



سازمان برنامه ریزی و توسعه فناوری اطلاعات
سازمان برنامه ریزی و توسعه فناوری اطلاعات

این برنامه برای
برنامه ریزی و توسعه فناوری اطلاعات

```
1 #include <Atmel/ioat91sam7s64.h>
```

سایر کتابخانه های مورد نیاز

```
2 #define (نیاز) (مورد نیاز) #define low AT91C_BASE_PIOA->PIO_CODR=(1<<2);
3 #define ON High AT91C_BASE_PIOA->PIO_SODR=(1<<2);
```

```
4 void main()
5 {
```

```
6 start-up
7 speed-up
```

```
8 while(1)
9 {
```

$$PLLCLKOUT = PLLCLKIN * \left(\frac{MUL+1}{DIV} \right)$$

تعیین start-up :
تایمر تکمیل خاموشی و ریست خارج فعال شود

```
AT91C_BASE_WDTC -> WDTC_WDMR = AT91C_WDTC_WDDIS; watch dog disable
```

```
AT91C_BASE_RSTC -> RSTC_RMR = 0xA5000000; ریست خارج فعال
```

تذکره: برای کدهای برنامه های که در سیستم این دو خط را برابر start-up بنویسیم.

تعیین speed-up :

مکانی: فرکانس اصلی از فرکانس کریستال خارج می‌شود و به بلورینده (Xtal) 18,432 MHz فرکانس می‌دهد.
 اصل برابر 48 MHz فاش (PLL فعال شود) generator

AT91C-BASE-CKGR → CKGR-MOR = 0x0F0; ;
 حالت استثناء از کریستال خارج می‌شود و دوباره XIM و XOM فعال می‌شود. state-up

AT91C-BASE-CKGR → CKGR-PLLR = 0x2903FFC;
 تنظیم PLL: $PLLCLKout = PLLCLKin \cdot \frac{MUL+1}{DIV}$
 تنظیم مقادیر MUL و DIV (18,432 MHz → 18,432 MHz) master clock

AT91C-BASE-PMC → PMC-MCKR = 0x03;
 منبع کلاک اصلی خروجی کلاک PLL است. اگر کلاک سایر کلاک‌ها کریستال باشد سیگنال 0x03 در تنظیم و در غیر این صورت خط می‌کشد.

AT91C-BASE-PMC → PMC-SCER = 0x01;
 کلاک CPU فعال می‌شود، کلاک استثنای آن می‌شود. system clock enable

تذکره: برای کلاک برنامه‌ها هر فرسبم این خط است تنظیم کلاک (مقادیر 0x03) نیاز می‌کشد.
 تذکره: رجیسترهای PMC کلاک مربوط به clock می‌کشد.

Handwritten signature or mark.



استفاده از کامپایلر IAR :

فهرست نرم افزار
فهرست نرم افزار هم بر سر CD تصویر
Auto Run و هم در کدها
ص ۱۴۸ تصویر

استفاده از کامپایلر IAR
آماره ساز شرایط اولیه
۱- محیط ایجاد پروژه جدید و اضافه کردن آن به workspace (فصل کار)

۱- File \ New \ workspace
(معمولاً با باز کردن برنامه ایجاد می شود)

۲- Project \ Creat New Project
۲- ایجاد پروژه جدید

۳- در پنجره باز شده مقابل tool chain با ARM انتخاب و گزینه Empty Project
انتخاب
select کردن. و با OK از کار صادر، سپس پروژه را مشخصی انتخاب کنید اسم دلخواه بزنید
انتخاب کنید. "Project 1" اسم اصلی برنامه. قبلاً فایل bin را آن پروگرام می کرد
در پنجره Release استخراجه فایل out " out File " در پنجره Release استخراجه فایل

۴- در پنجره workspace میله کلیک کنید Release را انتخاب کنید

۵- File \ save workspace
۵- save پنجره workspace با کلیک بر project

۶- انتخاب نوع میکرو، زبان برنامه نویسی، گزینه برنامه نویسی

۶- Project \ options
۶-۱- در زبان General
Target در منوی Device میکرو خود را از لیست انتخاب کنید
AT91sam7s64

۶-۲- در حالت C/C++ در زبان Language حالت را انتخاب کنید
Compiler
C/C++ در زبان Language حالت را انتخاب کنید
Compiler
C/C++ در زبان Language حالت را انتخاب کنید
Compiler
C/C++ در زبان Language حالت را انتخاب کنید



۳۴

کوتاه صفحه

۴-۲ - در حالت output converter فایل خروجی را بصورت binary منتقل کنید
↓
generate additional output
* قبل از آن

۴-۳ - در حالت linker از زبان config باید فایل دستور linker که مشخص کننده

اندازه و مکان بخش حافظه SRAM, Flash Ram و ... من باشد را مشخص کنیم
برای اینکار باید فایل را در محل پروژه ذخیره کرده و از طریق زبان config آنرا آدرس دهی

مثال: AT91SAM7S64_FLASH.icf

۷ - workspace را مجدداً save می‌کنیم
File \ save workspace
مثال: AT91SAM7S64_FLASH.icf
در workspace موجود می‌باشد
Example → Atmel → att91sam7s64
generic.icf را در محل نصب درج می‌کنیم
کنیم و آدرس دهی - (البته قبل از اینکار)

۸ - File \ New \ File
File \ save
main.cpp
انتخاب و سپس آنرا save می‌کنیم
نام و پسوند .cpp (یعنی C++) را برای آن انتخاب می‌کنیم
اسم دلخواه main.cpp

۱۰ - Project \ Add file
این کار را از طریق منوی

۱۱ - File \ save workspace
به پروژه اضافه می‌کنیم و workspace را save می‌کنیم
تذکره: اگر فایل از قبل داشته باشیم نیاز به ساخت فایل جدید نداریم و می‌توانیم آنرا مستقیم

```
#include <Atmel/ia91sam7s64.h>
void main()
{
    start_up;
    speed_up;
    AT91C_BASE_PMC->PMC_PCER = (1 << AT91C_ID_PIOA);
    AT91C_BASE_PIOA->PIO_PER = (1 << 0);
    AT91C_BASE_PIOA->PIO_OER = (1 << 0);
    AT91C_BASE_PIOA->PIO_SODR = (1 << 0);
    while(1)
    {}
}
```

Add file
پس اول فرم فرستاده شود
برنامه را بنویسید و اجرا کنید
مثال پین PA.0 را یک کنید
پس PA.0 بعنوان ورودی و خروجی
در هر خروجی
در برابر یک (۲، ۳، ۴) می‌شود
فعال سازید
پس PA.0
۱۱۰۰۰۱
ن در ۲ نوبت
پس PA.0 را یک کنید
پس PA.0 بعنوان ورودی و خروجی
در هر خروجی
در برابر یک (۲، ۳، ۴) می‌شود

POS

تست سازه اشکال برای برنامه

پیکره بندی Debug را از بخش work space انتخاب می کنیم

تذکره: برای هر مسیله Project \ Options در قسمت: Debugger حالت Simulator و حالت = گزینه RAM to main انتخاب شده باشد. (پس نرفتن فعال است)

از منو Project گزینه Debug را انتخاب می کنیم

تذکره: از منو View مرتباً انواع ^{ویژگیها} Tools ها را مشخص کرده

۱- برای دیدن متغیرها View \ watch ← نام متغیر را وارد می کنیم، مقدار آن در هر خط برنامه با نام مقابل آن نمایش داده می شود

۲- View \ Register ^{حسب هر متغیر} CPU

و سایر رجیسترها

برای دیدن سایر رجیسترها و نیز حالت

workSpace

Release

تذکره: برای اجرای Program از منوی فایل .bin از داخل پروژه

Release \ exe Load می کنیم توسط برنامه SAM-BA

با پورت Com یا پورت USB مطابق توضیحات کتاب ص ۱۷۰ به بعد