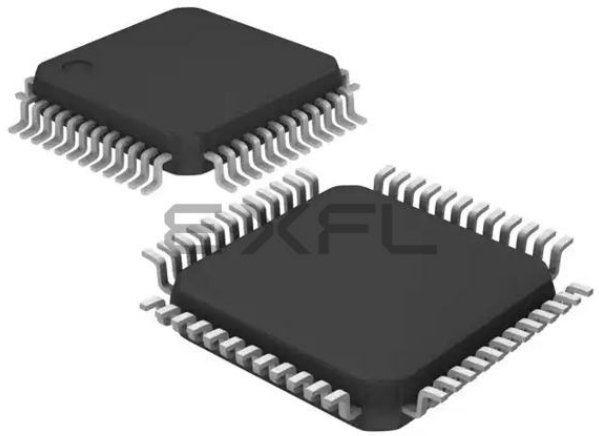


Welcome to E-XFL.COM

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	IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	28
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12vr48af0clf

Chapter 5
Background Debug Module (S12SBDMV1)

Chapter 6
S12S Debug Module (S12DBGV2)

1.6 Family Memory Map

Table 1-2 shows the MC9S12VR-Family register memory map.

Table 1-2. Device Register Memory Map

Address	Module	Size (Bytes)
0x0000–0x0009	PIM (port integration module)	10
0x000A–0x000B	MMC (memory map control)	2
0x000C–0x000D	PIM (port integration module)	2
0x000E–0x000F	Reserved	2
0x0010–0x0017	MMC (memory map control)	8
0x0018–0x0019	Reserved	2
0x001A–0x001B	Device ID register	2
0x001C–0x001F	PIM (port integration module)	4
0x0020–0x002F	DBG (debug module)	16
0x0030–0x0033	Reserved	4
0x0034–0x003F	CPMU (clock and power management)	12
0x0040–0x006F	TIM (timer module <= 4channels)	48
0x0070–0x009F	ADC (analog to digital converter <= 6 channels)	48
0x00A0–0x00C7	PWM (pulse-width modulator <= 2channels)	40
0x00C8–0x00CF	SCI0 (serial communication interface)	8
0x00D0–0x00D7	SCI1 (serial communication interface)	8
0x00D8–0x00DF	SPI (serial peripheral interface)	8
0x00E0–0x00FF	Reserved	32
0x0100–0x0113	FTMRG control registers	20
0x0114–0x011F	Reserved	12
0x0120	INT (interrupt module)	1
0x0121–0x013F	Reserved	31
0x0140-0x0147	HSDRV (high-side driver)	8
0x0148-0x014F	Reserved	8
0x0150-0x0157	LSDRV (low-side driver)	8
0x0158-0x015F	Reserved	8
0x0160-0x0167	LINPHY (LIN physical layer)	8
0x0168-0x016F	Reserved	8
0x0170-0x0177	BATS (Supply Voltage Sense)	8
0x0178–0x023F	Reserved	200
0x0240–0x027F	PIM (port integration module)	64

2.3.19 Port S Data Register (PTS)

Address 0x0248 (S12VR64/48) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
W								
Altern. Function	—	—	—	—	ECLK	—	—	—
	—	—	$\overline{SS}$	SCK	MOSI	MISO	—	—
	—	—	—	—	(TXD1)	(RXD1)	TXD1	RXD1
	—	—	—	—	(PWM5[2])	(PWM4[2])	(LPDR1)	—
	—	—	—	—	(ETRIG1)	(ETRIG0)	(TXD0)	(RXD0)
Reset	0	0	0	0	0	0	0	0

Figure 2-18. Port S Data Register (PTS - S12VR64/48)

1 Read: Anytime. The data source is depending on the data direction value.
Write: Anytime

2 PWM function available on this pin only if not used with a routed HSDRV or LSDRV function. Refer to Section 2.3.16, “Module Routing Register 0 (MODRR0)”

Address 0x0248 (S12VR32/16) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTS3	PTS2	0	0
W								
Altern. Function	—	—	—	—	ECLK	—	—	—
	—	—	—	—	MOSI	MISO	—	—
	—	—	—	—	(TXD1)	(RXD1)	—	—
	—	—	—	—	(PWM5[2])	(PWM4[2])	—	—
	—	—	—	—	(ETRIG1)	(ETRIG0)	—	—
Reset	0	0	0	0	0	0	0	0

Figure 2-19. Port S Data Register (PTS - S12VR32/16)

1 Read: Anytime. The data source is depending on the data direction value.
Write: Anytime

2 PWM function available on this pin only if not used with a routed HSDRV or LSDRV function. Refer to Section 2.3.16, “Module Routing Register 0 (MODRR0)”

2.3.20 Port S Input Register (PTIS)

Address 0x0249 (S12VR64/48) Access: User read only[1]

	7	6	5	4	3	2	1	0
R	0	0	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-20. Port S Input Register (PTIS - S12VR64/48)

[1] Read: Anytime
Write:Never

Address 0x0249 (S12VR32/16) Access: User read only[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTIS3	PTIS2	0	0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-21. Port S Input Register (PTIS - S12VR32/16)

[1] Read: Anytime
Write:Never

Table 2-18. PTIS Register Field Descriptions

Field	Description
5-0 PTIS	**PorT Input data register port S —** A read always returns the synchronized input state of the associated pin. It can be used to detect overload or short circuit conditions on output pins.

4.6.1.5 HTI - High Temperature Interrupt

In FPM the junction temperature T_J is monitored. Whenever T_J exceeds level T_{HTIA} the status bit HTDS is set to 1. Vice versa, HTDS is reset to 0 when T_J get below level T_{HTID}. An interrupt, indicated by flag HTIF = 1, is triggered by any change of the status bit HTDS, if interrupt enable bit HTIE = 1.

4.6.1.6 Autonomous Periodical Interrupt (API)

The API sub-block can generate periodical interrupts independent of the clock source of the MCU. To enable the timer, the bit APIFE needs to be set.

The API timer is either clocked by the Autonomous Clock (ACLK - trimmable internal RC oscillator) or the Bus Clock. Timer operation will freeze when MCU clock source is selected and Bus Clock is turned off. The clock source can be selected with bit APICLK. APICLK can only be written when APIFE is not set.

The APIR[15:0] bits determine the interrupt period. APIR[15:0] can only be written when APIFE is cleared. As soon as APIFE is set, the timer starts running for the period selected by APIR[15:0] bits. When the configured time has elapsed, the flag APIF is set. An interrupt, indicated by flag APIF = 1, is triggered if interrupt enable bit APIE = 1. The timer is re-started automatically again after it has set APIF.

The procedure to change APICLK or APIR[15:0] is first to clear APIFE, then write to APICLK or APIR[15:0], and afterwards set APIFE.

The API Trimming bits ACLKTR[5:0] must be set so the minimum period equals 0.2 ms if stable frequency is desired.

See Table 4-20 for the trimming effect of ACLKTR.

NOTE

The first period after enabling the counter by APIFE might be reduced by API start up delay t_{sdel}.

It is possible to generate with the API a waveform at the external pin API_EXTCLK by setting APIFE and enabling the external access with setting APIEA.

4.7 Initialization/Application Information

4.7.1 General Initialization information

Usually applications run in MCU Normal Mode.

It is recommended to write the CPMUCOP register in any case from the application program initialization routine after reset no matter if the COP is used in the application or not, even if a configuration is loaded via the flash memory after reset. By doing a “controlled” write access in MCU Normal Mode (with the right value for the application) the write once for the COP configuration bits (WCOP,CR[2:0]) takes place which protects these bits from further accidental change. In case of a program sequencing issue (code runaway) the COP configuration can not be accidentally modified anymore.

Table 5-6. Firmware Commands

Command[1]	Opcode (hex)	Data	Description
READ_NEXT[2]	62	16-bit data out	Increment X index register by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT[2]	42	16-bit data in	Increment X index register by 2 (X = X + 2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL[3]	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

1 If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

2 When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.

3 System stop disables the ACK function and ignored commands will not have an ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge if CPU executes the wait or stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 5.4.7, “Serial Interface Hardware Handshake Protocol” last note).

5.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word, depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, only one byte of which contains valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

- 4-stage state sequencer for trace buffer control
  - — Tracing session trigger linked to Final State of state sequencer
  - — Begin and End alignment of tracing to trigger

6.1.4 Modes of Operation

The DBG module can be used in all MCU functional modes.

During BDM hardware accesses and whilst the BDM module is active, CPU monitoring is disabled. When the CPU enters active BDM Mode through a BACKGROUND command, the DBG module, if already armed, remains armed.

The DBG module tracing is disabled if the MCU is secure, however, breakpoints can still be generated.

Table 6-2. Mode Dependent Restriction Summary

BDM Enable	BDM Active	MCU Secure	Comparator Matches Enabled	Breakpoints Possible	Tagging Possible	Tracing Possible
x	x	1	Yes	Yes	Yes	No
0	0	0	Yes	Only SWI	Yes	Yes
0	1	0	Active BDM not possible when not enabled			
1	0	0	Yes	Yes	Yes	Yes
1	1	0	No	No	No	No

6.1.5 Block Diagram

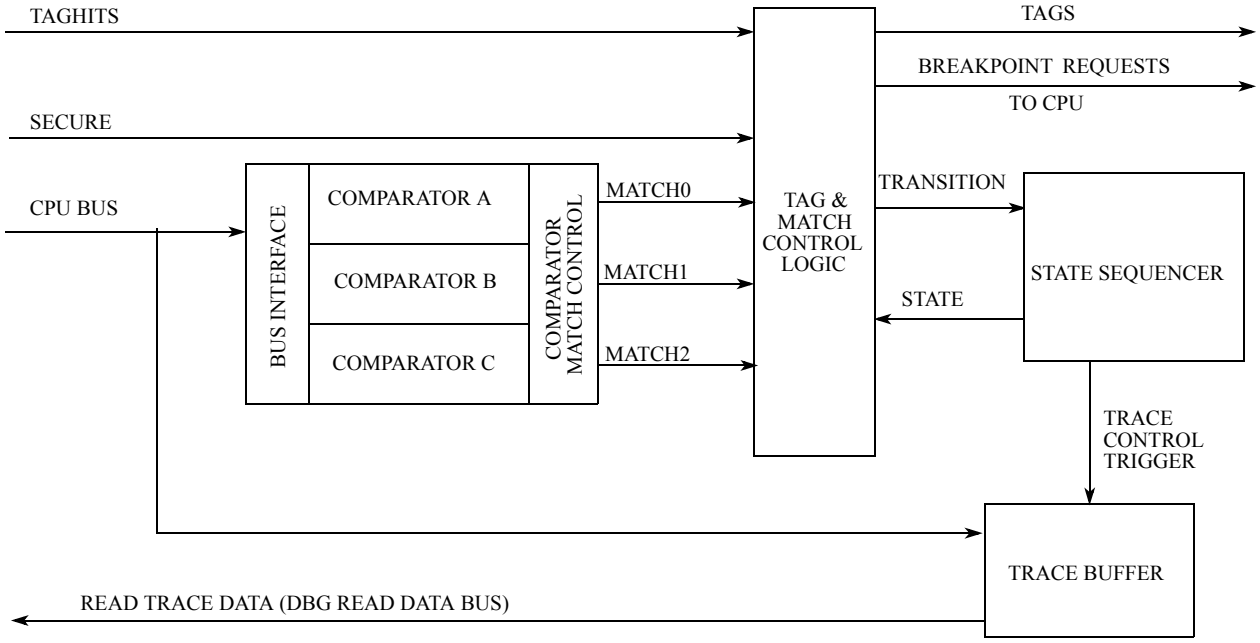


Figure 6-1. Debug Module Block Diagram

Table 6-3. DBGC1 Field Descriptions

Field	Description
7 ARM	**Arm Bit** — The ARM bit controls whether the DBG module is armed. This bit can be set and cleared by user software and is automatically cleared on completion of a debug session, or if a breakpoint is generated with tracing not enabled. On setting this bit the state sequencer enters State1. 0 Debugger disarmed 1 Debugger armed
6 TRIG	**Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate trigger independent of state sequencer status. When tracing is complete a forced breakpoint may be generated depending upon DBGBRK and BDM bit settings. This bit always reads back a 0. Writing a 0 to this bit has no effect. If the DBGTCR_TSOURCE bit is clear no tracing is carried out. If tracing has already commenced using BEGIN trigger alignment, it continues until the end of the tracing session as defined by the TALIGN bit, thus TRIG has no affect. In secure mode tracing is disabled and writing to this bit cannot initiate a tracing session. The session is ended by setting TRIG and ARM simultaneously. 0 Do not trigger until the state sequencer enters the Final State. 1 Trigger immediately
4 BDM	**Background Debug Mode Enable** — This bit determines if a breakpoint causes the system to enter Background Debug Mode (BDM) or initiate a Software Interrupt (SWI). If this bit is set but the BDM is not enabled by the ENBDM bit in the BDM module, then breakpoints default to SWI. 0 Breakpoint to Software Interrupt if BDM inactive. Otherwise no breakpoint. 1 Breakpoint to BDM, if BDM enabled. Otherwise breakpoint to SWI
3 DBGBRK	**S12DBGV2 Breakpoint Enable Bit** — The DBGBRK bit controls whether the debugger will request a breakpoint on reaching the state sequencer Final State. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. 0 No Breakpoint generated 1 Breakpoint generated
1–0 COMRV	**Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the S12SDBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which register is visible at the address 0x0027. See Table 6-4.

Table 6-4. COMRV Encoding

COMRV	Visible Comparator	Visible Register at 0x0027
00	Comparator A	DBGSCR1
01	Comparator B	DBGSCR2
10	Comparator C	DBGSCR3
11	None	DBGMFR

6.3.2.2 Debug Status Register (DBGSR)

```
        RTI                             ;
```

The execution flow taking into account the IRQ is as follows

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ;
IRQ_ISR LDAB    #$F0                    ;
        STAB    VAR_C1
        RTI                             ;
SUB_1   BRN     *
        NOP                             ;
ADDR1   DBNE    A,PART5                 ;
```

6.4.5.2.2 Loop1 Mode

Loop1 Mode, similarly to Normal Mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 Mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the DBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 Mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user's code that the DBG module is designed to help find.

6.4.5.2.3 Detail Mode

In Detail Mode, address and data for all memory and register accesses is stored in the trace buffer. This mode is intended to supply additional information on indexed, indirect addressing modes where storing only the destination address would not provide all information required for a user to determine where the code is in error. This mode also features information bit storage to the trace buffer, for each address byte storage. The information bits indicate the size of access (word or byte) and the type of access (read or write).

When tracing in Detail Mode, all cycles are traced except those when the CPU is either in a free or opcode fetch cycle.

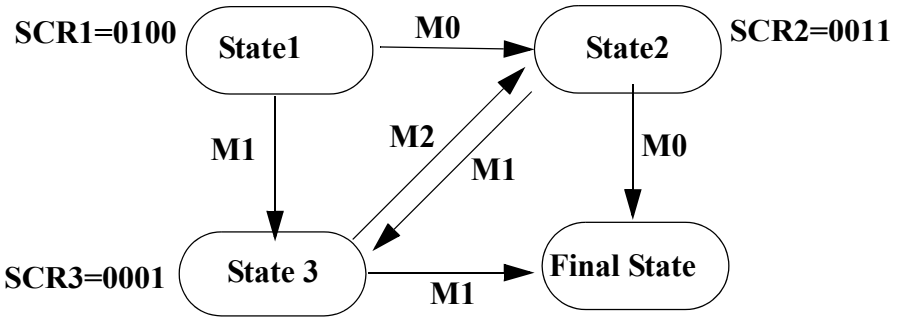
6.4.5.2.4 Compressed Pure PC Mode

In Compressed Pure PC Mode, the PC addresses of all executed opcodes, including illegal opcodes are stored. A compressed storage format is used to increase the effective depth of the trace buffer. This is achieved by storing the lower order bits each time and using 2 information bits to indicate if a 64 byte boundary has been crossed, in which case the full PC is stored.

Each Trace Buffer row consists of 2 information bits and 18 PC address bits

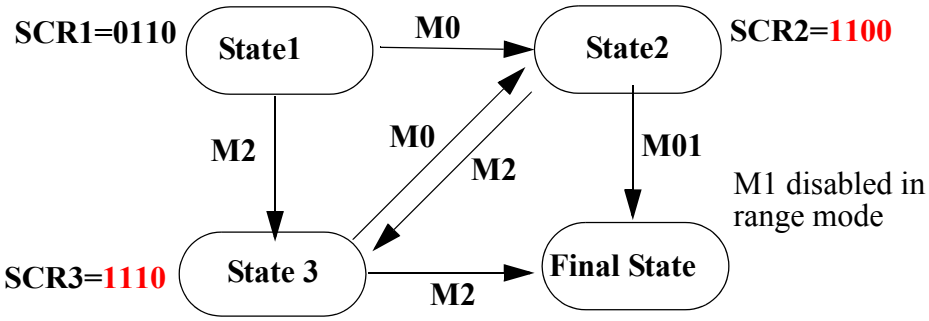
event B cause a trigger. Similarly 2 consecutive occurrences of event B without an intermediate event A cause a trigger. This is possible by using CompA and CompC to match on the same address as shown.

Figure 6-32. Scenario 4a



This scenario is currently not possible using 2 comparators only. S12SDBGV2 makes it possible with 2 comparators, State 3 allowing a M0 to return to state 2, whilst a M2 leads to final state as shown.

Figure 6-33. Scenario 4b (with 2 comparators)



The advantage of using only 2 channels is that now range comparisons can be included (channel0)

This however violates the S12SDBGV1 specification, which states that a match leading to final state always has priority in case of a simultaneous match, whilst priority is also given to the lowest channel number. For S12SDBG the corresponding CPU priority decoder is removed to support this, such that on simultaneous taghits, taghits pointing to final state have highest priority. If no taghit points to final state then the lowest channel number has priority. Thus with the above encoding from State3, the CPU and DBG would break on a simultaneous M0/M2.

is generated. Configuring CompA and CompC the same, it is possible to generate a breakpoint on the third consecutive occurrence of event M0 without a reset M1.

Figure 6-40. Scenario 10a

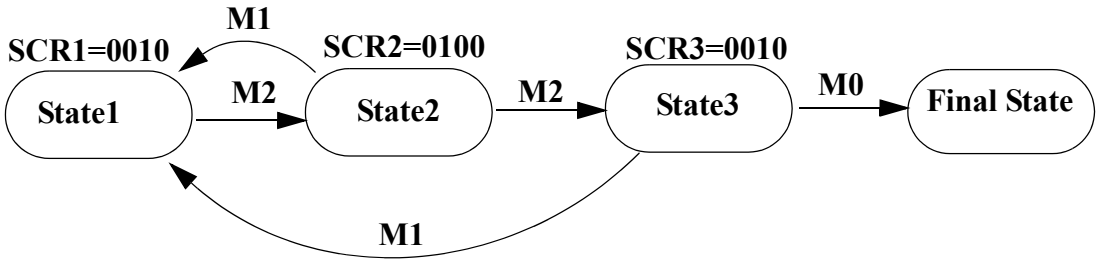
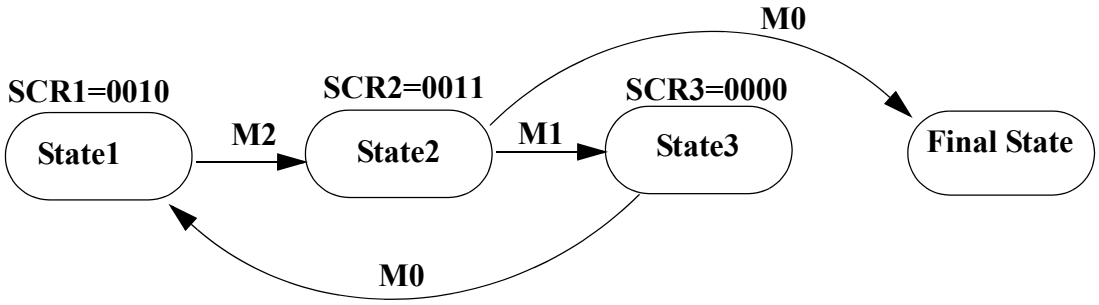


Figure 6-41. Scenario 10b



Scenario 10b shows the case that after M2 then M1 must occur before M0. Starting from a particular point in code, event M2 must always be followed by M1 before M0. If after any M2, event M0 occurs before M1 then a trigger is generated.

10.3.2.1 SCI Baud Rate Registers (SCIBDH, SCIBDL)

	7	6	5	4	3	2	1	0
R W	SBR15	SBR14	SBR13	SBR12	SBR11	SBR10	SBR9	SBR8
Reset	0	0	0	0	0	0	0	0

Figure 10-3. SCI Baud Rate Register (SCIBDH)

	7	6	5	4	3	2	1	0
R W	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
Reset	0	0	1	0	0	0	0	0

Figure 10-4. SCI Baud Rate Register (SCIBDL)

Read: Anytime, if AMAP = 0.

Write: Anytime, if AMAP = 0.

NOTE

Those two registers are only visible in the memory map if AMAP = 0 (reset condition).

The SCI baud rate register is used by to determine the baud rate of the SCI, and to control the infrared modulation/demodulation submodule.

Table 10-2. SCIBDH and SCIBDL Field Descriptions

Field	Description
SBR[15:0]	**SCI Baud Rate Bits** — The baud rate for the SCI is determined by the bits in this register. The baud rate is calculated two different ways depending on the state of the IREN bit. The formulas for calculating the baud rate are: When IREN = 0 then, SCI baud rate = SCI bus clock / (SBR[15:0]) When IREN = 1 then, SCI baud rate = SCI bus clock / (2 x SBR[15:1]) **Note:** The baud rate generator is disabled after reset and not started until the TE bit or the RE bit is set for the first time. The baud rate generator is disabled when (SBR[15:4] = 0 and IREN = 0) or (SBR[15:5] = 0 and IREN = 1). **Note:** . User should write SCIBD by word access. The updated SCIBD may take effect until next RT clock start, write SCIBDH or SCIBDL separately may cause baud generator load wrong data at that time,if second write later then RT clock.

10.3.2.2 SCI Control Register 1 (SCICR1)

	7	6	5	4	3	2	1	0
R W	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

Figure 10-5. SCI Control Register 1 (SCICR1)

10.4.1 Infrared Interface Submodule

This module provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the SCI. The IrDA physical layer specification defines a half-duplex infrared communication link for exchange data. The full standard includes data rates up to 16 Mbits/s. This design covers only data rates between 2.4 Kbits/s and 115.2 Kbits/s.

The infrared submodule consists of two major blocks: the transmit encoder and the receive decoder. The SCI transmits serial bits of data which are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses should be detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder (external from the MCU). The narrow pulses are then stretched by the infrared submodule to get back to a serial bit stream to be received by the SCI.The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses.

The infrared submodule receives its clock sources from the SCI. One of these two clocks are selected in the infrared submodule in order to generate either 3/16, 1/16, 1/32 or 1/4 narrow pulses during transmission. The infrared block receives two clock sources from the SCI, R16XCLK and R32XCLK, which are configured to generate the narrow pulse width during transmission. The R16XCLK and R32XCLK are internal clocks with frequencies 16 and 32 times the baud rate respectively. Both R16XCLK and R32XCLK clocks are used for transmitting data. The receive decoder uses only the R16XCLK clock.

10.4.1.1 Infrared Transmit Encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD pin. A narrow pulse is transmitted for a zero bit and no pulse for a one bit. The narrow pulse is sent in the middle of the bit with a duration of 1/32, 1/16, 3/16 or 1/4 of a bit time. A narrow high pulse is transmitted for a zero bit when TXPOL is cleared, while a narrow low pulse is transmitted for a zero bit when TXPOL is set.

10.4.1.2 Infrared Receive Decoder

The infrared receive block converts data from the RXD pin to the receive shift register. A narrow pulse is expected for each zero received and no pulse is expected for each one received. A narrow high pulse is expected for a zero bit when RXPOL is cleared, while a narrow low pulse is expected for a zero bit when RXPOL is set. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

10.4.2 LIN Support

This module provides some basic support for the LIN protocol. At first this is a break detect circuitry making it easier for the LIN software to distinguish a break character from an incoming data stream. As a further addition is supports a collision detection at the bit level as well as cancelling pending transmissions.

Chapter 13
High-Side Drivers - HSDRV (S12HSDRVV2) for S12VR64

Table 13-1. Revision History Table

Rev. No. (Item No.)	Date (Submitted By)	Sections Affected	Substantial Change(s)
V1.00	10 December 2010	All	- Initial
V2.00	07 Sep 2012	All	- Added description and register bits for over-current masking feature
V2.02	05 August 2013	All	- Removed open-load detection feature

13.1 Introduction

The HSDRV module provides two high-side drivers typically used to drive LED or resistive loads

13.1.1 Features

The HSDRV module includes two independent high-side drivers with common high voltage supply. Each driver has the following features:

- Selectable gate control of high-side switches: HSDR[1:0] register bits or PWM or timer channels.
- Over-current shutdown for the drivers, while they are enabled, comprising:
  - Interrupt flag generation
  - Driver shutdown
  - Optional masking window

13.1.2 Modes of Operation

The HSDRV module behaves as follows in the system power modes:

1. MCU run mode

   The activation of the HSE0 or HSE1 bits enable the related high-side driver. The driver is controlled by the selected source.
2. MCU stop mode

13.3.6 HSDRV Interrupt Enable Register (HSIE)

Module Base + 0x0006 Access: User read/write[1]

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

= Unimplemented

Figure 13-5. HSDRV Interrupt Enable Register (HSIE)

[1] Read: Anytime
Write: Anytime

Table 13-7. HSIE Register Field Descriptions

Field	Description
7 HSOCIE	**HSDRV Over-Current Interrupt Enable** 0 Interrupt request is disabled 1Interrupt is requested whenever a HSOCIFx flag is set

14.3.2 Register Definition

14.3.3 Port HS Data Register (HSDR)

Module Base + 0x0000 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	HSDR0
W								
Altern. Read Function	—	—	—	—	—	—	—	OC[2]
	—	—	—	—	—	—	—	PWM[2]
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-2. Port HS Data Register (HSDR)

1 Read: Anytime The data source (HSDRx or alternate function) depends on the HSE control bit settings.
Write: Anytime

2 See PIM chapter for detailed routing description.

Table 14-4. Port HS Data Register (HSDR) Field Descriptions

Field	Description
0 HSDR0	**Port HS Data** — Data register output or routed timer output or routed PWM output This register can be used to control the high-side driver if selected as control source. See PIM section for routing details. If the HSCR[HSE0] bit is set to 0, a read returns the value of the Port HS Data Register (HSDR[HSDRx]). If the HSCR[HSE0] bit is set to 1, a read returns the value of the selected control source for the driver. When entering in STOP mode the Port HS Data Register (HSDR) is cleared. 0 High-side driver is turned off 1 High-side driver is turned on **Note:** After enabling the high-side driver with the HSCR[HSE0] bit, software must wait for a minimum settling time $t_{HS\_settling}$ before turning on the high-side driver.

14.3.4 HSDRV1C Configuration Register (HSCR)

Module Base + 0x0001 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	HSOCME0	0	HSOLE0	0	HSE0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-3. HSDRV1C Configuration Register (HSCR)

17.4.2 Interrupts

This section describes the interrupt generated by the BATS module. The interrupt is only available in CPU run mode. Entering and exiting CPU stop mode has no effect on the interrupt flags.

To make sure the interrupt generation works properly the bus clock frequency must be higher than the Voltage Warning Low Pass Filter frequency (f_{VWLP_filter}).

The comparator outputs BVLC and BVHC are forced to zero if the comparator is disabled (configuration bits BSESE and BSUSE are cleared). If the software disables the comparator during a high or low Voltage condition (BVHC or BVLC active), then an additional interrupt is generated. To avoid this behavior the software must disable the interrupt generation before disabling the comparator.

The BATS interrupt vector is named in Table 17-6. Vector addresses and interrupt priorities are defined at MCU level.

The module internal interrupt sources are combined into one module interrupt signal.

Table 17-6. BATS Interrupt Sources

Module Interrupt Source	Module Internal Interrupt Source	Local Enable
BATS Interrupt (BATI)	BATS Voltage Low Condition Interrupt (BVLI)	BVLIE = 1
	BATS Voltage High Condition Interrupt (BVHI)	BVHIE = 1

17.4.2.1 BATS Voltage Low Condition Interrupt (BVLI)

To use the Voltage Low Interrupt the Level Sensing must be enabled (BSESE =1 or BSUSE =1).

If measured when

a) V_{LBI1} selected with BVLS[1:0] = 0x0
at selected pin $V_{measure} < V_{LBI1\_A}$ (falling edge) or $V_{measure} < V_{LBI1\_D}$ (rising edge)

or when

b) V_{LBI2} selected with BVLS[1:0] = 0x1
at selected pin $V_{measure} < V_{LBI2\_A}$ (falling edge) or $V_{measure} < V_{LBI2\_D}$ (rising edge)

or when

c) V_{LBI3} selected with BVLS[1:0] = 0x2
at selected pin $V_{measure} < V_{LBI3\_A}$ (falling edge) or $V_{measure} < V_{LBI3\_D}$ (rising edge)

or when

d) V_{LBI4} selected with BVLS[1:0] = 0x3
at selected pin $V_{measure} < V_{LBI4\_A}$ (falling edge) or $V_{measure} < V_{LBI4\_D}$ (rising edge)

then BVLC is set. BVLC status bit indicates that a low voltage at the selected pin is present. The Low Voltage Interrupt flag (BVLIF) is set to 1 when the Voltage Low Condition (BVLC) changes state . The

that will be loaded during the reset sequence, the P-Flash sector containing the EEPROM protection byte must be unprotected, then the EEPROM protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the EEPROM protection byte during the reset sequence, the DPOPEN bit will be cleared and DPS bits will be set to leave the EEPROM memory fully protected.

Trying to alter data in any protected area in the EEPROM memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. Block erase of the EEPROM memory is not possible if any of the EEPROM sectors are protected.

Table 18-22. EEPROT Field Descriptions

Field	Description
7 DPOPEN	**EEPROM Protection Control** 0 Enables EEPROM memory protection from program and erase with protected address range defined by DPS bits 1 Disables EEPROM memory protection from program and erase
3–0 DPS[3:0]	**EEPROM Protection Size** — The DPS[3:0] bits determine the size of the protected area in the EEPROM memory as shown inTable 18-23 .

Table 18-23. EEPROM Protection Address Range

DPS[3:0]	Global Address Range	Protected Size
0000	0x0_0400 – 0x0_041F	32 bytes
0001	0x0_0400 – 0x0_043F	64 bytes
0010	0x0_0400 – 0x0_045F	96 bytes
0011	0x0_0400 – 0x0_047F	128 bytes
0100	0x0_0400 – 0x0_049F	160 bytes
0101	0x0_0400 – 0x0_04BF	192 bytes
The Protection Size goes on enlarging in step of 32 bytes, for each DPS value increasing of one. . . .		
1111	0x0_0400 – 0x0_05FF	512 bytes

18.3.2.11 Flash Common Command Object Register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

Table 19-5. Program IFR Fields

Global Address	Size (Bytes)	Field Description
0x0_4000 – 0x0_4007	8	Reserved
0x0_4008 – 0x0_40B5	174	Reserved
0x0_40B6 – 0x0_40B7	2	Version ID[1]
0x0_40B8 – 0x0_40BF	8	Reserved
0x0_40C0 – 0x0_40FF	64	Program Once Field Refer to Section 19.4.6.6, “Program Once Command”

1 Used to track firmware patch versions, see Section 19.4.2 IFR Version ID Word

Table 19-6. Memory Controller Resource Fields (NVMRES[1]=1)

Global Address	Size (Bytes)	Description
0x0_4000 – 0x0_40FF	256	P-Flash IFR (see Table 19-5)
0x0_4100 – 0x0_41FF	256	Reserved.
0x0_4200 – 0x0_57FF		Reserved
0x0_5800 – 0x0_59FF	512	Reserved
0x0_5A00 – 0x0_5FFF	1,536	Reserved
0x0_6000 – 0x0_6BFF	3,072	Reserved
0x0_6C00 – 0x0_7FFF	5,120	Reserved

1 NVMRES - See Section 19.4.3 Internal NVM resource (NVMRES) for NVMRES (NVM Resource) detail.

Appendix H
LSDRV Electrical Specifications

This section provides electrical parametric and ratings for the LSDRV.

H.1 Static Characteristics

Table H-1. Static Characteristics - LSDRV

Characteristics noted under conditions 6V ≤ VSUP ≤ 18 V, -40°C ≤ T_J ≤ 150°C[1] unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25°C[2] under nominal conditions unless otherwise noted.						
Num	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	VSUP range for LSDRV compliant electrical characteristics	V_{SUP}	6	12	18	V
2	VSUP range within which the device is working without LSDRV compliant electrical characteristics	V_{SUP}		3.5 to 6 and 18 to 27		V
3	Output Drain-to-Source On Resistance T_J = 25°C, $I_{PLS0/1}$ = 150 mA T_J = 150°C, $I_{PLS0/1}$ = 150 mA	$R_{DS(ON)}$	 – –	 2.3 –	 – 4.5	Ω
4	Output Over-Current Threshold The threashold is valid for each LS-driver output. **Note:** **The low-side driver is NOT intended to switch capacitive loads. A significant capacitive load on PLS0/1 would induce a current when the low-side driver gate is turned on. This current will be sensed by the over-current circuitry and eventually lead to an immediate over-current shut down.**	I_{LIMLSX}	160	270	350	mA
5	Nominal Current for continuous operation. This value is valid for each LS-driver output.	I_{NOMLSX}	–	–	150	mA
5	Settling time after the low-side driver is enabled (write LSEx Bits)	$t_{LS\_settling}$	1	–		µs
7	High-Load Resistance Open-Load Detection Current (if low-side driver is enabled and gate turned off)	$I_{HLROLDC}$	30	40	50	µA
8	Leakage Current -40°C < T_J < 80°C Open Load Detection disabled.	I_{LEAK_L}	–	–	1	µA µA
9	Leakage Current -40°C < T_J< 150°C Open Load Detection disabled.	I_{LEAK_H}	–	–	10	µA µA