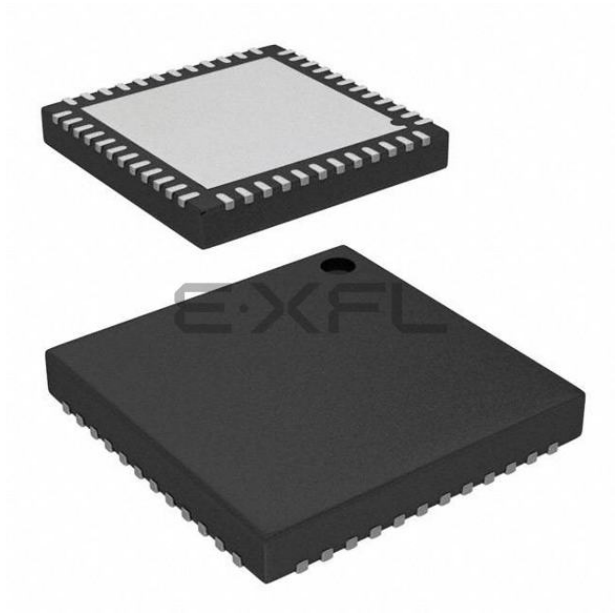


Welcome to E-XFL.COM

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	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	15MHz
Connectivity	LINbus, SPI, UART/USART, LINbus-SBC
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	10
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 1x17b Sigma Delta, 1x18b Sigma Delta
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-VQFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega64hve2-plqw

6. Functional Description

6.1 Pin Functions

For pin functions of the LIN SBC please refer to Section 2.1 “Pin Descriptions” on page 3.

6.2 Physical Layer Compatibility

Since the LIN physical layer is independent from higher LIN layers (e.g., the LIN protocol layer), all nodes with a LIN physical layer according to revision 2.x can be mixed with LIN physical layer nodes, which, according to older versions (i.e., LIN 1.0, LIN 1.1, LIN 1.2, LIN 1.3), are without any restrictions.

6.3 Wake-u Events from Sleep or Silent Mode

- LIN-bus
- EN pin

6.4 Ground Shift

The IC does not affect the LIN-bus in the event of GND disconnection. It is able to handle a ground shift up to 11.5% of VS. This is the mandatory system ground pin.

6.5 TXD Dominant Time-out Function

The TXD input has an internal pull-up resistor. An internal timer prevents the bus line from being driven permanently in dominant state. If TXD is forced to low for longer than t_{DOM} > 27ms, the LIN-bus driver is switched to recessive state. Nevertheless, when switching to Sleep Mode, the actual level at the TXD pin is relevant.

To reactivate the LIN bus driver, switch TXD to high (> 10µs).

10. Electrical Characteristics LIN SBC (Continued)

5V < V_S < 27V, –40°C < T_j < 150°C, unless otherwise specified. All values refer to GND pins

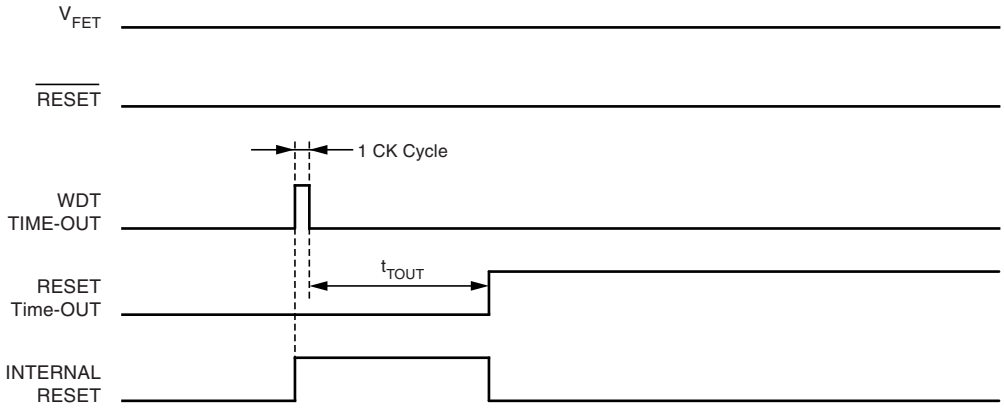
No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
11.5	Switch off leakage current	V_{NRES} = 5.5V	NRES		–3		+3	µA	A
12	Watchdog Oscillator								
12.1	Voltage at WD_OSC in Normal or Fail-safe Mode	I_{WD_OSC} = –200µA V_{VS} ≥ 4V	WD_OSC	V_{WD_OSC}	1.13	1.23	1.33	V	A
12.2	Possible values of resistor	Resistor ±1%	WD_OSC	R_{OSC}	34		120	kΩ	A
12.3	Oscillator period	R_{OSC} = 34kΩ		t_{OSC}	10.65	13.3	15.97	µs	A
12.4	Oscillator period	R_{OSC} = 51kΩ		t_{OSC}	15.68	19.6	23.52	µs	A
12.5	Oscillator period	R_{OSC} = 91kΩ		t_{OSC}	26.83	33.5	40.24	µs	A
12.6	Oscillator period	R_{OSC} = 120kΩ		t_{OSC}	34.2	42.8	51.4	µs	A
13	Watchdog Timing Relative to t_{OSC}								
13.1	Watchdog lead time after Reset			t_d		7895		cycles	A
13.2	Watchdog closed window			t_1		1053		cycles	A
13.3	Watchdog open window			t_2		1105		cycles	A
13.4	Watchdog reset time NRES		NRES	t_{nres}	3.2	4	4.8	ms	A
14	VREG Voltage Regulator in Normal/Fail-safe and Silent Mode, VREG and PVREG Short-circuited								
14.1	Output voltage VREG	4V < V_S < 18V (0mA to 50mA)	VREG	$VREG_{nor}$	3.234		3.366	V	A
14.2	Output voltage VREG at low VS	3V < V_S < 4V	VREG	$VREG_{low}$	V_S – V_D		3.366	V	A
14.3	Regulator drop voltage	V_S > 3V, I_{VREG} = –15mA	VS, VREG	V_D			200	mV	A
14.4	Regulator drop voltage	V_S > 3V, I_{VREG} = –50mA	VS, VREG	V_D		500	700	mV	A
14.5	Line regulation	4V < V_S < 18V	VREG	$VREG_{line}$		0.1	0.2	%	A
14.6	Load regulation	5mA < I_{VREG} < 50mA	VREG	$VREG_{load}$		0.1	0.5	%	A
14.7	Power supply ripple rejection	10Hz to 100kHz C_{VREG} = 10µF V_S = 14V, I_{VREG} = –15mA	VREG		50			dB	D
14.8	Output current limitation	V_S > 4V	VREG	$I_{VREGlim}$	–240	–160	–85	mA	A
14.9	Load capacity	0.2Ω < ESR < 5Ω at 100kHz	VREG	C_{load}	1.8	10		µF	D
14.10	VREG undervoltage threshold	Referred to VREG V_S > 4V	VREG	V_{thunN}	2.8		3.2	V	A
14.11	Hysteresis of undervoltage threshold	Referred to VREG V_S > 4V	VREG	$Vhys_{thun}$		150		mV	A
14.12	Ramp-up time V_S > 4V to V_{REG} = 3.3V	C_{VREG} = 2.2µF I_{load} = –5mA at VREG	VREG	T_{VREG}		320	500	µs	A
15	DIV_ON Input Pin								
15.1	Low-level voltage input		DIV_ON	V_{DIV_ON}	–0.3		+0.8	V	A
15.2	High-level voltage input		DIV_ON	V_{DIV_ON}	2		V_{REG} + 0.3	V	A
15.3	Pull-down resistor	V_{DIV_ON} = V_{REG}	DIV_ON	R_{DIV_ON}	125	250	400	kΩ	A
15.4	Low-level input current	V_{DIV_ON} = 0V	DIV_ON	I_{DIV_ON}	–3		+3	µA	A

*) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

17.2.3 Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to page 62 for details on operation of the Watchdog Timer.

Figure 17-5. Watchdog Reset During Operation



17.2.4 Brown-out Detection

The Atmel® AVR MCU has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a trigger level
$V_{BOT} = V_{BOTIDEAL} \times V_{REF}/1.1$. During start-up the VREF value will change, see Section 27. “Band Gap Reference and Temperature Sensor” on page 161. The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as
$V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} - V_{HYST}/2$.

The BOD is enabled by setting the BODEN fuse in low fuse byte, see Section 30.2 “Fuse Bits” on page 181. When the fuse is programmed the BOD will be enabled in all modes of operation, except in Power-off mode. For applications that do not have an external VCC monitor to generate a reset in case of low VCC, it is recommended to always enable the BOD in order to guarantee safe operating conditions for the device.

When the BOD is enabled, and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 17-6), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 17-6), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

Figure 17-6. Brown-out Reset During Operation

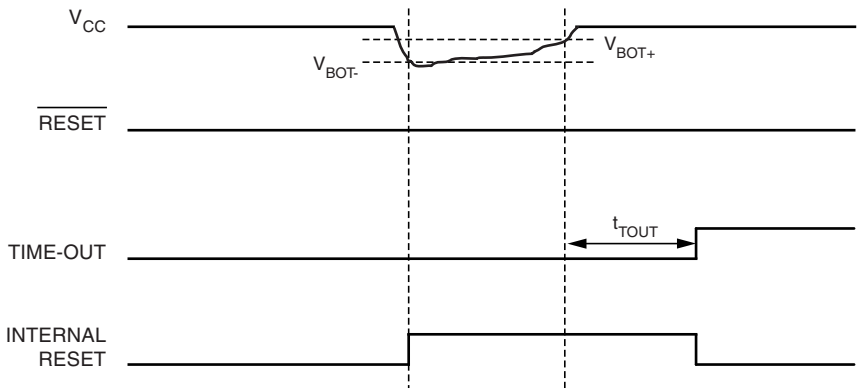


Table 17-1. Watchdog Timer Configuration

WDTON[1]	WDE	WDIE	Mode	Action on Time-out
1	0	0	Stopped	None
1	0	1	Interrupt Mode	Interrupt
1	1	0	System Reset Mode	Reset
1	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
0	x	x	System Reset Mode	Reset

Note: 1. WDTON Fuse set to “0” means programmed, “1” means unprogrammed.

- **Bit 5 – WDP3 : Watchdog Timer Prescaler 3**

  The WDP3..0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 17-2.
- **Bit 4 – WDCE: Watchdog Change Enable**

  This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE must be set.

  Once written to one, hardware will clear WDCE after four clock cycles.
- **Bit 3 – WDE: Watchdog System Reset Enable**

  WDE is overridden by WDRF in MCUSR. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.
- **Bits 2:0 – WDP 2:0: Watchdog Timer Prescaler 2, 1, and 0**

  The WDP3..0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 17-2 on page 66.

Table 17-2. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out[1]
0	0	0	0	2Kcycles	16ms
0	0	0	1	4Kcycles	32ms
0	0	1	0	8Kcycles	64ms
0	0	1	1	16Kcycles	0.13s
0	1	0	0	32Kcycles	0.26s
0	1	0	1	64Kcycles	0.51s
0	1	1	0	128Kcycles	1.0s
0	1	1	1	256Kcycles	2.0s
1	0	0	0	512Kcycles	4.1s
1	0	0	1	1024Kcycles	8.2s
1	0	1	0	Reserved	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Note: 1. The actual timeout value depends on the actual clock period of the Ultra Low Power RC Oscillator, refer to Section 15.2.3 “Ultra Low Power RC Oscillator” on page 50 for details.

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address     Labels      Code        Comments
0x0000      RESET:      ldi         r16,high(RAMEND     ; Main program start
0x0001                  out         SPH,r16             ; Set Stack Pointer to top
                                                        ; of RAM
0x0002                  ldi         r16,low(RAMEND)
0x0003                  out                             SPL,r16
0x0004                  sei                             ; Enable interrupts
0x0005                  <instr>     xxx
;
.org 0x4C02
0x4C02                  jmp         INT0                ; External Interrupt 0
                                                        ; Handler
...                     ...         ...                 ;
0x4C30                  jmp         PLL_LCHNG           ; PLL Lock Change Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2Kbytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address     Labels      Code        Comments
.org 0x0002
0x0002                  jmp         INT0                ; External Interrupt 0
                                                        ; Handler
...                     ...         ...                 ;
0x0030                  jmp         PLL_LCHNG           ; PLL Lock Change Handler
;
.org 0x4C00
0x4C00      RESET:      ldi         r16,high(RAMEND)    ; Main program start
0x4C01                  out         SPH,r16             ; Set Stack Pointer to top
                                                        ; of RAM
0x4C02                  ldi         r16,low(RAMEND)
0x4C03                  out                             SPL,r16
0x4C04                  sei                             ; Enable interrupts
0x4C05                  <instr>     xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

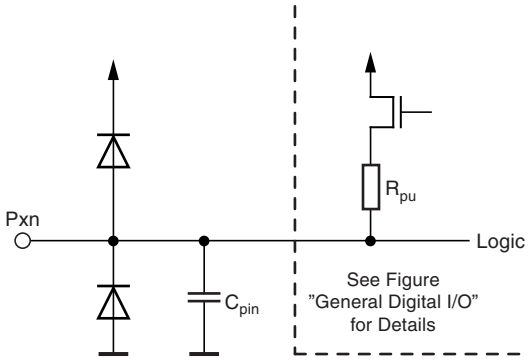
```
Address     Labels      Code        Comments
;
.org 0x4C00
0x4C00                  jmp         RESET               ; Reset Handler
0x4C02                  jmp         INT0                ; External Interrupt 0
                                                        ; Handler
...                     ...         ...                 ;
0x4C30                  jmp         PLL_LCHNG           ; PLL Lock Change Handler
;
0x4C2E      RESET:      ldi         r16,high(RAMEND)    ; Main program start
0x4C2F                  out         SPH,r16             ; Set Stack Pointer to top
                                                        ; of RAM
0x4C30                  ldi         r16,low(RAMEND)
0x4C31                  out         SPL,r16
0x4C32                  sei                             ; Enable interrupts
0x4C33                  <instr>     xxx
```

21. I/O-Ports

21.1 Overview

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 21-1. Refer to Section 31. “Electrical Characteristics AVR MCU” on page 192ff for a complete list of parameters.

Figure 21-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in Section 21.4 “Register Description” on page 88.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

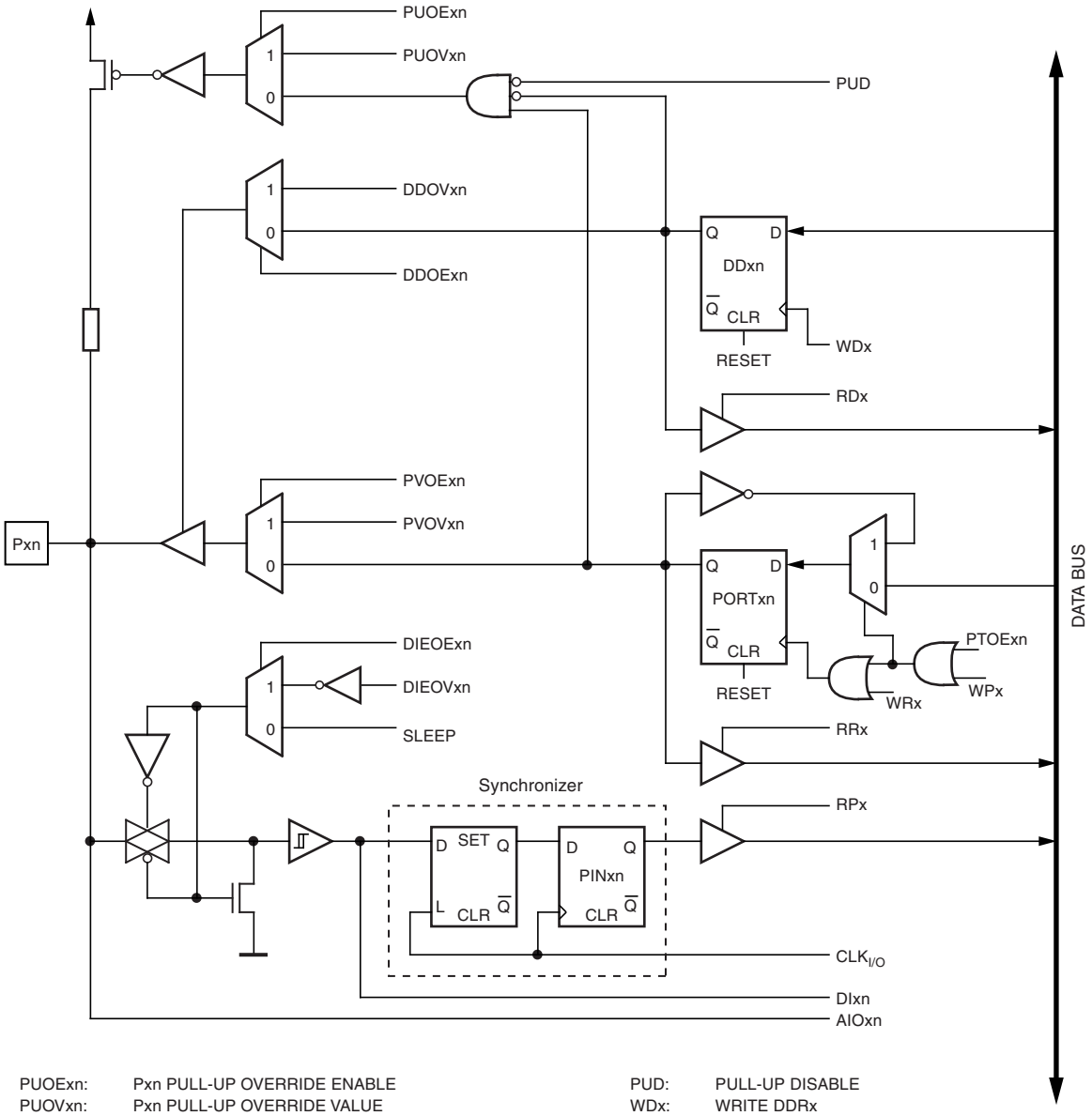
Using the I/O port as General Digital I/O is described in Section 21.2 “Ports as General Digital I/O” on page 79. Many port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in Section 21.3 “Alternate Port Functions” on page 83. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

21.3 Alternate Port Functions

Many port pins have alternate functions in addition to being general digital I/Os. Figure 21-5 shows how the port pin control signals from the simplified Figure 21-2 on page 79 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Figure 21-5. Alternate Port Functions(1)



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULL-UP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WRx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn DIGITAL INPUT ENABLE OVERRIDE ENABLE	WPx:	WRITE PINx
DIEOVxn:	Pxn DIGITAL INPUT ENABLE OVERRIDE VALUE	$CLK_{I/O}$:	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	DIGITAL INPUT PIN n ON PORTx
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

24.5 Register Description

24.5.1 SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIE: SPI Interrupt Enable**

  This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR Register is set and the if the Global Interrupt Enable bit in SREG is set.

- **Bit 6 – SPE: SPI Enable**

  When the SPE bit is written to one, the SPI is enabled. This bit must be set to enable any SPI operations.

- **Bit 5 – DORD: Data Order**

  When the DORD bit is written to one, the LSB of the data word is transmitted first.

  When the DORD bit is written to zero, the MSB of the data word is transmitted first.

- **Bit 4 – MSTR: Master/Slave Select**

  This bit selects Master SPI mode when written to one, and Slave SPI mode when written logic zero. If $\overline{SS}$ is configured as an input and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI Master mode.

- **Bit 3 – CPOL: Clock Polarity**

  When this bit is written to one, SCK is high when idle. When CPOL is written to zero, SCK is low when idle. Refer to Figure 24-3 and Figure 24-4 for an example. The CPOL functionality is summarized below.

Table 24-3. CPOL Functionality

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

- **Bit 2 – CPHA: Clock Phase**

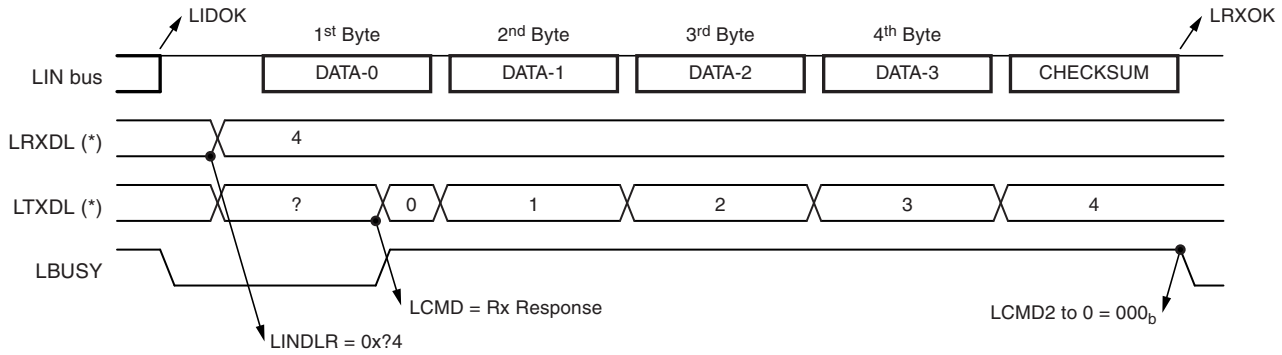
  The settings of the Clock Phase bit (CPHA) determine if data is sampled on the leading (first) or trailing (last) edge of SCK. Refer to Figure 24-3 and Figure 24-4 for an example. The CPHA functionality is summarized below.

Table 24-4. CPHA Functionality

CPHA	Leading Edge	Trailing Edge
0	Sample	Setup
1	Setup	Sample

25.7.6.3Data Length in Rx Response

Figure 25-9. LIN2.1 - Rx Response - No Error

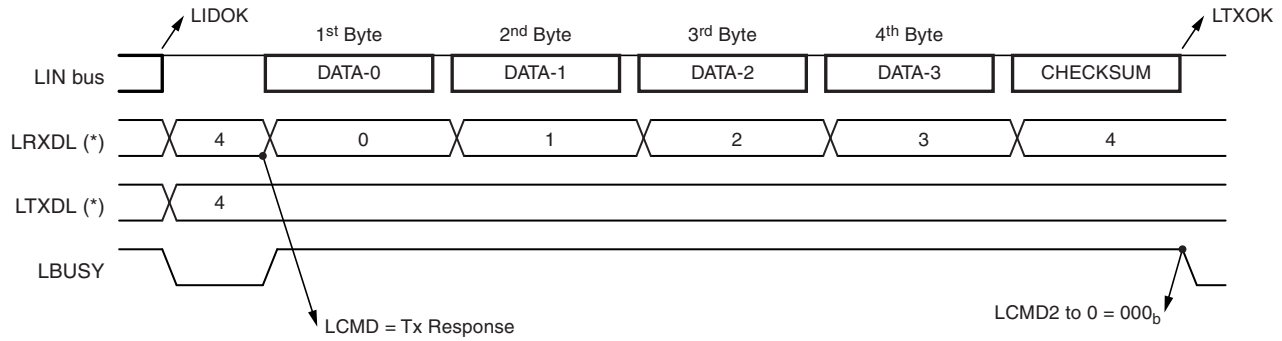


(*): LRXDL and LTXDL updated by user

- The user initializes LRXDL field before setting the Rx Response command,
- After setting the Rx Response command, LTXDL is reset by hardware,
- LRXDL field will remain unchanged during Rx (during busy signal),
- LTXDL field will count the number of received bytes (during busy signal),
- If an error occurs, Rx stops, the corresponding error flag is set and LTXDL will give the number of received bytes without error,
- If no error occurs, LRXOK is set after the reception of the CHECKSUM, LRXDL will be unchanged (and LTXDL = LRXDL).

25.7.6.4Data Length in Tx Response

Figure 25-10.LIN1.3 - Tx Response - No Error



(*): LRXDL and LTXDL updated by Rx Response or Tx Response task

- The user initializes LTXDL field before setting the Tx Response command,
- After setting the Tx Response command, LRXDL is reset by hardware,
- LTXDL will remain unchanged during Tx (during busy signal),
- LRXDL will count the number of transmitted bytes (during busy signal),
- If an error occurs, Tx stops, the corresponding error flag is set and LRXDL will give the number of transmitted bytes without error,
- If no error occurs, LTXOK is set after the transmission of the CHECKSUM, LTXDL will be unchanged (and LRXDL = LTXDL).

25.8 LIN / UART Register Description

25.8.1 LINCR – LIN Control Register

Bit	7	6	5	4	3	2	1	0	
(0xC0)	LSWRES	LIN13	LCONF1	LCONF0	LENA	LCMD2	LCMD1	LCMD0	LINCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – LSWRES: Software Reset**
  - 0 = No action,
  - 1 = Software reset (this bit is self-reset at the end of the reset procedure).
- **Bit 6 – LIN13: LIN 1.3 mode**
  - 0 = LIN 2.1 (default),
  - 1 = LIN 1.3.
- **Bit 5:4 – LCONF[1:0]: Configuration**
  1. LIN mode (default = 00):
     - 00 = LIN Standard configuration (listen mode “off”, CRC “on” and Frame_Time_Out “on”,
     - 01 = No CRC, no Time out (listen mode “off”),
     - 10 = No Frame_Time_Out (listen mode “off” and CRC “on”),
     - 11 = Listening mode (CRC “on” and Frame_Time_Out “on”).
  2. UART mode (default = 00):
     - 00 = 8-bit, no parity (listen mode “off”),
     - 01 = 8-bit, even parity (listen mode “off”),
     - 10 = 8-bit, odd parity (listen mode “off”),
     - 11 = Listening mode, 8-bit, no parity.
- **Bit 3 – LENA: Enable**
  - 0 = Disable (both LIN and UART modes),
  - 1 = Enable (both LIN and UART modes).
- **Bit 2:0 – LCMD[2:0]: Command and mode**

  The command is only available if LENA is set.
  - 000 = LIN Rx Header - LIN abort,
  - 001 = LIN Tx Header,
  - 010 = LIN Rx Response,
  - 011 = LIN Tx Response,
  - 100 = UART Rx and Tx Byte disable,
  - 11x = UART Rx Byte enable,
  - 1x1 = UART Tx Byte enable.

- **Bit 3 – LSERR: Synchronization Error Flag**
  - 0 = No error,
  - 1 = Synchronization error.

  This bit is cleared when LERR bit in LINSIR is cleared.
- **Bit 2 – LPERR: Parity Error Flag**
  - 0 = No error,
  - 1 = Parity error.

  This bit is cleared when LERR bit in LINSIR is cleared.
- **Bit 1 – LCERR: Checksum Error Flag**
  - 0 = No error,
  - 1 = Checksum error.

  This bit is cleared when LERR bit in LINSIR is cleared.
- **Bit 0 – LBERR: Bit Error Flag**
  - 0 = no error,
  - 1 = Bit error.

  This bit is cleared when LERR bit in LINSIR is cleared.

25.8.5 LINBTR – LIN Bit Timing Register

Bit	7	6	5	4	3	2	1	0	
(0xC4)	**LDISR**	–	**LBT5**	**LBT4**	**LBT3**	**LBT2**	**LBT1**	**LBT0**	**LINBTR**
Read/Write	R/W	R	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – LDISR: Disable Bit Timing Resynchronization**
  - 0 = Bit timing re-synchronization enabled (default),
  - 1 = Bit timing re-synchronization disabled.
- **Bits 5:0 – LBT[5:0]: LIN Bit Timing**

  Gives the number of samples of a bit.
  sample-time = (1 / $fclk_{i/o}$) x (LDIV[11..0] + 1)
  Default value: LBT[6:0]=32 — Min. value: LBT[6:0]=8 — Max. value: LBT[6:0]=63

25.8.6 LINBRR – LIN Baud Rate Register

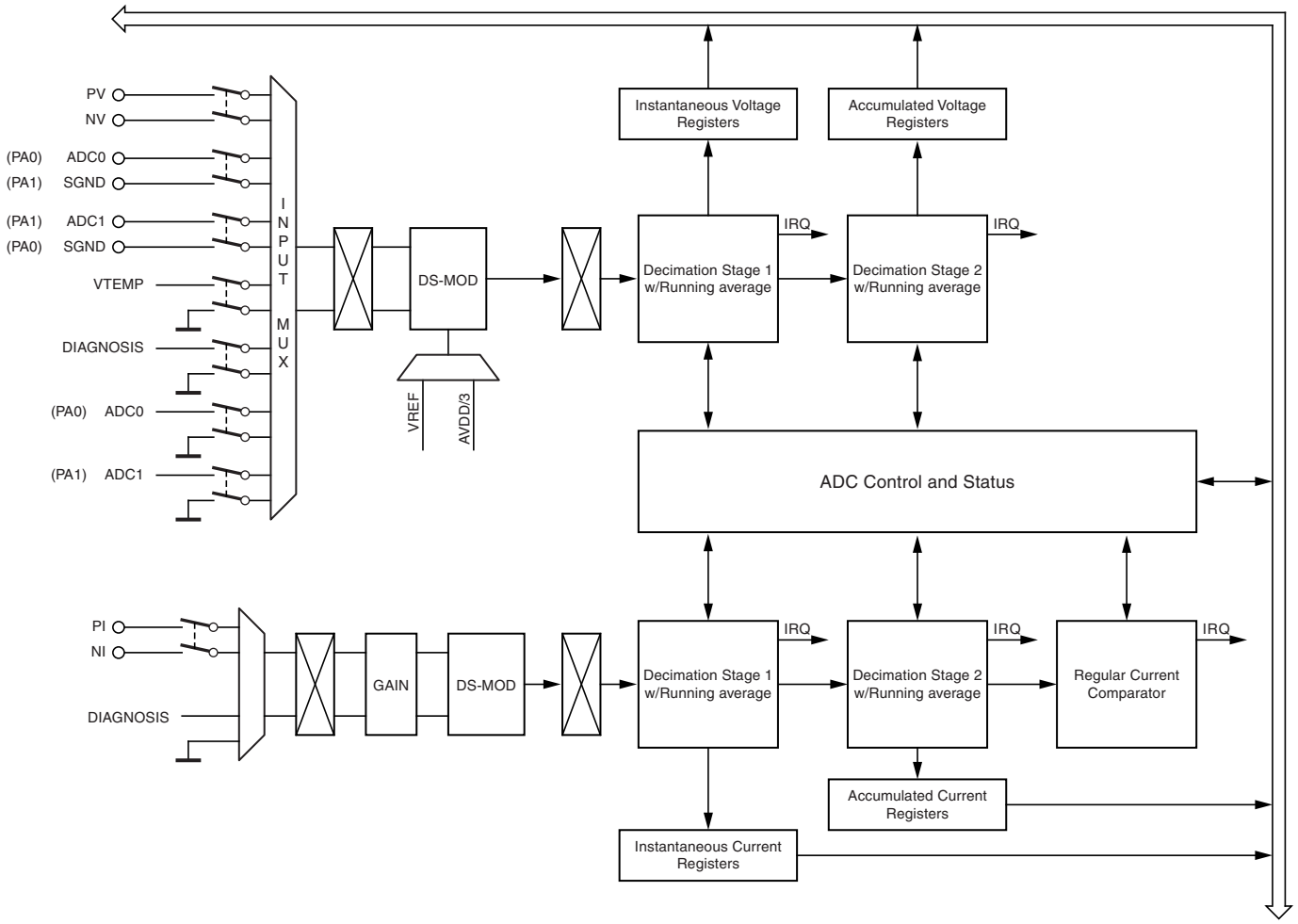
Bit	7	6	5	4	3	2	1	0	
(0xC5)	**LDIV7**	**LDIV6**	**LDIV5**	**LDIV4**	**LDIV3**	**LDIV2**	**LDIV1**	**LDIV0**	**LINBRRL**
(0xC6)	–	–	–	–	**LDIV11**	**LDIV10**	**LDIV9**	**LDIV8**	**LINBRRH**
Bit	15	14	13	12	11	10	9	8	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 15:12 – Reserved**

  These bits are reserved for future use. For compatibility with future devices, they must be written to zero when LINBRR is written.
- **Bits 11:0 – LDIV[11:0]: Scaling of $clk_{i/o}$ Frequency**

  The LDIV value is used to scale the entering $clk_{i/o}$ frequency to achieve appropriate LIN or UART baud rate.

Figure 26-1. ADC Overview



Both the C-ADC and V-ADC include chopper functionality to automatically cancel offset. The chopper can be configured to run automatically or it can be controlled directly by software. When running in automatic settings, the chopper will automatically switch the input polarity on regular intervals and calculate a running average to remove the offset. The automatic chopping can be configured to follow the Instantaneous Current or the Accumulated Conversion Complete.

To allow measurements on a wide range of current inputs, the C-ADC has programmable gain settings.

The C-ADC also includes a Regular Current Comparator. With the Regular Current Comparator the system can be configured to enter a low power mode where it wakes up when current exceeds a configurable trigger level.

The V-ADC is connected to seven different sources through the Input Multiplexer. The PV2/NV2 pins are dedicated pins for measuring the scaled battery terminal voltage. ADC0/1 are two general purpose inputs which can be configured for different external configurations. In addition to the external channels there are two internal channels for internal temperature sensor measurement and diagnosis function.

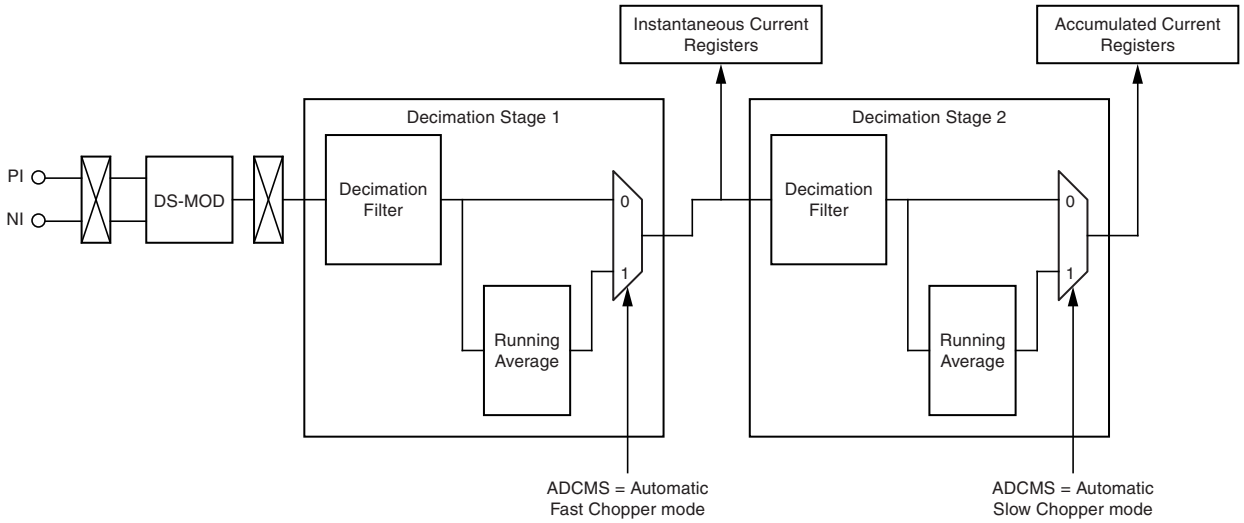
26.3.4 Programmable Chopper Control

Both the C-ADC and V-ADC have a chopper feature to cancel offset in the conversion data. The chopper can be configured by writing to the ADCMS1:0 IO bits in the Section 26.6.3 “ADCRA - ADC Control Register A” on page 151. If enabled, the chopper can be configured to run with in either

- Automatic Fast Chopper mode
- Automatic Slow Chopper mode
- Software Polarity Control mode

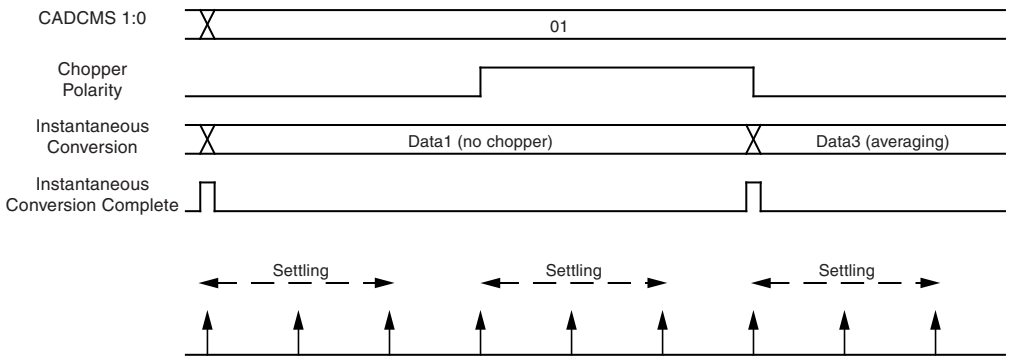
Figure 26-5 gives an overview of the chopper functionality.

Figure 26-5. Chopper Overview



If selecting the Automatic Fast Chopper mode the chopper will switch the polarity of the input on each Instantaneous Conversion and calculate running averaging on the last two conversion results. In Automatic Fast Chopper mode the ADC will automatically perform two settling conversions before using the 3rd Instantaneous Conversion for Running Average, hence the Instantaneous Conversion data rate is reduced by 3x compared to having the Fast Chopper off. To get accurate Accumulation Conversion result the settling conversions in the Instantaneous filter are not used by the second decimation filter stage; hence the data rate in the Accumulated Conversion is also reduced by 3. The Fast Chopper timing is illustrated in Figure 26-6.

Figure 26-6. Fast Chopper Timing



Note that the user software can never read any code that is located inside the RWW section during a Boot Loader software operation. The syntax “Read-While-Write section” refers to which section that is being programmed (erased or written), not which section that actually is being read during a Boot Loader software update.

29.4.1 RWW – Read-While-Write Section

If a Boot Loader software update is programming a page inside the RWW section, it is possible to read code from the Flash, but only code that is located in the NRWW section. During an on-going programming, the software must ensure that the RWW section never is being read. If the user software is trying to read code that is located inside the RWW section (i.e., by a call/jmp/lpm or an interrupt) during programming, the software might end up in an unknown state. To avoid this, the interrupts should either be disabled or moved to the Boot Loader section. The Boot Loader section is always located in the NRWW section. The RWW Section Busy bit (RWWSB) in the Store Program Memory Control and Status Register (SPMCSR) will be read as logical one as long as the RWW section is blocked for reading. After a programming is completed, the RWWSB must be cleared by software before reading code located in the RWW section. See Section 29.9.1 “SPMCSR – Store Program Memory Control and Status Register” on page 178 for details on how to clear RWWSB.

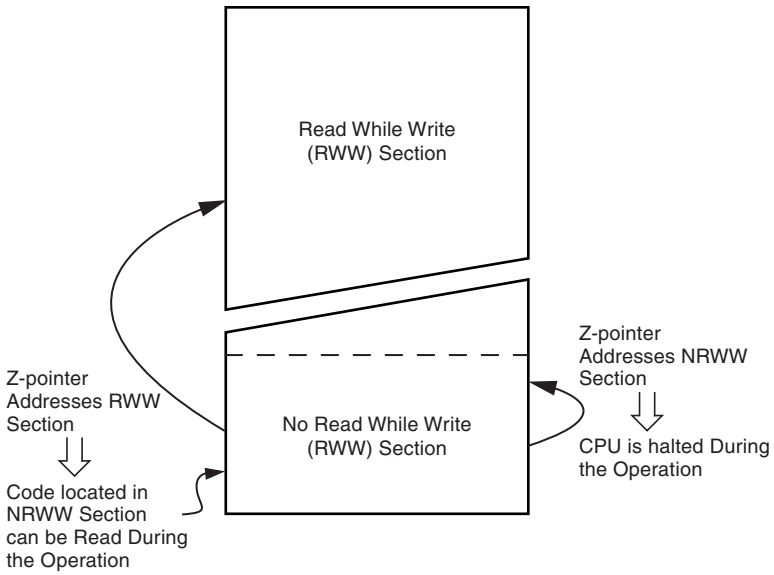
29.4.2 NRWW – No Read-While-Write Section

The code located in the NRWW section can be read when the Boot Loader software is updating a page in the RWW section. When the Boot Loader code updates the NRWW section, the CPU is halted during the entire Page Erase or Page Write operation.

Table 29-1. Read-While-Write Features

Which Section does the Z-pointer Address During the Programming?	Which Section Can be Read During Programming?	CPU Halted?	Read-While-Write Supported?
RWW Section	NRWW Section	No	Yes
NRWW Section	None	Yes	No

Figure 29-1. Read-While-Write versus No Read-While-Write



29.6 Entering the Boot Loader Program

Entering the Boot Loader takes place by a jump or call from the application program. This may be initiated by a trigger such as a command received via the SPI or LIN. Alternatively, the Boot Reset Fuse can be programmed so that the Reset Vector is pointing to the Boot Flash start address after a reset. In this case, the Boot Loader is started after a reset. After the application code is loaded, the program can start executing the application code. Note that the fuses cannot be changed by the MCU itself. This means that once the Boot Reset Fuse is programmed, the Reset Vector will always point to the Boot Loader Reset and the fuse can only be changed through the serial or parallel programming interface.

Table 29-2. Boot Reset Fuse(1)

BOOTRST	Reset Address
1	Reset Vector = Application Reset (address 0x0000)
0	Reset Vector = Boot Loader Reset (see Table 29-5 on page 176)

Note: 1. “1” means unprogrammed, “0” means programmed

29.7 Addressing the Flash During Self-Programming

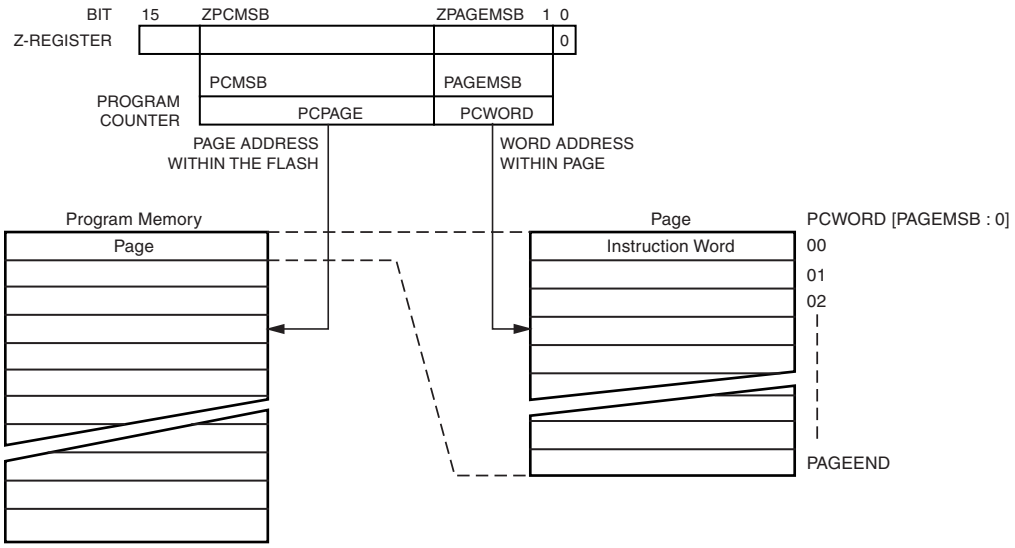
The Z-pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	**Z15**	**Z14**	**Z13**	**Z12**	**Z11**	**Z10**	**Z9**	**Z8**
ZL (R30)	**Z7**	**Z6**	**Z5**	**Z4**	**Z3**	**Z2**	**Z1**	**Z0**
	7	6	5	4	3	2	1	0

Since the Flash is organized in pages (see Section 30.2 “Fuse Bits” on page 181), the Program Counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in Figure 29-3. Note that the Page Erase and Page Write operations are addressed independently. Therefore it is of major importance that the Boot Loader software addresses the same page in both the Page Erase and Page Write operation. Once a programming operation is initiated, the address is latched and the Z-pointer can be used for other operations.

The only SPM operation that does not use the Z-pointer is Setting the Boot Loader Lock bits. The content of the Z-pointer is ignored and will have no effect on the operation. The LPM instruction does also use the Z-pointer to store the address. Since this instruction addresses the Flash byte-by-byte, also the LSB (bit Z0) of the Z-pointer is used.

Figure 29-3. Addressing the Flash During SPM(1)



Note: 1. The different variables used in Figure 29-3 are listed in Section 29.8.13 “Atmel ATmega32HVE Boot Loader Parameters” on page 176 and Section 29.8.14 “Atmel ATmega64HVE Boot Loader Parameters” on page 177.

30. Memory Programming

30.1 Program And Data Memory Lock Bits

The Atmel® AVR MCU provides six Lock bits which can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 30-2. The Lock bits can only be erased to “1” with the Chip Erase command.

Table 30-1. Lock Bit Byte(1)

Lock Bit Byte	Bit No	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12	5	Boot Lock bit	1 (unprogrammed)
BLB11	4	Boot Lock bit	1 (unprogrammed)
BLB02	3	Boot Lock bit	1 (unprogrammed)
BLB01	2	Boot Lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

“1” means unprogrammed, “0” means programmed

Table 30-2. Lock Bit Protection Modes(1)(2)

Memory Lock Bits			Protection Type
LB Mode	LB2	LB1	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.(1)
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Boot Lock bits and Fuse bits are locked in both Serial and Parallel Programming mode.(1)
BLB0 Mode	BLB02	BLB01	
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.

Notes:
1. Program the Fuse bits and Boot Lock bits before programming the LB1 and LB2.
2. “1” means unprogrammed, “0” means programmed

30.6.2 Serial Programming Instruction set

Table 30-10 on page 185 and Figure 30-2 on page 186 describes the Instruction set.

Table 30-10. Serial Programming Instruction Set

Instruction/Operation	Instruction Format			
	Byte 1	Byte 2	Byte 3	Byte4
Programming Enable	$AC	$53	$00	$00
Chip Erase (Program Memory/EEPROM)	$AC	$80	$00	$00
Poll RDY/$\overline{\text{BSY}}$	$F0	$00	$00	data byte out
Load Instructions				
Load Extended Address byte(1)	$4D	$00	Extended adr	$00
Load Program Memory Page, High byte	$48	adr MSB	adr LSB	high data byte in
Load Program Memory Page, Low byte	$40	adr MSB	adr LSB	low data byte in
Load EEPROM Memory Page (page access)	$C1	adr MSB	adr LSB	data byte in
Read Instructions				
Read Program Memory, High byte	$28	adr MSB	adr LSB	high data byte out
Read Program Memory, Low byte	$20	adr MSB	adr LSB	low data byte out
Read EEPROM Memory	$A0	adr MSB	adr LSB	data byte out
Read Lock bits	$58	$00	$00	data byte out
Read Signature Byte	$30	$00	adr LSB	data byte out
Read Fuse bits	$50	$00	$00	data byte out
Read Fuse High bits	$58	$08	$00	data byte out
Read Extended Fuse Bits	$50	$08	$00	data byte out
Read Calibration Byte	$38	$00	$00	data byte out
Write Instructions(6)				
Write Program Memory Page	$4C	adr MSB	adr LSB	$00
Write EEPROM Memory	$C0	adr MSB	adr LSB	data byte in
Write EEPROM Memory Page (page access)	$C2	adr MSB	adr LSB	$00
Write Lock bits	$AC	$E0	$00	data byte in
Write Fuse bits	$AC	$A0	$00	data byte in
Write Fuse High bits	$AC	$A8	$00	data byte in
Write Extended Fuse Bits	$AC	$A4	$00	data byte in

Notes:
1. Not all instructions are applicable for all parts.
2. a = address
3. Bits are programmed ‘0’, unprogrammed ‘1’.
4. To ensure future compatibility, unused Fuses and Lock bits should be unprogrammed (‘1’) .
5. Refer to the correspondig section for Fuse and Lock bits, Calibration and Signature bytes and Page size.
6. Instructions accessing program memory use word address. This address may be random within the page range.
7. See htt://www.atmel.com/avr for Application Notes regarding programming and programmers.

If the LSB in RDY/BSY data byte out is ‘1’, a programming operation is still pending. Wait until this bit returns ‘0’ before the next instruction is carried out.

Within the same page, the low data byte must be loaded prior to the high data byte.

After data is loaded to the page buffer, program the EEPROM page, see Figure 30-2 on page 186.

32. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xFF)	Reserved	–	–	–	–	–	–	–	–	
(0xFE)	Reserved	–	–	–	–	–	–	–	–	
(0xFD)	Reserved	–	–	–	–	–	–	–	–	
(0xFC)	Reserved	–	–	–	–	–	–	–	–	
(0xFB)	Reserved	–	–	–	–	–	–	–	–	
(0xFA)	Reserved	–	–	–	–	–	–	–	–	
(0xF9)	Reserved	–	–	–	–	–	–	–	–	
(0xF8)	Reserved	–	–	–	–	–	–	–	–	
(0xF7)	Reserved	–	–	–	–	–	–	–	–	
(0xF6)	VADAC3	VADAC3[31:24]								158
(0xF5)	VADAC2	VADAC2[23:16]								158
(0xF4)	VADAC1	VADAC1[15:8]								158
(0xF3)	VADAC0	VADAC0[7:0]								158
(0xF2)	VADICH	VADICH[15:8]								158
(0xF1)	VADICL	VADICL[7:0]								158
(0xF0)	CADAC3	CADAC3[31:24]								159
(0xEF)	CADAC2	CADAC2[23:16]								159
(0xEE)	CADAC1	CADAC1[15:8]								159
(0xED)	CADAC0	CADAC0[7:0]								159
(0xEC)	CADICH	CADICH[15:8]								159
(0xEB)	CADICL	CADICL[7:0]								159
(0xEA)	CADRCLH	CADRCLH[15:8]								157
(0xE9)	CADRCLL	CADRCLL[7:0]								157
(0xE8)	ADIMR	–	–	VADACIE	VADICIE	–	CADRCIE	CADACIE	CADICIE	157
(0xE7)	ADIFR	–	–	VADACIF	VADICIF	–	CADRCIF	CADACIF	CADICIF	156
(0xE6)	ADCRE	VADEN	–	VADREFS	VADPDM1	VADPDM0	VAMUX2	VAMUX1	VAMUX0	155
(0xE5)	ADCRD	–	–	CADG2	CADG1	CADG0	CADPDM1	CADPDM0	CADDSEL	154
(0xE4)	ADCRC	CADEN	–	CADRCM1	CADRCM0	CADRCT3	CADRCT2	CADRCT1	CADRCT0	153
(0xE3)	ADCRB	–	–	–	ADIDES1	ADIDES0	ADADES2	ADADES1	ADADES0	152
(0xE2)	ADCRA	–	–	–	–	ADPSEL	ADCMS1	ADCMS0	CKSEL	151
(0xE1)	ADSCSRB	–	VADICPS	VADACRB	VADICRB	–	CADICPS	CADACRB	CADICRB	150
(0xE0)	ADSCSRA	–	–	–	–	–	SBSY	SCMD1	SCMD0	149
(0xDF)	Reserved	–	–	–	–	–	–	–	–	
(0xDE)	Reserved	–	–	–	–	–	–	–	–	
(0xDD)	Reserved	–	–	–	–	–	–	–	–	
(0xDC)	PBOV	PBOVCE	–	–	–	PBOE3	–	–	PBOE0	89
(0xDB)	Reserved	–	–	–	–	–	–	–	–	
(0xDA)	Reserved	–	–	–	–	–	–	–	–	
(0xD9)	Reserved	–	–	–	–	–	–	–	–	
(0xD8)	PLLCSR	–	–	SWEN	LOCK	–	–	PLLCIF	PLLCIE	51
(0xD7)	Reserved	–	–	–	–	–	–	–	–	
(0xD6)	Reserved	–	–	–	–	–	–	–	–	
(0xD5)	Reserved	–	–	–	–	–	–	–	–	
(0xD4)	BGLR	–	–	–	–	–	–	BGPLE	BPGL	164
(0xD3)	BGCRA	BGCN[7:0]								163
(0xD2)	BGCRB	BGCL[7:0]								163
(0xD1)	BGCSRA	–	–	–	–	–		BGSC[2:0]		163
(0xD0)	Reserved	–	–	–	–	–	–	–	–	
(0xCF)	Reserved	–	–	–	–	–	–	–	–	
(0xCE)	Reserved	–	–	–	–	–	–	–	–	
(0xCD)	Reserved	–	–	–	–	–	–	–	–	

Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses $00 - $3F must be used. When addressing I/O registers as data space using LD and ST instructions, $20 must be added to these addresses. The Atmel AVR MCU is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from $60 - $FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

32. Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x99)	Reserved	–	–	–	–	–	–	–	–	
(0x98)	Reserved	–	–	–	–	–	–	–	–	
(0x97)	Reserved	–	–	–	–	–	–	–	–	
(0x96)	Reserved	–	–	–	–	–	–	–	–	
(0x95)	Reserved	–	–	–	–	–	–	–	–	
(0x94)	Reserved	–	–	–	–	–	–	–	–	
(0x93)	Reserved	–	–	–	–	–	–	–	–	
(0x92)	Reserved	–	–	–	–	–	–	–	–	
(0x91)	Reserved	–	–	–	–	–	–	–	–	
(0x90)	Reserved	–	–	–	–	–	–	–	–	
(0x8F)	Reserved	–	–	–	–	–	–	–	–	
(0x8E)	Reserved	–	–	–	–	–	–	–	–	
(0x8D)	Reserved	–	–	–	–	–	–	–	–	
(0x8C)	Reserved	–	–	–	–	–	–	–	–	
(0x8B)	Reserved	–	–	–	–	–	–	–	–	
(0x8A)	Reserved	–	–	–	–	–	–	–	–	
(0x89)	OCR1B	Timer/Counter1 – Output Compare Register B								106
(0x88)	OCR1A	Timer/Counter1 – Output Compare Register A								106
(0x87)	Reserved	–	–	–	–	–	–	–	–	
(0x86)	Reserved	–	–	–	–	–	–	–	–	
(0x85)	TCNT1H	Timer/Counter1 (8 Bit) High Byte								106
(0x84)	TCNT1L	Timer/Counter1 (8 Bit) Low Byte								105
(0x83)	Reserved	–	–	–	–	–	–	–	–	
(0x82)	TCCR1C	–	–	–	–	–	–	ICS11	ICS10	105
(0x81)	TCCR1B	–	–	–	–	–	CS12	CS11	CS10	92
(0x80)	TCCR1A	TCW1	ICEN1	ICNC1	ICES1	–	–	–	WGM10	104
(0x7F)	Reserved	–	–	–	–	–	–	–	–	
(0x7E)	DIDR0	–	–	–	–	–	–	PA1DID	PA0DID	160
(0x7D)	Reserved	–	–	–	–	–	–	–	–	
(0x7C)	Reserved	–	–	–	–	–	–	–	–	
(0x7B)	Reserved	–	–	–	–	–	–	–	–	
(0x7A)	Reserved	–	–	–	–	–	–	–	–	
(0x79)	Reserved	–	–	–	–	–	–	–	–	
(0x78)	Reserved	–	–	–	–	–	–	–	–	
(0x77)	Reserved	–	–	–	–	–	–	–	–	
(0x76)	Reserved	–	–	–	–	–	–	–	–	
(0x75)	Reserved	–	–	–	–	–	–	–	–	
(0x74)	Reserved	–	–	–	–	–	–	–	–	
(0x73)	Reserved	–	–	–	–	–	–	–	–	
(0x72)	Reserved	–	–	–	–	–	–	–	–	
(0x71)	Reserved	–	–	–	–	–	–	–	–	
(0x70)	Reserved	–	–	–	–	–	–	–	–	
(0x6F)	TIMSK1	–	–	–	–	ICIE1	OCIE1B	OCIE1A	TOIE1	106
(0x6E)	TIMSK0	–	–	–	–	ICIE0	OCIE0B	OCIE0A	TOIE0	106
(0x6D)	Reserved	–	–	–	–	–	–	–	–	
(0x6C)	PCMSK1	PCINT[9:2]								77
(0x6B)	PCMSK0	–	–	–	–	–	–	PCINT[1:0]		77
(0x6A)	Reserved	–	–	–	–	–	–	–	–	
(0x69)	EICRA	–	–	–	–	–	–	ISC01	ISC00	75
(0x68)	PCICR	–	–	–	–	–	–	PCIE1	PCIE0	76
(0x67)	SOSCCALB	Slow RC Oscillator Calibration Register B								51

Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses $00 - $3F must be used. When addressing I/O registers as data space using LD and ST instructions, $20 must be added to these addresses. The Atmel AVR MCU is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from $60 - $FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Roa
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Building
1-6-4 Osaki
Shinagawa-ku, Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370