ATMFLASH

JEDNOTKA PRO PRÁCI S PAMĚTMI FLASH FIRMY ATMEL

Příručka uživatele a programátora

***SofCon*®spol. s r.o.**
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: http://www.sofcon.cz

Verze dokumentu 3.00

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.
SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsané. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.
SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 08.01.2004
Datum posledního uložení dokumentu: 08.01.2004
(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:
V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
2.00	1.XX	Hv		První vydání.
2.10	2.XX	Tu	20.05.2003	Úprava dokumentu dle ISO9000
3.00	3.XX	Hv	05.01.2004	Změna deklarace funkcí, přidán prefix „**a“.** Zveřejnění proměnných ovlivňujících zápisové funkce. Funkce ATMWrite, ATMDestruct, ATMTotalDestruct neprovádí identifikaci paměti. Přidán příznak pro omezení zápisu stejných dat, používá LOADER. Oprava identifikace AT29C040A v 8b režimu. Zatím neopraven problém s identifikací AT29C040A v 16b režimu. Tuto chybu lze obejít pomocí nastavení nově zveřejněných konstant..

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky pro práci s pamětí FLASH firmy ATMEL.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, ale je potřeba se orientovat v používání programového vybavení SofCon.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

Hlavní předností pamětí FLASH je jejich schopnost pamatovat si uložená data i po vypnutém napájení. To předurčuje tyto paměti pro uchovávání konstant, dat a programů, které umožní snadný update na novou verzi.

V jednotce je poskytováno jednotné rozhraní pro paměti firmy ATMEL AT29C010, AT29C020, AT29C040 a AT29C040A, které umožní pracovat s pamětí FLASH jako se pracuje s pamětí RAM.

Rozhraní poskytuje následující funkce

- pro zápis nejmenších možných bloků, tzv. sektorů majících délku závislou na typu paměti.
- pro zápis paměťových bloků.
- pro identifikaci paměti a výrobce.
- pro vratné a nevratné poškození vybraného sektoru paměti. Tyto funkce jsou používány aplikačními programy, které se v běžném provozu automaticky spouštějí, ale zároveň za výjimečných situacích umožňují možnost zápisu nové verze, např. pomocí RETOS Debuggeru.

Na závěr bychom Vás chtěli upozornit na omezený počet zápisů do jednoho sektoru paměti FLASH a na jejich odezvu při neautorizovaném pokusu o zápis dat.

- V době vydání této příručky uváděla firma ATMEL ve svých manuálových listech pro paměti AT29C010, AT29C020, AT29C040 a AT29C040A min. hodnotu 10000 zápisů na jeden sektor. Proto doporučujeme provádět zápisy pomocí kruhového bufferu, který tento min. počet mnohonásobně zvýší.
- Aby nedošlo k poškození uložených dat, používají zápisové funkce této knihovny po každém ukončeném nahrávání dat do jednoho sektoru uzamykání celé paměti proti neautorizovanému zápisu pomocí algoritmu SDP (Software Data protection). Tato ochrana může v *„chodivé“* aplikaci v paměti EPROM případně simulátoru EPROM při neautorizovaném pokusu o zápis do této paměti při přenosu aplikace do paměti FLASH způsobovat její padání. Tento stav je způsoben právě touto ochranou, která způsobuje, že při každém neautorizovaném pokusu o zápis dat dojde k zablokování paměti na dobu 10 ms. V tomto okamžiku nelze z paměti číst aplikační program, což vede k jeho zhroucení.

4. Popis konstant a typů

```
cVerNo = např. $0251; { BCD formát }
cVer   = např. '02.51,07.08.2003';
```

Číslo verze jednotky v BCD tvaru a v textové podobě včetně datumu změny.

```
cMaxSectorLength  = 1024     konstanta určuje max. velikost sektoru,
                             s kterým umí tato jednotka pracovat.
                             Max. délka sektoru je určena pamětí
                             FLASH typu 29C040 při 16b přístupu.

cCheckBIOS : Boolean         proměnná zapíná kontrolu zda zápis do
                             paměti FLASH nemůže poškodit BIOS. Je-li
                             při zapnuté kontrole zjištěno, že zápis
                             by poškodil BIOS, je zápis neproveden a
                             je vrácena hodnota FALSE. Při vypnuté
                             kontrole je zápis vždy proveden.
```

	Implicitně je tato proměnná nastavena na TRUE.
cRepeat : Byte	proměnná udává počet opakovaných zápisů do jednoho sektoru, v případě že první zápis neproběhl bez chyb. Implicitně je tato hodnota nastavena na 2. Max. doporučená hodnota je 10.
cDelay : Word=$2000;	v této proměnné je uložena konstanta určující dobu pro zápis dat do paměti FLASH. Po tuto dobu paměť FLASH je zablokována a nelze z ní číst. Tato proměnná se nastaví podle typu procesoru a přístupu k paměti v inicializační části jednotky.
cFlag16b : Byte=$0;	v této proměnné je uložena konstanta určující přístup k pamětem FLASH – 8/16b. Implicitně je nastaven 8b přístup. Tato proměnná se nastaví podle přístupu k paměti v inicializační části jednotky.
cCPUFlag : Byte=$0;	v této proměnné je uložen typ procesoru. Tato proměnná se používá v inicializační části jednotky a při identifikaci paměti pro nastavení časů případně typu přístupu k paměti. Tato proměnná se nastaví podle typu procesoru v inicializační části jednotky.
cLenSect : Word=512;	v této proměnné je uložena velikost sektoru. Implicitně je velikost sektoru nastavena na 512B. Tato proměnná se nastaví při volání funkce ATMIdent.
cManDev : Word=0;	v této proměnné je uložena identifikace paměti. Tato proměnná se nastaví při volání funkce ATMIdent.
fUseMemCmp:Boolean = FALSE;	v této proměnné je uložen příznak, zda před zápisem do paměti FLASH provést porovnání paměti. V případě jeho nastavení a stejných dat se zápis neprovede. Při aktivaci tohoto příznaku se v některých případech sníží počet zápisů do paměti FLASH a zvýší se rychlost zápisu. (pozn. doba zápisu zpravidla činí 10ms)

5. Funkce

5.1. ATMIdent

Popis:
Funkce vrací pro paměti firmy Atmel typ paměti, identifikační kód výrobce a délku sektoru.
Syntaxe:
function ATMIdent(aBase: LongInt; var aManDev, aLen: Word):Boolean;
Vstup:

aBase udává bázovou adresu paměti FLASH v adresovém prostoru procesoru.

Výstup:

Funkce vrací TRUE, pokud identifikace byla úspěšná a výrobcem paměti je firma ATMEL (Lo(aManDev) musí mít hodnotu $1F). V tomto případě obsahují hodnoty v proměnných aManDev, aLen, cLenSect (nastavena na hodnotu aLen) a cManDev (nastavena na hodnotu aManDev) platné údaje.
V opačném případě vrací funkce FALSE, proměnná aManDev obsahuje neplatnou hodnotu a proměnné aLen (nastavena na hodnotu cLenSect), cLenSect a cManDev obsahují poslední nastavenou hodnotu zpravidla implicitní.

Pozn. Proměnná cLenSect je dále používána v dalších funkcích.

Při úspěšné identifikaci jsou proměnné aManDev a aLen nastaveny na dále popsané hodnoty.

aManDev udává typ paměti a výrobce. Jak už název proměnné napovídá horní byte udává výrobce a spodní byte udává výrobce paměti. (Pozn. Pro hodnoty proměnné cManDev platí stejné pravidlo.)

Hi(aManDev) typ paměti

Knihovna je určena pro paměti firmy ATMEL s těmito identifikačními čísly:

29C010	*$D5*
29C020	*$DA*
29C040	*$5B*
29C040A	*$A4*

Lo(aManDev) výrobce paměti

Identifikační číslo firmy ATMEL je $1F. Pro paměti s výše uvedenými identifikačními čísly a jejichž výrobce je firma ATMEL, lze používat všechny funkce této knihovny. V případě že budete používat paměť jiného výrobce, tak návratový kód rutin používající funkci ATMIdent bude neustále vracet FALSE.

aLen udává délku nejmenší možné oblasti, kterou lze zapsat do paměti FLASH. Tato jednotka se nazývá sektor.
(Pozn. Pro hodnoty proměnné cLenSect platí stejné pravidlo.)
Knihovna vrací pro uvedené paměti následující hodnoty:

29C010	*128B*
29C020	*256B*
29C040	*512B*
29C040A	*256B*

V případě jiného typu paměti nebo výrobce je tato hodnota nastavena na hodnotu cLenSect, což je zpravidla implicitní hodnota.

5.2. ATMWriteSec

Popis:
Funkce zapíše sektor z lineární adresy aSour (musí být v paměti RAM) na adresu aDest v paměti FLASH. Délka tohoto sektoru je zadána v proměnné aLenSect. Vypočtené lineární adresy by měly být dělitelné délkou sektoru, jinak hrozí ztráta zbývajících dat v sektoru.
Výpočet adresy sektoru z lineární adresy (Addr)

(Addr div aLenSect)*aLenSect

Výpočet adresy sektoru z bázové adresy (seg:off)

(((seg shl 4)+off) div aLenSect)*aLenSect

Syntaxe:
function ATMWriteSec(aBase, aSour, aDest: LongInt; aLenSect: Word):Boolean;
Vstup:

aBase udává bázovou adresu paměti FLASH v adresovém prostoru procesoru.
aSour lineární adresa pole bytů, které bude zapsáno do sektoru paměti FLASH. Pole musí být uloženo v paměti RAM.
Pokud toto pole nebude v paměti RAM, nezapíší se požadované data, protože v okamžiku zápisu je paměť Write Only.
aDest lineární adresa sektoru, do kterého budou zadaná data zapsána.
aLenSect udává velikost sektoru. Tato velikost musí odpovídat velikosti sektoru pro daný typ paměti FLASH. Tato hodnota se získá pomocí funkce ATMIdent.

Výstup:

Funkce vrací True, pokud dále uvedené kontroly, zápis dat do sektoru a jeho verifikace proběhla bez chyb.

Kontroly před a během zápisu

- Před vlastním zápisem se provede kontrola přepisu BIOS. Jestliže je nastaven příznak cCheckBIOS a je zjištěn požadavek na přepis BIOS, funkce se ukončí s návratovou hodnotou FALSE a zápis se neprovede. Jestliže tato kontrola není zapnuta, požadovaný zápis se provede a nejspíše se poškodí BIOS.
- Po kontrole přepisu BIOS a při zapnuté kontrole obsahu paměti fUseMemCmp, se provede kontrola obsahu dat. Jestliže zapisovaná data jsou stejná jako data v paměti, funkce vrátí TRUE a zápis se neprovede.

- Jestliže při vlastním zápisu (nedostatek STACK) případně při kontrole dat (poškozená paměť FLASH) je detekována chyba, vrací se FALSE.

5.3. ATMWrite

Popis:
Funkce zapisuje paměťový blok z adresy aSour (zdrojový blok může být umístěn jak v paměti RAM tak v paměti FLASH, viz funkce ATMWriteSect) délky menší jak 64kB do paměti FLASH na adresu zadanou proměnnou aDest. Při zápisu zadaného bloku dat se zapisovaný sektor vždy uloží do pomocného bufferu vytvořeného na STACK o maximální velikosti a poté se zapíše pomocí funkce ATMWriteSect. Před vlastním zápisem je ošetřen jak začátek tak konec bloku dat, aby při samotném zápisu nedošlo ke ztrátě dat v zapisovaných sektorech. Tzn. začátek a konec zapisovaného bloku dat není násobkem velikosti sektoru.
Upozornění: Před prvním použitím této funkce se musí zavolat funkce ATMIdent nebo správně nastavit proměnnou cLenSect.
Syntaxe:
function ATMWrite(aBase, aSour, aDest, aLen: LongInt):Boolean;
Vstup:

aBase	udává bázovou adresu paměti FLASH v adresovém prostoru procesoru.
aSour	lineární adresa paměťového bloku, který bude zapsán do paměti FLASH.
aDest	lineární adresa v paměti FLASH, na kterou bude zapsán paměťový blok určený proměnnou aSour.
aLen	udává velikost paměťového bloku.

Výstup:

Funkce	vrací TRUE, pokud při zápisu jednotlivých sektorů paměťového bloku dat a jeho verifikaci pomocí funkce ATMWriteSect nedošlo k chybě. V opačném případě se vrací FALSE.

5.4. ATMDestruct

Popis:
Funkce provede inverzi prvního bytu sektoru zadaného lineární adresou.
Upozornění: Před prvním použitím této funkce se musí zavolat funkce ATMIdent nebo správně nastavit proměnnou cLenSect.
Upozornění: Pokud lineární adresa sektoru není celočíselně dělitelná délkou sektoru, je tato adresa oříznuta na nejbližší nižší adresu celočíselně dělitelnou velikostí sektoru, viz popis funkce ATMWriteSect.
Syntaxe:
function ATMDestruct(aBase, aDest: LongInt):Boolean;
Vstup:

aBase	udává bázovou adresu paměti FLASH v adresovém prostoru procesoru.
aDest	lineární adresa sektoru, ve kterém bude invertována první položka.

Výstup:

Funkce vrací TRUE, pokud při zápisu dat a jeho verifikaci pomocí funkce ATMWriteSect nedošlo k chybě. V opačném případě funkce vrací FALSE.

5.5. ATMTotalDestruct

Popis:

Funkce nastaví první byte sektoru zadaného lineární adresou na hodnotu $33.

Upozornění: Před prvním použitím této funkce se musí zavolat funkce ATMIdent nebo správně nastavit proměnnou cLenSect.

Upozornění: Pokud lineární adresa sektoru není celočíselně dělitelná délkou sektoru, je tato adresa oříznuta na nejbližší nižší adresu celočíselně dělitelnou velikostí sektoru, viz popis funkce ATMWriteSect.

Syntaxe:

function ATMTotalDestruct(aBase, aDest: LongInt):Boolean;

Vstup:

aBase udává bázovou adresu paměti FLASH v adresovém prostoru procesoru.

aDest lineární adresa sektoru, ve kterém bude invertována první položka.

Výstup:

Funkce vrací TRUE, pokud při zápisu dat a jeho verifikaci pomocí funkce ATMWriteSect nedošlo k chybě. V opačném případě funkce vrací FALSE.