Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	86
Program Memory Size	192KB (192K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	11K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g192f0cllr

Table 1-29. 48-Pin LQFP Pinout for S12GA192 and S12GA240

	Function <----lowest-----PRIORITY-----highest---->					Power Supply	Internal Pull Resistor	
Package Pin	Pin	2nd Func.	3rd Func.	4th Func	5th Func		CTRL	Reset State
2	VDDXR	—	—	—	—	—	—	—
3	VSSX	—	—	—	—	—	—	—
4	PE0[1]	EXTAL	—	—	—	V_{DDX}	PUCR/PDPEE	Down
5	VSS	—	—	—	—	—	—	—
6	PE1[1]	XTAL	—	—	—	V_{DDX}	PUCR/PDPEE	Down
7	TEST	—	—	—	—	N.A.	$\overline{\text{RESET}}$ pin	Down
8	PJ0	KWJ0	PWM6	MISO1	—	V_{DDX}	PERJ/PPSJ	Up
9	PJ1	KWJ1	IOC6	MOSI1	—	V_{DDX}	PERJ/PPSJ	Up
10	PJ2	KWJ2	IOC7	SCK1	—	V_{DDX}	PERJ/PPSJ	Up
11	PJ3	KWJ3	PWM7	$\overline{\text{SS1}}$	—	V_{DDX}	PERJ/PPSJ	Up
12	BKGD	MODC	—	—	—	V_{DDX}	PUCR/BKPUE	Up
13	PP0	KWP0	ETRIG0	API_EXTCLK	PWM0	V_{DDX}	PERP/PPSP	Disabled
14	PP1	KWP1	ETRIG1	ECLKX2	PWM1	V_{DDX}	PERP/PPSP	Disabled
15	PP2	KWP2	ETRIG2	PWM2	—	V_{DDX}	PERP/PPSP	Disabled
16	PP3	KWP3	ETRIG3	PWM3	—	V_{DDX}	PERP/PPSP	Disabled
17	PP4	KWP4	PWM4	—	—	V_{DDX}	PERP/PPSP	Disabled
18	PP5	KWP5	PWM5	—	—	V_{DDX}	PERP/PPSP	Disabled
19	PT5	IOC5	—	—	—	V_{DDX}	PERT/PPST	Disabled
20	PT4	IOC4	—	—	—	V_{DDX}	PERT/PPST	Disabled
21	PT3	IOC3	—	—	—	V_{DDX}	PERT/PPST	Disabled
22	PT2	IOC2	—	—	—	V_{DDX}	PERT/PPST	Disabled
23	PT1	IOC1	$\overline{\text{IRQ}}$	—	—	V_{DDX}	PERT/PPST	Disabled
24	PT0	IOC0	$\overline{\text{XIRQ}}$	—	—	V_{DDX}	PERT/PPST	Disabled
25	PAD0	KWAD0	AN0	—	—	V_{DDA}	PER1AD/PPS1AD	Disabled
26	PAD8	KWAD8	AN8	—	—	V_{DDA}	PER0AD/PPS0AD	Disabled
27	PAD1	KWAD1	AN1	—	—	V_{DDA}	PER1AD/PPS1AD	Disabled
28	PAD9	KWAD9	AN9		—	V_{DDA}	PER0AD/PPS0AD	Disabled
29	PAD2	KWAD2	AN2	—	—	V_{DDA}	PER1AD/PPS1AD	Disabled

Table 2-21. Block Register Map (G3) (continued)

Global Address Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x027C PIE0AD	R	0	0	0	0	PIE0AD3	PIE0AD2	PIE0AD1	PIE0AD0
	W								
0x027D PIE1AD	R	PIE1AD7	PIE1AD6	PIE1AD5	PIE1AD4	PIE1AD3	PIE1AD2	PIE1AD1	PIE1AD0
	W								
0x027E PIF0AD	R	0	0	0	0	PIF0AD3	PIF0AD2	PIF0AD1	PIF0AD0
	W								
0x027F PIF1AD	R	PIF1AD7	PIF1AD6	PIF1AD5	PIF1AD4	PIF1AD3	PIF1AD2	PIF1AD1	PIF1AD0
	W								

= Unimplemented or Reserved

2.4.3 Register Descriptions

This section describes the details of all configuration registers. Every register has the same functionality in all groups if not specified separately. Refer to the register figures for reserved locations. If not stated differently, writing to reserved bits has not effect and read returns zero.

NOTE

- All register read accesses are synchronous to internal clocks
- General-purpose data output availability depends on prioritization; input data registers always reflect the pin status independent of the use
- Pull-device availability, pull-device polarity, wired-or mode, key-wakeup functionality are independent of the prioritization unless noted differently in section Section 2.3, “PIM Routing - Functional description”.

Table 2-82. AN Routing Options

PRR1AN	Associated Pins
0	AN10 - PAD10 AN11 - PAD11 AN13 - PAD13 AN14 - PAD14 AN15 - PAD15
1	AN10 - PC0 AN11 - PC1 AN13 - PC2 AN14 - PC3 AN15 - PC4

2.4.3.57 Port AD Pull Enable Register (PER0AD)

Address 0x0278 (G1, G2) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PER0AD7	PER0AD6	PER0AD5	PER0AD4	PER0AD3	PER0AD2	PER0AD1	PER0AD0
Reset	0	0	0	0	0	0	0	0

Address 0x0278 (G3) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	PER0AD3	PER0AD2	PER0AD1	PER0AD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-56. Port AD Pullup Enable Register (PER0AD)

[1] Read: Anytime
Write: Anytime

Table 2-83. PER0AD Register Field Descriptions

Field	Description
7-0 PER0AD	**Port AD pull enable**—Enable pull device on input pin This bit controls whether a pull device on the associated port input pin is active. If a pin is used as output this bit has no effect. The polarity is selected by the related polarity select register bit. 1 Pull device enabled 0 Pull device disabled

Table 11-11. ATD Behavior in Freeze Mode (Breakpoint)

FRZ1	FRZ0	Behavior in Freeze Mode
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze Immediately

11.3.2.5 ATD Control Register 4 (ATDCTL4)

Writes to this register will abort current conversion sequence.

Module Base + 0x0004

Figure 11-7. ATD Control Register 4 (ATDCTL4)

Read: Anytime

Write: Anytime

Table 11-12. ATDCTL4 Field Descriptions

Field	Description
7–5 SMP[2:0]	**Sample Time Select** — These three bits select the length of the sample time in units of ATD conversion clock cycles. Note that the ATD conversion clock period is itself a function of the prescaler value (bits PRS4-0). Table 11-13 lists the available sample time lengths.
4–0 PRS[4:0]	**ATD Clock Prescaler** — These 5 bits are the binary prescaler value PRS. The ATD conversion clock frequency is calculated as follows: $f_{ATDCLK} = \frac{f_{BUS}}{2 \times (PRS + 1)}$ Refer to Device Specification for allowed frequency range of f_{ATDCLK}.

Table 11-13. Sample Time Select

SMP2	SMP1	SMP0	Sample Time in Number of ATD Clock Cycles
0	0	0	4
0	0	1	6
0	1	0	8
0	1	1	10
1	0	0	12
1	0	1	16
1	1	0	20
1	1	1	24

Table 14-5. External Trigger Channel Select Coding

ETRIGSEL	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0	External trigger source is
0	0	0	0	0	AN0
0	0	0	0	1	AN1
0	0	0	1	0	AN2
0	0	0	1	1	AN3
0	0	1	0	0	AN4
0	0	1	0	1	AN5
0	0	1	1	0	AN6
0	0	1	1	1	AN7
0	1	0	0	0	AN8
0	1	0	0	1	AN9
0	1	0	1	0	AN10
0	1	0	1	1	AN11
0	1	1	0	0	AN11
0	1	1	0	1	AN11
0	1	1	1	0	AN11
0	1	1	1	1	AN11
1	0	0	0	0	ETRIG0[1]
1	0	0	0	1	ETRIG1[1]
1	0	0	1	0	ETRIG2[1]
1	0	0	1	1	ETRIG3[1]
1	0	1	X	X	Reserved
1	1	X	X	X	Reserved

[1] Only if ETRIG3-0 input option is available (see device specification), else ETRISEL is ignored, that means external trigger source is still on one of the AD channels selected by ETRIGCH3-0

14.3.2.3 ATD Control Register 2 (ATDCTL2)

Writes to this register will abort current conversion sequence.

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	0	AFFC	Reserved	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ACMPIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-5. ATD Control Register 2 (ATDCTL2)

Read: Anytime

Write: Anytime

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0003	ATDCTL3	R	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		W								
0x0004	ATDCTL4	R	SMP2	SMP1	SMP0	PRS[4:0]				
		W								
0x0005	ATDCTL5	R	0	SC	SCAN	MULT	CD	CC	CB	CA
		W								
0x0006	ATDSTAT0	R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
		W								
0x0007	Unimple-mented	R	0	0	0	0	0	0	0	0
		W								
0x0008	ATDCMPEH	R	CMPE[15:8]							
		W								
0x0009	ATDCMPEL	R	CMPE[7:0]							
		W								
0x000A	ATDSTAT2H	R	CCF[15:8]							
		W								
0x000B	ATDSTAT2L	R	CCF[7:0]							
		W								
0x000C	ATDDIENH	R	IEN[15:8]							
		W								
0x000D	ATDDIENL	R	IEN[7:0]							
		W								
0x000E	ATDCMPHTH	R	CMPHT[15:8]							
		W								
0x000F	ATDCMPHTL	R	CMPHT[7:0]							
		W								
0x0010	ATDDR0	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0012	ATDDR1	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0014	ATDDR2	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0016	ATDDR3	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0018	ATDDR4	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001A	ATDDR5	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001C	ATDDR6	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001E	ATDDR7	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0020	ATDDR8	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0022	ATDDR9	R	See Section 15.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 15.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								

= Unimplemented or Reserved

Figure 15-2. ADC10B16C Register Summary (Sheet 2 of 3)

Table 15-15. Analog Input Channel Select Coding

SC	CD	CC	CB	CA	Analog Input Channel
0	0	0	0	0	AN0
	0	0	0	1	AN1
	0	0	1	0	AN2
	0	0	1	1	AN3
	0	1	0	0	AN4
	0	1	0	1	AN5
	0	1	1	0	AN6
	0	1	1	1	AN7
	1	0	0	0	AN8
	1	0	0	1	AN9
	1	0	1	0	AN10
	1	0	1	1	AN11
	1	1	0	0	AN12
	1	1	0	1	AN13
	1	1	1	0	AN14
	1	1	1	1	AN15
1	0	0	0	0	Internal_6,
	0	0	0	1	Internal_7
	0	0	1	0	Internal_0
	0	0	1	1	Internal_1
	0	1	0	0	VRH
	0	1	0	1	VRL
	0	1	1	0	(VRH+VRL) / 2
	0	1	1	1	Reserved
	1	0	0	0	Internal_2
	1	0	0	1	Internal_3
	1	0	1	0	Internal_4
	1	0	1	1	Internal_5
	1	1	X	X	Reserved

Chapter 17
Digital Analog Converter (DAC_8B5V)

17.1 Revision History

Table 17-1. Revision History Table

Rev. No. (Item No.)	Data	Sections Affected	Substantial Change(s)
1.0	12-Apr.-10	1.4.2.1	Added DACCTL register bit DACDIEN
1.01	04-May-10,	Table 1.2, Section 1.4	Replaced VRL,VRL with variable correct wrong figure, table numbering
1.02	12-May-10	Section 1.4	replaced ipt_test_mode with ips_test_access new description/address of DACDEBUG register
1.1	25-May-10	17.4.2.1	Removed DACCTL register bit DACDIEN
1.2	25-Jun.-10	17.4	Correct table and figure title format
1.3	29-Jul.-10	17.2	Fixed typos
1.4	17-Nov.-10	17.2.2	Update the behavior of the DACU pin during stop mode
1.5	29-Aug.-13	17.2.2, 17.3	added note about settling time added link to DACM register inside section 17.3

Glossary

Table 17-2. Terminology

Term	Meaning
DAC	Digital to Analog Converter
VRL	Low Reference Voltage
VRH	High Reference Voltage
FVR	Full Voltage Range
SSC	Special Single Chip

17.2 Introduction

The DAC_8B5V module is a digital to analog converter. The converter works with a resolution of 8 bit and generates an output voltage between VRL and VRH.

The module consists of configuration registers and two analog functional units, a DAC resistor network and an operational amplifier.

- Any CAN node is able to send out a stream of scheduled messages without releasing the CAN bus between the two messages. Such nodes arbitrate for the CAN bus immediately after sending the previous message and only release the CAN bus in case of lost arbitration.
- The internal message queue within any CAN node is organized such that the highest priority message is sent out first, if more than one message is ready to be sent.

The behavior described in the bullets above cannot be achieved with a single transmit buffer. That buffer must be reloaded immediately after the previous message is sent. This loading process lasts a finite amount of time and must be completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme de-couples the reloading of the transmit buffer from the actual message sending and, therefore, reduces the reactiveness requirements of the CPU. Problems can arise if the sending of a message is finished while the CPU re-loads the second buffer. No buffer would then be ready for transmission, and the CAN bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN implements with the “local priority” concept described in Section 18.4.2.2, “Transmit Structures.”

18.4.2.2 Transmit Structures

The MSCAN triple transmit buffer scheme optimizes real-time performance by allowing multiple messages to be set up in advance. The three buffers are arranged as shown in Figure 18-39.

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see Section 18.3.3, “Programmer’s Model of Message Storage”). An additional Transmit Buffer Priority Register (TBPR) contains an 8-bit local priority field (PRIO) (see Section 18.3.3.4, “Transmit Buffer Priority Register (TBPR)”). The remaining two bytes are used for time stamping of a message, if required (see Section 18.3.3.5, “Time Stamp Register (TSRH–TSRL)”).

To transmit a message, the CPU must identify an available transmit buffer, which is indicated by a set transmitter buffer empty (TXEx) flag (see Section 18.3.2.7, “MSCAN Transmitter Flag Register (CANTFLG)”). If a transmit buffer is available, the CPU must set a pointer to this buffer by writing to the CANTBSEL register (see Section 18.3.2.11, “MSCAN Transmit Buffer Selection Register (CANTBSEL)”). This makes the respective buffer accessible within the CANTXFG address space (see Section 18.3.3, “Programmer’s Model of Message Storage”). The algorithmic feature associated with the CANTBSEL register simplifies the transmit buffer selection. In addition, this scheme makes the handler software simpler because only one address area is applicable for the transmit process, and the required address space is minimized.

The CPU then stores the identifier, the control bits, and the data content into one of the transmit buffers. Finally, the buffer is flagged as ready for transmission by clearing the associated TXE flag.

20.4.4 Baud Rate Generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and the transmitter. The value from 0 to 8191 written to the SBR12:SBR0 bits determines the bus clock divisor. The SBR bits are in the SCI baud rate registers (SCIBDH and SCIBDL). The baud rate clock is synchronized with the bus clock and drives the receiver. The baud rate clock divided by 16 drives the transmitter. The receiver has an acquisition rate of 16 samples per bit time.

Baud rate generation is subject to one source of error:

- Integer division of the bus clock may not give the exact target frequency.

Table 20-16 lists some examples of achieving target baud rates with a bus clock frequency of 25 MHz.

When IREN = 0 then,

SCI baud rate = SCI bus clock / (16 * SCIBR[12:0])

Table 20-16. Baud Rates (Example: Bus Clock = 25 MHz)

Bits SBR[12:0]	Receiver Clock (Hz)	Transmitter Clock (Hz)	Target Baud Rate	Error (%)
41	609,756.1	38,109.8	38,400	.76
81	308,642.0	19,290.1	19,200	.47
163	153,374.2	9585.9	9,600	.16
326	76,687.1	4792.9	4,800	.15
651	38,402.5	2400.2	2,400	.01
1302	19,201.2	1200.1	1,200	.01
2604	9600.6	600.0	600	.00
5208	4800.0	300.0	300	.00

Only bits related to implemented channels are valid.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 TIOS	R	RESERVED	RESERVED	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
	W								
0x0001 CFORC	R	0	0	0	0	0	0	0	0
	W	RESERVED	RESERVED	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
0x0004 TCNTH	R	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
	W								
0x0005 TCNTL	R	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
	W								
0x0006 TSCR1	R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
	W								
0x0007 TTOV	R	RESERVED	RESERVED	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
	W								
0x0008 TCTL1	R	RESERVED	RESERVED	RESERVED	RESERVED	OM5	OL5	OM4	OL4
	W								
0x0009 TCTL2	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	W								
0x000A TCTL3	R	RESERVED	RESERVED	RESERVED	RESERVED	EDG5B	EDG5A	EDG4B	EDG4A
	W								
0x000B TCTL4	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	W								
0x000C TIE	R	RESERVED	RESERVED	C5I	C4I	C3I	C2I	C1I	C0I
	W								
0x000D TSCR2	R	TOI	0	0	0	RESERVED	PR2	PR1	PR0
	W								
0x000E TFLG1	R	RESERVED	RESERVED	C5F	C4F	C3F	C2F	C1F	C0F
	W								
0x000F TFLG2	R	TOF	0	0	0	0	0	0	0
	W								
0x0010–0x001F TCxH–TCxL[1]	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x0024–0x002B Reserved	R								
	W								
0x002C OCPD	R	RESERVED	RESERVED	OCPD5	OCPD4	OCPD3	OCPD2	OCPD1	OCPD0
	W								
0x002D Reserved	R								
0x002E PTPSR	R	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
	W								
0x002F Reserved	R								
	W								

Figure 22-3. TIM16B6CV3 Register Summary

24.4.6.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify that an entire P-Flash or EEPROM block has been erased. The FCCOB FlashBlockSelectionCode[1:0] bits determine which block must be verified.

Table 24-31. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Flash block selection code [1:0]. See Table 24-32

Table 24-32. Flash block selection code description

Selection code[1:0]	Flash block to be verified
00	EEPROM
01	Invalid (ACCERR)
10	Invalid (ACCERR)
11	P-Flash

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or EEPROM block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.If the block is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 24-33. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if an invalid FlashBlockSelectionCode[1:0] is supplied[1]
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read[2] or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read[2] or if blank check failed.

1 As defined by the memory map for FTMRG32K1.

2 As found in the memory map for FTMRG32K1.

24.4.6.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

Table 24-39. Program P-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 24-25)
		Set if an invalid global address [17:0] is supplied see Table 24-3)[1]
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the global address [17:0] points to a protected area
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

1 As defined by the memory map for FTMRG32K1.

24.4.6.6 Program Once Command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash. The Program Once reserved field can be read using the Read Once command as described in Section 24.4.6.4. The Program Once command must only be issued once since the nonvolatile information register in P-Flash cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 24-40. Program Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x07	Not Required
001	Program Once phrase index (0x0000 - 0x0007)	
010	Program Once word 0 value	
011	Program Once word 1 value	
100	Program Once word 2 value	
101	Program Once word 3 value	

Upon clearing CCIF to launch the Program Once command, the Memory Controller first verifies that the selected phrase is erased. If erased, then the selected phrase will be programmed and then verified with read back. The CCIF flag will remain clear, setting only after the Program Once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash will return invalid data.

26.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU.

Offset Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-9. Flash Configuration Register (FCNFG)

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 26-13. FCNFG Field Descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 26.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 26.3.2.8). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated
1 FDFD	**Force Double Bit Fault Detect** — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. 0 Flash array read operations will set the DFDIF flag in the FERSTAT register only if a double bit fault is detected 1 Any Flash array read operation will force the DFDIF flag in the FERSTAT register to be set (see Section 26.3.2.7) and an interrupt will be generated as long as the DFDIE interrupt enable in the FERCNFG register is set (see Section 26.3.2.6)
0 FSFD	**Force Single Bit Fault Detect** — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. 0 Flash array read operations will set the SFDIF flag in the FERSTAT register only if a single bit fault is detected 1 Flash array read operation will force the SFDIF flag in the FERSTAT register to be set (see Section 26.3.2.7) and an interrupt will be generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Section 26.3.2.6)

26.3.2.6 Flash Error Configuration Register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

27.3 Memory Map and Registers

This section describes the memory map and registers for the Flash module. Read data from unimplemented memory space in the Flash module is undefined. Write access to unimplemented or reserved memory space in the Flash module will be ignored by the Flash module.

CAUTION

Writing to the Flash registers while a Flash command is executing (that is indicated when the value of flag CCIF reads as ’0’) is not allowed. If such action is attempted the write operation will not change the register value.

Writing to the Flash registers is allowed when the Flash is not busy executing commands (CCIF = 1) and during initialization right after reset, despite the value of flag CCIF in that case (refer to Section 27.6 for a complete description of the reset sequence).

.

Table 27-2. FTMRG Memory Map

Global Address (in Bytes)	Size (Bytes)	Description
0x0_0000 - 0x0_03FF	1,024	Register Space
0x0_0400 – 0x0_0BFF	2,048	EEPROM Memory
0x0_4000 – 0x0_7FFF	16,284	NVMRES[1]=1 : NVM Resource area (see Figure 27-3)
0x3_0000 – 0x3_FFFF	65,536	P-Flash Memory

[1] See NVMRES description in Section 27.4.3

27.3.1 Module Memory Map

The S12 architecture places the P-Flash memory between global addresses 0x3_0000 and 0x3_FFFF as shown in Table 27-3.The P-Flash memory map is shown in Figure 27-2.

Table 27-3. P-Flash Memory Addressing

Global Address	Size (Bytes)	Description
0x3_0000 – 0x3_FFFF	64 K	P-Flash Block Contains Flash Configuration Field (see Table 27-4)

Table 27-25. FOPT Field Descriptions

Field	Description
7–0 NV[7:0]	**Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the device user guide for proper use of the NV bits.

27.3.2.17 Flash Reserved5 Register (FRSV5)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0011

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 27-23. Flash Reserved5 Register (FRSV5)

All bits in the FRSV5 register read 0 and are not writable.

27.3.2.18 Flash Reserved6 Register (FRSV6)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0012

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 27-24. Flash Reserved6 Register (FRSV6)

All bits in the FRSV6 register read 0 and are not writable.

27.3.2.19 Flash Reserved7 Register (FRSV7)

This Flash register is reserved for factory testing.

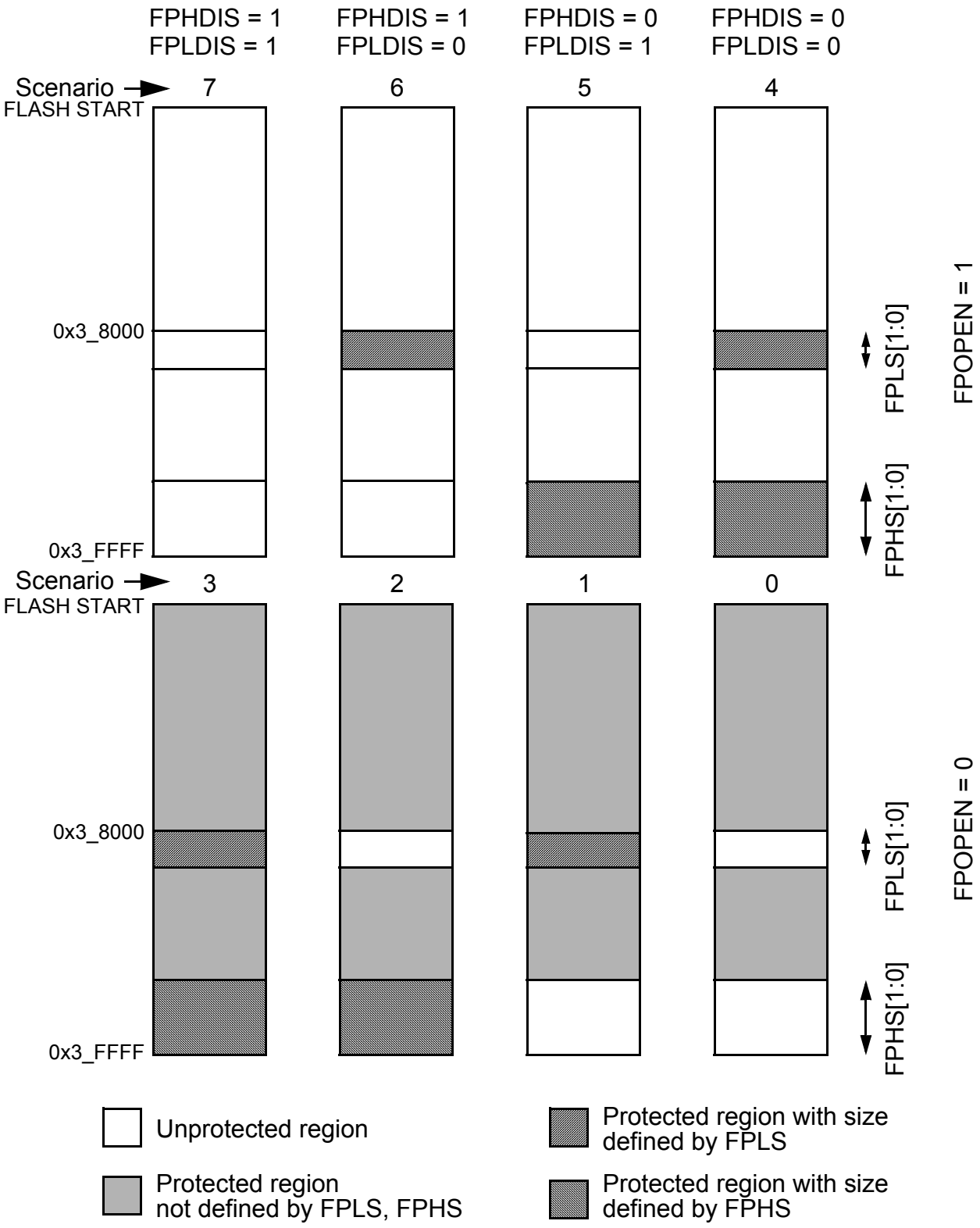

Figure 29-14. P-Flash Protection Scenarios

Address & Name		7	6	5	4	3	2	1	0
0x000A FCCOBHI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000B FCCOBLO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x000C FRSV1	R	0	0	0	0	0	0	0	0
	W								
0x000D FRSV2	R	0	0	0	0	0	0	0	0
	W								
0x000E FRSV3	R	0	0	0	0	0	0	0	0
	W								
0x000F FRSV4	R	0	0	0	0	0	0	0	0
	W								
0x0010 FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x0011 FRSV5	R	0	0	0	0	0	0	0	0
	W								
0x0012 FRSV6	R	0	0	0	0	0	0	0	0
	W								
0x0013 FRSV7	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 30-4. FTMRG192K2 Register Summary (continued)

30.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

indicated by reset condition F in Figure 30-6. If a double bit fault is detected while reading the P-Flash phrase containing the Flash security byte during the reset sequence, all bits in the FSEC register will be set to leave the Flash module in a secured state with backdoor key access disabled.

Table 30-9. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 30-10.
5–2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 30-11. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 30-10. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01	DISABLED[1]
10	ENABLED
11	DISABLED

1 Preferred KEYEN state to disable backdoor key access.

Table 30-11. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED[1]
10	UNSECURED
11	SECURED

1 Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 30.5.

30.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Offset Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCOBIX[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 30-7. FCCOB Index Register (FCCOBIX)

Table 30-14. FERCNFG Field Descriptions

Field	Description
1 DFDIE	**Double Bit Fault Detect Interrupt Enable** — The DFDIE bit controls interrupt generation when a double bit fault is detected during a Flash block read operation. 0 DFDIF interrupt disabled 1 An interrupt will be requested whenever the DFDIF flag is set (see Section 30.3.2.8)
0 SFDIE	**Single Bit Fault Detect Interrupt Enable** — The SFDIE bit controls interrupt generation when a single bit fault is detected during a Flash block read operation. 0 SFDIF interrupt disabled whenever the SFDIF flag is set (see Section 30.3.2.8) 1 An interrupt will be requested whenever the SFDIF flag is set (see Section 30.3.2.8)

30.3.2.7 Flash Status Register (FSTAT)

The FSTAT register reports the operational status of the Flash module.

Offset Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT[1:0]	
W								
Reset	1	0	0	0	0	0	0[1]	0[1]

= Unimplemented or Reserved

Figure 30-11. Flash Status Register (FSTAT)

[1] Reset value can deviate from the value shown if a double bit fault is detected during the reset sequence (see Section 30.6).

CCIF, ACCERR, and FPVIOL bits are readable and writable, MGBUSY and MGSTAT bits are readable but not writable, while remaining bits read 0 and are not writable.

Table 30-15. FSTAT Field Descriptions

Field	Description
7 CCIF	**Command Complete Interrupt Flag** — The CCIF flag indicates that a Flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command and CCIF will stay low until command completion or command violation. 0 Flash command in progress 1 Flash command has completed
5 ACCERR	**Flash Access Error Flag** — The ACCERR bit indicates an illegal access has occurred to the Flash memory caused by either a violation of the command write sequence (see Section 30.4.4.2) or issuing an illegal Flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR bit has no effect on ACCERR. 0 No access error detected 1 Access error detected
4 FPVIOL	**Flash Protection Violation Flag** —The FPVIOL bit indicates an attempt was made to program or erase an address in a protected area of P-Flash or EEPROM memory during a command write sequence. The FPVIOL bit is cleared by writing a 1 to FPVIOL. Writing a 0 to the FPVIOL bit has no effect on FPVIOL. While FPVIOL is set, it is not possible to launch a command or start a command write sequence. 0 No protection violation detected 1 Protection violation detected