آموزش پروگرم کردن میکروکنترلر LPC1768 با استفاده از بوت لودر و از طریق USB



همانطور که می دانیم به صورت پیش فرض کد خروجی کامپایلر Keil به صورت هگز می باشد در صورتیکه برای پروگرم کردن میکروکنترلر از طریق USB ، باید از کد باینری استفاده کنیم. بنابراین ابتدا به این نکته می پردازیم که چگونه می توان با استفاده از Keil کد باینری نیز ایجاد کرد.

: MDK Keil uVision نحوه ایجاد کد باینری توسط

به صورت پیش فرض Keil uVision نمی تواند کد باینری سازگار با بوت لودر USB ایجاد کند. به منظور ایجاد کد باینری سازگار با بوت لودر از یک خط دستور خارجی استفاده می کنیم. که به صورت زیر عمل می کنیم :

- ۱- در پنجره "Target Options" تب "User" را انتخاب کنید.
- ۲- در این پنجره در قسمت "Run User Programm After Build/Rebuild" تیک مربوط به "Run #1" را قرار در این پنجره در قسمت "Run #1" را قرار در این پنجره در قسمت "
 - ۳- دستور fromelf را به صورت زیر در کادر مقابل ":Run #1" وارد کنید.

fromelf --bin ".\test.axf" --output ".\test.bin"

مطابق با شکل زیر :

| C:\Users\ramin\Desktop\LCD3.2-LPC1768\test.uvproj - µVision4 | | _ _ X |
|---|---|-----------------|
| File Edit View Project Flash Debug Peripherals Tools SVCS Window | Help | |
| □ 😂 🖬 🗿 👗 🛍 🖄 🗢 🔍 🖛 → 🥐 際 務 務 (建 連) | //= //z 🖄 💌 🖳 🔹 🖉 🍳 🕒 🗢 🔗 🍓 🔳 🗨 | |
| 🔮 🖺 🕮 🥪 拱 🙀 Target 1 💽 💦 🛔 🗟 | | |
| Project 4 🛛 🔛 main.c | | ▼ × |
| B → Target1 1 #include <lpc17xx.h></lpc17xx.h> | 🗑 Options for Target 1' | Â |
| - | Device Target Output Listing User C/C++ Asm Linker Debug Utilities | |
| formal 6 #define LCD OFFSET 15 formal formation in the state of the formation of the state of the st | Run User Programs Before Compilation of a C/C++ File | |
| - core_cminstr. 9 #define TFTLCD_RS 1 - core_cmfunc 10 #define TFTLCD_WR 2 | Stop Build/Rebuild #1 on Exit Code: Not Specified | = |
| | Stop Build/Rebuild #2 on Exit Code: Not Specified | |
| - ☐ font_24x16.h 14 - ☐ inages.h 15 ⊟ void Delay (void) (- ☐ images.h 16 int i; | Run User Programs Before Build/Rebuild | |
| B- system_LPCI7xx. 17 for(i=0;i<2000000;i++); 18 } 19 ↓ | □ Run #2: | |
| 20 = int main() { 21 | Run User Programs After Build/Rebuild | |
| 22 SystemInit(); 23 LCD Init(); | Run #1: [fromelf -bin ".\test.axf" -output ".\test.bin" | |
| 24 LCD_Clear(RED); 25 Delay(); 26 LCD_Clear(Graen); | □ Run #2: □ □ DOS16 □ DOS16 □ DOS16 □ DOS16 □ Run #2: □ □ DOS16 □ DOS16 □ Run #2: | - |
| E P | | • |
| Build Output compiling system_LPC17xx.c | OK Cancel Defaults Help | ₽ ⊠ |
| linking Program Size: Code=5780 RO-data=167420 RW-data=24 ZI-data FromEF: creating hex file | =608 | 1 |
| User command \$1: fromelfbin ".\test.axf"output ".\te "test.axf" - 0 Error(s), 0 Warning(s). | est.bin" | - |
| <u>د</u> | | F |
| | Simulation CAP N | UM SCRL OVR R/W |

در دستور "test.bin"./test.axf" --output !./test.bin عبارت test مربوط به نام پروژه می باشد و برای استفاده از این دستور در هر پروژه، نام آن پروژه به جای این عبارت قرار می گیرد. به طور مثال در شکل بالا این دستور برای برنامه نمونه 1cd 3.2 که در قسمت محصولات سایت قرار داده شده به کار برده شده است که در آن برنامه نام پروژه test می باشد.

نحوه پروگرم کردن میکروکنترلر با استفاده از بوت لودر USB :

برای استفاده از بوت لودر USB برای پروگرم کردن میکرو باید بوت لودر USB در حافظه میکروکنترلر قرار گرفته باشد. این بوت لودر در آدرس 0x00000000 تا 0x00001FFF از حافظه فلش میکروکنترلر قرار گرفته است. در صورتیکه برای اولین بار می خواهیم از هدر برد استفاده کنیم، بوت لودر USB بر روی حافظه میکروکنترلر قرار داده شده است ولی اگر قبلاً میکروکنترلر توسط نرم افزار JLink پروگرم شده باشد ، بوت لودر از روی حافظه پاک شده است و باید برای استفاده از آن دوباره کد مربوط به بوت لودر که در سایت قرار گرفته (<u>http://www.eca.ir/products/lpc1768-header-board</u>) ، توسط پروگرمر JTAG در حافظه میکروکنترلر پروگرم شود. برای اینکه در استفاده های بعدی از بوت لودر برای پروگرم میکروکنترلر، کد بوت لودر توسط کد برنامه کاربر پاک نشود باید آدرس کد برنامه کاربر بعد از آدرس قرارگیری بوت لودر یعنی از آدرس 0x00002000 شروع شود. برای تنظیم آدرس شروع کد برنامه کاربر از پنجره "Target Options" و تب "Target" مطابق شکل زیر تنظیمات را اعمال می کنیم.

| 😮 C:\Users\ramin\Desktop\LCD3.2-LPC1768\test.uvproj - µVision4 | |
|---|------------|
| File Edit View Project Flash Debug Peripherals Tools SVCS Window Help | |
| 🗋 🖆 🖬 🕼 🔺 🗠 🖏 🤊 🗠 🖛 🔿 🥐 Target Options: /// 🖄 🖉 💿 🗟 🖉 🍳 🖕 📀 🔗 🏨 💷 🔦 | |
| 😵 🛅 🗃 🥔 🚉 🙀 Target 1 🔤 🕋 | |
| Project I maine | ▼ × |
| Tamet 1 1 final vda zinal 7 vz. by | |
| Les Source Group 1 2 | - Â |
| → Startup IPC17xx. 3 | |
| B main.c 4 #define family_lpc17; Device larger Output Listing User C/C++ Asm Linker Debug Utilities | |
| Ipc17xc.h S fdefine TFTLCD DATAP NXP (ounded by Philips) LPC1768 | |
| Code Generation | |
| - in stdint.h 8 fdefine TFILCD CS 22 | |
| - in core_cminstr. 9 #define TFTLCD_RS 23 Operating system: None | |
| Gore_cmfunc 10 #define TFTLCD_WR 24 Surbon Menure Ele (GE) Use MicroLLB Use MicroLLB Big Endian | = |
| system_lpc17 11 fdefine TTTLCD_RD 25 Jstein Viewe Tie (Sn). | |
| - dlach 12 Finclude Tica.n" | |
| | |
| Ent_24xLb.n 15 = void Delay (void) { Read/Only Memory Areas Read/Write Memory Areas | |
| Images.n 16 int i; default off-chip Start Size Startup default off-chip Start Size Noini | |
| H-M system_PCI/X. 17 for (i=0; i<2000000; BOM1: C BAM1: | |
| | |
| | |
| 21 C RAM3: C | |
| 22 SystemInit(); onchip on-chip | |
| 23 LCD_Init(); V IRMI: 0x00002000 0x8000 C V IRAMI: 0x1000000 0x8000 C | |
| 24 LD Clear(RED); 25 DD Liear(NED); 26 DD Liear(NED); 27 DD Liear(NED); 28 DD Liear(NED); 29 DD Liear(NED); 20 DD Liear(| |
| () DELAY(), CARACTER () CARAC | - |
| | • |
| Build Output OK Cancel Defaults Help | 4 |
| assembling startup_LPC17xx.s | * |
| linking | |
| Program Size: Code=5/94 KO-data=16/420 KW-data=24 Z1-data=608 | = |
| User command #1: fromelfbin ".\test.axf"output ".\test.bin" | |
| "test.axf" - 0 Error(s), 0 Warning(s). | + |
| 4 | Þ |
| Simulation CAP NUM SCRL | OVR R/W |

همچنین در پنجره "Target Options" ودر تب "Asm" مقابل کادر Define عبارت NO_CRP را وارد می کنیم.

مطابق با شکل زیر :

شماره تماس: ۴۱۱۵۵۳۳۷۳۲

| Options for T | arget 'Target 1' | x |
|---|--|---|
| Device Target | Output Listing User C/C++ Asm Linker Debug Utilities | |
| Conditional A Define: Undefine: | IND_CRP | |
| Language / | Code Generation Split Load and Store Multiple ly Position Independent ite Position Independent lode ings | |
| Include Paths Controls Assembler control string | -cpu Cortex-M3 -gapcs=interwork -I C:\Keil\\ARM\RV31\lnc -I C:\Keil\\ARM\CMSIS\lnclude -I C:\Keil\\ARM\lnc\NXP\LPC17xx -pd "NO_CRP SETA 1" -list "*.lst" -xref -o "*.o" -depend "*.d" | |

بعد از اینکه تنظیمات فوق انجام شد و برنامه بدون error کامپایل شد، برای پروگرم کردن میکروکنترلر از طریق USB باید مد بوت لودر USB فعال شود . برای این منظور ابتدا دیپ سوئیچ های USB و USBOT قرار گرفته روی هدر برد را در حالت ON قرار می دهیم و با کابل USB پورت usb میکروکنترلر را به پورت usb کامپیوتر وصل می کنیم و سپس تغذیه میکروکنترلر را on می کنیم . با این کار، هدربرد به صورت یک درایو در My Computer ظاهر می شود. مطابق شکل زیر



***اگر درایوی شناخته نشد دو علت می تواند داشته باشد :

۱- کد بوت لودر از حافظه میکروکنترلر پاک شده و دوباره باید پروگرم شود.

۲- کابل و یا پورت USB کامپیوتر خراب است.

سپس درایو شناخته شده را باز کرده و فایل firmware.bin را از داخل آن حذف می کنیم و فایل باینری تولیدی توسط کامپایلر Keil که در بالا توضیح داده شد را به جای آن قرار می دهیم.

4

| 🕞 🖉 🚃 + Computer + CRP DISABLE (S:) | ◆ + Search CRP DISABLE (0) | | |
|---|--|------------------------------|---------------------------|
| rganize 🕶 🔄 Open 🔹 Burn New folder | 8.00 | | |
| # BIN File (1) | | | |
| Desktop | | | |
| Downloads | | | |
| Recent Places | | | |
| 🗘 Dropbox | | | |
| | | | |
| Libraries | 20 | Computer + CRP DISABLE (G) | 44 Search CRP DISABLE (G) |
| Designs | | an e stalpast e stalpast en | A second second |
| Documents | Organiz | e 🕶 🎆 Open 💌 Burn New folder | 5 · |
| er Music | * 6 | A BIN File (1) | |
| Videos | | exitop tert his | |
| Per Construction | 1 C | ownloads all BIN File | |
| Homegroup | 30,7 | ecent Places 169 KB | |
| | 49 r | ropbax | |
| Computer | | | |
| 🚣 Local Disk (Ci) | 词 Lib | anes | |
| ma Softwares (D:) | | esigns | |
| Rose (E) | | ocuments | |
| IN CRP DISABLE (G) | | husic | |
| Marcal (| | ctures | |
| • Network | a. | 0601 | |
| | vil Hs | Theorem | |
| | | | |
| | r Co | nputer | |
| | <u> </u> | acal Disk (C) | |
| | Itta S | aftwares (D:) | |
| | al | ose (E:) | |
| firmware.bin Date modified: 1/11/2008 1/49 PM Date created: 1/23/2008 6/58 PM | | OF DDABLE (0) | |
| BN File Sine SOLKR | | | |
| and set of the | Se to | unark (| |

اکنون پنجره را بسته و دیپ سوئیچ ها را به حالت اول برمی گردانیم و سپس با ریست میکروکنترلر برنامه به اجرا در می آید.

لازم به ذکر است در صورتی مراحل بالا را ه درستی انجام ندهید، برنامه شما به درستی کار نخواهد کرد

وب سایت تخصصی برق و الکترونیک: www.ECA.ir

هد برد LPC1768 : LPC1768 هد برد http://eshop.eca.ir/product_info.php?products_id=876

انجمن های تخصصی مربوط به میکروکنترلرهای LPC: http://www.eca.ir/forum2/index.php?board=295.0/