

راهنمای سریع ساخت، مونتاژ و راه اندازی

LPC1768/2368 miniBoard V1.0

شروع کار با تراشه های LPC1700 با هسته Cortex-M3 و LPC2300 با هسته ARM7TDMI-S

به انضمام بحث توجیهی ARM

مسعود انصاریان

Masoud.Ansarian@gmail.com

پاییز ۱۳۸۹



حق کپی

در مورد طرح مدار، مطالب مطرح شده در این نوشتار، و برنامه های نمونه؛ هر گونه تغییر و استفاده آموزشی آزاد است. به سبب زحماتی که کشیده شده، ذکر منبع، موجب امتنان خواهد بود.

رفع ادعا

توجه! مولف هیچ گونه مسئولیتی در زمینه کلیه موارد مطرح یا اشاره شده در این نوشتار، طرح مدار، و دیگر موارد همراه آن ندارد. همچنین کلیه افراد، نام ها، سایت ها، و فروشنده های مطرح شده در این جا تنها به منظور آشنایی مخاطب، مطرح شده و مولف قصد هیچ گونه تبلیغ یا تایید آن ها را به هیچ وجه و صورتی ندارد.

درباره مولف

در حال حاضر، دانشجوی کارشناسی ارشد مهندسی کامپیوتر- معماری سیستم های کامپیوتری هستم و اندکی هم آشنایی و تجربه کاری در زمینه های طراحی سیستم های (ELEC/HW/SW) SoPC/embedded، طراحی/توسعه پردازنده های SoftCore مبتنی بر FPGA، برنامه نویسی / توسعه نرم افزارهای embedded، RTOS، processor، cross tool chain، سیستم های واسط، برنامه نویسی شبکه و سیستمی لینوکس/یونیکس (USP)، و طراحی PCB دارم.

برسر آنم که گر ز دست بر آید دست به کاری زنم که غصه سر آید

فهرست

- ۱- مقدمه
 - ۱-۱- اهداف
 - ۲-۱- تشکرات
 - ۳-۱- موارد مورد نیاز
 - ۴-۱- امکانات برد
 - ۵-۱- برآورد قیمتی
 - ۶-۱- دیگر محتویات همراه این نوشتار
- ۲- نکات ساخت و مونتاژ
- ۳- راه اندازی و برنامه نویسی چیپ
 - ۳-۱- برنامه ریزی از طریق نرم افزار Flash Magic
 - ۳-۱-۱- برنامه ریزی به صورت منفرد
 - ۳-۱-۲- برنامه ریزی از درون Keil
 - ۳-۲- برنامه ریزی توسط JLINK/ULINK2
 - ۳-۳- دیباگ توسط JLINK/ULINK2
 - ۳-۴- - شبیه سازی در Keil
- ۴- ادامه
- ۵- منابع
 - ۵-۱- منابع طراحی مدار و PCB
 - ۵-۲- دیگر منابع
 - ۵-۳- منابع مفید
- ضمیمه ۱- چرا ARM؟

۱- مقدمه

این نوشتار راهنمای مختصری است برای ساخت و راه اندازی یک نمونه برد ساده برای LPC1768/2368. سعی شده است که به کلیه مواردی که مفید به نظر می رسیده، به صورت اجمالی و مقدماتی پرداخته شود. شایان توجه است که این یک طراحی کاملاً ساده، مقدماتی، و غیر حرفه ای است که با اهداف بیان شده در بخش زیر، به صورت بازمتن منتشر می شود و پیش از انتشار، قسمت های مختلف آن با هر دو تا چیپ تست شده است. امیدوارم تا حد امکان خالی از اشکالات عمده باشد و برای دوستان مفید واقع شود. قبلاً از اظهار نظرها و مطرح ساختن اشکالات، متشکراً!

۱-۱- اهداف

برد LPC1768/2368 miniBoard یک برد کاملاً ساده و غیر حرفه ای است با اهداف ذیل:

- راهنمایی سریع، ساده، و مستقیم برای شروع کار، راه اندازی، و کسب تجربه کاری با دو میکروکنترلر با فرکانس و امکانات درست و حسابی و قابلیت های قدرتمند، و بروز در کلاس کاری خودش (ضمیمه ۱ را ببینید).
- طرح پایه ای باشد برای طراحی های سفارشی و کامل تر در این زمینه.
- کمکی هرچند اندک به ادامه توسعه جریان بازمتن برای سخت افزار/نرم افزار. (دست کم برای انتشار موارد پایه ای و مشترک، و نه محصولات نهایی، مدل بازمتن بسیار مفید به نظر می رسد)
- کمک به آشنایی هر چه بیشتر مهندسین طراح با چیپ های قدرتمند و مفید با هسته ARM، به ویژه تراشه های سری LPC1700 و LPC2300 از Philips(NXP)؛ و تا حد ممکن توجه استفاده از این تراشه ها در محصولات نهایی.
- هدف، هیچ یک از موارد آموزش برنامه نویسی و کار با Kiel، راه اندازی ماژول ها و امکانات چیپ، آموزش طراحی مدار یا PCB و مانند آن نبوده است بلکه منظور، ارائه یک پلتفرم حتی امکان ساده و ارزان همراه با سورس، برای شروع کار در موارد مطرح شده است. (بخش های ۴ و ۵ را ببینید)

۱-۲- تشکرات

- تشکر از محتویات و سایت ها و کاربرانی که کارهایشان را در زمینه ARM و دیگر زمینه ها به صورت باز متن ارائه می دهند (گذشته، حال، آینده! مثلاً سایت هایی مانند eca.ir و iranmicro.ir حاوی مطالب و البته لینک های مفیدی هستند).

- تشکر از همه دوستان که راهنماییم کردن به ویژه آقای بهنام اسکندریون.

- تشکر از شرکت های ARM و Philips!

- یک حکایت بامزه! 😊: تشکر از آقای جوان الکترونیک که یک LPC1768 رو که پایه ۹۵ را کم داشت اشانسیون داد، تا رسیدم خونه دیدم که پین ۹۵، gpio و Ethernet Transmit هست و نه پایه VCC/GND. روی همین برد با فرکانس 100MHz راهش انداختم!!!!

۱-۳- موارد مورد نیاز

- منبع تغذیه 9V-12V DC

- امکانات مونتاژ SMD (در صورت نیاز، مجله 6-7 pmm سایت iranmicro.ir را ببینید)

- PC با پورت سریال ، در صورتی که از لپ تاپ استفاده می کنید می توانید از مبدل USB-UART استفاده کنید. حدود ۱۰,۰۰۰ تومان از جمهوری می توانید تهیه کنید یا می توان با چیپ های FTDI یکی درست کرد.

- نرم افزار Keil که می توانید از سایت Keil دانلود کنید. برای دانلود crack و یا خود نرم افزار نیز از طریق لینک های سایت های فوق اقدام کنید. (بیش از 100MB)

- نرم افزار Flash Magic برای برنامه ریزی، که رایگان است و می توانید از سایتش دانلود کنید.(حدود 6.7MB www.flashmagictool.com)

- برای کارهای جدی، دیباگر ARM، که ULINK2 حدود ۱۲۰,۰۰۰ تومان و نسخه های مختلف JLINK حدود ۱۵۰,۰۰۰ تومان تا ۴۰,۰۰۰ تومان بسته به نسخه و قابلیت هاش، تو جمهوری هست یا از طریق سایت ها تهیه کنید. توجه کنید که از ULINK2 تنها در نرم افزار Keil می توانید استفاده کنید. من با هر دوتاش با keil کار کردم و برای خودم JLINK خریدم از بل الکترونیک- جمهوری- پاساژ فرشته.

به نظر من برای کارهای حرفه ای وجود یک دیباگر کامل و پایدار ضروری است وگرنه گزینه wiggler هم برای برنامه ریزی و دیباگ از طریق پورت موازی وجود دارد که من تست نکردم. (دیباگرهای بالا USB اند)

- مستندات مورد نیاز شامل دیتا شیت ها و appNote ها و مجموعه مثال های(ی Keil) که بسیار مفید هستند و همگی قابل دانلود از سایت NXP می باشند (کافی است LPC1768 یا LPC2368 را تو گوگل جستجو کنید)

- برای شروع کار؛ آشنایی با زبان C، مبانی الکترونیک، و مبانی میکروکنترلرها.

۱-۴- امکانات برد

- قابلیت به کارگیری هر یک از دو نوع میکروکنترلر یعنی LPC1768 با هسته Cortex-m3 و فرکانس 100MHz، و LPC2368 با هسته ARM7TDMI-S و فرکانس 72MHz. پیکربندی پایه های این دو چیپ یکسان است. (AN10878: Migrating to the LPC1700 series بخش Pin compatibility 8.1.1 را نگاه کنید).

- همه IO ها بیرون کشیده شده

- یک LED تست رو پایه ۳۴ یعنی P1p20

- UART0,1 به عنوان stdio

- سخت افزار لازم برای برنامه ریزی از طریق نرم افزار FlashMagic

- سخت افزار لازم برای برنامه ریزی و دیباگ از طریق ارتباط JTAG با دیباگرهایی نظیر ULINK2/JLINK

۱-۵- برآورد قیمتی (در زمان تدوین این نوشته)

- LPC1768FBD100 که ۱۴,۰۰۰ تومان جوان داره.

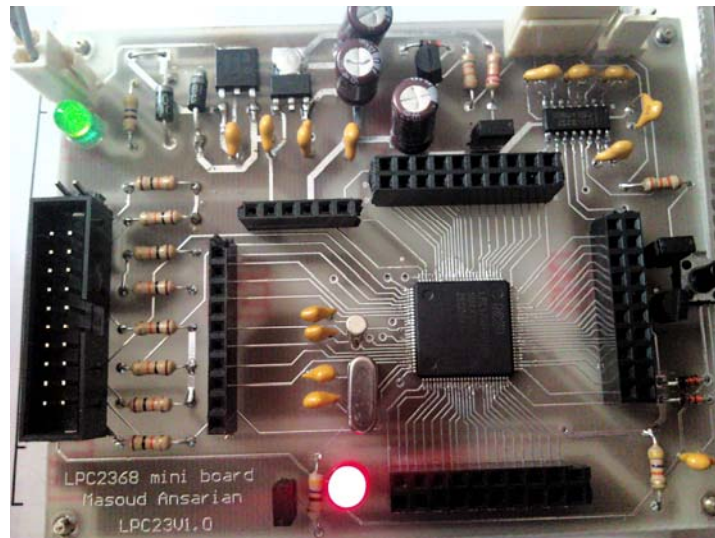
- LPC2368FBD100 هم بین ۱۰,۰۰۰ تا ۱۲,۰۰۰

- MAX3232CSE, LD1117, 7805 که روی هم ۲,۰۰۰ تومان همیشه

- خازن های دکوپلاژ 100nF که 0603 هستن که برای کارای آموزشی و راه اندازی ساده، ضروری نیست، میشه مونتاژ نکرد.

بقیه قطعات شامل دیودها، خازن ها و ... همگی dip هستن (برای طراحی های جدی، حتی الامکان نسخه های SMD پیشنهاد می شود).

- PCB، ۸×۱۰، دو رو، و متالیزه (برای نمونه سازی اولی، بدون چاپ سبز و راهنما زدم، البته یک سری راهنما رو لایه بالا گذاشتم. تصویر زیر را ببینید. به هر حال مراقب باشید که در این حالت مسیرها و پولی عایق نیستند و استحکام مسیرها مانند حالت با چاپ سبز نیست. البته توی یک نمونه بدون چاپ سبز، سه بار چیپ رو عوض کردم، مشکلی نبود! برای ساخت هم به آلفا مدار دادم، تک نمونه احتمالا حدود ۱۱،۰۰۰ تومان بشود. فکر کنم جاهای ارزون تر هم باشه).



برای شروع می توان از دیباگر ARM صرفنظر کرد و از نرم افزار Flash Magic برای برنامه ریزی برد استفاده کرد. به هر حال برای کارهای حرفه ای تر، وجود یک دیباگر خوب، به منظور on-Chip Debugging، ضرورت مطلق دارد.

۱-۶- دیگر محتویات همراه این نوشتار

- فایل های پروژه، شماتیک، و PCB برای Altium08/09
- برنامه های Keil-C نمونه برای LED تست و راه اندازی UART برای هر دو تا چیپ
- لیست جداگانه قطعات برد با توضیحات

۲- نکات ساخت و مونتاژ

(هرچه که به نظرم مفید برسه، مطرح می کنم)

- برای LED های تغذیه و تست ترجیحا از دو رنگ متفاوت استفاده شود
- برای ورودی تغذیه از کانکتور مخابراتی معمولی استفاده کردم.



- برای ارتباط سریال از کانکتور مخابراتی معمولی استفاده شده و از طریق مثلا سیم فلت (۵ تایی برای JCOM0 و ۳ تایی برای JCOM1) به کانکتور Female DB9 متصل کردم. (برای طراحی های جدی، شاید بهتر باشد که PCB تغییر داده شود و از کانکتور روی برد استفاده شود. بخش ۵-۱ را ببینید)



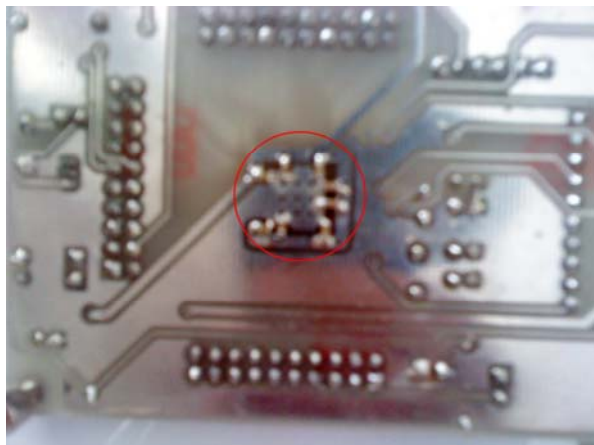
- ترتیب پایه های JCOM0 برای اتصال به کانکتور DB9

DB9	JCOM0
2 RxD	1
7 RTS	2
3 TxD	3
4 DTR	4
5 GND	5

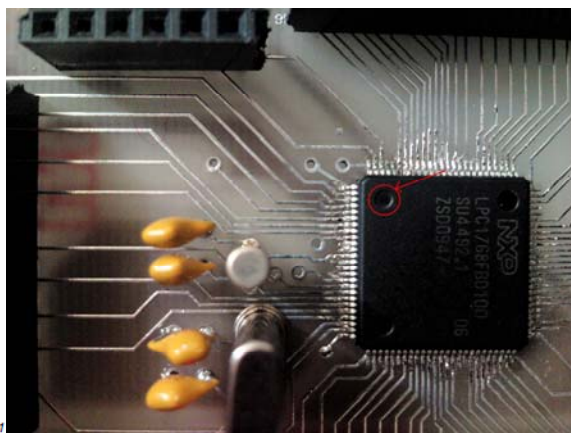
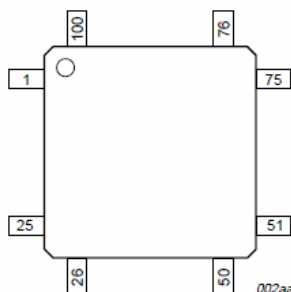
- ترتیب پایه های JCOM1 برای اتصال به کانکتور DB9

DB9	JCOM1
2 RxD	2
3 TxD	3
5 GND	1

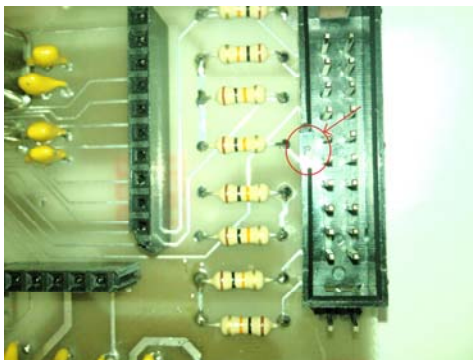
- قطعاتی که توان می سوزانند یعنی D100, D101 را قدری بالاتر از فیبر مونتاژ کنید.
- برای یک راه اندازی ساده و استفاده ی آموزشی، مونتاژ خازن های دکوپلاژ پایه های تغذیه میکرو که SMD(0603) هستند، ضرورت ندارد. (البته برای کارهای جدی، دو سه تا خازن مثلا 10uF نیز، علاوه بر این خازن ها، مفید است!)



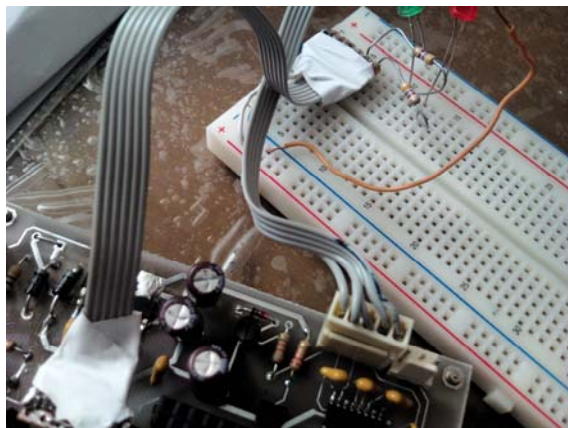
- هنگام مونتاژ، به جهت MAX3232 و خصوصا خود میکرو دقت شود. برای LPC1768/2368 FBD100، اگر آن را روبه روی خود بگیرید، علامت، در پایین سمت چپ قرار دارد. علامت جهت میکرو نسبت به دو علامت دیگر، گودی بیشتری دارد.



- IDC 20-PIN که برای ارتباط JTAG هست باید طوری مونتاژ شود که علامت آن به سمت داخل برد باشد.



- برای ارتباط برد با مثلاً یک برد دیگر یا bread board، می توان یک کابل رابط با استفاده از سیم فلت، فیبر ماتریسی، و پین هدر درست کرد. خودم روی برد، پین هدر مادگی مونتاژ کردم و کابل رابط را با فیبر ماتریسی و پین هدر نرگی ساختم. خیلی خوب محکم و فیت می شود.



- هدر IO4(1x6) شامل دو خروجی gnd و دو خروجی 3.3V می باشد که می تواند برای ارائه تغذیه به بردها و ماژول های دیگر به کار رود.

- اگر نیاز به راهنمایی درباره تجهیزات و مونتاژ قطعات SMD بود، می توانید به مجله 6-7 pmm سایت iranmicro رجوع کنید.

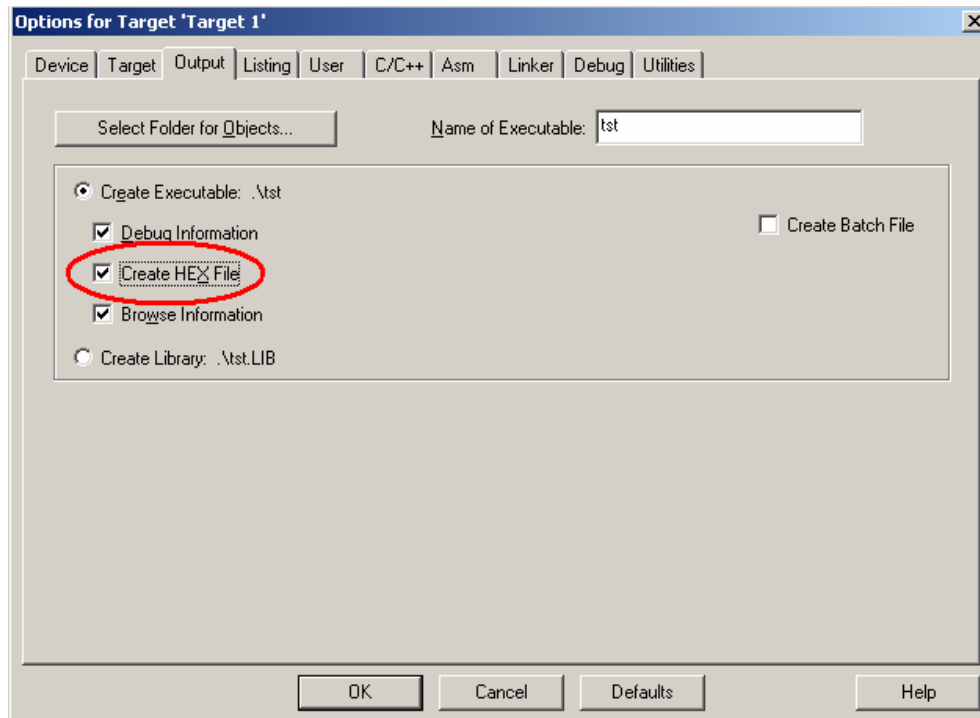
۳- راه اندازی و برنامه نویسی چیپ

(باز هم هر نکته ای که فکر کنم مفید هست را مطرح می کنم)

۳-۱- برنامه ریزی از طریق نرم افزار Flash Magic

- نخست پروژه gpio_lpc1768 یا gpio_lpc2368 را وابسته به میکروبی که روی برد مونتاژ کردید، در Keil کنید.

- از طریق منوی Flash-Configure Flash Tools... یا Project-Options for Target... دیالوگ ذیل را باز کنید و در برگه Output، گزینه مشخص شده در تصویر زیر را انتخاب نمایید.



و سپس از منوی Project-Build Target فایل خروجی را تولید کنید.

- تغذیه مدار را حدود ۹ تا ۱۲ ولت، متصل کنید.

- جامپرهای J201 و J202 و J300 را متصل نمایید.

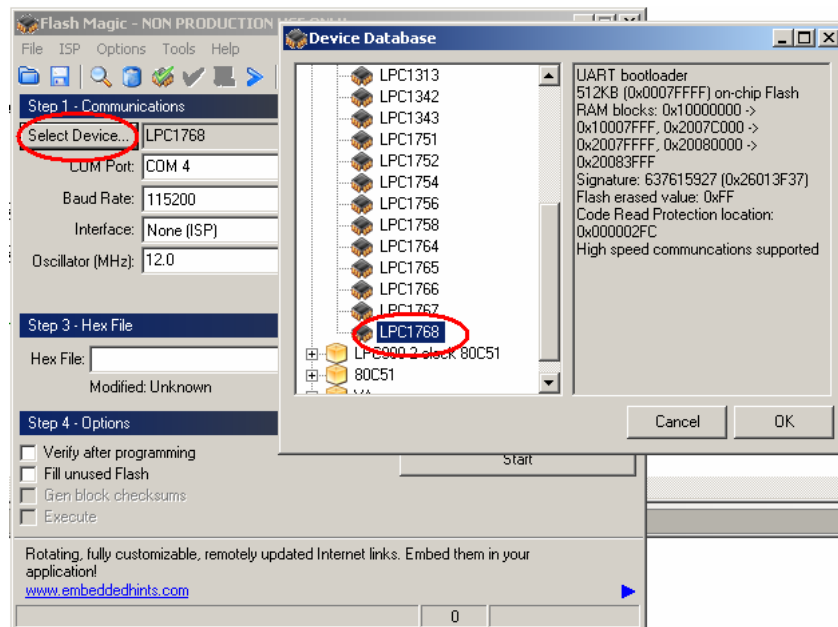
- JCOM0 (اتصال ۵ سیمه) را به پورت سریال PC متصل کنید.

برنامه ریزی را می توان به دو صورت انجام داد.

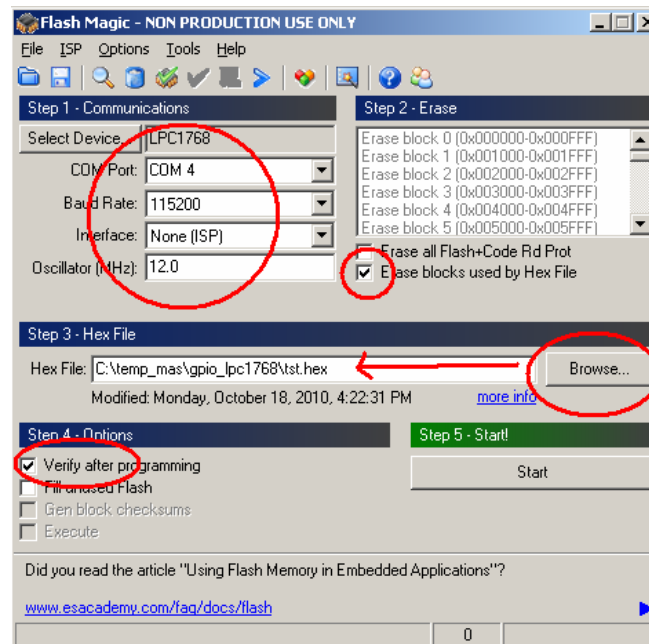
۳-۱-۱- برنامه ریزی به صورت منفرد

- برنامه Flash Magic را نصب و اجرا نمایید و با فشردن دکمه Select Device... چیپ مورد نظر را از شاخه

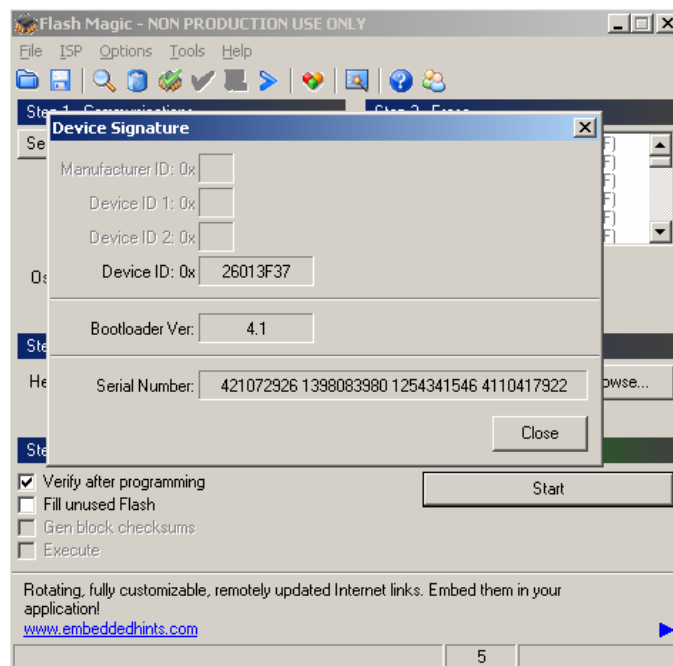
ARM7 برای LPC2368 و یا از شاخه ARM Cortex برای LPC1768 انتخاب کنید



بقیه گزینه ها را مطابق با شکل ذیل تنظیم کنید. در مورد Com Port شماره پورت سریال PC خودتون را انتخاب کنید. و فایل هگز را از مسیر پروژه gpio_lpc1768 یا gpio_lpc2368 انتخاب نمایید. در step2 می توانید هر یک از دو چک باکس را انتخاب کنید.



- برای تست ارتباط می توان از منوی ISP-Read Device Signature استفاده کرد.



- برای برنامه ریزی دگمه **Start** را بزنید.
 نکته: گاهی ممکن است در طول این فرآیند، اشکال پیش آید لذا در صورت لزوم، دو یا سه بار تلاش غیر عادی نیست!
 در صورت بروز چنین وضعی، معمولاً فشردن دگمه **RST** روی برد مفید واقع می شود.



۳-۱-۲- برنامه ریزی از درون Keil

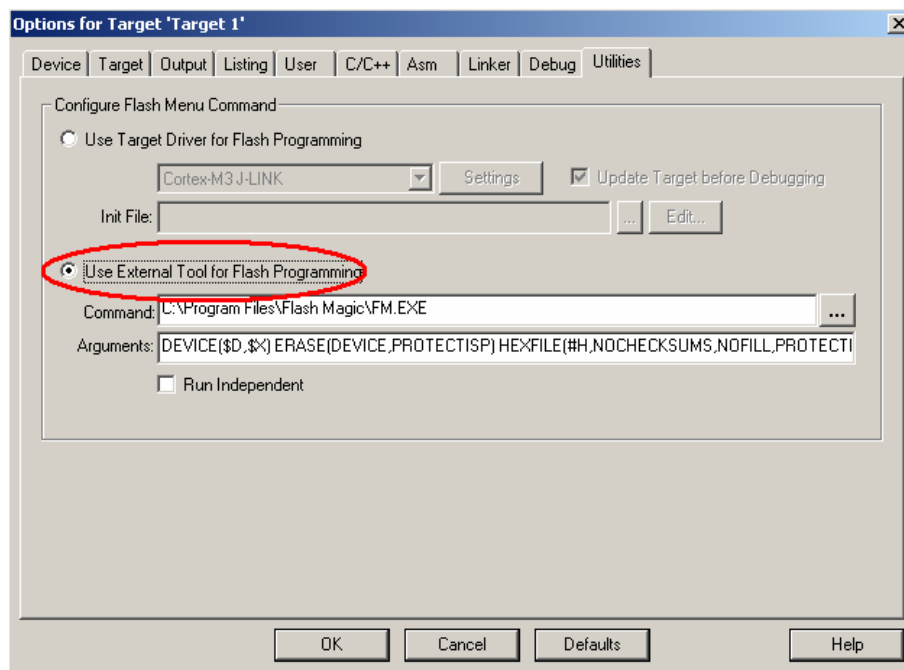
از قابلیت های جالب برنامه Flash Magic امکان اجرا از درون keil می باشد. بدین منظور نخست از منوی **Flash-Utilities** گزینه مشخص شده در تصویر زیر را انتخاب نمایید.
 برای فیلد **Command** فایل **FM.EXE** را که در مسیر برنامه Flash Magic قرار دارد انتخاب کنید و فیلد آرگومان را هم مطابق عبارت زیر پر نمایید:

```
DEVICE($D,$X) ERASE(DEVICE,PROTECTISP)
HEXFILE(#H,NOCHECKSUMS,NOFILL,PROTECTISP) COM(4,115200)
HARDWARE(BOOTEXEC,100,50)
```

این عبارت از منوی **Help-uVision Help** در صفحه تحت شاخه (فایل CHM)

MCB2300 User's Guide-Writing Programs-Programing Flash-Downloading with Flash Magic- External Toolin uVision

استخراج شده است و کافی است که تنها عبارت های قرمز رنگ در آن اصلاح شود.



حالا از منوی Flash-Download تراشه را برنامه ریزی نمایید. نکته بخش قبل در این جا هم صادق است.

```
Build Output
ARM UART Driver Version 2.14.1783
ARM Cortex UART Driver Version 1.28.1827
ARM Ethernet Driver Version 1.08.1770
ARM CAN Driver Version 1.01.1770
(C) Embedded Systems Academy 2000-2009 All rights reserved
NON PRODUCTION USE ONLY
Connected
Device selected
Erase complete (DEVICE)
Hex file programming complete (C:\temp_mas\gpio_lpc1768\tst.HEX)
```

نکته: برای LPC2368 (و در واقع سری LPC2000)، مطابق با AN10744، از نرم افزار Ethernet Flash Utility هم می توان برای برنامه ریزی تراشه استفاده کرد. این نرم افزار نیاز به NET2 دارد.

۳-۲- برنامه ریزی توسط JLINK/ULINK2

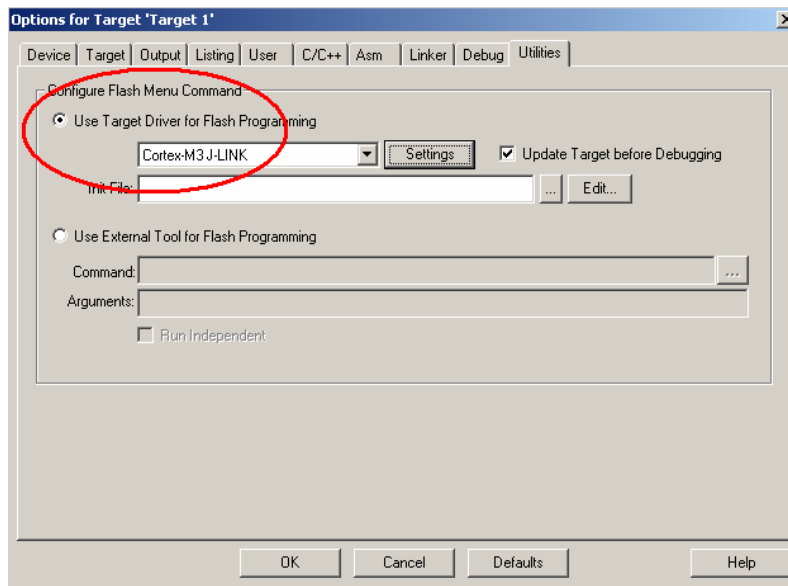
نخست برنامه gpio_lpc1768 یا gpio_lpc2368 را وابسته به میکرویی که روی برد مونتاژ کردید، در Keil باز کنید و از طریق منوی Project-Build Target فایل خروجی را تولید کنید. از طریق منوی Flash-Configure Flash Tools... یا Project-Options for Target... دیالوگ ذیل را باز کنید و در برگه Utilities گزینه مشخص شده در تصویر زیر را انتخاب نمایید. برای LPC1768 :

- اگر از JLINK استفاده می کنید گزینه Cortex-M3 J-LLINK

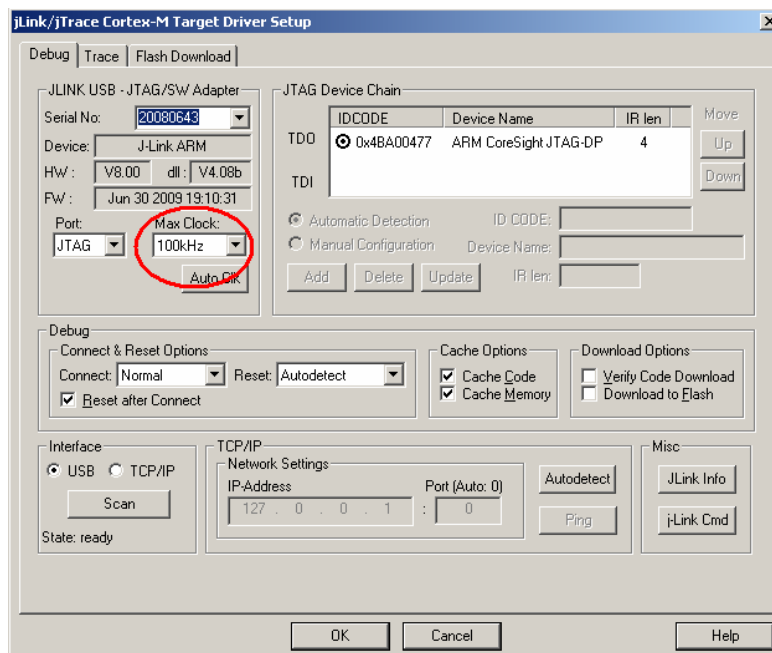
- اگر از ULINK2 استفاده می کنید گزینه ULINK Cortex Debugger

و برای LPC2368

- اگر از JLINK استفاده می کنید گزینه J-LLINK / J-TRACE
- اگر از ULINK2 استفاده می کنید گزینه ULINK ARM Debugger
- را از کامیوباکس انتخاب کنید.

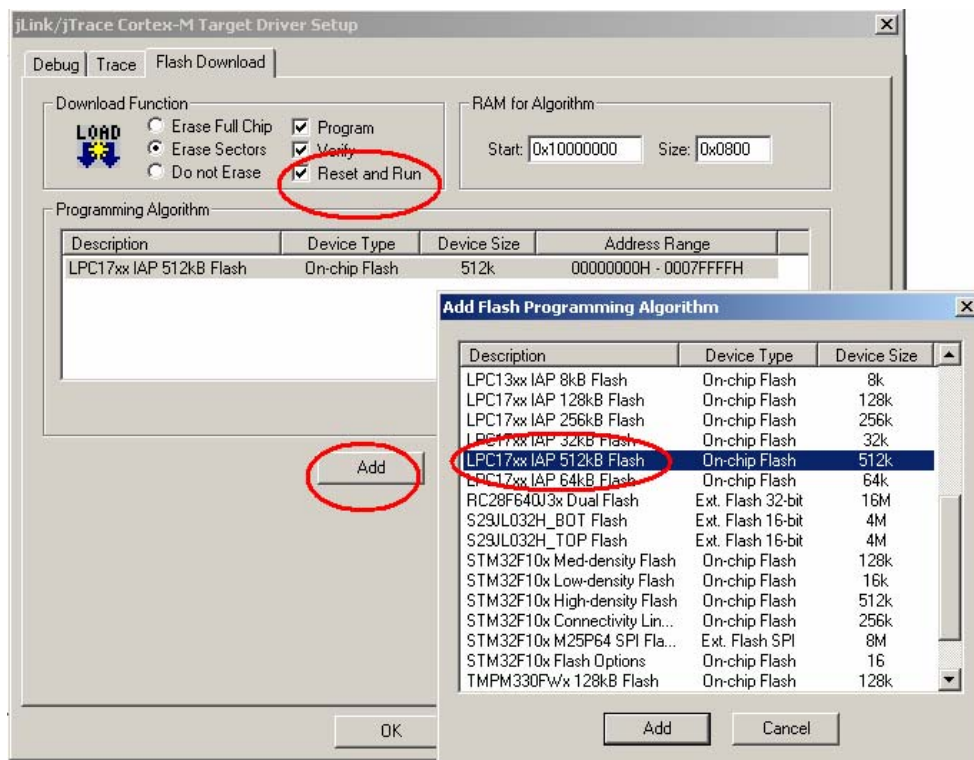


- سپس با زدن دکمه Settings در دیالوگ باز شده نخست بهتر است که در برگه Debug
- اگر از JLINK استفاده می کنید گزینه Max Clock را مثلاً 100KHz
- اگر از ULINK2 استفاده می کنید گزینه Max JTAG Clock را مثلاً 200KHz
- انتخاب کنید (مقادیر خیلی بالا کار نمی کند).



- وبعد در برگه Flash Download تنظیمات را مطابق تصویر زیر انجام دهید:
- گزینه Reset and Run را انتخاب کنید
- با فشردن دکمه Add از کادر باز شده، گزینه LPC17xx IAP 512kB Flash را برای LPC1768 و یا

LPC2000 IAP2 512kB Flash را برای LPC2368 انتخاب نمایید
توجه شود که این مرحله برای هردوی JLINK و ULINK2 تقریباً یکی است.



- تغذیه مدار را حدود ۹ الی ۱۲ ولت متصل کنید.
- جامپر tstLED یعنی J300 را متصل نمایید.
- اتصال JLINK یا ULINK2 را از طریق USB با PC و از طریق IDC با برد برقرار کنید.
- حالا کافی است که منوی Flash-Download را انتخاب کنید تا چیپ برنامه ریزی شود. توجه گردد که در این مرحله نیازی به فعال کردن گزینه Create HEX File نیست.
- در مورد JLINK نیازی به اجرای دستی GDB Server نیست و مانند ULINK2 کارها خودکار انجام می گیرد.

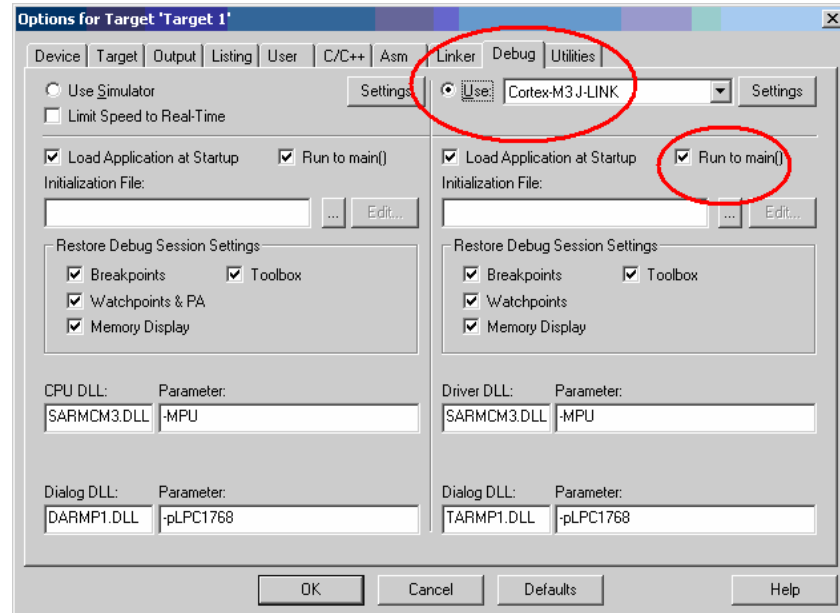
```
Build Output
* JLink Info:  FPUnit: 6 code (BP) slots and 2 literal slots
Erase Done.
Programming Done.
Verify OK.
* JLink Info: TotalIRLen = 4, IRPrint = 0x01
* JLink Info: Found Cortex-M3 r2p0, Little endian.
* JLink Info: TPIU fitted.
* JLink Info: ETM fitted.
* JLink Info:  FPUnit: 6 code (BP) slots and 2 literal slots
Application running ...
```

۳-۳- دیباگ توسط JLINK/ULINK2 (Emulation or on-chip Debugging)

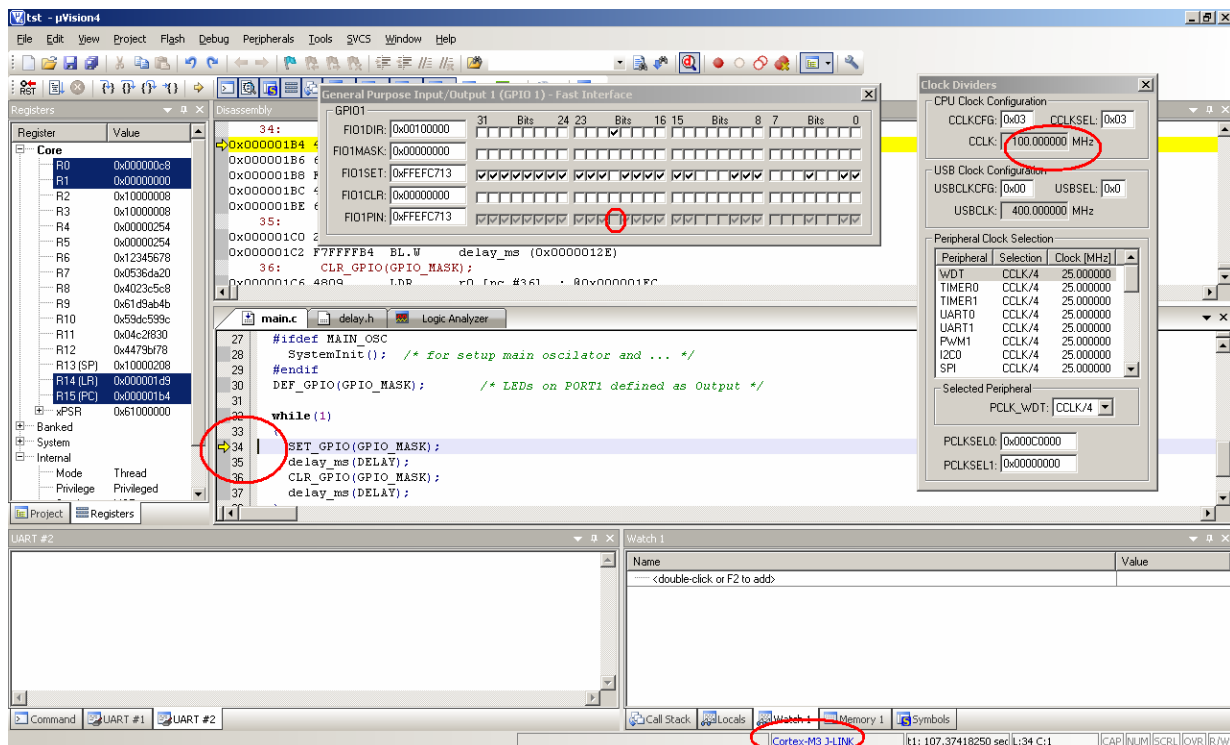
قابلیت اشکال زدایی برنامه به صورت درون-تراشه ای از قابلیت های قدرتمند و بسیار ضروری در پروژه های جدی است. با استفاده از سخت افزار مناسب یعنی JLINK یا ULINK2 می توان به سادگی برنامه را در محیط Keil،

حالی که بر روی چیپ مقصد در حال اجراست، ردیابی یا متوقف کرد، وضعیت و محتویات بخش های مختلف تراشه از قبیل رجیسترها یا خانه های حافظه (متغیرها در C) را مشاهده نمود یا تغییر داد، از نقاط شکست در برنامه استفاده نمود و خلاصه عملکرد برنامه را در شرایط واقعی بازیابی و اصلاح کرد.

از طریق منوی Flash-Configure Flash Tools... یا Project-Options for Target... دیالوگ ذیل را باز کنید و علاوه بر تنظیمات برگه Utilities و دگمه Settings که در بخش قبل آمد، در برگه Debug نیز گزینه های مشخص شده در تصویر زیر را انتخاب نمایید. انتخاب محتویات کامیوباکس مانند بخش قبل است.

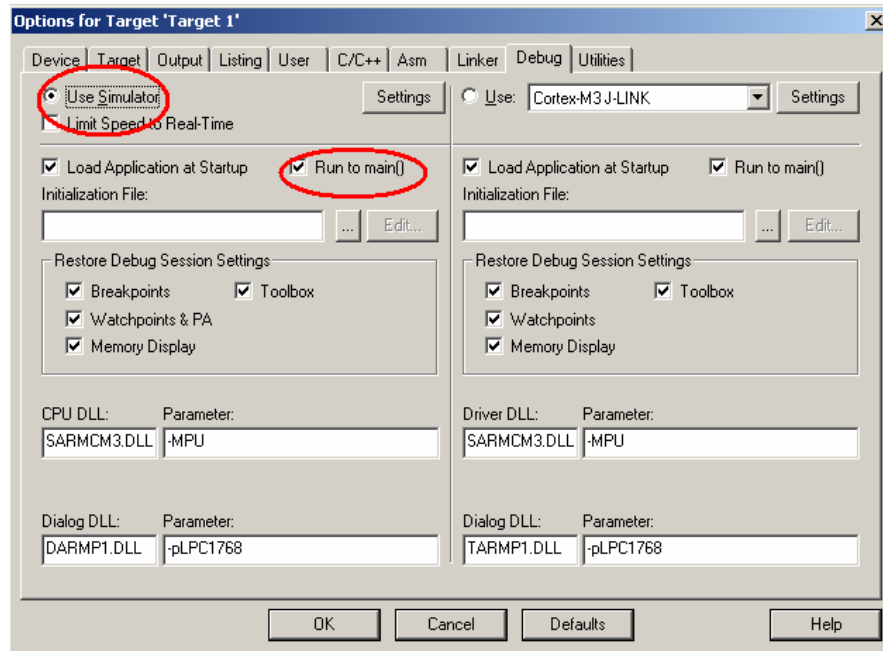


حال می توان بعد از برنامه ریزی تراشه؛ از منوی Debug-Start/Stop Debug Session وارد مد دیباگ شد. امکانات مورد نظر را می توانید از منوی Peripherals انتخاب کنید. تصویر زیر، چیپ LPC1768 را در حال دیباگ برنامه gpio_lpc1768 در Keil نمایش می دهد.

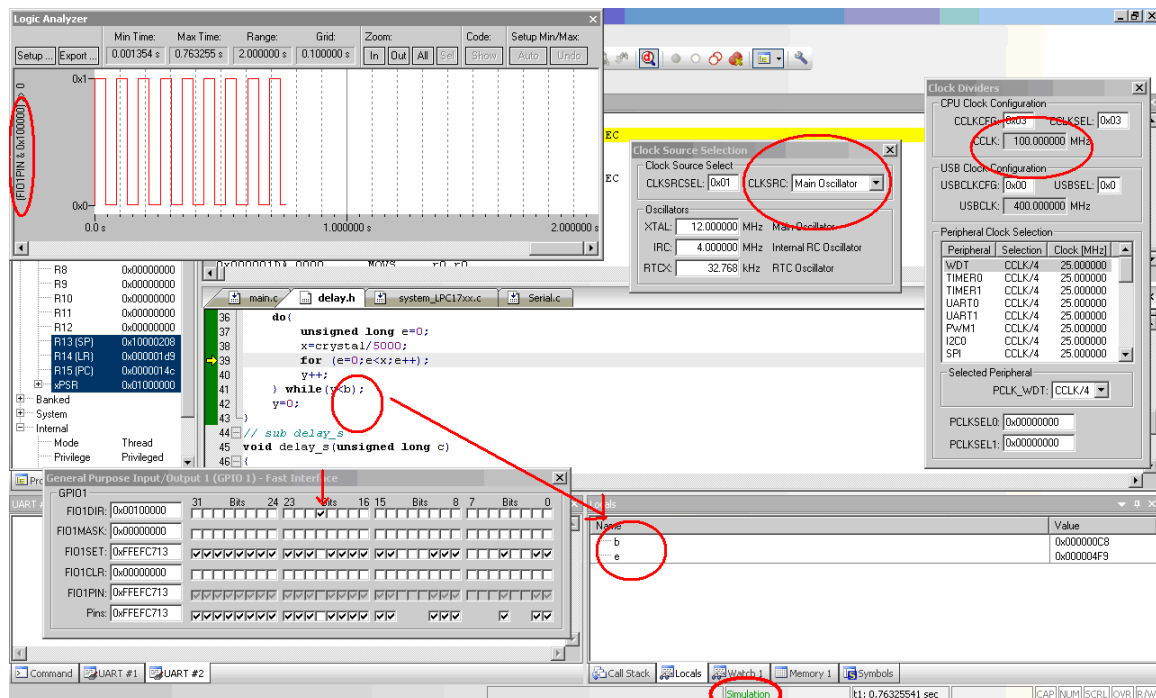


۴-۳- شبیه سازی در Keil (Simulation)

بهرتر است پس از نوشتن برنامه و پیش از برنامه ریزی تراشه، برنامه را به صورت شبیه سازی مورد آزمون قرار داد و در واقع آن را به صورت off-Chip دیباگ نمود. در این زمینه نیز نرم افزار Keil دارای امکانات قدرتمندی است. برای این منظور کافی است که از طریق منوی Project-Options for Flash-Configure Flash Tools یا Target... دیالوگ ذیل را باز کنید و در برگه Debug گزینه های مشخص شده در تصویر زیر را انتخاب نمایید.



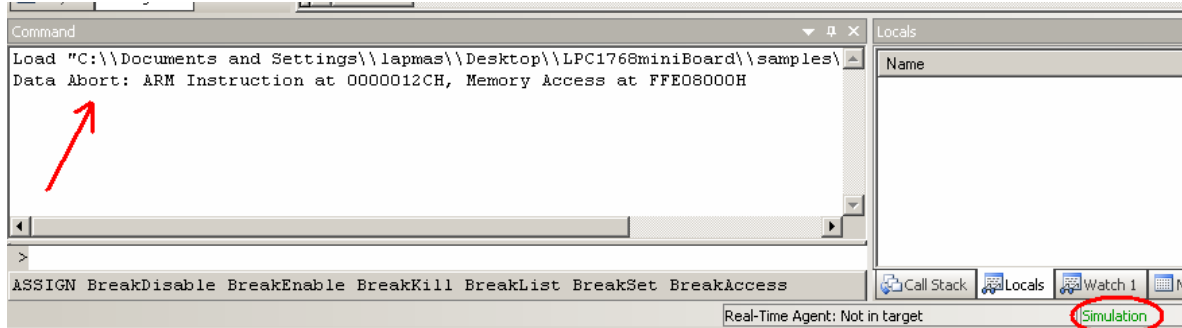
حال می توان از همان منوی Debug-Start/Stop Debug Session وارد مد شبیه سازی شد. باز هم امکانات مورد نظر را می توانید از منوی Peripherals انتخاب کنید. تصویر زیر، چپ LPC1768 را در حال شبیه سازی برنامه gpio_lpc1768 در Keil نمایش می دهد. به ویژه به قابلیت جالب Logic Analyzer توجه کنید.



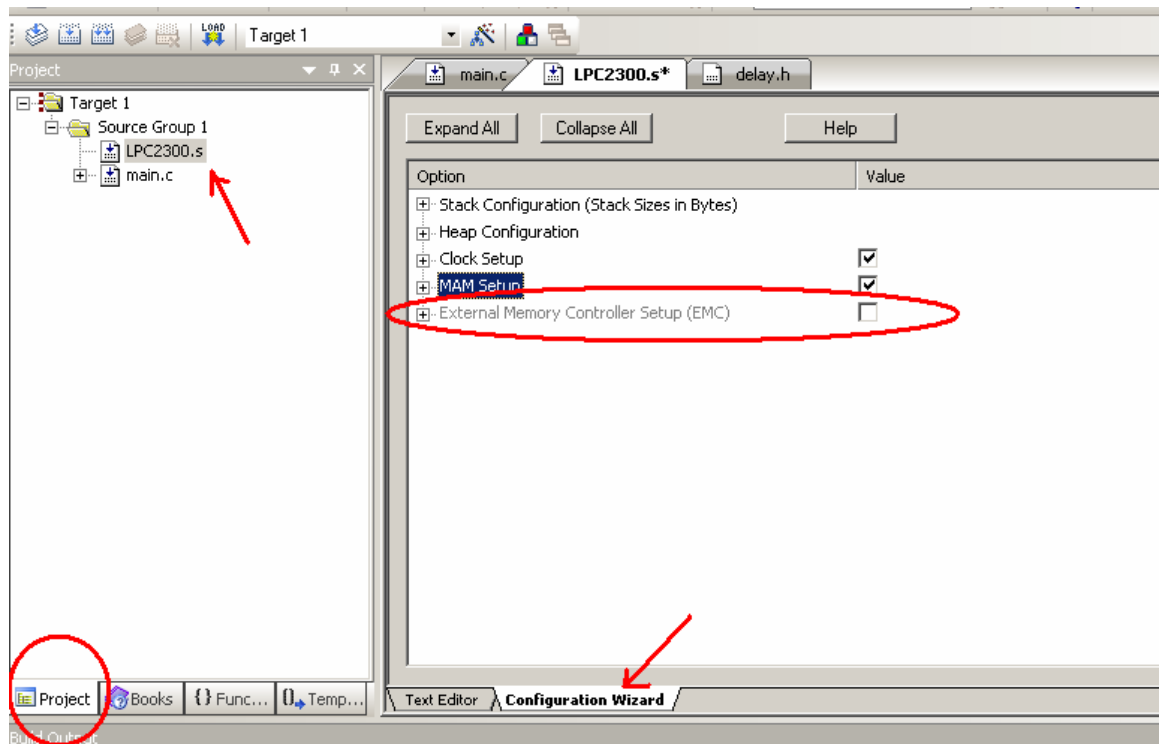
نکته ۱: LPC1768 در حالت عادی با اسیلاتور داخلی 4MHz کار می کند، هم در شبیه سازی و هم در اجرا روی چیپ. در Keil 4.01 برای این تراشه، ظاهراً در Configuration Wizard گزینه ای برای تنظیم اسیلاتور و PLL موجود نیست (برای LPC2368 هست). از این رو در برنامه های نمونه همراه این نوشته، برای این تراشه، از تابع SystemInit() استفاده شده که CLKSRC را به اسیلاتور اصلی تغییر می دهد و فرکانس CCLK را 100MHz می نماید. (تفاوت سرعت چشمک زدن LED تست در دو حالت وجود/عدم این تابع در برنامه gpio_lpc1768 به وضوح قابل مشاهده است). این تابع را از مثال Blinky مربوط به برد Keil MCB1700 برداشتم.

\Keil\ARM\Boards\Keil\MCB1700\Blinky

نکته ۲: در هنگام شبیه سازی در Keil (و نه حالت دیباگ در تراشه)، ممکن است برای تراشه LPC2368 به خطای زیر برخورد کنید و لذا امکان شبیه سازی و ردیابی برنامه وجود نخواهد داشت.

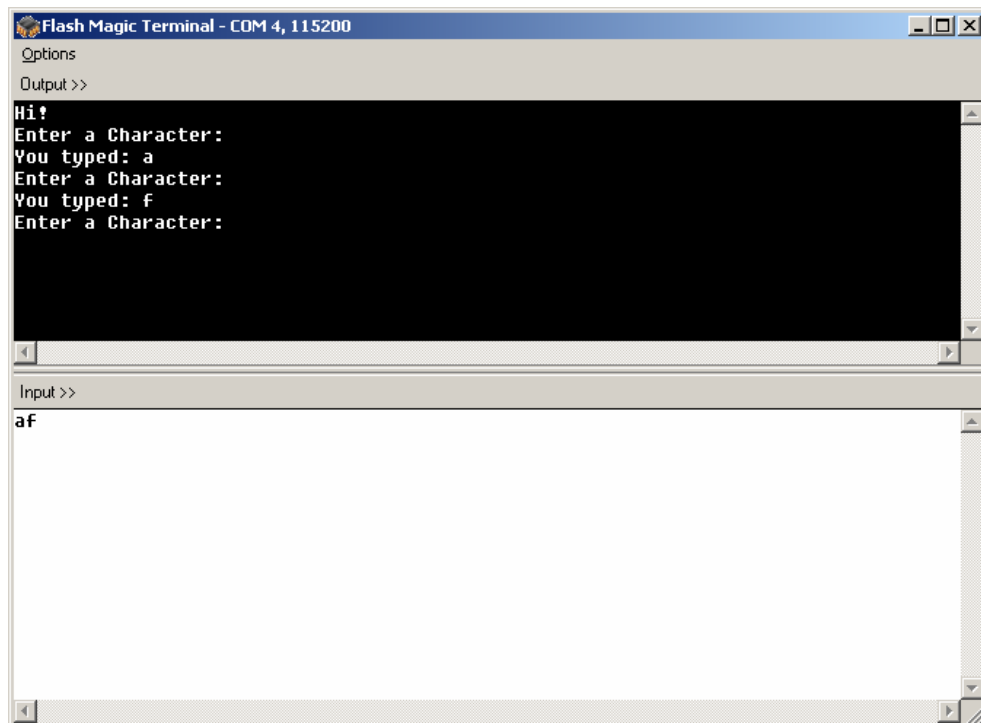
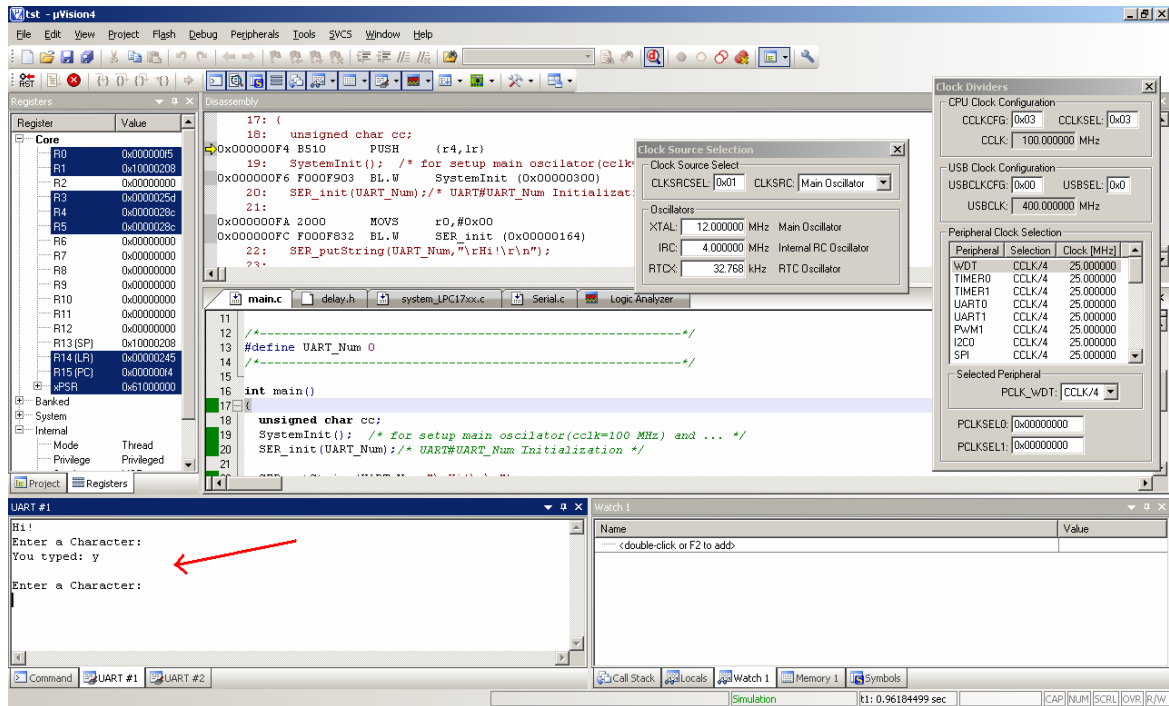


برای رفع مشکل، از حالت شبیه سازی خارج شوید، در برگه project در سمت راست بر روی LPC2300.s دوبار کلیک کنید تا این فایل باز شود. سپس زبانه Configuration Wizard را انتخاب نموده و تیک عبارت External Memory Controller Setup را بردارید. حالا پروژه را دوباره Build کرده، و بعد شبیه سازی کنید. (این مشکل از آن جا ناشی می شود که در سری LPC236x بر خلاف سری LPC237x، ماژول EMC وجود ندارد ولی هنگام شبیه سازی، بدون توجه به این موضوع، این ماژول فعال است).



۳-۵- برنامه تست ارتباط سریال

اکنون باید به سادگی بتوانید برد را با برنامه `simple_uart` برنامه ریزی و تست کنید. نمونه ای از اجرای این برنامه به صورت شبیه سازی و نیز خروجی اجرای روی برد با نرخ جالب 115200 برای LPC1768 در پنجره Flash Magic Terminal که از منوی Tools-Terminal برنامه Flash Magic قابل دست یابی است، در تصاویر ذیل نشان داده شده است. (می توان از Hyper Terminal یا miniCom و ... استفاده کرد)



برای یادگیری بیشتر در خصوص کار با نرم افزار Keil، علاوه بر راهنمای نسبتاً مناسب آن، می توان راهنمای uv4_keil_Getting Started.pdf را نیز از سایت Keil دانلود کرد. مطالب زیادی نیز به فارسی در مجله pmm سایت iranmicro و انجمن های این سایت و سایت eca.ir موجود است.

۴- ادامه

یک راهنمای نسبتاً جامع از مقدماتی تا پیشرفته برای ARM از جمله شامل استفاده از نرم افزارهای GNU و همراه با دیگر طرح بردهای کامل تر و راه اندازی ماژول های مختلف، راه اندازی RTOS ها و ... در دست تهیه است.

۵- منابع

۱-۵- منابع طراحی مدار و PCB:

- برد NXP_DEV از آقای بهنام اسکندریون که تو انجمن ARM سایت eca.ir هست.
- Keil MCB2300 Evaluation Boards -
- CP-JR ARM7 LPC2368 Microcontroller Board -

۲-۵- دیگر منابع:

- مستندات بسیار مفید LPC1700 و LPC2300 از سایت NXP (Datasheets, Application Notes, Sample Codes , ...)
- مستندات Keil -

۳-۵- منابع مفید

- The Insider's Guide to the NXP 2300/2400 Base Microcontrollers, An Engineer's Guide to the LPC2300/2400 Series, www.hitex.co.uk, 2007.
- The Definitive Guide to the ARM Cortex-M3, Joseph Yiu, 2007, Elsevier Inc.
- ARM System Developer's Guide, Andrew N. Sloss and et al, 2004 by Elsevier Inc.
- ARM System-On-Chip Architecture, Steve Furber.
- مستندات سایت ARM برای ARM7TDMI-S و Cortex-M3
- انجمن های ARM از سایت های eca.ir و iranmicro.ir
- کارهای خوب آقای علی تروشه در زمینه ARM ، LPC2378 و ...، electrorc.blogfa.com

ضمیمه ۱- چرا ARM؟!

الف) گسترده‌ی شرکت‌های ارائه دهنده چیپ‌های با هسته ARM، در دسترس بودن و قیمت مناسب (دو پارامتر اخیر به نظر می‌رسد که در ایران در حال بهبودی هرچه بیشتر است و باید باور کنیم بیشتر بیشتر به خود ما طراحان بستگی دارد تا به وارد کننده‌ها و فروشندگان‌ها. ما اگر بخواهیم از محصولات حساسی و با کیفیت استفاده کنیم، اون‌ها هم به تقاضا نگاه می‌کنند....)

هسته‌های ARM را شرکت ARM تولید می‌کند و به صورت IP^۱ (کد HDL^۲ یا ماسک) به فروش می‌رساند. بعد شرکت‌های دیگری این هسته‌ها را خریداری کرده و از آن در تراشه‌های خود استفاده می‌کنند. نمونه‌ای از شرکت‌هایی که تراشه‌ها/محصولات با هسته ARM ارائه می‌دهند:

Philips(NXP), Sumsung, STMicroelectronics, Texas Instrument, Atmel, Actel, Altera, Luminart Micro, Intel, Toshiba, Sharp, Zilog, Nokia, CISCO, etc.

جالب است که تعدادی از این شرکت‌ها علاوه بر تولید چیپ، داری محصولات متنوع و معروف تجاری و صنعتی و نظامی و فضایی هستند و از تراشه‌های خودشان در این محصولات استفاده می‌کنند. آیا این لیست نمی‌تواند دلیلی بر اعتماد پذیری و قدرت هسته ARM باشد؟ در ضمن برای مثال، تراشه‌های

با هسته ARM7 از Philips و Atmel،

با هسته Cortex-M3 از Philips و STMicroelectronics،

با هسته Cortex-M0 از Philips و به صورت SoftCore روی FPGA های Altera و Actel

با هسته ARM9, ARM11 از Sumsung،

با هسته Cortex-A8 از Texas

به صورت تک تراشه یا دمو برد یا هر دو در ایران موجود است.

ب) گستره وسیع کاربرد تراشه‌های با هسته ARM در انواع سیستم‌های نهفته^۳، کنترلر هاردیسک‌ها، کارت‌های هوشمند، تقریباً تمام تلفن‌های همراه، PDA ها، NetBook ها، انواع تجهیزات شبکه مانند سوئیچ‌ها و روترها و بسیاری موارد دیگر.

تصویر زیر را از کتاب سازمان کامپیوتر پترسون-هنسی ۲۰۰۴ آوردم که سهم فروش پردازنده‌ها با معماری‌های مختلف را تا سال ۲۰۰۲ نشان می‌دهد.

(Computer Organization and Design, the hardware/software interface, Patterson-Hennessy)

^۱ (دارایی فکری) Intellectual Property

^۲ Hardware Description Language

^۳ embedded

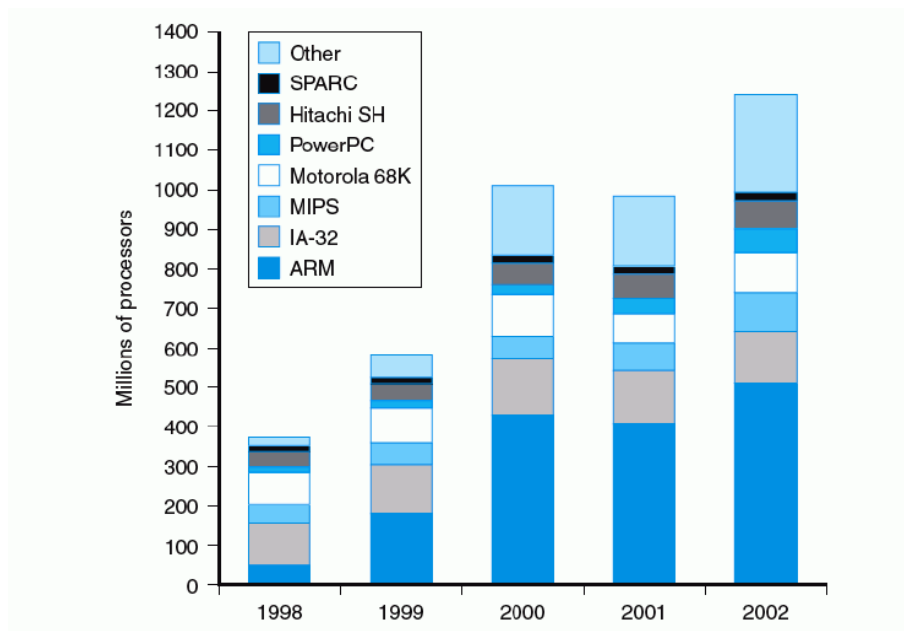


FIGURE 1.2 Sales of microprocessors between 1998 and 2002 by instruction set architecture combining all uses. The "other" category refers to processors that are either application-specific or customized architectures. In the case of ARM, roughly 80% of the sales are for cell phones, where an ARM core is used in conjunction with application-specific logic on a chip.

به سهم رو به رشد ARM نگاه کنید. این سهم بیشتر برای تلفن های همراه بوده، الان که ARM به 1GHz رسیده و از بازار سیستم های نهفته به بازار سرور وارد می شود آن هم با کارایی و مصرف توان بسیار کمتر نسبت به معماری هایی مانند X86. ویرایش های جدید تر کتاب معماری کامپیوتر استالینگ (فکر کنم ۲۰۰۷ به بعد) علاوه بر بحث مفصل معماری X86، بخشی را به بررسی معماری ARM اختصاص داده. خلاصه تنوع کاربرد ARM در کلاس میکروکنترلر و میکروپروسسور قابل مقایسه با هیچ محصول و معماری دیگری نیست. در واقع از بین معماری های عمده سیستم های نهفته مانند X86، PowerPC، و ARM، سومی گستره وسیع تری، از میکروکنترلر های کوچک تا پردازنده های به نسبت پر قدرت را پوشش می دهد و به ویژه با معرفی نسخه آسنکرون آن، AMULET، این طیف کامل تر شد.

جمع بندی این که، اولاً محصولاتی مانند AVR8,32، PIC8,16,32، XMEGA^۴ در بازار میکروکنترلر، بدون توجه به بحث مقایسه کارایی و کیفیت، توسط یک شرکت خاص ارائه می شود در حالیکه وسعت شرکت های ارائه دهنده چیپ های با هسته ARM آن هم شرکت های درست و حسابی، و ثانیاً طیف وسیع و منحصر به فرد محصولات ARM از میکروکنترلرهای ساده و ارزان تا پردازنده های با توانایی بالا انگیزه بسیار مناسبی برای انتخاب این چیپ ها به نظر می رسد.

(پ) قدرت و امکانات عالی این تراشه ها (یک SoC حسابی!)

- تراشه ای ۳۲ بیتی با قابلیت اجرای کدهای ۱۶ بیتی (کارایی در برابر فشردگی حجم کد)

- فرکانس کاری و کارایی بسیار مناسب

- وجود انواع امکانات به صورت System-on-Chip در محصولات با هسته ARM

برای نمونه به مشخصات NXP LPC1768 اشاره می کنیم و موارد مهم را بیشتر توضیح می دهیم.

⁴ RD = XMEGA یا Roa یعنی همون پیکان با موتور تقویت شده و به خرده امکانات جانبی! ☺

ARM **32-bit** Cortex-M3 Microcontroller
 with MPU,
 CPU clock up to **100MHz**,
512kB on-chip Flash ROM with enhanced Flash Memory Accelerator,
 In-System Programming (ISP),
 In-Application Programming (IAP),
64kB RAM,
 Nested Vectored Interrupt Controller,
Eight channel General purpose DMA controller,
 AHB Matrix,
 APB,
Ethernet 10/100 MAC with RII interface and dedicated DMA,
USB 2.0 full-speed Device controller and Host/OTG controller with DMA,
 CAN 2.0B with **two channels**,
Four UARTs, one with full Modem interface,
Three I2C serial interfaces,
Three SPI/SSP serial interfaces,
 I2S interface,
70 General purpose I/O pins,
12-bit ADC with **8 channels**,
 10-bit **DAC**,
Four 32-bit Timers with capture/compare,
 Standard PWM Timer block,
 Motor control PWM for three-phase Motor control,
 Quadrature Encoder,
 Watchdog Timer,
 Real Time Clock with optional Battery backup,
 System Tick Timer,
 Repetitive Interrupt Timer,
 Brown-out detect circuit,
 Power-On Reset,
 Power Management Unit,
 Wakeup Interrupt Controller,
 Crystal oscillator,
 4MHz internal RC oscillator,
 PLL,
 JTAG and Serial Wire Debug/Trace Port with ETM

هسته Cortex-M3 با کارایی بالا، مصرف توان مناسب و ویژگی های جدید برای جایگزینی محصولات با هسته ARM7 ارائه شده است. (AN10878 Migrating to the LPC1700 series را ببینید)
 در مورد سرعت و در واقع کارایی پردازنده (بر حسب MIPS معمولاً)، چند پارامتر با هم موثرند:
 - معماری 32-bit یعنی در یک سیکل، 32-bit داده می تواند واکنشی/پردازش شود (با معماری های 8-bit مقایسه کنید. جالب این که LPC1768 یک میکروکنترلر عادی هست نه یک پردازنده سرور!)
 - فرکانس کاری 100MHz هسته چیپ (100MHz واقعاً عالی! حتی LPC1769 تا 120MHz میره!!!)
 SoftCore ها روی FPGA های معمول، عموماً به حدود 50MHz یا کمتر می رسند، مثلاً PLASMA و YACC از opencores.org؛ مگر با بهینه سازی های مخصوص مدل FPGA مقصد و نیز بهینه سازی در سطح Place/route، مثل Xilinx MicroBlaze یا Altera NiosII. البته قدرت طراحی سخت افزارهای سفارشی به

صورت موازی با سخت افزار پردازنده ی اصلی، برای FPGA محفوظ هست ولی در ازای قیمت بیشتر، و مشکل و طولانی تر بودن فرآیند طراحی نسبت به میکروکنترلر؛ که انتخاب هر گزینه دقیقاً به کاربرد مورد نظر و نیازمندی های آن بستگی دارد).

- واحدی به نام Flash Memory Accelerator که سرعت واکنشی کد/داده را از حافظه فلش افزایش می دهد - میزان حافظه وسیع 64KB RAM

مورد جالب توجه دیگر، قابلیت برنامه ریزی PLL است که می تواند از طریق کد برنامه انجام شود لذا می توان برای رسیدن به نرخ مناسب یک ماژول خاص (مثلاً CAN) و حتی بسته به موقعیت، در حین برنامه برای مصالحه بین سرعت پردازش و مصرف توان کمتر، فرکانس هسته و به تبع آن فرکانس ماژول ها را تغییر داد.

قابلیت IAP نیز ویژگی شایان توجهی است و باعث می شود که در طول اجرای برنامه بتوان بخشی از فلش را تغییر داد، مفید برای استفاده به عنوان حافظه غیر فرار برای مصارفی مانند ثوابت کم تغییر و وابسته به کاربر. به این ترتیب دیگر نیازی به حافظه ای جداگانه مانند EEPROM داخلی یا خارجی نیست، از همان 512kB حافظه فلش برنامه با مزیت استفاده از قابلیت هایی مانند واحد تسریع دسترسی (Flash Memory Accelerator) می توان برای مصارف Lookup, log و ... نیز بهره برد.

خصوصیت مهم دیگر، واحد DMA است که برای تسریع مبادله داده بدون درگیری پردازنده و افزایش کارایی برنامه بسیار مفید است. مثلاً برای مبادله داده از طریق اترنت، کارت SD، USB، و غیره. در این تراشه، ۸ کانال DMA وجود دارد!

یکی از قابلیت های اساسی که سیستم عامل ها ارائه می دهند قابلیت اجرا و مدیریت همزمان چند کار یا فرآیند است. در سیستم های نهفته و کنترلی که سیستم عامل های نهفته یا بلادرنگ (RTOS⁵) را به کار می برند، این ویژگی معمولاً به صورت چند نخه (Multi-Thread) مطرح می شود^۶. پیاده سازی ویژگی چند نخه، در حالت بدون RTOS، در یک سیستم میکروکنترلی، معمولاً با استفاده مستقیم از تایمر ها و وقفه های مربوطه انجام می شود. کد نویسی (درست و کارا)، مدیریت، و توسعه چنین برنامه ای جز در موارد ساده، کاری است دشوار و مستعد خطا. این جاست که RTOS وارد می شود (البته نه فقط این جا!). از سوی دیگر، برای افزایش کارایی و امکان مدیریت صحیح و دقیق چند فرآیند، و جلوگیری از تداخل آن ها، وجود سخت افزار مناسب این کار در پردازنده ضروری است. این سخت افزار که معمولاً واحد مدیریت حافظه، MMU^۷ نام دارد و در برخی نسخه های ARM7 و نیز ARM9,11 و ...، موجود هست، بخش نسبتاً قابل توجهی از مساحت (ولتا قیمت) و مصرف توان تراشه را به خود اختصاص می دهد. واحد حفاظت حافظه، MPU^۸ ماژولی است که یک سری قابلیت های پایه ای MMU را ارائه می دهد که از یک طرف برای اجرای RTOS ها کافی است و از سوی دیگر، مساحت و مصرف توان کمتری را دارد و لذا ایده آل برای سیستم های میکروکنترلی می باشد. تراشه LPC1768 دارای یک واحد MPU است (این قابلیت از ویژگی های اختیاری هسته Cortex-M3 می باشد) لذا می توان RTOS هایی مانند FreeRTOS, uC/OS, eCos, RTEMS, TinyOS را روی تراشه های فاقد قابلیت های مدیریت حافظه قابل اجرا هستند را با قابلیت های بیشتری اجرا کرد و همچنین سیستم های عاملی مانند Linux و FreeBSD را نیز بر روی آن ها بالا آورد (به شرط تامین میزان حافظه مورد نیاز). برای نمونه پروژه LinuxM3، لینوکس را برای هسته Cortex-M3 پورت می کند.

⁵ Real Time Operating System

⁶ در این نوشته مفاهیم چند وظیفگی، چند کاری، چند فرآیندی و چند نخه تقریباً به یک معنا به کار می روند.

⁷ Memory Management Unit

⁸ Memory Protection Unit

تمام موارد بالا، انگیزه های مهمی در فراگیری و به کارگیری چیپ های ARM در مقایسه با دیگر محصولات (و حتی بدون مقایسه با آن ها!) می باشد.

در انتهای این بخش کمی هم درباره انتخاب چیپ های Philips:

- قابلیت ها و کارایی بسیار مناسب این چیپ ها نسبت به قیمت و دسترس پذیری و عمومیت نسبی آن ها
- مستندات و مثال های بسیار مفید و واضح

- پشتیبانی خوب Keil از این تراشه ها (هم مثال ها و هم دمو بردها)

- ارائه توسط یک شرکت معتبر؛ همچنین شرکت Philips علاوه بر تولید چیپ، دارای محصولات متنوع تجاری و صنعتی است و در آن ها از همین تراشه ها استفاده می کند که به نظر می آید این یک مزیت عمده برای این تراشه ها، نسبت به چیپ های شرکت هایی مانند Atmel است. (بحث مقایسه امکانات، قدرت، و قیمت تراشه و دمو بردها که بماند!)