



# MiDAS Family

Application Note #012 (AN012-V1.2)

## [MiDAS2.0] How to use IAP

### IAP (In-Application Programming) for FLASH (62KB) & EEPROM (2KB)

V1.2

August 2005

- ◆ CoreRiver Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice.
- ◆ Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
- ◆ The CoreRiver Semiconductor products listed in this document are intended for usage in general electronics applications. These CoreRiver Semiconductor products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury.

# Contents

1. Flow Chart : IAP
2. IAP (In-Application Programming)
3. Solution for Program Fail
4. Example
  - 1) Program Code Byte (FLASH)
  - 2) Program Code n Byte (FLASH)
  - 3) Erase Code Byte (FLASH)
  - 4) Erase Code Sector

# 1. Flow Chart : IAP (In-Application Programming)

## IAP Function Setting

: Refer to next slide how to use ACC, B, and DPTR.

### [ FLASH (62KB) ]

- ◆ Program Code Byte
- ◆ Erase Code Byte / Sector(256B)

### [ EEPROM (2KB) ]

- ◆ Program EEPROM Byte
- ◆ Erase EEPROM Byte / Sector(256B)



## Call IAP Routine

IAP Routine Access Enable (FAEN.0)  
Backup Stack Pointer (SP).  
SP = 0x07

LCALL IAP Routine (LCALL Address : FFF0h)

\* If user wants to program code/EEPROM byte more, set the additional byte, and then LCALL IAP routine again.

Restore Stack Pointer (SP).  
IAP Routine Access Disable (FAEN.0)



- ◆ FLASH for Code ROM (62Kbytes) can be programmed / erased during the operation of MCU.
- ◆ EEPROM for Data ROM (2Kbytes) can be programmed / erased during the operation of MCU.
- ◆ Program / Erase Time
  - ✓ Program : 50 usec except for IAP code execution time.
  - ✓ Erase : 10 msec except for IAP code execution time.
- ◆ Program / Erase Unit
  - ✓ Program : 1 byte
  - ✓ Erase : 1 byte or 1 sector (256 bytes)
- ◆ IAP SFR : Refer to FAEN (F7h)

## Check IAP Return Value

Success?

- Fail → Refer to ACC SFR ]
- ◆ Program Fail : FCh
  - ◆ Address Fail : FDh
  - ◆ Lock Fail : FEh
  - ◆ Command Fail : FFh

- Success → Refer to ACC SFR ]
- ◆ Erase Code Sector : 81h
  - ◆ Erase Code Byte : 82h
  - ◆ Program Code Byte : 83h
  - ◆ Erase EEPROM Sector : 84h
  - ◆ Erase EEPROM Byte : 85h
  - ◆ Program EEPROM Byte : 86h

## 2. IAP (In-Application Programming)

### ◆ Function Setting

- ✓ ACC (Accumulator) : - code/data byte to be programmed  
- Return Value
- ✓ B Register : - IAP Function Setting

### ◆ IAP SFR

- ✓ FAEN : IAP Routine Access Enable Register  
(Default Value = 00h)

#### ■ FAEN (F7h) : IAP Routine Access Enable Register

-	-	-	-	-	-	-	FLASH_AEN
							R/W(0)

- ◆ FLASH\_AEN : IAP routine access enable.

### ◆ Return Value

- ✓ Success : [ACC] 8Xh
- ✓ Program Fail : [ACC] FCh  
- It occurs if the data is written to the address whose content is not FFh (not blank).
- ✓ Address Fail : [ACC] FDh  
- It occurs when the incorrect address is assigned to DPTR in assembly. (or DPH & DPL in C).  
- Address Region
  - FLASH Region for Code ROM : 0000h ~ F7FFh
  - EEPROM Region for Data ROM : F800h ~ FFFFh

[ Table. IAP : Function Setting & Return Value ]

LCALL Address	Command	Function	B	ACC	DPTR (Address)	Used XRAM Area	Return Value (ACC)
FFF0h	Program	Program Code Byte <sup>1)</sup>	03h	Programmed code	FLASH (Byte)	No	83h / FCh / FDh / FEh / FFh
		Program EEPROM Byte	06h	Programmed data	EEPROM (Byte)	No	86h / FCh / FDh / FEh / FFh
	Erase	Erase Code Sector <sup>2)</sup>	01h	Don't care	FLASH (Sector)	No	81h / FDh / FEh / FFh
		Erase Code Byte <sup>3)</sup>	02h	Don't care	FLASH (Byte)	0600h ~ 06ffh	82h / FDh / FEh / FFh
		Erase EEPROM Sector	04h	Don't care	EEPROM (Sector)	No	84h / FDh / FEh / FFh
		Erase EEPROM Byte	05h	Don't care	EEPROM (Byte)	0600h ~ 06FFh	85h / FDh / FEh / FFh

- ◆ For examples, refer to [slide 6 & 7 for 1\)](#), [slide 8 for 2\)](#), and [slide 9 for 3\)](#). In additional, the example code is under "c:\gentos\example\MiDAS2.0"

## 3. Solution for Program Fail

### ◆ FLASH / EEPROM

- ✓ FLASH memory and EEPROM can change the value "1" to "0".  
But, they can't change the value "0" to "1".
- ✓ If the data is written to the address whose content is not FFh (not blank), it causes program fail.

### ◆ Solution for Program Fail

- ✓ Before write the data, erase the data of the address.  
(the data for the address → FFh)
- ✓ Then, user can write the data to that address.
- ✓ User can program the function.

## 4. Example : 1) Program Code Byte (FLASH)

### IAP Function Setting

#### [ FLASH (62KB) ]

- ◆ Program Code Byte (B = 0x03)

```
unsigned char at 0x08 backup_stack[3];
```

```
B = 0x03; // IAP Function Setting
ACC = 0x55; // Data to be Programmed
DPH = 0x80; // Address to be Programmed
DPL = 0x00; // DPTR = 0x8000
```

### LCALL IAP Routine

IAP Routine Access Enable (FAEN.0)  
Backup Stack Pointer (SP).  
SP = 0x07

```
FAEN = 0x01; // IAP Routine Access Enable
backup_stack[2] = SP; // Backup Stack Pointer (SP)
SP = 0x07; // SP = 7
```

LCALL IAP Routine (Call Address : FFF0h)

```
_asm // (In-line Assembly)
LCALL 0FFF0h // LCALL IAP Routine
_endasm
```

Restore Stack Pointer (SP).  
IAP Routine Access Disable (FAEN.0)

```
SP = backup_stack[2]; // Restore Stack Pointer (SP)
FAEN = 0x00; // IAP Routine Access Disable
```

### Check IAP Return Value

Success?

[ Success → Refer to ACC SFR ]

- ◆ Program Code Byte : 83h

[ Fail → Refer to ACC SFR ]

- ◆ Program Fail : FCh
- ◆ Address Fail : FDh
- ◆ Lock Fail : FEh
- ◆ Command Fail : FFh

```
If (ACC != 0x83) { // Fail : Program Code Byte
    // User Code (Action for Fail)
} else { // OK : Program Code Byte
    // User Code (Action for OK)
}
```

- ◆ To use the Data Pointer, user can use the DPTR in assembly.
- ◆ And user can use the DPH and DPL registers in C language.

## 4. Example : 2) Program Code n Byte (FLASH)

### IAP Function Setting

#### [ FLASH (62KB) ]

- ◆ Program Code Byte (B = 0x03)

```
unsigned char at 0x08 backup_stack[3];
```

```
B = 0x03; // IAP Function Setting
ACC = 0x55; // Data to be Programmed
DPH = 0x80; // Address to be Programmed
DPL = 0x00; // DPTR = 0x8000
```

### LCALL IAP Routine

IAP Routine Access Enable (FAEN.0)  
Backup Stack Pointer (SP).  
SP = 0x07

```
FAEN = 0x01; // IAP Routine Access Enable
backup_stack[2] = SP; // Backup Stack Pointer (SP)
SP = 0x07; // SP = 7
```

LCALL IAP Routine (Call Address : FFF0h)

```
_asm // (In-line Assembly)
LCALL 0FFF0h // Call IAP Routine
_endasm
```

Data (ACC), Address (DPTR) Assignment  
LCALL IAP Routine

```
ACC = 0xAA; // Data to be Programmed
DPL = 0x01; // DPTR = 0x8001
_asm // (In-line Assembly)
LCALL 0FFF0h // Call IAP Routine
_endasm
```

•  
•  
•  
n Bytes

Restore Stack Pointer (SP).  
IAP Routine Access Disable (FAEN.0)

```
SP = backup_stack[2]; // Restore Stack Pointer (SP)
FAEN = 0x00; // IAP Routine Access Disable
```

Check IAP Return Value

- ◆ User can insert code to program n bytes to FLASH / EEPROM using Data (ACC), Address Assignment (DPTR), & LCALL IAP Routine.
- ◆ User can code [the loop function](#) to program n bytes to FLASH / EEPROM.

## 4. Example : 3) Erase Code Byte (FLASH)

### IAP Function Setting

#### [ FLASH (62KB) ]

- ◆ Erase Code Byte (B = 0x02)

```
unsigned char at 0x08 backup_stack[3];
```

```
B      = 0x02;           // IAP Function Setting
DPH    = 0x80;           // Address to be Erased
DPL    = 0x00;           // DPTR = 0x8000
```



### LCALL IAP Routine

IAP Routine Access Enable (FAEN.0)  
Backup Stack Pointer (SP).  
SP = 0x07

```
FAEN = 0x01;           // IAP Routine Access Enable
backup_stack[2] = SP;  // Backup Stack Pointer (SP)
SP    = 0x07;           // SP = 7
```

LCALL IAP Routine (Call Address : FFF0h)

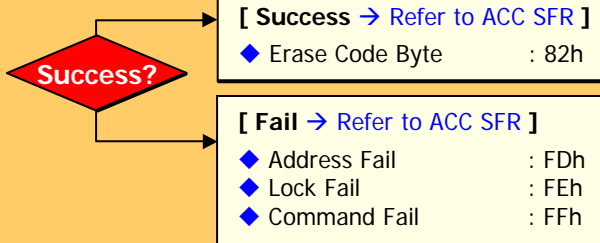
```
_asm                   // (In-line Assembly)
LCALL 0FFF0h           // LCALL IAP Routine
_endasm
```

Restore Stack Pointer (SP).  
IAP Routine Access Disable (FAEN.0)

```
SP    = backup_stack[2]; // Restore Stack Pointer (SP)
FAEN  = 0x00;           // IAP Routine Access Disable
```



### Check IAP Return Value



```
If (ACC != 0x82) {     // Fail : Erase Code Byte
    // User Code (Action for Fail)
} else {                // OK : Erase Code Byte
    // User Code (Action for OK)
}
```



## 4. Example : 3) Erase Code Sector (FLASH)

### IAP Function Setting

#### [ FLASH (62KB) ]

- ◆ Erase Code Sector (B = 0x01)

```
unsigned char at 0x08 backup_stack[3];
```

```
B = 0x01; // IAP Function Setting
DPH = 0x80; // Address to be Erased
DPL = 0x00; // DPTR = 0x8000
```



### LCALL IAP Routine

IAP Routine Access Enable (FAEN.0)  
Backup Stack Pointer (SP).  
SP = 0x07

```
FAEN = 0x01; // IAP Routine Access Enable
backup_stack[2] = SP; // Backup Stack Pointer (SP)
SP = 0x07; // SP = 7
```

LCALL IAP Routine (Call Address : FFF0h)

```
_asm // (In-line Assembly)
LCALL 0FFF0h // LCALL IAP Routine
_endasm
```

Restore Stack Pointer (SP).  
IAP Routine Access Disable (FAEN.0)

```
SP = backup_stack[2]; // Restore Stack Pointer (SP)
FAEN = 0x00; // IAP Routine Access Disable
```



### Check IAP Return Value

Success?

[ Success → Refer to ACC SFR ]

- ◆ Erase Code Sector : 81h

```
If (ACC != 0x81) { // Fail : Erase Code Sector
    // User Code (Action for Fail)
} else { // OK : Erase Code Sector
    // User Code (Action for OK)
}
```

[ Fail → Refer to ACC SFR ]

- ◆ Address Fail : FDh
- ◆ Lock Fail : FEh
- ◆ Command Fail : FFh

- ◆ Sector (256 bytes) Address is the upper byte of address (DPH).

- Example) Byte Address : F700h
- Example) Sector Address : F7h