

ตัวอย่างการเชื่อมต่อ ET-MINI MP3 ด้วย CP-JR ARM7 LPC2138/LPC2148 จาก อีทีที

สำหรับตัวอย่างนี้จะแสดงให้เห็นการใช้พอร์ต SPI ของ LPC2138/LPC2148 ในการเชื่อมต่อสัญญาณไอซี VS1002D ซึ่งเป็น MP3 Decoder โดยตัวอย่างโปรแกรมจะมุ่งเน้นให้ผู้ใช้งานเข้ากับการติดต่อกับและสัญญาณไอซี VS1002D เพื่อเล่นเพลงหรือเสียงเท่านั้น ส่วนในเรื่องของการจัดการกับระบบไฟล์แบบต่างๆ นั้นจะไม่ได้กล่าวถึงในที่นี้ด้วย ซึ่งผู้ใช้งานจำเป็นต้องไปค้นคว้าหากรรมวิธีในการที่จะอ่านไฟล์ข้อมูล มาส่งให้กับ MP3 Decoder เพื่อแปลงกลับเป็นเสียงเอง เช่น การเลือกใช้นหน่วยความจำแบบ SD/MMC Memory หรือหน่วยความจำแบบอื่นๆ

โดยในตัวอย่างนี้จะใช้วิธีการนำไฟล์แบบ Wave ขนาดเล็กๆ ซึ่งมีความยาวไฟล์จะไม่เกิน 16KB จำนวน 11 ไฟล์ โดยเป็นเสียงประโยคการนับเลข “0” ถึง “10” มาเก็บไว้ในหน่วยความจำ Flash ของ LPC2138/LPC2148 จากนั้นก็จะทำการอ้างตำแหน่งของหน่วยความจำที่ใส่เก็บไฟล์ไว้แล้วส่งให้ MP3 Decoder ทำการแปลงเป็นเสียงให้ ซึ่งในครั้งแรกผู้เขียนตั้งใจจะนำไฟล์มาจัดเรียงเป็นตารางแบบ Array รวมไว้ใน Code โปรแกรมด้วย เพื่อความสะดวกในการเรียกใช้งาน แต่เนื่องจากว่า โปรแกรมภาษาซี (Keil-CARM) ที่ใช้สำหรับ Compiler นั้นเป็นรุ่นทดลองใช้ ซึ่งมีข้อจำกัดไม่สามารถ Compiler Source Code ที่มีขนาดเกิน 16 KByte ได้ ดังนั้นจึงจำเป็นต้องใช้วิธีการทางอ้อมในการแก้ปัญหาที่ไปก่อน โดยแยกส่วนของ Code Program และ ส่วนของไฟล์เสียง ออกจากกันแล้ว เพื่อให้ไฟล์มีขนาดไม่เกิน 16KByte จากนั้นจึงนำไฟล์ที่ส่งแปลเป็น HEX แล้วมารวมกันในภายหลัง โดยโครงสร้างการจัดแบ่งพื้นที่ของหน่วยความจำสำหรับเก็บ Code Program และ Code ของไฟล์เสียง กำหนดไว้เป็นดังนี้

ตำแหน่งของหน่วยความจำ(Code)	การใช้งาน
0x00000 - 0x03FFF (16KB)	Monitor Code Program
0x04000 - 0x07FFF (16KB)	เสียง “Zero” (0f.wav)
0x08000 - 0x0BFFF (16KB)	เสียง “One” (1f.wav)
0x0C000 - 0x0FFFF (16KB)	เสียง “Two” (2f.wav)
0x10000 - 0x13FFF (16KB)	เสียง “Three” (3f.wav)
0x14000 - 0x17FFF (16KB)	เสียง “Four” (4f.wav)
0x18000 - 0x1BFFF (16KB)	เสียง “Five” (5f.wav)
0x1C000 - 0x1FFFF (16KB)	เสียง “Six” (6f.wav)
0x20000 - 0x23FFF (16KB)	เสียง “Seven” (7f.wav)
0x24000 - 0x27FFF (16KB)	เสียง “Eight” (8f.wav)
0x28000 - 0x2BFFF (16KB)	เสียง “Nine” (9f.wav)
0x2C000 - 0x2FFFF (16KB)	เสียง “Ten” (10f.wav)

ตารางแสดง การจัดพื้นที่หน่วยความจำของ LPC2138/LPC2148 ที่ใช้ในตัวอย่าง

สำหรับวิธีการแปลงไฟล์เสียงให้เป็น HEX นั้น สามารถทำได้หลายวิธีการ เช่น อาจใช้โปรแกรม Utility ของเครื่องโปรแกรม EPROM ต่างๆ โดยสั่ง Open File ของเสียงขึ้นมาในรูปแบบของ Binary File แล้วเก็บไว้ใน Buffer ของโปรแกรมจากนั้นก็สั่ง Save กลับในรูปแบบของ HEX File โดยกำหนดตำแหน่ง Offset เริ่มต้นในการจัดเก็บให้ เริ่มต้นที่ 0x4000 เป็นต้นไป ดังที่แสดงไว้ในตารางข้างต้น แต่ในที่นี้จะขอใช้ โปรแกรม Utility ชื่อ "BIN2HEX" ซึ่งผู้เขียนได้ Download มาจาก Website ของ Keil โดยสามารถ Download มาใช้ได้ฟรี ซึ่ง BIN2HEX เป็นโปรแกรมที่มีการทำงานแบบ Command Line โดยวิธีการใช้งานสามารถศึกษาได้จาก Help ของโปรแกรม โดยเพื่อสั่ง Run โปรแกรมที่ Dos Prompt โปรแกรมจะแสดงวิธีการใช้งานให้เห็น แต่อย่างไรก็ตามเพื่อความสะดวกในการทดลอง ผู้เขียนได้สร้าง Batch File ชื่อ "VOICE.BAT" สำหรับสั่งแปลงไฟล์เสียงให้เป็น HEX File ตามตำแหน่งแอดเดรสที่กำหนดไว้ดังตารางข้างต้น เตรียมไว้ให้ด้วยแล้ว โดยเก็บอยู่ใน Folder ชื่อ "AudioData" ซึ่ง Folder ดังกล่าวจะซ่อนอยู่ภายใน Folder ของ Source Code ภาษาซี อีกชั้นหนึ่ง โดย Batch File ดังกล่าว มีรายละเอียดดังนี้

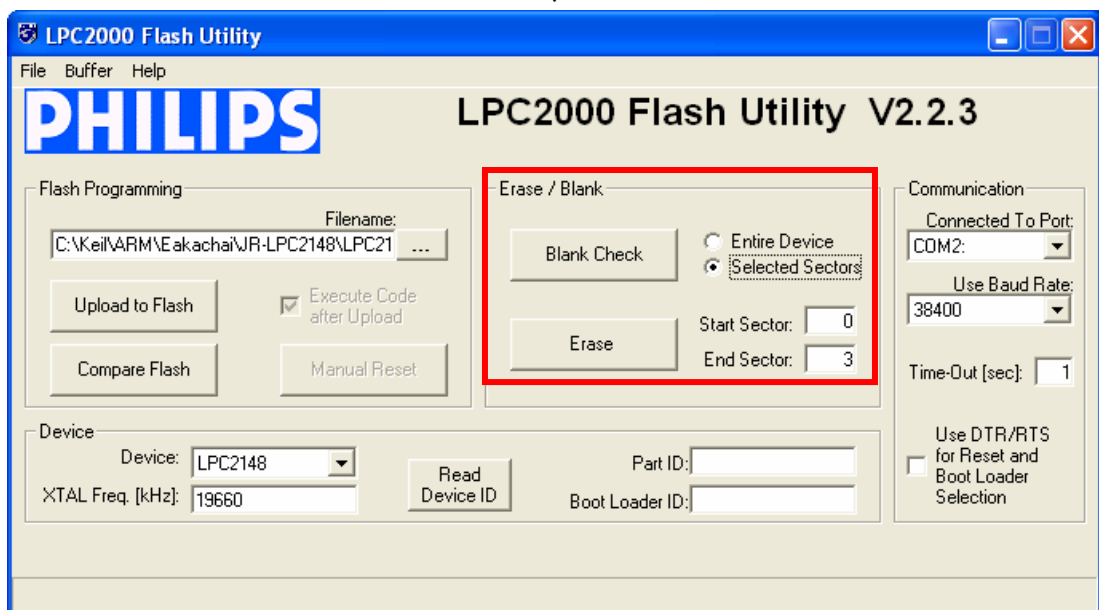
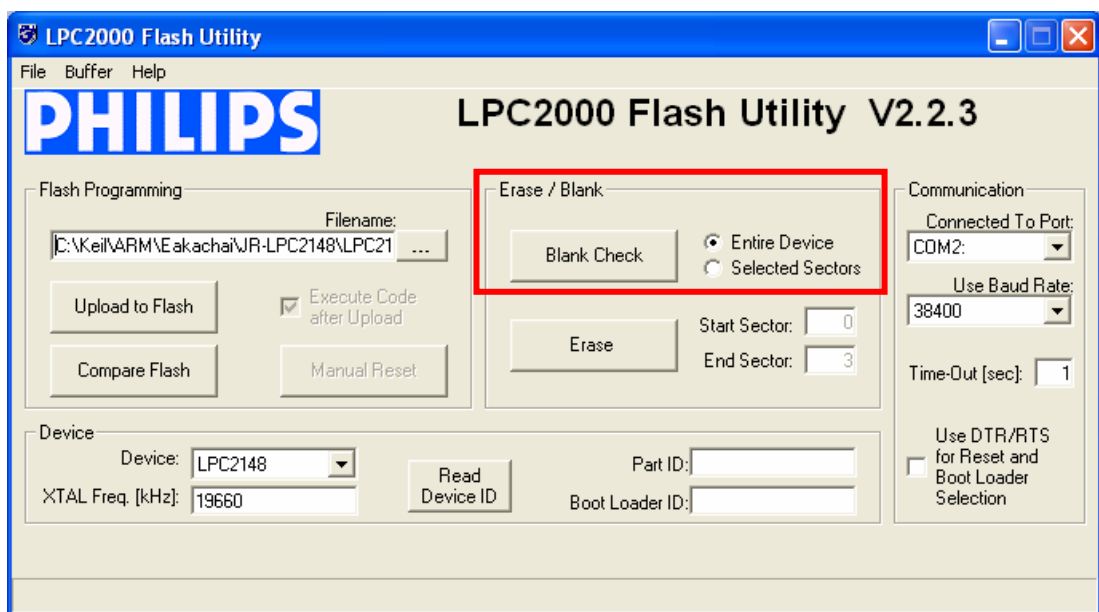
```
ECHO OFF
ECHO Generating VOICE.HEX with Wave Files...
DEL VOICE.HEX
BIN2HEX /L16384 /O16384 /4 /T /Q 0f.wav VOICE.HEX
BIN2HEX /L16384 /O32768 /4 /A /T /Q 1f.wav VOICE.HEX
BIN2HEX /L16384 /O49152 /4 /A /T /Q 2f.wav VOICE.HEX
BIN2HEX /L16384 /O65536 /4 /A /T /Q 3f.wav VOICE.HEX
BIN2HEX /L16384 /O81920 /4 /A /T /Q 4f.wav VOICE.HEX
BIN2HEX /L16384 /O98304 /4 /A /T /Q 5f.wav VOICE.HEX
BIN2HEX /L16384 /O114688 /4 /A /T /Q 6f.wav VOICE.HEX
BIN2HEX /L16384 /O131072 /4 /A /T /Q 7f.wav VOICE.HEX
BIN2HEX /L16384 /O147456 /4 /A /T /Q 8f.wav VOICE.HEX
BIN2HEX /L16384 /O163840 /4 /A /T /Q 9f.wav VOICE.HEX
BIN2HEX /L16384 /O180224 /4 /A /Q 10f.wav VOICE.HEX
```

แสดง คำสั่งใน Batch File เพื่อสั่งแปลงไฟล์เสียงให้เป็น HEX

โดยเมื่อเรียกใช้ Batch File จะทำให้เราได้ไฟล์ ชื่อ "VOICE.HEX" ซึ่งเป็นส่วนของไฟล์เสียง จำนวน 11 ไฟล์จัดเรียงกันไว้ในหน่วยความจำไฟล์ละ 16 KB ตามลำดับ ดังได้แสดงไว้ในตารางข้างต้น

ซึ่งหลังจากที่เราได้ทั้ง HEX File ในส่วนที่เป็น Code Program (ได้จากการแปลของ Keil-CARM) และส่วน
ของ HEX File ที่เป็นของเสียงที่ได้จากการแปลของ Batch File (VOICE.BAT) แล้วในขั้นตอนถัดไปก็คือ การ
Download HEX File ทั้ง 2 ไฟล์ให้กับ CPU โดยใช้โปรแกรม LPC2000 ของ Philips เป็นตัวจัดการให้ โดยใน
ขั้นตอนนี้สามารถทำได้ 2 วิธีการคือ

1. สั่ง Download HEX File ทีละไฟล์ โดยเมื่อสั่งเปิด HEX File ที่เป็นของ ไฟล์เสียงให้เลือกรูปแบบของ
การลบหน่วยความจำเป็น Enter Device ซึ่งเป็นการสั่งลบหน่วยความจำทั้งหมดแล้วจึงสั่ง Download
ไฟล์เสียงก่อน จากนั้นให้สั่งเปิด HEX File ที่เป็นของ Source Code แต่ให้เลือกรูปแบบของการลบเป็น
แบบ Select Sectors โดยให้กำหนดตำแหน่ง Sector เป็น 0..3 แล้วสั่ง Download ดังรูป



2. ใช้วิธีการนำ HEX File ทั้ง 2 ไฟล์รวมกันเป็น FILE เดียว แล้วสั่ง Download เพียงครั้งเดียว ซึ่งสามารถทำได้ดังนี้

- a. สั่งเปิดไฟล์ HEX ที่เป็นของ Source Code ที่ได้จากการแปลของ Keil-CARM ด้วยโปรแกรม Text Editor เช่น Notepad จากนั้นให้ลบ บรรทัด สุดท้ายซึ่งเป็นตัวบ่งบอกการจบไฟล์ของ Intel HEX (End of HEX Record) ออก โดยบรรทัดดังกล่าวจะมีลักษณะดังนี้

```
:00000001FF
```

- b. สั่งเปิดไฟล์ HEX ที่เป็นของเสียงที่ได้จากการแปลของ Batch File (VOICE.BAT) แล้วทำการ Copy ข้อมูลทั้งหมดใน HEX File ไปต่อท้าย HEX File ของ Source Code แล้วสั่งบันทึก HEX File ที่เป็น Source Code ที่ได้รวมไฟล์แล้วไว้
- c. สั่ง Download HEX File ที่ทำการรวมไฟล์แล้วให้กับบอร์ด

****หมายเหตุ**** สำหรับ HEX File ที่ได้ทำการรวมไฟล์ไว้แล้วจะชื่อ "LPC2148_MP3_PLAY_WAVE.HEX" และ "LPC2138_MP3_PLAY_WAVE.HEX" โดยเก็บอยู่ใน Folder ชื่อ FINAL_HEX_TEST ซึ่ง Folder ดังกล่าวจะซ่อนอยู่ภายใน Folder ของ Source Code ภาษาซี อีกชั้นหนึ่ง ส่วน Hex File ที่อยู่ใน Folder เดียวกันกับ Source Code นั้นเป็น Hex File ของ Source Code ที่ยังไม่ได้รวมกับ Hex File ของเสียงเข้าไปด้วย โดยผู้ใช้สามารถนำ Hex File ใน Folder ชื่อ FINAL_HEX_TEST ไปทำการ Download เพื่อทดลองได้ทันที โดยถ้าทุกอย่างถูกต้อง หลังจาก Download เสร็จแล้ว เมื่อกดรีเซ็ตและต่อชุดหูฟังหรือชุดขยายเสียงของ คอมพิวเตอร์ PC เป็นที่เรียบร้อยแล้ว จะมีเสียงนับเลขเป็นภาษาอังกฤษ จาก ศูนย์ ถึง สิบ วนรอบไม่รู้จบให้ได้ยินอยู่ตลอดเวลา

การเชื่อมต่อสัญญาณระหว่างบอร์ด

บอร์ด CP-JR ARM7 LPC2138 CP-JR ARM7 USB-LPC2148		บอร์ด ET-MINI MP3 (VS1002D)
[+3V3]	↔	[+3V3]
[GPIO0.2 (Input)]	←	[DREQ]
[GPIO0.3 (Output)]	→	[RES#]
[GPIO0.4 (SPI0-SCLK)]	→	[SCLK]
[GPIO0.5 (SPI0-MISO)]	←	[SO]
[GPIO0.6 (SPI0-MOSI)]	→	[SI]
[GPIO0.7 (Output)]	→	[XCS#]
[GPIO0.8 (Output)]	→	[BSYNC]
[GND]	↔	[GND]

ตาราง แสดงทิศทางและสัญญาณการเชื่อมต่อระหว่าง LPC2138/LPC2148 กับ MP3 Decoder