TINY\_AVR 产品开发注意事项

1. FUSE

A, BOD

BODLEVEL: 根据使用的电源选择2.7V 还是4.3V或1.8V ，一般建议5V系统选择2.7V以上，3.3V系统选择1.8V以上，这个主要由整板的系统设计决定，BODLEVEL 定义的是系统工作电压，当外部电源电压低于设定值时，MCU进入RESET状态，以防止部分外设或外部信号在设定电压以下可能导致的出错。

B, WatchDog

作为产品，我们建议使用WatchDog防止由于干扰或程序缺陷导致的不可预知的后果，这个可以WDTON 来使能， 当WDTON使能后，程序要定期刷新WDT 寄存器，如果超时，MCU会被复位。

ASF 没有给出完整的watchdog驱动， 需自己添加部分驱动，以下是两种关watchdog的方式：

void WDT\_off(void)

{

#if 0

//DISABLE\_INTERRUPTS();

ENTER\_CRITICAL(UNUSED);

\_\_watchdog\_reset();

/\* Clear WDRF in MCUSR \*/

MCUSR &= ~(1<<WDRF);

/\* Write logical one to WDCE and WDE \*/

/\* Keep old prescaler setting to prevent unintentional time-out \*/

WDTCSR |= (1<<WDCE) | (1<<WDE);

/\* Turn off WDT \*/

WDTCSR = 0x00;

//ENABLE\_INTERRUPTS();

EXIT\_CRITICAL(UNUSED);

#else

MCUSR &= ~(1 << WDRF);

protected\_write\_io((void \*)&WDTCSR,

(1 << WDCE) | (1 << WDE),

(0x00 /\* Oscillator Cycles 2K \*/) | (0 << WDE /\* Watch Dog Enable: enabled \*/)

| (0 << WDIE /\* Watchdog Timeout Interrupt Enable: disabled \*/));

#endif

}

喂狗：ATmega324pb 没有watchdog Timer 寄存器，但有专门的清狗指令：

#define \_\_watchdog\_reset() \_\_asm\_\_ \_\_volatile\_\_ ( "wdr" ::: "memory")

修改watchdog 溢出周期，修改程序中WDP3，WDP2，WDP1，WDP0对应值（0/1）：

void WDT\_Prescaler\_Change(void)

{

ENTER\_CRITICAL(UNUSED);;

\_\_watchdog\_reset();

/\* Start timed sequence \*/

WDTCSR |= (1<<WDCE) | (1<<WDE);

/\* Set new prescaler(time-out) value = 64K cycles (~0.5 s) \*/

WDTCSR = (1<<WDE) | (1<<WDP3);

EXIT\_CRITICAL(UNUSED);

}

C, 在程序中设置fuse 及eeprom

Fuse 可以预先定义在程序中，如果有预先要定义的存储在EEPROM 的数据，可参考如下：

#if 1

// Example data for ATmega324PB

const char eeprdata[] \_\_attribute\_\_ ((section (".eeprom"))) =

"Hello EEPROM";

// The order of the fuse values is from low to high. 0x42 is written to Fuse byte 0, 0x81 to byte 1...

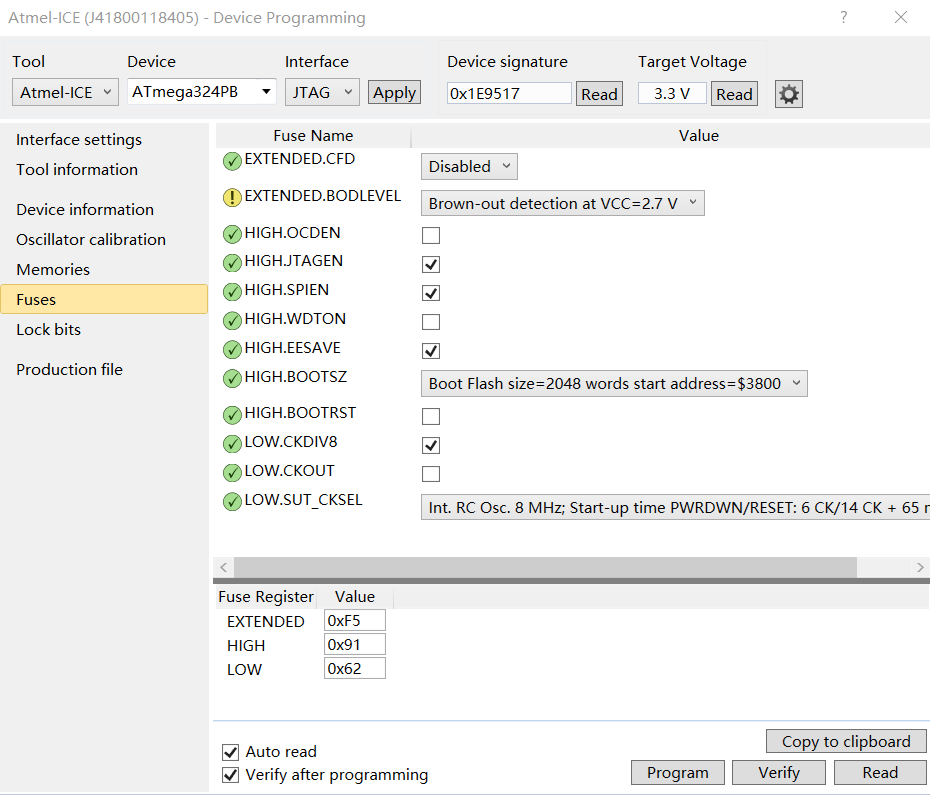
const *uint8\_t* fusedata[] \_\_attribute\_\_ ((section (".fuse"))) =

{0x42, 0x81, 0xF5};

//const char userdata[] \_\_attribute\_\_ ((section (".user\_signatures"))) =

//"Hello User Signatures";

#endif



2， 量产最终目标文件

以上定义的FUSE及EEPROM 在编译后不会体现在HEX文件中， 但会存在于ELF文件中，可以用直接生成的elf文件用于量产烧写，但这个文件包含可调试信息，不利于程序的保密。可以图中Save to ELF production file 部分重新生成elf文件（包括下面的Flash，EEPROM，Fuses等选项）。

