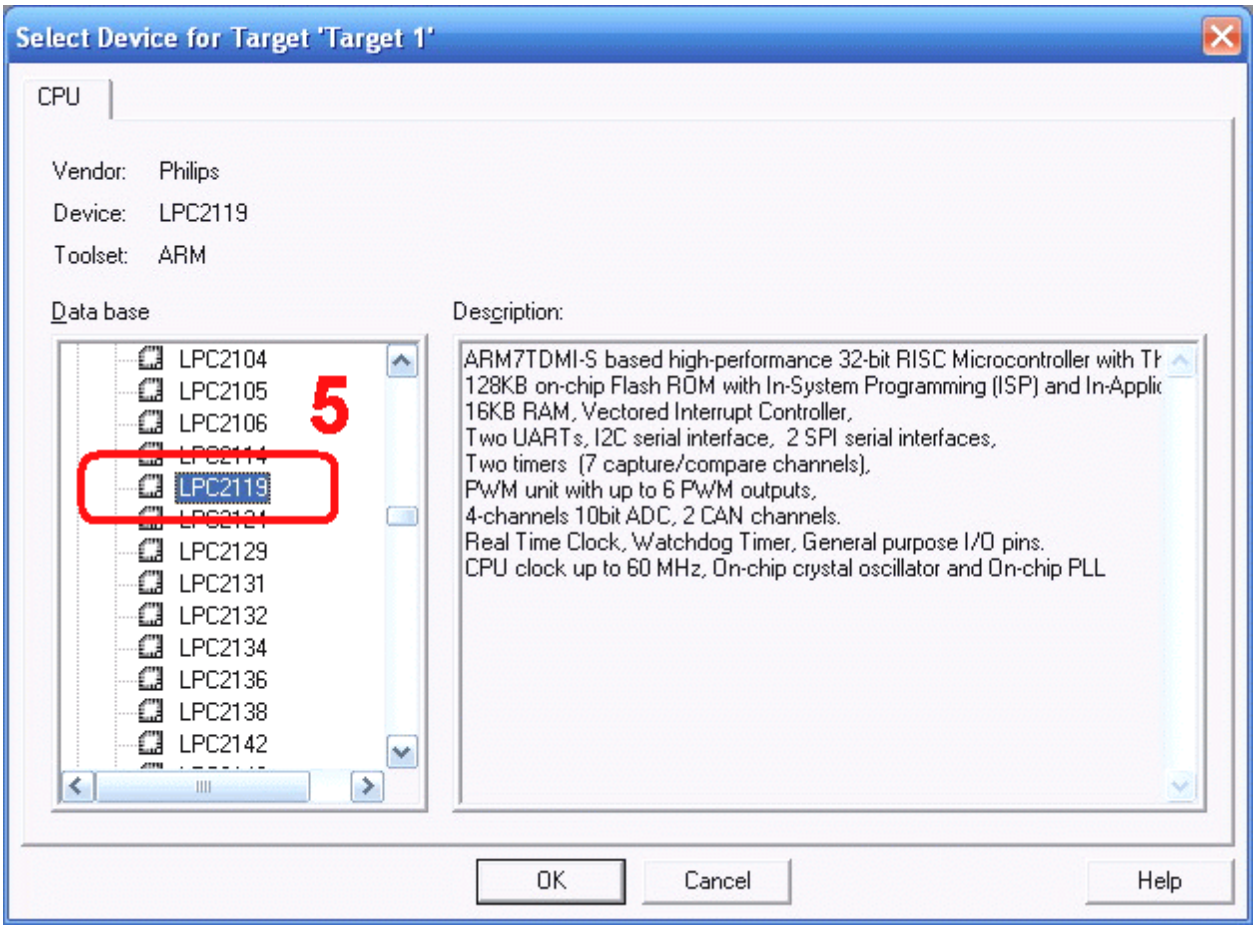
รูปที่ 3 แสดงการสร้าง Folder สำหรับโปรเจ็ค

หลังจากที่ทำการกำหนดชื่อ Project (หมายเลข 3) เรียบร้อยแล้ว ให้คลิกที่ Save (หมายเลข 4) ดังแสดงในรูปที่ 4



รูปที่ 4 แสดงการตั้งชื่อ Project

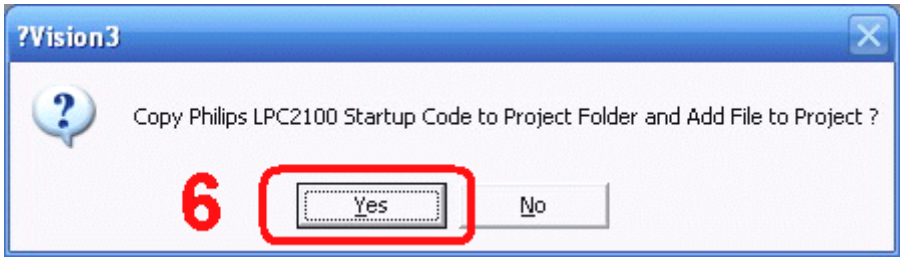
เมื่อตั้งชื่อ Project แล้ว โปรแกรมจะแสดงหน้าต่างขึ้นมาให้เลือกเบอร์ของ MCU ที่จะใช้งานกับ Project ในที่นี้เลือกเบอร์ Philips → LPC2119 (หมายเลข 5) ดังแสดงในรูปที่ 5 แล้วคลิกเลือกที่ “OK”



รูปที่ 5 แสดงการเลือกเบอร์ CPU

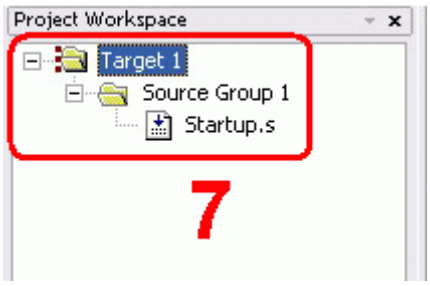
เมื่อเลือกเบอร์ CPU แล้ว µVision3 IDE จะถามว่าต้องการ Copy และ Add file Startup code ไปที่ Project Folder หรือไม่ ตอบ Yes เพื่อยอมรับ (หมายเลข 6) ดังแสดงในรูปที่ 6 ซึ่งตัว Startup code นี้คือ ไฟล์ซึ่งบรรจุโปรแกรมภาษา แอสแซมบลี้ สำหรับใช้กำหนดค่าเริ่มต้นการทำงาน หรือค่า Configurations ของ MCU ก่อนที่จะสั่งให้ MCU กระโดดไปเริ่มต้นทำงานตามโปรแกรมที่ผู้ใช้เขียนขึ้นใน Main Program นั่นเอง ซึ่งถ้าไม่มีการ Add file Startup code เข้าไปใน Project ด้วย ผู้ใช้จะต้องเขียนโปรแกรมเพื่อกำหนดค่าเริ่มต้นการทำงานของ MCU เองทั้งหมด เช่น การกำหนดค่าของ Stack Pointer การกำหนดค่าการทำงานของวงจร Phase-Lock-Loop เป็นต้น

แต่อย่างไรก็ตามค่าเริ่มต้นการทำงานใน Startup code ของ Keil ก็อาจมีค่าไม่เหมาะสมกับระบบฮาร์ดแวร์ของบอร์ดที่เราจะใช้งานก็ได้ ซึ่งผู้ใช้มีทางเลือก 2 ทาง คือ สั่ง Add Startup Code จาก Keil ไปก่อน แล้วค่อยไปแก้ไขค่าต่างๆให้ถูกต้องและเหมาะสมกับระบบฮาร์ดแวร์ของบอร์ดในภายหลัง หรืออาจสั่งยกเลิก (เลือก “No”) แล้ว Copy ไฟล์ Startup code ที่กำหนดไว้สำหรับระบบฮาร์ดแวร์ที่ใช้งานอยู่ มาใช้ใน Project เอง แล้วไปสั่ง Add File ดังกล่าวเองในภายหลัง แต่เพื่อความสะดวกในการใช้งาน ในที่นี้แนะนำให้เลือก “Yes” ไปก่อน ดังรูป



รูปที่ 6 แสดงการ Add file Startup code

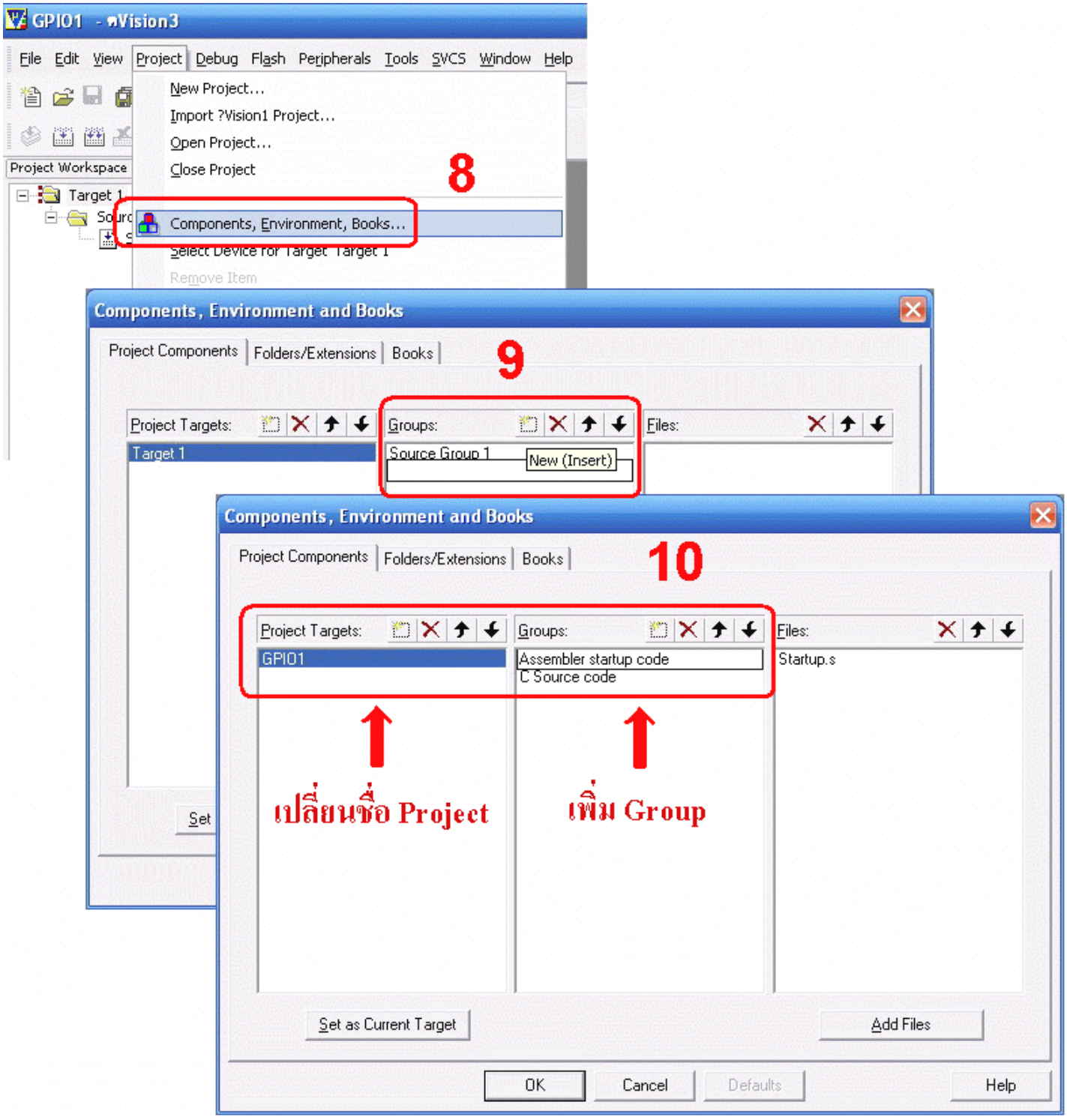
เมื่อทำการสั่ง Add file Startup code เสร็จเรียบร้อยแล้ว จะได้ผลลัพธ์ดังรูปที่ 7 (หมายเลข 7) ซึ่งโปรแกรมจะแสดงชื่อไฟล์ที่ Add ให้เห็น ใน Project Workspace



รูปที่ 6 แสดง Project Workspace

สร้าง Group ของ Project

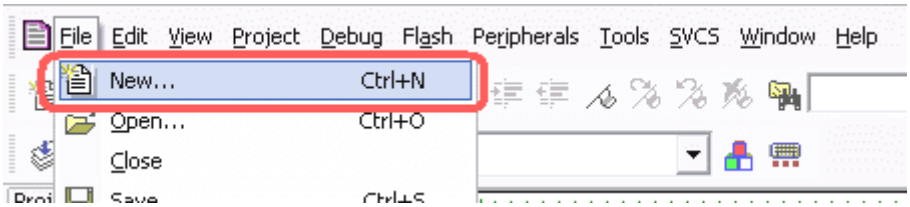
เมื่อสร้าง Project เรียบร้อยแล้ว จากนั้นก็ทำการสร้าง Group ซึ่งจะทำให้เป็นระเบียบ อาจจะแบ่งตามที่เราเข้าใจ ดังแสดงในรูปที่ 7 โดยไปที่ Project → Components, Environment, Books… (หมายเลข 8) เมื่อปรากฏหน้าต่างขึ้นมา ก็ให้เพิ่มกลุ่ม (หมายเลข 9) โดยตั้งชื่อตามความเหมาะสม (หมายเลข 10) หรือท่านสามารถเปลี่ยนชื่อ ลบ เพิ่ม Project Targets, Groups ได้เช่นกัน



รูปที่ 7 แสดงการสร้าง Group ของ Project

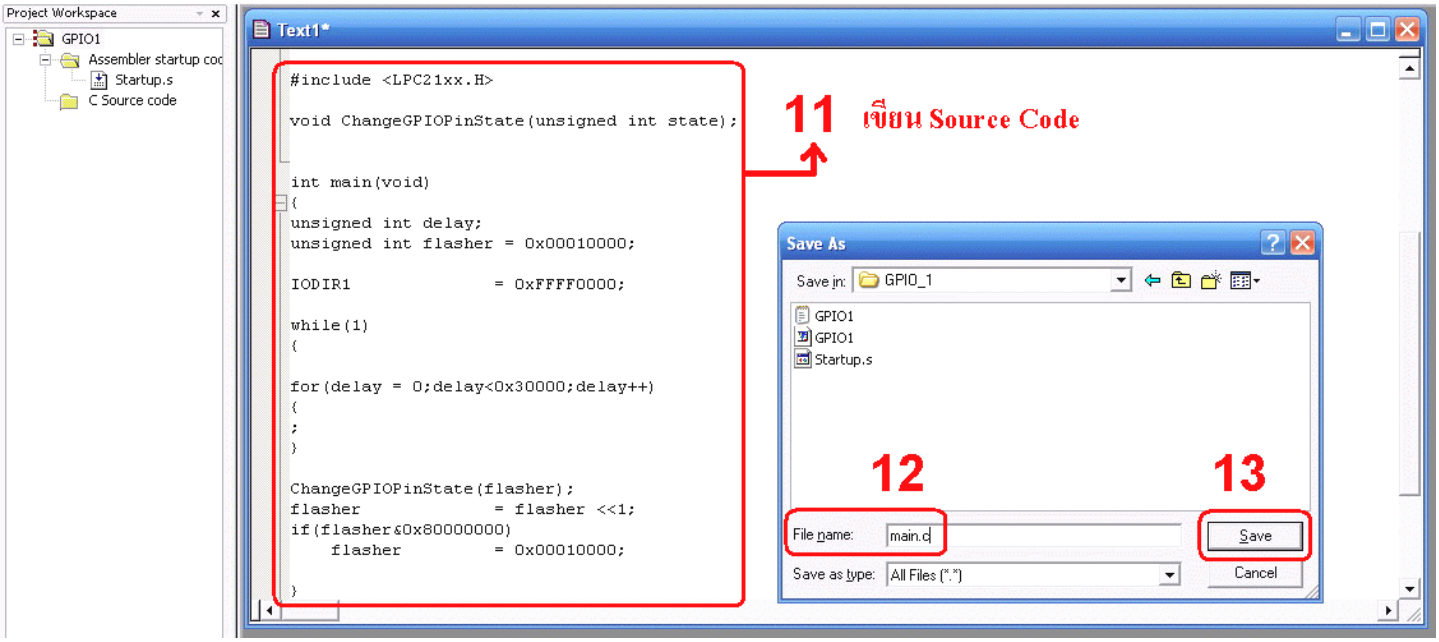
เขียน Source Code

เมื่อเราสร้าง Group แล้วก็ทำการเขียน Source Code ใน Text Editor ซึ่ง µVision3 ได้ตระเตรียมไว้เรียบร้อยแล้ว โดยไปที่ File → New… ดังแสดงในรูปที่ 8 ซึ่งเมื่อคลิกแล้วจะแสดงหน้าต่างขึ้นมาดังแสดงในรูปที่ 9



รูปที่ 8 แสดงการเลือก Text Editor ใหม่

จากรูปที่ 9 ให้ทำการเขียน Source Code ภาษา C ที่ Text Editor (หมายเลข 11) ดังตัวอย่าง Source Code ด้านล่างนี้ แล้วทำการ Save โดยไปที่ File → Save As… เลือกเก็บไว้ใน Directory ของ GPIO_1 การตั้งชื่อให้ตั้งตามความเหมาะสม ในที่นี้ตั้งเป็น main.c (หมายเลข 12) จากนั้นทำการ Save File ภาษา C ไว้ (หมายเลข 13)



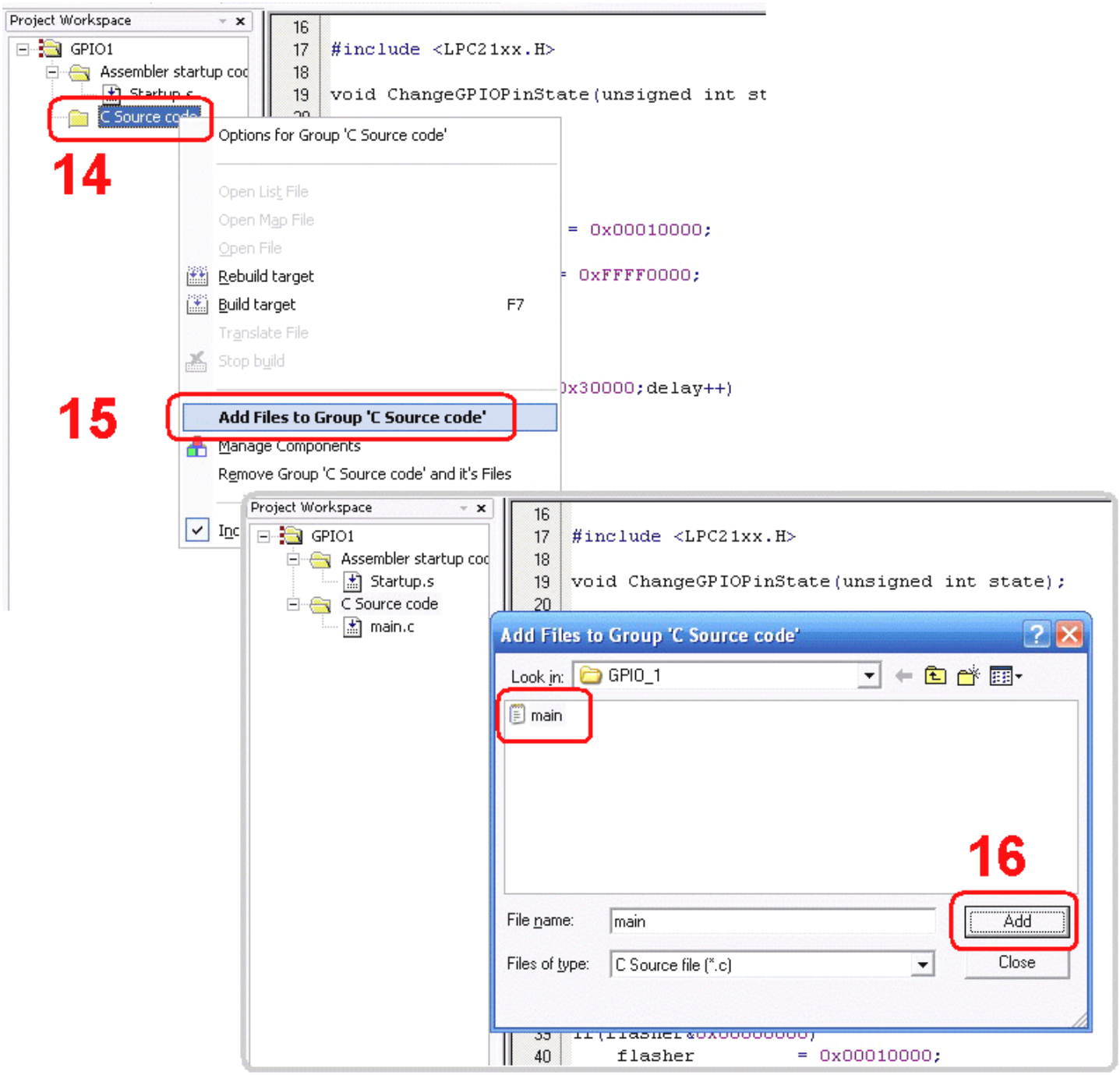
รูปที่ 9 แสดงตัวอย่างการเขียน Source Code

```
#include <LPC21xx.H>

void ChangeGPIOPinState(unsigned int state);


int main(void)
{
  unsigned int delay;
  unsigned int flasher = 0x00010000;          // define locals

  IODIR1             = 0x00FF0000;            // set P1.16…P1.23 = output

  while(1)
  {

    for(delay = 0;delay<0xA0000;delay++)      //simple delay loop
    {;;}

    ChangeGPIOPinState(flasher);              //set state of the ports
    flasher                = flasher <<1;     //shift the active led
    if(flasher&0x00800000)
    flasher          = 0x00010000;            //Increment flasher led

   }
}


void ChangeGPIOPinState(unsigned int state)
{
  IOCLR1        = ~state;                     //clear output pins
  IOSET1        =  state;                     //set output pins
}
```

ตัวอย่าง Source Code ภาษาC ไฟวิ่ง 8 ดวง (PIN P1.16-P1.23)

Add File to Group

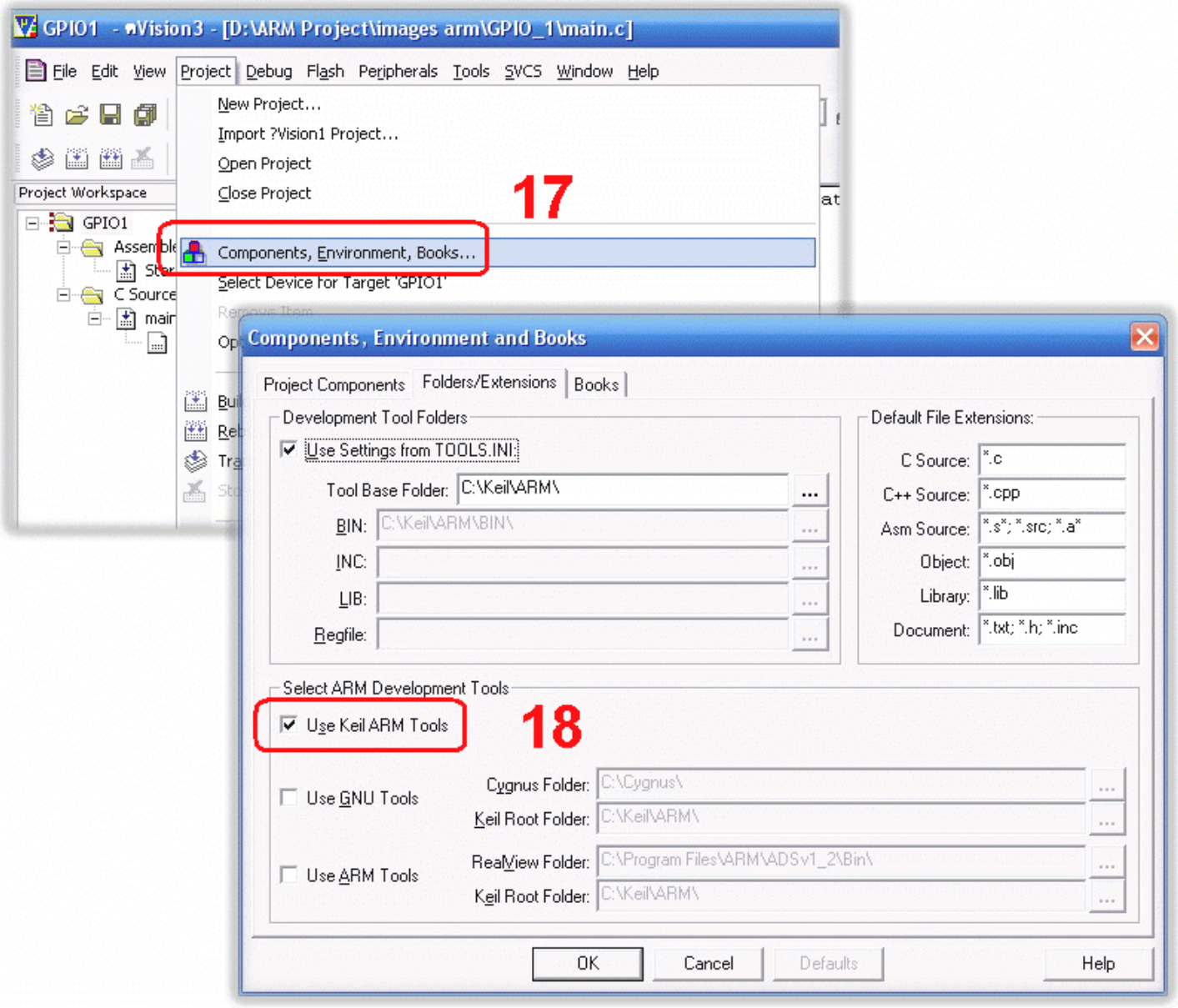
หลังจากสร้าง Group และ Source Code แล้ว ต้องทำการ Add file Source Code “main.c” เข้ามาที่ Group ดังแสดงในรูปที่ 10 ซึ่งมีขั้นตอนดังนี้ เลือก Group ที่ต้องการจะเพิ่มไฟล์เข้าไป คลิกขวาที่ Group (หมายเลข 14) ในที่นี้เลือก **C Source code → Add Files to Group ‘C Source code’** (หมายเลข 15) จากนั้นจะมีหน้าต่างแสดงขึ้นมา ให้ไปที่เราเก็บ main.c ไว้ แล้วคลิก Add (หมายเลข 16) โปรดสังเกตที่ Project Workspace จะมี “main.c” เพิ่มเข้ามาที่ Group ‘C Source code’



รูปที่ 10 แสดงการเพิ่มไฟล์ไปที่ Group

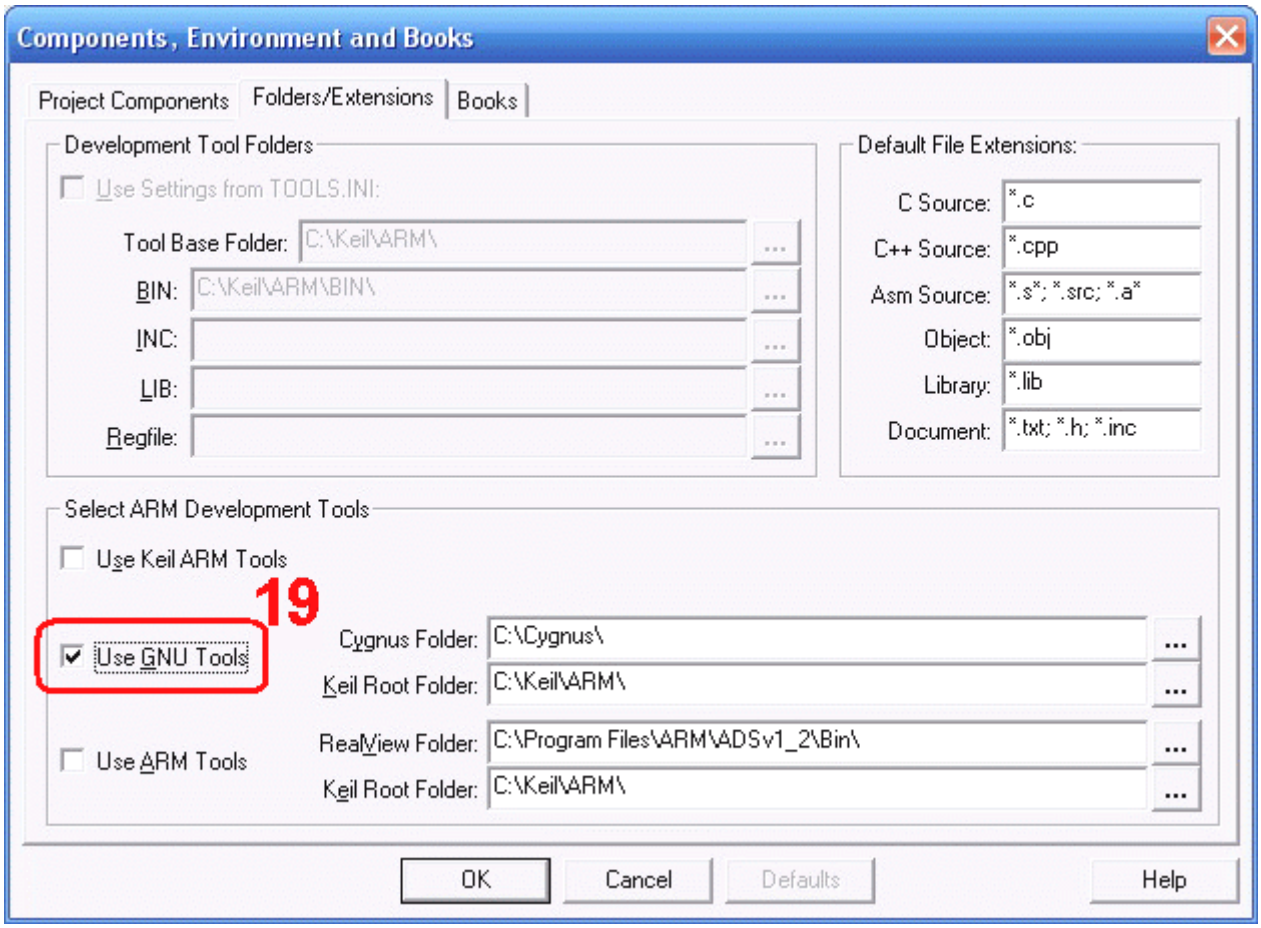
กำหนดรูปแบบการ Compile

การคอมไพล์ (Compile) นั้นสามารถเรียกใช้ Compiler ของ Keil เอง หรือให้ Link ไป Compile ด้วย GCC, ADS, หรือ RVDS ก็ได้ โดยการกำหนดค่าให้ไปที่ Project → Components, Environment, Books… (หมายเลข 17) ดังแสดงในรูปที่ 11 แล้วจะมีหน้าต่างแสดงขึ้นมา ให้เลือกที่แท็ป Folders/Extensions จะเห็นว่าค่าติดตั้งเดิมจะใช้ Keil ARM Tool (หมายเลข 18)



รูปที่ 11 แสดงการ Link กับ Compiler

ถ้าท่านต้องการ Compile ด้วย GCC ARM หรืออื่นๆ ให้ท่านเลือกที่ Use GNU Tools ดังแสดงในรูปที่ 12 (หมายเลข 19) โดยที่เราต้องติดตั้งโปรแกรมดังกล่าวด้วย สามารถดาวน์โหลดโปรแกรมดังกล่าวที่ www.keil.com/demo/eval/arm.htm เมื่อติดตั้งแล้วจะชื่อว่า Cygnus แล้วคลิก OK เพื่อยืนยันการเปลี่ยนแปลง



รูปที่ 12 แสดงการ Link ไป Compile กับ GCC ARM

หมายเหตุ

สำหรับตัวอย่างวิธีการพัฒนาโปรแกรมด้วย Keil CARM และ GCC ด้วย μVision3 แบบ Step-by-Step นั้น สามารถอ่านเพิ่มเติมได้จากเอกสารเรื่อง “ตัวอย่างการพัฒนาโปรแกรม ET-ARM STAMP LPC2119 ด้วย KEIL-ARM” และ “ตัวอย่างการพัฒนาโปรแกรม ET-ARM STAMP LPC2119 ด้วย GCCARM” ที่ทางอีทีที จัดทำไว้ได้

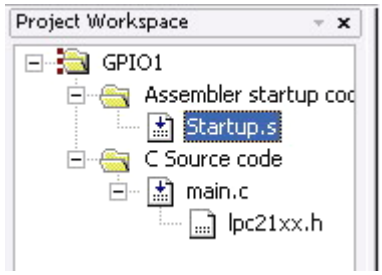
การกำหนดค่าให้กับ Startup File

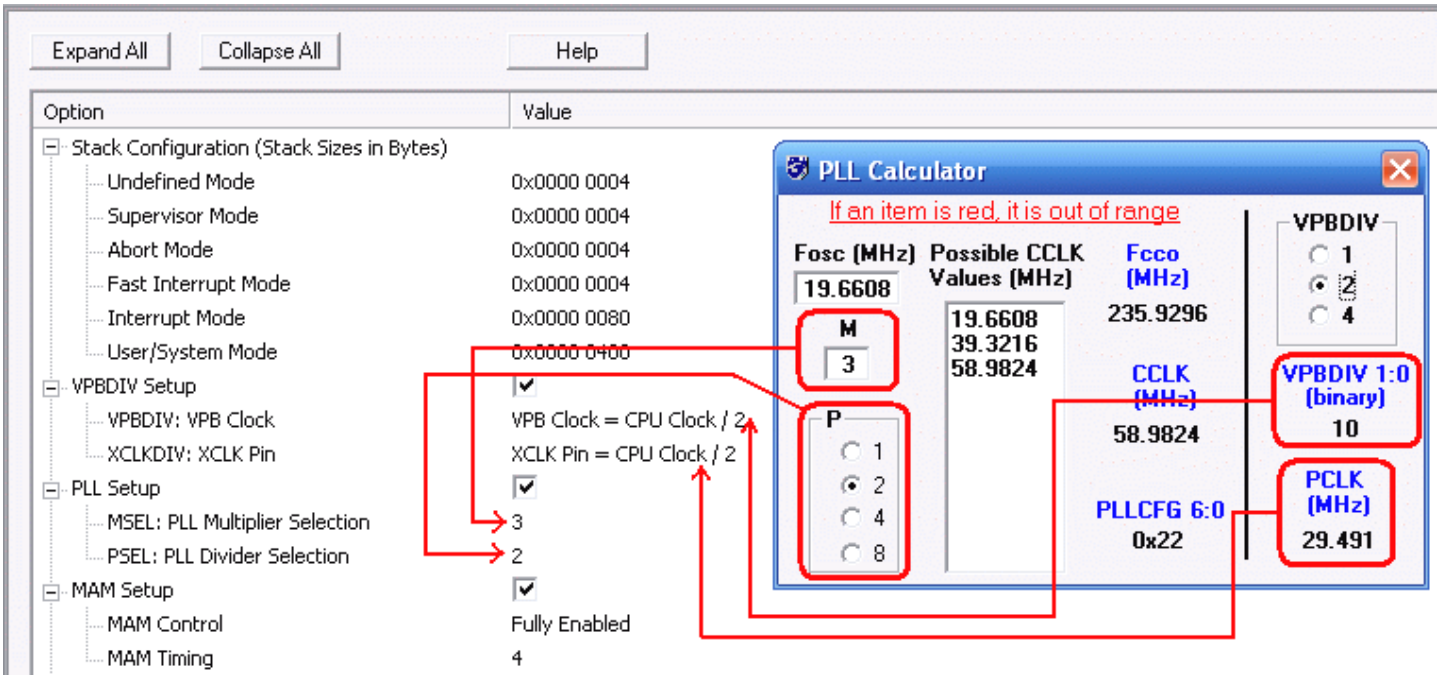
ดังได้กล่าว อธิบายมาแล้วในเรื่องการสั่ง Add Startup Code ไฟล์ (หมายเลข 6) ซึ่งในที่นี้จะขอแนะนำให้ทราบถึงวิธีการแก้ไขและกำหนดค่าต่างๆใน Startup ไฟล์จาก Keil เพื่อให้สามารถใช้งานกับ MCU เบอร์ LPC2119 ที่ใช้ในระบบฮาร์ดแวร์ของ ET-ARM STAMP LPC2119 ได้อย่างถูกต้อง ซึ่งเพื่อให้การใช้งานบอร์ดมีประสิทธิภาพสูงสุด ขอแนะนำให้ทำการกำหนดค่าตัวเลือกต่างๆเป็นดังนี้

- กำหนดค่าต่างๆให้กับวงจร Phase-Lock-Loop (PLL) เพื่อให้สามารถสร้างความถี่ของสัญญาณนาฬิกาให้กับระบบประมวลผลของ MCU (cclk) ให้ได้ค่า 58.9824 MHz โดยใช้ความถี่ XTAL = 19.6608 MHz ซึ่งค่าที่เหมาะสม ที่จะใช้ในการกำหนดค่าให้กับ Startup ในส่วนที่กำหนดการทำงานของ Phase-Lock-Loop ได้แก่ กำหนดค่า PLL Multiply (MSEL) = 3 และ กำหนดค่า PLL Divider (PSEL) = 2
- กำหนดค่าความถี่ของสัญญาณนาฬิกา สำหรับวงจร Pheri-Pheral อื่นๆ (pclk Clock) ให้มีค่าเป็นครึ่งหนึ่งของความถี่สัญญาณนาฬิกาของระบบประมวลผล (VPB Clock = CPU Clock / 2) ซึ่งจะทำให้ให้ค่าความถี่ของสัญญาณนาฬิกาที่จะใช้กับวงจร I/O ต่างๆใน MCU มีค่าเท่ากับ 29.4912 MHz
- กำหนดค่าการทำงานของ MAM Function ให้เป็น Full Enable และกำหนด Timing เป็น 3 หรือ 4

การตั้งค่าให้ Startup File

เรียกไฟล์ Startup.s ขึ้นมา ดังแสดงในรูป จะเห็นว่า Osc ของ ET-ARM STAMP LPC2119 คือ 19.6608 ดังนั้นเมื่อทำการคำนวณค่าความถี่ แล้วจะได้ดังรูป ซึ่งอาจคำนวณด้วยสูตร หรือจะใช้โปรแกรม PLL Calculator ก็ได้ แล้วนำค่านั้นๆ ดังในรูปมากำหนดที่ Startup.s ต่อไป ซึ่งตัวอย่างโปรแกรมของ ETT จะใช้ค่านี้เป็นค่าเริ่มต้น



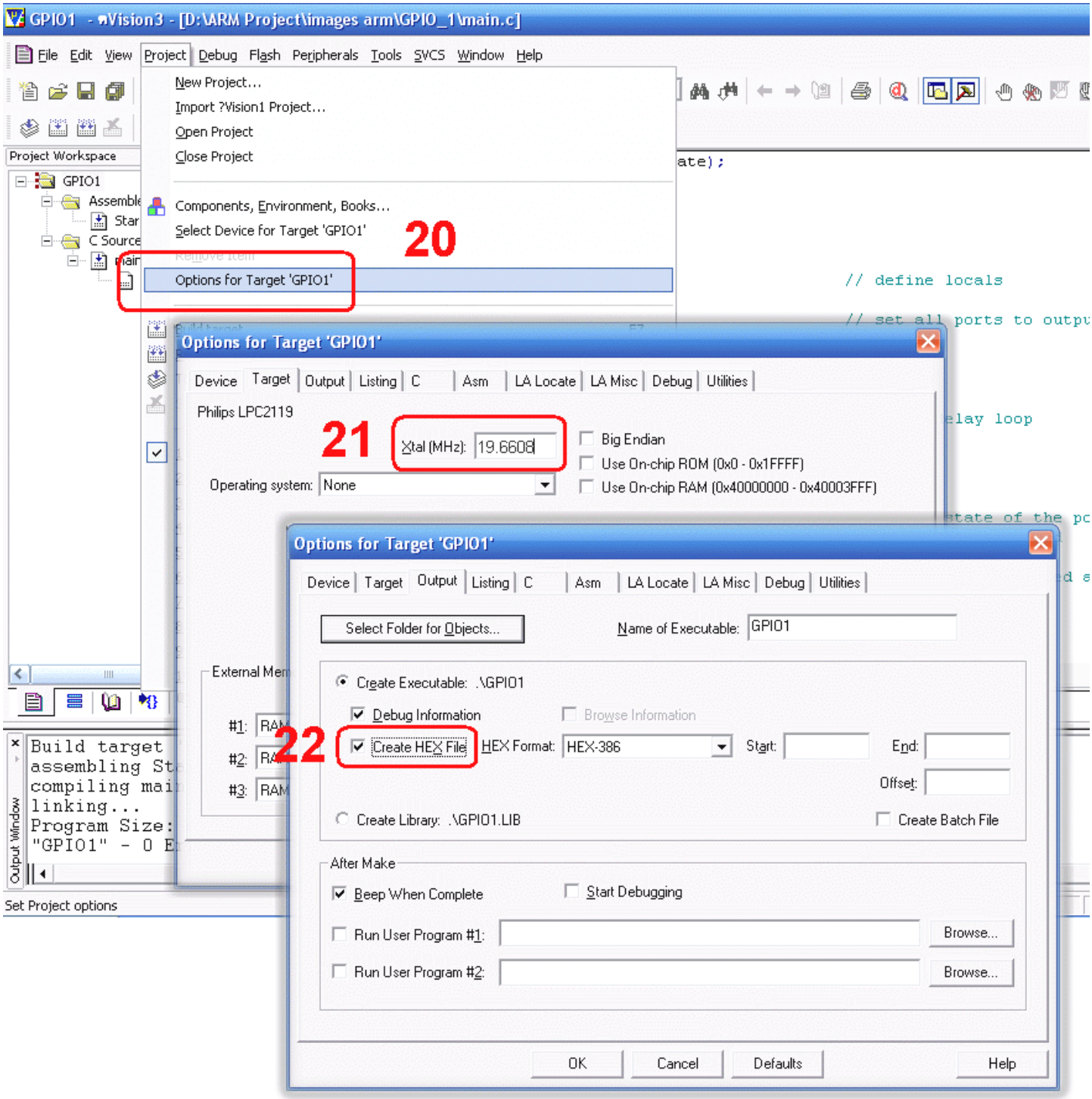


รูปแสดงการกำหนดค่าให้ Startup File

วิธีการ Compile Program

ก่อนอื่นต้องตั้งค่าต่างๆให้ตรงกับบอร์ดดังนี้ ไปที่ Project → Options for Target 'GPIO1' (หมายเลข 20) เมื่อมีหน้าต่างแสดงขึ้นมา ที่แท็ป Target ให้เปลี่ยนค่า Xtal (MHz) เท่ากับ 19.6608 (หมายเลข 21) ให้ตรงกับบอร์ดที่ใช้งาน แล้วคลิก OK

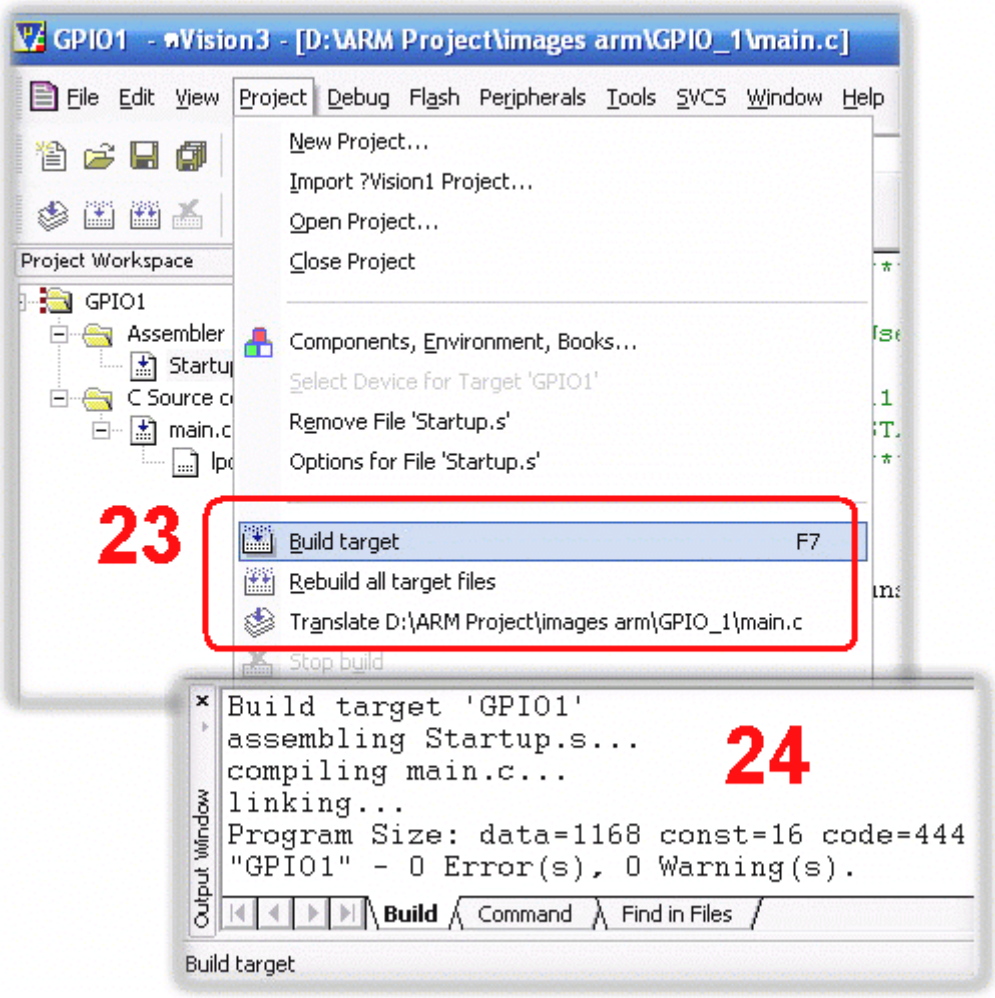
ต่อไปเลือกแท็ป Output ให้ทำการเลือกที่ Create HEX File (หมายเลข 22) เพื่อให้โปรแกรมสร้าง Hex File แล้วนำไปโหลดลงบอร์ดด้วยโปรแกรม LPC2000 Flash Utility ของ Philips ต่อไป จากนั้นคลิก OK



รูปที่ 13 แสดงการกำหนดค่าก่อน Compile

เมื่อกำหนดค่าให้กับบอร์ดแล้ว ก็ทำการ Compile โดยไปที่ Project → Translate... ก่อน จากนั้นรอสักครู่ แล้วจึงสั่ง Project → Build target หรือกด F7 (หมายเลข 23) ดังแสดงในรูปที่ 14

การ Compile ผ่านหรือไม่ผ่าน ให้ดูที่ Output Window (หมายเลข 24) ถ้ามี Error เกิดขึ้นก็จะแสดงที่หน้าต่างนี้ ให้คลิกที่ Error นั้นๆ µVision3 IDE จะแสดงจุดที่ผิดพลาดให้เราได้แก้ไข

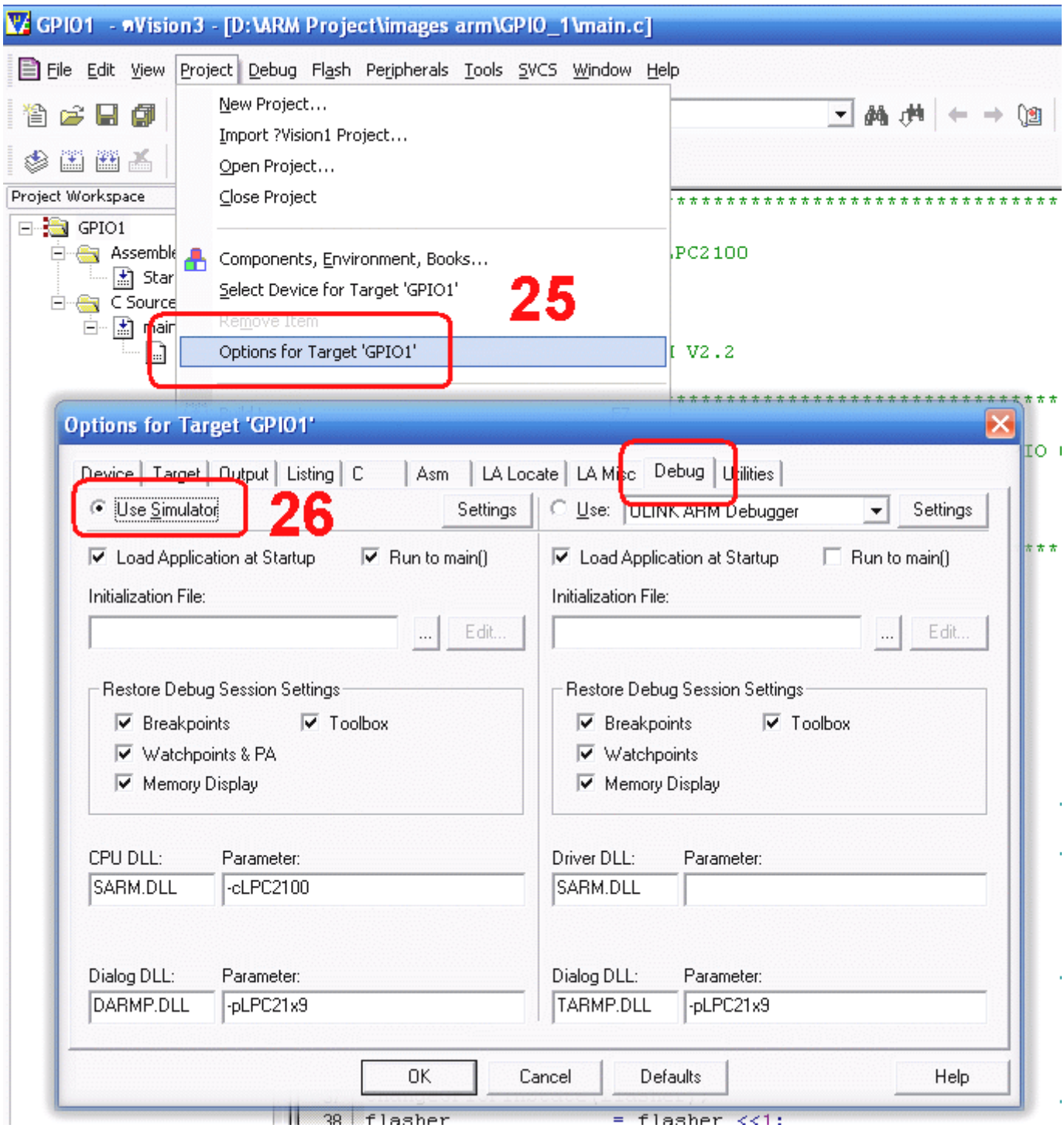


รูปที่ 14 แสดงการ Compile Program

การ Simulator โปรแกรม

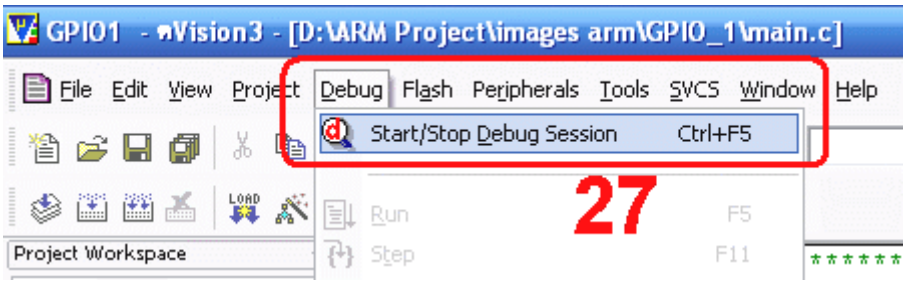
เราสามารถจำลองการทำงานของโปรแกรมที่เขียนขึ้นได้ ด้วยฟังก์ชั่น Simulator ของ µVision3 IDE เพื่อดูผลการทำงานของโปรแกรม ดังตัวอย่างของโปรแกรม อันดับแรกต้องไปตั้งค่าที่ Project → Options for Target 'GPIO1' (หมายเลข 25) เมื่อมีหน้าต่างแสดงขึ้นดังแสดงในรูปที่ 15 ให้เลือกไปที่ Debug → Use Simulator (หมายเลข 26) แล้วคลิก OK

อนึ่งในหน้าต่างที่กำลังแสดงขึ้นนี้ จะมีฟังก์ชั่นทางขวามืออีกอันคือ การ Debugger ด้วย Hardware ซึ่งสามารถเลือกใช้งานได้ถ้ามี ULINK ARM Debugger ของ Keil หรือ Hardware จากที่อื่น แต่ไม่ขอกล่าวถึงวิธีการใช้งานในที่นี้



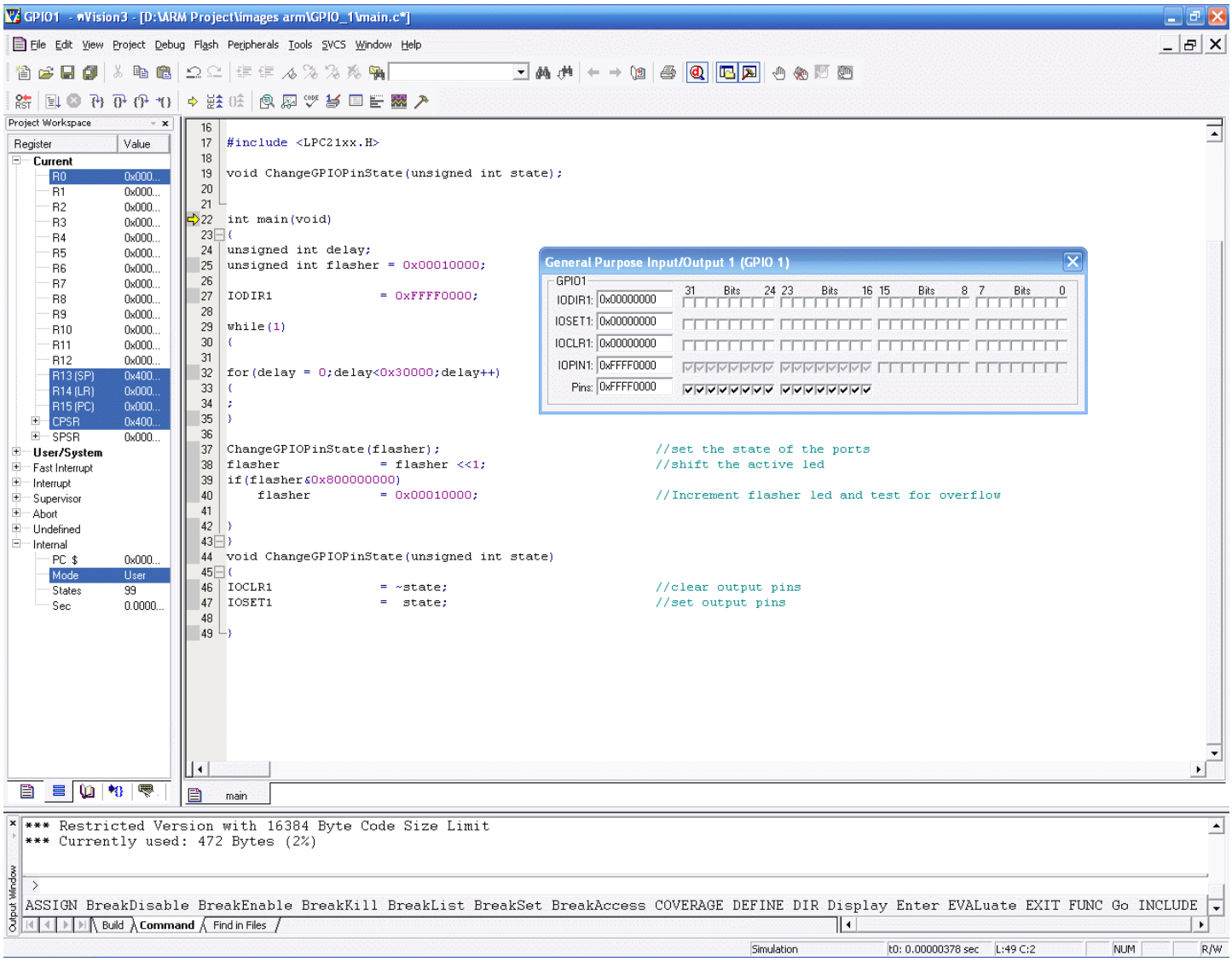
รูปที่ 15 แสดงการกำหนดค่าก่อน Simulator

เมื่อกำหนดค่าแล้วให้ไปที่ Debug → Start/Stop Debug Session ดังแสดงในรูปที่ 16 (หมายเลข 27) เพื่อเริ่มการทำงานของโหมด Simulator หรือ Debug



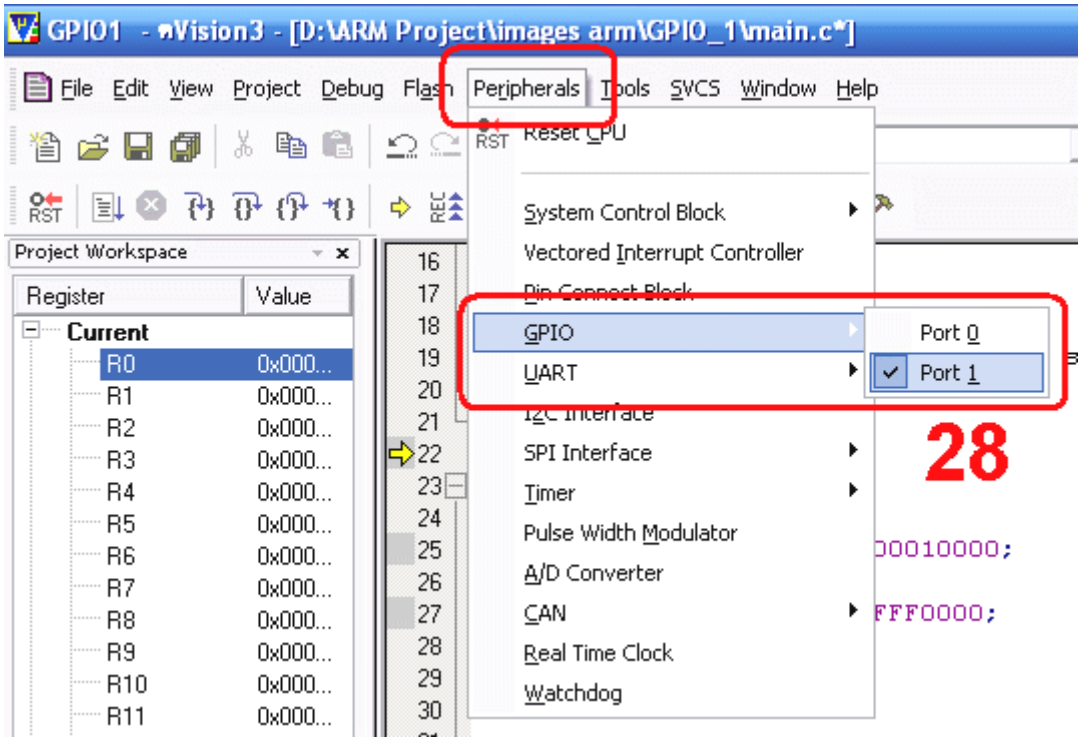
รูปที่ 16 แสดงการเริ่มต้นเข้าสู่ โหมด Simulator หรือ Debug

เรียบร้อยแล้วจะได้โปรแกรมในสถานะที่พร้อม Simulator ดังแสดงในรูปที่ 17



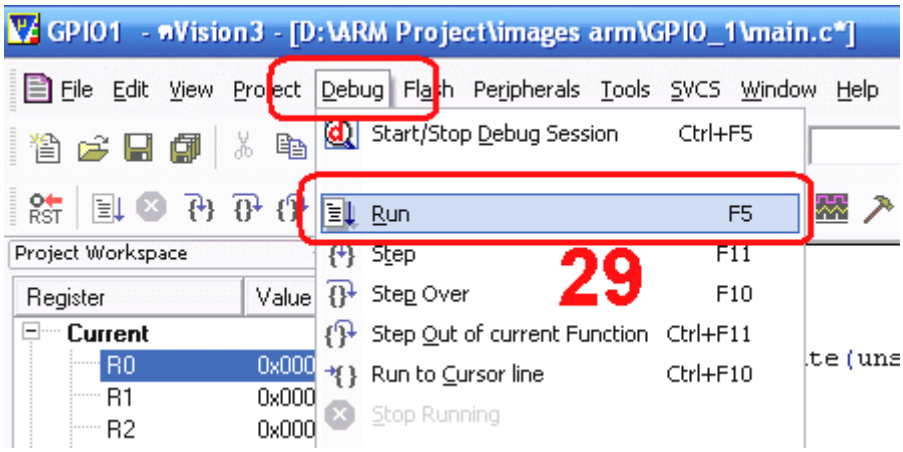
รูปที่ 17 แสดงสถานะเตรียมพร้อม Simulator

จากรูปที่ 17 ถ้าหน้าต่าง Peripherals ไม่ขึ้นให้ไปคลิกเลือกได้ที่ Peripherals ในตัวอย่างนี้จะใช้ Peripherals → GPIO → Port 1 (หมายเลข 28) ดังแสดงในรูปที่ 18



รูปที่ 18 แสดงการเลือก Peripherals I/O

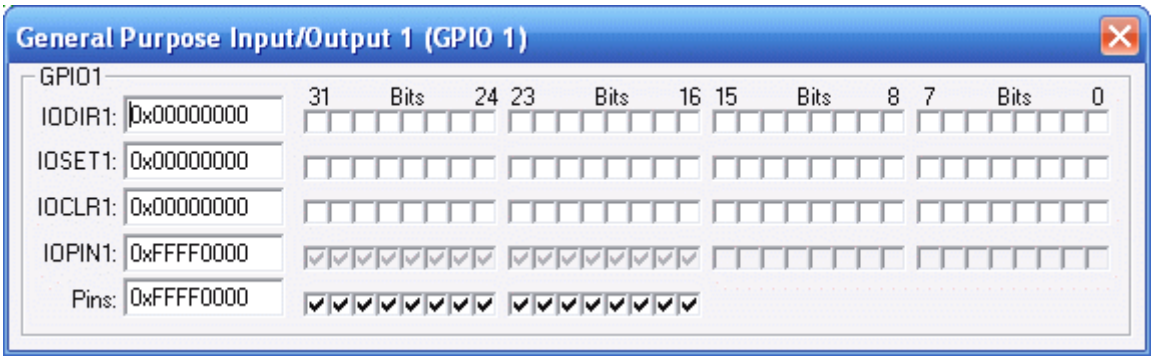
ตอนนี้เราก็สามารถ Run โปรแกรมได้แล้ว โดยไปที่ Debug → Run หรือกด F5 (หมายเลข 29) ดังแสดงในรูปที่ 19



รูปที่ 19 แสดงการ Run Simulator / Debug

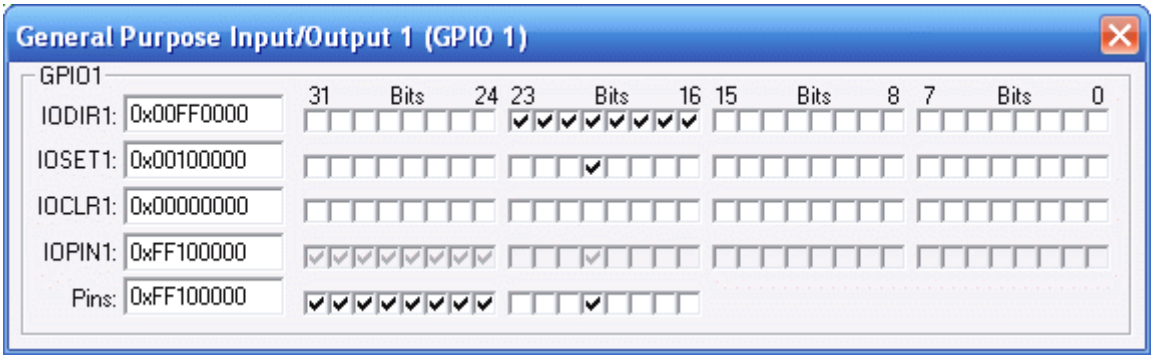
ผลการ Simulator โปรแกรม

จากรูปที่ 20 เป็นรูปแสดงก่อนการคลิก Run เพื่อ Simulator โปรแกรม จะอยู่ในสถานะเตรียมพร้อมจะทำงาน หน้าต่างนี้เปรียบเสมือนบอร์ดแสดงผล LED บอร์ดหนึ่ง ที่จะมีไฟติดไปเรื่อยๆทีละดวง ทีละบิต



รูปที่ 20 แสดงก่อนการ Run Simulator / Debug

เมื่อมาเปรียบเทียบกับรูปที่ 21 ที่ IODIR1 จะทำการ Set Bit ที่ P1.16-P1.23 เป็น Output ทั้งหมด จากนั้นที่ IOSET1 จะทำการ Set Bit ทีล่ะ Bit ไปเรื่อยๆ พร้อมกันนั้น IOCLR1 ก็จะทำการ Clear Bit หลังจาก Set bit จึงแลดูเหมือนโปรแกรมไฟวิ่ง เมื่อนำไปใช้งานกับบอร์ดจึงจะเห็นผลที่ชัดเจนขึ้น



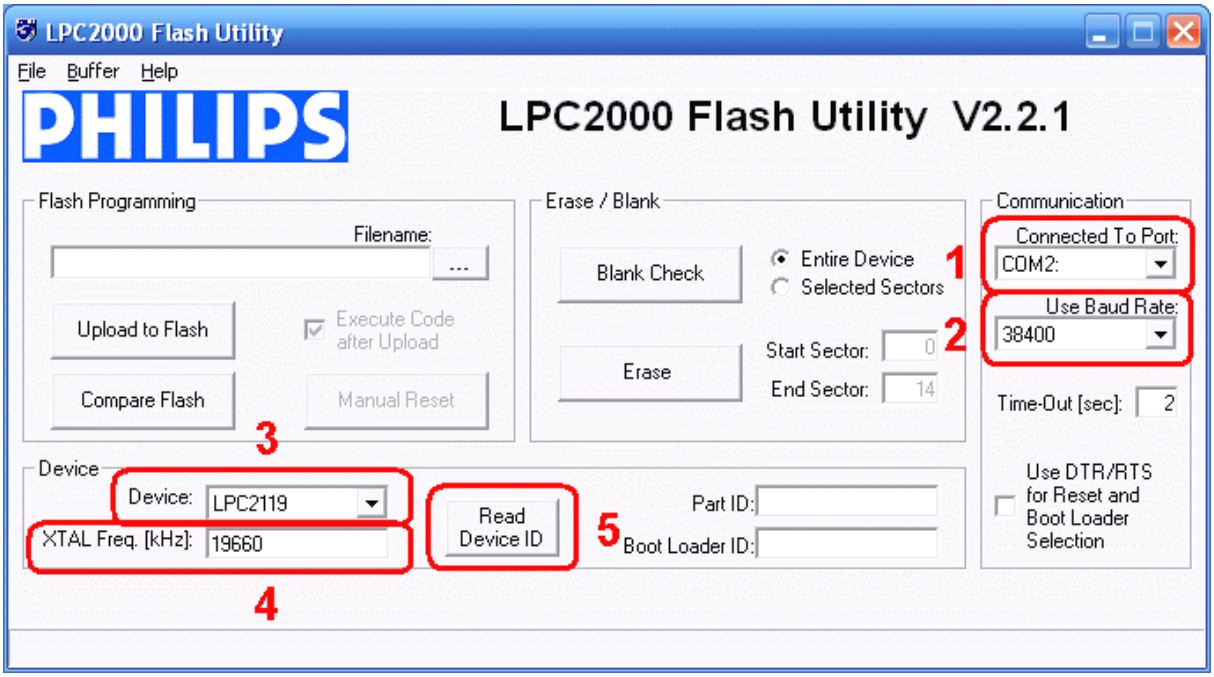
รูปที่ 21 แสดงผลลัพธ์หลังการ Run Simulator / Debug

การ Download Hex file ให้กับ MCU ของบอร์ด

การ Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ดนั้น จะใช้โปรแกรมชื่อ LPC2000 Flash Utility ของ Philips ซึ่งจะติดต่อกับ MCU ผ่าน Serial Port ของคอมพิวเตอร์ PC โดยโปรแกรมดังกล่าวสามารถดาวน์โหลดโปรแกรมได้ที่ www.semiconductors.philips.com

ขั้นตอนการ Download HEX File ให้กับ MCU

1. ต่อสายสัญญาณ RS232 ระหว่างพอร์ตสื่อสารอนุกรม RS232 ของ PC และบอร์ด (ET-RS232)
2. จ่ายไฟเลี้ยงวงจรขนาด +3.3V ให้กับบอร์ด ซึ่งจะสังเกตเห็น LED สีแดง (PWR) ติดสว่างให้เห็น
3. สั่ง Run โปรแกรม LPC2000 Flash Utility ของ Philips ซึ่งจะได้ผลดังรูป

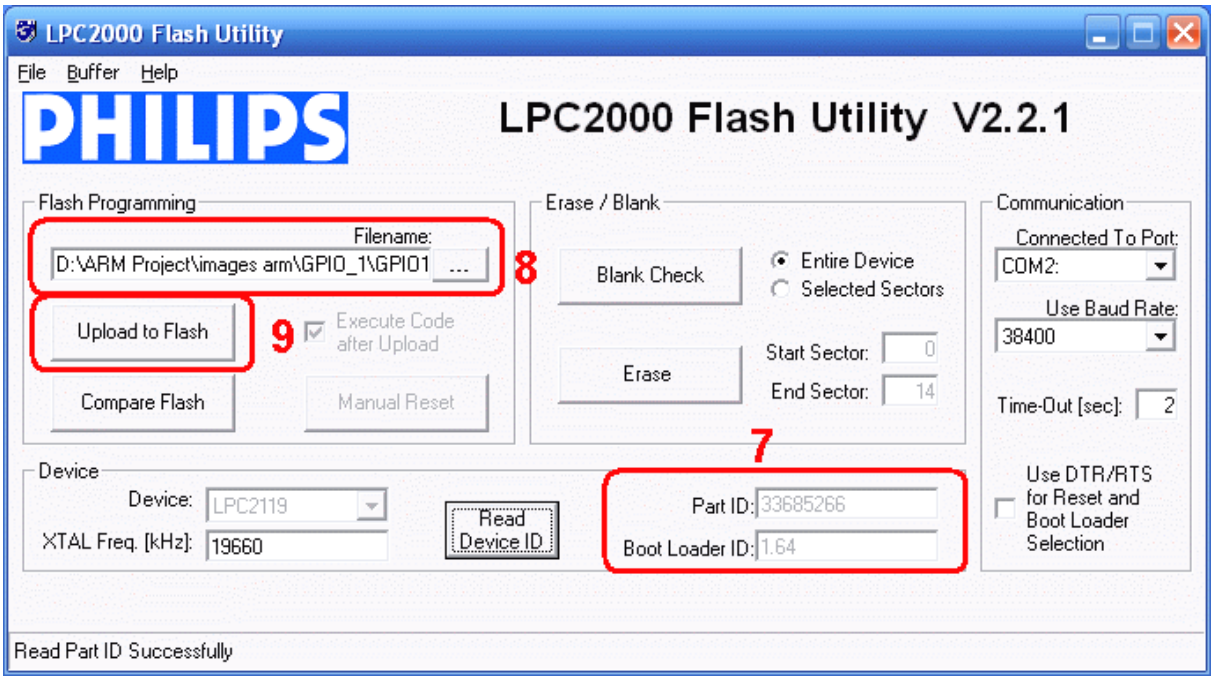


4. เริ่มต้นกำหนดค่าตัวเลือกต่างๆให้กับโปรแกรมตามต้องการ ซึ่งในกรณีที่ใช้กับ MCU เบอร์ LPC2119 ที่ใช้งานกับบอร์ดรุ่น ET-ARM STAMP LPC2119 ของ อีทีที ให้เลือกกำหนดค่าต่างๆให้โปรแกรมดังนี้
   1) เลือก COM Port ให้ตรงกับหมายเลข Com Port ที่ใช้งานจริง (ในตัวอย่าง COM2)
   2) ตั้งค่า Baud Rate อยู่ที่ระหว่าง 4800 – 38400 ซึ่งเป็นค่าที่ทดสอบแล้วใช้ได้โดยไม่เกิดปัญหา หรือใช้ค่าความเร็วมาตรฐานคือ 9600
   3) เลือกกำหนดเบอร์ MCU ในการติดต่อ ในที่นี้คือ LPC2119
   4) กำหนดค่าคริสตอล ออสซิลเลเตอร์ ให้ตรงกับที่ใช้ในจริงภายในบอร์ด โดยกำหนดให้มีหน่วยเป็น KHz และห้ามใส่ค่าเกิน 5 หลัก ในที่นี้ใช้ค่า 19.6608MHz ซึ่งเท่ากับ 19660
   5) คลิกเมาส์ที่ปุ่มคำสั่ง Read Device ID เพื่อติดต่อกับ CPU ซึ่งจะมีข้อความขึ้นมาเตือนให้ Reset การทำงานของ MCU ให้เข้าสู่ Boot Loader Mode ดังแสดงในรูป

6) ให้กดสวิตช์ RESET และ LOAD (BSL) ที่บอร์ด ET-ARM STAMP LPC2119 เพื่อทำการ Reset ให้ MCU ทำงานใน Boot Loader Mode ตามขั้นตอนดังต่อไปนี้

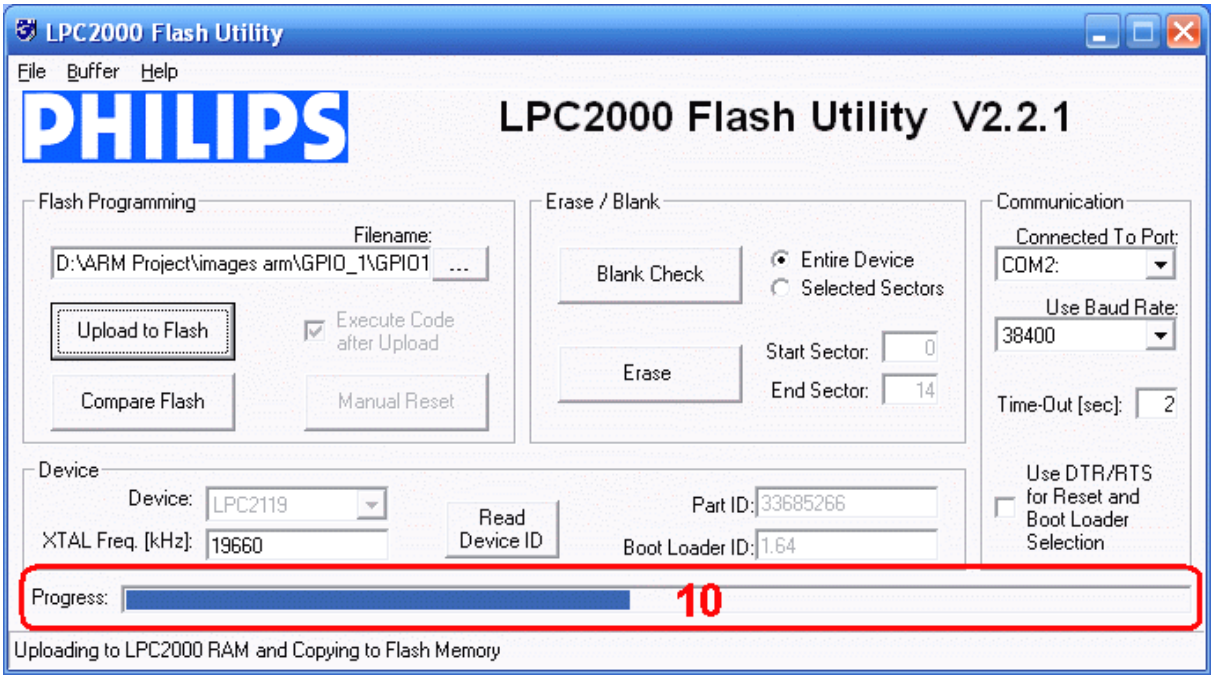
- กดสวิตช์ LOAD (BSL) ค้างไว้
- กดสวิตช์ RESET โดยที่สวิตช์ LOAD (BSL) ยังกดค้างอยู่
- ปล่อยสวิตช์ RESET โดยที่สวิตช์ LOAD (BSL) ยังกดค้างอยู่
- ปล่อยสวิตช์ LOAD (BSL) เป็นลำดับสุดท้าย เสร็จแล้วจึงคลิกเมาส์ที่ “OK.

7) เมื่อติดต่อกับ CPU ได้ จะปากฏรายละเอียด Part ID และ Boot Loader ID ดังรูป



8) ให้ทำการเลือกกำหนด HEX File ที่จะทำการสั่งโปรแกรม

9) ให้ทำการคลิกเมาส์ที่ “Upload to Flash” ซึ่งจะเห็นว่าโปรแกรม LPC2000 จะเริ่มต้นทำการ Download ข้อมูลให้กับ MCU ทันที โดยในขั้นตอนนี้ให้รอจนกว่าการทำงานของโปรแกรมจะเสร็จสมบูรณ์ ดังรูป



10) เมื่อการทำงานของโปรแกรมเสร็จเรียบร้อยแล้ว ให้กดสวิตช์ Reset ที่บอร์ด ซึ่ง MCU จะเริ่มต้นทำงานตามโปรแกรมที่สั่ง Download ให้ทันที

********** WWW.ETT.CO.TH **********

Referents

1. Keil Software, Inc

Download Evaluation Software : http://www.keil.com/demo/evaldl.asp

Limitations ARM Tools

Evaluation Kits are code-limited and have the following restrictions:

- You may not use the Evaluation Version of the µVision IDE/Debugger to create commercial products.
- The GNU ARM tools (compiler, assembler, and so on) that are provided are not limited or restricted in any way.
- The CARM compiler, assembler, and linker are limited to 16K Bytes of object code. Source code may be of any size.
- Programs that generate more than 16K Bytes of object code will not compile, assemble, or link.
- The debugger supports programs that are 16K Bytes or smaller.

2. GNU GCC ARM: www.gnuarm.org

3. Philips Semiconductor

LPC2000 Flash Utility

www.semiconductors.philips.com/files/products/standard/microcontrollers/utilities/lpc2000_flash_utility.zip

---

www.ett.co.th
www.etteam.com

บริษัท อีทีที จำกัด

1112/96-98 ถ. สุขุมวิท แขวงพระโขนง เขตคลองเตย กรุงเทพ 10110

ETT CO., LTD.

1112/96-98 Sukhumvit Rd., Phrakanong, Bangkok 10110 Thailand

Tel. : (66)02-7121120 , FAX. : (66)02-3917216
