```
;*****************************************************************
;* A P P L I C A T I O N   N O T E   F O R   T H E   A V R   F A M I L Y
;*
;* Number            : AVR000
;* File Name         : ATxmega128A1Udef.inc
;* Title             : Register/Bit Definitions for the ATxmega128A1U
;* Date              : Jan 01 2008
;* Version           : 1.00
;* Support E-mail    : avr@atmel.com
;* Target MCU        : ATxmega128A1U
;*
;* DESCRIPTION
;* When including this file in the assembly program file, all I/O register
;* names and I/O register bit names appearing in the data book can be used.
;* In addition, the six registers forming the three data pointers X, Y and
;* Z have been assigned names XL - ZH. Highest RAM address for Internal
;* SRAM is also defined
;*
;*****************************************************************

#ifndef _ATxmega128A1UDEF_INC_
#define _ATxmega128A1UDEF_INC_


#pragma partinc 0

; ***** SPECIFY DEVICE ***************************************************
.device ATxmega128A1U


.equ    SIGNATURE_000 = 0x1E
.equ    SIGNATURE_001 = 0x97
.equ    SIGNATURE_002 = 0x4C

#pragma AVRPART ADMIN PART_NAME ATxmega128A1U
#pragma AVRPART CORE CORE_VERSION V3XJ


; ***** ABSOLUTE I/O REGISTER LOCATIONS *********************************


;***************************************************************************
;** GPIO - General Purpose IO Registers
;***************************************************************************

.equ GPIO_GPIOR0 = 0        // General Purpose IO Register 0
.equ GPIO_GPIOR1 = 1        // General Purpose IO Register 1
.equ GPIO_GPIOR2 = 2        // General Purpose IO Register 2
.equ GPIO_GPIOR3 = 3        // General Purpose IO Register 3
.equ GPIO_GPIOR4 = 4        // General Purpose IO Register 4
.equ GPIO_GPIOR5 = 5        // General Purpose IO Register 5
.equ GPIO_GPIOR6 = 6        // General Purpose IO Register 6
.equ GPIO_GPIOR7 = 7        // General Purpose IO Register 7
.equ GPIO_GPIOR8 = 8        // General Purpose IO Register 8
.equ GPIO_GPIOR9 = 9        // General Purpose IO Register 9
.equ GPIO_GPIORA = 10       // General Purpose IO Register 10
.equ GPIO_GPIORB = 11       // General Purpose IO Register 11
.equ GPIO_GPIORC = 12       // General Purpose IO Register 12
.equ GPIO_GPIORD = 13       // General Purpose IO Register 13
.equ GPIO_GPIORE = 14       // General Purpose IO Register 14
.equ GPIO_GPIORF = 15       // General Purpose IO Register 15

;***************************************************************************
;** VPORT0 - Virtual Port 0
```

```
67  ;***************************************************************************
68
69  .equ VPORT0_DIR = 16        // I/O Port Data Direction
70  .equ VPORT0_OUT = 17        // I/O Port Output
71  .equ VPORT0_IN = 18     // I/O Port Input
72  .equ VPORT0_INTFLAGS = 19       // Interrupt Flag Register
73
74  ;***************************************************************************
75  ;** VPORT1 - Virtual Port 1
76  ;***************************************************************************
77
78  .equ VPORT1_DIR = 20        // I/O Port Data Direction
79  .equ VPORT1_OUT = 21        // I/O Port Output
80  .equ VPORT1_IN = 22     // I/O Port Input
81  .equ VPORT1_INTFLAGS = 23       // Interrupt Flag Register
82
83  ;***************************************************************************
84  ;** VPORT2 - Virtual Port 2
85  ;***************************************************************************
86
87  .equ VPORT2_DIR = 24        // I/O Port Data Direction
88  .equ VPORT2_OUT = 25        // I/O Port Output
89  .equ VPORT2_IN = 26     // I/O Port Input
90  .equ VPORT2_INTFLAGS = 27       // Interrupt Flag Register
91
92  ;***************************************************************************
93  ;** VPORT3 - Virtual Port 3
94  ;***************************************************************************
95
96  .equ VPORT3_DIR = 28        // I/O Port Data Direction
97  .equ VPORT3_OUT = 29        // I/O Port Output
98  .equ VPORT3_IN = 30     // I/O Port Input
99  .equ VPORT3_INTFLAGS = 31       // Interrupt Flag Register
100
101 ;***************************************************************************
102 ;** OCD - On-Chip Debug System
103 ;***************************************************************************
104
105 .equ OCD_OCDR0 = 46     // OCD Register 0
106 .equ OCD_OCDR1 = 47     // OCD Register 1
107
108 ;***************************************************************************
109 ;** CPU - CPU Registers
110 ;***************************************************************************
111
112 .equ CPU_CCP = 52       // Configuration Change Protection
113 .equ CPU_RAMPD = 56     // Ramp D
114 .equ CPU_RAMPX = 57     // Ramp X
115 .equ CPU_RAMPY = 58     // Ramp Y
116 .equ CPU_RAMPZ = 59     // Ramp Z
117 .equ CPU_EIND = 60      // Extended Indirect Jump
118 .equ CPU_SPL = 61       // Stack Pointer Low
119 .equ CPU_SPH = 62       // Stack Pointer High
120 .equ CPU_SREG = 63      // Status Register
121
122 ;***************************************************************************
123 ;** CLK - Clock System
124 ;***************************************************************************
125
126 .equ CLK_CTRL = 64      // Control Register
127 .equ CLK_PSCTRL = 65        // Prescaler Control Register
128 .equ CLK_LOCK = 66      // Lock register
129 .equ CLK_RTCCTRL = 67       // RTC Control Register
130 .equ CLK_USBCTRL = 68       // USB Control Register
131
132 ;***************************************************************************
```

```
;** SLEEP - Sleep Controller
;****************************************************************************

.equ SLEEP_CTRL = 72        // Control Register

;****************************************************************************
;** OSC - Oscillator Control
;****************************************************************************

.equ OSC_CTRL = 80      // Control Register
.equ OSC_STATUS = 81        // Status Register
.equ OSC_XOSCCTRL = 82      // External Oscillator Control Register
.equ OSC_XOSCFAIL = 83      // Oscillator Failure Detection Register
.equ OSC_RC32KCAL = 84      // 32.768 kHz Internal Oscillator Calibration Register
.equ OSC_PLLCTRL = 85       // PLL Control Register
.equ OSC_DFLLCTRL = 86      // DFLL Control Register

;****************************************************************************
;** DFLLRC32M - DFLL for 32MHz RC Oscillator
;****************************************************************************

.equ DFLLRC32M_CTRL = 96        // Control Register
.equ DFLLRC32M_CALA = 98        // Calibration Register A
.equ DFLLRC32M_CALB = 99        // Calibration Register B
.equ DFLLRC32M_COMP0 = 100      // Oscillator Compare Register 0
.equ DFLLRC32M_COMP1 = 101      // Oscillator Compare Register 1
.equ DFLLRC32M_COMP2 = 102      // Oscillator Compare Register 2

;****************************************************************************
;** DFLLRC2M - DFLL for 2MHz RC Oscillator
;****************************************************************************

.equ DFLLRC2M_CTRL = 104        // Control Register
.equ DFLLRC2M_CALA = 106        // Calibration Register A
.equ DFLLRC2M_CALB = 107        // Calibration Register B
.equ DFLLRC2M_COMP0 = 108       // Oscillator Compare Register 0
.equ DFLLRC2M_COMP1 = 109       // Oscillator Compare Register 1
.equ DFLLRC2M_COMP2 = 110       // Oscillator Compare Register 2

;****************************************************************************
;** PR - Power Reduction
;****************************************************************************

.equ PR_PRGEN = 112     // General Power Reduction
.equ PR_PRPA = 113      // Power Reduction Port A
.equ PR_PRPB = 114      // Power Reduction Port B
.equ PR_PRPC = 115      // Power Reduction Port C
.equ PR_PRPD = 116      // Power Reduction Port D
.equ PR_PRPE = 117      // Power Reduction Port E
.equ PR_PRPF = 118      // Power Reduction Port F

;****************************************************************************
;** RST - Reset Controller
;****************************************************************************

.equ RST_STATUS = 120       // Status Register
.equ RST_CTRL = 121     // Control Register

;****************************************************************************
;** WDT - Watch-Dog Timer
;****************************************************************************

.equ WDT_CTRL = 128     // Control
.equ WDT_WINCTRL = 129      // Windowed Mode Control
.equ WDT_STATUS = 130       // Status

```

```
;**************************************************************************
;** MCU - MCU Control
;**************************************************************************

.equ MCU_DEVID0 = 144       // Device ID byte 0
.equ MCU_DEVID1 = 145       // Device ID byte 1
.equ MCU_DEVID2 = 146       // Device ID byte 2
.equ MCU_REVID = 147        // Revision ID
.equ MCU_JTAGUID = 148      // JTAG User ID
.equ MCU_MCUCR = 150        // MCU Control
.equ MCU_ANAINIT = 151      // Analog Startup Delay
.equ MCU_EVSYSLOCK = 152        // Event System Lock
.equ MCU_AWEXLOCK = 153     // AWEX Lock

;**************************************************************************
;** PMIC - Programmable Interrupt Controller
;**************************************************************************

.equ PMIC_STATUS = 160      // Status Register
.equ PMIC_INTPRI = 161      // Interrupt Priority
.equ PMIC_CTRL = 162        // Control Register

;**************************************************************************
;** PORTCFG - Port Configuration
;**************************************************************************

.equ PORTCFG_MPCMASK = 176      // Multi-pin Configuration Mask
.equ PORTCFG_VPCTRLA = 178      // Virtual Port Control Register A
.equ PORTCFG_VPCTRLB = 179      // Virtual Port Control Register B
.equ PORTCFG_CLKEVOUT = 180     // Clock and Event Out Register
.equ PORTCFG_EVOUTSEL = 182     // Event Output Select

;**************************************************************************
;** AES - AES Crypto Module
;**************************************************************************

.equ AES_CTRL = 192     // AES Control Register
.equ AES_STATUS = 193       // AES Status Register
.equ AES_STATE = 194        // AES State Register
.equ AES_KEY = 195      // AES Key Register
.equ AES_INTCTRL = 196      // AES Interrupt Control Register

;**************************************************************************
;** CRC - CRC Module
;**************************************************************************

.equ CRC_CTRL = 208     // Control Register
.equ CRC_STATUS = 209       // Status Register
.equ CRC_DATAIN = 211       // Data Input
.equ CRC_CHECKSUM0 = 212        // Checksum byte 0
.equ CRC_CHECKSUM1 = 213        // Checksum byte 1
.equ CRC_CHECKSUM2 = 214        // Checksum byte 2
.equ CRC_CHECKSUM3 = 215        // Checksum byte 3

;**************************************************************************
;** DMA - DMA Controller
;**************************************************************************

.equ DMA_CTRL = 256     // Control
.equ DMA_INTFLAGS = 259     // Transfer Interrupt Status
.equ DMA_STATUS = 260       // Status
.equ DMA_TEMP = 262     // Temporary Register For 16/24-bit Access
.equ DMA_CH0_CTRLA = 272        // Channel Control
.equ DMA_CH0_CTRLB = 273        // Channel Control
.equ DMA_CH0_ADDRCTRL = 274     // Address Control
.equ DMA_CH0_TRIGSRC = 275      // Channel Trigger Source
```

```
.equ DMA_CH0_TRFCNT = 276       // Channel Block Transfer Count
.equ DMA_CH0_REPCNT = 278       // Channel Repeat Count
.equ DMA_CH0_SRCADDR0 = 280     // Channel Source Address 0
.equ DMA_CH0_SRCADDR1 = 281     // Channel Source Address 1
.equ DMA_CH0_SRCADDR2 = 282     // Channel Source Address 2
.equ DMA_CH0_DESTADDR0 = 284        // Channel Destination Address 0
.equ DMA_CH0_DESTADDR1 = 285        // Channel Destination Address 1
.equ DMA_CH0_DESTADDR2 = 286        // Channel Destination Address 2
.equ DMA_CH1_CTRLA = 288        // Channel Control
.equ DMA_CH1_CTRLB = 289        // Channel Control
.equ DMA_CH1_ADDRCTRL = 290     // Address Control
.equ DMA_CH1_TRIGSRC = 291      // Channel Trigger Source
.equ DMA_CH1_TRFCNT = 292       // Channel Block Transfer Count
.equ DMA_CH1_REPCNT = 294       // Channel Repeat Count
.equ DMA_CH1_SRCADDR0 = 296     // Channel Source Address 0
.equ DMA_CH1_SRCADDR1 = 297     // Channel Source Address 1
.equ DMA_CH1_SRCADDR2 = 298     // Channel Source Address 2
.equ DMA_CH1_DESTADDR0 = 300        // Channel Destination Address 0
.equ DMA_CH1_DESTADDR1 = 301        // Channel Destination Address 1
.equ DMA_CH1_DESTADDR2 = 302        // Channel Destination Address 2
.equ DMA_CH2_CTRLA = 304        // Channel Control
.equ DMA_CH2_CTRLB = 305        // Channel Control
.equ DMA_CH2_ADDRCTRL = 306     // Address Control
.equ DMA_CH2_TRIGSRC = 307      // Channel Trigger Source
.equ DMA_CH2_TRFCNT = 308       // Channel Block Transfer Count
.equ DMA_CH2_REPCNT = 310       // Channel Repeat Count
.equ DMA_CH2_SRCADDR0 = 312     // Channel Source Address 0
.equ DMA_CH2_SRCADDR1 = 313     // Channel Source Address 1
.equ DMA_CH2_SRCADDR2 = 314     // Channel Source Address 2
.equ DMA_CH2_DESTADDR0 = 316        // Channel Destination Address 0
.equ DMA_CH2_DESTADDR1 = 317        // Channel Destination Address 1
.equ DMA_CH2_DESTADDR2 = 318        // Channel Destination Address 2
.equ DMA_CH3_CTRLA = 320        // Channel Control
.equ DMA_CH3_CTRLB = 321        // Channel Control
.equ DMA_CH3_ADDRCTRL = 322     // Address Control
.equ DMA_CH3_TRIGSRC = 323      // Channel Trigger Source
.equ DMA_CH3_TRFCNT = 324       // Channel Block Transfer Count
.equ DMA_CH3_REPCNT = 326       // Channel Repeat Count
.equ DMA_CH3_SRCADDR0 = 328     // Channel Source Address 0
.equ DMA_CH3_SRCADDR1 = 329     // Channel Source Address 1
.equ DMA_CH3_SRCADDR2 = 330     // Channel Source Address 2
.equ DMA_CH3_DESTADDR0 = 332        // Channel Destination Address 0
.equ DMA_CH3_DESTADDR1 = 333        // Channel Destination Address 1
.equ DMA_CH3_DESTADDR2 = 334        // Channel Destination Address 2

;**************************************************************************
;** EVSYS - Event System
;**************************************************************************

.equ EVSYS_CH0MUX = 384     // Event Channel 0 Multiplexer
.equ EVSYS_CH1MUX = 385     // Event Channel 1 Multiplexer
.equ EVSYS_CH2MUX = 386     // Event Channel 2 Multiplexer
.equ EVSYS_CH3MUX = 387     // Event Channel 3 Multiplexer
.equ EVSYS_CH4MUX = 388     // Event Channel 4 Multiplexer
.equ EVSYS_CH5MUX = 389     // Event Channel 5 Multiplexer
.equ EVSYS_CH6MUX = 390     // Event Channel 6 Multiplexer
.equ EVSYS_CH7MUX = 391     // Event Channel 7 Multiplexer
.equ EVSYS_CH0CTRL = 392        // Channel 0 Control Register
.equ EVSYS_CH1CTRL = 393        // Channel 1 Control Register
.equ EVSYS_CH2CTRL = 394        // Channel 2 Control Register
.equ EVSYS_CH3CTRL = 395        // Channel 3 Control Register
.equ EVSYS_CH4CTRL = 396        // Channel 4 Control Register
.equ EVSYS_CH5CTRL = 397        // Channel 5 Control Register
.equ EVSYS_CH6CTRL = 398        // Channel 6 Control Register
.equ EVSYS_CH7CTRL = 399        // Channel 7 Control Register
.equ EVSYS_STROBE = 400     // Event Strobe
```

```
.equ EVSYS_DATA = 401       // Event Data

;***************************************************************************
;** NVM - Non Volatile Memory
;***************************************************************************

.equ NVM_ADDR0 = 448        // Address Register 0
.equ NVM_ADDR1 = 449        // Address Register 1
.equ NVM_ADDR2 = 450        // Address Register 2
.equ NVM_DATA0 = 452        // Data Register 0
.equ NVM_DATA1 = 453        // Data Register 1
.equ NVM_DATA2 = 454        // Data Register 2
.equ NVM_CMD = 458      // Command
.equ NVM_CTRLA = 459        // Control Register A
.equ NVM_CTRLB = 460        // Control Register B
.equ NVM_INTCTRL = 461      // Interrupt Control
.equ NVM_STATUS = 463       // Status
.equ NVM_LOCKBITS = 464     // Lock Bits

;***************************************************************************
;** ADCA - Analog to Digital Converter A
;***************************************************************************

.equ ADCA_CTRLA = 512       // Control Register A
.equ ADCA_CTRLB = 513       // Control Register B
.equ ADCA_REFCTRL = 514     // Reference Control
.equ ADCA_EVCTRL = 515      // Event Control
.equ ADCA_PRESCALER = 516       // Clock Prescaler
.equ ADCA_INTFLAGS = 518        // Interrupt Flags
.equ ADCA_TEMP = 519        // Temporary Register
.equ ADCA_CAL = 524     // Calibration Value
.equ ADCA_CH0RES = 528      // Channel 0 Result
.equ ADCA_CH1RES = 530      // Channel 1 Result
.equ ADCA_CH2RES = 532      // Channel 2 Result
.equ ADCA_CH3RES = 534      // Channel 3 Result
.equ ADCA_CMP = 536     // Compare Value
.equ ADCA_CH0_CTRL = 544        // Control Register
.equ ADCA_CH0_MUXCTRL = 545     // MUX Control
.equ ADCA_CH0_INTCTRL = 546     // Channel Interrupt Control Register
.equ ADCA_CH0_INTFLAGS = 547        // Interrupt Flags
.equ ADCA_CH0_RES = 548     // Channel Result
.equ ADCA_CH0_SCAN = 550        // Input Channel Scan
.equ ADCA_CH1_CTRL = 552        // Control Register
.equ ADCA_CH1_MUXCTRL = 553     // MUX Control
.equ ADCA_CH1_INTCTRL = 554     // Channel Interrupt Control Register
.equ ADCA_CH1_INTFLAGS = 555        // Interrupt Flags
.equ ADCA_CH1_RES = 556     // Channel Result
.equ ADCA_CH1_SCAN = 558        // Input Channel Scan
.equ ADCA_CH2_CTRL = 560        // Control Register
.equ ADCA_CH2_MUXCTRL = 561     // MUX Control
.equ ADCA_CH2_INTCTRL = 562     // Channel Interrupt Control Register
.equ ADCA_CH2_INTFLAGS = 563        // Interrupt Flags
.equ ADCA_CH2_RES = 564     // Channel Result
.equ ADCA_CH2_SCAN = 566        // Input Channel Scan
.equ ADCA_CH3_CTRL = 568        // Control Register
.equ ADCA_CH3_MUXCTRL = 569     // MUX Control
.equ ADCA_CH3_INTCTRL = 570     // Channel Interrupt Control Register
.equ ADCA_CH3_INTFLAGS = 571        // Interrupt Flags
.equ ADCA_CH3_RES = 572     // Channel Result
.equ ADCA_CH3_SCAN = 574        // Input Channel Scan

;***************************************************************************
;** ADCB - Analog to Digital Converter B
;***************************************************************************

.equ ADCB_CTRLA = 576       // Control Register A
```

```
.equ ADCB_CTRLB = 577       // Control Register B
.equ ADCB_REFCTRL = 578     // Reference Control
.equ ADCB_EVCTRL = 579      // Event Control
.equ ADCB_PRESCALER = 580       // Clock Prescaler
.equ ADCB_INTFLAGS = 582        // Interrupt Flags
.equ ADCB_TEMP = 583        // Temporary Register
.equ ADCB_CAL = 588     // Calibration Value
.equ ADCB_CH0RES = 592      // Channel 0 Result
.equ ADCB_CH1RES = 594      // Channel 1 Result
.equ ADCB_CH2RES = 596      // Channel 2 Result
.equ ADCB_CH3RES = 598      // Channel 3 Result
.equ ADCB_CMP = 600     // Compare Value
.equ ADCB_CH0_CTRL = 608        // Control Register
.equ ADCB_CH0_MUXCTRL = 609     // MUX Control
.equ ADCB_CH0_INTCTRL = 610     // Channel Interrupt Control Register
.equ ADCB_CH0_INTFLAGS = 611        // Interrupt Flags
.equ ADCB_CH0_RES = 612     // Channel Result
.equ ADCB_CH0_SCAN = 614        // Input Channel Scan
.equ ADCB_CH1_CTRL = 616        // Control Register
.equ ADCB_CH1_MUXCTRL = 617     // MUX Control
.equ ADCB_CH1_INTCTRL = 618     // Channel Interrupt Control Register
.equ ADCB_CH1_INTFLAGS = 619        // Interrupt Flags
.equ ADCB_CH1_RES = 620     // Channel Result
.equ ADCB_CH1_SCAN = 622        // Input Channel Scan
.equ ADCB_CH2_CTRL = 624        // Control Register
.equ ADCB_CH2_MUXCTRL = 625     // MUX Control
.equ ADCB_CH2_INTCTRL = 626     // Channel Interrupt Control Register
.equ ADCB_CH2_INTFLAGS = 627        // Interrupt Flags
.equ ADCB_CH2_RES = 628     // Channel Result
.equ ADCB_CH2_SCAN = 630        // Input Channel Scan
.equ ADCB_CH3_CTRL = 632        // Control Register
.equ ADCB_CH3_MUXCTRL = 633     // MUX Control
.equ ADCB_CH3_INTCTRL = 634     // Channel Interrupt Control Register
.equ ADCB_CH3_INTFLAGS = 635        // Interrupt Flags
.equ ADCB_CH3_RES = 636     // Channel Result
.equ ADCB_CH3_SCAN = 638        // Input Channel Scan

;**************************************************************************
;** DACA - Digital to Analog Converter A
;**************************************************************************

.equ DACA_CTRLA = 768       // Control Register A
.equ DACA_CTRLB = 769       // Control Register B
.equ DACA_CTRLC = 770       // Control Register C
.equ DACA_EVCTRL = 771      // Event Input Control
.equ DACA_TIMCTRL = 772     // Timing Control
.equ DACA_STATUS = 773      // Status
.equ DACA_CH0GAINCAL = 776      // Gain Calibration
.equ DACA_CH0OFFSETCAL = 777        // Offset Calibration
.equ DACA_CH1GAINCAL = 778      // Gain Calibration
.equ DACA_CH1OFFSETCAL = 779        // Offset Calibration
.equ DACA_CH0DATA = 792     // Channel 0 Data
.equ DACA_CH1DATA = 794     // Channel 1 Data

;**************************************************************************
;** DACB - Digital to Analog Converter B
;**************************************************************************

.equ DACB_CTRLA = 800       // Control Register A
.equ DACB_CTRLB = 801       // Control Register B
.equ DACB_CTRLC = 802       // Control Register C
.equ DACB_EVCTRL = 803      // Event Input Control
.equ DACB_TIMCTRL = 804     // Timing Control
.equ DACB_STATUS = 805      // Status
.equ DACB_CH0GAINCAL = 808      // Gain Calibration
.equ DACB_CH0OFFSETCAL = 809        // Offset Calibration
```

```
.equ DACB_CH1GAINCAL = 810      // Gain Calibration
.equ DACB_CH1OFFSETCAL = 811        // Offset Calibration
.equ DACB_CH0DATA = 824     // Channel 0 Data
.equ DACB_CH1DATA = 826     // Channel 1 Data

;***************************************************************************
;** ACA - Analog Comparator A
;***************************************************************************

.equ ACA_AC0CTRL = 896      // Analog Comparator 0 Control
.equ ACA_AC1CTRL = 897      // Analog Comparator 1 Control
.equ ACA_AC0MUXCTRL = 898       // Analog Comparator 0 MUX Control
.equ ACA_AC1MUXCTRL = 899       // Analog Comparator 1 MUX Control
.equ ACA_CTRLA = 900        // Control Register A
.equ ACA_CTRLB = 901        // Control Register B
.equ ACA_WINCTRL = 902      // Window Mode Control
.equ ACA_STATUS = 903       // Status

;***************************************************************************
;** ACB - Analog Comparator B
;***************************************************************************

.equ ACB_AC0CTRL = 912      // Analog Comparator 0 Control
.equ ACB_AC1CTRL = 913      // Analog Comparator 1 Control
.equ ACB_AC0MUXCTRL = 914       // Analog Comparator 0 MUX Control
.equ ACB_AC1MUXCTRL = 915       // Analog Comparator 1 MUX Control
.equ ACB_CTRLA = 916        // Control Register A
.equ ACB_CTRLB = 917        // Control Register B
.equ ACB_WINCTRL = 918      // Window Mode Control
.equ ACB_STATUS = 919       // Status

;***************************************************************************
;** RTC - Real-Time Counter
;***************************************************************************

.equ RTC_CTRL = 1024        // Control Register
.equ RTC_STATUS = 1025      // Status Register
.equ RTC_INTCTRL = 1026     // Interrupt Control Register
.equ RTC_INTFLAGS = 1027        // Interrupt Flags
.equ RTC_TEMP = 1028        // Temporary register
.equ RTC_CNT = 1032     // Count Register
.equ RTC_PER = 1034     // Period Register
.equ RTC_COMP = 1036        // Compare Register

;***************************************************************************
;** EBI - External Bus Interface
;***************************************************************************

.equ EBI_CTRL = 1088        // Control
.equ EBI_SDRAMCTRLA = 1089      // SDRAM Control Register A
.equ EBI_REFRESH = 1092     // SDRAM Refresh Period
.equ EBI_INITDLY = 1094     // SDRAM Initialization Delay
.equ EBI_SDRAMCTRLB = 1096      // SDRAM Control Register B
.equ EBI_SDRAMCTRLC = 1097      // SDRAM Control Register C
.equ EBI_CS0_CTRLA = 1104       // Chip Select Control Register A
.equ EBI_CS0_CTRLB = 1105       // Chip Select Control Register B
.equ EBI_CS0_BASEADDR = 1106        // Chip Select Base Address
.equ EBI_CS1_CTRLA = 1108       // Chip Select Control Register A
.equ EBI_CS1_CTRLB = 1109       // Chip Select Control Register B
.equ EBI_CS1_BASEADDR = 1110        // Chip Select Base Address
.equ EBI_CS2_CTRLA = 1112       // Chip Select Control Register A
.equ EBI_CS2_CTRLB = 1113       // Chip Select Control Register B
.equ EBI_CS2_BASEADDR = 1114        // Chip Select Base Address
.equ EBI_CS3_CTRLA = 1116       // Chip Select Control Register A
.equ EBI_CS3_CTRLB = 1117       // Chip Select Control Register B
.equ EBI_CS3_BASEADDR = 1118        // Chip Select Base Address
```

```
529
530   ;**************************************************************************
531   ;** TWIC - Two-Wire Interface C
532   ;**************************************************************************
533
534   .equ TWIC_CTRL = 1152       // TWI Common Control Register
535   .equ TWIC_MASTER_CTRLA = 1153       // Control Register A
536   .equ TWIC_MASTER_CTRLB = 1154       // Control Register B
537   .equ TWIC_MASTER_CTRLC = 1155       // Control Register C
538   .equ TWIC_MASTER_STATUS = 1156      // Status Register
539   .equ TWIC_MASTER_BAUD = 1157        // Baurd Rate Control Register
540   .equ TWIC_MASTER_ADDR = 1158        // Address Register
541   .equ TWIC_MASTER_DATA = 1159        // Data Register
542   .equ TWIC_SLAVE_CTRLA = 1160        // Control Register A
543   .equ TWIC_SLAVE_CTRLB = 1161        // Control Register B
544   .equ TWIC_SLAVE_STATUS = 1162       // Status Register
545   .equ TWIC_SLAVE_ADDR = 1163     // Address Register
546   .equ TWIC_SLAVE_DATA = 1164     // Data Register
547   .equ TWIC_SLAVE_ADDRMASK = 1165     // Address Mask Register
548
549   ;**************************************************************************
550   ;** TWID - Two-Wire Interface D
551   ;**************************************************************************
552
553   .equ TWID_CTRL = 1168       // TWI Common Control Register
554   .equ TWID_MASTER_CTRLA = 1169       // Control Register A
555   .equ TWID_MASTER_CTRLB = 1170       // Control Register B
556   .equ TWID_MASTER_CTRLC = 1171       // Control Register C
557   .equ TWID_MASTER_STATUS = 1172      // Status Register
558   .equ TWID_MASTER_BAUD = 1173        // Baurd Rate Control Register
559   .equ TWID_MASTER_ADDR = 1174        // Address Register
560   .equ TWID_MASTER_DATA = 1175        // Data Register
561   .equ TWID_SLAVE_CTRLA = 1176        // Control Register A
562   .equ TWID_SLAVE_CTRLB = 1177        // Control Register B
563   .equ TWID_SLAVE_STATUS = 1178       // Status Register
564   .equ TWID_SLAVE_ADDR = 1179     // Address Register
565   .equ TWID_SLAVE_DATA = 1180     // Data Register
566   .equ TWID_SLAVE_ADDRMASK = 1181     // Address Mask Register
567
568   ;**************************************************************************
569   ;** TWIE - Two-Wire Interface E
570   ;**************************************************************************
571
572   .equ TWIE_CTRL = 1184       // TWI Common Control Register
573   .equ TWIE_MASTER_CTRLA = 1185       // Control Register A
574   .equ TWIE_MASTER_CTRLB = 1186       // Control Register B
575   .equ TWIE_MASTER_CTRLC = 1187       // Control Register C
576   .equ TWIE_MASTER_STATUS = 1188      // Status Register
577   .equ TWIE_MASTER_BAUD = 1189        // Baurd Rate Control Register
578   .equ TWIE_MASTER_ADDR = 1190        // Address Register
579   .equ TWIE_MASTER_DATA = 1191        // Data Register
580   .equ TWIE_SLAVE_CTRLA = 1192        // Control Register A
581   .equ TWIE_SLAVE_CTRLB = 1193        // Control Register B
582   .equ TWIE_SLAVE_STATUS = 1194       // Status Register
583   .equ TWIE_SLAVE_ADDR = 1195     // Address Register
584   .equ TWIE_SLAVE_DATA = 1196     // Data Register
585   .equ TWIE_SLAVE_ADDRMASK = 1197     // Address Mask Register
586
587   ;**************************************************************************
588   ;** TWIF - Two-Wire Interface F
589   ;**************************************************************************
590
591   .equ TWIF_CTRL = 1200       // TWI Common Control Register
592   .equ TWIF_MASTER_CTRLA = 1201       // Control Register A
593   .equ TWIF_MASTER_CTRLB = 1202       // Control Register B
594   .equ TWIF_MASTER_CTRLC = 1203       // Control Register C
```

```
595   .equ TWIF_MASTER_STATUS = 1204      // Status Register
596   .equ TWIF_MASTER_BAUD = 1205        // Baurd Rate Control Register
597   .equ TWIF_MASTER_ADDR = 1206        // Address Register
598   .equ TWIF_MASTER_DATA = 1207        // Data Register
599   .equ TWIF_SLAVE_CTRLA = 1208        // Control Register A
600   .equ TWIF_SLAVE_CTRLB = 1209        // Control Register B
601   .equ TWIF_SLAVE_STATUS = 1210       // Status Register
602   .equ TWIF_SLAVE_ADDR = 1211     // Address Register
603   .equ TWIF_SLAVE_DATA = 1212     // Data Register
604   .equ TWIF_SLAVE_ADDRMASK = 1213     // Address Mask Register
605
606   ;**************************************************************************
607   ;** USB - Universal Serial Bus
608   ;**************************************************************************
609
610   .equ USB_CTRLA = 1216       // Control Register A
611   .equ USB_CTRLB = 1217       // Control Register B
612   .equ USB_STATUS = 1218      // Status Register
613   .equ USB_ADDR = 1219        // Address Register
614   .equ USB_FIFOWP = 1220      // FIFO Write Pointer Register
615   .equ USB_FIFORP = 1221      // FIFO Read Pointer Register
616   .equ USB_EPPTR = 1222       // Endpoint Configuration Table Pointer
617   .equ USB_INTCTRLA = 1224        // Interrupt Control Register A
618   .equ USB_INTCTRLB = 1225        // Interrupt Control Register B
619   .equ USB_INTFLAGSACLR = 1226        // Clear Interrupt Flag Register A
620   .equ USB_INTFLAGSASET = 1227        // Set Interrupt Flag Register A
621   .equ USB_INTFLAGSBCLR = 1228        // Clear Interrupt Flag Register B
622   .equ USB_INTFLAGSBSET = 1229        // Set Interrupt Flag Register B
623   .equ USB_CAL0 = 1274        // Calibration Byte 0
624   .equ USB_CAL1 = 1275        // Calibration Byte 1
625
626   ;**************************************************************************
627   ;** PORTA - Port A
628   ;**************************************************************************
629
630   .equ PORTA_DIR = 1536       // I/O Port Data Direction
631   .equ PORTA_DIRSET = 1537        // I/O Port Data Direction Set
632   .equ PORTA_DIRCLR = 1538        // I/O Port Data Direction Clear
633   .equ PORTA_DIRTGL = 1539        // I/O Port Data Direction Toggle
634   .equ PORTA_OUT = 1540       // I/O Port Output
635   .equ PORTA_OUTSET = 1541        // I/O Port Output Set
636   .equ PORTA_OUTCLR = 1542        // I/O Port Output Clear
637   .equ PORTA_OUTTGL = 1543        // I/O Port Output Toggle
638   .equ PORTA_IN = 1544        // I/O port Input
639   .equ PORTA_INTCTRL = 1545       // Interrupt Control Register
640   .equ PORTA_INT0MASK = 1546      // Port Interrupt 0 Mask
641   .equ PORTA_INT1MASK = 1547      // Port Interrupt 1 Mask
642   .equ PORTA_INTFLAGS = 1548      // Interrupt Flag Register
643   .equ PORTA_REMAP = 1550     // I/O Port Pin Remap Register
644   .equ PORTA_PIN0CTRL = 1552      // Pin 0 Control Register
645   .equ PORTA_PIN1CTRL = 1553      // Pin 1 Control Register
646   .equ PORTA_PIN2CTRL = 1554      // Pin 2 Control Register
647   .equ PORTA_PIN3CTRL = 1555      // Pin 3 Control Register
648   .equ PORTA_PIN4CTRL = 1556      // Pin 4 Control Register
649   .equ PORTA_PIN5CTRL = 1557      // Pin 5 Control Register
650   .equ PORTA_PIN6CTRL = 1558      // Pin 6 Control Register
651   .equ PORTA_PIN7CTRL = 1559      // Pin 7 Control Register
652
653   ;**************************************************************************
654   ;** PORTB - Port B
655   ;**************************************************************************
656
657   .equ PORTB_DIR = 1568       // I/O Port Data Direction
658   .equ PORTB_DIRSET = 1569        // I/O Port Data Direction Set
659   .equ PORTB_DIRCLR = 1570        // I/O Port Data Direction Clear
660   .equ PORTB_DIRTGL = 1571        // I/O Port Data Direction Toggle
```

```
.equ PORTB_OUT = 1572       // I/O Port Output
.equ PORTB_OUTSET = 1573        // I/O Port Output Set
.equ PORTB_OUTCLR = 1574        // I/O Port Output Clear
.equ PORTB_OUTTGL = 1575        // I/O Port Output Toggle
.equ PORTB_IN = 1576        // I/O port Input
.equ PORTB_INTCTRL = 1577       // Interrupt Control Register
.equ PORTB_INT0MASK = 1578      // Port Interrupt 0 Mask
.equ PORTB_INT1MASK = 1579      // Port Interrupt 1 Mask
.equ PORTB_INTFLAGS = 1580      // Interrupt Flag Register
.equ PORTB_REMAP = 1582     // I/O Port Pin Remap Register
.equ PORTB_PIN0CTRL = 1584      // Pin 0 Control Register
.equ PORTB_PIN1CTRL = 1585      // Pin 1 Control Register
.equ PORTB_PIN2CTRL = 1586      // Pin 2 Control Register
.equ PORTB_PIN3CTRL = 1587      // Pin 3 Control Register
.equ PORTB_PIN4CTRL = 1588      // Pin 4 Control Register
.equ PORTB_PIN5CTRL = 1589      // Pin 5 Control Register
.equ PORTB_PIN6CTRL = 1590      // Pin 6 Control Register
.equ PORTB_PIN7CTRL = 1591      // Pin 7 Control Register

;***************************************************************************
;** PORTC - Port C
;***************************************************************************

.equ PORTC_DIR = 1600       // I/O Port Data Direction
.equ PORTC_DIRSET = 1601        // I/O Port Data Direction Set
.equ PORTC_DIRCLR = 1602        // I/O Port Data Direction Clear
.equ PORTC_DIRTGL = 1603        // I/O Port Data Direction Toggle
.equ PORTC_OUT = 1604       // I/O Port Output
.equ PORTC_OUTSET = 1605        // I/O Port Output Set
.equ PORTC_OUTCLR = 1606        // I/O Port Output Clear
.equ PORTC_OUTTGL = 1607        // I/O Port Output Toggle
.equ PORTC_IN = 1608        // I/O port Input
.equ PORTC_INTCTRL = 1609       // Interrupt Control Register
.equ PORTC_INT0MASK = 1610      // Port Interrupt 0 Mask
.equ PORTC_INT1MASK = 1611      // Port Interrupt 1 Mask
.equ PORTC_INTFLAGS = 1612      // Interrupt Flag Register
.equ PORTC_REMAP = 1614     // I/O Port Pin Remap Register
.equ PORTC_PIN0CTRL = 1616      // Pin 0 Control Register
.equ PORTC_PIN1CTRL = 1617      // Pin 1 Control Register
.equ PORTC_PIN2CTRL = 1618      // Pin 2 Control Register
.equ PORTC_PIN3CTRL = 1619      // Pin 3 Control Register
.equ PORTC_PIN4CTRL = 1620      // Pin 4 Control Register
.equ PORTC_PIN5CTRL = 1621      // Pin 5 Control Register
.equ PORTC_PIN6CTRL = 1622      // Pin 6 Control Register
.equ PORTC_PIN7CTRL = 1623      // Pin 7 Control Register

;***************************************************************************
;** PORTD - Port D
;***************************************************************************

.equ PORTD_DIR = 1632       // I/O Port Data Direction
.equ PORTD_DIRSET = 1633        // I/O Port Data Direction Set
.equ PORTD_DIRCLR = 1634        // I/O Port Data Direction Clear
.equ PORTD_DIRTGL = 1635        // I/O Port Data Direction Toggle
.equ PORTD_OUT = 1636       // I/O Port Output
.equ PORTD_OUTSET = 1637        // I/O Port Output Set
.equ PORTD_OUTCLR = 1638        // I/O Port Output Clear
.equ PORTD_OUTTGL = 1639        // I/O Port Output Toggle
.equ PORTD_IN = 1640        // I/O port Input
.equ PORTD_INTCTRL = 1641       // Interrupt Control Register
.equ PORTD_INT0MASK = 1642      // Port Interrupt 0 Mask
.equ PORTD_INT1MASK = 1643      // Port Interrupt 1 Mask
.equ PORTD_INTFLAGS = 1644      // Interrupt Flag Register
.equ PORTD_REMAP = 1646     // I/O Port Pin Remap Register
.equ PORTD_PIN0CTRL = 1648      // Pin 0 Control Register
.equ PORTD_PIN1CTRL = 1649      // Pin 1 Control Register
```

```
727   .equ PORTD_PIN2CTRL = 1650      // Pin 2 Control Register
728   .equ PORTD_PIN3CTRL = 1651      // Pin 3 Control Register
729   .equ PORTD_PIN4CTRL = 1652      // Pin 4 Control Register
730   .equ PORTD_PIN5CTRL = 1653      // Pin 5 Control Register
731   .equ PORTD_PIN6CTRL = 1654      // Pin 6 Control Register
732   .equ PORTD_PIN7CTRL = 1655      // Pin 7 Control Register
733
734   ;**************************************************************************
735   ;** PORTE - Port E
736   ;**************************************************************************
737
738   .equ PORTE_DIR = 1664       // I/O Port Data Direction
739   .equ PORTE_DIRSET = 1665        // I/O Port Data Direction Set
740   .equ PORTE_DIRCLR = 1666        // I/O Port Data Direction Clear
741   .equ PORTE_DIRTGL = 1667        // I/O Port Data Direction Toggle
742   .equ PORTE_OUT = 1668       // I/O Port Output
743   .equ PORTE_OUTSET = 1669        // I/O Port Output Set
744   .equ PORTE_OUTCLR = 1670        // I/O Port Output Clear
745   .equ PORTE_OUTTGL = 1671        // I/O Port Output Toggle
746   .equ PORTE_IN = 1672        // I/O port Input
747   .equ PORTE_INTCTRL = 1673       // Interrupt Control Register
748   .equ PORTE_INT0MASK = 1674      // Port Interrupt 0 Mask
749   .equ PORTE_INT1MASK = 1675      // Port Interrupt 1 Mask
750   .equ PORTE_INTFLAGS = 1676      // Interrupt Flag Register
751   .equ PORTE_REMAP = 1678     // I/O Port Pin Remap Register
752   .equ PORTE_PIN0CTRL = 1680      // Pin 0 Control Register
753   .equ PORTE_PIN1CTRL = 1681      // Pin 1 Control Register
754   .equ PORTE_PIN2CTRL = 1682      // Pin 2 Control Register
755   .equ PORTE_PIN3CTRL = 1683      // Pin 3 Control Register
756   .equ PORTE_PIN4CTRL = 1684      // Pin 4 Control Register
757   .equ PORTE_PIN5CTRL = 1685      // Pin 5 Control Register
758   .equ PORTE_PIN6CTRL = 1686      // Pin 6 Control Register
759   .equ PORTE_PIN7CTRL = 1687      // Pin 7 Control Register
760
761   ;**************************************************************************
762   ;** PORTF - Port F
763   ;**************************************************************************
764
765   .equ PORTF_DIR = 1696       // I/O Port Data Direction
766   .equ PORTF_DIRSET = 1697        // I/O Port Data Direction Set
767   .equ PORTF_DIRCLR = 1698        // I/O Port Data Direction Clear
768   .equ PORTF_DIRTGL = 1699        // I/O Port Data Direction Toggle
769   .equ PORTF_OUT = 1700       // I/O Port Output
770   .equ PORTF_OUTSET = 1701        // I/O Port Output Set
771   .equ PORTF_OUTCLR = 1702        // I/O Port Output Clear
772   .equ PORTF_OUTTGL = 1703        // I/O Port Output Toggle
773   .equ PORTF_IN = 1704        // I/O port Input
774   .equ PORTF_INTCTRL = 1705       // Interrupt Control Register
775   .equ PORTF_INT0MASK = 1706      // Port Interrupt 0 Mask
776   .equ PORTF_INT1MASK = 1707      // Port Interrupt 1 Mask
777   .equ PORTF_INTFLAGS = 1708      // Interrupt Flag Register
778   .equ PORTF_REMAP = 1710     // I/O Port Pin Remap Register
779   .equ PORTF_PIN0CTRL = 1712      // Pin 0 Control Register
780   .equ PORTF_PIN1CTRL = 1713      // Pin 1 Control Register
781   .equ PORTF_PIN2CTRL = 1714      // Pin 2 Control Register
782   .equ PORTF_PIN3CTRL = 1715      // Pin 3 Control Register
783   .equ PORTF_PIN4CTRL = 1716      // Pin 4 Control Register
784   .equ PORTF_PIN5CTRL = 1717      // Pin 5 Control Register
785   .equ PORTF_PIN6CTRL = 1718      // Pin 6 Control Register
786   .equ PORTF_PIN7CTRL = 1719      // Pin 7 Control Register
787
788   ;**************************************************************************
789   ;** PORTH - Port H
790   ;**************************************************************************
791
792   .equ PORTH_DIR = 1760       // I/O Port Data Direction
```

```
793   .equ PORTH_DIRSET = 1761        // I/O Port Data Direction Set
794   .equ PORTH_DIRCLR = 1762        // I/O Port Data Direction Clear
795   .equ PORTH_DIRTGL = 1763        // I/O Port Data Direction Toggle
796   .equ PORTH_OUT = 1764       // I/O Port Output
797   .equ PORTH_OUTSET = 1765        // I/O Port Output Set
798   .equ PORTH_OUTCLR = 1766        // I/O Port Output Clear
799   .equ PORTH_OUTTGL = 1767        // I/O Port Output Toggle
800   .equ PORTH_IN = 1768        // I/O port Input
801   .equ PORTH_INTCTRL = 1769       // Interrupt Control Register
802   .equ PORTH_INT0MASK = 1770      // Port Interrupt 0 Mask
803   .equ PORTH_INT1MASK = 1771      // Port Interrupt 1 Mask
804   .equ PORTH_INTFLAGS = 1772      // Interrupt Flag Register
805   .equ PORTH_REMAP = 1774     // I/O Port Pin Remap Register
806   .equ PORTH_PIN0CTRL = 1776      // Pin 0 Control Register
807   .equ PORTH_PIN1CTRL = 1777      // Pin 1 Control Register
808   .equ PORTH_PIN2CTRL = 1778      // Pin 2 Control Register
809   .equ PORTH_PIN3CTRL = 1779      // Pin 3 Control Register
810   .equ PORTH_PIN4CTRL = 1780      // Pin 4 Control Register
811   .equ PORTH_PIN5CTRL = 1781      // Pin 5 Control Register
812   .equ PORTH_PIN6CTRL = 1782      // Pin 6 Control Register
813   .equ PORTH_PIN7CTRL = 1783      // Pin 7 Control Register
814
815   ;**************************************************************************
816   ;** PORTJ - Port J
817   ;**************************************************************************
818
819   .equ PORTJ_DIR = 1792       // I/O Port Data Direction
820   .equ PORTJ_DIRSET = 1793        // I/O Port Data Direction Set
821   .equ PORTJ_DIRCLR = 1794        // I/O Port Data Direction Clear
822   .equ PORTJ_DIRTGL = 1795        // I/O Port Data Direction Toggle
823   .equ PORTJ_OUT = 1796       // I/O Port Output
824   .equ PORTJ_OUTSET = 1797        // I/O Port Output Set
825   .equ PORTJ_OUTCLR = 1798        // I/O Port Output Clear
826   .equ PORTJ_OUTTGL = 1799        // I/O Port Output Toggle
827   .equ PORTJ_IN = 1800        // I/O port Input
828   .equ PORTJ_INTCTRL = 1801       // Interrupt Control Register
829   .equ PORTJ_INT0MASK = 1802      // Port Interrupt 0 Mask
830   .equ PORTJ_INT1MASK = 1803      // Port Interrupt 1 Mask
831   .equ PORTJ_INTFLAGS = 1804      // Interrupt Flag Register
832   .equ PORTJ_REMAP = 1806     // I/O Port Pin Remap Register
833   .equ PORTJ_PIN0CTRL = 1808      // Pin 0 Control Register
834   .equ PORTJ_PIN1CTRL = 1809      // Pin 1 Control Register
835   .equ PORTJ_PIN2CTRL = 1810      // Pin 2 Control Register
836   .equ PORTJ_PIN3CTRL = 1811      // Pin 3 Control Register
837   .equ PORTJ_PIN4CTRL = 1812      // Pin 4 Control Register
838   .equ PORTJ_PIN5CTRL = 1813      // Pin 5 Control Register
839   .equ PORTJ_PIN6CTRL = 1814      // Pin 6 Control Register
840   .equ PORTJ_PIN7CTRL = 1815      // Pin 7 Control Register
841
842   ;**************************************************************************
843   ;** PORTK - Port K
844   ;**************************************************************************
845
846   .equ PORTK_DIR = 1824       // I/O Port Data Direction
847   .equ PORTK_DIRSET = 1825        // I/O Port Data Direction Set
848   .equ PORTK_DIRCLR = 1826        // I/O Port Data Direction Clear
849   .equ PORTK_DIRTGL = 1827        // I/O Port Data Direction Toggle
850   .equ PORTK_OUT = 1828       // I/O Port Output
851   .equ PORTK_OUTSET = 1829        // I/O Port Output Set
852   .equ PORTK_OUTCLR = 1830        // I/O Port Output Clear
853   .equ PORTK_OUTTGL = 1831        // I/O Port Output Toggle
854   .equ PORTK_IN = 1832        // I/O port Input
855   .equ PORTK_INTCTRL = 1833       // Interrupt Control Register
856   .equ PORTK_INT0MASK = 1834      // Port Interrupt 0 Mask
857   .equ PORTK_INT1MASK = 1835      // Port Interrupt 1 Mask
858   .equ PORTK_INTFLAGS = 1836      // Interrupt Flag Register
```

```
.equ PORTK_REMAP = 1838     // I/O Port Pin Remap Register
.equ PORTK_PIN0CTRL = 1840      // Pin 0 Control Register
.equ PORTK_PIN1CTRL = 1841      // Pin 1 Control Register
.equ PORTK_PIN2CTRL = 1842      // Pin 2 Control Register
.equ PORTK_PIN3CTRL = 1843      // Pin 3 Control Register
.equ PORTK_PIN4CTRL = 1844      // Pin 4 Control Register
.equ PORTK_PIN5CTRL = 1845      // Pin 5 Control Register
.equ PORTK_PIN6CTRL = 1846      // Pin 6 Control Register
.equ PORTK_PIN7CTRL = 1847      // Pin 7 Control Register

;**************************************************************************
;** PORTQ - Port Q
;**************************************************************************

.equ PORTQ_DIR = 1984       // I/O Port Data Direction
.equ PORTQ_DIRSET = 1985        // I/O Port Data Direction Set
.equ PORTQ_DIRCLR = 1986        // I/O Port Data Direction Clear
.equ PORTQ_DIRTGL = 1987        // I/O Port Data Direction Toggle
.equ PORTQ_OUT = 1988       // I/O Port Output
.equ PORTQ_OUTSET = 1989        // I/O Port Output Set
.equ PORTQ_OUTCLR = 1990        // I/O Port Output Clear
.equ PORTQ_OUTTGL = 1991        // I/O Port Output Toggle
.equ PORTQ_IN = 1992        // I/O port Input
.equ PORTQ_INTCTRL = 1993       // Interrupt Control Register
.equ PORTQ_INT0MASK = 1994      // Port Interrupt 0 Mask
.equ PORTQ_INT1MASK = 1995      // Port Interrupt 1 Mask
.equ PORTQ_INTFLAGS = 1996      // Interrupt Flag Register
.equ PORTQ_REMAP = 1998     // I/O Port Pin Remap Register
.equ PORTQ_PIN0CTRL = 2000      // Pin 0 Control Register
.equ PORTQ_PIN1CTRL = 2001      // Pin 1 Control Register
.equ PORTQ_PIN2CTRL = 2002      // Pin 2 Control Register
.equ PORTQ_PIN3CTRL = 2003      // Pin 3 Control Register
.equ PORTQ_PIN4CTRL = 2004      // Pin 4 Control Register
.equ PORTQ_PIN5CTRL = 2005      // Pin 5 Control Register
.equ PORTQ_PIN6CTRL = 2006      // Pin 6 Control Register
.equ PORTQ_PIN7CTRL = 2007      // Pin 7 Control Register

;**************************************************************************
;** PORTR - Port R
;**************************************************************************

.equ PORTR_DIR = 2016       // I/O Port Data Direction
.equ PORTR_DIRSET = 2017        // I/O Port Data Direction Set
.equ PORTR_DIRCLR = 2018        // I/O Port Data Direction Clear
.equ PORTR_DIRTGL = 2019        // I/O Port Data Direction Toggle
.equ PORTR_OUT = 2020       // I/O Port Output
.equ PORTR_OUTSET = 2021        // I/O Port Output Set
.equ PORTR_OUTCLR = 2022        // I/O Port Output Clear
.equ PORTR_OUTTGL = 2023        // I/O Port Output Toggle
.equ PORTR_IN = 2024        // I/O port Input
.equ PORTR_INTCTRL = 2025       // Interrupt Control Register
.equ PORTR_INT0MASK = 2026      // Port Interrupt 0 Mask
.equ PORTR_INT1MASK = 2027      // Port Interrupt 1 Mask
.equ PORTR_INTFLAGS = 2028      // Interrupt Flag Register
.equ PORTR_REMAP = 2030     // I/O Port Pin Remap Register
.equ PORTR_PIN0CTRL = 2032      // Pin 0 Control Register
.equ PORTR_PIN1CTRL = 2033      // Pin 1 Control Register
.equ PORTR_PIN2CTRL = 2034      // Pin 2 Control Register
.equ PORTR_PIN3CTRL = 2035      // Pin 3 Control Register
.equ PORTR_PIN4CTRL = 2036      // Pin 4 Control Register
.equ PORTR_PIN5CTRL = 2037      // Pin 5 Control Register
.equ PORTR_PIN6CTRL = 2038      // Pin 6 Control Register
.equ PORTR_PIN7CTRL = 2039      // Pin 7 Control Register

;**************************************************************************
;** TCC0 - Timer/Counter C0
```

```
;*****************************************************************************

.equ TCC0_CTRLA = 2048      // Control  Register A
.equ TCC0_CTRLB = 2049      // Control Register B
.equ TCC0_CTRLC = 2050      // Control register C
.equ TCC0_CTRLD = 2051      // Control Register D
.equ TCC0_CTRLE = 2052      // Control Register E
.equ TCC0_INTCTRLA = 2054       // Interrupt Control Register A
.equ TCC0_INTCTRLB = 2055       // Interrupt Control Register B
.equ TCC0_CTRLFCLR = 2056       // Control Register F Clear
.equ TCC0_CTRLFSET = 2057       // Control Register F Set
.equ TCC0_CTRLGCLR = 2058       // Control Register G Clear
.equ TCC0_CTRLGSET = 2059       // Control Register G Set
.equ TCC0_INTFLAGS = 2060       // Interrupt Flag Register
.equ TCC0_TEMP = 2063       // Temporary Register For 16-bit Access
.equ TCC0_CNT = 2080        // Count
.equ TCC0_PER = 2086        // Period
.equ TCC0_CCA = 2088        // Compare or Capture A
.equ TCC0_CCB = 2090        // Compare or Capture B
.equ TCC0_CCC = 2092        // Compare or Capture C
.equ TCC0_CCD = 2094        // Compare or Capture D
.equ TCC0_PERBUF = 2102     // Period Buffer
.equ TCC0_CCABUF = 2104     // Compare Or Capture A Buffer
.equ TCC0_CCBBUF = 2106     // Compare Or Capture B Buffer
.equ TCC0_CCCBUF = 2108     // Compare Or Capture C Buffer
.equ TCC0_CCDBUF = 2110     // Compare Or Capture D Buffer

;*****************************************************************************
;** TCC2 - Timer/Counter C2
;*****************************************************************************

.equ TCC2_CTRLA = 2048      // Control Register A
.equ TCC2_CTRLB = 2049      // Control Register B
.equ TCC2_CTRLC = 2050      // Control register C
.equ TCC2_CTRLE = 2052      // Control Register E
.equ TCC2_INTCTRLA = 2054       // Interrupt Control Register A
.equ TCC2_INTCTRLB = 2055       // Interrupt Control Register B
.equ TCC2_CTRLF = 2057      // Control Register F
.equ TCC2_INTFLAGS = 2060       // Interrupt Flag Register
.equ TCC2_LCNT = 2080       // Low Byte Count
.equ TCC2_HCNT = 2081       // High Byte Count
.equ TCC2_LPER = 2086       // Low Byte Period
.equ TCC2_HPER = 2087       // High Byte Period
.equ TCC2_LCMPA = 2088      // Low Byte Compare A
.equ TCC2_HCMPA = 2089      // High Byte Compare A
.equ TCC2_LCMPB = 2090      // Low Byte Compare B
.equ TCC2_HCMPB = 2091      // High Byte Compare B
.equ TCC2_LCMPC = 2092      // Low Byte Compare C
.equ TCC2_HCMPC = 2093      // High Byte Compare C
.equ TCC2_LCMPD = 2094      // Low Byte Compare D
.equ TCC2_HCMPD = 2095      // High Byte Compare D

;*****************************************************************************
;** TCC1 - Timer/Counter C1
;*****************************************************************************

.equ TCC1_CTRLA = 2112      // Control  Register A
.equ TCC1_CTRLB = 2113      // Control Register B
.equ TCC1_CTRLC = 2114      // Control register C
.equ TCC1_CTRLD = 2115      // Control Register D
.equ TCC1_CTRLE = 2116      // Control Register E
.equ TCC1_INTCTRLA = 2118       // Interrupt Control Register A
.equ TCC1_INTCTRLB = 2119       // Interrupt Control Register B
.equ TCC1_CTRLFCLR = 2120       // Control Register F Clear
.equ TCC1_CTRLFSET = 2121       // Control Register F Set
.equ TCC1_CTRLGCLR = 2122       // Control Register G Clear
```

```
.equ TCC1_CTRLGSET = 2123       // Control Register G Set
.equ TCC1_INTFLAGS = 2124       // Interrupt Flag Register
.equ TCC1_TEMP = 2127       // Temporary Register For 16-bit Access
.equ TCC1_CNT = 2144        // Count
.equ TCC1_PER = 2150        // Period
.equ TCC1_CCA = 2152        // Compare or Capture A
.equ TCC1_CCB = 2154        // Compare or Capture B
.equ TCC1_PERBUF = 2166     // Period Buffer
.equ TCC1_CCABUF = 2168     // Compare Or Capture A Buffer
.equ TCC1_CCBBUF = 2170     // Compare Or Capture B Buffer

;**************************************************************************
;** AWEXC - Advanced Waveform Extension C
;**************************************************************************

.equ AWEXC_CTRL = 2176      // Control Register
.equ AWEXC_FDEMASK = 2178       // Fault Detection Event Mask
.equ AWEXC_FDCTRL = 2179        // Fault Detection Control Register
.equ AWEXC_STATUS = 2180        // Status Register
.equ AWEXC_STATUSSET = 2181     // Status Set Register
.equ AWEXC_DTBOTH = 2182        // Dead Time Both Sides
.equ AWEXC_DTBOTHBUF = 2183     // Dead Time Both Sides Buffer
.equ AWEXC_DTLS = 2184      // Dead Time Low Side
.equ AWEXC_DTHS = 2185      // Dead Time High Side
.equ AWEXC_DTLSBUF = 2186       // Dead Time Low Side Buffer
.equ AWEXC_DTHSBUF = 2187       // Dead Time High Side Buffer
.equ AWEXC_OUTOVEN = 2188       // Output Override Enable

;**************************************************************************
;** HIRESC - High-Resolution Extension C
;**************************************************************************

.equ HIRESC_CTRLA = 2192        // Control Register

;**************************************************************************
;** USARTC0 - Universal Asynchronous Receiver-Transmitter C0
;**************************************************************************

.equ USARTC0_DATA = 2208        // Data Register
.equ USARTC0_STATUS = 2209      // Status Register
.equ USARTC0_CTRLA = 2211       // Control Register A
.equ USARTC0_CTRLB = 2212       // Control Register B
.equ USARTC0_CTRLC = 2213       // Control Register C
.equ USARTC0_BAUDCTRLA = 2214       // Baud Rate Control Register A
.equ USARTC0_BAUDCTRLB = 2215       // Baud Rate Control Register B

;**************************************************************************
;** USARTC1 - Universal Asynchronous Receiver-Transmitter C1
;**************************************************************************

.equ USARTC1_DATA = 2224        // Data Register
.equ USARTC1_STATUS = 2225      // Status Register
.equ USARTC1_CTRLA = 2227       // Control Register A
.equ USARTC1_CTRLB = 2228       // Control Register B
.equ USARTC1_CTRLC = 2229       // Control Register C
.equ USARTC1_BAUDCTRLA = 2230       // Baud Rate Control Register A
.equ USARTC1_BAUDCTRLB = 2231       // Baud Rate Control Register B

;**************************************************************************
;** SPIC - Serial Peripheral Interface C
;**************************************************************************

.equ SPIC_CTRL = 2240       // Control Register
.equ SPIC_INTCTRL = 2241        // Interrupt Control Register
.equ SPIC_STATUS = 2242     // Status Register
.equ SPIC_DATA = 2243       // Data Register
```

```
;***************************************************************************
;** IRCOM - IR Communication Module
;***************************************************************************

.equ IRCOM_CTRL = 2296      // Control Register
.equ IRCOM_TXPLCTRL = 2297      // IrDA Transmitter Pulse Length Control Register
.equ IRCOM_RXPLCTRL = 2298      // IrDA Receiver Pulse Length Control Register

;***************************************************************************
;** TCD0 - Timer/Counter D0
;***************************************************************************

.equ TCD0_CTRLA = 2304      // Control  Register A
.equ TCD0_CTRLB = 2305      // Control Register B
.equ TCD0_CTRLC = 2306      // Control register C
.equ TCD0_CTRLD = 2307      // Control Register D
.equ TCD0_CTRLE = 2308      // Control Register E
.equ TCD0_INTCTRLA = 2310       // Interrupt Control Register A
.equ TCD0_INTCTRLB = 2311       // Interrupt Control Register B
.equ TCD0_CTRLFCLR = 2312       // Control Register F Clear
.equ TCD0_CTRLFSET = 2313       // Control Register F Set
.equ TCD0_CTRLGCLR = 2314       // Control Register G Clear
.equ TCD0_CTRLGSET = 2315       // Control Register G Set
.equ TCD0_INTFLAGS = 2316       // Interrupt Flag Register
.equ TCD0_TEMP = 2319       // Temporary Register For 16-bit Access
.equ TCD0_CNT = 2336        // Count
.equ TCD0_PER = 2342        // Period
.equ TCD0_CCA = 2344        // Compare or Capture A
.equ TCD0_CCB = 2346        // Compare or Capture B
.equ TCD0_CCC = 2348        // Compare or Capture C
.equ TCD0_CCD = 2350        // Compare or Capture D
.equ TCD0_PERBUF = 2358     // Period Buffer
.equ TCD0_CCABUF = 2360     // Compare Or Capture A Buffer
.equ TCD0_CCBBUF = 2362     // Compare Or Capture B Buffer
.equ TCD0_CCCBUF = 2364     // Compare Or Capture C Buffer
.equ TCD0_CCDBUF = 2366     // Compare Or Capture D Buffer

;***************************************************************************
;** TCD2 - Timer/Counter D2
;***************************************************************************

.equ TCD2_CTRLA = 2304      // Control Register A
.equ TCD2_CTRLB = 2305      // Control Register B
.equ TCD2_CTRLC = 2306      // Control register C
.equ TCD2_CTRLE = 2308      // Control Register E
.equ TCD2_INTCTRLA = 2310       // Interrupt Control Register A
.equ TCD2_INTCTRLB = 2311       // Interrupt Control Register B
.equ TCD2_CTRLF = 2313      // Control Register F
.equ TCD2_INTFLAGS = 2316       // Interrupt Flag Register
.equ TCD2_LCNT = 2336       // Low Byte Count
.equ TCD2_HCNT = 2337       // High Byte Count
.equ TCD2_LPER = 2342       // Low Byte Period
.equ TCD2_HPER = 2343       // High Byte Period
.equ TCD2_LCMPA = 2344      // Low Byte Compare A
.equ TCD2_HCMPA = 2345      // High Byte Compare A
.equ TCD2_LCMPB = 2346      // Low Byte Compare B
.equ TCD2_HCMPB = 2347      // High Byte Compare B
.equ TCD2_LCMPC = 2348      // Low Byte Compare C
.equ TCD2_HCMPC = 2349      // High Byte Compare C
.equ TCD2_LCMPD = 2350      // Low Byte Compare D
.equ TCD2_HCMPD = 2351      // High Byte Compare D

;***************************************************************************
;** TCD1 - Timer/Counter D1
;***************************************************************************
```

```
1123
1124    .equ TCD1_CTRLA = 2368      // Control  Register A
1125    .equ TCD1_CTRLB = 2369      // Control Register B
1126    .equ TCD1_CTRLC = 2370      // Control register C
1127    .equ TCD1_CTRLD = 2371      // Control Register D
1128    .equ TCD1_CTRLE = 2372      // Control Register E
1129    .equ TCD1_INTCTRLA = 2374       // Interrupt Control Register A
1130    .equ TCD1_INTCTRLB = 2375       // Interrupt Control Register B
1131    .equ TCD1_CTRLFCLR = 2376       // Control Register F Clear
1132    .equ TCD1_CTRLFSET = 2377       // Control Register F Set
1133    .equ TCD1_CTRLGCLR = 2378       // Control Register G Clear
1134    .equ TCD1_CTRLGSET = 2379       // Control Register G Set
1135    .equ TCD1_INTFLAGS = 2380       // Interrupt Flag Register
1136    .equ TCD1_TEMP = 2383       // Temporary Register For 16-bit Access
1137    .equ TCD1_CNT = 2400        // Count
1138    .equ TCD1_PER = 2406        // Period
1139    .equ TCD1_CCA = 2408        // Compare or Capture A
1140    .equ TCD1_CCB = 2410        // Compare or Capture B
1141    .equ TCD1_PERBUF = 2422     // Period Buffer
1142    .equ TCD1_CCABUF = 2424     // Compare Or Capture A Buffer
1143    .equ TCD1_CCBBUF = 2426     // Compare Or Capture B Buffer
1144
1145    ;**************************************************************************
1146    ;** HIRESD - High-Resolution Extension D
1147    ;**************************************************************************
1148
1149    .equ HIRESD_CTRLA = 2448        // Control Register
1150
1151    ;**************************************************************************
1152    ;** USARTD0 - Universal Asynchronous Receiver-Transmitter D0
1153    ;**************************************************************************
1154
1155    .equ USARTD0_DATA = 2464        // Data Register
1156    .equ USARTD0_STATUS = 2465      // Status Register
1157    .equ USARTD0_CTRLA = 2467       // Control Register A
1158    .equ USARTD0_CTRLB = 2468       // Control Register B
1159    .equ USARTD0_CTRLC = 2469       // Control Register C
1160    .equ USARTD0_BAUDCTRLA = 2470       // Baud Rate Control Register A
1161    .equ USARTD0_BAUDCTRLB = 2471       // Baud Rate Control Register B
1162
1163    ;**************************************************************************
1164    ;** USARTD1 - Universal Asynchronous Receiver-Transmitter D1
1165    ;**************************************************************************
1166
1167    .equ USARTD1_DATA = 2480        // Data Register
1168    .equ USARTD1_STATUS = 2481      // Status Register
1169    .equ USARTD1_CTRLA = 2483       // Control Register A
1170    .equ USARTD1_CTRLB = 2484       // Control Register B
1171    .equ USARTD1_CTRLC = 2485       // Control Register C
1172    .equ USARTD1_BAUDCTRLA = 2486       // Baud Rate Control Register A
1173    .equ USARTD1_BAUDCTRLB = 2487       // Baud Rate Control Register B
1174
1175    ;**************************************************************************
1176    ;** SPID - Serial Peripheral Interface D
1177    ;**************************************************************************
1178
1179    .equ SPID_CTRL = 2496       // Control Register
1180    .equ SPID_INTCTRL = 2497        // Interrupt Control Register
1181    .equ SPID_STATUS = 2498     // Status Register
1182    .equ SPID_DATA = 2499       // Data Register
1183
1184    ;**************************************************************************
1185    ;** TCE0 - Timer/Counter E0
1186    ;**************************************************************************
1187
1188    .equ TCE0_CTRLA = 2560      // Control  Register A
```

```
1189  .equ TCE0_CTRLB = 2561      // Control Register B
1190  .equ TCE0_CTRLC = 2562      // Control register C
1191  .equ TCE0_CTRLD = 2563      // Control Register D
1192  .equ TCE0_CTRLE = 2564      // Control Register E
1193  .equ TCE0_INTCTRLA = 2566       // Interrupt Control Register A
1194  .equ TCE0_INTCTRLB = 2567       // Interrupt Control Register B
1195  .equ TCE0_CTRLFCLR = 2568       // Control Register F Clear
1196  .equ TCE0_CTRLFSET = 2569       // Control Register F Set
1197  .equ TCE0_CTRLGCLR = 2570       // Control Register G Clear
1198  .equ TCE0_CTRLGSET = 2571       // Control Register G Set
1199  .equ TCE0_INTFLAGS = 2572       // Interrupt Flag Register
1200  .equ TCE0_TEMP = 2575       // Temporary Register For 16-bit Access
1201  .equ TCE0_CNT = 2592        // Count
1202  .equ TCE0_PER = 2598        // Period
1203  .equ TCE0_CCA = 2600        // Compare or Capture A
1204  .equ TCE0_CCB = 2602        // Compare or Capture B
1205  .equ TCE0_CCC = 2604        // Compare or Capture C
1206  .equ TCE0_CCD = 2606        // Compare or Capture D
1207  .equ TCE0_PERBUF = 2614     // Period Buffer
1208  .equ TCE0_CCABUF = 2616     // Compare Or Capture A Buffer
1209  .equ TCE0_CCBBUF = 2618     // Compare Or Capture B Buffer
1210  .equ TCE0_CCCBUF = 2620     // Compare Or Capture C Buffer
1211  .equ TCE0_CCDBUF = 2622     // Compare Or Capture D Buffer
1212
1213  ;**************************************************************************
1214  ;** TCE2 - Timer/Counter E2
1215  ;**************************************************************************
1216
1217  .equ TCE2_CTRLA = 2560      // Control Register A
1218  .equ TCE2_CTRLB = 2561      // Control Register B
1219  .equ TCE2_CTRLC = 2562      // Control register C
1220  .equ TCE2_CTRLE = 2564      // Control Register E
1221  .equ TCE2_INTCTRLA = 2566       // Interrupt Control Register A
1222  .equ TCE2_INTCTRLB = 2567       // Interrupt Control Register B
1223  .equ TCE2_CTRLF = 2569      // Control Register F
1224  .equ TCE2_INTFLAGS = 2572       // Interrupt Flag Register
1225  .equ TCE2_LCNT = 2592       // Low Byte Count
1226  .equ TCE2_HCNT = 2593       // High Byte Count
1227  .equ TCE2_LPER = 2598       // Low Byte Period
1228  .equ TCE2_HPER = 2599       // High Byte Period
1229  .equ TCE2_LCMPA = 2600      // Low Byte Compare A
1230  .equ TCE2_HCMPA = 2601      // High Byte Compare A
1231  .equ TCE2_LCMPB = 2602      // Low Byte Compare B
1232  .equ TCE2_HCMPB = 2603      // High Byte Compare B
1233  .equ TCE2_LCMPC = 2604      // Low Byte Compare C
1234  .equ TCE2_HCMPC = 2605      // High Byte Compare C
1235  .equ TCE2_LCMPD = 2606      // Low Byte Compare D
1236  .equ TCE2_HCMPD = 2607      // High Byte Compare D
1237
1238  ;**************************************************************************
1239  ;** TCE1 - Timer/Counter E1
1240  ;**************************************************************************
1241
1242  .equ TCE1_CTRLA = 2624      // Control  Register A
1243  .equ TCE1_CTRLB = 2625      // Control Register B
1244  .equ TCE1_CTRLC = 2626      // Control register C
1245  .equ TCE1_CTRLD = 2627      // Control Register D
1246  .equ TCE1_CTRLE = 2628      // Control Register E
1247  .equ TCE1_INTCTRLA = 2630       // Interrupt Control Register A
1248  .equ TCE1_INTCTRLB = 2631       // Interrupt Control Register B
1249  .equ TCE1_CTRLFCLR = 2632       // Control Register F Clear
1250  .equ TCE1_CTRLFSET = 2633       // Control Register F Set
1251  .equ TCE1_CTRLGCLR = 2634       // Control Register G Clear
1252  .equ TCE1_CTRLGSET = 2635       // Control Register G Set
1253  .equ TCE1_INTFLAGS = 2636       // Interrupt Flag Register
1254  .equ TCE1_TEMP = 2639       // Temporary Register For 16-bit Access
```

```
1255   .equ TCE1_CNT = 2656        // Count
1256   .equ TCE1_PER = 2662        // Period
1257   .equ TCE1_CCA = 2664        // Compare or Capture A
1258   .equ TCE1_CCB = 2666        // Compare or Capture B
1259   .equ TCE1_PERBUF = 2678     // Period Buffer
1260   .equ TCE1_CCABUF = 2680     // Compare Or Capture A Buffer
1261   .equ TCE1_CCBBUF = 2682     // Compare Or Capture B Buffer
1262
1263   ;**************************************************************************
1264   ;** AWEXE - Advanced Waveform Extension E
1265   ;**************************************************************************
1266
1267   .equ AWEXE_CTRL = 2688      // Control Register
1268   .equ AWEXE_FDEMASK = 2690       // Fault Detection Event Mask
1269   .equ AWEXE_FDCTRL = 2691        // Fault Detection Control Register
1270   .equ AWEXE_STATUS = 2692        // Status Register
1271   .equ AWEXE_STATUSSET = 2693     // Status Set Register
1272   .equ AWEXE_DTBOTH = 2694        // Dead Time Both Sides
1273   .equ AWEXE_DTBOTHBUF = 2695     // Dead Time Both Sides Buffer
1274   .equ AWEXE_DTLS = 2696      // Dead Time Low Side
1275   .equ AWEXE_DTHS = 2697      // Dead Time High Side
1276   .equ AWEXE_DTLSBUF = 2698       // Dead Time Low Side Buffer
1277   .equ AWEXE_DTHSBUF = 2699       // Dead Time High Side Buffer
1278   .equ AWEXE_OUTOVEN = 2700       // Output Override Enable
1279
1280   ;**************************************************************************
1281   ;** HIRESE - High-Resolution Extension E
1282   ;**************************************************************************
1283
1284   .equ HIRESE_CTRLA = 2704        // Control Register
1285
1286   ;**************************************************************************
1287   ;** USARTE0 - Universal Asynchronous Receiver-Transmitter E0
1288   ;**************************************************************************
1289
1290   .equ USARTE0_DATA = 2720        // Data Register
1291   .equ USARTE0_STATUS = 2721      // Status Register
1292   .equ USARTE0_CTRLA = 2723       // Control Register A
1293   .equ USARTE0_CTRLB = 2724       // Control Register B
1294   .equ USARTE0_CTRLC = 2725       // Control Register C
1295   .equ USARTE0_BAUDCTRLA = 2726       // Baud Rate Control Register A
1296   .equ USARTE0_BAUDCTRLB = 2727       // Baud Rate Control Register B
1297
1298   ;**************************************************************************
1299   ;** USARTE1 - Universal Asynchronous Receiver-Transmitter E1
1300   ;**************************************************************************
1301
1302   .equ USARTE1_DATA = 2736        // Data Register
1303   .equ USARTE1_STATUS = 2737      // Status Register
1304   .equ USARTE1_CTRLA = 2739       // Control Register A
1305   .equ USARTE1_CTRLB = 2740       // Control Register B
1306   .equ USARTE1_CTRLC = 2741       // Control Register C
1307   .equ USARTE1_BAUDCTRLA = 2742       // Baud Rate Control Register A
1308   .equ USARTE1_BAUDCTRLB = 2743       // Baud Rate Control Register B
1309
1310   ;**************************************************************************
1311   ;** SPIE - Serial Peripheral Interface E
1312   ;**************************************************************************
1313
1314   .equ SPIE_CTRL = 2752       // Control Register
1315   .equ SPIE_INTCTRL = 2753        // Interrupt Control Register
1316   .equ SPIE_STATUS = 2754     // Status Register
1317   .equ SPIE_DATA = 2755       // Data Register
1318
1319   ;**************************************************************************
1320   ;** TCF0 - Timer/Counter F0
```

```
;*****************************************************************************

.equ TCF0_CTRLA = 2816      // Control  Register A
.equ TCF0_CTRLB = 2817      // Control Register B
.equ TCF0_CTRLC = 2818      // Control register C
.equ TCF0_CTRLD = 2819      // Control Register D
.equ TCF0_CTRLE = 2820      // Control Register E
.equ TCF0_INTCTRLA = 2822       // Interrupt Control Register A
.equ TCF0_INTCTRLB = 2823       // Interrupt Control Register B
.equ TCF0_CTRLFCLR = 2824       // Control Register F Clear
.equ TCF0_CTRLFSET = 2825       // Control Register F Set
.equ TCF0_CTRLGCLR = 2826       // Control Register G Clear
.equ TCF0_CTRLGSET = 2827       // Control Register G Set
.equ TCF0_INTFLAGS = 2828       // Interrupt Flag Register
.equ TCF0_TEMP = 2831       // Temporary Register For 16-bit Access
.equ TCF0_CNT = 2848        // Count
.equ TCF0_PER = 2854        // Period
.equ TCF0_CCA = 2856        // Compare or Capture A
.equ TCF0_CCB = 2858        // Compare or Capture B
.equ TCF0_CCC = 2860        // Compare or Capture C
.equ TCF0_CCD = 2862        // Compare or Capture D
.equ TCF0_PERBUF = 2870     // Period Buffer
.equ TCF0_CCABUF = 2872     // Compare Or Capture A Buffer
.equ TCF0_CCBBUF = 2874     // Compare Or Capture B Buffer
.equ TCF0_CCCBUF = 2876     // Compare Or Capture C Buffer
.equ TCF0_CCDBUF = 2878     // Compare Or Capture D Buffer

;*****************************************************************************
;** TCF2 - Timer/Counter F2
;*****************************************************************************

.equ TCF2_CTRLA = 2816      // Control Register A
.equ TCF2_CTRLB = 2817      // Control Register B
.equ TCF2_CTRLC = 2818      // Control register C
.equ TCF2_CTRLE = 2820      // Control Register E
.equ TCF2_INTCTRLA = 2822       // Interrupt Control Register A
.equ TCF2_INTCTRLB = 2823       // Interrupt Control Register B
.equ TCF2_CTRLF = 2825      // Control Register F
.equ TCF2_INTFLAGS = 2828       // Interrupt Flag Register
.equ TCF2_LCNT = 2848       // Low Byte Count
.equ TCF2_HCNT = 2849       // High Byte Count
.equ TCF2_LPER = 2854       // Low Byte Period
.equ TCF2_HPER = 2855       // High Byte Period
.equ TCF2_LCMPA = 2856      // Low Byte Compare A
.equ TCF2_HCMPA = 2857      // High Byte Compare A
.equ TCF2_LCMPB = 2858      // Low Byte Compare B
.equ TCF2_HCMPB = 2859      // High Byte Compare B
.equ TCF2_LCMPC = 2860      // Low Byte Compare C
.equ TCF2_HCMPC = 2861      // High Byte Compare C
.equ TCF2_LCMPD = 2862      // Low Byte Compare D
.equ TCF2_HCMPD = 2863      // High Byte Compare D

;*****************************************************************************
;** TCF1 - Timer/Counter F1
;*****************************************************************************

.equ TCF1_CTRLA = 2880      // Control  Register A
.equ TCF1_CTRLB = 2881      // Control Register B
.equ TCF1_CTRLC = 2882      // Control register C
.equ TCF1_CTRLD = 2883      // Control Register D
.equ TCF1_CTRLE = 2884      // Control Register E
.equ TCF1_INTCTRLA = 2886       // Interrupt Control Register A
.equ TCF1_INTCTRLB = 2887       // Interrupt Control Register B
.equ TCF1_CTRLFCLR = 2888       // Control Register F Clear
.equ TCF1_CTRLFSET = 2889       // Control Register F Set
.equ TCF1_CTRLGCLR = 2890       // Control Register G Clear
```

```
.equ TCF1_CTRLGSET = 2891       // Control Register G Set
.equ TCF1_INTFLAGS = 2892       // Interrupt Flag Register
.equ TCF1_TEMP = 2895       // Temporary Register For 16-bit Access
.equ TCF1_CNT = 2912        // Count
.equ TCF1_PER = 2918        // Period
.equ TCF1_CCA = 2920        // Compare or Capture A
.equ TCF1_CCB = 2922        // Compare or Capture B
.equ TCF1_PERBUF = 2934     // Period Buffer
.equ TCF1_CCABUF = 2936     // Compare Or Capture A Buffer
.equ TCF1_CCBBUF = 2938     // Compare Or Capture B Buffer

;**************************************************************************
;** HIRESF - High-Resolution Extension F
;**************************************************************************

.equ HIRESF_CTRLA = 2960        // Control Register

;**************************************************************************
;** USARTF0 - Universal Asynchronous Receiver-Transmitter F0
;**************************************************************************

.equ USARTF0_DATA = 2976        // Data Register
.equ USARTF0_STATUS = 2977      // Status Register
.equ USARTF0_CTRLA = 2979       // Control Register A
.equ USARTF0_CTRLB = 2980       // Control Register B
.equ USARTF0_CTRLC = 2981       // Control Register C
.equ USARTF0_BAUDCTRLA = 2982       // Baud Rate Control Register A
.equ USARTF0_BAUDCTRLB = 2983       // Baud Rate Control Register B

;**************************************************************************
;** USARTF1 - Universal Asynchronous Receiver-Transmitter F1
;**************************************************************************

.equ USARTF1_DATA = 2992        // Data Register
.equ USARTF1_STATUS = 2993      // Status Register
.equ USARTF1_CTRLA = 2995       // Control Register A
.equ USARTF1_CTRLB = 2996       // Control Register B
.equ USARTF1_CTRLC = 2997       // Control Register C
.equ USARTF1_BAUDCTRLA = 2998       // Baud Rate Control Register A
.equ USARTF1_BAUDCTRLB = 2999       // Baud Rate Control Register B

;**************************************************************************
;** SPIF - Serial Peripheral Interface F
;**************************************************************************

.equ SPIF_CTRL = 3008       // Control Register
.equ SPIF_INTCTRL = 3009        // Interrupt Control Register
.equ SPIF_STATUS = 3010     // Status Register
.equ SPIF_DATA = 3011       // Data Register


; ***** ALL MODULE BASE ADRESSES ****************************************

.equ GPIO_base = 0x0000     // General Purpose IO Registers
.equ VPORT0_base = 0x0010       // Virtual Port 0
.equ VPORT1_base = 0x0014       // Virtual Port 1
.equ VPORT2_base = 0x0018       // Virtual Port 2
.equ VPORT3_base = 0x001C       // Virtual Port 3
.equ OCD_base = 0x002E      // On-Chip Debug System
.equ CPU_base = 0x0030      // CPU Registers
.equ CLK_base = 0x0040      // Clock System
.equ SLEEP_base = 0x0048        // Sleep Controller
.equ OSC_base = 0x0050      // Oscillator Control
.equ DFLLRC32M_base = 0x0060        // DFLL for 32MHz RC Oscillator
.equ DFLLRC2M_base = 0x0068     // DFLL for 2MHz RC Oscillator
.equ PR_base = 0x0070       // Power Reduction
```

```
1453   .equ RST_base = 0x0078      // Reset Controller
1454   .equ WDT_base = 0x0080      // Watch-Dog Timer
1455   .equ MCU_base = 0x0090      // MCU Control
1456   .equ PMIC_base = 0x00A0     // Programmable Interrupt Controller
1457   .equ PORTCFG_base = 0x00B0      // Port Configuration
1458   .equ AES_base = 0x0C0       // AES Crypto Module
1459   .equ CRC_base = 0x0D0       // CRC Module
1460   .equ DMA_base = 0x0100      // DMA Controller
1461   .equ EVSYS_base = 0x0180        // Event System
1462   .equ NVM_base = 0x01C0      // Non Volatile Memory Controller
1463   .equ ADCA_base = 0x0200     // Analog to Digital Converter A
1464   .equ ADCB_base = 0x0240     // Analog to Digital Converter B
1465   .equ DACA_base = 0x0300     // Digital to Analog Converter A
1466   .equ DACB_base = 0x0320     // Digital to Analog Converter B
1467   .equ ACA_base = 0x0380      // Analog Comparator A
1468   .equ ACB_base = 0x0390      // Analog Comparator B
1469   .equ RTC_base = 0x0400      // Real-Time Counter
1470   .equ EBI_base = 0x0440      // External Bus Interface
1471   .equ TWIC_base = 0x480      // Two-Wire Interface C
1472   .equ TWID_base = 0x490      // Two-Wire Interface D
1473   .equ TWIE_base = 0x4A0      // Two-Wire Interface E
1474   .equ TWIF_base = 0x4B0      // Two-Wire Interface F
1475   .equ USB_base = 0x4C0       // Universal Serial Bus
1476   .equ PORTA_base = 0x0600        // Port A
1477   .equ PORTB_base = 0x0620        // Port B
1478   .equ PORTC_base = 0x0640        // Port C
1479   .equ PORTD_base = 0x0660        // Port D
1480   .equ PORTE_base = 0x0680        // Port E
1481   .equ PORTF_base = 0x06A0        // Port F
1482   .equ PORTH_base = 0x06E0        // Port H
1483   .equ PORTJ_base = 0x0700        // Port J
1484   .equ PORTK_base = 0x0720        // Port K
1485   .equ PORTQ_base = 0x07C0        // Port Q
1486   .equ PORTR_base = 0x07E0        // Port R
1487   .equ TCC0_base = 0x800      // Timer/Counter C0
1488   .equ TCC2_base = 0x800      // Timer/Counter C2
1489   .equ TCC1_base = 0x840      // Timer/Counter C1
1490   .equ AWEXC_base = 0x880     // Advanced Waveform Extension C
1491   .equ HIRESC_base = 0x890        // High-Resolution Extension C
1492   .equ USARTC0_base = 0x8A0       // Universal Asynchronous Receiver-Transmitter C0
1493   .equ USARTC1_base = 0x8B0       // Universal Asynchronous Receiver-Transmitter C1
1494   .equ SPIC_base = 0x8C0      // Serial Peripheral Interface C
1495   .equ IRCOM_base = 0x8F8     // IR Communication Module
1496   .equ TCD0_base = 0x900      // Timer/Counter D0
1497   .equ TCD2_base = 0x900      // Timer/Counter D2
1498   .equ TCD1_base = 0x940      // Timer/Counter D1
1499   .equ HIRESD_base = 0x990        // High-Resolution Extension D
1500   .equ USARTD0_base = 0x9A0       // Universal Asynchronous Receiver-Transmitter D0
1501   .equ USARTD1_base = 0x9B0       // Universal Asynchronous Receiver-Transmitter D1
1502   .equ SPID_base = 0x9C0      // Serial Peripheral Interface D
1503   .equ TCE0_base = 0xA00      // Timer/Counter E0
1504   .equ TCE2_base = 0xA00      // Timer/Counter E2
1505   .equ TCE1_base = 0xA40      // Timer/Counter E1
1506   .equ AWEXE_base = 0xA80     // Advanced Waveform Extension E
1507   .equ HIRESE_base = 0xA90        // High-Resolution Extension E
1508   .equ USARTE0_base = 0xAA0       // Universal Asynchronous Receiver-Transmitter E0
1509   .equ USARTE1_base = 0xAB0       // Universal Asynchronous Receiver-Transmitter E1
1510   .equ SPIE_base = 0xAC0      // Serial Peripheral Interface E
1511   .equ TCF0_base = 0xB00      // Timer/Counter F0
1512   .equ TCF2_base = 0xB00      // Timer/Counter F2
1513   .equ TCF1_base = 0xB40      // Timer/Counter F1
1514   .equ HIRESF_base = 0xB90        // High-Resolution Extension F
1515   .equ USARTF0_base = 0xBA0       // Universal Asynchronous Receiver-Transmitter F0
1516   .equ USARTF1_base = 0xBB0       // Universal Asynchronous Receiver-Transmitter F1
1517   .equ SPIF_base = 0xBC0      // Serial Peripheral Interface F
1518
```

```
1519
1520  ; ***** IO REGISTER OFFSETS ***************************************************
1521
1522
1523  ;**************************************************************************
1524  ;** GPIO - General Purpose IO
1525  ;**************************************************************************/
1526  .equ GPIO_GPIOR0_offset = 0x00      // General Purpose IO Register 0
1527  .equ GPIO_GPIOR1_offset = 0x01      // General Purpose IO Register 1
1528  .equ GPIO_GPIOR2_offset = 0x02      // General Purpose IO Register 2
1529  .equ GPIO_GPIOR3_offset = 0x03      // General Purpose IO Register 3
1530  .equ GPIO_GPIOR4_offset = 0x04      // General Purpose IO Register 4
1531  .equ GPIO_GPIOR5_offset = 0x05      // General Purpose IO Register 5
1532  .equ GPIO_GPIOR6_offset = 0x06      // General Purpose IO Register 6
1533  .equ GPIO_GPIOR7_offset = 0x07      // General Purpose IO Register 7
1534  .equ GPIO_GPIOR8_offset = 0x08      // General Purpose IO Register 8
1535  .equ GPIO_GPIOR9_offset = 0x09      // General Purpose IO Register 9
1536  .equ GPIO_GPIORA_offset = 0x0A      // General Purpose IO Register 10
1537  .equ GPIO_GPIORB_offset = 0x0B      // General Purpose IO Register 11
1538  .equ GPIO_GPIORC_offset = 0x0C      // General Purpose IO Register 12
1539  .equ GPIO_GPIORD_offset = 0x0D      // General Purpose IO Register 13
1540  .equ GPIO_GPIORE_offset = 0x0E      // General Purpose IO Register 14
1541  .equ GPIO_GPIORF_offset = 0x0F      // General Purpose IO Register 15
1542
1543  ;**************************************************************************
1544  ;** VPORT - Virtual Ports
1545  ;**************************************************************************/
1546  .equ VPORT_DIR_offset = 0x00        // I/O Port Data Direction
1547  .equ VPORT_OUT_offset = 0x01        // I/O Port Output
1548  .equ VPORT_IN_offset = 0x02     // I/O Port Input
1549  .equ VPORT_INTFLAGS_offset = 0x03       // Interrupt Flag Register
1550
1551  ;**************************************************************************
1552  ;** XOCD - On-Chip Debug System
1553  ;**************************************************************************/
1554  .equ OCD_OCDR0_offset = 0x00        // OCD Register 0
1555  .equ OCD_OCDR1_offset = 0x01        // OCD Register 1
1556
1557  ;**************************************************************************
1558  ;** CPU - CPU
1559  ;**************************************************************************/
1560  .equ CPU_CCP_offset = 0x04      // Configuration Change Protection
1561  .equ CPU_RAMPD_offset = 0x08        // Ramp D
1562  .equ CPU_RAMPX_offset = 0x09        // Ramp X
1563  .equ CPU_RAMPY_offset = 0x0A        // Ramp Y
1564  .equ CPU_RAMPZ_offset = 0x0B        // Ramp Z
1565  .equ CPU_EIND_offset = 0x0C     // Extended Indirect Jump
1566  .equ CPU_SPL_offset = 0x0D      // Stack Pointer Low
1567  .equ CPU_SPH_offset = 0x0E      // Stack Pointer High
1568  .equ CPU_SREG_offset = 0x0F     // Status Register
1569
1570  ;**************************************************************************
1571  ;** CLK - Clock System
1572  ;**************************************************************************/
1573  .equ CLK_CTRL_offset = 0x00     // Control Register
1574  .equ CLK_PSCTRL_offset = 0x01       // Prescaler Control Register
1575  .equ CLK_LOCK_offset = 0x02     // Lock register
1576  .equ CLK_RTCCTRL_offset = 0x03      // RTC Control Register
1577  .equ CLK_USBCTRL_offset = 0x04      // USB Control Register
1578  .equ PR_PRGEN_offset = 0x00     // General Power Reduction
1579  .equ PR_PRPA_offset = 0x01      // Power Reduction Port A
1580  .equ PR_PRPB_offset = 0x02      // Power Reduction Port B
1581  .equ PR_PRPC_offset = 0x03      // Power Reduction Port C
1582  .equ PR_PRPD_offset = 0x04      // Power Reduction Port D
1583  .equ PR_PRPE_offset = 0x05      // Power Reduction Port E
1584  .equ PR_PRPF_offset = 0x06      // Power Reduction Port F
```

```
1585
1586  ;**************************************************************************
1587  ;** SLEEP - Sleep Controller
1588  ;**************************************************************************/
1589  .equ SLEEP_CTRL_offset = 0x00       // Control Register
1590
1591  ;**************************************************************************
1592  ;** OSC - Oscillator
1593  ;**************************************************************************/
1594  .equ OSC_CTRL_offset = 0x00     // Control Register
1595  .equ OSC_STATUS_offset = 0x01       // Status Register
1596  .equ OSC_XOSCCTRL_offset = 0x02     // External Oscillator Control Register
1597  .equ OSC_XOSCFAIL_offset = 0x03     // Oscillator Failure Detection Register
1598  .equ OSC_RC32KCAL_offset = 0x04     // 32.768 kHz Internal Oscillator Calibration
     Register
1599  .equ OSC_PLLCTRL_offset = 0x05      // PLL Control Register
1600  .equ OSC_DFLLCTRL_offset = 0x06     // DFLL Control Register
1601
1602  ;**************************************************************************
1603  ;** DFLL - DFLL
1604  ;**************************************************************************/
1605  .equ DFLL_CTRL_offset = 0x00        // Control Register
1606  .equ DFLL_CALA_offset = 0x02        // Calibration Register A
1607  .equ DFLL_CALB_offset = 0x03        // Calibration Register B
1608  .equ DFLL_COMP0_offset = 0x04       // Oscillator Compare Register 0
1609  .equ DFLL_COMP1_offset = 0x05       // Oscillator Compare Register 1
1610  .equ DFLL_COMP2_offset = 0x06       // Oscillator Compare Register 2
1611
1612  ;**************************************************************************
1613  ;** RST - Reset
1614  ;**************************************************************************/
1615  .equ RST_STATUS_offset = 0x00       // Status Register
1616  .equ RST_CTRL_offset = 0x01     // Control Register
1617
1618  ;**************************************************************************
1619  ;** WDT - Watch-Dog Timer
1620  ;**************************************************************************/
1621  .equ WDT_CTRL_offset = 0x00     // Control
1622  .equ WDT_WINCTRL_offset = 0x01      // Windowed Mode Control
1623  .equ WDT_STATUS_offset = 0x02       // Status
1624
1625  ;**************************************************************************
1626  ;** MCU - MCU Control
1627  ;**************************************************************************/
1628  .equ MCU_DEVID0_offset = 0x00       // Device ID byte 0
1629  .equ MCU_DEVID1_offset = 0x01       // Device ID byte 1
1630  .equ MCU_DEVID2_offset = 0x02       // Device ID byte 2
1631  .equ MCU_REVID_offset = 0x03        // Revision ID
1632  .equ MCU_JTAGUID_offset = 0x04      // JTAG User ID
1633  .equ MCU_MCUCR_offset = 0x06        // MCU Control
1634  .equ MCU_ANAINIT_offset = 0x07      // Analog Startup Delay
1635  .equ MCU_EVSYSLOCK_offset = 0x08        // Event System Lock
1636  .equ MCU_AWEXLOCK_offset = 0x09     // AWEX Lock
1637
1638  ;**************************************************************************
1639  ;** PMIC - Programmable Multi-level Interrupt Controller
1640  ;**************************************************************************/
1641  .equ PMIC_STATUS_offset = 0x00      // Status Register
1642  .equ PMIC_INTPRI_offset = 0x01      // Interrupt Priority
1643  .equ PMIC_CTRL_offset = 0x02        // Control Register
1644
1645  ;**************************************************************************
1646  ;** PORTCFG - Port Configuration
1647  ;**************************************************************************/
1648  .equ PORTCFG_MPCMASK_offset = 0x00      // Multi-pin Configuration Mask
1649  .equ PORTCFG_VPCTRLA_offset = 0x02      // Virtual Port Control Register A
```

```
1650  .equ PORTCFG_VPCTRLB_offset = 0x03      // Virtual Port Control Register B
1651  .equ PORTCFG_CLKEVOUT_offset = 0x04     // Clock and Event Out Register
1652  .equ PORTCFG_EVOUTSEL_offset = 0x06     // Event Output Select
1653
1654  ;**************************************************************************
1655  ;** AES - AES Module
1656  ;**************************************************************************/
1657  .equ AES_CTRL_offset = 0x00     // AES Control Register
1658  .equ AES_STATUS_offset = 0x01       // AES Status Register
1659  .equ AES_STATE_offset = 0x02        // AES State Register
1660  .equ AES_KEY_offset = 0x03      // AES Key Register
1661  .equ AES_INTCTRL_offset = 0x04      // AES Interrupt Control Register
1662
1663  ;**************************************************************************
1664  ;** CRC - Cyclic Redundancy Checker
1665  ;**************************************************************************/
1666  .equ CRC_CTRL_offset = 0x00     // Control Register
1667  .equ CRC_STATUS_offset = 0x01       // Status Register
1668  .equ CRC_DATAIN_offset = 0x03       // Data Input
1669  .equ CRC_CHECKSUM0_offset = 0x04        // Checksum byte 0
1670  .equ CRC_CHECKSUM1_offset = 0x05        // Checksum byte 1
1671  .equ CRC_CHECKSUM2_offset = 0x06        // Checksum byte 2
1672  .equ CRC_CHECKSUM3_offset = 0x07        // Checksum byte 3
1673
1674  ;**************************************************************************
1675  ;** DMA - DMA Controller
1676  ;**************************************************************************/
1677  .equ DMA_CH_CTRLA_offset = 0x00     // Channel Control
1678  .equ DMA_CH_CTRLB_offset = 0x01     // Channel Control
1679  .equ DMA_CH_ADDRCTRL_offset = 0x02      // Address Control
1680  .equ DMA_CH_TRIGSRC_offset = 0x03       // Channel Trigger Source
1681  .equ DMA_CH_TRFCNT_offset = 0x04        // Channel Block Transfer Count
1682  .equ DMA_CH_REPCNT_offset = 0x06        // Channel Repeat Count
1683  .equ DMA_CH_SRCADDR0_offset = 0x08      // Channel Source Address 0
1684  .equ DMA_CH_SRCADDR1_offset = 0x09      // Channel Source Address 1
1685  .equ DMA_CH_SRCADDR2_offset = 0x0A      // Channel Source Address 2
1686  .equ DMA_CH_DESTADDR0_offset = 0x0C     // Channel Destination Address 0
1687  .equ DMA_CH_DESTADDR1_offset = 0x0D     // Channel Destination Address 1
1688  .equ DMA_CH_DESTADDR2_offset = 0x0E     // Channel Destination Address 2
1689  .equ DMA_CTRL_offset = 0x00     // Control
1690  .equ DMA_INTFLAGS_offset = 0x03     // Transfer Interrupt Status
1691  .equ DMA_STATUS_offset = 0x04       // Status
1692  .equ DMA_TEMP_offset = 0x06     // Temporary Register For 16/24-bit Access
1693  .equ DMA_CH0_offset = 0x10      // DMA Channel 0
1694  .equ DMA_CH1_offset = 0x20      // DMA Channel 1
1695  .equ DMA_CH2_offset = 0x30      // DMA Channel 2
1696  .equ DMA_CH3_offset = 0x40      // DMA Channel 3
1697
1698  ;**************************************************************************
1699  ;** EVSYS - Event System
1700  ;**************************************************************************/
1701  .equ EVSYS_CH0MUX_offset = 0x00     // Event Channel 0 Multiplexer
1702  .equ EVSYS_CH1MUX_offset = 0x01     // Event Channel 1 Multiplexer
1703  .equ EVSYS_CH2MUX_offset = 0x02     // Event Channel 2 Multiplexer
1704  .equ EVSYS_CH3MUX_offset = 0x03     // Event Channel 3 Multiplexer
1705  .equ EVSYS_CH4MUX_offset = 0x04     // Event Channel 4 Multiplexer
1706  .equ EVSYS_CH5MUX_offset = 0x05     // Event Channel 5 Multiplexer
1707  .equ EVSYS_CH6MUX_offset = 0x06     // Event Channel 6 Multiplexer
1708  .equ EVSYS_CH7MUX_offset = 0x07     // Event Channel 7 Multiplexer
1709  .equ EVSYS_CH0CTRL_offset = 0x08        // Channel 0 Control Register
1710  .equ EVSYS_CH1CTRL_offset = 0x09        // Channel 1 Control Register
1711  .equ EVSYS_CH2CTRL_offset = 0x0A        // Channel 2 Control Register
1712  .equ EVSYS_CH3CTRL_offset = 0x0B        // Channel 3 Control Register
1713  .equ EVSYS_CH4CTRL_offset = 0x0C        // Channel 4 Control Register
1714  .equ EVSYS_CH5CTRL_offset = 0x0D        // Channel 5 Control Register
1715  .equ EVSYS_CH6CTRL_offset = 0x0E        // Channel 6 Control Register
```

```
1716  .equ EVSYS_CH7CTRL_offset = 0x0F        // Channel 7 Control Register
1717  .equ EVSYS_STROBE_offset = 0x10     // Event Strobe
1718  .equ EVSYS_DATA_offset = 0x11       // Event Data
1719
1720  ;**************************************************************************
1721  ;** NVM - Non Volatile Memory Controller
1722  ;**************************************************************************/
1723  .equ NVM_ADDR0_offset = 0x00        // Address Register 0
1724  .equ NVM_ADDR1_offset = 0x01        // Address Register 1
1725  .equ NVM_ADDR2_offset = 0x02        // Address Register 2
1726  .equ NVM_DATA0_offset = 0x04        // Data Register 0
1727  .equ NVM_DATA1_offset = 0x05        // Data Register 1
1728  .equ NVM_DATA2_offset = 0x06        // Data Register 2
1729  .equ NVM_CMD_offset = 0x0A      // Command
1730  .equ NVM_CTRLA_offset = 0x0B        // Control Register A
1731  .equ NVM_CTRLB_offset = 0x0C        // Control Register B
1732  .equ NVM_INTCTRL_offset = 0x0D      // Interrupt Control
1733  .equ NVM_STATUS_offset = 0x0F       // Status
1734  .equ NVM_LOCKBITS_offset = 0x10     // Lock Bits
1735
1736  ;**************************************************************************
1737  ;** ADC - Analog/Digital Converter
1738  ;**************************************************************************/
1739  .equ ADC_CH_CTRL_offset = 0x00      // Control Register
1740  .equ ADC_CH_MUXCTRL_offset = 0x01       // MUX Control
1741  .equ ADC_CH_INTCTRL_offset = 0x02       // Channel Interrupt Control Register
1742  .equ ADC_CH_INTFLAGS_offset = 0x03      // Interrupt Flags
1743  .equ ADC_CH_RES_offset = 0x04       // Channel Result
1744  .equ ADC_CH_SCAN_offset = 0x06      // Input Channel Scan
1745  .equ ADC_CTRLA_offset = 0x00        // Control Register A
1746  .equ ADC_CTRLB_offset = 0x01        // Control Register B
1747  .equ ADC_REFCTRL_offset = 0x02      // Reference Control
1748  .equ ADC_EVCTRL_offset = 0x03       // Event Control
1749  .equ ADC_PRESCALER_offset = 0x04        // Clock Prescaler
1750  .equ ADC_INTFLAGS_offset = 0x06     // Interrupt Flags
1751  .equ ADC_TEMP_offset = 0x07     // Temporary Register
1752  .equ ADC_CAL_offset = 0x0C      // Calibration Value
1753  .equ ADC_CH0RES_offset = 0x10       // Channel 0 Result
1754  .equ ADC_CH1RES_offset = 0x12       // Channel 1 Result
1755  .equ ADC_CH2RES_offset = 0x14       // Channel 2 Result
1756  .equ ADC_CH3RES_offset = 0x16       // Channel 3 Result
1757  .equ ADC_CMP_offset = 0x18      // Compare Value
1758  .equ ADC_CH0_offset = 0x20      // ADC Channel 0
1759  .equ ADC_CH1_offset = 0x28      // ADC Channel 1
1760  .equ ADC_CH2_offset = 0x30      // ADC Channel 2
1761  .equ ADC_CH3_offset = 0x38      // ADC Channel 3
1762
1763  ;**************************************************************************
1764  ;** DAC - Digital/Analog Converter
1765  ;**************************************************************************/
1766  .equ DAC_CTRLA_offset = 0x00        // Control Register A
1767  .equ DAC_CTRLB_offset = 0x01        // Control Register B
1768  .equ DAC_CTRLC_offset = 0x02        // Control Register C
1769  .equ DAC_EVCTRL_offset = 0x03       // Event Input Control
1770  .equ DAC_TIMCTRL_offset = 0x04      // Timing Control
1771  .equ DAC_STATUS_offset = 0x05       // Status
1772  .equ DAC_CH0GAINCAL_offset = 0x08       // Gain Calibration
1773  .equ DAC_CH0OFFSETCAL_offset = 0x09     // Offset Calibration
1774  .equ DAC_CH1GAINCAL_offset = 0x0A       // Gain Calibration
1775  .equ DAC_CH1OFFSETCAL_offset = 0x0B     // Offset Calibration
1776  .equ DAC_CH0DATA_offset = 0x18      // Channel 0 Data
1777  .equ DAC_CH1DATA_offset = 0x1A      // Channel 1 Data
1778
1779  ;**************************************************************************
1780  ;** AC - Analog Comparator
1781  ;**************************************************************************/
```

```
.equ AC_AC0CTRL_offset = 0x00       // Analog Comparator 0 Control
.equ AC_AC1CTRL_offset = 0x01       // Analog Comparator 1 Control
.equ AC_AC0MUXCTRL_offset = 0x02        // Analog Comparator 0 MUX Control
.equ AC_AC1MUXCTRL_offset = 0x03        // Analog Comparator 1 MUX Control
.equ AC_CTRLA_offset = 0x04     // Control Register A
.equ AC_CTRLB_offset = 0x05     // Control Register B
.equ AC_WINCTRL_offset = 0x06       // Window Mode Control
.equ AC_STATUS_offset = 0x07        // Status

;***************************************************************************
;** RTC - Real-Time Clounter
;***************************************************************************/
.equ RTC_CTRL_offset = 0x00     // Control Register
.equ RTC_STATUS_offset = 0x01       // Status Register
.equ RTC_INTCTRL_offset = 0x02      // Interrupt Control Register
.equ RTC_INTFLAGS_offset = 0x03     // Interrupt Flags
.equ RTC_TEMP_offset = 0x04     // Temporary register
.equ RTC_CNT_offset = 0x08      // Count Register
.equ RTC_PER_offset = 0x0A      // Period Register
.equ RTC_COMP_offset = 0x0C     // Compare Register

;***************************************************************************
;** EBI - External Bus Interface
;***************************************************************************/
.equ EBI_CS_CTRLA_offset = 0x00     // Chip Select Control Register A
.equ EBI_CS_CTRLB_offset = 0x01     // Chip Select Control Register B
.equ EBI_CS_BASEADDR_offset = 0x02      // Chip Select Base Address
.equ EBI_CTRL_offset = 0x00     // Control
.equ EBI_SDRAMCTRLA_offset = 0x01       // SDRAM Control Register A
.equ EBI_REFRESH_offset = 0x04      // SDRAM Refresh Period
.equ EBI_INITDLY_offset = 0x06      // SDRAM Initialization Delay
.equ EBI_SDRAMCTRLB_offset = 0x08       // SDRAM Control Register B
.equ EBI_SDRAMCTRLC_offset = 0x09       // SDRAM Control Register C
.equ EBI_CS0_offset = 0x10      // Chip Select 0
.equ EBI_CS1_offset = 0x14      // Chip Select 1
.equ EBI_CS2_offset = 0x18      // Chip Select 2
.equ EBI_CS3_offset = 0x1C      // Chip Select 3

;***************************************************************************
;** TWI - Two-Wire Interface
;***************************************************************************/
.equ TWI_MASTER_CTRLA_offset = 0x00     // Control Register A
.equ TWI_MASTER_CTRLB_offset = 0x01     // Control Register B
.equ TWI_MASTER_CTRLC_offset = 0x02     // Control Register C
.equ TWI_MASTER_STATUS_offset = 0x03        // Status Register
.equ TWI_MASTER_BAUD_offset = 0x04      // Baurd Rate Control Register
.equ TWI_MASTER_ADDR_offset = 0x05      // Address Register
.equ TWI_MASTER_DATA_offset = 0x06      // Data Register
.equ TWI_SLAVE_CTRLA_offset = 0x00      // Control Register A
.equ TWI_SLAVE_CTRLB_offset = 0x01      // Control Register B
.equ TWI_SLAVE_STATUS_offset = 0x02     // Status Register
.equ TWI_SLAVE_ADDR_offset = 0x03       // Address Register
.equ TWI_SLAVE_DATA_offset = 0x04       // Data Register
.equ TWI_SLAVE_ADDRMASK_offset = 0x05       // Address Mask Register
.equ TWI_CTRL_offset = 0x00     // TWI Common Control Register
.equ TWI_MASTER_offset = 0x0001     // TWI master module
.equ TWI_SLAVE_offset = 0x0008      // TWI slave module

;***************************************************************************
;** USB - USB
;***************************************************************************/
.equ USB_EP_STATUS_offset = 0x00        // Endpoint Status
.equ USB_EP_CTRL_offset = 0x01      // Endpoint Control
.equ USB_EP_CNT_offset = 0x02       // USB Endpoint Counter
.equ USB_EP_DATAPTR_offset = 0x04       // Data Pointer
.equ USB_EP_AUXDATA_offset = 0x06       // Auxiliary Data
```

```
.equ USB_CTRLA_offset = 0x00        // Control Register A
.equ USB_CTRLB_offset = 0x01        // Control Register B
.equ USB_STATUS_offset = 0x02       // Status Register
.equ USB_ADDR_offset = 0x03     // Address Register
.equ USB_FIFOWP_offset = 0x04       // FIFO Write Pointer Register
.equ USB_FIFORP_offset = 0x05       // FIFO Read Pointer Register
.equ USB_EPPTR_offset = 0x06        // Endpoint Configuration Table Pointer
.equ USB_INTCTRLA_offset = 0x08     // Interrupt Control Register A
.equ USB_INTCTRLB_offset = 0x09     // Interrupt Control Register B
.equ USB_INTFLAGSACLR_offset = 0x0A     // Clear Interrupt Flag Register A
.equ USB_INTFLAGSASET_offset = 0x0B     // Set Interrupt Flag Register A
.equ USB_INTFLAGSBCLR_offset = 0x0C     // Clear Interrupt Flag Register B
.equ USB_INTFLAGSBSET_offset = 0x0D     // Set Interrupt Flag Register B
.equ USB_CAL0_offset = 0x3A     // Calibration Byte 0
.equ USB_CAL1_offset = 0x3B     // Calibration Byte 1
.equ USB_EP_TABLE_EP0OUT_offset = 0x00      // Endpoint 0
.equ USB_EP_TABLE_EP0IN_offset = 0x08       // Endpoint 0
.equ USB_EP_TABLE_EP1OUT_offset = 0x10      // Endpoint 1
.equ USB_EP_TABLE_EP1IN_offset = 0x18       // Endpoint 1
.equ USB_EP_TABLE_EP2OUT_offset = 0x20      // Endpoint 2
.equ USB_EP_TABLE_EP2IN_offset = 0x28       // Endpoint 2
.equ USB_EP_TABLE_EP3OUT_offset = 0x30      // Endpoint 3
.equ USB_EP_TABLE_EP3IN_offset = 0x38       // Endpoint 3
.equ USB_EP_TABLE_EP4OUT_offset = 0x40      // Endpoint 4
.equ USB_EP_TABLE_EP4IN_offset = 0x48       // Endpoint 4
.equ USB_EP_TABLE_EP5OUT_offset = 0x50      // Endpoint 5
.equ USB_EP_TABLE_EP5IN_offset = 0x58       // Endpoint 5
.equ USB_EP_TABLE_EP6OUT_offset = 0x60      // Endpoint 6
.equ USB_EP_TABLE_EP6IN_offset = 0x68       // Endpoint 6
.equ USB_EP_TABLE_EP7OUT_offset = 0x70      // Endpoint 7
.equ USB_EP_TABLE_EP7IN_offset = 0x78       // Endpoint 7
.equ USB_EP_TABLE_EP8OUT_offset = 0x80      // Endpoint 8
.equ USB_EP_TABLE_EP8IN_offset = 0x88       // Endpoint 8
.equ USB_EP_TABLE_EP9OUT_offset = 0x90      // Endpoint 9
.equ USB_EP_TABLE_EP9IN_offset = 0x98       // Endpoint 9
.equ USB_EP_TABLE_EP10OUT_offset = 0xA0     // Endpoint 10
.equ USB_EP_TABLE_EP10IN_offset = 0xA8      // Endpoint 10
.equ USB_EP_TABLE_EP11OUT_offset = 0xB0     // Endpoint 11
.equ USB_EP_TABLE_EP11IN_offset = 0xB8      // Endpoint 11
.equ USB_EP_TABLE_EP12OUT_offset = 0xC0     // Endpoint 12
.equ USB_EP_TABLE_EP12IN_offset = 0xC8      // Endpoint 12
.equ USB_EP_TABLE_EP13OUT_offset = 0xD0     // Endpoint 13
.equ USB_EP_TABLE_EP13IN_offset = 0xD8      // Endpoint 13
.equ USB_EP_TABLE_EP14OUT_offset = 0xE0     // Endpoint 14
.equ USB_EP_TABLE_EP14IN_offset = 0xE8      // Endpoint 14
.equ USB_EP_TABLE_EP15OUT_offset = 0xF0     // Endpoint 15
.equ USB_EP_TABLE_EP15IN_offset = 0xF8      // Endpoint 15
.equ USB_EP_TABLE_FRAMENUML_offset = 0x110      // Frame Number Low Byte
.equ USB_EP_TABLE_FRAMENUMH_offset = 0x111      // Frame Number High Byte

;--------------------------------------------------------------------------
;** PORT - I/O Port Configuration
;--------------------------------------------------------------------------*/
.equ PORT_DIR_offset = 0x00     // I/O Port Data Direction
.equ PORT_DIRSET_offset = 0x01      // I/O Port Data Direction Set
.equ PORT_DIRCLR_offset = 0x02      // I/O Port Data Direction Clear
.equ PORT_DIRTGL_offset = 0x03      // I/O Port Data Direction Toggle
.equ PORT_OUT_offset = 0x04     // I/O Port Output
.equ PORT_OUTSET_offset = 0x05      // I/O Port Output Set
.equ PORT_OUTCLR_offset = 0x06      // I/O Port Output Clear
.equ PORT_OUTTGL_offset = 0x07      // I/O Port Output Toggle
.equ PORT_IN_offset = 0x08      // I/O port Input
.equ PORT_INTCTRL_offset = 0x09     // Interrupt Control Register
.equ PORT_INT0MASK_offset = 0x0A        // Port Interrupt 0 Mask
.equ PORT_INT1MASK_offset = 0x0B        // Port Interrupt 1 Mask
.equ PORT_INTFLAGS_offset = 0x0C        // Interrupt Flag Register
```

```
1914   .equ PORT_REMAP_offset = 0x0E       // I/O Port Pin Remap Register
1915   .equ PORT_PIN0CTRL_offset = 0x10        // Pin 0 Control Register
1916   .equ PORT_PIN1CTRL_offset = 0x11        // Pin 1 Control Register
1917   .equ PORT_PIN2CTRL_offset = 0x12        // Pin 2 Control Register
1918   .equ PORT_PIN3CTRL_offset = 0x13        // Pin 3 Control Register
1919   .equ PORT_PIN4CTRL_offset = 0x14        // Pin 4 Control Register
1920   .equ PORT_PIN5CTRL_offset = 0x15        // Pin 5 Control Register
1921   .equ PORT_PIN6CTRL_offset = 0x16        // Pin 6 Control Register
1922   .equ PORT_PIN7CTRL_offset = 0x17        // Pin 7 Control Register
1923
1924   ;**************************************************************************
1925   ;** TC - 16-bit Timer/Counter With PWM
1926   ;**************************************************************************/
1927   .equ TC0_CTRLA_offset = 0x00        // Control  Register A
1928   .equ TC0_CTRLB_offset = 0x01        // Control Register B
1929   .equ TC0_CTRLC_offset = 0x02        // Control register C
1930   .equ TC0_CTRLD_offset = 0x03        // Control Register D
1931   .equ TC0_CTRLE_offset = 0x04        // Control Register E
1932   .equ TC0_INTCTRLA_offset = 0x06     // Interrupt Control Register A
1933   .equ TC0_INTCTRLB_offset = 0x07     // Interrupt Control Register B
1934   .equ TC0_CTRLFCLR_offset = 0x08     // Control Register F Clear
1935   .equ TC0_CTRLFSET_offset = 0x09     // Control Register F Set
1936   .equ TC0_CTRLGCLR_offset = 0x0A     // Control Register G Clear
1937   .equ TC0_CTRLGSET_offset = 0x0B     // Control Register G Set
1938   .equ TC0_INTFLAGS_offset = 0x0C     // Interrupt Flag Register
1939   .equ TC0_TEMP_offset = 0x0F     // Temporary Register For 16-bit Access
1940   .equ TC0_CNT_offset = 0x20      // Count
1941   .equ TC0_PER_offset = 0x26      // Period
1942   .equ TC0_CCA_offset = 0x28      // Compare or Capture A
1943   .equ TC0_CCB_offset = 0x2A      // Compare or Capture B
1944   .equ TC0_CCC_offset = 0x2C      // Compare or Capture C
1945   .equ TC0_CCD_offset = 0x2E      // Compare or Capture D
1946   .equ TC0_PERBUF_offset = 0x36       // Period Buffer
1947   .equ TC0_CCABUF_offset = 0x38       // Compare Or Capture A Buffer
1948   .equ TC0_CCBBUF_offset = 0x3A       // Compare Or Capture B Buffer
1949   .equ TC0_CCCBUF_offset = 0x3C       // Compare Or Capture C Buffer
1950   .equ TC0_CCDBUF_offset = 0x3E       // Compare Or Capture D Buffer
1951   .equ TC1_CTRLA_offset = 0x00        // Control  Register A
1952   .equ TC1_CTRLB_offset = 0x01        // Control Register B
1953   .equ TC1_CTRLC_offset = 0x02        // Control register C
1954   .equ TC1_CTRLD_offset = 0x03        // Control Register D
1955   .equ TC1_CTRLE_offset = 0x04        // Control Register E
1956   .equ TC1_INTCTRLA_offset = 0x06     // Interrupt Control Register A
1957   .equ TC1_INTCTRLB_offset = 0x07     // Interrupt Control Register B
1958   .equ TC1_CTRLFCLR_offset = 0x08     // Control Register F Clear
1959   .equ TC1_CTRLFSET_offset = 0x09     // Control Register F Set
1960   .equ TC1_CTRLGCLR_offset = 0x0A     // Control Register G Clear
1961   .equ TC1_CTRLGSET_offset = 0x0B     // Control Register G Set
1962   .equ TC1_INTFLAGS_offset = 0x0C     // Interrupt Flag Register
1963   .equ TC1_TEMP_offset = 0x0F     // Temporary Register For 16-bit Access
1964   .equ TC1_CNT_offset = 0x20      // Count
1965   .equ TC1_PER_offset = 0x26      // Period
1966   .equ TC1_CCA_offset = 0x28      // Compare or Capture A
1967   .equ TC1_CCB_offset = 0x2A      // Compare or Capture B
1968   .equ TC1_PERBUF_offset = 0x36       // Period Buffer
1969   .equ TC1_CCABUF_offset = 0x38       // Compare Or Capture A Buffer
1970   .equ TC1_CCBBUF_offset = 0x3A       // Compare Or Capture B Buffer
1971
1972   ;**************************************************************************
1973   ;** TC2 - 16-bit Timer/Counter type 2
1974   ;**************************************************************************/
1975   .equ TC2_CTRLA_offset = 0x00        // Control Register A
1976   .equ TC2_CTRLB_offset = 0x01        // Control Register B
1977   .equ TC2_CTRLC_offset = 0x02        // Control register C
1978   .equ TC2_CTRLE_offset = 0x04        // Control Register E
1979   .equ TC2_INTCTRLA_offset = 0x06     // Interrupt Control Register A
```

```
.equ TC2_INTCTRLB_offset = 0x07     // Interrupt Control Register B
.equ TC2_CTRLF_offset = 0x09        // Control Register F
.equ TC2_INTFLAGS_offset = 0x0C     // Interrupt Flag Register
.equ TC2_LCNT_offset = 0x20     // Low Byte Count
.equ TC2_HCNT_offset = 0x21     // High Byte Count
.equ TC2_LPER_offset = 0x26     // Low Byte Period
.equ TC2_HPER_offset = 0x27     // High Byte Period
.equ TC2_LCMPA_offset = 0x28        // Low Byte Compare A
.equ TC2_HCMPA_offset = 0x29        // High Byte Compare A
.equ TC2_LCMPB_offset = 0x2A        // Low Byte Compare B
.equ TC2_HCMPB_offset = 0x2B        // High Byte Compare B
.equ TC2_LCMPC_offset = 0x2C        // Low Byte Compare C
.equ TC2_HCMPC_offset = 0x2D        // High Byte Compare C
.equ TC2_LCMPD_offset = 0x2E        // Low Byte Compare D
.equ TC2_HCMPD_offset = 0x2F        // High Byte Compare D

;**************************************************************************
;** AWEX - Timer/Counter Advanced Waveform Extension
;**************************************************************************/
.equ AWEX_CTRL_offset = 0x00        // Control Register
.equ AWEX_FDEMASK_offset = 0x02     // Fault Detection Event Mask
.equ AWEX_FDCTRL_offset = 0x03      // Fault Detection Control Register
.equ AWEX_STATUS_offset = 0x04      // Status Register
.equ AWEX_STATUSSET_offset = 0x05       // Status Set Register
.equ AWEX_DTBOTH_offset = 0x06      // Dead Time Both Sides
.equ AWEX_DTBOTHBUF_offset = 0x07       // Dead Time Both Sides Buffer
.equ AWEX_DTLS_offset = 0x08        // Dead Time Low Side
.equ AWEX_DTHS_offset = 0x09        // Dead Time High Side
.equ AWEX_DTLSBUF_offset = 0x0A     // Dead Time Low Side Buffer
.equ AWEX_DTHSBUF_offset = 0x0B     // Dead Time High Side Buffer
.equ AWEX_OUTOVEN_offset = 0x0C     // Output Override Enable

;**************************************************************************
;** HIRES - Timer/Counter High-Resolution Extension
;**************************************************************************/
.equ HIRES_CTRLA_offset = 0x00      // Control Register

;**************************************************************************
;** USART - Universal Asynchronous Receiver-Transmitter
;**************************************************************************/
.equ USART_DATA_offset = 0x00       // Data Register
.equ USART_STATUS_offset = 0x01     // Status Register
.equ USART_CTRLA_offset = 0x03      // Control Register A
.equ USART_CTRLB_offset = 0x04      // Control Register B
.equ USART_CTRLC_offset = 0x05      // Control Register C
.equ USART_BAUDCTRLA_offset = 0x06      // Baud Rate Control Register A
.equ USART_BAUDCTRLB_offset = 0x07      // Baud Rate Control Register B

;**************************************************************************
;** SPI - Serial Peripheral Interface
;**************************************************************************/
.equ SPI_CTRL_offset = 0x00     // Control Register
.equ SPI_INTCTRL_offset = 0x01      // Interrupt Control Register
.equ SPI_STATUS_offset = 0x02       // Status Register
.equ SPI_DATA_offset = 0x03     // Data Register

;**************************************************************************
;** IRCOM - IR Communication Module
;**************************************************************************/
.equ IRCOM_CTRL_offset = 0x00       // Control Register
.equ IRCOM_TXPLCTRL_offset = 0x01       // IrDA Transmitter Pulse Length Control Register
.equ IRCOM_RXPLCTRL_offset = 0x02       // IrDA Receiver Pulse Length Control Register

;**************************************************************************
;** FUSE - Fuses and Lockbits
;**************************************************************************/
```

```
2046  .equ NVM_LOCKBITS_LOCKBITS_offset = 0x00        // Lock Bits
2047  .equ NVM_FUSES_FUSEBYTE0_offset = 0x00      // JTAG User ID
2048  .equ NVM_FUSES_FUSEBYTE1_offset = 0x01      // Watchdog Configuration
2049  .equ NVM_FUSES_FUSEBYTE2_offset = 0x02      // Reset Configuration
2050  .equ NVM_FUSES_FUSEBYTE4_offset = 0x04      // Start-up Configuration
2051  .equ NVM_FUSES_FUSEBYTE5_offset = 0x05      // EESAVE and BOD Level
2052
2053  ;**************************************************************************
2054  ;** SIGROW - Signature Row
2055  ;**************************************************************************/
2056  .equ NVM_PROD_SIGNATURES_RCOSC2M_offset = 0x00      // RCOSC 2 MHz Calibration Value B
2057  .equ NVM_PROD_SIGNATURES_RCOSC2MA_offset = 0x01     // RCOSC 2 MHz Calibration Value A
2058  .equ NVM_PROD_SIGNATURES_RCOSC32K_offset = 0x02     // RCOSC 32.768 kHz Calibration Value
2059  .equ NVM_PROD_SIGNATURES_RCOSC32M_offset = 0x03     // RCOSC 32 MHz Calibration Value B
2060  .equ NVM_PROD_SIGNATURES_RCOSC32MA_offset = 0x04        // RCOSC 32 MHz Calibration
      Value A
2061  .equ NVM_PROD_SIGNATURES_LOTNUM0_offset = 0x08      // Lot Number Byte 0, ASCII
2062  .equ NVM_PROD_SIGNATURES_LOTNUM1_offset = 0x09      // Lot Number Byte 1, ASCII
2063  .equ NVM_PROD_SIGNATURES_LOTNUM2_offset = 0x0A      // Lot Number Byte 2, ASCII
2064  .equ NVM_PROD_SIGNATURES_LOTNUM3_offset = 0x0B      // Lot Number Byte 3, ASCII
2065  .equ NVM_PROD_SIGNATURES_LOTNUM4_offset = 0x0C      // Lot Number Byte 4, ASCII
2066  .equ NVM_PROD_SIGNATURES_LOTNUM5_offset = 0x0D      // Lot Number Byte 5, ASCII
2067  .equ NVM_PROD_SIGNATURES_WAFNUM_offset = 0x10       // Wafer Number
2068  .equ NVM_PROD_SIGNATURES_COORDX0_offset = 0x12      // Wafer Coordinate X Byte 0
2069  .equ NVM_PROD_SIGNATURES_COORDX1_offset = 0x13      // Wafer Coordinate X Byte 1
2070  .equ NVM_PROD_SIGNATURES_COORDY0_offset = 0x14      // Wafer Coordinate Y Byte 0
2071  .equ NVM_PROD_SIGNATURES_COORDY1_offset = 0x15      // Wafer Coordinate Y Byte 1
2072  .equ NVM_PROD_SIGNATURES_USBCAL0_offset = 0x1A      // USB Calibration Byte 0
2073  .equ NVM_PROD_SIGNATURES_USBCAL1_offset = 0x1B      // USB Calibration Byte 1
2074  .equ NVM_PROD_SIGNATURES_USBRCOSC_offset = 0x1C     // USB RCOSC Calibration Value B
2075  .equ NVM_PROD_SIGNATURES_USBRCOSCA_offset = 0x1D        // USB RCOSC Calibration Value A
2076  .equ NVM_PROD_SIGNATURES_ADCACAL0_offset = 0x20     // ADCA Calibration Byte 0
2077  .equ NVM_PROD_SIGNATURES_ADCACAL1_offset = 0x21     // ADCA Calibration Byte 1
2078  .equ NVM_PROD_SIGNATURES_ADCBCAL0_offset = 0x24     // ADCB Calibration Byte 0
2079  .equ NVM_PROD_SIGNATURES_ADCBCAL1_offset = 0x25     // ADCB Calibration Byte 1
2080  .equ NVM_PROD_SIGNATURES_TEMPSENSE0_offset = 0x2E       // Temperature Sensor
      Calibration Byte 0
2081  .equ NVM_PROD_SIGNATURES_TEMPSENSE1_offset = 0x2F       // Temperature Sensor
      Calibration Byte 1
2082  .equ NVM_PROD_SIGNATURES_DACA0OFFCAL_offset = 0x30      // DACA0 Calibration Byte 0
2083  .equ NVM_PROD_SIGNATURES_DACA0GAINCAL_offset = 0x31     // DACA0 Calibration Byte 1
2084  .equ NVM_PROD_SIGNATURES_DACB0OFFCAL_offset = 0x32      // DACB0 Calibration Byte 0
2085  .equ NVM_PROD_SIGNATURES_DACB0GAINCAL_offset = 0x33     // DACB0 Calibration Byte 1
2086  .equ NVM_PROD_SIGNATURES_DACA1OFFCAL_offset = 0x34      // DACA1 Calibration Byte 0
2087  .equ NVM_PROD_SIGNATURES_DACA1GAINCAL_offset = 0x35     // DACA1 Calibration Byte 1
2088  .equ NVM_PROD_SIGNATURES_DACB1OFFCAL_offset = 0x36      // DACB1 Calibration Byte 0
2089  .equ NVM_PROD_SIGNATURES_DACB1GAINCAL_offset = 0x37     // DACB1 Calibration Byte 1
2090
2091
2092  ; ***** LOCKBIT REGISTER LOCATIONS ************************************
2093
2094
2095  ;**************************************************************************
2096  ;** LOCKBIT - Lockbits
2097  ;**************************************************************************
2098
2099  .equ LOCKBIT_LOCKBITS = 0       // Lock Bits
2100
2101
2102  ; ***** FUSE REGISTER LOCATIONS ***************************************
2103
2104
2105  ;**************************************************************************
2106  ;** FUSE - Fuses
2107  ;**************************************************************************
2108
```

```
2109  .equ FUSE_FUSEBYTE0 = 0     // JTAG User ID
2110  .equ FUSE_FUSEBYTE1 = 1     // Watchdog Configuration
2111  .equ FUSE_FUSEBYTE2 = 2     // Reset Configuration
2112  .equ FUSE_FUSEBYTE4 = 4     // Start-up Configuration
2113  .equ FUSE_FUSEBYTE5 = 5     // EESAVE and BOD Level
2114
2115
2116  ; ***** BIT AND VALUE DEFINITIONS ************************************
2117
2118
2119  ;**************************************************************************
2120  ;** GPIO - General Purpose IO
2121  ;**************************************************************************/
2122
2123
2124  ;**************************************************************************
2125  ;** VPORT - Virtual Ports
2126  ;**************************************************************************/
2127
2128  ; VPORT_INTFLAGS masks
2129  .equ VPORT_INT1IF_bm = 0x02 ; Port Interrupt 1 Flag bit mask
2130  .equ VPORT_INT1IF_bp = 1 ; Port Interrupt 1 Flag bit position
2131  .equ VPORT_INT0IF_bm = 0x01 ; Port Interrupt 0 Flag bit mask
2132  .equ VPORT_INT0IF_bp = 0 ; Port Interrupt 0 Flag bit position
2133
2134
2135  ;**************************************************************************
2136  ;** XOCD - On-Chip Debug System
2137  ;**************************************************************************/
2138
2139  ; OCD_OCDR1 masks
2140  .equ OCD_OCDRD_bm = 0x01 ; OCDR Dirty bit mask
2141  .equ OCD_OCDRD_bp = 0 ; OCDR Dirty bit position
2142
2143
2144  ;**************************************************************************
2145  ;** CPU - CPU
2146  ;**************************************************************************/
2147
2148  ; CPU_CCP masks
2149  .equ CPU_CCP_gm = 0xFF ; CCP signature group mask
2150  .equ CPU_CCP_gp = 0 ; CCP signature group position
2151  .equ CPU_CCP0_bm = (1<<0) ; CCP signature bit 0 mask
2152  .equ CPU_CCP0_bp = 0 ; CCP signature bit 0 position
2153  .equ CPU_CCP1_bm = (1<<1) ; CCP signature bit 1 mask
2154  .equ CPU_CCP1_bp = 1 ; CCP signature bit 1 position
2155  .equ CPU_CCP2_bm = (1<<2) ; CCP signature bit 2 mask
2156  .equ CPU_CCP2_bp = 2 ; CCP signature bit 2 position
2157  .equ CPU_CCP3_bm = (1<<3) ; CCP signature bit 3 mask
2158  .equ CPU_CCP3_bp = 3 ; CCP signature bit 3 position
2159  .equ CPU_CCP4_bm = (1<<4) ; CCP signature bit 4 mask
2160  .equ CPU_CCP4_bp = 4 ; CCP signature bit 4 position
2161  .equ CPU_CCP5_bm = (1<<5) ; CCP signature bit 5 mask
2162  .equ CPU_CCP5_bp = 5 ; CCP signature bit 5 position
2163  .equ CPU_CCP6_bm = (1<<6) ; CCP signature bit 6 mask
2164  .equ CPU_CCP6_bp = 6 ; CCP signature bit 6 position
2165  .equ CPU_CCP7_bm = (1<<7) ; CCP signature bit 7 mask
2166  .equ CPU_CCP7_bp = 7 ; CCP signature bit 7 position
2167
2168  ; CPU_SREG masks
2169  .equ CPU_I_bm = 0x80 ; Global Interrupt Enable Flag bit mask
2170  .equ CPU_I_bp = 7 ; Global Interrupt Enable Flag bit position
2171  .equ CPU_T_bm = 0x40 ; Transfer Bit bit mask
2172  .equ CPU_T_bp = 6 ; Transfer Bit bit position
2173  .equ CPU_H_bm = 0x20 ; Half Carry Flag bit mask
2174  .equ CPU_H_bp = 5 ; Half Carry Flag bit position
```

```
2175   .equ CPU_S_bm = 0x10 ; N Exclusive Or V Flag bit mask
2176   .equ CPU_S_bp = 4 ; N Exclusive Or V Flag bit position
2177   .equ CPU_V_bm = 0x08 ; Two's Complement Overflow Flag bit mask
2178   .equ CPU_V_bp = 3 ; Two's Complement Overflow Flag bit position
2179   .equ CPU_N_bm = 0x04 ; Negative Flag bit mask
2180   .equ CPU_N_bp = 2 ; Negative Flag bit position
2181   .equ CPU_Z_bm = 0x02 ; Zero Flag bit mask
2182   .equ CPU_Z_bp = 1 ; Zero Flag bit position
2183   .equ CPU_C_bm = 0x01 ; Carry Flag bit mask
2184   .equ CPU_C_bp = 0 ; Carry Flag bit position
2185
2186   ; CCP signatures
2187   .equ CCP_SPM_gc = (0x9D<<0) ; SPM Instruction Protection
2188   .equ CCP_IOREG_gc = (0xD8<<0) ; IO Register Protection
2189
2190
2191   ;**************************************************************************
2192   ;** CLK - Clock System
2193   ;**************************************************************************/
2194
2195   ; CLK_CTRL masks
2196   .equ CLK_SCLKSEL_gm = 0x07 ; System Clock Selection group mask
2197   .equ CLK_SCLKSEL_gp = 0 ; System Clock Selection group position
2198   .equ CLK_SCLKSEL0_bm = (1<<0) ; System Clock Selection bit 0 mask
2199   .equ CLK_SCLKSEL0_bp = 0 ; System Clock Selection bit 0 position
2200   .equ CLK_SCLKSEL1_bm = (1<<1) ; System Clock Selection bit 1 mask
2201   .equ CLK_SCLKSEL1_bp = 1 ; System Clock Selection bit 1 position
2202   .equ CLK_SCLKSEL2_bm = (1<<2) ; System Clock Selection bit 2 mask
2203   .equ CLK_SCLKSEL2_bp = 2 ; System Clock Selection bit 2 position
2204
2205   ; CLK_PSCTRL masks
2206   .equ CLK_PSADIV_gm = 0x7C ; Prescaler A Division Factor group mask
2207   .equ CLK_PSADIV_gp = 2 ; Prescaler A Division Factor group position
2208   .equ CLK_PSADIV0_bm = (1<<2) ; Prescaler A Division Factor bit 0 mask
2209   .equ CLK_PSADIV0_bp = 2 ; Prescaler A Division Factor bit 0 position
2210   .equ CLK_PSADIV1_bm = (1<<3) ; Prescaler A Division Factor bit 1 mask
2211   .equ CLK_PSADIV1_bp = 3 ; Prescaler A Division Factor bit 1 position
2212   .equ CLK_PSADIV2_bm = (1<<4) ; Prescaler A Division Factor bit 2 mask
2213   .equ CLK_PSADIV2_bp = 4 ; Prescaler A Division Factor bit 2 position
2214   .equ CLK_PSADIV3_bm = (1<<5) ; Prescaler A Division Factor bit 3 mask
2215   .equ CLK_PSADIV3_bp = 5 ; Prescaler A Division Factor bit 3 position
2216   .equ CLK_PSADIV4_bm = (1<<6) ; Prescaler A Division Factor bit 4 mask
2217   .equ CLK_PSADIV4_bp = 6 ; Prescaler A Division Factor bit 4 position
2218   .equ CLK_PSBCDIV_gm = 0x03 ; Prescaler B and C Division factor group mask
2219   .equ CLK_PSBCDIV_gp = 0 ; Prescaler B and C Division factor group position
2220   .equ CLK_PSBCDIV0_bm = (1<<0) ; Prescaler B and C Division factor bit 0 mask
2221   .equ CLK_PSBCDIV0_bp = 0 ; Prescaler B and C Division factor bit 0 position
2222   .equ CLK_PSBCDIV1_bm = (1<<1) ; Prescaler B and C Division factor bit 1 mask
2223   .equ CLK_PSBCDIV1_bp = 1 ; Prescaler B and C Division factor bit 1 position
2224
2225   ; CLK_LOCK masks
2226   .equ CLK_LOCK_bm = 0x01 ; Clock System Lock bit mask
2227   .equ CLK_LOCK_bp = 0 ; Clock System Lock bit position
2228
2229   ; CLK_RTCCTRL masks
2230   .equ CLK_RTCSRC_gm = 0x0E ; Clock Source group mask
2231   .equ CLK_RTCSRC_gp = 1 ; Clock Source group position
2232   .equ CLK_RTCSRC0_bm = (1<<1) ; Clock Source bit 0 mask
2233   .equ CLK_RTCSRC0_bp = 1 ; Clock Source bit 0 position
2234   .equ CLK_RTCSRC1_bm = (1<<2) ; Clock Source bit 1 mask
2235   .equ CLK_RTCSRC1_bp = 2 ; Clock Source bit 1 position
2236   .equ CLK_RTCSRC2_bm = (1<<3) ; Clock Source bit 2 mask
2237   .equ CLK_RTCSRC2_bp = 3 ; Clock Source bit 2 position
2238   .equ CLK_RTCEN_bm = 0x01 ; Clock Source Enable bit mask
2239   .equ CLK_RTCEN_bp = 0 ; Clock Source Enable bit position
2240
```

```
; CLK_USBCTRL masks
.equ CLK_USBPSDIV_gm = 0x38 ; Prescaler Division Factor group mask
.equ CLK_USBPSDIV_gp = 3 ; Prescaler Division Factor group position
.equ CLK_USBPSDIV0_bm = (1<<3) ; Prescaler Division Factor bit 0 mask
.equ CLK_USBPSDIV0_bp = 3 ; Prescaler Division Factor bit 0 position
.equ CLK_USBPSDIV1_bm = (1<<4) ; Prescaler Division Factor bit 1 mask
.equ CLK_USBPSDIV1_bp = 4 ; Prescaler Division Factor bit 1 position
.equ CLK_USBPSDIV2_bm = (1<<5) ; Prescaler Division Factor bit 2 mask
.equ CLK_USBPSDIV2_bp = 5 ; Prescaler Division Factor bit 2 position
.equ CLK_USBSRC_gm = 0x06 ; Clock Source group mask
.equ CLK_USBSRC_gp = 1 ; Clock Source group position
.equ CLK_USBSRC0_bm = (1<<1) ; Clock Source bit 0 mask
.equ CLK_USBSRC0_bp = 1 ; Clock Source bit 0 position
.equ CLK_USBSRC1_bm = (1<<2) ; Clock Source bit 1 mask
.equ CLK_USBSRC1_bp = 2 ; Clock Source bit 1 position
.equ CLK_USBSEN_bm = 0x01 ; Clock Source Enable bit mask
.equ CLK_USBSEN_bp = 0 ; Clock Source Enable bit position

; PR_PRGEN masks
.equ PR_USB_bm = 0x40 ; USB bit mask
.equ PR_USB_bp = 6 ; USB bit position
.equ PR_AES_bm = 0x10 ; AES bit mask
.equ PR_AES_bp = 4 ; AES bit position
.equ PR_EBI_bm = 0x08 ; External Bus Interface bit mask
.equ PR_EBI_bp = 3 ; External Bus Interface bit position
.equ PR_RTC_bm = 0x04 ; Real-time Counter bit mask
.equ PR_RTC_bp = 2 ; Real-time Counter bit position
.equ PR_EVSYS_bm = 0x02 ; Event System bit mask
.equ PR_EVSYS_bp = 1 ; Event System bit position
.equ PR_DMA_bm = 0x01 ; DMA-Controller bit mask
.equ PR_DMA_bp = 0 ; DMA-Controller bit position

; PR_PRPA masks
.equ PR_DAC_bm = 0x04 ; Port A DAC bit mask
.equ PR_DAC_bp = 2 ; Port A DAC bit position
.equ PR_ADC_bm = 0x02 ; Port A ADC bit mask
.equ PR_ADC_bp = 1 ; Port A ADC bit position
.equ PR_AC_bm = 0x01 ; Port A Analog Comparator bit mask
.equ PR_AC_bp = 0 ; Port A Analog Comparator bit position

; PR_PRPB masks
; Masks for DAC aready defined
; Masks for ADC aready defined
; Masks for AC aready defined

; PR_PRPC masks
.equ PR_TWI_bm = 0x40 ; Port C Two-wire Interface bit mask
.equ PR_TWI_bp = 6 ; Port C Two-wire Interface bit position
.equ PR_USART1_bm = 0x20 ; Port C USART1 bit mask
.equ PR_USART1_bp = 5 ; Port C USART1 bit position
.equ PR_USART0_bm = 0x10 ; Port C USART0 bit mask
.equ PR_USART0_bp = 4 ; Port C USART0 bit position
.equ PR_SPI_bm = 0x08 ; Port C SPI bit mask
.equ PR_SPI_bp = 3 ; Port C SPI bit position
.equ PR_HIRES_bm = 0x04 ; Port C AWEX bit mask
.equ PR_HIRES_bp = 2 ; Port C AWEX bit position
.equ PR_TC1_bm = 0x02 ; Port C Timer/Counter1 bit mask
.equ PR_TC1_bp = 1 ; Port C Timer/Counter1 bit position
.equ PR_TC0_bm = 0x01 ; Port C Timer/Counter0 bit mask
.equ PR_TC0_bp = 0 ; Port C Timer/Counter0 bit position

; PR_PRPD masks
; Masks for TWI aready defined
; Masks for USART1 aready defined
; Masks for USART0 aready defined
; Masks for SPI aready defined
```

```
2307   ; Masks for HIRES aready defined
2308   ; Masks for TC1 aready defined
2309   ; Masks for TC0 aready defined
2310
2311   ; PR_PRPE masks
2312   ; Masks for TWI aready defined
2313   ; Masks for USART1 aready defined
2314   ; Masks for USART0 aready defined
2315   ; Masks for SPI aready defined
2316   ; Masks for HIRES aready defined
2317   ; Masks for TC1 aready defined
2318   ; Masks for TC0 aready defined
2319
2320   ; PR_PRPF masks
2321   ; Masks for TWI aready defined
2322   ; Masks for USART1 aready defined
2323   ; Masks for USART0 aready defined
2324   ; Masks for SPI aready defined
2325   ; Masks for HIRES aready defined
2326   ; Masks for TC1 aready defined
2327   ; Masks for TC0 aready defined
2328
2329   ; System Clock Selection
2330   .equ CLK_SCLKSEL_RC2M_gc = (0x00<<0) ; Internal 2 MHz RC Oscillator
2331   .equ CLK_SCLKSEL_RC32M_gc = (0x01<<0) ; Internal 32 MHz RC Oscillator
2332   .equ CLK_SCLKSEL_RC32K_gc = (0x02<<0) ; Internal 32.768 kHz RC Oscillator
2333   .equ CLK_SCLKSEL_XOSC_gc = (0x03<<0) ; External Crystal Oscillator or Clock
2334   .equ CLK_SCLKSEL_PLL_gc = (0x04<<0) ; Phase Locked Loop
2335
2336   ; Prescaler A Division Factor
2337   .equ CLK_PSADIV_1_gc = (0x00<<2) ; Divide by 1
2338   .equ CLK_PSADIV_2_gc = (0x01<<2) ; Divide by 2
2339   .equ CLK_PSADIV_4_gc = (0x03<<2) ; Divide by 4
2340   .equ CLK_PSADIV_8_gc = (0x05<<2) ; Divide by 8
2341   .equ CLK_PSADIV_16_gc = (0x07<<2) ; Divide by 16
2342   .equ CLK_PSADIV_32_gc = (0x09<<2) ; Divide by 32
2343   .equ CLK_PSADIV_64_gc = (0x0B<<2) ; Divide by 64
2344   .equ CLK_PSADIV_128_gc = (0x0D<<2) ; Divide by 128
2345   .equ CLK_PSADIV_256_gc = (0x0F<<2) ; Divide by 256
2346   .equ CLK_PSADIV_512_gc = (0x11<<2) ; Divide by 512
2347
2348   ; Prescaler B and C Division Factor
2349   .equ CLK_PSBCDIV_1_1_gc = (0x00<<0) ; Divide B by 1 and C by 1
2350   .equ CLK_PSBCDIV_1_2_gc = (0x01<<0) ; Divide B by 1 and C by 2
2351   .equ CLK_PSBCDIV_4_1_gc = (0x02<<0) ; Divide B by 4 and C by 1
2352   .equ CLK_PSBCDIV_2_2_gc = (0x03<<0) ; Divide B by 2 and C by 2
2353
2354   ; RTC Clock Source
2355   .equ CLK_RTCSRC_ULP_gc = (0x00<<1) ; 1.024 kHz from internal 32kHz ULP
2356   .equ CLK_RTCSRC_TOSC_gc = (0x01<<1) ; 1.024 kHz from 32.768 kHz crystal oscillator on
       TOSC
2357   .equ CLK_RTCSRC_RCOSC_gc = (0x02<<1) ; 1.024 kHz from internal 32.768 kHz RC oscillator
2358   .equ CLK_RTCSRC_TOSC32_gc = (0x05<<1) ; 32.768 kHz from 32.768 kHz crystal oscillator
       on TOSC
2359   .equ CLK_RTCSRC_RCOSC32_gc = (0x06<<1) ; 32.768 kHz from internal 32.768 kHz RC
       oscillator
2360   .equ CLK_RTCSRC_EXTCLK_gc = (0x07<<1) ; External Clock from TOSC1
2361
2362   ; USB Prescaler Division Factor
2363   .equ CLK_USBPSDIV_1_gc = (0x00<<3) ; Divide by 1
2364   .equ CLK_USBPSDIV_2_gc = (0x01<<3) ; Divide by 2
2365   .equ CLK_USBPSDIV_4_gc = (0x02<<3) ; Divide by 4
2366   .equ CLK_USBPSDIV_8_gc = (0x03<<3) ; Divide by 8
2367   .equ CLK_USBPSDIV_16_gc = (0x04<<3) ; Divide by 16
2368   .equ CLK_USBPSDIV_32_gc = (0x05<<3) ; Divide by 32
2369
```

```
; USB Clock Source
.equ CLK_USBSRC_PLL_gc = (0x00<<1) ; PLL
.equ CLK_USBSRC_RC32M_gc = (0x01<<1) ; Internal 32 MHz RC Oscillator


;**************************************************************************
;** SLEEP - Sleep Controller
;**************************************************************************/

; SLEEP_CTRL masks
.equ SLEEP_SMODE_gm = 0x0E ; Sleep Mode group mask
.equ SLEEP_SMODE_gp = 1 ; Sleep Mode group position
.equ SLEEP_SMODE0_bm = (1<<1) ; Sleep Mode bit 0 mask
.equ SLEEP_SMODE0_bp = 1 ; Sleep Mode bit 0 position
.equ SLEEP_SMODE1_bm = (1<<2) ; Sleep Mode bit 1 mask
.equ SLEEP_SMODE1_bp = 2 ; Sleep Mode bit 1 position
.equ SLEEP_SMODE2_bm = (1<<3) ; Sleep Mode bit 2 mask
.equ SLEEP_SMODE2_bp = 3 ; Sleep Mode bit 2 position
.equ SLEEP_SEN_bm = 0x01 ; Sleep Enable bit mask
.equ SLEEP_SEN_bp = 0 ; Sleep Enable bit position

; Sleep Mode
.equ SLEEP_SMODE_IDLE_gc = (0x00<<1) ; Idle mode
.equ SLEEP_SMODE_PDOWN_gc = (0x02<<1) ; Power-down Mode
.equ SLEEP_SMODE_PSAVE_gc = (0x03<<1) ; Power-save Mode
.equ SLEEP_SMODE_STDBY_gc = (0x06<<1) ; Standby Mode
.equ SLEEP_SMODE_ESTDBY_gc = (0x07<<1) ; Extended Standby Mode


;**************************************************************************
;** OSC - Oscillator
;**************************************************************************/

; OSC_CTRL masks
.equ OSC_PLLEN_bm = 0x10 ; PLL Enable bit mask
.equ OSC_PLLEN_bp = 4 ; PLL Enable bit position
.equ OSC_XOSCEN_bm = 0x08 ; External Oscillator Enable bit mask
.equ OSC_XOSCEN_bp = 3 ; External Oscillator Enable bit position
.equ OSC_RC32KEN_bm = 0x04 ; Internal 32.768 kHz RC Oscillator Enable bit mask
.equ OSC_RC32KEN_bp = 2 ; Internal 32.768 kHz RC Oscillator Enable bit position
.equ OSC_RC32MEN_bm = 0x02 ; Internal 32 MHz RC Oscillator Enable bit mask
.equ OSC_RC32MEN_bp = 1 ; Internal 32 MHz RC Oscillator Enable bit position
.equ OSC_RC2MEN_bm = 0x01 ; Internal 2 MHz RC Oscillator Enable bit mask
.equ OSC_RC2MEN_bp = 0 ; Internal 2 MHz RC Oscillator Enable bit position

; OSC_STATUS masks
.equ OSC_PLLRDY_bm = 0x10 ; PLL Ready bit mask
.equ OSC_PLLRDY_bp = 4 ; PLL Ready bit position
.equ OSC_XOSCRDY_bm = 0x08 ; External Oscillator Ready bit mask
.equ OSC_XOSCRDY_bp = 3 ; External Oscillator Ready bit position
.equ OSC_RC32KRDY_bm = 0x04 ; Internal 32.768 kHz RC Oscillator Ready bit mask
.equ OSC_RC32KRDY_bp = 2 ; Internal 32.768 kHz RC Oscillator Ready bit position
.equ OSC_RC32MRDY_bm = 0x02 ; Internal 32 MHz RC Oscillator Ready bit mask
.equ OSC_RC32MRDY_bp = 1 ; Internal 32 MHz RC Oscillator Ready bit position
.equ OSC_RC2MRDY_bm = 0x01 ; Internal 2 MHz RC Oscillator Ready bit mask
.equ OSC_RC2MRDY_bp = 0 ; Internal 2 MHz RC Oscillator Ready bit position

; OSC_XOSCCTRL masks
.equ OSC_FRQRANGE_gm = 0xC0 ; Frequency Range group mask
.equ OSC_FRQRANGE_gp = 6 ; Frequency Range group position
.equ OSC_FRQRANGE0_bm = (1<<6) ; Frequency Range bit 0 mask
.equ OSC_FRQRANGE0_bp = 6 ; Frequency Range bit 0 position
.equ OSC_FRQRANGE1_bm = (1<<7) ; Frequency Range bit 1 mask
.equ OSC_FRQRANGE1_bp = 7 ; Frequency Range bit 1 position
.equ OSC_X32KLPM_bm = 0x20 ; 32.768 kHz XTAL OSC Low-power Mode bit mask
.equ OSC_X32KLPM_bp = 5 ; 32.768 kHz XTAL OSC Low-power Mode bit position
```

```
.equ OSC_XOSCPWR_bm = 0x10 ; 16 MHz Crystal Oscillator High Power mode bit mask
.equ OSC_XOSCPWR_bp = 4 ; 16 MHz Crystal Oscillator High Power mode bit position
.equ OSC_XOSCSEL_gm = 0x0F ; External Oscillator Selection and Startup Time group mask
.equ OSC_XOSCSEL_gp = 0 ; External Oscillator Selection and Startup Time group position
.equ OSC_XOSCSEL0_bm = (1<<0) ; External Oscillator Selection and Startup Time bit 0 mask
.equ OSC_XOSCSEL0_bp = 0 ; External Oscillator Selection and Startup Time bit 0 position
.equ OSC_XOSCSEL1_bm = (1<<1) ; External Oscillator Selection and Startup Time bit 1 mask
.equ OSC_XOSCSEL1_bp = 1 ; External Oscillator Selection and Startup Time bit 1 position
.equ OSC_XOSCSEL2_bm = (1<<2) ; External Oscillator Selection and Startup Time bit 2 mask
.equ OSC_XOSCSEL2_bp = 2 ; External Oscillator Selection and Startup Time bit 2 position
.equ OSC_XOSCSEL3_bm = (1<<3) ; External Oscillator Selection and Startup Time bit 3 mask
.equ OSC_XOSCSEL3_bp = 3 ; External Oscillator Selection and Startup Time bit 3 position

; OSC_XOSCFAIL masks
.equ OSC_PLLFDIF_bm = 0x08 ; PLL Failure Detection Interrupt Flag bit mask
.equ OSC_PLLFDIF_bp = 3 ; PLL Failure Detection Interrupt Flag bit position
.equ OSC_PLLFDEN_bm = 0x04 ; PLL Failure Detection Enable bit mask
.equ OSC_PLLFDEN_bp = 2 ; PLL Failure Detection Enable bit position
.equ OSC_XOSCFDIF_bm = 0x02 ; XOSC Failure Detection Interrupt Flag bit mask
.equ OSC_XOSCFDIF_bp = 1 ; XOSC Failure Detection Interrupt Flag bit position
.equ OSC_XOSCFDEN_bm = 0x01 ; XOSC Failure Detection Enable bit mask
.equ OSC_XOSCFDEN_bp = 0 ; XOSC Failure Detection Enable bit position

; OSC_PLLCTRL masks
.equ OSC_PLLSRC_gm = 0xC0 ; Clock Source group mask
.equ OSC_PLLSRC_gp = 6 ; Clock Source group position
.equ OSC_PLLSRC0_bm = (1<<6) ; Clock Source bit 0 mask
.equ OSC_PLLSRC0_bp = 6 ; Clock Source bit 0 position
.equ OSC_PLLSRC1_bm = (1<<7) ; Clock Source bit 1 mask
.equ OSC_PLLSRC1_bp = 7 ; Clock Source bit 1 position
.equ OSC_PLLDIV_bm = 0x20 ; Divide by 2 bit mask
.equ OSC_PLLDIV_bp = 5 ; Divide by 2 bit position
.equ OSC_PLLFAC_gm = 0x1F ; Multiplication Factor group mask
.equ OSC_PLLFAC_gp = 0 ; Multiplication Factor group position
.equ OSC_PLLFAC0_bm = (1<<0) ; Multiplication Factor bit 0 mask
.equ OSC_PLLFAC0_bp = 0 ; Multiplication Factor bit 0 position
.equ OSC_PLLFAC1_bm = (1<<1) ; Multiplication Factor bit 1 mask
.equ OSC_PLLFAC1_bp = 1 ; Multiplication Factor bit 1 position
.equ OSC_PLLFAC2_bm = (1<<2) ; Multiplication Factor bit 2 mask
.equ OSC_PLLFAC2_bp = 2 ; Multiplication Factor bit 2 position
.equ OSC_PLLFAC3_bm = (1<<3) ; Multiplication Factor bit 3 mask
.equ OSC_PLLFAC3_bp = 3 ; Multiplication Factor bit 3 position
.equ OSC_PLLFAC4_bm = (1<<4) ; Multiplication Factor bit 4 mask
.equ OSC_PLLFAC4_bp = 4 ; Multiplication Factor bit 4 position

; OSC_DFLLCTRL masks
.equ OSC_RC32MCREF_gm = 0x06 ; 32 MHz DFLL Calibration Reference group mask
.equ OSC_RC32MCREF_gp = 1 ; 32 MHz DFLL Calibration Reference group position
.equ OSC_RC32MCREF0_bm = (1<<1) ; 32 MHz DFLL Calibration Reference bit 0 mask
.equ OSC_RC32MCREF0_bp = 1 ; 32 MHz DFLL Calibration Reference bit 0 position
.equ OSC_RC32MCREF1_bm = (1<<2) ; 32 MHz DFLL Calibration Reference bit 1 mask
.equ OSC_RC32MCREF1_bp = 2 ; 32 MHz DFLL Calibration Reference bit 1 position
.equ OSC_RC2MCREF_bm = 0x01 ; 2 MHz DFLL Calibration Reference bit mask
.equ OSC_RC2MCREF_bp = 0 ; 2 MHz DFLL Calibration Reference bit position

; Oscillator Frequency Range
.equ OSC_FRQRANGE_04TO2_gc = (0x00<<6) ; 0.4 - 2 MHz
.equ OSC_FRQRANGE_2TO9_gc = (0x01<<6) ; 2 - 9 MHz
.equ OSC_FRQRANGE_9TO12_gc = (0x02<<6) ; 9 - 12 MHz
.equ OSC_FRQRANGE_12TO16_gc = (0x03<<6) ; 12 - 16 MHz

; External Oscillator Selection and Startup Time
.equ OSC_XOSCSEL_EXTCLK_gc = (0x00<<0) ; External Clock - 6 CLK
.equ OSC_XOSCSEL_32KHz_gc = (0x02<<0) ; 32.768 kHz TOSC - 32K CLK
.equ OSC_XOSCSEL_XTAL_256CLK_gc = (0x03<<0) ; 0.4-16 MHz XTAL - 256 CLK
.equ OSC_XOSCSEL_XTAL_1KCLK_gc = (0x07<<0) ; 0.4-16 MHz XTAL - 1K CLK
```

```
.equ OSC_XOSCSEL_XTAL_16KCLK_gc = (0x0B<<0) ; 0.4-16 MHz XTAL - 16K CLK

; PLL Clock Source
.equ OSC_PLLSRC_RC2M_gc = (0x00<<6) ; Internal 2 MHz RC Oscillator
.equ OSC_PLLSRC_RC32M_gc = (0x02<<6) ; Internal 32 MHz RC Oscillator
.equ OSC_PLLSRC_XOSC_gc = (0x03<<6) ; External Oscillator

; 2 MHz DFLL Calibration Reference
.equ OSC_RC2MCREF_RC32K_gc = (0x00<<0) ; Internal 32.768 kHz RC Oscillator
.equ OSC_RC2MCREF_XOSC32K_gc = (0x01<<0) ; External 32.768 kHz Crystal Oscillator

; 32 MHz DFLL Calibration Reference
.equ OSC_RC32MCREF_RC32K_gc = (0x00<<1) ; Internal 32.768 kHz RC Oscillator
.equ OSC_RC32MCREF_XOSC32K_gc = (0x01<<1) ; External 32.768 kHz Crystal Oscillator
.equ OSC_RC32MCREF_USBSOF_gc = (0x02<<1) ; USB Start of Frame


;**************************************************************************
;** DFLL - DFLL
;**************************************************************************/

; DFLL_CTRL masks
.equ DFLL_ENABLE_bm = 0x01 ; DFLL Enable bit mask
.equ DFLL_ENABLE_bp = 0 ; DFLL Enable bit position

; DFLL_CALA masks
.equ DFLL_CALL_gm = 0x7F ; DFLL Calibration Value A group mask
.equ DFLL_CALL_gp = 0 ; DFLL Calibration Value A group position
.equ DFLL_CALL0_bm = (1<<0) ; DFLL Calibration Value A bit 0 mask
.equ DFLL_CALL0_bp = 0 ; DFLL Calibration Value A bit 0 position
.equ DFLL_CALL1_bm = (1<<1) ; DFLL Calibration Value A bit 1 mask
.equ DFLL_CALL1_bp = 1 ; DFLL Calibration Value A bit 1 position
.equ DFLL_CALL2_bm = (1<<2) ; DFLL Calibration Value A bit 2 mask
.equ DFLL_CALL2_bp = 2 ; DFLL Calibration Value A bit 2 position
.equ DFLL_CALL3_bm = (1<<3) ; DFLL Calibration Value A bit 3 mask
.equ DFLL_CALL3_bp = 3 ; DFLL Calibration Value A bit 3 position
.equ DFLL_CALL4_bm = (1<<4) ; DFLL Calibration Value A bit 4 mask
.equ DFLL_CALL4_bp = 4 ; DFLL Calibration Value A bit 4 position
.equ DFLL_CALL5_bm = (1<<5) ; DFLL Calibration Value A bit 5 mask
.equ DFLL_CALL5_bp = 5 ; DFLL Calibration Value A bit 5 position
.equ DFLL_CALL6_bm = (1<<6) ; DFLL Calibration Value A bit 6 mask
.equ DFLL_CALL6_bp = 6 ; DFLL Calibration Value A bit 6 position

; DFLL_CALB masks
.equ DFLL_CALH_gm = 0x3F ; DFLL Calibration Value B group mask
.equ DFLL_CALH_gp = 0 ; DFLL Calibration Value B group position
.equ DFLL_CALH0_bm = (1<<0) ; DFLL Calibration Value B bit 0 mask
.equ DFLL_CALH0_bp = 0 ; DFLL Calibration Value B bit 0 position
.equ DFLL_CALH1_bm = (1<<1) ; DFLL Calibration Value B bit 1 mask
.equ DFLL_CALH1_bp = 1 ; DFLL Calibration Value B bit 1 position
.equ DFLL_CALH2_bm = (1<<2) ; DFLL Calibration Value B bit 2 mask
.equ DFLL_CALH2_bp = 2 ; DFLL Calibration Value B bit 2 position
.equ DFLL_CALH3_bm = (1<<3) ; DFLL Calibration Value B bit 3 mask
.equ DFLL_CALH3_bp = 3 ; DFLL Calibration Value B bit 3 position
.equ DFLL_CALH4_bm = (1<<4) ; DFLL Calibration Value B bit 4 mask
.equ DFLL_CALH4_bp = 4 ; DFLL Calibration Value B bit 4 position
.equ DFLL_CALH5_bm = (1<<5) ; DFLL Calibration Value B bit 5 mask
.equ DFLL_CALH5_bp = 5 ; DFLL Calibration Value B bit 5 position


;**************************************************************************
;** RST - Reset
;**************************************************************************/

; RST_STATUS masks
.equ RST_SDRF_bm = 0x40 ; Spike Detection Reset Flag bit mask
```

```
.equ RST_SDRF_bp = 6 ; Spike Detection Reset Flag bit position
.equ RST_SRF_bm = 0x20 ; Software Reset Flag bit mask
.equ RST_SRF_bp = 5 ; Software Reset Flag bit position
.equ RST_PDIRF_bm = 0x10 ; Programming and Debug Interface Interface Reset Flag bit mask
.equ RST_PDIRF_bp = 4 ; Programming and Debug Interface Interface Reset Flag bit position
.equ RST_WDRF_bm = 0x08 ; Watchdog Reset Flag bit mask
.equ RST_WDRF_bp = 3 ; Watchdog Reset Flag bit position
.equ RST_BORF_bm = 0x04 ; Brown-out Reset Flag bit mask
.equ RST_BORF_bp = 2 ; Brown-out Reset Flag bit position
.equ RST_EXTRF_bm = 0x02 ; External Reset Flag bit mask
.equ RST_EXTRF_bp = 1 ; External Reset Flag bit position
.equ RST_PORF_bm = 0x01 ; Power-on Reset Flag bit mask
.equ RST_PORF_bp = 0 ; Power-on Reset Flag bit position

; RST_CTRL masks
.equ RST_SWRST_bm = 0x01 ; Software Reset bit mask
.equ RST_SWRST_bp = 0 ; Software Reset bit position


;**************************************************************************
;** WDT - Watch-Dog Timer
;**************************************************************************/

; WDT_CTRL masks
.equ WDT_PER_gm = 0x3C ; Period group mask
.equ WDT_PER_gp = 2 ; Period group position
.equ WDT_PER0_bm = (1<<2) ; Period bit 0 mask
.equ WDT_PER0_bp = 2 ; Period bit 0 position
.equ WDT_PER1_bm = (1<<3) ; Period bit 1 mask
.equ WDT_PER1_bp = 3 ; Period bit 1 position
.equ WDT_PER2_bm = (1<<4) ; Period bit 2 mask
.equ WDT_PER2_bp = 4 ; Period bit 2 position
.equ WDT_PER3_bm = (1<<5) ; Period bit 3 mask
.equ WDT_PER3_bp = 5 ; Period bit 3 position
.equ WDT_ENABLE_bm = 0x02 ; Enable bit mask
.equ WDT_ENABLE_bp = 1 ; Enable bit position
.equ WDT_CEN_bm = 0x01 ; Change Enable bit mask
.equ WDT_CEN_bp = 0 ; Change Enable bit position

; WDT_WINCTRL masks
.equ WDT_WPER_gm = 0x3C ; Windowed Mode Period group mask
.equ WDT_WPER_gp = 2 ; Windowed Mode Period group position
.equ WDT_WPER0_bm = (1<<2) ; Windowed Mode Period bit 0 mask
.equ WDT_WPER0_bp = 2 ; Windowed Mode Period bit 0 position
.equ WDT_WPER1_bm = (1<<3) ; Windowed Mode Period bit 1 mask
.equ WDT_WPER1_bp = 3 ; Windowed Mode Period bit 1 position
.equ WDT_WPER2_bm = (1<<4) ; Windowed Mode Period bit 2 mask
.equ WDT_WPER2_bp = 4 ; Windowed Mode Period bit 2 position
.equ WDT_WPER3_bm = (1<<5) ; Windowed Mode Period bit 3 mask
.equ WDT_WPER3_bp = 5 ; Windowed Mode Period bit 3 position
.equ WDT_WEN_bm = 0x02 ; Windowed Mode Enable bit mask
.equ WDT_WEN_bp = 1 ; Windowed Mode Enable bit position
.equ WDT_WCEN_bm = 0x01 ; Windowed Mode Change Enable bit mask
.equ WDT_WCEN_bp = 0 ; Windowed Mode Change Enable bit position

; WDT_STATUS masks
.equ WDT_SYNCBUSY_bm = 0x01 ; Syncronization busy bit mask
.equ WDT_SYNCBUSY_bp = 0 ; Syncronization busy bit position

; Period setting
.equ WDT_PER_8CLK_gc = (0x00<<2) ; 8 cycles (8ms @ 3.3V)
.equ WDT_PER_16CLK_gc = (0x01<<2) ; 16 cycles (16ms @ 3.3V)
.equ WDT_PER_32CLK_gc = (0x02<<2) ; 32 cycles (32ms @ 3.3V)
.equ WDT_PER_64CLK_gc = (0x03<<2) ; 64 cycles (64ms @ 3.3V)
.equ WDT_PER_125CLK_gc = (0x04<<2) ; 125 cycles (0.125s @ 3.3V)
.equ WDT_PER_250CLK_gc = (0x05<<2) ; 250 cycles (0.25s @ 3.3V)
```

```
.equ WDT_PER_500CLK_gc = (0x06<<2) ; 500 cycles (0.5s @ 3.3V)
.equ WDT_PER_1KCLK_gc = (0x07<<2) ; 1K cycles (1s @ 3.3V)
.equ WDT_PER_2KCLK_gc = (0x08<<2) ; 2K cycles (2s @ 3.3V)
.equ WDT_PER_4KCLK_gc = (0x09<<2) ; 4K cycles (4s @ 3.3V)
.equ WDT_PER_8KCLK_gc = (0x0A<<2) ; 8K cycles (8s @ 3.3V)

; Closed window period
.equ WDT_WPER_8CLK_gc = (0x00<<2) ; 8 cycles (8ms @ 3.3V)
.equ WDT_WPER_16CLK_gc = (0x01<<2) ; 16 cycles (16ms @ 3.3V)
.equ WDT_WPER_32CLK_gc = (0x02<<2) ; 32 cycles (32ms @ 3.3V)
.equ WDT_WPER_64CLK_gc = (0x03<<2) ; 64 cycles (64ms @ 3.3V)
.equ WDT_WPER_125CLK_gc = (0x04<<2) ; 125 cycles (0.125s @ 3.3V)
.equ WDT_WPER_250CLK_gc = (0x05<<2) ; 250 cycles (0.25s @ 3.3V)
.equ WDT_WPER_500CLK_gc = (0x06<<2) ; 500 cycles (0.5s @ 3.3V)
.equ WDT_WPER_1KCLK_gc = (0x07<<2) ; 1K cycles (1s @ 3.3V)
.equ WDT_WPER_2KCLK_gc = (0x08<<2) ; 2K cycles (2s @ 3.3V)
.equ WDT_WPER_4KCLK_gc = (0x09<<2) ; 4K cycles (4s @ 3.3V)
.equ WDT_WPER_8KCLK_gc = (0x0A<<2) ; 8K cycles (8s @ 3.3V)


;**************************************************************************
;** MCU - MCU Control
;**************************************************************************/

; MCU_MCUCR masks
.equ MCU_JTAGD_bm = 0x01 ; JTAG Disable bit mask
.equ MCU_JTAGD_bp = 0 ; JTAG Disable bit position

; MCU_ANAINIT masks
.equ MCU_STARTUPDLYB_gm = 0x0C ; Analog startup delay Port B group mask
.equ MCU_STARTUPDLYB_gp = 2 ; Analog startup delay Port B group position
.equ MCU_STARTUPDLYB0_bm = (1<<2) ; Analog startup delay Port B bit 0 mask
.equ MCU_STARTUPDLYB0_bp = 2 ; Analog startup delay Port B bit 0 position
.equ MCU_STARTUPDLYB1_bm = (1<<3) ; Analog startup delay Port B bit 1 mask
.equ MCU_STARTUPDLYB1_bp = 3 ; Analog startup delay Port B bit 1 position
.equ MCU_STARTUPDLYA_gm = 0x03 ; Analog startup delay Port A group mask
.equ MCU_STARTUPDLYA_gp = 0 ; Analog startup delay Port A group position
.equ MCU_STARTUPDLYA0_bm = (1<<0) ; Analog startup delay Port A bit 0 mask
.equ MCU_STARTUPDLYA0_bp = 0 ; Analog startup delay Port A bit 0 position
.equ MCU_STARTUPDLYA1_bm = (1<<1) ; Analog startup delay Port A bit 1 mask
.equ MCU_STARTUPDLYA1_bp = 1 ; Analog startup delay Port A bit 1 position

; MCU_EVSYSLOCK masks
.equ MCU_EVSYS1LOCK_bm = 0x10 ; Event Channel 4-7 Lock bit mask
.equ MCU_EVSYS1LOCK_bp = 4 ; Event Channel 4-7 Lock bit position
.equ MCU_EVSYS0LOCK_bm = 0x01 ; Event Channel 0-3 Lock bit mask
.equ MCU_EVSYS0LOCK_bp = 0 ; Event Channel 0-3 Lock bit position

; MCU_AWEXLOCK masks
.equ MCU_AWEXFLOCK_bm = 0x08 ; AWeX on T/C F0 Lock bit mask
.equ MCU_AWEXFLOCK_bp = 3 ; AWeX on T/C F0 Lock bit position
.equ MCU_AWEXELOCK_bm = 0x04 ; AWeX on T/C E0 Lock bit mask
.equ MCU_AWEXELOCK_bp = 2 ; AWeX on T/C E0 Lock bit position
.equ MCU_AWEXDLOCK_bm = 0x02 ; AWeX on T/C D0 Lock bit mask
.equ MCU_AWEXDLOCK_bp = 1 ; AWeX on T/C D0 Lock bit position
.equ MCU_AWEXCLOCK_bm = 0x01 ; AWeX on T/C C0 Lock bit mask
.equ MCU_AWEXCLOCK_bp = 0 ; AWeX on T/C C0 Lock bit position


;**************************************************************************
;** PMIC - Programmable Multi-level Interrupt Controller
;**************************************************************************/

; PMIC_STATUS masks
.equ PMIC_NMIEX_bm = 0x80 ; Non-maskable Interrupt Executing bit mask
.equ PMIC_NMIEX_bp = 7 ; Non-maskable Interrupt Executing bit position
```

```
2700  .equ PMIC_HILVLEX_bm = 0x04 ; High Level Interrupt Executing bit mask
2701  .equ PMIC_HILVLEX_bp = 2 ; High Level Interrupt Executing bit position
2702  .equ PMIC_MEDLVLEX_bm = 0x02 ; Medium Level Interrupt Executing bit mask
2703  .equ PMIC_MEDLVLEX_bp = 1 ; Medium Level Interrupt Executing bit position
2704  .equ PMIC_LOLVLEX_bm = 0x01 ; Low Level Interrupt Executing bit mask
2705  .equ PMIC_LOLVLEX_bp = 0 ; Low Level Interrupt Executing bit position
2706
2707  ; PMIC_CTRL masks
2708  .equ PMIC_RREN_bm = 0x80 ; Round-Robin Priority Enable bit mask
2709  .equ PMIC_RREN_bp = 7 ; Round-Robin Priority Enable bit position
2710  .equ PMIC_IVSEL_bm = 0x40 ; Interrupt Vector Select bit mask
2711  .equ PMIC_IVSEL_bp = 6 ; Interrupt Vector Select bit position
2712  .equ PMIC_HILVLEN_bm = 0x04 ; High Level Enable bit mask
2713  .equ PMIC_HILVLEN_bp = 2 ; High Level Enable bit position
2714  .equ PMIC_MEDLVLEN_bm = 0x02 ; Medium Level Enable bit mask
2715  .equ PMIC_MEDLVLEN_bp = 1 ; Medium Level Enable bit position
2716  .equ PMIC_LOLVLEN_bm = 0x01 ; Low Level Enable bit mask
2717  .equ PMIC_LOLVLEN_bp = 0 ; Low Level Enable bit position
2718
2719
2720  ;**************************************************************************
2721  ;** PORTCFG - Port Configuration
2722  ;*************************************************************************/
2723
2724  ; PORTCFG_VPCTRLA masks
2725  .equ PORTCFG_VP1MAP_gm = 0xF0 ; Virtual Port 1 Mapping group mask
2726  .equ PORTCFG_VP1MAP_gp = 4 ; Virtual Port 1 Mapping group position
2727  .equ PORTCFG_VP1MAP0_bm = (1<<4) ; Virtual Port 1 Mapping bit 0 mask
2728  .equ PORTCFG_VP1MAP0_bp = 4 ; Virtual Port 1 Mapping bit 0 position
2729  .equ PORTCFG_VP1MAP1_bm = (1<<5) ; Virtual Port 1 Mapping bit 1 mask
2730  .equ PORTCFG_VP1MAP1_bp = 5 ; Virtual Port 1 Mapping bit 1 position
2731  .equ PORTCFG_VP1MAP2_bm = (1<<6) ; Virtual Port 1 Mapping bit 2 mask
2732  .equ PORTCFG_VP1MAP2_bp = 6 ; Virtual Port 1 Mapping bit 2 position
2733  .equ PORTCFG_VP1MAP3_bm = (1<<7) ; Virtual Port 1 Mapping bit 3 mask
2734  .equ PORTCFG_VP1MAP3_bp = 7 ; Virtual Port 1 Mapping bit 3 position
2735  .equ PORTCFG_VP0MAP_gm = 0x0F ; Virtual Port 0 Mapping group mask
2736  .equ PORTCFG_VP0MAP_gp = 0 ; Virtual Port 0 Mapping group position
2737  .equ PORTCFG_VP0MAP0_bm = (1<<0) ; Virtual Port 0 Mapping bit 0 mask
2738  .equ PORTCFG_VP0MAP0_bp = 0 ; Virtual Port 0 Mapping bit 0 position
2739  .equ PORTCFG_VP0MAP1_bm = (1<<1) ; Virtual Port 0 Mapping bit 1 mask
2740  .equ PORTCFG_VP0MAP1_bp = 1 ; Virtual Port 0 Mapping bit 1 position
2741  .equ PORTCFG_VP0MAP2_bm = (1<<2) ; Virtual Port 0 Mapping bit 2 mask
2742  .equ PORTCFG_VP0MAP2_bp = 2 ; Virtual Port 0 Mapping bit 2 position
2743  .equ PORTCFG_VP0MAP3_bm = (1<<3) ; Virtual Port 0 Mapping bit 3 mask
2744  .equ PORTCFG_VP0MAP3_bp = 3 ; Virtual Port 0 Mapping bit 3 position
2745
2746  ; PORTCFG_VPCTRLB masks
2747  .equ PORTCFG_VP3MAP_gm = 0xF0 ; Virtual Port 3 Mapping group mask
2748  .equ PORTCFG_VP3MAP_gp = 4 ; Virtual Port 3 Mapping group position
2749  .equ PORTCFG_VP3MAP0_bm = (1<<4) ; Virtual Port 3 Mapping bit 0 mask
2750  .equ PORTCFG_VP3MAP0_bp = 4 ; Virtual Port 3 Mapping bit 0 position
2751  .equ PORTCFG_VP3MAP1_bm = (1<<5) ; Virtual Port 3 Mapping bit 1 mask
2752  .equ PORTCFG_VP3MAP1_bp = 5 ; Virtual Port 3 Mapping bit 1 position
2753  .equ PORTCFG_VP3MAP2_bm = (1<<6) ; Virtual Port 3 Mapping bit 2 mask
2754  .equ PORTCFG_VP3MAP2_bp = 6 ; Virtual Port 3 Mapping bit 2 position
2755  .equ PORTCFG_VP3MAP3_bm = (1<<7) ; Virtual Port 3 Mapping bit 3 mask
2756  .equ PORTCFG_VP3MAP3_bp = 7 ; Virtual Port 3 Mapping bit 3 position
2757  .equ PORTCFG_VP2MAP_gm = 0x0F ; Virtual Port 2 Mapping group mask
2758  .equ PORTCFG_VP2MAP_gp = 0 ; Virtual Port 2 Mapping group position
2759  .equ PORTCFG_VP2MAP0_bm = (1<<0) ; Virtual Port 2 Mapping bit 0 mask
2760  .equ PORTCFG_VP2MAP0_bp = 0 ; Virtual Port 2 Mapping bit 0 position
2761  .equ PORTCFG_VP2MAP1_bm = (1<<1) ; Virtual Port 2 Mapping bit 1 mask
2762  .equ PORTCFG_VP2MAP1_bp = 1 ; Virtual Port 2 Mapping bit 1 position
2763  .equ PORTCFG_VP2MAP2_bm = (1<<2) ; Virtual Port 2 Mapping bit 2 mask
2764  .equ PORTCFG_VP2MAP2_bp = 2 ; Virtual Port 2 Mapping bit 2 position
2765  .equ PORTCFG_VP2MAP3_bm = (1<<3) ; Virtual Port 2 Mapping bit 3 mask
```

```
.equ PORTCFG_VP2MAP3_bp = 3 ; Virtual Port 2 Mapping bit 3 position

; PORTCFG_CLKEVOUT masks
.equ PORTCFG_CLKOUT_gm = 0x03 ; Peripheral Clock Output Port group mask
.equ PORTCFG_CLKOUT_gp = 0 ; Peripheral Clock Output Port group position
.equ PORTCFG_CLKOUT0_bm = (1<<0) ; Peripheral Clock Output Port bit 0 mask
.equ PORTCFG_CLKOUT0_bp = 0 ; Peripheral Clock Output Port bit 0 position
.equ PORTCFG_CLKOUT1_bm = (1<<1) ; Peripheral Clock Output Port bit 1 mask
.equ PORTCFG_CLKOUT1_bp = 1 ; Peripheral Clock Output Port bit 1 position
.equ PORTCFG_CLKOUTSEL_gm = 0x0C ; Peripheral Clock Output Select group mask
.equ PORTCFG_CLKOUTSEL_gp = 2 ; Peripheral Clock Output Select group position
.equ PORTCFG_CLKOUTSEL0_bm = (1<<2) ; Peripheral Clock Output Select bit 0 mask
.equ PORTCFG_CLKOUTSEL0_bp = 2 ; Peripheral Clock Output Select bit 0 position
.equ PORTCFG_CLKOUTSEL1_bm = (1<<3) ; Peripheral Clock Output Select bit 1 mask
.equ PORTCFG_CLKOUTSEL1_bp = 3 ; Peripheral Clock Output Select bit 1 position
.equ PORTCFG_EVOUT_gm = 0x30 ; Event Output Port group mask
.equ PORTCFG_EVOUT_gp = 4 ; Event Output Port group position
.equ PORTCFG_EVOUT0_bm = (1<<4) ; Event Output Port bit 0 mask
.equ PORTCFG_EVOUT0_bp = 4 ; Event Output Port bit 0 position
.equ PORTCFG_EVOUT1_bm = (1<<5) ; Event Output Port bit 1 mask
.equ PORTCFG_EVOUT1_bp = 5 ; Event Output Port bit 1 position
.equ PORTCFG_RTCOUT_bm = 0x40 ; RTC Clock Output bit mask
.equ PORTCFG_RTCOUT_bp = 6 ; RTC Clock Output bit position
.equ PORTCFG_CLKEVPIN_bm = 0x80 ; Peripheral Clock and Event Output pin Select bit mask
.equ PORTCFG_CLKEVPIN_bp = 7 ; Peripheral Clock and Event Output pin Select bit position

; PORTCFG_EVOUTSEL masks
.equ PORTCFG_EVOUTSEL_gm = 0x07 ; Event Output Select group mask
.equ PORTCFG_EVOUTSEL_gp = 0 ; Event Output Select group position
.equ PORTCFG_EVOUTSEL0_bm = (1<<0) ; Event Output Select bit 0 mask
.equ PORTCFG_EVOUTSEL0_bp = 0 ; Event Output Select bit 0 position
.equ PORTCFG_EVOUTSEL1_bm = (1<<1) ; Event Output Select bit 1 mask
.equ PORTCFG_EVOUTSEL1_bp = 1 ; Event Output Select bit 1 position
.equ PORTCFG_EVOUTSEL2_bm = (1<<2) ; Event Output Select bit 2 mask
.equ PORTCFG_EVOUTSEL2_bp = 2 ; Event Output Select bit 2 position

; Virtual Port Mapping
.equ PORTCFG_VP02MAP_PORTA_gc = (0x00<<0) ; Mapped To PORTA
.equ PORTCFG_VP02MAP_PORTB_gc = (0x01<<0) ; Mapped To PORTB
.equ PORTCFG_VP02MAP_PORTC_gc = (0x02<<0) ; Mapped To PORTC
.equ PORTCFG_VP02MAP_PORTD_gc = (0x03<<0) ; Mapped To PORTD
.equ PORTCFG_VP02MAP_PORTE_gc = (0x04<<0) ; Mapped To PORTE
.equ PORTCFG_VP02MAP_PORTF_gc = (0x05<<0) ; Mapped To PORTF
.equ PORTCFG_VP02MAP_PORTG_gc = (0x06<<0) ; Mapped To PORTG
.equ PORTCFG_VP02MAP_PORTH_gc = (0x07<<0) ; Mapped To PORTH
.equ PORTCFG_VP02MAP_PORTJ_gc = (0x08<<0) ; Mapped To PORTJ
.equ PORTCFG_VP02MAP_PORTK_gc = (0x09<<0) ; Mapped To PORTK
.equ PORTCFG_VP02MAP_PORTL_gc = (0x0A<<0) ; Mapped To PORTL
.equ PORTCFG_VP02MAP_PORTM_gc = (0x0B<<0) ; Mapped To PORTM
.equ PORTCFG_VP02MAP_PORTN_gc = (0x0C<<0) ; Mapped To PORTN
.equ PORTCFG_VP02MAP_PORTP_gc = (0x0D<<0) ; Mapped To PORTP
.equ PORTCFG_VP02MAP_PORTQ_gc = (0x0E<<0) ; Mapped To PORTQ
.equ PORTCFG_VP02MAP_PORTR_gc = (0x0F<<0) ; Mapped To PORTR

; Virtual Port Mapping
.equ PORTCFG_VP13MAP_PORTA_gc = (0x00<<4) ; Mapped To PORTA
.equ PORTCFG_VP13MAP_PORTB_gc = (0x01<<4) ; Mapped To PORTB
.equ PORTCFG_VP13MAP_PORTC_gc = (0x02<<4) ; Mapped To PORTC
.equ PORTCFG_VP13MAP_PORTD_gc = (0x03<<4) ; Mapped To PORTD
.equ PORTCFG_VP13MAP_PORTE_gc = (0x04<<4) ; Mapped To PORTE
.equ PORTCFG_VP13MAP_PORTF_gc = (0x05<<4) ; Mapped To PORTF
.equ PORTCFG_VP13MAP_PORTG_gc = (0x06<<4) ; Mapped To PORTG
.equ PORTCFG_VP13MAP_PORTH_gc = (0x07<<4) ; Mapped To PORTH
.equ PORTCFG_VP13MAP_PORTJ_gc = (0x08<<4) ; Mapped To PORTJ
.equ PORTCFG_VP13MAP_PORTK_gc = (0x09<<4) ; Mapped To PORTK
.equ PORTCFG_VP13MAP_PORTL_gc = (0x0A<<4) ; Mapped To PORTL
```

```
.equ PORTCFG_VP13MAP_PORTM_gc = (0x0B<<4) ; Mapped To PORTM
.equ PORTCFG_VP13MAP_PORTN_gc = (0x0C<<4) ; Mapped To PORTN
.equ PORTCFG_VP13MAP_PORTP_gc = (0x0D<<4) ; Mapped To PORTP
.equ PORTCFG_VP13MAP_PORTQ_gc = (0x0E<<4) ; Mapped To PORTQ
.equ PORTCFG_VP13MAP_PORTR_gc = (0x0F<<4) ; Mapped To PORTR

; System Clock Output Port
.equ PORTCFG_CLKOUT_OFF_gc = (0x00<<0) ; System Clock Output Disabled
.equ PORTCFG_CLKOUT_PC7_gc = (0x01<<0) ; System Clock Output on Port C pin 7
.equ PORTCFG_CLKOUT_PD7_gc = (0x02<<0) ; System Clock Output on Port D pin 7
.equ PORTCFG_CLKOUT_PE7_gc = (0x03<<0) ; System Clock Output on Port E pin 7

; Peripheral Clock Output Select
.equ PORTCFG_CLKOUTSEL_CLK1X_gc = (0x00<<2) ; 1x Peripheral Clock Output to pin
.equ PORTCFG_CLKOUTSEL_CLK2X_gc = (0x01<<2) ; 2x Peripheral Clock Output to pin
.equ PORTCFG_CLKOUTSEL_CLK4X_gc = (0x02<<2) ; 4x Peripheral Clock Output to pin

; Event Output Port
.equ PORTCFG_EVOUT_OFF_gc = (0x00<<4) ; Event Output Disabled
.equ PORTCFG_EVOUT_PC7_gc = (0x01<<4) ; Event Channel 7 Output on Port C pin 7
.equ PORTCFG_EVOUT_PD7_gc = (0x02<<4) ; Event Channel 7 Output on Port D pin 7
.equ PORTCFG_EVOUT_PE7_gc = (0x03<<4) ; Event Channel 7 Output on Port E pin 7

; Clock and Event Output Port
.equ PORTCFG_CLKEVPIN_PIN7_gc = (0x00<<7) ; Clock and Event Ouput on PIN 7
.equ PORTCFG_CLKEVPIN_PIN4_gc = (0x01<<7) ; Clock and Event Ouput on PIN 4

; Event Output Select
.equ PORTCFG_EVOUTSEL_0_gc = (0x00<<0) ; Event Channel 0 output to pin
.equ PORTCFG_EVOUTSEL_1_gc = (0x01<<0) ; Event Channel 1 output to pin
.equ PORTCFG_EVOUTSEL_2_gc = (0x02<<0) ; Event Channel 2 output to pin
.equ PORTCFG_EVOUTSEL_3_gc = (0x03<<0) ; Event Channel 3 output to pin
.equ PORTCFG_EVOUTSEL_4_gc = (0x04<<0) ; Event Channel 4 output to pin
.equ PORTCFG_EVOUTSEL_5_gc = (0x05<<0) ; Event Channel 5 output to pin
.equ PORTCFG_EVOUTSEL_6_gc = (0x06<<0) ; Event Channel 6 output to pin
.equ PORTCFG_EVOUTSEL_7_gc = (0x07<<0) ; Event Channel 7 output to pin


;**************************************************************************
;** AES - AES Module
;**************************************************************************/

; AES_CTRL masks
.equ AES_START_bm = 0x80 ; Start/Run bit mask
.equ AES_START_bp = 7 ; Start/Run bit position
.equ AES_AUTO_bm = 0x40 ; Auto Start Trigger bit mask
.equ AES_AUTO_bp = 6 ; Auto Start Trigger bit position
.equ AES_RESET_bm = 0x20 ; AES Software Reset bit mask
.equ AES_RESET_bp = 5 ; AES Software Reset bit position
.equ AES_DECRYPT_bm = 0x10 ; Decryption / Direction bit mask
.equ AES_DECRYPT_bp = 4 ; Decryption / Direction bit position
.equ AES_XOR_bm = 0x04 ; State XOR Load Enable bit mask
.equ AES_XOR_bp = 2 ; State XOR Load Enable bit position

; AES_STATUS masks
.equ AES_ERROR_bm = 0x80 ; AES Error bit mask
.equ AES_ERROR_bp = 7 ; AES Error bit position
.equ AES_SRIF_bm = 0x01 ; State Ready Interrupt Flag bit mask
.equ AES_SRIF_bp = 0 ; State Ready Interrupt Flag bit position

; AES_INTCTRL masks
.equ AES_INTLVL_gm = 0x03 ; Interrupt level group mask
.equ AES_INTLVL_gp = 0 ; Interrupt level group position
.equ AES_INTLVL0_bm = (1<<0) ; Interrupt level bit 0 mask
.equ AES_INTLVL0_bp = 0 ; Interrupt level bit 0 position
.equ AES_INTLVL1_bm = (1<<1) ; Interrupt level bit 1 mask
```

```
2898  .equ AES_INTLVL1_bp = 1 ; Interrupt level bit 1 position
2899
2900  ; Interrupt level
2901  .equ AES_INTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
2902  .equ AES_INTLVL_LO_gc = (0x01<<0) ; Low Level
2903  .equ AES_INTLVL_MED_gc = (0x02<<0) ; Medium Level
2904  .equ AES_INTLVL_HI_gc = (0x03<<0) ; High Level
2905
2906
2907  ;**************************************************************************
2908  ;** CRC - Cyclic Redundancy Checker
2909  ;**************************************************************************/
2910
2911  ; CRC_CTRL masks
2912  .equ CRC_RESET_gm = 0xC0 ; Reset group mask
2913  .equ CRC_RESET_gp = 6 ; Reset group position
2914  .equ CRC_RESET0_bm = (1<<6) ; Reset bit 0 mask
2915  .equ CRC_RESET0_bp = 6 ; Reset bit 0 position
2916  .equ CRC_RESET1_bm = (1<<7) ; Reset bit 1 mask
2917  .equ CRC_RESET1_bp = 7 ; Reset bit 1 position
2918  .equ CRC_CRC32_bm = 0x20 ; CRC Mode bit mask
2919  .equ CRC_CRC32_bp = 5 ; CRC Mode bit position
2920  .equ CRC_SOURCE_gm = 0x0F ; Input Source group mask
2921  .equ CRC_SOURCE_gp = 0 ; Input Source group position
2922  .equ CRC_SOURCE0_bm = (1<<0) ; Input Source bit 0 mask
2923  .equ CRC_SOURCE0_bp = 0 ; Input Source bit 0 position
2924  .equ CRC_SOURCE1_bm = (1<<1) ; Input Source bit 1 mask
2925  .equ CRC_SOURCE1_bp = 1 ; Input Source bit 1 position
2926  .equ CRC_SOURCE2_bm = (1<<2) ; Input Source bit 2 mask
2927  .equ CRC_SOURCE2_bp = 2 ; Input Source bit 2 position
2928  .equ CRC_SOURCE3_bm = (1<<3) ; Input Source bit 3 mask
2929  .equ CRC_SOURCE3_bp = 3 ; Input Source bit 3 position
2930
2931  ; CRC_STATUS masks
2932  .equ CRC_ZERO_bm = 0x02 ; Zero detection bit mask
2933  .equ CRC_ZERO_bp = 1 ; Zero detection bit position
2934  .equ CRC_BUSY_bm = 0x01 ; Busy bit mask
2935  .equ CRC_BUSY_bp = 0 ; Busy bit position
2936
2937  ; Reset
2938  .equ CRC_RESET_NO_gc = (0x00<<6) ; No Reset
2939  .equ CRC_RESET_RESET0_gc = (0x02<<6) ; Reset CRC with CHECKSUM to all zeros
2940  .equ CRC_RESET_RESET1_gc = (0x03<<6) ; Reset CRC with CHECKSUM to all ones
2941
2942  ; Input Source
2943  .equ CRC_SOURCE_DISABLE_gc = (0x00<<0) ; Disabled
2944  .equ CRC_SOURCE_IO_gc = (0x01<<0) ; I/O Interface
2945  .equ CRC_SOURCE_FLASH_gc = (0x02<<0) ; Flash
2946  .equ CRC_SOURCE_DMAC0_gc = (0x04<<0) ; DMAC Channel 0
2947  .equ CRC_SOURCE_DMAC1_gc = (0x05<<0) ; DMAC Channel 1
2948  .equ CRC_SOURCE_DMAC2_gc = (0x06<<0) ; DMAC Channel 2
2949  .equ CRC_SOURCE_DMAC3_gc = (0x07<<0) ; DMAC Channel 3
2950
2951
2952  ;**************************************************************************
2953  ;** DMA - DMA Controller
2954  ;**************************************************************************/
2955
2956  ; DMA_CH_CTRLA masks
2957  .equ DMA_CH_ENABLE_bm = 0x80 ; Channel Enable bit mask
2958  .equ DMA_CH_ENABLE_bp = 7 ; Channel Enable bit position
2959  .equ DMA_CH_RESET_bm = 0x40 ; Channel Software Reset bit mask
2960  .equ DMA_CH_RESET_bp = 6 ; Channel Software Reset bit position
2961  .equ DMA_CH_REPEAT_bm = 0x20 ; Channel Repeat Mode bit mask
2962  .equ DMA_CH_REPEAT_bp = 5 ; Channel Repeat Mode bit position
2963  .equ DMA_CH_TRFREQ_bm = 0x10 ; Channel Transfer Request bit mask
```

```
.equ DMA_CH_TRFREQ_bp = 4 ; Channel Transfer Request bit position
.equ DMA_CH_SINGLE_bm = 0x04 ; Channel Single Shot Data Transfer bit mask
.equ DMA_CH_SINGLE_bp = 2 ; Channel Single Shot Data Transfer bit position
.equ DMA_CH_BURSTLEN_gm = 0x03 ; Channel Transfer Mode group mask
.equ DMA_CH_BURSTLEN_gp = 0 ; Channel Transfer Mode group position
.equ DMA_CH_BURSTLEN0_bm = (1<<0) ; Channel Transfer Mode bit 0 mask
.equ DMA_CH_BURSTLEN0_bp = 0 ; Channel Transfer Mode bit 0 position
.equ DMA_CH_BURSTLEN1_bm = (1<<1) ; Channel Transfer Mode bit 1 mask
.equ DMA_CH_BURSTLEN1_bp = 1 ; Channel Transfer Mode bit 1 position

; DMA_CH_CTRLB masks
.equ DMA_CH_CHBUSY_bm = 0x80 ; Block Transfer Busy bit mask
.equ DMA_CH_CHBUSY_bp = 7 ; Block Transfer Busy bit position
.equ DMA_CH_CHPEND_bm = 0x40 ; Block Transfer Pending bit mask
.equ DMA_CH_CHPEND_bp = 6 ; Block Transfer Pending bit position
.equ DMA_CH_ERRIF_bm = 0x20 ; Block Transfer Error Interrupt Flag bit mask
.equ DMA_CH_ERRIF_bp = 5 ; Block Transfer Error Interrupt Flag bit position
.equ DMA_CH_TRNIF_bm = 0x10 ; Transaction Complete Interrup Flag bit mask
.equ DMA_CH_TRNIF_bp = 4 ; Transaction Complete Interrup Flag bit position
.equ DMA_CH_ERRINTLVL_gm = 0x0C ; Transfer Error Interrupt Level group mask
.equ DMA_CH_ERRINTLVL_gp = 2 ; Transfer Error Interrupt Level group position
.equ DMA_CH_ERRINTLVL0_bm = (1<<2) ; Transfer Error Interrupt Level bit 0 mask
.equ DMA_CH_ERRINTLVL0_bp = 2 ; Transfer Error Interrupt Level bit 0 position
.equ DMA_CH_ERRINTLVL1_bm = (1<<3) ; Transfer Error Interrupt Level bit 1 mask
.equ DMA_CH_ERRINTLVL1_bp = 3 ; Transfer Error Interrupt Level bit 1 position
.equ DMA_CH_TRNINTLVL_gm = 0x03 ; Transaction Complete Interrupt Level group mask
.equ DMA_CH_TRNINTLVL_gp = 0 ; Transaction Complete Interrupt Level group position
.equ DMA_CH_TRNINTLVL0_bm = (1<<0) ; Transaction Complete Interrupt Level bit 0 mask
.equ DMA_CH_TRNINTLVL0_bp = 0 ; Transaction Complete Interrupt Level bit 0 position
.equ DMA_CH_TRNINTLVL1_bm = (1<<1) ; Transaction Complete Interrupt Level bit 1 mask
.equ DMA_CH_TRNINTLVL1_bp = 1 ; Transaction Complete Interrupt Level bit 1 position

; DMA_CH_ADDRCTRL masks
.equ DMA_CH_SRCRELOAD_gm = 0xC0 ; Channel Source Address Reload group mask
.equ DMA_CH_SRCRELOAD_gp = 6 ; Channel Source Address Reload group position
.equ DMA_CH_SRCRELOAD0_bm = (1<<6) ; Channel Source Address Reload bit 0 mask
.equ DMA_CH_SRCRELOAD0_bp = 6 ; Channel Source Address Reload bit 0 position
.equ DMA_CH_SRCRELOAD1_bm = (1<<7) ; Channel Source Address Reload bit 1 mask
.equ DMA_CH_SRCRELOAD1_bp = 7 ; Channel Source Address Reload bit 1 position
.equ DMA_CH_SRCDIR_gm = 0x30 ; Channel Source Address Mode group mask
.equ DMA_CH_SRCDIR_gp = 4 ; Channel Source Address Mode group position
.equ DMA_CH_SRCDIR0_bm = (1<<4) ; Channel Source Address Mode bit 0 mask
.equ DMA_CH_SRCDIR0_bp = 4 ; Channel Source Address Mode bit 0 position
.equ DMA_CH_SRCDIR1_bm = (1<<5) ; Channel Source Address Mode bit 1 mask
.equ DMA_CH_SRCDIR1_bp = 5 ; Channel Source Address Mode bit 1 position
.equ DMA_CH_DESTRELOAD_gm = 0x0C ; Channel Destination Address Reload group mask
.equ DMA_CH_DESTRELOAD_gp = 2 ; Channel Destination Address Reload group position
.equ DMA_CH_DESTRELOAD0_bm = (1<<2) ; Channel Destination Address Reload bit 0 mask
.equ DMA_CH_DESTRELOAD0_bp = 2 ; Channel Destination Address Reload bit 0 position
.equ DMA_CH_DESTRELOAD1_bm = (1<<3) ; Channel Destination Address Reload bit 1 mask
.equ DMA_CH_DESTRELOAD1_bp = 3 ; Channel Destination Address Reload bit 1 position
.equ DMA_CH_DESTDIR_gm = 0x03 ; Channel Destination Address Mode group mask
.equ DMA_CH_DESTDIR_gp = 0 ; Channel Destination Address Mode group position
.equ DMA_CH_DESTDIR0_bm = (1<<0) ; Channel Destination Address Mode bit 0 mask
.equ DMA_CH_DESTDIR0_bp = 0 ; Channel Destination Address Mode bit 0 position
.equ DMA_CH_DESTDIR1_bm = (1<<1) ; Channel Destination Address Mode bit 1 mask
.equ DMA_CH_DESTDIR1_bp = 1 ; Channel Destination Address Mode bit 1 position

; DMA_CH_TRIGSRC masks
.equ DMA_CH_TRIGSRC_gm = 0xFF ; Channel Trigger Source group mask
.equ DMA_CH_TRIGSRC_gp = 0 ; Channel Trigger Source group position
.equ DMA_CH_TRIGSRC0_bm = (1<<0) ; Channel Trigger Source bit 0 mask
.equ DMA_CH_TRIGSRC0_bp = 0 ; Channel Trigger Source bit 0 position
.equ DMA_CH_TRIGSRC1_bm = (1<<1) ; Channel Trigger Source bit 1 mask
.equ DMA_CH_TRIGSRC1_bp = 1 ; Channel Trigger Source bit 1 position
.equ DMA_CH_TRIGSRC2_bm = (1<<2) ; Channel Trigger Source bit 2 mask
```

```
3030   .equ DMA_CH_TRIGSRC2_bp = 2 ; Channel Trigger Source bit 2 position
3031   .equ DMA_CH_TRIGSRC3_bm = (1<<3) ; Channel Trigger Source bit 3 mask
3032   .equ DMA_CH_TRIGSRC3_bp = 3 ; Channel Trigger Source bit 3 position
3033   .equ DMA_CH_TRIGSRC4_bm = (1<<4) ; Channel Trigger Source bit 4 mask
3034   .equ DMA_CH_TRIGSRC4_bp = 4 ; Channel Trigger Source bit 4 position
3035   .equ DMA_CH_TRIGSRC5_bm = (1<<5) ; Channel Trigger Source bit 5 mask
3036   .equ DMA_CH_TRIGSRC5_bp = 5 ; Channel Trigger Source bit 5 position
3037   .equ DMA_CH_TRIGSRC6_bm = (1<<6) ; Channel Trigger Source bit 6 mask
3038   .equ DMA_CH_TRIGSRC6_bp = 6 ; Channel Trigger Source bit 6 position
3039   .equ DMA_CH_TRIGSRC7_bm = (1<<7) ; Channel Trigger Source bit 7 mask
3040   .equ DMA_CH_TRIGSRC7_bp = 7 ; Channel Trigger Source bit 7 position
3041
3042   ; DMA_CTRL masks
3043   .equ DMA_ENABLE_bm = 0x80 ; Enable bit mask
3044   .equ DMA_ENABLE_bp = 7 ; Enable bit position
3045   .equ DMA_RESET_bm = 0x40 ; Software Reset bit mask
3046   .equ DMA_RESET_bp = 6 ; Software Reset bit position
3047   .equ DMA_DBUFMODE_gm = 0x0C ; Double Buffering Mode group mask
3048   .equ DMA_DBUFMODE_gp = 2 ; Double Buffering Mode group position
3049   .equ DMA_DBUFMODE0_bm = (1<<2) ; Double Buffering Mode bit 0 mask
3050   .equ DMA_DBUFMODE0_bp = 2 ; Double Buffering Mode bit 0 position
3051   .equ DMA_DBUFMODE1_bm = (1<<3) ; Double Buffering Mode bit 1 mask
3052   .equ DMA_DBUFMODE1_bp = 3 ; Double Buffering Mode bit 1 position
3053   .equ DMA_PRIMODE_gm = 0x03 ; Channel Priority Mode group mask
3054   .equ DMA_PRIMODE_gp = 0 ; Channel Priority Mode group position
3055   .equ DMA_PRIMODE0_bm = (1<<0) ; Channel Priority Mode bit 0 mask
3056   .equ DMA_PRIMODE0_bp = 0 ; Channel Priority Mode bit 0 position
3057   .equ DMA_PRIMODE1_bm = (1<<1) ; Channel Priority Mode bit 1 mask
3058   .equ DMA_PRIMODE1_bp = 1 ; Channel Priority Mode bit 1 position
3059
3060   ; DMA_INTFLAGS masks
3061   .equ DMA_CH3ERRIF_bm = 0x80 ; Channel 3 Block Transfer Error Interrupt Flag bit mask
3062   .equ DMA_CH3ERRIF_bp = 7 ; Channel 3 Block Transfer Error Interrupt Flag bit position
3063   .equ DMA_CH2ERRIF_bm = 0x40 ; Channel 2 Block Transfer Error Interrupt Flag bit mask
3064   .equ DMA_CH2ERRIF_bp = 6 ; Channel 2 Block Transfer Error Interrupt Flag bit position
3065   .equ DMA_CH1ERRIF_bm = 0x20 ; Channel 1 Block Transfer Error Interrupt Flag bit mask
3066   .equ DMA_CH1ERRIF_bp = 5 ; Channel 1 Block Transfer Error Interrupt Flag bit position
3067   .equ DMA_CH0ERRIF_bm = 0x10 ; Channel 0 Block Transfer Error Interrupt Flag bit mask
3068   .equ DMA_CH0ERRIF_bp = 4 ; Channel 0 Block Transfer Error Interrupt Flag bit position
3069   .equ DMA_CH3TRNIF_bm = 0x08 ; Channel 3 Transaction Complete Interrupt Flag bit mask
3070   .equ DMA_CH3TRNIF_bp = 3 ; Channel 3 Transaction Complete Interrupt Flag bit position
3071   .equ DMA_CH2TRNIF_bm = 0x04 ; Channel 2 Transaction Complete Interrupt Flag bit mask
3072   .equ DMA_CH2TRNIF_bp = 2 ; Channel 2 Transaction Complete Interrupt Flag bit position
3073   .equ DMA_CH1TRNIF_bm = 0x02 ; Channel 1 Transaction Complete Interrupt Flag bit mask
3074   .equ DMA_CH1TRNIF_bp = 1 ; Channel 1 Transaction Complete Interrupt Flag bit position
3075   .equ DMA_CH0TRNIF_bm = 0x01 ; Channel 0 Transaction Complete Interrupt Flag bit mask
3076   .equ DMA_CH0TRNIF_bp = 0 ; Channel 0 Transaction Complete Interrupt Flag bit position
3077
3078   ; DMA_STATUS masks
3079   .equ DMA_CH3BUSY_bm = 0x80 ; Channel 3 Block Transfer Busy bit mask
3080   .equ DMA_CH3BUSY_bp = 7 ; Channel 3 Block Transfer Busy bit position
3081   .equ DMA_CH2BUSY_bm = 0x40 ; Channel 2 Block Transfer Busy bit mask
3082   .equ DMA_CH2BUSY_bp = 6 ; Channel 2 Block Transfer Busy bit position
3083   .equ DMA_CH1BUSY_bm = 0x20 ; Channel 1 Block Transfer Busy bit mask
3084   .equ DMA_CH1BUSY_bp = 5 ; Channel 1 Block Transfer Busy bit position
3085   .equ DMA_CH0BUSY_bm = 0x10 ; Channel 0 Block Transfer Busy bit mask
3086   .equ DMA_CH0BUSY_bp = 4 ; Channel 0 Block Transfer Busy bit position
3087   .equ DMA_CH3PEND_bm = 0x08 ; Channel 3 Block Transfer Pending bit mask
3088   .equ DMA_CH3PEND_bp = 3 ; Channel 3 Block Transfer Pending bit position
3089   .equ DMA_CH2PEND_bm = 0x04 ; Channel 2 Block Transfer Pending bit mask
3090   .equ DMA_CH2PEND_bp = 2 ; Channel 2 Block Transfer Pending bit position
3091   .equ DMA_CH1PEND_bm = 0x02 ; Channel 1 Block Transfer Pending bit mask
3092   .equ DMA_CH1PEND_bp = 1 ; Channel 1 Block Transfer Pending bit position
3093   .equ DMA_CH0PEND_bm = 0x01 ; Channel 0 Block Transfer Pending bit mask
3094   .equ DMA_CH0PEND_bp = 0 ; Channel 0 Block Transfer Pending bit position
3095
```

```
; Burst mode
.equ DMA_CH_BURSTLEN_1BYTE_gc = (0x00<<0) ; 1-byte burst mode
.equ DMA_CH_BURSTLEN_2BYTE_gc = (0x01<<0) ; 2-byte burst mode
.equ DMA_CH_BURSTLEN_4BYTE_gc = (0x02<<0) ; 4-byte burst mode
.equ DMA_CH_BURSTLEN_8BYTE_gc = (0x03<<0) ; 8-byte burst mode

; Source address reload mode
.equ DMA_CH_SRCRELOAD_NONE_gc = (0x00<<6) ; No reload
.equ DMA_CH_SRCRELOAD_BLOCK_gc = (0x01<<6) ; Reload at end of block
.equ DMA_CH_SRCRELOAD_BURST_gc = (0x02<<6) ; Reload at end of burst
.equ DMA_CH_SRCRELOAD_TRANSACTION_gc = (0x03<<6) ; Reload at end of transaction

; Source addressing mode
.equ DMA_CH_SRCDIR_FIXED_gc = (0x00<<4) ; Fixed
.equ DMA_CH_SRCDIR_INC_gc = (0x01<<4) ; Increment
.equ DMA_CH_SRCDIR_DEC_gc = (0x02<<4) ; Decrement

; Destination adress reload mode
.equ DMA_CH_DESTRELOAD_NONE_gc = (0x00<<2) ; No reload
.equ DMA_CH_DESTRELOAD_BLOCK_gc = (0x01<<2) ; Reload at end of block
.equ DMA_CH_DESTRELOAD_BURST_gc = (0x02<<2) ; Reload at end of burst
.equ DMA_CH_DESTRELOAD_TRANSACTION_gc = (0x03<<2) ; Reload at end of transaction

; Destination adressing mode
.equ DMA_CH_DESTDIR_FIXED_gc = (0x00<<0) ; Fixed
.equ DMA_CH_DESTDIR_INC_gc = (0x01<<0) ; Increment
.equ DMA_CH_DESTDIR_DEC_gc = (0x02<<0) ; Decrement

; Transfer trigger source
.equ DMA_CH_TRIGSRC_OFF_gc = (0x00<<0) ; Off software triggers only
.equ DMA_CH_TRIGSRC_EVSYS_CH0_gc = (0x01<<0) ; Event System Channel 0
.equ DMA_CH_TRIGSRC_EVSYS_CH1_gc = (0x02<<0) ; Event System Channel 1
.equ DMA_CH_TRIGSRC_EVSYS_CH2_gc = (0x03<<0) ; Event System Channel 2
.equ DMA_CH_TRIGSRC_ADCA_CH0_gc = (0x10<<0) ; ADCA Channel 0
.equ DMA_CH_TRIGSRC_ADCA_CH1_gc = (0x11<<0) ; ADCA Channel 1
.equ DMA_CH_TRIGSRC_ADCA_CH2_gc = (0x12<<0) ; ADCA Channel 2
.equ DMA_CH_TRIGSRC_ADCA_CH3_gc = (0x13<<0) ; ADCA Channel 3
.equ DMA_CH_TRIGSRC_ADCA_CH4_gc = (0x14<<0) ; ADCA Channel 0,1,2,3 combined
.equ DMA_CH_TRIGSRC_DACA_CH0_gc = (0x15<<0) ; DACA Channel 0
.equ DMA_CH_TRIGSRC_DACA_CH1_gc = (0x16<<0) ; DACA Channel 1
.equ DMA_CH_TRIGSRC_ADCB_CH0_gc = (0x20<<0) ; ADCB Channel 0
.equ DMA_CH_TRIGSRC_ADCB_CH1_gc = (0x21<<0) ; ADCB Channel 1
.equ DMA_CH_TRIGSRC_ADCB_CH2_gc = (0x22<<0) ; ADCB Channel 2
.equ DMA_CH_TRIGSRC_ADCB_CH3_gc = (0x23<<0) ; ADCB Channel 3
.equ DMA_CH_TRIGSRC_ADCB_CH4_gc = (0x24<<0) ; ADCB Channel 0,1,2,3 combined
.equ DMA_CH_TRIGSRC_DACB_CH0_gc = (0x25<<0) ; DACB Channel 0
.equ DMA_CH_TRIGSRC_DACB_CH1_gc = (0x26<<0) ; DACB Channel 1
.equ DMA_CH_TRIGSRC_TCC0_OVF_gc = (0x40<<0) ; Timer/Counter C0 Overflow
.equ DMA_CH_TRIGSRC_TCC0_ERR_gc = (0x41<<0) ; Timer/Counter C0 Error
.equ DMA_CH_TRIGSRC_TCC0_CCA_gc = (0x42<<0) ; Timer/Counter C0 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCC0_CCB_gc = (0x43<<0) ; Timer/Counter C0 Compare or Capture B
.equ DMA_CH_TRIGSRC_TCC0_CCC_gc = (0x44<<0) ; Timer/Counter C0 Compare or Capture C
.equ DMA_CH_TRIGSRC_TCC0_CCD_gc = (0x45<<0) ; Timer/Counter C0 Compare or Capture D
.equ DMA_CH_TRIGSRC_TCC1_OVF_gc = (0x46<<0) ; Timer/Counter C1 Overflow
.equ DMA_CH_TRIGSRC_TCC1_ERR_gc = (0x47<<0) ; Timer/Counter C1 Error
.equ DMA_CH_TRIGSRC_TCC1_CCA_gc = (0x48<<0) ; Timer/Counter C1 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCC1_CCB_gc = (0x49<<0) ; Timer/Counter C1 Compare or Capture B
.equ DMA_CH_TRIGSRC_SPIC_gc = (0x4A<<0) ; SPI C Transfer Complete
.equ DMA_CH_TRIGSRC_USARTC0_RXC_gc = (0x4B<<0) ; USART C0 Receive Complete
.equ DMA_CH_TRIGSRC_USARTC0_DRE_gc = (0x4C<<0) ; USART C0 Data Register Empty
.equ DMA_CH_TRIGSRC_USARTC1_RXC_gc = (0x4E<<0) ; USART C1 Receive Complete
.equ DMA_CH_TRIGSRC_USARTC1_DRE_gc = (0x4F<<0) ; USART C1 Data Register Empty
.equ DMA_CH_TRIGSRC_TCD0_OVF_gc = (0x60<<0) ; Timer/Counter D0 Overflow
.equ DMA_CH_TRIGSRC_TCD0_ERR_gc = (0x61<<0) ; Timer/Counter D0 Error
.equ DMA_CH_TRIGSRC_TCD0_CCA_gc = (0x62<<0) ; Timer/Counter D0 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCD0_CCB_gc = (0x63<<0) ; Timer/Counter D0 Compare or Capture B
```

```
.equ DMA_CH_TRIGSRC_TCD0_CCC_gc = (0x64<<0) ; Timer/Counter D0 Compare or Capture C
.equ DMA_CH_TRIGSRC_TCD0_CCD_gc = (0x65<<0) ; Timer/Counter D0 Compare or Capture D
.equ DMA_CH_TRIGSRC_TCD1_OVF_gc = (0x66<<0) ; Timer/Counter D1 Overflow
.equ DMA_CH_TRIGSRC_TCD1_ERR_gc = (0x67<<0) ; Timer/Counter D1 Error
.equ DMA_CH_TRIGSRC_TCD1_CCA_gc = (0x68<<0) ; Timer/Counter D1 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCD1_CCB_gc = (0x69<<0) ; Timer/Counter D1 Compare or Capture B
.equ DMA_CH_TRIGSRC_SPID_gc = (0x6A<<0) ; SPI D Transfer Complete
.equ DMA_CH_TRIGSRC_USARTD0_RXC_gc = (0x6B<<0) ; USART D0 Receive Complete
.equ DMA_CH_TRIGSRC_USARTD0_DRE_gc = (0x6C<<0) ; USART D0 Data Register Empty
.equ DMA_CH_TRIGSRC_USARTD1_RXC_gc = (0x6E<<0) ; USART D1 Receive Complete
.equ DMA_CH_TRIGSRC_USARTD1_DRE_gc = (0x6F<<0) ; USART D1 Data Register Empty
.equ DMA_CH_TRIGSRC_TCE0_OVF_gc = (0x80<<0) ; Timer/Counter E0 Overflow
.equ DMA_CH_TRIGSRC_TCE0_ERR_gc = (0x81<<0) ; Timer/Counter E0 Error
.equ DMA_CH_TRIGSRC_TCE0_CCA_gc = (0x82<<0) ; Timer/Counter E0 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCE0_CCB_gc = (0x83<<0) ; Timer/Counter E0 Compare or Capture B
.equ DMA_CH_TRIGSRC_TCE0_CCC_gc = (0x84<<0) ; Timer/Counter E0 Compare or Capture C
.equ DMA_CH_TRIGSRC_TCE0_CCD_gc = (0x85<<0) ; Timer/Counter E0 Compare or Capture D
.equ DMA_CH_TRIGSRC_TCE1_OVF_gc = (0x86<<0) ; Timer/Counter E1 Overflow
.equ DMA_CH_TRIGSRC_TCE1_ERR_gc = (0x87<<0) ; Timer/Counter E1 Error
.equ DMA_CH_TRIGSRC_TCE1_CCA_gc = (0x88<<0) ; Timer/Counter E1 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCE1_CCB_gc = (0x89<<0) ; Timer/Counter E1 Compare or Capture B
.equ DMA_CH_TRIGSRC_SPIE_gc = (0x8A<<0) ; SPI E Transfer Complete
.equ DMA_CH_TRIGSRC_USARTE0_RXC_gc = (0x8B<<0) ; USART E0 Receive Complete
.equ DMA_CH_TRIGSRC_USARTE0_DRE_gc = (0x8C<<0) ; USART E0 Data Register Empty
.equ DMA_CH_TRIGSRC_USARTE1_RXC_gc = (0x8E<<0) ; USART E1 Receive Complete
.equ DMA_CH_TRIGSRC_USARTE1_DRE_gc = (0x8F<<0) ; USART E1 Data Register Empty
.equ DMA_CH_TRIGSRC_TCF0_OVF_gc = (0xA0<<0) ; Timer/Counter F0 Overflow
.equ DMA_CH_TRIGSRC_TCF0_ERR_gc = (0xA1<<0) ; Timer/Counter F0 Error
.equ DMA_CH_TRIGSRC_TCF0_CCA_gc = (0xA2<<0) ; Timer/Counter F0 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCF0_CCB_gc = (0xA3<<0) ; Timer/Counter F0 Compare or Capture B
.equ DMA_CH_TRIGSRC_TCF0_CCC_gc = (0xA4<<0) ; Timer/Counter F0 Compare or Capture C
.equ DMA_CH_TRIGSRC_TCF0_CCD_gc = (0xA5<<0) ; Timer/Counter F0 Compare or Capture D
.equ DMA_CH_TRIGSRC_TCF1_OVF_gc = (0xA6<<0) ; Timer/Counter F1 Overflow
.equ DMA_CH_TRIGSRC_TCF1_ERR_gc = (0xA7<<0) ; Timer/Counter F1 Error
.equ DMA_CH_TRIGSRC_TCF1_CCA_gc = (0xA8<<0) ; Timer/Counter F1 Compare or Capture A
.equ DMA_CH_TRIGSRC_TCF1_CCB_gc = (0xA9<<0) ; Timer/Counter F1 Compare or Capture B
.equ DMA_CH_TRIGSRC_SPIF_gc = (0xAA<<0) ; SPI F Transfer Complete
.equ DMA_CH_TRIGSRC_USARTF0_RXC_gc = (0xAB<<0) ; USART F0 Receive Complete
.equ DMA_CH_TRIGSRC_USARTF0_DRE_gc = (0xAC<<0) ; USART F0 Data Register Empty
.equ DMA_CH_TRIGSRC_USARTF1_RXC_gc = (0xAE<<0) ; USART F1 Receive Complete
.equ DMA_CH_TRIGSRC_USARTF1_DRE_gc = (0xAF<<0) ; USART F1 Data Register Empty

; Double buffering mode
.equ DMA_DBUFMODE_DISABLED_gc = (0x00<<2) ; Double buffering disabled
.equ DMA_DBUFMODE_CH01_gc = (0x01<<2) ; Double buffering enabled on channel 0/1
.equ DMA_DBUFMODE_CH23_gc = (0x02<<2) ; Double buffering enabled on channel 2/3
.equ DMA_DBUFMODE_CH01CH23_gc = (0x03<<2) ; Double buffering enabled on ch. 0/1 and ch.
2/3

; Priority mode
.equ DMA_PRIMODE_RR0123_gc = (0x00<<0) ; Round Robin
.equ DMA_PRIMODE_CH0RR123_gc = (0x01<<0) ; Channel 0 > Round Robin on channel 1/2/3
.equ DMA_PRIMODE_CH01RR23_gc = (0x02<<0) ; Channel 0 > channel 1 > Round Robin on
channel 2/3
.equ DMA_PRIMODE_CH0123_gc = (0x03<<0) ; Channel 0 > channel 1 > channel 2 > channel 3

; Interrupt level
.equ DMA_CH_ERRINTLVL_OFF_gc = (0x00<<2) ; Interrupt disabled
.equ DMA_CH_ERRINTLVL_LO_gc = (0x01<<2) ; Low level
.equ DMA_CH_ERRINTLVL_MED_gc = (0x02<<2) ; Medium level
.equ DMA_CH_ERRINTLVL_HI_gc = (0x03<<2) ; High level

; Interrupt level
.equ DMA_CH_TRNINTLVL_OFF_gc = (0x00<<0) ; Interrupt disabled
.equ DMA_CH_TRNINTLVL_LO_gc = (0x01<<0) ; Low level
.equ DMA_CH_TRNINTLVL_MED_gc = (0x02<<0) ; Medium level
```

```
3226  .equ DMA_CH_TRNINTLVL_HI_gc = (0x03<<0) ; High level
3227
3228
3229  ;**************************************************************************
3230  ;** EVSYS - Event System
3231  ;**************************************************************************/
3232
3233  ; EVSYS_CH0MUX masks
3234  .equ EVSYS_CHMUX_gm = 0xFF ; Event Channel 0 Multiplexer group mask
3235  .equ EVSYS_CHMUX_gp = 0 ; Event Channel 0 Multiplexer group position
3236  .equ EVSYS_CHMUX0_bm = (1<<0) ; Event Channel 0 Multiplexer bit 0 mask
3237  .equ EVSYS_CHMUX0_bp = 0 ; Event Channel 0 Multiplexer bit 0 position
3238  .equ EVSYS_CHMUX1_bm = (1<<1) ; Event Channel 0 Multiplexer bit 1 mask
3239  .equ EVSYS_CHMUX1_bp = 1 ; Event Channel 0 Multiplexer bit 1 position
3240  .equ EVSYS_CHMUX2_bm = (1<<2) ; Event Channel 0 Multiplexer bit 2 mask
3241  .equ EVSYS_CHMUX2_bp = 2 ; Event Channel 0 Multiplexer bit 2 position
3242  .equ EVSYS_CHMUX3_bm = (1<<3) ; Event Channel 0 Multiplexer bit 3 mask
3243  .equ EVSYS_CHMUX3_bp = 3 ; Event Channel 0 Multiplexer bit 3 position
3244  .equ EVSYS_CHMUX4_bm = (1<<4) ; Event Channel 0 Multiplexer bit 4 mask
3245  .equ EVSYS_CHMUX4_bp = 4 ; Event Channel 0 Multiplexer bit 4 position
3246  .equ EVSYS_CHMUX5_bm = (1<<5) ; Event Channel 0 Multiplexer bit 5 mask
3247  .equ EVSYS_CHMUX5_bp = 5 ; Event Channel 0 Multiplexer bit 5 position
3248  .equ EVSYS_CHMUX6_bm = (1<<6) ; Event Channel 0 Multiplexer bit 6 mask
3249  .equ EVSYS_CHMUX6_bp = 6 ; Event Channel 0 Multiplexer bit 6 position
3250  .equ EVSYS_CHMUX7_bm = (1<<7) ; Event Channel 0 Multiplexer bit 7 mask
3251  .equ EVSYS_CHMUX7_bp = 7 ; Event Channel 0 Multiplexer bit 7 position
3252
3253  ; EVSYS_CH1MUX masks
3254  ; Masks for CHMUX aready defined
3255
3256  ; EVSYS_CH2MUX masks
3257  ; Masks for CHMUX aready defined
3258
3259  ; EVSYS_CH3MUX masks
3260  ; Masks for CHMUX aready defined
3261
3262  ; EVSYS_CH4MUX masks
3263  ; Masks for CHMUX aready defined
3264
3265  ; EVSYS_CH5MUX masks
3266  ; Masks for CHMUX aready defined
3267
3268  ; EVSYS_CH6MUX masks
3269  ; Masks for CHMUX aready defined
3270
3271  ; EVSYS_CH7MUX masks
3272  ; Masks for CHMUX aready defined
3273
3274  ; EVSYS_CH0CTRL masks
3275  .equ EVSYS_QDIRM_gm = 0x60 ; Quadrature Decoder Index Recognition Mode group mask
3276  .equ EVSYS_QDIRM_gp = 5 ; Quadrature Decoder Index Recognition Mode group position
3277  .equ EVSYS_QDIRM0_bm = (1<<5) ; Quadrature Decoder Index Recognition Mode bit 0 mask
3278  .equ EVSYS_QDIRM0_bp = 5 ; Quadrature Decoder Index Recognition Mode bit 0 position
3279  .equ EVSYS_QDIRM1_bm = (1<<6) ; Quadrature Decoder Index Recognition Mode bit 1 mask
3280  .equ EVSYS_QDIRM1_bp = 6 ; Quadrature Decoder Index Recognition Mode bit 1 position
3281  .equ EVSYS_QDIEN_bm = 0x10 ; Quadrature Decoder Index Enable bit mask
3282  .equ EVSYS_QDIEN_bp = 4 ; Quadrature Decoder Index Enable bit position
3283  .equ EVSYS_QDEN_bm = 0x08 ; Quadrature Decoder Enable bit mask
3284  .equ EVSYS_QDEN_bp = 3 ; Quadrature Decoder Enable bit position
3285  .equ EVSYS_DIGFILT_gm = 0x07 ; Digital Filter group mask
3286  .equ EVSYS_DIGFILT_gp = 0 ; Digital Filter group position
3287  .equ EVSYS_DIGFILT0_bm = (1<<0) ; Digital Filter bit 0 mask
3288  .equ EVSYS_DIGFILT0_bp = 0 ; Digital Filter bit 0 position
3289  .equ EVSYS_DIGFILT1_bm = (1<<1) ; Digital Filter bit 1 mask
3290  .equ EVSYS_DIGFILT1_bp = 1 ; Digital Filter bit 1 position
3291  .equ EVSYS_DIGFILT2_bm = (1<<2) ; Digital Filter bit 2 mask
```

```
3292  .equ EVSYS_DIGFILT2_bp = 2 ; Digital Filter bit 2 position
3293
3294  ; EVSYS_CH1CTRL masks
3295  ; Masks for DIGFILT aready defined
3296
3297  ; EVSYS_CH2CTRL masks
3298  ; Masks for QDIRM aready defined
3299  ; Masks for QDIEN aready defined
3300  ; Masks for QDEN aready defined
3301  ; Masks for DIGFILT aready defined
3302
3303  ; EVSYS_CH3CTRL masks
3304  ; Masks for DIGFILT aready defined
3305
3306  ; EVSYS_CH4CTRL masks
3307  ; Masks for QDIRM aready defined
3308  ; Masks for QDIEN aready defined
3309  ; Masks for QDEN aready defined
3310  ; Masks for DIGFILT aready defined
3311
3312  ; EVSYS_CH5CTRL masks
3313  ; Masks for DIGFILT aready defined
3314
3315  ; EVSYS_CH6CTRL masks
3316  ; Masks for DIGFILT aready defined
3317
3318  ; EVSYS_CH7CTRL masks
3319  ; Masks for DIGFILT aready defined
3320
3321  ; Quadrature Decoder Index Recognition Mode
3322  .equ EVSYS_QDIRM_00_gc = (0x00<<5) ; QDPH0 = 0, QDPH90 = 0
3323  .equ EVSYS_QDIRM_01_gc = (0x01<<5) ; QDPH0 = 0, QDPH90 = 1
3324  .equ EVSYS_QDIRM_10_gc = (0x02<<5) ; QDPH0 = 1, QDPH90 = 0
3325  .equ EVSYS_QDIRM_11_gc = (0x03<<5) ; QDPH0 = 1, QDPH90 = 1
3326
3327  ; Digital filter coefficient
3328  .equ EVSYS_DIGFILT_1SAMPLE_gc = (0x00<<0) ; 1 SAMPLE
3329  .equ EVSYS_DIGFILT_2SAMPLES_gc = (0x01<<0) ; 2 SAMPLES
3330  .equ EVSYS_DIGFILT_3SAMPLES_gc = (0x02<<0) ; 3 SAMPLES
3331  .equ EVSYS_DIGFILT_4SAMPLES_gc = (0x03<<0) ; 4 SAMPLES
3332  .equ EVSYS_DIGFILT_5SAMPLES_gc = (0x04<<0) ; 5 SAMPLES
3333  .equ EVSYS_DIGFILT_6SAMPLES_gc = (0x05<<0) ; 6 SAMPLES
3334  .equ EVSYS_DIGFILT_7SAMPLES_gc = (0x06<<0) ; 7 SAMPLES
3335  .equ EVSYS_DIGFILT_8SAMPLES_gc = (0x07<<0) ; 8 SAMPLES
3336
3337  ; Event Channel multiplexer input selection
3338  .equ EVSYS_CHMUX_OFF_gc = (0x00<<0) ; Off
3339  .equ EVSYS_CHMUX_RTC_OVF_gc = (0x08<<0) ; RTC Overflow
3340  .equ EVSYS_CHMUX_RTC_CMP_gc = (0x09<<0) ; RTC Compare Match
3341  .equ EVSYS_CHMUX_USB_gc = (0x0A<<0) ; USB Setup, SOF, CRC error and UNF/OVF
3342  .equ EVSYS_CHMUX_ACA_CH0_gc = (0x10<<0) ; Analog Comparator A Channel 0
3343  .equ EVSYS_CHMUX_ACA_CH1_gc = (0x11<<0) ; Analog Comparator A Channel 1
3344  .equ EVSYS_CHMUX_ACA_WIN_gc = (0x12<<0) ; Analog Comparator A Window
3345  .equ EVSYS_CHMUX_ACB_CH0_gc = (0x13<<0) ; Analog Comparator B Channel 0
3346  .equ EVSYS_CHMUX_ACB_CH1_gc = (0x14<<0) ; Analog Comparator B Channel 1
3347  .equ EVSYS_CHMUX_ACB_WIN_gc = (0x15<<0) ; Analog Comparator B Window
3348  .equ EVSYS_CHMUX_ADCA_CH0_gc = (0x20<<0) ; ADC A Channel 0
3349  .equ EVSYS_CHMUX_ADCA_CH1_gc = (0x21<<0) ; ADC A Channel 1
3350  .equ EVSYS_CHMUX_ADCA_CH2_gc = (0x22<<0) ; ADC A Channel 2
3351  .equ EVSYS_CHMUX_ADCA_CH3_gc = (0x23<<0) ; ADC A Channel 3
3352  .equ EVSYS_CHMUX_ADCB_CH0_gc = (0x24<<0) ; ADC B Channel 0
3353  .equ EVSYS_CHMUX_ADCB_CH1_gc = (0x25<<0) ; ADC B Channel 1
3354  .equ EVSYS_CHMUX_ADCB_CH2_gc = (0x26<<0) ; ADC B Channel 2
3355  .equ EVSYS_CHMUX_ADCB_CH3_gc = (0x27<<0) ; ADC B Channel 3
3356  .equ EVSYS_CHMUX_PORTA_PIN0_gc = (0x50<<0) ; Port A, Pin0
3357  .equ EVSYS_CHMUX_PORTA_PIN1_gc = (0x51<<0) ; Port A, Pin1
```

```
3358   .equ EVSYS_CHMUX_PORTA_PIN2_gc = (0x52<<0) ; Port A, Pin2
3359   .equ EVSYS_CHMUX_PORTA_PIN3_gc = (0x53<<0) ; Port A, Pin3
3360   .equ EVSYS_CHMUX_PORTA_PIN4_gc = (0x54<<0) ; Port A, Pin4
3361   .equ EVSYS_CHMUX_PORTA_PIN5_gc = (0x55<<0) ; Port A, Pin5
3362   .equ EVSYS_CHMUX_PORTA_PIN6_gc = (0x56<<0) ; Port A, Pin6
3363   .equ EVSYS_CHMUX_PORTA_PIN7_gc = (0x57<<0) ; Port A, Pin7
3364   .equ EVSYS_CHMUX_PORTB_PIN0_gc = (0x58<<0) ; Port B, Pin0
3365   .equ EVSYS_CHMUX_PORTB_PIN1_gc = (0x59<<0) ; Port B, Pin1
3366   .equ EVSYS_CHMUX_PORTB_PIN2_gc = (0x5A<<0) ; Port B, Pin2
3367   .equ EVSYS_CHMUX_PORTB_PIN3_gc = (0x5B<<0) ; Port B, Pin3
3368   .equ EVSYS_CHMUX_PORTB_PIN4_gc = (0x5C<<0) ; Port B, Pin4
3369   .equ EVSYS_CHMUX_PORTB_PIN5_gc = (0x5D<<0) ; Port B, Pin5
3370   .equ EVSYS_CHMUX_PORTB_PIN6_gc = (0x5E<<0) ; Port B, Pin6
3371   .equ EVSYS_CHMUX_PORTB_PIN7_gc = (0x5F<<0) ; Port B, Pin7
3372   .equ EVSYS_CHMUX_PORTC_PIN0_gc = (0x60<<0) ; Port C, Pin0
3373   .equ EVSYS_CHMUX_PORTC_PIN1_gc = (0x61<<0) ; Port C, Pin1
3374   .equ EVSYS_CHMUX_PORTC_PIN2_gc = (0x62<<0) ; Port C, Pin2
3375   .equ EVSYS_CHMUX_PORTC_PIN3_gc = (0x63<<0) ; Port C, Pin3
3376   .equ EVSYS_CHMUX_PORTC_PIN4_gc = (0x64<<0) ; Port C, Pin4
3377   .equ EVSYS_CHMUX_PORTC_PIN5_gc = (0x65<<0) ; Port C, Pin5
3378   .equ EVSYS_CHMUX_PORTC_PIN6_gc = (0x66<<0) ; Port C, Pin6
3379   .equ EVSYS_CHMUX_PORTC_PIN7_gc = (0x67<<0) ; Port C, Pin7
3380   .equ EVSYS_CHMUX_PORTD_PIN0_gc = (0x68<<0) ; Port D, Pin0
3381   .equ EVSYS_CHMUX_PORTD_PIN1_gc = (0x69<<0) ; Port D, Pin1
3382   .equ EVSYS_CHMUX_PORTD_PIN2_gc = (0x6A<<0) ; Port D, Pin2
3383   .equ EVSYS_CHMUX_PORTD_PIN3_gc = (0x6B<<0) ; Port D, Pin3
3384   .equ EVSYS_CHMUX_PORTD_PIN4_gc = (0x6C<<0) ; Port D, Pin4
3385   .equ EVSYS_CHMUX_PORTD_PIN5_gc = (0x6D<<0) ; Port D, Pin5
3386   .equ EVSYS_CHMUX_PORTD_PIN6_gc = (0x6E<<0) ; Port D, Pin6
3387   .equ EVSYS_CHMUX_PORTD_PIN7_gc = (0x6F<<0) ; Port D, Pin7
3388   .equ EVSYS_CHMUX_PORTE_PIN0_gc = (0x70<<0) ; Port E, Pin0
3389   .equ EVSYS_CHMUX_PORTE_PIN1_gc = (0x71<<0) ; Port E, Pin1
3390   .equ EVSYS_CHMUX_PORTE_PIN2_gc = (0x72<<0) ; Port E, Pin2
3391   .equ EVSYS_CHMUX_PORTE_PIN3_gc = (0x73<<0) ; Port E, Pin3
3392   .equ EVSYS_CHMUX_PORTE_PIN4_gc = (0x74<<0) ; Port E, Pin4
3393   .equ EVSYS_CHMUX_PORTE_PIN5_gc = (0x75<<0) ; Port E, Pin5
3394   .equ EVSYS_CHMUX_PORTE_PIN6_gc = (0x76<<0) ; Port E, Pin6
3395   .equ EVSYS_CHMUX_PORTE_PIN7_gc = (0x77<<0) ; Port E, Pin7
3396   .equ EVSYS_CHMUX_PORTF_PIN0_gc = (0x78<<0) ; Port F, Pin0
3397   .equ EVSYS_CHMUX_PORTF_PIN1_gc = (0x79<<0) ; Port F, Pin1
3398   .equ EVSYS_CHMUX_PORTF_PIN2_gc = (0x7A<<0) ; Port F, Pin2
3399   .equ EVSYS_CHMUX_PORTF_PIN3_gc = (0x7B<<0) ; Port F, Pin3
3400   .equ EVSYS_CHMUX_PORTF_PIN4_gc = (0x7C<<0) ; Port F, Pin4
3401   .equ EVSYS_CHMUX_PORTF_PIN5_gc = (0x7D<<0) ; Port F, Pin5
3402   .equ EVSYS_CHMUX_PORTF_PIN6_gc = (0x7E<<0) ; Port F, Pin6
3403   .equ EVSYS_CHMUX_PORTF_PIN7_gc = (0x7F<<0) ; Port F, Pin7
3404   .equ EVSYS_CHMUX_PRESCALER_1_gc = (0x80<<0) ; Prescaler, divide by 1
3405   .equ EVSYS_CHMUX_PRESCALER_2_gc = (0x81<<0) ; Prescaler, divide by 2
3406   .equ EVSYS_CHMUX_PRESCALER_4_gc = (0x82<<0) ; Prescaler, divide by 4
3407   .equ EVSYS_CHMUX_PRESCALER_8_gc = (0x83<<0) ; Prescaler, divide by 8
3408   .equ EVSYS_CHMUX_PRESCALER_16_gc = (0x84<<0) ; Prescaler, divide by 16
3409   .equ EVSYS_CHMUX_PRESCALER_32_gc = (0x85<<0) ; Prescaler, divide by 32
3410   .equ EVSYS_CHMUX_PRESCALER_64_gc = (0x86<<0) ; Prescaler, divide by 64
3411   .equ EVSYS_CHMUX_PRESCALER_128_gc = (0x87<<0) ; Prescaler, divide by 128
3412   .equ EVSYS_CHMUX_PRESCALER_256_gc = (0x88<<0) ; Prescaler, divide by 256
3413   .equ EVSYS_CHMUX_PRESCALER_512_gc = (0x89<<0) ; Prescaler, divide by 512
3414   .equ EVSYS_CHMUX_PRESCALER_1024_gc = (0x8A<<0) ; Prescaler, divide by 1024
3415   .equ EVSYS_CHMUX_PRESCALER_2048_gc = (0x8B<<0) ; Prescaler, divide by 2048
3416   .equ EVSYS_CHMUX_PRESCALER_4096_gc = (0x8C<<0) ; Prescaler, divide by 4096
3417   .equ EVSYS_CHMUX_PRESCALER_8192_gc = (0x8D<<0) ; Prescaler, divide by 8192
3418   .equ EVSYS_CHMUX_PRESCALER_16384_gc = (0x8E<<0) ; Prescaler, divide by 16384
3419   .equ EVSYS_CHMUX_PRESCALER_32768_gc = (0x8F<<0) ; Prescaler, divide by 32768
3420   .equ EVSYS_CHMUX_TCC0_OVF_gc = (0xC0<<0) ; Timer/Counter C0 Overflow
3421   .equ EVSYS_CHMUX_TCC0_ERR_gc = (0xC1<<0) ; Timer/Counter C0 Error
3422   .equ EVSYS_CHMUX_TCC0_CCA_gc = (0xC4<<0) ; Timer/Counter C0 Compare or Capture A
3423   .equ EVSYS_CHMUX_TCC0_CCB_gc = (0xC5<<0) ; Timer/Counter C0 Compare or Capture B
```

```
.equ EVSYS_CHMUX_TCC0_CCC_gc = (0xC6<<0) ; Timer/Counter C0 Compare or Capture C
.equ EVSYS_CHMUX_TCC0_CCD_gc = (0xC7<<0) ; Timer/Counter C0 Compare or Capture D
.equ EVSYS_CHMUX_TCC1_OVF_gc = (0xC8<<0) ; Timer/Counter C1 Overflow
.equ EVSYS_CHMUX_TCC1_ERR_gc = (0xC9<<0) ; Timer/Counter C1 Error
.equ EVSYS_CHMUX_TCC1_CCA_gc = (0xCC<<0) ; Timer/Counter C1 Compare or Capture A
.equ EVSYS_CHMUX_TCC1_CCB_gc = (0xCD<<0) ; Timer/Counter C1 Compare or Capture B
.equ EVSYS_CHMUX_TCD0_OVF_gc = (0xD0<<0) ; Timer/Counter D0 Overflow
.equ EVSYS_CHMUX_TCD0_ERR_gc = (0xD1<<0) ; Timer/Counter D0 Error
.equ EVSYS_CHMUX_TCD0_CCA_gc = (0xD4<<0) ; Timer/Counter D0 Compare or Capture A
.equ EVSYS_CHMUX_TCD0_CCB_gc = (0xD5<<0) ; Timer/Counter D0 Compare or Capture B
.equ EVSYS_CHMUX_TCD0_CCC_gc = (0xD6<<0) ; Timer/Counter D0 Compare or Capture C
.equ EVSYS_CHMUX_TCD0_CCD_gc = (0xD7<<0) ; Timer/Counter D0 Compare or Capture D
.equ EVSYS_CHMUX_TCD1_OVF_gc = (0xD8<<0) ; Timer/Counter D1 Overflow
.equ EVSYS_CHMUX_TCD1_ERR_gc = (0xD9<<0) ; Timer/Counter D1 Error
.equ EVSYS_CHMUX_TCD1_CCA_gc = (0xDC<<0) ; Timer/Counter D1 Compare or Capture A
.equ EVSYS_CHMUX_TCD1_CCB_gc = (0xDD<<0) ; Timer/Counter D1 Compare or Capture B
.equ EVSYS_CHMUX_TCE0_OVF_gc = (0xE0<<0) ; Timer/Counter E0 Overflow
.equ EVSYS_CHMUX_TCE0_ERR_gc = (0xE1<<0) ; Timer/Counter E0 Error
.equ EVSYS_CHMUX_TCE0_CCA_gc = (0xE4<<0) ; Timer/Counter E0 Compare or Capture A
.equ EVSYS_CHMUX_TCE0_CCB_gc = (0xE5<<0) ; Timer/Counter E0 Compare or Capture B
.equ EVSYS_CHMUX_TCE0_CCC_gc = (0xE6<<0) ; Timer/Counter E0 Compare or Capture C
.equ EVSYS_CHMUX_TCE0_CCD_gc = (0xE7<<0) ; Timer/Counter E0 Compare or Capture D
.equ EVSYS_CHMUX_TCE1_OVF_gc = (0xE8<<0) ; Timer/Counter E1 Overflow
.equ EVSYS_CHMUX_TCE1_ERR_gc = (0xE9<<0) ; Timer/Counter E1 Error
.equ EVSYS_CHMUX_TCE1_CCA_gc = (0xEC<<0) ; Timer/Counter E1 Compare or Capture A
.equ EVSYS_CHMUX_TCE1_CCB_gc = (0xED<<0) ; Timer/Counter E1 Compare or Capture B
.equ EVSYS_CHMUX_TCF0_OVF_gc = (0xF0<<0) ; Timer/Counter F0 Overflow
.equ EVSYS_CHMUX_TCF0_ERR_gc = (0xF1<<0) ; Timer/Counter F0 Error
.equ EVSYS_CHMUX_TCF0_CCA_gc = (0xF4<<0) ; Timer/Counter F0 Compare or Capture A
.equ EVSYS_CHMUX_TCF0_CCB_gc = (0xF5<<0) ; Timer/Counter F0 Compare or Capture B
.equ EVSYS_CHMUX_TCF0_CCC_gc = (0xF6<<0) ; Timer/Counter F0 Compare or Capture C
.equ EVSYS_CHMUX_TCF0_CCD_gc = (0xF7<<0) ; Timer/Counter F0 Compare or Capture D
.equ EVSYS_CHMUX_TCF1_OVF_gc = (0xF8<<0) ; Timer/Counter F1 Overflow
.equ EVSYS_CHMUX_TCF1_ERR_gc = (0xF9<<0) ; Timer/Counter F1 Error
.equ EVSYS_CHMUX_TCF1_CCA_gc = (0xFC<<0) ; Timer/Counter F1 Compare or Capture A
.equ EVSYS_CHMUX_TCF1_CCB_gc = (0xFD<<0) ; Timer/Counter F1 Compare or Capture B


;**************************************************************************
;** NVM - Non Volatile Memory Controller
;**************************************************************************/

; NVM_CMD masks
.equ NVM_CMD_gm = 0x7F ; Command group mask
.equ NVM_CMD_gp = 0 ; Command group position
.equ NVM_CMD0_bm = (1<<0) ; Command bit 0 mask
.equ NVM_CMD0_bp = 0 ; Command bit 0 position
.equ NVM_CMD1_bm = (1<<1) ; Command bit 1 mask
.equ NVM_CMD1_bp = 1 ; Command bit 1 position
.equ NVM_CMD2_bm = (1<<2) ; Command bit 2 mask
.equ NVM_CMD2_bp = 2 ; Command bit 2 position
.equ NVM_CMD3_bm = (1<<3) ; Command bit 3 mask
.equ NVM_CMD3_bp = 3 ; Command bit 3 position
.equ NVM_CMD4_bm = (1<<4) ; Command bit 4 mask
.equ NVM_CMD4_bp = 4 ; Command bit 4 position
.equ NVM_CMD5_bm = (1<<5) ; Command bit 5 mask
.equ NVM_CMD5_bp = 5 ; Command bit 5 position
.equ NVM_CMD6_bm = (1<<6) ; Command bit 6 mask
.equ NVM_CMD6_bp = 6 ; Command bit 6 position

; NVM_CTRLA masks
.equ NVM_CMDEX_bm = 0x01 ; Command Execute bit mask
.equ NVM_CMDEX_bp = 0 ; Command Execute bit position

; NVM_CTRLB masks
.equ NVM_EEMAPEN_bm = 0x08 ; EEPROM Mapping Enable bit mask
```

```
.equ NVM_EEMAPEN_bp = 3 ; EEPROM Mapping Enable bit position
.equ NVM_FPRM_bm = 0x04 ; Flash Power Reduction Enable bit mask
.equ NVM_FPRM_bp = 2 ; Flash Power Reduction Enable bit position
.equ NVM_EPRM_bm = 0x02 ; EEPROM Power Reduction Enable bit mask
.equ NVM_EPRM_bp = 1 ; EEPROM Power Reduction Enable bit position
.equ NVM_SPMLOCK_bm = 0x01 ; SPM Lock bit mask
.equ NVM_SPMLOCK_bp = 0 ; SPM Lock bit position

; NVM_INTCTRL masks
.equ NVM_SPMLVL_gm = 0x0C ; SPM Interrupt Level group mask
.equ NVM_SPMLVL_gp = 2 ; SPM Interrupt Level group position
.equ NVM_SPMLVL0_bm = (1<<2) ; SPM Interrupt Level bit 0 mask
.equ NVM_SPMLVL0_bp = 2 ; SPM Interrupt Level bit 0 position
.equ NVM_SPMLVL1_bm = (1<<3) ; SPM Interrupt Level bit 1 mask
.equ NVM_SPMLVL1_bp = 3 ; SPM Interrupt Level bit 1 position
.equ NVM_EELVL_gm = 0x03 ; EEPROM Interrupt Level group mask
.equ NVM_EELVL_gp = 0 ; EEPROM Interrupt Level group position
.equ NVM_EELVL0_bm = (1<<0) ; EEPROM Interrupt Level bit 0 mask
.equ NVM_EELVL0_bp = 0 ; EEPROM Interrupt Level bit 0 position
.equ NVM_EELVL1_bm = (1<<1) ; EEPROM Interrupt Level bit 1 mask
.equ NVM_EELVL1_bp = 1 ; EEPROM Interrupt Level bit 1 position

; NVM_STATUS masks
.equ NVM_NVMBUSY_bm = 0x80 ; Non-volatile Memory Busy bit mask
.equ NVM_NVMBUSY_bp = 7 ; Non-volatile Memory Busy bit position
.equ NVM_FBUSY_bm = 0x40 ; Flash Memory Busy bit mask
.equ NVM_FBUSY_bp = 6 ; Flash Memory Busy bit position
.equ NVM_EELOAD_bm = 0x02 ; EEPROM Page Buffer Active Loading bit mask
.equ NVM_EELOAD_bp = 1 ; EEPROM Page Buffer Active Loading bit position
.equ NVM_FLOAD_bm = 0x01 ; Flash Page Buffer Active Loading bit mask
.equ NVM_FLOAD_bp = 0 ; Flash Page Buffer Active Loading bit position

; NVM_LOCKBITS masks
.equ NVM_BLBB_gm = 0xC0 ; Boot Lock Bits - Boot Section group mask
.equ NVM_BLBB_gp = 6 ; Boot Lock Bits - Boot Section group position
.equ NVM_BLBB0_bm = (1<<6) ; Boot Lock Bits - Boot Section bit 0 mask
.equ NVM_BLBB0_bp = 6 ; Boot Lock Bits - Boot Section bit 0 position
.equ NVM_BLBB1_bm = (1<<7) ; Boot Lock Bits - Boot Section bit 1 mask
.equ NVM_BLBB1_bp = 7 ; Boot Lock Bits - Boot Section bit 1 position
.equ NVM_BLBA_gm = 0x30 ; Boot Lock Bits - Application Section group mask
.equ NVM_BLBA_gp = 4 ; Boot Lock Bits - Application Section group position
.equ NVM_BLBA0_bm = (1<<4) ; Boot Lock Bits - Application Section bit 0 mask
.equ NVM_BLBA0_bp = 4 ; Boot Lock Bits - Application Section bit 0 position
.equ NVM_BLBA1_bm = (1<<5) ; Boot Lock Bits - Application Section bit 1 mask
.equ NVM_BLBA1_bp = 5 ; Boot Lock Bits - Application Section bit 1 position
.equ NVM_BLBAT_gm = 0x0C ; Boot Lock Bits - Application Table group mask
.equ NVM_BLBAT_gp = 2 ; Boot Lock Bits - Application Table group position
.equ NVM_BLBAT0_bm = (1<<2) ; Boot Lock Bits - Application Table bit 0 mask
.equ NVM_BLBAT0_bp = 2 ; Boot Lock Bits - Application Table bit 0 position
.equ NVM_BLBAT1_bm = (1<<3) ; Boot Lock Bits - Application Table bit 1 mask
.equ NVM_BLBAT1_bp = 3 ; Boot Lock Bits - Application Table bit 1 position
.equ NVM_LB_gm = 0x03 ; Lock Bits group mask
.equ NVM_LB_gp = 0 ; Lock Bits group position
.equ NVM_LB0_bm = (1<<0) ; Lock Bits bit 0 mask
.equ NVM_LB0_bp = 0 ; Lock Bits bit 0 position
.equ NVM_LB1_bm = (1<<1) ; Lock Bits bit 1 mask
.equ NVM_LB1_bp = 1 ; Lock Bits bit 1 position

; NVM Command
.equ NVM_CMD_NO_OPERATION_gc = (0x00<<0) ; Noop/Ordinary LPM
.equ NVM_CMD_READ_USER_SIG_ROW_gc = (0x01<<0) ; Read user signature row
.equ NVM_CMD_READ_CALIB_ROW_gc = (0x02<<0) ; Read calibration row
.equ NVM_CMD_READ_EEPROM_gc = (0x06<<0) ; Read EEPROM
.equ NVM_CMD_READ_FUSES_gc = (0x07<<0) ; Read fuse byte
.equ NVM_CMD_WRITE_LOCK_BITS_gc = (0x08<<0) ; Write lock bits
.equ NVM_CMD_ERASE_USER_SIG_ROW_gc = (0x18<<0) ; Erase user signature row
```

```
3556   .equ NVM_CMD_WRITE_USER_SIG_ROW_gc = (0x1A<<0) ; Write user signature row
3557   .equ NVM_CMD_ERASE_APP_gc = (0x20<<0) ; Erase Application Section
3558   .equ NVM_CMD_ERASE_APP_PAGE_gc = (0x22<<0) ; Erase Application Section page
3559   .equ NVM_CMD_LOAD_FLASH_BUFFER_gc = (0x23<<0) ; Load Flash page buffer
3560   .equ NVM_CMD_WRITE_APP_PAGE_gc = (0x24<<0) ; Write Application Section page
3561   .equ NVM_CMD_ERASE_WRITE_APP_PAGE_gc = (0x25<<0) ; Erase-and-write Application Section
      page
3562   .equ NVM_CMD_ERASE_FLASH_BUFFER_gc = (0x26<<0) ; Erase/flush Flash page buffer
3563   .equ NVM_CMD_ERASE_BOOT_PAGE_gc = (0x2A<<0) ; Erase Boot Section page
3564   .equ NVM_CMD_ERASE_FLASH_PAGE_gc = (0x2B<<0) ; Erase Flash Page
3565   .equ NVM_CMD_WRITE_BOOT_PAGE_gc = (0x2C<<0) ; Write Boot Section page
3566   .equ NVM_CMD_ERASE_WRITE_BOOT_PAGE_gc = (0x2D<<0) ; Erase-and-write Boot Section page
3567   .equ NVM_CMD_WRITE_FLASH_PAGE_gc = (0x2E<<0) ; Write Flash Page
3568   .equ NVM_CMD_ERASE_WRITE_FLASH_PAGE_gc = (0x2F<<0) ; Erase-and-write Flash Page
3569   .equ NVM_CMD_ERASE_EEPROM_gc = (0x30<<0) ; Erase EEPROM
3570   .equ NVM_CMD_ERASE_EEPROM_PAGE_gc = (0x32<<0) ; Erase EEPROM page
3571   .equ NVM_CMD_LOAD_EEPROM_BUFFER_gc = (0x33<<0) ; Load EEPROM page buffer
3572   .equ NVM_CMD_WRITE_EEPROM_PAGE_gc = (0x34<<0) ; Write EEPROM page
3573   .equ NVM_CMD_ERASE_WRITE_EEPROM_PAGE_gc = (0x35<<0) ; Erase-and-write EEPROM page
3574   .equ NVM_CMD_ERASE_EEPROM_BUFFER_gc = (0x36<<0) ; Erase/flush EEPROM page buffer
3575   .equ NVM_CMD_APP_CRC_gc = (0x38<<0) ; Application section CRC
3576   .equ NVM_CMD_BOOT_CRC_gc = (0x39<<0) ;  Boot Section CRC
3577   .equ NVM_CMD_FLASH_RANGE_CRC_gc = (0x3A<<0) ; Flash Range CRC
3578   .equ NVM_CMD_CHIP_ERASE_gc = (0x40<<0) ; Erase Chip
3579   .equ NVM_CMD_READ_NVM_gc = (0x43<<0) ; Read NVM
3580   .equ NVM_CMD_WRITE_FUSE_gc = (0x4C<<0) ; Write Fuse byte
3581   .equ NVM_CMD_ERASE_BOOT_gc = (0x68<<0) ; Erase Boot Section
3582   .equ NVM_CMD_FLASH_CRC_gc = (0x78<<0) ; Flash CRC
3583
3584   ; SPM ready interrupt level
3585   .equ NVM_SPMLVL_OFF_gc = (0x00<<2) ; Interrupt disabled
3586   .equ NVM_SPMLVL_LO_gc = (0x01<<2) ; Low level
3587   .equ NVM_SPMLVL_MED_gc = (0x02<<2) ; Medium level
3588   .equ NVM_SPMLVL_HI_gc = (0x03<<2) ; High level
3589
3590   ; EEPROM ready interrupt level
3591   .equ NVM_EELVL_OFF_gc = (0x00<<0) ; Interrupt disabled
3592   .equ NVM_EELVL_LO_gc = (0x01<<0) ; Low level
3593   .equ NVM_EELVL_MED_gc = (0x02<<0) ; Medium level
3594   .equ NVM_EELVL_HI_gc = (0x03<<0) ; High level
3595
3596   ; Boot lock bits - boot setcion
3597   .equ NVM_BLBB_RWLOCK_gc = (0x00<<6) ; Read and write not allowed
3598   .equ NVM_BLBB_RLOCK_gc = (0x01<<6) ; Read not allowed
3599   .equ NVM_BLBB_WLOCK_gc = (0x02<<6) ; Write not allowed
3600   .equ NVM_BLBB_NOLOCK_gc = (0x03<<6) ; No locks
3601
3602   ; Boot lock bits - application section
3603   .equ NVM_BLBA_RWLOCK_gc = (0x00<<4) ; Read and write not allowed
3604   .equ NVM_BLBA_RLOCK_gc = (0x01<<4) ; Read not allowed
3605   .equ NVM_BLBA_WLOCK_gc = (0x02<<4) ; Write not allowed
3606   .equ NVM_BLBA_NOLOCK_gc = (0x03<<4) ; No locks
3607
3608   ; Boot lock bits - application table section
3609   .equ NVM_BLBAT_RWLOCK_gc = (0x00<<2) ; Read and write not allowed
3610   .equ NVM_BLBAT_RLOCK_gc = (0x01<<2) ; Read not allowed
3611   .equ NVM_BLBAT_WLOCK_gc = (0x02<<2) ; Write not allowed
3612   .equ NVM_BLBAT_NOLOCK_gc = (0x03<<2) ; No locks
3613
3614   ; Lock bits
3615   .equ NVM_LB_RWLOCK_gc = (0x00<<0) ; Read and write not allowed
3616   .equ NVM_LB_WLOCK_gc = (0x02<<0) ; Write not allowed
3617   .equ NVM_LB_NOLOCK_gc = (0x03<<0) ; No locks
3618
3619
3620   ;**************************************************************************
```

```
;** ADC - Analog/Digital Converter
;**************************************************************************/

; ADC_CH_CTRL masks
.equ ADC_CH_START_bm = 0x80 ; Channel Start Conversion bit mask
.equ ADC_CH_START_bp = 7 ; Channel Start Conversion bit position
.equ ADC_CH_GAIN_gm = 0x1C ; Gain Factor group mask
.equ ADC_CH_GAIN_gp = 2 ; Gain Factor group position
.equ ADC_CH_GAIN0_bm = (1<<2) ; Gain Factor bit 0 mask
.equ ADC_CH_GAIN0_bp = 2 ; Gain Factor bit 0 position
.equ ADC_CH_GAIN1_bm = (1<<3) ; Gain Factor bit 1 mask
.equ ADC_CH_GAIN1_bp = 3 ; Gain Factor bit 1 position
.equ ADC_CH_GAIN2_bm = (1<<4) ; Gain Factor bit 2 mask
.equ ADC_CH_GAIN2_bp = 4 ; Gain Factor bit 2 position
.equ ADC_CH_INPUTMODE_gm = 0x03 ; Input Mode Select group mask
.equ ADC_CH_INPUTMODE_gp = 0 ; Input Mode Select group position
.equ ADC_CH_INPUTMODE0_bm = (1<<0) ; Input Mode Select bit 0 mask
.equ ADC_CH_INPUTMODE0_bp = 0 ; Input Mode Select bit 0 position
.equ ADC_CH_INPUTMODE1_bm = (1<<1) ; Input Mode Select bit 1 mask
.equ ADC_CH_INPUTMODE1_bp = 1 ; Input Mode Select bit 1 position

; ADC_CH_MUXCTRL masks
.equ ADC_CH_MUXPOS_gm = 0x78 ; MUX selection on Positive ADC input group mask
.equ ADC_CH_MUXPOS_gp = 3 ; MUX selection on Positive ADC input group position
.equ ADC_CH_MUXPOS0_bm = (1<<3) ; MUX selection on Positive ADC input bit 0 mask
.equ ADC_CH_MUXPOS0_bp = 3 ; MUX selection on Positive ADC input bit 0 position
.equ ADC_CH_MUXPOS1_bm = (1<<4) ; MUX selection on Positive ADC input bit 1 mask
.equ ADC_CH_MUXPOS1_bp = 4 ; MUX selection on Positive ADC input bit 1 position
.equ ADC_CH_MUXPOS2_bm = (1<<5) ; MUX selection on Positive ADC input bit 2 mask
.equ ADC_CH_MUXPOS2_bp = 5 ; MUX selection on Positive ADC input bit 2 position
.equ ADC_CH_MUXPOS3_bm = (1<<6) ; MUX selection on Positive ADC input bit 3 mask
.equ ADC_CH_MUXPOS3_bp = 6 ; MUX selection on Positive ADC input bit 3 position
.equ ADC_CH_MUXINT_gm = 0x78 ; MUX selection on Internal ADC input group mask
.equ ADC_CH_MUXINT_gp = 3 ; MUX selection on Internal ADC input group position
.equ ADC_CH_MUXINT0_bm = (1<<3) ; MUX selection on Internal ADC input bit 0 mask
.equ ADC_CH_MUXINT0_bp = 3 ; MUX selection on Internal ADC input bit 0 position
.equ ADC_CH_MUXINT1_bm = (1<<4) ; MUX selection on Internal ADC input bit 1 mask
.equ ADC_CH_MUXINT1_bp = 4 ; MUX selection on Internal ADC input bit 1 position
.equ ADC_CH_MUXINT2_bm = (1<<5) ; MUX selection on Internal ADC input bit 2 mask
.equ ADC_CH_MUXINT2_bp = 5 ; MUX selection on Internal ADC input bit 2 position
.equ ADC_CH_MUXINT3_bm = (1<<6) ; MUX selection on Internal ADC input bit 3 mask
.equ ADC_CH_MUXINT3_bp = 6 ; MUX selection on Internal ADC input bit 3 position
.equ ADC_CH_MUXNEG_gm = 0x03 ; MUX selection on Negative ADC input group mask
.equ ADC_CH_MUXNEG_gp = 0 ; MUX selection on Negative ADC input group position
.equ ADC_CH_MUXNEG0_bm = (1<<0) ; MUX selection on Negative ADC input bit 0 mask
.equ ADC_CH_MUXNEG0_bp = 0 ; MUX selection on Negative ADC input bit 0 position
.equ ADC_CH_MUXNEG1_bm = (1<<1) ; MUX selection on Negative ADC input bit 1 mask
.equ ADC_CH_MUXNEG1_bp = 1 ; MUX selection on Negative ADC input bit 1 position

; ADC_CH_INTCTRL masks
.equ ADC_CH_INTMODE_gm = 0x0C ; Interrupt Mode group mask
.equ ADC_CH_INTMODE_gp = 2 ; Interrupt Mode group position
.equ ADC_CH_INTMODE0_bm = (1<<2) ; Interrupt Mode bit 0 mask
.equ ADC_CH_INTMODE0_bp = 2 ; Interrupt Mode bit 0 position
.equ ADC_CH_INTMODE1_bm = (1<<3) ; Interrupt Mode bit 1 mask
.equ ADC_CH_INTMODE1_bp = 3 ; Interrupt Mode bit 1 position
.equ ADC_CH_INTLVL_gm = 0x03 ; Interrupt Level group mask
.equ ADC_CH_INTLVL_gp = 0 ; Interrupt Level group position
.equ ADC_CH_INTLVL0_bm = (1<<0) ; Interrupt Level bit 0 mask
.equ ADC_CH_INTLVL0_bp = 0 ; Interrupt Level bit 0 position
.equ ADC_CH_INTLVL1_bm = (1<<1) ; Interrupt Level bit 1 mask
.equ ADC_CH_INTLVL1_bp = 1 ; Interrupt Level bit 1 position

; ADC_CH_INTFLAGS masks
.equ ADC_CH_CHIF_bm = 0x01 ; Channel Interrupt Flag bit mask
.equ ADC_CH_CHIF_bp = 0 ; Channel Interrupt Flag bit position
```

```
3687
3688  ; ADC_CH_SCAN masks
3689  .equ ADC_CH_OFFSET_gm = 0xF0 ; Positive MUX setting offset group mask
3690  .equ ADC_CH_OFFSET_gp = 4 ; Positive MUX setting offset group position
3691  .equ ADC_CH_OFFSET0_bm = (1<<4) ; Positive MUX setting offset bit 0 mask
3692  .equ ADC_CH_OFFSET0_bp = 4 ; Positive MUX setting offset bit 0 position
3693  .equ ADC_CH_OFFSET1_bm = (1<<5) ; Positive MUX setting offset bit 1 mask
3694  .equ ADC_CH_OFFSET1_bp = 5 ; Positive MUX setting offset bit 1 position
3695  .equ ADC_CH_OFFSET2_bm = (1<<6) ; Positive MUX setting offset bit 2 mask
3696  .equ ADC_CH_OFFSET2_bp = 6 ; Positive MUX setting offset bit 2 position
3697  .equ ADC_CH_OFFSET3_bm = (1<<7) ; Positive MUX setting offset bit 3 mask
3698  .equ ADC_CH_OFFSET3_bp = 7 ; Positive MUX setting offset bit 3 position
3699  .equ ADC_CH_SCANNUM_gm = 0x0F ; Number of Channels included in scan group mask
3700  .equ ADC_CH_SCANNUM_gp = 0 ; Number of Channels included in scan group position
3701  .equ ADC_CH_SCANNUM0_bm = (1<<0) ; Number of Channels included in scan bit 0 mask
3702  .equ ADC_CH_SCANNUM0_bp = 0 ; Number of Channels included in scan bit 0 position
3703  .equ ADC_CH_SCANNUM1_bm = (1<<1) ; Number of Channels included in scan bit 1 mask
3704  .equ ADC_CH_SCANNUM1_bp = 1 ; Number of Channels included in scan bit 1 position
3705  .equ ADC_CH_SCANNUM2_bm = (1<<2) ; Number of Channels included in scan bit 2 mask
3706  .equ ADC_CH_SCANNUM2_bp = 2 ; Number of Channels included in scan bit 2 position
3707  .equ ADC_CH_SCANNUM3_bm = (1<<3) ; Number of Channels included in scan bit 3 mask
3708  .equ ADC_CH_SCANNUM3_bp = 3 ; Number of Channels included in scan bit 3 position
3709
3710  ; ADC_CTRLA masks
3711  .equ ADC_DMASEL_gm = 0xC0 ; DMA Selection group mask
3712  .equ ADC_DMASEL_gp = 6 ; DMA Selection group position
3713  .equ ADC_DMASEL0_bm = (1<<6) ; DMA Selection bit 0 mask
3714  .equ ADC_DMASEL0_bp = 6 ; DMA Selection bit 0 position
3715  .equ ADC_DMASEL1_bm = (1<<7) ; DMA Selection bit 1 mask
3716  .equ ADC_DMASEL1_bp = 7 ; DMA Selection bit 1 position
3717  .equ ADC_CH3START_bm = 0x20 ; Channel 3 Start Conversion bit mask
3718  .equ ADC_CH3START_bp = 5 ; Channel 3 Start Conversion bit position
3719  .equ ADC_CH2START_bm = 0x10 ; Channel 2 Start Conversion bit mask
3720  .equ ADC_CH2START_bp = 4 ; Channel 2 Start Conversion bit position
3721  .equ ADC_CH1START_bm = 0x08 ; Channel 1 Start Conversion bit mask
3722  .equ ADC_CH1START_bp = 3 ; Channel 1 Start Conversion bit position
3723  .equ ADC_CH0START_bm = 0x04 ; Channel 0 Start Conversion bit mask
3724  .equ ADC_CH0START_bp = 2 ; Channel 0 Start Conversion bit position
3725  .equ ADC_FLUSH_bm = 0x02 ; Flush Pipeline bit mask
3726  .equ ADC_FLUSH_bp = 1 ; Flush Pipeline bit position
3727  .equ ADC_ENABLE_bm = 0x01 ; Enable ADC bit mask
3728  .equ ADC_ENABLE_bp = 0 ; Enable ADC bit position
3729
3730  ; ADC_CTRLB masks
3731  .equ ADC_IMPMODE_bm = 0x80 ; Gain Stage Impedance Mode bit mask
3732  .equ ADC_IMPMODE_bp = 7 ; Gain Stage Impedance Mode bit position
3733  .equ ADC_CURRLIMIT_gm = 0x60 ; Current Limitation group mask
3734  .equ ADC_CURRLIMIT_gp = 5 ; Current Limitation group position
3735  .equ ADC_CURRLIMIT0_bm = (1<<5) ; Current Limitation bit 0 mask
3736  .equ ADC_CURRLIMIT0_bp = 5 ; Current Limitation bit 0 position
3737  .equ ADC_CURRLIMIT1_bm = (1<<6) ; Current Limitation bit 1 mask
3738  .equ ADC_CURRLIMIT1_bp = 6 ; Current Limitation bit 1 position
3739  .equ ADC_CONMODE_bm = 0x10 ; Conversion Mode bit mask
3740  .equ ADC_CONMODE_bp = 4 ; Conversion Mode bit position
3741  .equ ADC_FREERUN_bm = 0x08 ; Free Running Mode Enable bit mask
3742  .equ ADC_FREERUN_bp = 3 ; Free Running Mode Enable bit position
3743  .equ ADC_RESOLUTION_gm = 0x06 ; Result Resolution group mask
3744  .equ ADC_RESOLUTION_gp = 1 ; Result Resolution group position
3745  .equ ADC_RESOLUTION0_bm = (1<<1) ; Result Resolution bit 0 mask
3746  .equ ADC_RESOLUTION0_bp = 1 ; Result Resolution bit 0 position
3747  .equ ADC_RESOLUTION1_bm = (1<<2) ; Result Resolution bit 1 mask
3748  .equ ADC_RESOLUTION1_bp = 2 ; Result Resolution bit 1 position
3749
3750  ; ADC_REFCTRL masks
3751  .equ ADC_REFSEL_gm = 0x70 ; Reference Selection group mask
3752  .equ ADC_REFSEL_gp = 4 ; Reference Selection group position
```

```
3753   .equ ADC_REFSEL0_bm = (1<<4) ; Reference Selection bit 0 mask
3754   .equ ADC_REFSEL0_bp = 4 ; Reference Selection bit 0 position
3755   .equ ADC_REFSEL1_bm = (1<<5) ; Reference Selection bit 1 mask
3756   .equ ADC_REFSEL1_bp = 5 ; Reference Selection bit 1 position
3757   .equ ADC_REFSEL2_bm = (1<<6) ; Reference Selection bit 2 mask
3758   .equ ADC_REFSEL2_bp = 6 ; Reference Selection bit 2 position
3759   .equ ADC_BANDGAP_bm = 0x02 ; Bandgap enable bit mask
3760   .equ ADC_BANDGAP_bp = 1 ; Bandgap enable bit position
3761   .equ ADC_TEMPREF_bm = 0x01 ; Temperature Reference Enable bit mask
3762   .equ ADC_TEMPREF_bp = 0 ; Temperature Reference Enable bit position
3763
3764   ; ADC_EVCTRL masks
3765   .equ ADC_SWEEP_gm = 0xC0 ; Channel Sweep Selection group mask
3766   .equ ADC_SWEEP_gp = 6 ; Channel Sweep Selection group position
3767   .equ ADC_SWEEP0_bm = (1<<6) ; Channel Sweep Selection bit 0 mask
3768   .equ ADC_SWEEP0_bp = 6 ; Channel Sweep Selection bit 0 position
3769   .equ ADC_SWEEP1_bm = (1<<7) ; Channel Sweep Selection bit 1 mask
3770   .equ ADC_SWEEP1_bp = 7 ; Channel Sweep Selection bit 1 position
3771   .equ ADC_EVSEL_gm = 0x38 ; Event Input Select group mask
3772   .equ ADC_EVSEL_gp = 3 ; Event Input Select group position
3773   .equ ADC_EVSEL0_bm = (1<<3) ; Event Input Select bit 0 mask
3774   .equ ADC_EVSEL0_bp = 3 ; Event Input Select bit 0 position
3775   .equ ADC_EVSEL1_bm = (1<<4) ; Event Input Select bit 1 mask
3776   .equ ADC_EVSEL1_bp = 4 ; Event Input Select bit 1 position
3777   .equ ADC_EVSEL2_bm = (1<<5) ; Event Input Select bit 2 mask
3778   .equ ADC_EVSEL2_bp = 5 ; Event Input Select bit 2 position
3779   .equ ADC_EVACT_gm = 0x07 ; Event Action Select group mask
3780   .equ ADC_EVACT_gp = 0 ; Event Action Select group position
3781   .equ ADC_EVACT0_bm = (1<<0) ; Event Action Select bit 0 mask
3782   .equ ADC_EVACT0_bp = 0 ; Event Action Select bit 0 position
3783   .equ ADC_EVACT1_bm = (1<<1) ; Event Action Select bit 1 mask
3784   .equ ADC_EVACT1_bp = 1 ; Event Action Select bit 1 position
3785   .equ ADC_EVACT2_bm = (1<<2) ; Event Action Select bit 2 mask
3786   .equ ADC_EVACT2_bp = 2 ; Event Action Select bit 2 position
3787
3788   ; ADC_PRESCALER masks
3789   .equ ADC_PRESCALER_gm = 0x07 ; Clock Prescaler Selection group mask
3790   .equ ADC_PRESCALER_gp = 0 ; Clock Prescaler Selection group position
3791   .equ ADC_PRESCALER0_bm = (1<<0) ; Clock Prescaler Selection bit 0 mask
3792   .equ ADC_PRESCALER0_bp = 0 ; Clock Prescaler Selection bit 0 position
3793   .equ ADC_PRESCALER1_bm = (1<<1) ; Clock Prescaler Selection bit 1 mask
3794   .equ ADC_PRESCALER1_bp = 1 ; Clock Prescaler Selection bit 1 position
3795   .equ ADC_PRESCALER2_bm = (1<<2) ; Clock Prescaler Selection bit 2 mask
3796   .equ ADC_PRESCALER2_bp = 2 ; Clock Prescaler Selection bit 2 position
3797
3798   ; ADC_INTFLAGS masks
3799   .equ ADC_CH3IF_bm = 0x08 ; Channel 3 Interrupt Flag bit mask
3800   .equ ADC_CH3IF_bp = 3 ; Channel 3 Interrupt Flag bit position
3801   .equ ADC_CH2IF_bm = 0x04 ; Channel 2 Interrupt Flag bit mask
3802   .equ ADC_CH2IF_bp = 2 ; Channel 2 Interrupt Flag bit position
3803   .equ ADC_CH1IF_bm = 0x02 ; Channel 1 Interrupt Flag bit mask
3804   .equ ADC_CH1IF_bp = 1 ; Channel 1 Interrupt Flag bit position
3805   .equ ADC_CH0IF_bm = 0x01 ; Channel 0 Interrupt Flag bit mask
3806   .equ ADC_CH0IF_bp = 0 ; Channel 0 Interrupt Flag bit position
3807
3808   ; Positive input multiplexer selection
3809   .equ ADC_CH_MUXPOS_PIN0_gc = (0x00<<3) ; Input pin 0
3810   .equ ADC_CH_MUXPOS_PIN1_gc = (0x01<<3) ; Input pin 1
3811   .equ ADC_CH_MUXPOS_PIN2_gc = (0x02<<3) ; Input pin 2
3812   .equ ADC_CH_MUXPOS_PIN3_gc = (0x03<<3) ; Input pin 3
3813   .equ ADC_CH_MUXPOS_PIN4_gc = (0x04<<3) ; Input pin 4
3814   .equ ADC_CH_MUXPOS_PIN5_gc = (0x05<<3) ; Input pin 5
3815   .equ ADC_CH_MUXPOS_PIN6_gc = (0x06<<3) ; Input pin 6
3816   .equ ADC_CH_MUXPOS_PIN7_gc = (0x07<<3) ; Input pin 7
3817   .equ ADC_CH_MUXPOS_PIN8_gc = (0x08<<3) ; Input pin 8
3818   .equ ADC_CH_MUXPOS_PIN9_gc = (0x09<<3) ; Input pin 9
```

```
3819   .equ ADC_CH_MUXPOS_PIN10_gc = (0x0A<<3) ; Input pin 10
3820   .equ ADC_CH_MUXPOS_PIN11_gc = (0x0B<<3) ; Input pin 11
3821   .equ ADC_CH_MUXPOS_PIN12_gc = (0x0C<<3) ; Input pin 12
3822   .equ ADC_CH_MUXPOS_PIN13_gc = (0x0D<<3) ; Input pin 13
3823   .equ ADC_CH_MUXPOS_PIN14_gc = (0x0E<<3) ; Input pin 14
3824   .equ ADC_CH_MUXPOS_PIN15_gc = (0x0F<<3) ; Input pin 15
3825
3826   ; Internal input multiplexer selections
3827   .equ ADC_CH_MUXINT_TEMP_gc = (0x00<<3) ; Temperature Reference
3828   .equ ADC_CH_MUXINT_BANDGAP_gc = (0x01<<3) ; Bandgap Reference
3829   .equ ADC_CH_MUXINT_SCALEDVCC_gc = (0x02<<3) ; 1/10 scaled VCC
3830   .equ ADC_CH_MUXINT_DAC_gc = (0x03<<3) ; DAC output
3831
3832   ; Negative input multiplexer selection
3833   .equ ADC_CH_MUXNEG_PIN0_gc = (0x00<<0) ; Input pin 0
3834   .equ ADC_CH_MUXNEG_PIN1_gc = (0x01<<0) ; Input pin 1
3835   .equ ADC_CH_MUXNEG_PIN2_gc = (0x02<<0) ; Input pin 2
3836   .equ ADC_CH_MUXNEG_PIN3_gc = (0x03<<0) ; Input pin 3
3837   .equ ADC_CH_MUXNEG_PIN4_gc = (0x00<<0) ; Input pin 4
3838   .equ ADC_CH_MUXNEG_PIN5_gc = (0x01<<0) ; Input pin 5
3839   .equ ADC_CH_MUXNEG_PIN6_gc = (0x02<<0) ; Input pin 6
3840   .equ ADC_CH_MUXNEG_PIN7_gc = (0x03<<0) ; Input pin 7
3841
3842   ; Input mode
3843   .equ ADC_CH_INPUTMODE_INTERNAL_gc = (0x00<<0) ; Internal inputs, no gain
3844   .equ ADC_CH_INPUTMODE_SINGLEENDED_gc = (0x01<<0) ; Single-ended input, no gain
3845   .equ ADC_CH_INPUTMODE_DIFF_gc = (0x02<<0) ; Differential input, no gain
3846   .equ ADC_CH_INPUTMODE_DIFFWGAIN_gc = (0x03<<0) ; Differential input, with gain
3847
3848   ; Gain factor
3849   .equ ADC_CH_GAIN_1X_gc = (0x00<<2) ; 1x gain
3850   .equ ADC_CH_GAIN_2X_gc = (0x01<<2) ; 2x gain
3851   .equ ADC_CH_GAIN_4X_gc = (0x02<<2) ; 4x gain
3852   .equ ADC_CH_GAIN_8X_gc = (0x03<<2) ; 8x gain
3853   .equ ADC_CH_GAIN_16X_gc = (0x04<<2) ; 16x gain
3854   .equ ADC_CH_GAIN_32X_gc = (0x05<<2) ; 32x gain
3855   .equ ADC_CH_GAIN_64X_gc = (0x06<<2) ; 64x gain
3856   .equ ADC_CH_GAIN_DIV2_gc = (0x07<<2) ; x/2 gain
3857
3858   ; Conversion result resolution
3859   .equ ADC_RESOLUTION_12BIT_gc = (0x00<<1) ; 12-bit right-adjusted result
3860   .equ ADC_RESOLUTION_8BIT_gc = (0x02<<1) ; 8-bit right-adjusted result
3861   .equ ADC_RESOLUTION_LEFT12BIT_gc = (0x03<<1) ; 12-bit left-adjusted result
3862
3863   ; Current Limitation Mode
3864   .equ ADC_CURRLIMIT_NO_gc = (0x00<<5) ; No Current Reduction
3865   .equ ADC_CURRLIMIT_SMALL_gc = (0x01<<5) ; 10% current reduction
3866   .equ ADC_CURRLIMIT_MEDIUM_gc = (0x02<<5) ; 20% current reduction
3867   .equ ADC_CURRLIMIT_LARGE_gc = (0x03<<5) ; 30% current reduction
3868
3869   ; Voltage reference selection
3870   .equ ADC_REFSEL_INT1V_gc = (0x00<<4) ; Internal 1V
3871   .equ ADC_REFSEL_VCC_gc = (0x01<<4) ; Internal VCC / 1.6
3872   .equ ADC_REFSEL_AREFA_gc = (0x02<<4) ; External reference on PORT A
3873   .equ ADC_REFSEL_AREFB_gc = (0x03<<4) ; External reference on PORT B
3874   .equ ADC_REFSEL_VCCDIV2_gc = (0x04<<4) ; Internal VCC / 2
3875
3876   ; Channel sweep selection
3877   .equ ADC_SWEEP_0_gc = (0x00<<6) ; ADC Channel 0
3878   .equ ADC_SWEEP_01_gc = (0x01<<6) ; ADC Channel 0,1
3879   .equ ADC_SWEEP_012_gc = (0x02<<6) ; ADC Channel 0,1,2
3880   .equ ADC_SWEEP_0123_gc = (0x03<<6) ; ADC Channel 0,1,2,3
3881
3882   ; Event channel input selection
3883   .equ ADC_EVSEL_0123_gc = (0x00<<3) ; Event Channel 0,1,2,3
3884   .equ ADC_EVSEL_1234_gc = (0x01<<3) ; Event Channel 1,2,3,4
```

```
3885  .equ ADC_EVSEL_2345_gc = (0x02<<3) ; Event Channel 2,3,4,5
3886  .equ ADC_EVSEL_3456_gc = (0x03<<3) ; Event Channel 3,4,5,6
3887  .equ ADC_EVSEL_4567_gc = (0x04<<3) ; Event Channel 4,5,6,7
3888  .equ ADC_EVSEL_567_gc = (0x05<<3) ; Event Channel 5,6,7
3889  .equ ADC_EVSEL_67_gc = (0x06<<3) ; Event Channel 6,7
3890  .equ ADC_EVSEL_7_gc = (0x07<<3) ; Event Channel 7
3891
3892  ; Event action selection
3893  .equ ADC_EVACT_NONE_gc = (0x00<<0) ; No event action
3894  .equ ADC_EVACT_CH0_gc = (0x01<<0) ; First event triggers channel 0
3895  .equ ADC_EVACT_CH01_gc = (0x02<<0) ; First two events trigger channel 0,1
3896  .equ ADC_EVACT_CH012_gc = (0x03<<0) ; First three events trigger channel 0,1,2
3897  .equ ADC_EVACT_CH0123_gc = (0x04<<0) ; Events trigger channel 0,1,2,3
3898  .equ ADC_EVACT_SWEEP_gc = (0x05<<0) ; First event triggers sweep
3899  .equ ADC_EVACT_SYNCHSWEEP_gc = (0x06<<0) ; First event triggers synchronized sweep
3900
3901  ; Interupt mode
3902  .equ ADC_CH_INTMODE_COMPLETE_gc = (0x00<<2) ; Interrupt on conversion complete
3903  .equ ADC_CH_INTMODE_BELOW_gc = (0x01<<2) ; Interrupt on result below compare value
3904  .equ ADC_CH_INTMODE_ABOVE_gc = (0x03<<2) ; Interrupt on result above compare value
3905
3906  ; Interrupt level
3907  .equ ADC_CH_INTLVL_OFF_gc = (0x00<<0) ; Interrupt disabled
3908  .equ ADC_CH_INTLVL_LO_gc = (0x01<<0) ; Low level
3909  .equ ADC_CH_INTLVL_MED_gc = (0x02<<0) ; Medium level
3910  .equ ADC_CH_INTLVL_HI_gc = (0x03<<0) ; High level
3911
3912  ; DMA request selection
3913  .equ ADC_DMASEL_OFF_gc = (0x00<<6) ; Combined DMA request OFF
3914  .equ ADC_DMASEL_CH01_gc = (0x01<<6) ; ADC Channel 0 or 1
3915  .equ ADC_DMASEL_CH012_gc = (0x02<<6) ; ADC Channel 0 or 1 or 2
3916  .equ ADC_DMASEL_CH0123_gc = (0x03<<6) ; ADC Channel 0 or 1 or 2 or 3
3917
3918  ; Clock prescaler
3919  .equ ADC_PRESCALER_DIV4_gc = (0x00<<0) ; Divide clock by 4
3920  .equ ADC_PRESCALER_DIV8_gc = (0x01<<0) ; Divide clock by 8
3921  .equ ADC_PRESCALER_DIV16_gc = (0x02<<0) ; Divide clock by 16
3922  .equ ADC_PRESCALER_DIV32_gc = (0x03<<0) ; Divide clock by 32
3923  .equ ADC_PRESCALER_DIV64_gc = (0x04<<0) ; Divide clock by 64
3924  .equ ADC_PRESCALER_DIV128_gc = (0x05<<0) ; Divide clock by 128
3925  .equ ADC_PRESCALER_DIV256_gc = (0x06<<0) ; Divide clock by 256
3926  .equ ADC_PRESCALER_DIV512_gc = (0x07<<0) ; Divide clock by 512
3927
3928
3929  ;**************************************************************************
3930  ;** DAC - Digital/Analog Converter
3931  ;*************************************************************************/
3932
3933  ; DAC_CTRLA masks
3934  .equ DAC_IDOEN_bm = 0x10 ; Internal Output Enable bit mask
3935  .equ DAC_IDOEN_bp = 4 ; Internal Output Enable bit position
3936  .equ DAC_CH1EN_bm = 0x08 ; Channel 1 Output Enable bit mask
3937  .equ DAC_CH1EN_bp = 3 ; Channel 1 Output Enable bit position
3938  .equ DAC_CH0EN_bm = 0x04 ; Channel 0 Output Enable bit mask
3939  .equ DAC_CH0EN_bp = 2 ; Channel 0 Output Enable bit position
3940  .equ DAC_LPMODE_bm = 0x02 ; Low Power Mode bit mask
3941  .equ DAC_LPMODE_bp = 1 ; Low Power Mode bit position
3942  .equ DAC_ENABLE_bm = 0x01 ; Enable bit mask
3943  .equ DAC_ENABLE_bp = 0 ; Enable bit position
3944
3945  ; DAC_CTRLB masks
3946  .equ DAC_CHSEL_gm = 0x60 ; Channel Select group mask
3947  .equ DAC_CHSEL_gp = 5 ; Channel Select group position
3948  .equ DAC_CHSEL0_bm = (1<<5) ; Channel Select bit 0 mask
3949  .equ DAC_CHSEL0_bp = 5 ; Channel Select bit 0 position
3950  .equ DAC_CHSEL1_bm = (1<<6) ; Channel Select bit 1 mask
```

```
3951  .equ DAC_CHSEL1_bp = 6 ; Channel Select bit 1 position
3952  .equ DAC_CH1TRIG_bm = 0x02 ; Channel 1 Event Trig Enable bit mask
3953  .equ DAC_CH1TRIG_bp = 1 ; Channel 1 Event Trig Enable bit position
3954  .equ DAC_CH0TRIG_bm = 0x01 ; Channel 0 Event Trig Enable bit mask
3955  .equ DAC_CH0TRIG_bp = 0 ; Channel 0 Event Trig Enable bit position
3956
3957  ; DAC_CTRLC masks
3958  .equ DAC_REFSEL_gm = 0x18 ; Reference Select group mask
3959  .equ DAC_REFSEL_gp = 3 ; Reference Select group position
3960  .equ DAC_REFSEL0_bm = (1<<3) ; Reference Select bit 0 mask
3961  .equ DAC_REFSEL0_bp = 3 ; Reference Select bit 0 position
3962  .equ DAC_REFSEL1_bm = (1<<4) ; Reference Select bit 1 mask
3963  .equ DAC_REFSEL1_bp = 4 ; Reference Select bit 1 position
3964  .equ DAC_LEFTADJ_bm = 0x01 ; Left-adjust Result bit mask
3965  .equ DAC_LEFTADJ_bp = 0 ; Left-adjust Result bit position
3966
3967  ; DAC_EVCTRL masks
3968  .equ DAC_EVSPLIT_bm = 0x08 ; Separate Event Channel Input for Channel 1 bit mask
3969  .equ DAC_EVSPLIT_bp = 3 ; Separate Event Channel Input for Channel 1 bit position
3970  .equ DAC_EVSEL_gm = 0x07 ; Event Input Selection group mask
3971  .equ DAC_EVSEL_gp = 0 ; Event Input Selection group position
3972  .equ DAC_EVSEL0_bm = (1<<0) ; Event Input Selection bit 0 mask
3973  .equ DAC_EVSEL0_bp = 0 ; Event Input Selection bit 0 position
3974  .equ DAC_EVSEL1_bm = (1<<1) ; Event Input Selection bit 1 mask
3975  .equ DAC_EVSEL1_bp = 1 ; Event Input Selection bit 1 position
3976  .equ DAC_EVSEL2_bm = (1<<2) ; Event Input Selection bit 2 mask
3977  .equ DAC_EVSEL2_bp = 2 ; Event Input Selection bit 2 position
3978
3979  ; DAC_TIMCTRL masks
3980  .equ DAC_CONINTVAL_gm = 0x70 ; Conversion Intercal group mask
3981  .equ DAC_CONINTVAL_gp = 4 ; Conversion Intercal group position
3982  .equ DAC_CONINTVAL0_bm = (1<<4) ; Conversion Intercal bit 0 mask
3983  .equ DAC_CONINTVAL0_bp = 4 ; Conversion Intercal bit 0 position
3984  .equ DAC_CONINTVAL1_bm = (1<<5) ; Conversion Intercal bit 1 mask
3985  .equ DAC_CONINTVAL1_bp = 5 ; Conversion Intercal bit 1 position
3986  .equ DAC_CONINTVAL2_bm = (1<<6) ; Conversion Intercal bit 2 mask
3987  .equ DAC_CONINTVAL2_bp = 6 ; Conversion Intercal bit 2 position
3988  .equ DAC_REFRESH_gm = 0x0F ; Refresh Timing Control group mask
3989  .equ DAC_REFRESH_gp = 0 ; Refresh Timing Control group position
3990  .equ DAC_REFRESH0_bm = (1<<0) ; Refresh Timing Control bit 0 mask
3991  .equ DAC_REFRESH0_bp = 0 ; Refresh Timing Control bit 0 position
3992  .equ DAC_REFRESH1_bm = (1<<1) ; Refresh Timing Control bit 1 mask
3993  .equ DAC_REFRESH1_bp = 1 ; Refresh Timing Control bit 1 position
3994  .equ DAC_REFRESH2_bm = (1<<2) ; Refresh Timing Control bit 2 mask
3995  .equ DAC_REFRESH2_bp = 2 ; Refresh Timing Control bit 2 position
3996  .equ DAC_REFRESH3_bm = (1<<3) ; Refresh Timing Control bit 3 mask
3997  .equ DAC_REFRESH3_bp = 3 ; Refresh Timing Control bit 3 position
3998
3999  ; DAC_STATUS masks
4000  .equ DAC_CH1DRE_bm = 0x02 ; Channel 1 Data Register Empty bit mask
4001  .equ DAC_CH1DRE_bp = 1 ; Channel 1 Data Register Empty bit position
4002  .equ DAC_CH0DRE_bm = 0x01 ; Channel 0 Data Register Empty bit mask
4003  .equ DAC_CH0DRE_bp = 0 ; Channel 0 Data Register Empty bit position
4004
4005  ; DAC_CH0GAINCAL masks
4006  .equ DAC_CH0GAINCAL_gm = 0x7F ; Gain Calibration group mask
4007  .equ DAC_CH0GAINCAL_gp = 0 ; Gain Calibration group position
4008  .equ DAC_CH0GAINCAL0_bm = (1<<0) ; Gain Calibration bit 0 mask
4009  .equ DAC_CH0GAINCAL0_bp = 0 ; Gain Calibration bit 0 position
4010  .equ DAC_CH0GAINCAL1_bm = (1<<1) ; Gain Calibration bit 1 mask
4011  .equ DAC_CH0GAINCAL1_bp = 1 ; Gain Calibration bit 1 position
4012  .equ DAC_CH0GAINCAL2_bm = (1<<2) ; Gain Calibration bit 2 mask
4013  .equ DAC_CH0GAINCAL2_bp = 2 ; Gain Calibration bit 2 position
4014  .equ DAC_CH0GAINCAL3_bm = (1<<3) ; Gain Calibration bit 3 mask
4015  .equ DAC_CH0GAINCAL3_bp = 3 ; Gain Calibration bit 3 position
4016  .equ DAC_CH0GAINCAL4_bm = (1<<4) ; Gain Calibration bit 4 mask
```

```
4017   .equ DAC_CH0GAINCAL4_bp = 4 ; Gain Calibration bit 4 position
4018   .equ DAC_CH0GAINCAL5_bm = (1<<5) ; Gain Calibration bit 5 mask
4019   .equ DAC_CH0GAINCAL5_bp = 5 ; Gain Calibration bit 5 position
4020   .equ DAC_CH0GAINCAL6_bm = (1<<6) ; Gain Calibration bit 6 mask
4021   .equ DAC_CH0GAINCAL6_bp = 6 ; Gain Calibration bit 6 position
4022
4023   ; DAC_CH0OFFSETCAL masks
4024   .equ DAC_CH0OFFSETCAL_gm = 0x7F ; Offset Calibration group mask
4025   .equ DAC_CH0OFFSETCAL_gp = 0 ; Offset Calibration group position
4026   .equ DAC_CH0OFFSETCAL0_bm = (1<<0) ; Offset Calibration bit 0 mask
4027   .equ DAC_CH0OFFSETCAL0_bp = 0 ; Offset Calibration bit 0 position
4028   .equ DAC_CH0OFFSETCAL1_bm = (1<<1) ; Offset Calibration bit 1 mask
4029   .equ DAC_CH0OFFSETCAL1_bp = 1 ; Offset Calibration bit 1 position
4030   .equ DAC_CH0OFFSETCAL2_bm = (1<<2) ; Offset Calibration bit 2 mask
4031   .equ DAC_CH0OFFSETCAL2_bp = 2 ; Offset Calibration bit 2 position
4032   .equ DAC_CH0OFFSETCAL3_bm = (1<<3) ; Offset Calibration bit 3 mask
4033   .equ DAC_CH0OFFSETCAL3_bp = 3 ; Offset Calibration bit 3 position
4034   .equ DAC_CH0OFFSETCAL4_bm = (1<<4) ; Offset Calibration bit 4 mask
4035   .equ DAC_CH0OFFSETCAL4_bp = 4 ; Offset Calibration bit 4 position
4036   .equ DAC_CH0OFFSETCAL5_bm = (1<<5) ; Offset Calibration bit 5 mask
4037   .equ DAC_CH0OFFSETCAL5_bp = 5 ; Offset Calibration bit 5 position
4038   .equ DAC_CH0OFFSETCAL6_bm = (1<<6) ; Offset Calibration bit 6 mask
4039   .equ DAC_CH0OFFSETCAL6_bp = 6 ; Offset Calibration bit 6 position
4040
4041   ; DAC_CH1GAINCAL masks
4042   .equ DAC_CH1GAINCAL_gm = 0x7F ; Gain Calibration group mask
4043   .equ DAC_CH1GAINCAL_gp = 0 ; Gain Calibration group position
4044   .equ DAC_CH1GAINCAL0_bm = (1<<0) ; Gain Calibration bit 0 mask
4045   .equ DAC_CH1GAINCAL0_bp = 0 ; Gain Calibration bit 0 position
4046   .equ DAC_CH1GAINCAL1_bm = (1<<1) ; Gain Calibration bit 1 mask
4047   .equ DAC_CH1GAINCAL1_bp = 1 ; Gain Calibration bit 1 position
4048   .equ DAC_CH1GAINCAL2_bm = (1<<2) ; Gain Calibration bit 2 mask
4049   .equ DAC_CH1GAINCAL2_bp = 2 ; Gain Calibration bit 2 position
4050   .equ DAC_CH1GAINCAL3_bm = (1<<3) ; Gain Calibration bit 3 mask
4051   .equ DAC_CH1GAINCAL3_bp = 3 ; Gain Calibration bit 3 position
4052   .equ DAC_CH1GAINCAL4_bm = (1<<4) ; Gain Calibration bit 4 mask
4053   .equ DAC_CH1GAINCAL4_bp = 4 ; Gain Calibration bit 4 position
4054   .equ DAC_CH1GAINCAL5_bm = (1<<5) ; Gain Calibration bit 5 mask
4055   .equ DAC_CH1GAINCAL5_bp = 5 ; Gain Calibration bit 5 position
4056   .equ DAC_CH1GAINCAL6_bm = (1<<6) ; Gain Calibration bit 6 mask
4057   .equ DAC_CH1GAINCAL6_bp = 6 ; Gain Calibration bit 6 position
4058
4059   ; DAC_CH1OFFSETCAL masks
4060   .equ DAC_CH1OFFSETCAL_gm = 0x7F ; Offset Calibration group mask
4061   .equ DAC_CH1OFFSETCAL_gp = 0 ; Offset Calibration group position
4062   .equ DAC_CH1OFFSETCAL0_bm = (1<<0) ; Offset Calibration bit 0 mask
4063   .equ DAC_CH1OFFSETCAL0_bp = 0 ; Offset Calibration bit 0 position
4064   .equ DAC_CH1OFFSETCAL1_bm = (1<<1) ; Offset Calibration bit 1 mask
4065   .equ DAC_CH1OFFSETCAL1_bp = 1 ; Offset Calibration bit 1 position
4066   .equ DAC_CH1OFFSETCAL2_bm = (1<<2) ; Offset Calibration bit 2 mask
4067   .equ DAC_CH1OFFSETCAL2_bp = 2 ; Offset Calibration bit 2 position
4068   .equ DAC_CH1OFFSETCAL3_bm = (1<<3) ; Offset Calibration bit 3 mask
4069   .equ DAC_CH1OFFSETCAL3_bp = 3 ; Offset Calibration bit 3 position
4070   .equ DAC_CH1OFFSETCAL4_bm = (1<<4) ; Offset Calibration bit 4 mask
4071   .equ DAC_CH1OFFSETCAL4_bp = 4 ; Offset Calibration bit 4 position
4072   .equ DAC_CH1OFFSETCAL5_bm = (1<<5) ; Offset Calibration bit 5 mask
4073   .equ DAC_CH1OFFSETCAL5_bp = 5 ; Offset Calibration bit 5 position
4074   .equ DAC_CH1OFFSETCAL6_bm = (1<<6) ; Offset Calibration bit 6 mask
4075   .equ DAC_CH1OFFSETCAL6_bp = 6 ; Offset Calibration bit 6 position
4076
4077   ; Output channel selection
4078   .equ DAC_CHSEL_SINGLE_gc = (0x00<<5) ; Single channel operation (Channel 0 only)
4079   .equ DAC_CHSEL_SINGLE1_gc = (0x01<<5) ; Single channel operation (Channel 1 only)
4080   .equ DAC_CHSEL_DUAL_gc = (0x02<<5) ; Dual channel operation (Channel 0 and channel 1)
4081
4082   ; Reference voltage selection
```

```
4083  .equ DAC_REFSEL_INT1V_gc = (0x00<<3) ; Internal 1V
4084  .equ DAC_REFSEL_AVCC_gc = (0x01<<3) ; Analog supply voltage
4085  .equ DAC_REFSEL_AREFA_gc = (0x02<<3) ; External reference on AREF on PORTA
4086  .equ DAC_REFSEL_AREFB_gc = (0x03<<3) ; External reference on AREF on PORTB
4087
4088  ; Event channel selection
4089  .equ DAC_EVSEL_0_gc = (0x00<<0) ; Event Channel 0
4090  .equ DAC_EVSEL_1_gc = (0x01<<0) ; Event Channel 1
4091  .equ DAC_EVSEL_2_gc = (0x02<<0) ; Event Channel 2
4092  .equ DAC_EVSEL_3_gc = (0x03<<0) ; Event Channel 3
4093  .equ DAC_EVSEL_4_gc = (0x04<<0) ; Event Channel 4
4094  .equ DAC_EVSEL_5_gc = (0x05<<0) ; Event Channel 5
4095  .equ DAC_EVSEL_6_gc = (0x06<<0) ; Event Channel 6
4096  .equ DAC_EVSEL_7_gc = (0x07<<0) ; Event Channel 7
4097
4098  ; Conversion interval
4099  .equ DAC_CONINTVAL_1CLK_gc = (0x00<<4) ; 1 CLK / 2 CLK in S/H mode
4100  .equ DAC_CONINTVAL_2CLK_gc = (0x01<<4) ; 2 CLK / 3 CLK in S/H mode
4101  .equ DAC_CONINTVAL_4CLK_gc = (0x02<<4) ; 4 CLK / 6 CLK in S/H mode
4102  .equ DAC_CONINTVAL_8CLK_gc = (0x03<<4) ; 8 CLK / 12 CLK in S/H mode
4103  .equ DAC_CONINTVAL_16CLK_gc = (0x04<<4) ; 16 CLK / 24 CLK in S/H mode
4104  .equ DAC_CONINTVAL_32CLK_gc = (0x05<<4) ; 32 CLK / 48 CLK in S/H mode
4105  .equ DAC_CONINTVAL_64CLK_gc = (0x06<<4) ; 64 CLK / 96 CLK in S/H mode
4106  .equ DAC_CONINTVAL_128CLK_gc = (0x07<<4) ; 128 CLK / 192 CLK in S/H mode
4107
4108  ; Refresh rate
4109  .equ DAC_REFRESH_16CLK_gc = (0x00<<0) ; 16 CLK
4110  .equ DAC_REFRESH_32CLK_gc = (0x01<<0) ; 32 CLK
4111  .equ DAC_REFRESH_64CLK_gc = (0x02<<0) ; 64 CLK
4112  .equ DAC_REFRESH_128CLK_gc = (0x03<<0) ; 128 CLK
4113  .equ DAC_REFRESH_256CLK_gc = (0x04<<0) ; 256 CLK
4114  .equ DAC_REFRESH_512CLK_gc = (0x05<<0) ; 512 CLK
4115  .equ DAC_REFRESH_1024CLK_gc = (0x06<<0) ; 1024 CLK
4116  .equ DAC_REFRESH_2048CLK_gc = (0x07<<0) ; 2048 CLK
4117  .equ DAC_REFRESH_4096CLK_gc = (0x08<<0) ; 4096 CLK
4118  .equ DAC_REFRESH_8192CLK_gc = (0x09<<0) ; 8192 CLK
4119  .equ DAC_REFRESH_16384CLK_gc = (0x0A<<0) ; 16384 CLK
4120  .equ DAC_REFRESH_32768CLK_gc = (0x0B<<0) ; 32768 CLK
4121  .equ DAC_REFRESH_65536CLK_gc = (0x0C<<0) ; 65536 CLK
4122  .equ DAC_REFRESH_OFF_gc = (0x0F<<0) ; Auto refresh OFF
4123
4124
4125  ;**************************************************************************
4126  ;** AC - Analog Comparator
4127  ;*************************************************************************/
4128
4129  ; AC_AC0CTRL masks
4130  .equ AC_INTMODE_gm = 0xC0 ; Interrupt Mode group mask
4131  .equ AC_INTMODE_gp = 6 ; Interrupt Mode group position
4132  .equ AC_INTMODE0_bm = (1<<6) ; Interrupt Mode bit 0 mask
4133  .equ AC_INTMODE0_bp = 6 ; Interrupt Mode bit 0 position
4134  .equ AC_INTMODE1_bm = (1<<7) ; Interrupt Mode bit 1 mask
4135  .equ AC_INTMODE1_bp = 7 ; Interrupt Mode bit 1 position
4136  .equ AC_INTLVL_gm = 0x30 ; Interrupt Level group mask
4137  .equ AC_INTLVL_gp = 4 ; Interrupt Level group position
4138  .equ AC_INTLVL0_bm = (1<<4) ; Interrupt Level bit 0 mask
4139  .equ AC_INTLVL0_bp = 4 ; Interrupt Level bit 0 position
4140  .equ AC_INTLVL1_bm = (1<<5) ; Interrupt Level bit 1 mask
4141  .equ AC_INTLVL1_bp = 5 ; Interrupt Level bit 1 position
4142  .equ AC_HSMODE_bm = 0x08 ; High-speed Mode bit mask
4143  .equ AC_HSMODE_bp = 3 ; High-speed Mode bit position
4144  .equ AC_HYSMODE_gm = 0x06 ; Hysteresis Mode group mask
4145  .equ AC_HYSMODE_gp = 1 ; Hysteresis Mode group position
4146  .equ AC_HYSMODE0_bm = (1<<1) ; Hysteresis Mode bit 0 mask
4147  .equ AC_HYSMODE0_bp = 1 ; Hysteresis Mode bit 0 position
4148  .equ AC_HYSMODE1_bm = (1<<2) ; Hysteresis Mode bit 1 mask
```

```
4149  .equ AC_HYSMODE1_bp = 2 ; Hysteresis Mode bit 1 position
4150  .equ AC_ENABLE_bm = 0x01 ; Enable bit mask
4151  .equ AC_ENABLE_bp = 0 ; Enable bit position
4152
4153  ; AC_AC1CTRL masks
4154  ; Masks for INTMODE aready defined
4155  ; Masks for INTLVL aready defined
4156  ; Masks for HSMODE aready defined
4157  ; Masks for HYSMODE aready defined
4158  ; Masks for ENABLE aready defined
4159
4160  ; AC_AC0MUXCTRL masks
4161  .equ AC_MUXPOS_gm = 0x38 ; MUX Positive Input group mask
4162  .equ AC_MUXPOS_gp = 3 ; MUX Positive Input group position
4163  .equ AC_MUXPOS0_bm = (1<<3) ; MUX Positive Input bit 0 mask
4164  .equ AC_MUXPOS0_bp = 3 ; MUX Positive Input bit 0 position
4165  .equ AC_MUXPOS1_bm = (1<<4) ; MUX Positive Input bit 1 mask
4166  .equ AC_MUXPOS1_bp = 4 ; MUX Positive Input bit 1 position
4167  .equ AC_MUXPOS2_bm = (1<<5) ; MUX Positive Input bit 2 mask
4168  .equ AC_MUXPOS2_bp = 5 ; MUX Positive Input bit 2 position
4169  .equ AC_MUXNEG_gm = 0x07 ; MUX Negative Input group mask
4170  .equ AC_MUXNEG_gp = 0 ; MUX Negative Input group position
4171  .equ AC_MUXNEG0_bm = (1<<0) ; MUX Negative Input bit 0 mask
4172  .equ AC_MUXNEG0_bp = 0 ; MUX Negative Input bit 0 position
4173  .equ AC_MUXNEG1_bm = (1<<1) ; MUX Negative Input bit 1 mask
4174  .equ AC_MUXNEG1_bp = 1 ; MUX Negative Input bit 1 position
4175  .equ AC_MUXNEG2_bm = (1<<2) ; MUX Negative Input bit 2 mask
4176  .equ AC_MUXNEG2_bp = 2 ; MUX Negative Input bit 2 position
4177
4178  ; AC_AC1MUXCTRL masks
4179  ; Masks for MUXPOS aready defined
4180  ; Masks for MUXNEG aready defined
4181
4182  ; AC_CTRLA masks
4183  .equ AC_AC1OUT_bm = 0x02 ; Analog Comparator 1 Output Enable bit mask
4184  .equ AC_AC1OUT_bp = 1 ; Analog Comparator 1 Output Enable bit position
4185  .equ AC_AC0OUT_bm = 0x01 ; Analog Comparator 0 Output Enable bit mask
4186  .equ AC_AC0OUT_bp = 0 ; Analog Comparator 0 Output Enable bit position
4187
4188  ; AC_CTRLB masks
4189  .equ AC_SCALEFAC_gm = 0x3F ; VCC Voltage Scaler Factor group mask
4190  .equ AC_SCALEFAC_gp = 0 ; VCC Voltage Scaler Factor group position
4191  .equ AC_SCALEFAC0_bm = (1<<0) ; VCC Voltage Scaler Factor bit 0 mask
4192  .equ AC_SCALEFAC0_bp = 0 ; VCC Voltage Scaler Factor bit 0 position
4193  .equ AC_SCALEFAC1_bm = (1<<1) ; VCC Voltage Scaler Factor bit 1 mask
4194  .equ AC_SCALEFAC1_bp = 1 ; VCC Voltage Scaler Factor bit 1 position
4195  .equ AC_SCALEFAC2_bm = (1<<2) ; VCC Voltage Scaler Factor bit 2 mask
4196  .equ AC_SCALEFAC2_bp = 2 ; VCC Voltage Scaler Factor bit 2 position
4197  .equ AC_SCALEFAC3_bm = (1<<3) ; VCC Voltage Scaler Factor bit 3 mask
4198  .equ AC_SCALEFAC3_bp = 3 ; VCC Voltage Scaler Factor bit 3 position
4199  .equ AC_SCALEFAC4_bm = (1<<4) ; VCC Voltage Scaler Factor bit 4 mask
4200  .equ AC_SCALEFAC4_bp = 4 ; VCC Voltage Scaler Factor bit 4 position
4201  .equ AC_SCALEFAC5_bm = (1<<5) ; VCC Voltage Scaler Factor bit 5 mask
4202  .equ AC_SCALEFAC5_bp = 5 ; VCC Voltage Scaler Factor bit 5 position
4203
4204  ; AC_WINCTRL masks
4205  .equ AC_WEN_bm = 0x10 ; Window Mode Enable bit mask
4206  .equ AC_WEN_bp = 4 ; Window Mode Enable bit position
4207  .equ AC_WINTMODE_gm = 0x0C ; Window Interrupt Mode group mask
4208  .equ AC_WINTMODE_gp = 2 ; Window Interrupt Mode group position
4209  .equ AC_WINTMODE0_bm = (1<<2) ; Window Interrupt Mode bit 0 mask
4210  .equ AC_WINTMODE0_bp = 2 ; Window Interrupt Mode bit 0 position
4211  .equ AC_WINTMODE1_bm = (1<<3) ; Window Interrupt Mode bit 1 mask
4212  .equ AC_WINTMODE1_bp = 3 ; Window Interrupt Mode bit 1 position
4213  .equ AC_WINTLVL_gm = 0x03 ; Window Interrupt Level group mask
4214  .equ AC_WINTLVL_gp = 0 ; Window Interrupt Level group position
```

```
4215  .equ AC_WINTLVL0_bm = (1<<0) ; Window Interrupt Level bit 0 mask
4216  .equ AC_WINTLVL0_bp = 0 ; Window Interrupt Level bit 0 position
4217  .equ AC_WINTLVL1_bm = (1<<1) ; Window Interrupt Level bit 1 mask
4218  .equ AC_WINTLVL1_bp = 1 ; Window Interrupt Level bit 1 position
4219
4220  ; AC_STATUS masks
4221  .equ AC_WSTATE_gm = 0xC0 ; Window Mode State group mask
4222  .equ AC_WSTATE_gp = 6 ; Window Mode State group position
4223  .equ AC_WSTATE0_bm = (1<<6) ; Window Mode State bit 0 mask
4224  .equ AC_WSTATE0_bp = 6 ; Window Mode State bit 0 position
4225  .equ AC_WSTATE1_bm = (1<<7) ; Window Mode State bit 1 mask
4226  .equ AC_WSTATE1_bp = 7 ; Window Mode State bit 1 position
4227  .equ AC_AC1STATE_bm = 0x20 ; Analog Comparator 1 State bit mask
4228  .equ AC_AC1STATE_bp = 5 ; Analog Comparator 1 State bit position
4229  .equ AC_AC0STATE_bm = 0x10 ; Analog Comparator 0 State bit mask
4230  .equ AC_AC0STATE_bp = 4 ; Analog Comparator 0 State bit position
4231  .equ AC_WIF_bm = 0x04 ; Window Mode Interrupt Flag bit mask
4232  .equ AC_WIF_bp = 2 ; Window Mode Interrupt Flag bit position
4233  .equ AC_AC1IF_bm = 0x02 ; Analog Comparator 1 Interrupt Flag bit mask
4234  .equ AC_AC1IF_bp = 1 ; Analog Comparator 1 Interrupt Flag bit position
4235  .equ AC_AC0IF_bm = 0x01 ; Analog Comparator 0 Interrupt Flag bit mask
4236  .equ AC_AC0IF_bp = 0 ; Analog Comparator 0 Interrupt Flag bit position
4237
4238  ; Interrupt mode
4239  .equ AC_INTMODE_BOTHEDGES_gc = (0x00<<6) ; Interrupt on both edges
4240  .equ AC_INTMODE_FALLING_gc = (0x02<<6) ; Interrupt on falling edge
4241  .equ AC_INTMODE_RISING_gc = (0x03<<6) ; Interrupt on rising edge
4242
4243  ; Interrupt level
4244  .equ AC_INTLVL_OFF_gc = (0x00<<4) ; Interrupt disabled
4245  .equ AC_INTLVL_LO_gc = (0x01<<4) ; Low level
4246  .equ AC_INTLVL_MED_gc = (0x02<<4) ; Medium level
4247  .equ AC_INTLVL_HI_gc = (0x03<<4) ; High level
4248
4249  ; Hysteresis mode selection
4250  .equ AC_HYSMODE_NO_gc = (0x00<<1) ; No hysteresis
4251  .equ AC_HYSMODE_SMALL_gc = (0x01<<1) ; Small hysteresis
4252  .equ AC_HYSMODE_LARGE_gc = (0x02<<1) ; Large hysteresis
4253
4254  ; Positive input multiplexer selection
4255  .equ AC_MUXPOS_PIN0_gc = (0x00<<3) ; Pin 0
4256  .equ AC_MUXPOS_PIN1_gc = (0x01<<3) ; Pin 1
4257  .equ AC_MUXPOS_PIN2_gc = (0x02<<3) ; Pin 2
4258  .equ AC_MUXPOS_PIN3_gc = (0x03<<3) ; Pin 3
4259  .equ AC_MUXPOS_PIN4_gc = (0x04<<3) ; Pin 4
4260  .equ AC_MUXPOS_PIN5_gc = (0x05<<3) ; Pin 5
4261  .equ AC_MUXPOS_PIN6_gc = (0x06<<3) ; Pin 6
4262  .equ AC_MUXPOS_DAC_gc = (0x07<<3) ; DAC output
4263
4264  ; Negative input multiplexer selection
4265  .equ AC_MUXNEG_PIN0_gc = (0x00<<0) ; Pin 0
4266  .equ AC_MUXNEG_PIN1_gc = (0x01<<0) ; Pin 1
4267  .equ AC_MUXNEG_PIN3_gc = (0x02<<0) ; Pin 3
4268  .equ AC_MUXNEG_PIN5_gc = (0x03<<0) ; Pin 5
4269  .equ AC_MUXNEG_PIN7_gc = (0x04<<0) ; Pin 7
4270  .equ AC_MUXNEG_DAC_gc = (0x05<<0) ; DAC output
4271  .equ AC_MUXNEG_BANDGAP_gc = (0x06<<0) ; Bandgap Reference
4272  .equ AC_MUXNEG_SCALER_gc = (0x07<<0) ; Internal voltage scaler
4273
4274  ; Windows interrupt mode
4275  .equ AC_WINTMODE_ABOVE_gc = (0x00<<2) ; Interrupt on above window
4276  .equ AC_WINTMODE_INSIDE_gc = (0x01<<2) ; Interrupt on inside window
4277  .equ AC_WINTMODE_BELOW_gc = (0x02<<2) ; Interrupt on below window
4278  .equ AC_WINTMODE_OUTSIDE_gc = (0x03<<2) ; Interrupt on outside window
4279
4280  ; Window interrupt level
```

```
4281  .equ AC_WINTLVL_OFF_gc = (0x00<<0) ; Interrupt disabled
4282  .equ AC_WINTLVL_LO_gc = (0x01<<0) ; Low priority
4283  .equ AC_WINTLVL_MED_gc = (0x02<<0) ; Medium priority
4284  .equ AC_WINTLVL_HI_gc = (0x03<<0) ; High priority
4285
4286  ; Window mode state
4287  .equ AC_WSTATE_ABOVE_gc = (0x00<<6) ; Signal above window
4288  .equ AC_WSTATE_INSIDE_gc = (0x01<<6) ; Signal inside window
4289  .equ AC_WSTATE_BELOW_gc = (0x02<<6) ; Signal below window
4290
4291
4292  ;**************************************************************************
4293  ;** RTC - Real-Time Clounter
4294  ;**************************************************************************/
4295
4296  ; RTC_CTRL masks
4297  .equ RTC_PRESCALER_gm = 0x07 ; Prescaling Factor group mask
4298  .equ RTC_PRESCALER_gp = 0 ; Prescaling Factor group position
4299  .equ RTC_PRESCALER0_bm = (1<<0) ; Prescaling Factor bit 0 mask
4300  .equ RTC_PRESCALER0_bp = 0 ; Prescaling Factor bit 0 position
4301  .equ RTC_PRESCALER1_bm = (1<<1) ; Prescaling Factor bit 1 mask
4302  .equ RTC_PRESCALER1_bp = 1 ; Prescaling Factor bit 1 position
4303  .equ RTC_PRESCALER2_bm = (1<<2) ; Prescaling Factor bit 2 mask
4304  .equ RTC_PRESCALER2_bp = 2 ; Prescaling Factor bit 2 position
4305
4306  ; RTC_STATUS masks
4307  .equ RTC_SYNCBUSY_bm = 0x01 ; Synchronization Busy Flag bit mask
4308  .equ RTC_SYNCBUSY_bp = 0 ; Synchronization Busy Flag bit position
4309
4310  ; RTC_INTCTRL masks
4311  .equ RTC_COMPINTLVL_gm = 0x0C ; Compare Match Interrupt Level group mask
4312  .equ RTC_COMPINTLVL_gp = 2 ; Compare Match Interrupt Level group position
4313  .equ RTC_COMPINTLVL0_bm = (1<<2) ; Compare Match Interrupt Level bit 0 mask
4314  .equ RTC_COMPINTLVL0_bp = 2 ; Compare Match Interrupt Level bit 0 position
4315  .equ RTC_COMPINTLVL1_bm = (1<<3) ; Compare Match Interrupt Level bit 1 mask
4316  .equ RTC_COMPINTLVL1_bp = 3 ; Compare Match Interrupt Level bit 1 position
4317  .equ RTC_OVFINTLVL_gm = 0x03 ; Overflow Interrupt Level group mask
4318  .equ RTC_OVFINTLVL_gp = 0 ; Overflow Interrupt Level group position
4319  .equ RTC_OVFINTLVL0_bm = (1<<0) ; Overflow Interrupt Level bit 0 mask
4320  .equ RTC_OVFINTLVL0_bp = 0 ; Overflow Interrupt Level bit 0 position
4321  .equ RTC_OVFINTLVL1_bm = (1<<1) ; Overflow Interrupt Level bit 1 mask
4322  .equ RTC_OVFINTLVL1_bp = 1 ; Overflow Interrupt Level bit 1 position
4323
4324  ; RTC_INTFLAGS masks
4325  .equ RTC_COMPIF_bm = 0x02 ; Compare Match Interrupt Flag bit mask
4326  .equ RTC_COMPIF_bp = 1 ; Compare Match Interrupt Flag bit position
4327  .equ RTC_OVFIF_bm = 0x01 ; Overflow Interrupt Flag bit mask
4328  .equ RTC_OVFIF_bp = 0 ; Overflow Interrupt Flag bit position
4329
4330  ; Prescaler Factor
4331  .equ RTC_PRESCALER_OFF_gc = (0x00<<0) ; RTC Off
4332  .equ RTC_PRESCALER_DIV1_gc = (0x01<<0) ; RTC Clock
4333  .equ RTC_PRESCALER_DIV2_gc = (0x02<<0) ; RTC Clock / 2
4334  .equ RTC_PRESCALER_DIV8_gc = (0x03<<0) ; RTC Clock / 8
4335  .equ RTC_PRESCALER_DIV16_gc = (0x04<<0) ; RTC Clock / 16
4336  .equ RTC_PRESCALER_DIV64_gc = (0x05<<0) ; RTC Clock / 64
4337  .equ RTC_PRESCALER_DIV256_gc = (0x06<<0) ; RTC Clock / 256
4338  .equ RTC_PRESCALER_DIV1024_gc = (0x07<<0) ; RTC Clock / 1024
4339
4340  ; Compare Interrupt level
4341  .equ RTC_COMPINTLVL_OFF_gc = (0x00<<2) ; Interrupt Disabled
4342  .equ RTC_COMPINTLVL_LO_gc = (0x01<<2) ; Low Level
4343  .equ RTC_COMPINTLVL_MED_gc = (0x02<<2) ; Medium Level
4344  .equ RTC_COMPINTLVL_HI_gc = (0x03<<2) ; High Level
4345
4346  ; Overflow Interrupt level
```

```
4347  .equ RTC_OVFINTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
4348  .equ RTC_OVFINTLVL_LO_gc = (0x01<<0) ; Low Level
4349  .equ RTC_OVFINTLVL_MED_gc = (0x02<<0) ; Medium Level
4350  .equ RTC_OVFINTLVL_HI_gc = (0x03<<0) ; High Level
4351
4352
4353  ;**************************************************************************
4354  ;** EBI - External Bus Interface
4355  ;**************************************************************************/
4356
4357  ; EBI_CS_CTRLA masks
4358  .equ EBI_CS_ASPACE_gm = 0x7C ; Address Space group mask
4359  .equ EBI_CS_ASPACE_gp = 2 ; Address Space group position
4360  .equ EBI_CS_ASPACE0_bm = (1<<2) ; Address Space bit 0 mask
4361  .equ EBI_CS_ASPACE0_bp = 2 ; Address Space bit 0 position
4362  .equ EBI_CS_ASPACE1_bm = (1<<3) ; Address Space bit 1 mask
4363  .equ EBI_CS_ASPACE1_bp = 3 ; Address Space bit 1 position
4364  .equ EBI_CS_ASPACE2_bm = (1<<4) ; Address Space bit 2 mask
4365  .equ EBI_CS_ASPACE2_bp = 4 ; Address Space bit 2 position
4366  .equ EBI_CS_ASPACE3_bm = (1<<5) ; Address Space bit 3 mask
4367  .equ EBI_CS_ASPACE3_bp = 5 ; Address Space bit 3 position
4368  .equ EBI_CS_ASPACE4_bm = (1<<6) ; Address Space bit 4 mask
4369  .equ EBI_CS_ASPACE4_bp = 6 ; Address Space bit 4 position
4370  .equ EBI_CS_MODE_gm = 0x03 ; Memory Mode group mask
4371  .equ EBI_CS_MODE_gp = 0 ; Memory Mode group position
4372  .equ EBI_CS_MODE0_bm = (1<<0) ; Memory Mode bit 0 mask
4373  .equ EBI_CS_MODE0_bp = 0 ; Memory Mode bit 0 position
4374  .equ EBI_CS_MODE1_bm = (1<<1) ; Memory Mode bit 1 mask
4375  .equ EBI_CS_MODE1_bp = 1 ; Memory Mode bit 1 position
4376
4377  ; EBI_CS_CTRLB masks
4378  .equ EBI_CS_SRWS_gm = 0x07 ; SRAM Wait State Cycles group mask
4379  .equ EBI_CS_SRWS_gp = 0 ; SRAM Wait State Cycles group position
4380  .equ EBI_CS_SRWS0_bm = (1<<0) ; SRAM Wait State Cycles bit 0 mask
4381  .equ EBI_CS_SRWS0_bp = 0 ; SRAM Wait State Cycles bit 0 position
4382  .equ EBI_CS_SRWS1_bm = (1<<1) ; SRAM Wait State Cycles bit 1 mask
4383  .equ EBI_CS_SRWS1_bp = 1 ; SRAM Wait State Cycles bit 1 position
4384  .equ EBI_CS_SRWS2_bm = (1<<2) ; SRAM Wait State Cycles bit 2 mask
4385  .equ EBI_CS_SRWS2_bp = 2 ; SRAM Wait State Cycles bit 2 position
4386  .equ EBI_CS_SDINITDONE_bm = 0x80 ; SDRAM Initialization Done bit mask
4387  .equ EBI_CS_SDINITDONE_bp = 7 ; SDRAM Initialization Done bit position
4388  .equ EBI_CS_SDSREN_bm = 0x04 ; SDRAM Self-refresh Enable bit mask
4389  .equ EBI_CS_SDSREN_bp = 2 ; SDRAM Self-refresh Enable bit position
4390  .equ EBI_CS_SDMODE_gm = 0x03 ; SDRAM Mode group mask
4391  .equ EBI_CS_SDMODE_gp = 0 ; SDRAM Mode group position
4392  .equ EBI_CS_SDMODE0_bm = (1<<0) ; SDRAM Mode bit 0 mask
4393  .equ EBI_CS_SDMODE0_bp = 0 ; SDRAM Mode bit 0 position
4394  .equ EBI_CS_SDMODE1_bm = (1<<1) ; SDRAM Mode bit 1 mask
4395  .equ EBI_CS_SDMODE1_bp = 1 ; SDRAM Mode bit 1 position
4396
4397  ; EBI_CTRL masks
4398  .equ EBI_SDDATAW_gm = 0xC0 ; SDRAM Data Width Setting group mask
4399  .equ EBI_SDDATAW_gp = 6 ; SDRAM Data Width Setting group position
4400  .equ EBI_SDDATAW0_bm = (1<<6) ; SDRAM Data Width Setting bit 0 mask
4401  .equ EBI_SDDATAW0_bp = 6 ; SDRAM Data Width Setting bit 0 position
4402  .equ EBI_SDDATAW1_bm = (1<<7) ; SDRAM Data Width Setting bit 1 mask
4403  .equ EBI_SDDATAW1_bp = 7 ; SDRAM Data Width Setting bit 1 position
4404  .equ EBI_LPCMODE_gm = 0x30 ; SRAM LPC Mode group mask
4405  .equ EBI_LPCMODE_gp = 4 ; SRAM LPC Mode group position
4406  .equ EBI_LPCMODE0_bm = (1<<4) ; SRAM LPC Mode bit 0 mask
4407  .equ EBI_LPCMODE0_bp = 4 ; SRAM LPC Mode bit 0 position
4408  .equ EBI_LPCMODE1_bm = (1<<5) ; SRAM LPC Mode bit 1 mask
4409  .equ EBI_LPCMODE1_bp = 5 ; SRAM LPC Mode bit 1 position
4410  .equ EBI_SRMODE_gm = 0x0C ; SRAM Mode group mask
4411  .equ EBI_SRMODE_gp = 2 ; SRAM Mode group position
4412  .equ EBI_SRMODE0_bm = (1<<2) ; SRAM Mode bit 0 mask
```

```
4413   .equ EBI_SRMODE0_bp = 2 ; SRAM Mode bit 0 position
4414   .equ EBI_SRMODE1_bm = (1<<3) ; SRAM Mode bit 1 mask
4415   .equ EBI_SRMODE1_bp = 3 ; SRAM Mode bit 1 position
4416   .equ EBI_IFMODE_gm = 0x03 ; Interface Mode group mask
4417   .equ EBI_IFMODE_gp = 0 ; Interface Mode group position
4418   .equ EBI_IFMODE0_bm = (1<<0) ; Interface Mode bit 0 mask
4419   .equ EBI_IFMODE0_bp = 0 ; Interface Mode bit 0 position
4420   .equ EBI_IFMODE1_bm = (1<<1) ; Interface Mode bit 1 mask
4421   .equ EBI_IFMODE1_bp = 1 ; Interface Mode bit 1 position
4422
4423   ; EBI_SDRAMCTRLA masks
4424   .equ EBI_SDCAS_bm = 0x08 ; SDRAM CAS Latency Setting bit mask
4425   .equ EBI_SDCAS_bp = 3 ; SDRAM CAS Latency Setting bit position
4426   .equ EBI_SDROW_bm = 0x04 ; SDRAM ROW Bits Setting bit mask
4427   .equ EBI_SDROW_bp = 2 ; SDRAM ROW Bits Setting bit position
4428   .equ EBI_SDCOL_gm = 0x03 ; SDRAM Column Bits Setting group mask
4429   .equ EBI_SDCOL_gp = 0 ; SDRAM Column Bits Setting group position
4430   .equ EBI_SDCOL0_bm = (1<<0) ; SDRAM Column Bits Setting bit 0 mask
4431   .equ EBI_SDCOL0_bp = 0 ; SDRAM Column Bits Setting bit 0 position
4432   .equ EBI_SDCOL1_bm = (1<<1) ; SDRAM Column Bits Setting bit 1 mask
4433   .equ EBI_SDCOL1_bp = 1 ; SDRAM Column Bits Setting bit 1 position
4434
4435   ; EBI_SDRAMCTRLB masks
4436   .equ EBI_MRDLY_gm = 0xC0 ; SDRAM Mode Register Delay group mask
4437   .equ EBI_MRDLY_gp = 6 ; SDRAM Mode Register Delay group position
4438   .equ EBI_MRDLY0_bm = (1<<6) ; SDRAM Mode Register Delay bit 0 mask
4439   .equ EBI_MRDLY0_bp = 6 ; SDRAM Mode Register Delay bit 0 position
4440   .equ EBI_MRDLY1_bm = (1<<7) ; SDRAM Mode Register Delay bit 1 mask
4441   .equ EBI_MRDLY1_bp = 7 ; SDRAM Mode Register Delay bit 1 position
4442   .equ EBI_ROWCYCDLY_gm = 0x38 ; SDRAM Row Cycle Delay group mask
4443   .equ EBI_ROWCYCDLY_gp = 3 ; SDRAM Row Cycle Delay group position
4444   .equ EBI_ROWCYCDLY0_bm = (1<<3) ; SDRAM Row Cycle Delay bit 0 mask
4445   .equ EBI_ROWCYCDLY0_bp = 3 ; SDRAM Row Cycle Delay bit 0 position
4446   .equ EBI_ROWCYCDLY1_bm = (1<<4) ; SDRAM Row Cycle Delay bit 1 mask
4447   .equ EBI_ROWCYCDLY1_bp = 4 ; SDRAM Row Cycle Delay bit 1 position
4448   .equ EBI_ROWCYCDLY2_bm = (1<<5) ; SDRAM Row Cycle Delay bit 2 mask
4449   .equ EBI_ROWCYCDLY2_bp = 5 ; SDRAM Row Cycle Delay bit 2 position
4450   .equ EBI_RPDLY_gm = 0x07 ; SDRAM Row-to-Precharge Delay group mask
4451   .equ EBI_RPDLY_gp = 0 ; SDRAM Row-to-Precharge Delay group position
4452   .equ EBI_RPDLY0_bm = (1<<0) ; SDRAM Row-to-Precharge Delay bit 0 mask
4453   .equ EBI_RPDLY0_bp = 0 ; SDRAM Row-to-Precharge Delay bit 0 position
4454   .equ EBI_RPDLY1_bm = (1<<1) ; SDRAM Row-to-Precharge Delay bit 1 mask
4455   .equ EBI_RPDLY1_bp = 1 ; SDRAM Row-to-Precharge Delay bit 1 position
4456   .equ EBI_RPDLY2_bm = (1<<2) ; SDRAM Row-to-Precharge Delay bit 2 mask
4457   .equ EBI_RPDLY2_bp = 2 ; SDRAM Row-to-Precharge Delay bit 2 position
4458
4459   ; EBI_SDRAMCTRLC masks
4460   .equ EBI_WRDLY_gm = 0xC0 ; SDRAM Write Recovery Delay group mask
4461   .equ EBI_WRDLY_gp = 6 ; SDRAM Write Recovery Delay group position
4462   .equ EBI_WRDLY0_bm = (1<<6) ; SDRAM Write Recovery Delay bit 0 mask
4463   .equ EBI_WRDLY0_bp = 6 ; SDRAM Write Recovery Delay bit 0 position
4464   .equ EBI_WRDLY1_bm = (1<<7) ; SDRAM Write Recovery Delay bit 1 mask
4465   .equ EBI_WRDLY1_bp = 7 ; SDRAM Write Recovery Delay bit 1 position
4466   .equ EBI_ESRDLY_gm = 0x38 ; SDRAM Exit-Self-refresh-to-Active Delay group mask
4467   .equ EBI_ESRDLY_gp = 3 ; SDRAM Exit-Self-refresh-to-Active Delay group position
4468   .equ EBI_ESRDLY0_bm = (1<<3) ; SDRAM Exit-Self-refresh-to-Active Delay bit 0 mask
4469   .equ EBI_ESRDLY0_bp = 3 ; SDRAM Exit-Self-refresh-to-Active Delay bit 0 position
4470   .equ EBI_ESRDLY1_bm = (1<<4) ; SDRAM Exit-Self-refresh-to-Active Delay bit 1 mask
4471   .equ EBI_ESRDLY1_bp = 4 ; SDRAM Exit-Self-refresh-to-Active Delay bit 1 position
4472   .equ EBI_ESRDLY2_bm = (1<<5) ; SDRAM Exit-Self-refresh-to-Active Delay bit 2 mask
4473   .equ EBI_ESRDLY2_bp = 5 ; SDRAM Exit-Self-refresh-to-Active Delay bit 2 position
4474   .equ EBI_ROWCOLDLY_gm = 0x07 ; SDRAM Row-to-Column Delay group mask
4475   .equ EBI_ROWCOLDLY_gp = 0 ; SDRAM Row-to-Column Delay group position
4476   .equ EBI_ROWCOLDLY0_bm = (1<<0) ; SDRAM Row-to-Column Delay bit 0 mask
4477   .equ EBI_ROWCOLDLY0_bp = 0 ; SDRAM Row-to-Column Delay bit 0 position
4478   .equ EBI_ROWCOLDLY1_bm = (1<<1) ; SDRAM Row-to-Column Delay bit 1 mask
```

```
.equ EBI_ROWCOLDLY1_bp = 1 ; SDRAM Row-to-Column Delay bit 1 position
.equ EBI_ROWCOLDLY2_bm = (1<<2) ; SDRAM Row-to-Column Delay bit 2 mask
.equ EBI_ROWCOLDLY2_bp = 2 ; SDRAM Row-to-Column Delay bit 2 position

; Chip Select adress space
.equ EBI_CS_ASPACE_256B_gc = (0x00<<2) ; 256 bytes
.equ EBI_CS_ASPACE_512B_gc = (0x01<<2) ; 512 bytes
.equ EBI_CS_ASPACE_1KB_gc = (0x02<<2) ; 1K bytes
.equ EBI_CS_ASPACE_2KB_gc = (0x03<<2) ; 2K bytes
.equ EBI_CS_ASPACE_4KB_gc = (0x04<<2) ; 4K bytes
.equ EBI_CS_ASPACE_8KB_gc = (0x05<<2) ; 8K bytes
.equ EBI_CS_ASPACE_16KB_gc = (0x06<<2) ; 16K bytes
.equ EBI_CS_ASPACE_32KB_gc = (0x07<<2) ; 32K bytes
.equ EBI_CS_ASPACE_64KB_gc = (0x08<<2) ; 64K bytes
.equ EBI_CS_ASPACE_128KB_gc = (0x09<<2) ; 128K bytes
.equ EBI_CS_ASPACE_256KB_gc = (0x0A<<2) ; 256K bytes
.equ EBI_CS_ASPACE_512KB_gc = (0x0B<<2) ; 512K bytes
.equ EBI_CS_ASPACE_1MB_gc = (0x0C<<2) ; 1M bytes
.equ EBI_CS_ASPACE_2MB_gc = (0x0D<<2) ; 2M bytes
.equ EBI_CS_ASPACE_4MB_gc = (0x0E<<2) ; 4M bytes
.equ EBI_CS_ASPACE_8MB_gc = (0x0F<<2) ; 8M bytes
.equ EBI_CS_ASPACE_16M_gc = (0x10<<2) ; 16M bytes

; SRAM Wait State Selection
.equ EBI_CS_SRWS_0CLK_gc = (0x00<<0) ; 0 cycles
.equ EBI_CS_SRWS_1CLK_gc = (0x01<<0) ; 1 cycle
.equ EBI_CS_SRWS_2CLK_gc = (0x02<<0) ; 2 cycles
.equ EBI_CS_SRWS_3CLK_gc = (0x03<<0) ; 3 cycles
.equ EBI_CS_SRWS_4CLK_gc = (0x04<<0) ; 4 cycles
.equ EBI_CS_SRWS_5CLK_gc = (0x05<<0) ; 5 cycles
.equ EBI_CS_SRWS_6CLK_gc = (0x06<<0) ; 6 cycles
.equ EBI_CS_SRWS_7CLK_gc = (0x07<<0) ; 7 cycles

; Chip Select address mode
.equ EBI_CS_MODE_DISABLED_gc = (0x00<<0) ; Chip Select Disabled
.equ EBI_CS_MODE_SRAM_gc = (0x01<<0) ; Chip Select in SRAM mode
.equ EBI_CS_MODE_LPC_gc = (0x02<<0) ; Chip Select in SRAM LPC mode
.equ EBI_CS_MODE_SDRAM_gc = (0x03<<0) ; Chip Select in SDRAM mode

; Chip Select SDRAM mode
.equ EBI_CS_SDMODE_NORMAL_gc = (0x00<<0) ; Normal mode
.equ EBI_CS_SDMODE_LOAD_gc = (0x01<<0) ; Load Mode Register command mode

;
.equ EBI_SDDATAW_4BIT_gc = (0x00<<6) ; 4-bit data bus
.equ EBI_SDDATAW_8BIT_gc = (0x01<<6) ; 8-bit data bus

;
.equ EBI_LPCMODE_ALE1_gc = (0x00<<4) ; Data muxed with addr byte 0
.equ EBI_LPCMODE_ALE12_gc = (0x02<<4) ; Data muxed with addr byte 0 and 1

;
.equ EBI_SRMODE_ALE1_gc = (0x00<<2) ; Addr byte 0 muxed with 1
.equ EBI_SRMODE_ALE2_gc = (0x01<<2) ; Addr byte 0 muxed with 2
.equ EBI_SRMODE_ALE12_gc = (0x02<<2) ; Addr byte 0 muxed with 1 and 2
.equ EBI_SRMODE_NOALE_gc = (0x03<<2) ; No addr muxing

;
.equ EBI_IFMODE_DISABLED_gc = (0x00<<0) ; EBI Disabled
.equ EBI_IFMODE_3PORT_gc = (0x01<<0) ; 3-port mode
.equ EBI_IFMODE_4PORT_gc = (0x02<<0) ; 4-port mode
.equ EBI_IFMODE_2PORT_gc = (0x03<<0) ; 2-port mode

;
.equ EBI_SDCOL_8BIT_gc = (0x00<<0) ; 8 column bits
.equ EBI_SDCOL_9BIT_gc = (0x01<<0) ; 9 column bits
```

```
4545   .equ EBI_SDCOL_10BIT_gc = (0x02<<0) ; 10 column bits
4546   .equ EBI_SDCOL_11BIT_gc = (0x03<<0) ; 11 column bits
4547
4548   ; SDRAM Load Mode to Active delay
4549   .equ EBI_MRDLY_0CLK_gc = (0x00<<6) ; 0 cycles
4550   .equ EBI_MRDLY_1CLK_gc = (0x01<<6) ; 1 cycle
4551   .equ EBI_MRDLY_2CLK_gc = (0x02<<6) ; 2 cycles
4552   .equ EBI_MRDLY_3CLK_gc = (0x03<<6) ; 3 cycles
4553
4554   ; SDRAM Row Cycle Delay
4555   .equ EBI_ROWCYCDLY_0CLK_gc = (0x00<<3) ; 0 cycles
4556   .equ EBI_ROWCYCDLY_1CLK_gc = (0x01<<3) ; 1 cycle
4557   .equ EBI_ROWCYCDLY_2CLK_gc = (0x02<<3) ; 2 cycles
4558   .equ EBI_ROWCYCDLY_3CLK_gc = (0x03<<3) ; 3 cycles
4559   .equ EBI_ROWCYCDLY_4CLK_gc = (0x04<<3) ; 4 cycles
4560   .equ EBI_ROWCYCDLY_5CLK_gc = (0x05<<3) ; 5 cycles
4561   .equ EBI_ROWCYCDLY_6CLK_gc = (0x06<<3) ; 6 cycles
4562   .equ EBI_ROWCYCDLY_7CLK_gc = (0x07<<3) ; 7 cycles
4563
4564   ; SDRAM Row to Precharge Delay
4565   .equ EBI_RPDLY_0CLK_gc = (0x00<<0) ; 0 cycles
4566   .equ EBI_RPDLY_1CLK_gc = (0x01<<0) ; 1 cycle
4567   .equ EBI_RPDLY_2CLK_gc = (0x02<<0) ; 2 cycles
4568   .equ EBI_RPDLY_3CLK_gc = (0x03<<0) ; 3 cycles
4569   .equ EBI_RPDLY_4CLK_gc = (0x04<<0) ; 4 cycles
4570   .equ EBI_RPDLY_5CLK_gc = (0x05<<0) ; 5 cycles
4571   .equ EBI_RPDLY_6CLK_gc = (0x06<<0) ; 6 cycles
4572   .equ EBI_RPDLY_7CLK_gc = (0x07<<0) ; 7 cycles
4573
4574   ; SDRAM Write Recovery Delay
4575   .equ EBI_WRDLY_0CLK_gc = (0x00<<6) ; 0 cycles
4576   .equ EBI_WRDLY_1CLK_gc = (0x01<<6) ; 1 cycle
4577   .equ EBI_WRDLY_2CLK_gc = (0x02<<6) ; 2 cycles
4578   .equ EBI_WRDLY_3CLK_gc = (0x03<<6) ; 3 cycles
4579
4580   ; SDRAM Exit Self Refresh to Active Delay
4581   .equ EBI_ESRDLY_0CLK_gc = (0x00<<3) ; 0 cycles
4582   .equ EBI_ESRDLY_1CLK_gc = (0x01<<3) ; 1 cycle
4583   .equ EBI_ESRDLY_2CLK_gc = (0x02<<3) ; 2 cycles
4584   .equ EBI_ESRDLY_3CLK_gc = (0x03<<3) ; 3 cycles
4585   .equ EBI_ESRDLY_4CLK_gc = (0x04<<3) ; 4 cycles
4586   .equ EBI_ESRDLY_5CLK_gc = (0x05<<3) ; 5 cycles
4587   .equ EBI_ESRDLY_6CLK_gc = (0x06<<3) ; 6 cycles
4588   .equ EBI_ESRDLY_7CLK_gc = (0x07<<3) ; 7 cycles
4589
4590   ; SDRAM Row to Column Delay
4591   .equ EBI_ROWCOLDLY_0CLK_gc = (0x00<<0) ; 0 cycles
4592   .equ EBI_ROWCOLDLY_1CLK_gc = (0x01<<0) ; 1 cycle
4593   .equ EBI_ROWCOLDLY_2CLK_gc = (0x02<<0) ; 2 cycles
4594   .equ EBI_ROWCOLDLY_3CLK_gc = (0x03<<0) ; 3 cycles
4595   .equ EBI_ROWCOLDLY_4CLK_gc = (0x04<<0) ; 4 cycles
4596   .equ EBI_ROWCOLDLY_5CLK_gc = (0x05<<0) ; 5 cycles
4597   .equ EBI_ROWCOLDLY_6CLK_gc = (0x06<<0) ; 6 cycles
4598   .equ EBI_ROWCOLDLY_7CLK_gc = (0x07<<0) ; 7 cycles
4599
4600
4601   ;**************************************************************************
4602   ;** TWI - Two-Wire Interface
4603   ;**************************************************************************/
4604
4605   ; TWI_MASTER_CTRLA masks
4606   .equ TWI_MASTER_INTLVL_gm = 0xC0 ; Interrupt Level group mask
4607   .equ TWI_MASTER_INTLVL_gp = 6 ; Interrupt Level group position
4608   .equ TWI_MASTER_INTLVL0_bm = (1<<6) ; Interrupt Level bit 0 mask
4609   .equ TWI_MASTER_INTLVL0_bp = 6 ; Interrupt Level bit 0 position
4610   .equ TWI_MASTER_INTLVL1_bm = (1<<7) ; Interrupt Level bit 1 mask
```

```
4611   .equ TWI_MASTER_INTLVL1_bp = 7 ; Interrupt Level bit 1 position
4612   .equ TWI_MASTER_RIEN_bm = 0x20 ; Read Interrupt Enable bit mask
4613   .equ TWI_MASTER_RIEN_bp = 5 ; Read Interrupt Enable bit position
4614   .equ TWI_MASTER_WIEN_bm = 0x10 ; Write Interrupt Enable bit mask
4615   .equ TWI_MASTER_WIEN_bp = 4 ; Write Interrupt Enable bit position
4616   .equ TWI_MASTER_ENABLE_bm = 0x08 ; Enable TWI Master bit mask
4617   .equ TWI_MASTER_ENABLE_bp = 3 ; Enable TWI Master bit position
4618
4619   ; TWI_MASTER_CTRLB masks
4620   .equ TWI_MASTER_TIMEOUT_gm = 0x0C ; Inactive Bus Timeout group mask
4621   .equ TWI_MASTER_TIMEOUT_gp = 2 ; Inactive Bus Timeout group position
4622   .equ TWI_MASTER_TIMEOUT0_bm = (1<<2) ; Inactive Bus Timeout bit 0 mask
4623   .equ TWI_MASTER_TIMEOUT0_bp = 2 ; Inactive Bus Timeout bit 0 position
4624   .equ TWI_MASTER_TIMEOUT1_bm = (1<<3) ; Inactive Bus Timeout bit 1 mask
4625   .equ TWI_MASTER_TIMEOUT1_bp = 3 ; Inactive Bus Timeout bit 1 position
4626   .equ TWI_MASTER_QCEN_bm = 0x02 ; Quick Command Enable bit mask
4627   .equ TWI_MASTER_QCEN_bp = 1 ; Quick Command Enable bit position
4628   .equ TWI_MASTER_SMEN_bm = 0x01 ; Smart Mode Enable bit mask
4629   .equ TWI_MASTER_SMEN_bp = 0 ; Smart Mode Enable bit position
4630
4631   ; TWI_MASTER_CTRLC masks
4632   .equ TWI_MASTER_ACKACT_bm = 0x04 ; Acknowledge Action bit mask
4633   .equ TWI_MASTER_ACKACT_bp = 2 ; Acknowledge Action bit position
4634   .equ TWI_MASTER_CMD_gm = 0x03 ; Command group mask
4635   .equ TWI_MASTER_CMD_gp = 0 ; Command group position
4636   .equ TWI_MASTER_CMD0_bm = (1<<0) ; Command bit 0 mask
4637   .equ TWI_MASTER_CMD0_bp = 0 ; Command bit 0 position
4638   .equ TWI_MASTER_CMD1_bm = (1<<1) ; Command bit 1 mask
4639   .equ TWI_MASTER_CMD1_bp = 1 ; Command bit 1 position
4640
4641   ; TWI_MASTER_STATUS masks
4642   .equ TWI_MASTER_RIF_bm = 0x80 ; Read Interrupt Flag bit mask
4643   .equ TWI_MASTER_RIF_bp = 7 ; Read Interrupt Flag bit position
4644   .equ TWI_MASTER_WIF_bm = 0x40 ; Write Interrupt Flag bit mask
4645   .equ TWI_MASTER_WIF_bp = 6 ; Write Interrupt Flag bit position
4646   .equ TWI_MASTER_CLKHOLD_bm = 0x20 ; Clock Hold bit mask
4647   .equ TWI_MASTER_CLKHOLD_bp = 5 ; Clock Hold bit position
4648   .equ TWI_MASTER_RXACK_bm = 0x10 ; Received Acknowledge bit mask
4649   .equ TWI_MASTER_RXACK_bp = 4 ; Received Acknowledge bit position
4650   .equ TWI_MASTER_ARBLOST_bm = 0x08 ; Arbitration Lost bit mask
4651   .equ TWI_MASTER_ARBLOST_bp = 3 ; Arbitration Lost bit position
4652   .equ TWI_MASTER_BUSERR_bm = 0x04 ; Bus Error bit mask
4653   .equ TWI_MASTER_BUSERR_bp = 2 ; Bus Error bit position
4654   .equ TWI_MASTER_BUSSTATE_gm = 0x03 ; Bus State group mask
4655   .equ TWI_MASTER_BUSSTATE_gp = 0 ; Bus State group position
4656   .equ TWI_MASTER_BUSSTATE0_bm = (1<<0) ; Bus State bit 0 mask
4657   .equ TWI_MASTER_BUSSTATE0_bp = 0 ; Bus State bit 0 position
4658   .equ TWI_MASTER_BUSSTATE1_bm = (1<<1) ; Bus State bit 1 mask
4659   .equ TWI_MASTER_BUSSTATE1_bp = 1 ; Bus State bit 1 position
4660
4661   ; TWI_SLAVE_CTRLA masks
4662   .equ TWI_SLAVE_INTLVL_gm = 0xC0 ; Interrupt Level group mask
4663   .equ TWI_SLAVE_INTLVL_gp = 6 ; Interrupt Level group position
4664   .equ TWI_SLAVE_INTLVL0_bm = (1<<6) ; Interrupt Level bit 0 mask
4665   .equ TWI_SLAVE_INTLVL0_bp = 6 ; Interrupt Level bit 0 position
4666   .equ TWI_SLAVE_INTLVL1_bm = (1<<7) ; Interrupt Level bit 1 mask
4667   .equ TWI_SLAVE_INTLVL1_bp = 7 ; Interrupt Level bit 1 position
4668   .equ TWI_SLAVE_DIEN_bm = 0x20 ; Data Interrupt Enable bit mask
4669   .equ TWI_SLAVE_DIEN_bp = 5 ; Data Interrupt Enable bit position
4670   .equ TWI_SLAVE_APIEN_bm = 0x10 ; Address/Stop Interrupt Enable bit mask
4671   .equ TWI_SLAVE_APIEN_bp = 4 ; Address/Stop Interrupt Enable bit position
4672   .equ TWI_SLAVE_ENABLE_bm = 0x08 ; Enable TWI Slave bit mask
4673   .equ TWI_SLAVE_ENABLE_bp = 3 ; Enable TWI Slave bit position
4674   .equ TWI_SLAVE_PIEN_bm = 0x04 ; Stop Interrupt Enable bit mask
4675   .equ TWI_SLAVE_PIEN_bp = 2 ; Stop Interrupt Enable bit position
4676   .equ TWI_SLAVE_PMEN_bm = 0x02 ; Promiscuous Mode Enable bit mask
```

```
4677  .equ TWI_SLAVE_PMEN_bp = 1 ; Promiscuous Mode Enable bit position
4678  .equ TWI_SLAVE_SMEN_bm = 0x01 ; Smart Mode Enable bit mask
4679  .equ TWI_SLAVE_SMEN_bp = 0 ; Smart Mode Enable bit position
4680
4681  ; TWI_SLAVE_CTRLB masks
4682  .equ TWI_SLAVE_ACKACT_bm = 0x04 ; Acknowledge Action bit mask
4683  .equ TWI_SLAVE_ACKACT_bp = 2 ; Acknowledge Action bit position
4684  .equ TWI_SLAVE_CMD_gm = 0x03 ; Command group mask
4685  .equ TWI_SLAVE_CMD_gp = 0 ; Command group position
4686  .equ TWI_SLAVE_CMD0_bm = (1<<0) ; Command bit 0 mask
4687  .equ TWI_SLAVE_CMD0_bp = 0 ; Command bit 0 position
4688  .equ TWI_SLAVE_CMD1_bm = (1<<1) ; Command bit 1 mask
4689  .equ TWI_SLAVE_CMD1_bp = 1 ; Command bit 1 position
4690
4691  ; TWI_SLAVE_STATUS masks
4692  .equ TWI_SLAVE_DIF_bm = 0x80 ; Data Interrupt Flag bit mask
4693  .equ TWI_SLAVE_DIF_bp = 7 ; Data Interrupt Flag bit position
4694  .equ TWI_SLAVE_APIF_bm = 0x40 ; Address/Stop Interrupt Flag bit mask
4695  .equ TWI_SLAVE_APIF_bp = 6 ; Address/Stop Interrupt Flag bit position
4696  .equ TWI_SLAVE_CLKHOLD_bm = 0x20 ; Clock Hold bit mask
4697  .equ TWI_SLAVE_CLKHOLD_bp = 5 ; Clock Hold bit position
4698  .equ TWI_SLAVE_RXACK_bm = 0x10 ; Received Acknowledge bit mask
4699  .equ TWI_SLAVE_RXACK_bp = 4 ; Received Acknowledge bit position
4700  .equ TWI_SLAVE_COLL_bm = 0x08 ; Collision bit mask
4701  .equ TWI_SLAVE_COLL_bp = 3 ; Collision bit position
4702  .equ TWI_SLAVE_BUSERR_bm = 0x04 ; Bus Error bit mask
4703  .equ TWI_SLAVE_BUSERR_bp = 2 ; Bus Error bit position
4704  .equ TWI_SLAVE_DIR_bm = 0x02 ; Read/Write Direction bit mask
4705  .equ TWI_SLAVE_DIR_bp = 1 ; Read/Write Direction bit position
4706  .equ TWI_SLAVE_AP_bm = 0x01 ; Slave Address or Stop bit mask
4707  .equ TWI_SLAVE_AP_bp = 0 ; Slave Address or Stop bit position
4708
4709  ; TWI_SLAVE_ADDRMASK masks
4710  .equ TWI_SLAVE_ADDRMASK_gm = 0xFE ; Address Mask group mask
4711  .equ TWI_SLAVE_ADDRMASK_gp = 1 ; Address Mask group position
4712  .equ TWI_SLAVE_ADDRMASK0_bm = (1<<1) ; Address Mask bit 0 mask
4713  .equ TWI_SLAVE_ADDRMASK0_bp = 1 ; Address Mask bit 0 position
4714  .equ TWI_SLAVE_ADDRMASK1_bm = (1<<2) ; Address Mask bit 1 mask
4715  .equ TWI_SLAVE_ADDRMASK1_bp = 2 ; Address Mask bit 1 position
4716  .equ TWI_SLAVE_ADDRMASK2_bm = (1<<3) ; Address Mask bit 2 mask
4717  .equ TWI_SLAVE_ADDRMASK2_bp = 3 ; Address Mask bit 2 position
4718  .equ TWI_SLAVE_ADDRMASK3_bm = (1<<4) ; Address Mask bit 3 mask
4719  .equ TWI_SLAVE_ADDRMASK3_bp = 4 ; Address Mask bit 3 position
4720  .equ TWI_SLAVE_ADDRMASK4_bm = (1<<5) ; Address Mask bit 4 mask
4721  .equ TWI_SLAVE_ADDRMASK4_bp = 5 ; Address Mask bit 4 position
4722  .equ TWI_SLAVE_ADDRMASK5_bm = (1<<6) ; Address Mask bit 5 mask
4723  .equ TWI_SLAVE_ADDRMASK5_bp = 6 ; Address Mask bit 5 position
4724  .equ TWI_SLAVE_ADDRMASK6_bm = (1<<7) ; Address Mask bit 6 mask
4725  .equ TWI_SLAVE_ADDRMASK6_bp = 7 ; Address Mask bit 6 position
4726  .equ TWI_SLAVE_ADDREN_bm = 0x01 ; Address Enable bit mask
4727  .equ TWI_SLAVE_ADDREN_bp = 0 ; Address Enable bit position
4728
4729  ; TWI_CTRL masks
4730  .equ TWI_SDAHOLD_bm = 0x02 ; SDA Hold Time Enable bit mask
4731  .equ TWI_SDAHOLD_bp = 1 ; SDA Hold Time Enable bit position
4732  .equ TWI_EDIEN_bm = 0x01 ; External Driver Interface Enable bit mask
4733  .equ TWI_EDIEN_bp = 0 ; External Driver Interface Enable bit position
4734
4735  ; Master Interrupt Level
4736  .equ TWI_MASTER_INTLVL_OFF_gc = (0x00<<6) ; Interrupt Disabled
4737  .equ TWI_MASTER_INTLVL_LO_gc = (0x01<<6) ; Low Level
4738  .equ TWI_MASTER_INTLVL_MED_gc = (0x02<<6) ; Medium Level
4739  .equ TWI_MASTER_INTLVL_HI_gc = (0x03<<6) ; High Level
4740
4741  ; Inactive Timeout
4742  .equ TWI_MASTER_TIMEOUT_DISABLED_gc = (0x00<<2) ; Bus Timeout Disabled
```

```
4743  .equ TWI_MASTER_TIMEOUT_50US_gc = (0x01<<2) ; 50 Microseconds
4744  .equ TWI_MASTER_TIMEOUT_100US_gc = (0x02<<2) ; 100 Microseconds
4745  .equ TWI_MASTER_TIMEOUT_200US_gc = (0x03<<2) ; 200 Microseconds
4746
4747  ; Master Command
4748  .equ TWI_MASTER_CMD_NOACT_gc = (0x00<<0) ; No Action
4749  .equ TWI_MASTER_CMD_REPSTART_gc = (0x01<<0) ; Issue Repeated Start Condition
4750  .equ TWI_MASTER_CMD_RECVTRANS_gc = (0x02<<0) ; Receive or Transmit Data
4751  .equ TWI_MASTER_CMD_STOP_gc = (0x03<<0) ; Issue Stop Condition
4752
4753  ; Master Bus State
4754  .equ TWI_MASTER_BUSSTATE_UNKNOWN_gc = (0x00<<0) ; Unknown Bus State
4755  .equ TWI_MASTER_BUSSTATE_IDLE_gc = (0x01<<0) ; Bus is Idle
4756  .equ TWI_MASTER_BUSSTATE_OWNER_gc = (0x02<<0) ; This Module Controls The Bus
4757  .equ TWI_MASTER_BUSSTATE_BUSY_gc = (0x03<<0) ; The Bus is Busy
4758
4759  ; Slave Interrupt Level
4760  .equ TWI_SLAVE_INTLVL_OFF_gc = (0x00<<6) ; Interrupt Disabled
4761  .equ TWI_SLAVE_INTLVL_LO_gc = (0x01<<6) ; Low Level
4762  .equ TWI_SLAVE_INTLVL_MED_gc = (0x02<<6) ; Medium Level
4763  .equ TWI_SLAVE_INTLVL_HI_gc = (0x03<<6) ; High Level
4764
4765  ; Slave Command
4766  .equ TWI_SLAVE_CMD_NOACT_gc = (0x00<<0) ; No Action
4767  .equ TWI_SLAVE_CMD_COMPTRANS_gc = (0x02<<0) ; Used To Complete a Transaction
4768  .equ TWI_SLAVE_CMD_RESPONSE_gc = (0x03<<0) ; Used in Response to Address/Data Interrupt
4769
4770
4771  ;**************************************************************************
4772  ;** USB - USB
4773  ;**************************************************************************/
4774
4775  ; USB_EP_STATUS masks
4776  .equ USB_EP_STALLF_bm = 0x80 ; Endpoint Stall Flag bit mask
4777  .equ USB_EP_STALLF_bp = 7 ; Endpoint Stall Flag bit position
4778  .equ USB_EP_CRC_bm = 0x80 ; CRC Error Flag bit mask
4779  .equ USB_EP_CRC_bp = 7 ; CRC Error Flag bit position
4780  .equ USB_EP_UNF_bm = 0x40 ; Underflow Enpoint FLag bit mask
4781  .equ USB_EP_UNF_bp = 6 ; Underflow Enpoint FLag bit position
4782  .equ USB_EP_OVF_bm = 0x40 ; Overflow Enpoint Flag for Output Endpoints bit mask
4783  .equ USB_EP_OVF_bp = 6 ; Overflow Enpoint Flag for Output Endpoints bit position
4784  .equ USB_EP_TRNCOMPL0_bm = 0x20 ; Transaction Complete 0 Flag bit mask
4785  .equ USB_EP_TRNCOMPL0_bp = 5 ; Transaction Complete 0 Flag bit position
4786  .equ USB_EP_TRNCOMPL1_bm = 0x10 ; Transaction Complete 1 Flag bit mask
4787  .equ USB_EP_TRNCOMPL1_bp = 4 ; Transaction Complete 1 Flag bit position
4788  .equ USB_EP_SETUP_bm = 0x10 ; SETUP Transaction Complete Flag bit mask
4789  .equ USB_EP_SETUP_bp = 4 ; SETUP Transaction Complete Flag bit position
4790  .equ USB_EP_BANK_bm = 0x08 ; Bank Select bit mask
4791  .equ USB_EP_BANK_bp = 3 ; Bank Select bit position
4792  .equ USB_EP_BUSNACK1_bm = 0x04 ; Data Buffer 1 Not Acknowledge bit mask
4793  .equ USB_EP_BUSNACK1_bp = 2 ; Data Buffer 1 Not Acknowledge bit position
4794  .equ USB_EP_BUSNACK0_bm = 0x02 ; Data Buffer 0 Not Acknowledge bit mask
4795  .equ USB_EP_BUSNACK0_bp = 1 ; Data Buffer 0 Not Acknowledge bit position
4796  .equ USB_EP_TOGGLE_bm = 0x01 ; Data Toggle bit mask
4797  .equ USB_EP_TOGGLE_bp = 0 ; Data Toggle bit position
4798
4799  ; USB_EP_CTRL masks
4800  .equ USB_EP_TYPE_gm = 0xC0 ; Endpoint Type group mask
4801  .equ USB_EP_TYPE_gp = 6 ; Endpoint Type group position
4802  .equ USB_EP_TYPE0_bm = (1<<6) ; Endpoint Type bit 0 mask
4803  .equ USB_EP_TYPE0_bp = 6 ; Endpoint Type bit 0 position
4804  .equ USB_EP_TYPE1_bm = (1<<7) ; Endpoint Type bit 1 mask
4805  .equ USB_EP_TYPE1_bp = 7 ; Endpoint Type bit 1 position
4806  .equ USB_EP_MULTIPKT_bm = 0x20 ; Multi Packet Transfer Enable bit mask
4807  .equ USB_EP_MULTIPKT_bp = 5 ; Multi Packet Transfer Enable bit position
4808  .equ USB_EP_PINGPONG_bm = 0x10 ; Ping-Pong Enable bit mask
```

```
4809   .equ USB_EP_PINGPONG_bp = 4 ; Ping-Pong Enable bit position
4810   .equ USB_EP_INTDSBL_bm = 0x08 ; Interrupt Disable bit mask
4811   .equ USB_EP_INTDSBL_bp = 3 ; Interrupt Disable bit position
4812   .equ USB_EP_STALL_bm = 0x04 ; Data Stall bit mask
4813   .equ USB_EP_STALL_bp = 2 ; Data Stall bit position
4814   .equ USB_EP_BUFSIZE_gm = 0x07 ; Data Buffer Size group mask
4815   .equ USB_EP_BUFSIZE_gp = 0 ; Data Buffer Size group position
4816   .equ USB_EP_BUFSIZE0_bm = (1<<0) ; Data Buffer Size bit 0 mask
4817   .equ USB_EP_BUFSIZE0_bp = 0 ; Data Buffer Size bit 0 position
4818   .equ USB_EP_BUFSIZE1_bm = (1<<1) ; Data Buffer Size bit 1 mask
4819   .equ USB_EP_BUFSIZE1_bp = 1 ; Data Buffer Size bit 1 position
4820   .equ USB_EP_BUFSIZE2_bm = (1<<2) ; Data Buffer Size bit 2 mask
4821   .equ USB_EP_BUFSIZE2_bp = 2 ; Data Buffer Size bit 2 position
4822
4823   ; USB_EP_CNT masks
4824   .equ USB_EP_ZLP_bm = 0x8000 ; Zero Length Packet bit mask
4825   .equ USB_EP_ZLP_bp = 15 ; Zero Length Packet bit position
4826
4827   ; USB_CTRLA masks
4828   .equ USB_ENABLE_bm = 0x80 ; USB Enable bit mask
4829   .equ USB_ENABLE_bp = 7 ; USB Enable bit position
4830   .equ USB_SPEED_bm = 0x40 ; Speed Select bit mask
4831   .equ USB_SPEED_bp = 6 ; Speed Select bit position
4832   .equ USB_FIFOEN_bm = 0x20 ; USB FIFO Enable bit mask
4833   .equ USB_FIFOEN_bp = 5 ; USB FIFO Enable bit position
4834   .equ USB_STFRNUM_bm = 0x10 ; Store Frame Number Enable bit mask
4835   .equ USB_STFRNUM_bp = 4 ; Store Frame Number Enable bit position
4836   .equ USB_MAXEP_gm = 0x0F ; Maximum Endpoint Addresses group mask
4837   .equ USB_MAXEP_gp = 0 ; Maximum Endpoint Addresses group position
4838   .equ USB_MAXEP0_bm = (1<<0) ; Maximum Endpoint Addresses bit 0 mask
4839   .equ USB_MAXEP0_bp = 0 ; Maximum Endpoint Addresses bit 0 position
4840   .equ USB_MAXEP1_bm = (1<<1) ; Maximum Endpoint Addresses bit 1 mask
4841   .equ USB_MAXEP1_bp = 1 ; Maximum Endpoint Addresses bit 1 position
4842   .equ USB_MAXEP2_bm = (1<<2) ; Maximum Endpoint Addresses bit 2 mask
4843   .equ USB_MAXEP2_bp = 2 ; Maximum Endpoint Addresses bit 2 position
4844   .equ USB_MAXEP3_bm = (1<<3) ; Maximum Endpoint Addresses bit 3 mask
4845   .equ USB_MAXEP3_bp = 3 ; Maximum Endpoint Addresses bit 3 position
4846
4847   ; USB_CTRLB masks
4848   .equ USB_PULLRST_bm = 0x10 ; Pull during Reset bit mask
4849   .equ USB_PULLRST_bp = 4 ; Pull during Reset bit position
4850   .equ USB_RWAKEUP_bm = 0x04 ; Remote Wake-up bit mask
4851   .equ USB_RWAKEUP_bp = 2 ; Remote Wake-up bit position
4852   .equ USB_GNACK_bm = 0x02 ; Global NACK bit mask
4853   .equ USB_GNACK_bp = 1 ; Global NACK bit position
4854   .equ USB_ATTACH_bm = 0x01 ; Attach bit mask
4855   .equ USB_ATTACH_bp = 0 ; Attach bit position
4856
4857   ; USB_STATUS masks
4858   .equ USB_URESUME_bm = 0x08 ; Upstream Resume bit mask
4859   .equ USB_URESUME_bp = 3 ; Upstream Resume bit position
4860   .equ USB_RESUME_bm = 0x04 ; Resume bit mask
4861   .equ USB_RESUME_bp = 2 ; Resume bit position
4862   .equ USB_SUSPEND_bm = 0x02 ; Bus Suspended bit mask
4863   .equ USB_SUSPEND_bp = 1 ; Bus Suspended bit position
4864   .equ USB_BUSRST_bm = 0x01 ; Bus Reset bit mask
4865   .equ USB_BUSRST_bp = 0 ; Bus Reset bit position
4866
4867   ; USB_ADDR masks
4868   .equ USB_ADDR_gm = 0x7F ; Device Address group mask
4869   .equ USB_ADDR_gp = 0 ; Device Address group position
4870   .equ USB_ADDR0_bm = (1<<0) ; Device Address bit 0 mask
4871   .equ USB_ADDR0_bp = 0 ; Device Address bit 0 position
4872   .equ USB_ADDR1_bm = (1<<1) ; Device Address bit 1 mask
4873   .equ USB_ADDR1_bp = 1 ; Device Address bit 1 position
4874   .equ USB_ADDR2_bm = (1<<2) ; Device Address bit 2 mask
```

```
4875  .equ USB_ADDR2_bp = 2 ; Device Address bit 2 position
4876  .equ USB_ADDR3_bm = (1<<3) ; Device Address bit 3 mask
4877  .equ USB_ADDR3_bp = 3 ; Device Address bit 3 position
4878  .equ USB_ADDR4_bm = (1<<4) ; Device Address bit 4 mask
4879  .equ USB_ADDR4_bp = 4 ; Device Address bit 4 position
4880  .equ USB_ADDR5_bm = (1<<5) ; Device Address bit 5 mask
4881  .equ USB_ADDR5_bp = 5 ; Device Address bit 5 position
4882  .equ USB_ADDR6_bm = (1<<6) ; Device Address bit 6 mask
4883  .equ USB_ADDR6_bp = 6 ; Device Address bit 6 position
4884
4885  ; USB_FIFOWP masks
4886  .equ USB_FIFOWP_gm = 0x1F ; FIFO Write Pointer group mask
4887  .equ USB_FIFOWP_gp = 0 ; FIFO Write Pointer group position
4888  .equ USB_FIFOWP0_bm = (1<<0) ; FIFO Write Pointer bit 0 mask
4889  .equ USB_FIFOWP0_bp = 0 ; FIFO Write Pointer bit 0 position
4890  .equ USB_FIFOWP1_bm = (1<<1) ; FIFO Write Pointer bit 1 mask
4891  .equ USB_FIFOWP1_bp = 1 ; FIFO Write Pointer bit 1 position
4892  .equ USB_FIFOWP2_bm = (1<<2) ; FIFO Write Pointer bit 2 mask
4893  .equ USB_FIFOWP2_bp = 2 ; FIFO Write Pointer bit 2 position
4894  .equ USB_FIFOWP3_bm = (1<<3) ; FIFO Write Pointer bit 3 mask
4895  .equ USB_FIFOWP3_bp = 3 ; FIFO Write Pointer bit 3 position
4896  .equ USB_FIFOWP4_bm = (1<<4) ; FIFO Write Pointer bit 4 mask
4897  .equ USB_FIFOWP4_bp = 4 ; FIFO Write Pointer bit 4 position
4898
4899  ; USB_FIFORP masks
4900  .equ USB_FIFORP_gm = 0x1F ; FIFO Read Pointer group mask
4901  .equ USB_FIFORP_gp = 0 ; FIFO Read Pointer group position
4902  .equ USB_FIFORP0_bm = (1<<0) ; FIFO Read Pointer bit 0 mask
4903  .equ USB_FIFORP0_bp = 0 ; FIFO Read Pointer bit 0 position
4904  .equ USB_FIFORP1_bm = (1<<1) ; FIFO Read Pointer bit 1 mask
4905  .equ USB_FIFORP1_bp = 1 ; FIFO Read Pointer bit 1 position
4906  .equ USB_FIFORP2_bm = (1<<2) ; FIFO Read Pointer bit 2 mask
4907  .equ USB_FIFORP2_bp = 2 ; FIFO Read Pointer bit 2 position
4908  .equ USB_FIFORP3_bm = (1<<3) ; FIFO Read Pointer bit 3 mask
4909  .equ USB_FIFORP3_bp = 3 ; FIFO Read Pointer bit 3 position
4910  .equ USB_FIFORP4_bm = (1<<4) ; FIFO Read Pointer bit 4 mask
4911  .equ USB_FIFORP4_bp = 4 ; FIFO Read Pointer bit 4 position
4912
4913  ; USB_INTCTRLA masks
4914  .equ USB_SOFIE_bm = 0x80 ; Start Of Frame Interrupt Enable bit mask
4915  .equ USB_SOFIE_bp = 7 ; Start Of Frame Interrupt Enable bit position
4916  .equ USB_BUSEVIE_bm = 0x40 ; Bus Event Interrupt Enable bit mask
4917  .equ USB_BUSEVIE_bp = 6 ; Bus Event Interrupt Enable bit position
4918  .equ USB_BUSERRIE_bm = 0x20 ; Bus Error Interrupt Enable bit mask
4919  .equ USB_BUSERRIE_bp = 5 ; Bus Error Interrupt Enable bit position
4920  .equ USB_STALLIE_bm = 0x10 ; STALL Interrupt Enable bit mask
4921  .equ USB_STALLIE_bp = 4 ; STALL Interrupt Enable bit position
4922  .equ USB_INTLVL_gm = 0x03 ; Interrupt Level group mask
4923  .equ USB_INTLVL_gp = 0 ; Interrupt Level group position
4924  .equ USB_INTLVL0_bm = (1<<0) ; Interrupt Level bit 0 mask
4925  .equ USB_INTLVL0_bp = 0 ; Interrupt Level bit 0 position
4926  .equ USB_INTLVL1_bm = (1<<1) ; Interrupt Level bit 1 mask
4927  .equ USB_INTLVL1_bp = 1 ; Interrupt Level bit 1 position
4928
4929  ; USB_INTCTRLB masks
4930  .equ USB_TRNIE_bm = 0x02 ; Transaction Complete Interrupt Enable bit mask
4931  .equ USB_TRNIE_bp = 1 ; Transaction Complete Interrupt Enable bit position
4932  .equ USB_SETUPIE_bm = 0x01 ; SETUP Transaction Complete Interrupt Enable bit mask
4933  .equ USB_SETUPIE_bp = 0 ; SETUP Transaction Complete Interrupt Enable bit position
4934
4935  ; USB_INTFLAGSACLR masks
4936  .equ USB_SOFIF_bm = 0x80 ; Start Of Frame Interrupt Flag bit mask
4937  .equ USB_SOFIF_bp = 7 ; Start Of Frame Interrupt Flag bit position
4938  .equ USB_SUSPENDIF_bm = 0x40 ; Suspend Interrupt Flag bit mask
4939  .equ USB_SUSPENDIF_bp = 6 ; Suspend Interrupt Flag bit position
4940  .equ USB_RESUMEIF_bm = 0x20 ; Resume Interrupt Flag bit mask
```

```
4941  .equ USB_RESUMEIF_bp = 5 ; Resume Interrupt Flag bit position
4942  .equ USB_RSTIF_bm = 0x10 ; Reset Interrupt Flag bit mask
4943  .equ USB_RSTIF_bp = 4 ; Reset Interrupt Flag bit position
4944  .equ USB_CRCIF_bm = 0x08 ; Isochronous CRC Error Interrupt Flag bit mask
4945  .equ USB_CRCIF_bp = 3 ; Isochronous CRC Error Interrupt Flag bit position
4946  .equ USB_UNFIF_bm = 0x04 ; Underflow Interrupt Flag bit mask
4947  .equ USB_UNFIF_bp = 2 ; Underflow Interrupt Flag bit position
4948  .equ USB_OVFIF_bm = 0x02 ; Overflow Interrupt Flag bit mask
4949  .equ USB_OVFIF_bp = 1 ; Overflow Interrupt Flag bit position
4950  .equ USB_STALLIF_bm = 0x01 ; STALL Interrupt Flag bit mask
4951  .equ USB_STALLIF_bp = 0 ; STALL Interrupt Flag bit position
4952
4953  ; USB_INTFLAGSASET masks
4954  ; Masks for SOFIF aready defined
4955  ; Masks for SUSPENDIF aready defined
4956  ; Masks for RESUMEIF aready defined
4957  ; Masks for RSTIF aready defined
4958  ; Masks for CRCIF aready defined
4959  ; Masks for UNFIF aready defined
4960  ; Masks for OVFIF aready defined
4961  ; Masks for STALLIF aready defined
4962
4963  ; USB_INTFLAGSBCLR masks
4964  .equ USB_TRNIF_bm = 0x02 ; Transaction Complete Interrupt Flag bit mask
4965  .equ USB_TRNIF_bp = 1 ; Transaction Complete Interrupt Flag bit position
4966  .equ USB_SETUPIF_bm = 0x01 ; SETUP Transaction Complete Interrupt Flag bit mask
4967  .equ USB_SETUPIF_bp = 0 ; SETUP Transaction Complete Interrupt Flag bit position
4968
4969  ; USB_INTFLAGSBSET masks
4970  ; Masks for TRNIF aready defined
4971  ; Masks for SETUPIF aready defined
4972
4973  ; Interrupt level
4974  .equ USB_INTLVL_OFF_gc = (0x00<<0) ; Interrupt disabled
4975  .equ USB_INTLVL_LO_gc = (0x01<<0) ; Low level
4976  .equ USB_INTLVL_MED_gc = (0x02<<0) ; Medium level
4977  .equ USB_INTLVL_HI_gc = (0x03<<0) ; High level
4978
4979  ; USB Endpoint Type
4980  .equ USB_EP_TYPE_DISABLE_gc = (0x00<<6) ; Endpoint Disabled
4981  .equ USB_EP_TYPE_CONTROL_gc = (0x01<<6) ; Control
4982  .equ USB_EP_TYPE_BULK_gc = (0x02<<6) ; Bulk/Interrupt
4983  .equ USB_EP_TYPE_ISOCHRONOUS_gc = (0x03<<6) ; Isochronous
4984
4985  ; USB Endpoint Buffersize
4986  .equ USB_EP_BUFSIZE_8_gc = (0x00<<0) ; 8 bytes buffer size
4987  .equ USB_EP_BUFSIZE_16_gc = (0x01<<0) ; 16 bytes buffer size
4988  .equ USB_EP_BUFSIZE_32_gc = (0x02<<0) ; 32 bytes buffer size
4989  .equ USB_EP_BUFSIZE_64_gc = (0x03<<0) ; 64 bytes buffer size
4990  .equ USB_EP_BUFSIZE_128_gc = (0x04<<0) ; 128 bytes buffer size
4991  .equ USB_EP_BUFSIZE_256_gc = (0x05<<0) ; 256 bytes buffer size
4992  .equ USB_EP_BUFSIZE_512_gc = (0x06<<0) ; 512 bytes buffer size
4993  .equ USB_EP_BUFSIZE_1023_gc = (0x07<<0) ; 1023 bytes buffer size
4994
4995
4996  ;**************************************************************************
4997  ;** PORT - I/O Port Configuration
4998  ;*************************************************************************/
4999
5000  ; PORT_INTCTRL masks
5001  .equ PORT_INT1LVL_gm = 0x0C ; Port Interrupt 1 Level group mask
5002  .equ PORT_INT1LVL_gp = 2 ; Port Interrupt 1 Level group position
5003  .equ PORT_INT1LVL0_bm = (1<<2) ; Port Interrupt 1 Level bit 0 mask
5004  .equ PORT_INT1LVL0_bp = 2 ; Port Interrupt 1 Level bit 0 position
5005  .equ PORT_INT1LVL1_bm = (1<<3) ; Port Interrupt 1 Level bit 1 mask
5006  .equ PORT_INT1LVL1_bp = 3 ; Port Interrupt 1 Level bit 1 position
```

```
5007  .equ PORT_INT0LVL_gm = 0x03 ; Port Interrupt 0 Level group mask
5008  .equ PORT_INT0LVL_gp = 0 ; Port Interrupt 0 Level group position
5009  .equ PORT_INT0LVL0_bm = (1<<0) ; Port Interrupt 0 Level bit 0 mask
5010  .equ PORT_INT0LVL0_bp = 0 ; Port Interrupt 0 Level bit 0 position
5011  .equ PORT_INT0LVL1_bm = (1<<1) ; Port Interrupt 0 Level bit 1 mask
5012  .equ PORT_INT0LVL1_bp = 1 ; Port Interrupt 0 Level bit 1 position
5013
5014  ; PORT_INTFLAGS masks
5015  .equ PORT_INT1IF_bm = 0x02 ; Port Interrupt 1 Flag bit mask
5016  .equ PORT_INT1IF_bp = 1 ; Port Interrupt 1 Flag bit position
5017  .equ PORT_INT0IF_bm = 0x01 ; Port Interrupt 0 Flag bit mask
5018  .equ PORT_INT0IF_bp = 0 ; Port Interrupt 0 Flag bit position
5019
5020  ; PORT_REMAP masks
5021  .equ PORT_SPI_bm = 0x20 ; SPI bit mask
5022  .equ PORT_SPI_bp = 5 ; SPI bit position
5023  .equ PORT_USART0_bm = 0x10 ; USART0 bit mask
5024  .equ PORT_USART0_bp = 4 ; USART0 bit position
5025  .equ PORT_TC0D_bm = 0x08 ; Timer/Counter 0 Output Compare D bit mask
5026  .equ PORT_TC0D_bp = 3 ; Timer/Counter 0 Output Compare D bit position
5027  .equ PORT_TC0C_bm = 0x04 ; Timer/Counter 0 Output Compare C bit mask
5028  .equ PORT_TC0C_bp = 2 ; Timer/Counter 0 Output Compare C bit position
5029  .equ PORT_TC0B_bm = 0x02 ; Timer/Counter 0 Output Compare B bit mask
5030  .equ PORT_TC0B_bp = 1 ; Timer/Counter 0 Output Compare B bit position
5031  .equ PORT_TC0A_bm = 0x01 ; Timer/Counter 0 Output Compare A bit mask
5032  .equ PORT_TC0A_bp = 0 ; Timer/Counter 0 Output Compare A bit position
5033
5034  ; PORT_PIN0CTRL masks
5035  .equ PORT_SRLEN_bm = 0x80 ; Slew Rate Enable bit mask
5036  .equ PORT_SRLEN_bp = 7 ; Slew Rate Enable bit position
5037  .equ PORT_INVEN_bm = 0x40 ; Inverted I/O Enable bit mask
5038  .equ PORT_INVEN_bp = 6 ; Inverted I/O Enable bit position
5039  .equ PORT_OPC_gm = 0x38 ; Output/Pull Configuration group mask
5040  .equ PORT_OPC_gp = 3 ; Output/Pull Configuration group position
5041  .equ PORT_OPC0_bm = (1<<3) ; Output/Pull Configuration bit 0 mask
5042  .equ PORT_OPC0_bp = 3 ; Output/Pull Configuration bit 0 position
5043  .equ PORT_OPC1_bm = (1<<4) ; Output/Pull Configuration bit 1 mask
5044  .equ PORT_OPC1_bp = 4 ; Output/Pull Configuration bit 1 position
5045  .equ PORT_OPC2_bm = (1<<5) ; Output/Pull Configuration bit 2 mask
5046  .equ PORT_OPC2_bp = 5 ; Output/Pull Configuration bit 2 position
5047  .equ PORT_ISC_gm = 0x07 ; Input/Sense Configuration group mask
5048  .equ PORT_ISC_gp = 0 ; Input/Sense Configuration group position
5049  .equ PORT_ISC0_bm = (1<<0) ; Input/Sense Configuration bit 0 mask
5050  .equ PORT_ISC0_bp = 0 ; Input/Sense Configuration bit 0 position
5051  .equ PORT_ISC1_bm = (1<<1) ; Input/Sense Configuration bit 1 mask
5052  .equ PORT_ISC1_bp = 1 ; Input/Sense Configuration bit 1 position
5053  .equ PORT_ISC2_bm = (1<<2) ; Input/Sense Configuration bit 2 mask
5054  .equ PORT_ISC2_bp = 2 ; Input/Sense Configuration bit 2 position
5055
5056  ; PORT_PIN1CTRL masks
5057  ; Masks for SRLEN aready defined
5058  ; Masks for INVEN aready defined
5059  ; Masks for OPC aready defined
5060  ; Masks for ISC aready defined
5061
5062  ; PORT_PIN2CTRL masks
5063  ; Masks for SRLEN aready defined
5064  ; Masks for INVEN aready defined
5065  ; Masks for OPC aready defined
5066  ; Masks for ISC aready defined
5067
5068  ; PORT_PIN3CTRL masks
5069  ; Masks for SRLEN aready defined
5070  ; Masks for INVEN aready defined
5071  ; Masks for OPC aready defined
5072  ; Masks for ISC aready defined
```

```
5073
5074   ; PORT_PIN4CTRL masks
5075   ; Masks for SRLEN aready defined
5076   ; Masks for INVEN aready defined
5077   ; Masks for OPC aready defined
5078   ; Masks for ISC aready defined
5079
5080   ; PORT_PIN5CTRL masks
5081   ; Masks for SRLEN aready defined
5082   ; Masks for INVEN aready defined
5083   ; Masks for OPC aready defined
5084   ; Masks for ISC aready defined
5085
5086   ; PORT_PIN6CTRL masks
5087   ; Masks for SRLEN aready defined
5088   ; Masks for INVEN aready defined
5089   ; Masks for OPC aready defined
5090   ; Masks for ISC aready defined
5091
5092   ; PORT_PIN7CTRL masks
5093   ; Masks for SRLEN aready defined
5094   ; Masks for INVEN aready defined
5095   ; Masks for OPC aready defined
5096   ; Masks for ISC aready defined
5097
5098   ; Port Interrupt 0 Level
5099   .equ PORT_INT0LVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
5100   .equ PORT_INT0LVL_LO_gc = (0x01<<0) ; Low Level
5101   .equ PORT_INT0LVL_MED_gc = (0x02<<0) ; Medium Level
5102   .equ PORT_INT0LVL_HI_gc = (0x03<<0) ; High Level
5103
5104   ; Port Interrupt 1 Level
5105   .equ PORT_INT1LVL_OFF_gc = (0x00<<2) ; Interrupt Disabled
5106   .equ PORT_INT1LVL_LO_gc = (0x01<<2) ; Low Level
5107   .equ PORT_INT1LVL_MED_gc = (0x02<<2) ; Medium Level
5108   .equ PORT_INT1LVL_HI_gc = (0x03<<2) ; High Level
5109
5110   ; Output/Pull Configuration
5111   .equ PORT_OPC_TOTEM_gc = (0x00<<3) ; Totempole
5112   .equ PORT_OPC_BUSKEEPER_gc = (0x01<<3) ; Totempole w/ Bus keeper on Input and Output
5113   .equ PORT_OPC_PULLDOWN_gc = (0x02<<3) ; Totempole w/ Pull-down on Input
5114   .equ PORT_OPC_PULLUP_gc = (0x03<<3) ; Totempole w/ Pull-up on Input
5115   .equ PORT_OPC_WIREDOR_gc = (0x04<<3) ; Wired OR
5116   .equ PORT_OPC_WIREDAND_gc = (0x05<<3) ; Wired AND
5117   .equ PORT_OPC_WIREDORPULL_gc = (0x06<<3) ; Wired OR w/ Pull-down
5118   .equ PORT_OPC_WIREDANDPULL_gc = (0x07<<3) ; Wired AND w/ Pull-up
5119
5120   ; Input/Sense Configuration
5121   .equ PORT_ISC_BOTHEDGES_gc = (0x00<<0) ; Sense Both Edges
5122   .equ PORT_ISC_RISING_gc = (0x01<<0) ; Sense Rising Edge
5123   .equ PORT_ISC_FALLING_gc = (0x02<<0) ; Sense Falling Edge
5124   .equ PORT_ISC_LEVEL_gc = (0x03<<0) ; Sense Level (Transparent For Events)
5125   .equ PORT_ISC_INPUT_DISABLE_gc = (0x07<<0) ; Disable Digital Input Buffer
5126
5127
5128   ;**************************************************************************
5129   ;** TC - 16-bit Timer/Counter With PWM
5130   ;**************************************************************************/
5131
5132   ; TC0_CTRLA masks
5133   .equ TC0_CLKSEL_gm = 0x0F ; Clock Selection group mask
5134   .equ TC0_CLKSEL_gp = 0 ; Clock Selection group position
5135   .equ TC0_CLKSEL0_bm = (1<<0) ; Clock Selection bit 0 mask
5136   .equ TC0_CLKSEL0_bp = 0 ; Clock Selection bit 0 position
5137   .equ TC0_CLKSEL1_bm = (1<<1) ; Clock Selection bit 1 mask
5138   .equ TC0_CLKSEL1_bp = 1 ; Clock Selection bit 1 position
```

```
5139  .equ TC0_CLKSEL2_bm = (1<<2) ; Clock Selection bit 2 mask
5140  .equ TC0_CLKSEL2_bp = 2 ; Clock Selection bit 2 position
5141  .equ TC0_CLKSEL3_bm = (1<<3) ; Clock Selection bit 3 mask
5142  .equ TC0_CLKSEL3_bp = 3 ; Clock Selection bit 3 position
5143
5144  ; TC0_CTRLB masks
5145  .equ TC0_CCDEN_bm = 0x80 ; Compare or Capture D Enable bit mask
5146  .equ TC0_CCDEN_bp = 7 ; Compare or Capture D Enable bit position
5147  .equ TC0_CCCEN_bm = 0x40 ; Compare or Capture C Enable bit mask
5148  .equ TC0_CCCEN_bp = 6 ; Compare or Capture C Enable bit position
5149  .equ TC0_CCBEN_bm = 0x20 ; Compare or Capture B Enable bit mask
5150  .equ TC0_CCBEN_bp = 5 ; Compare or Capture B Enable bit position
5151  .equ TC0_CCAEN_bm = 0x10 ; Compare or Capture A Enable bit mask
5152  .equ TC0_CCAEN_bp = 4 ; Compare or Capture A Enable bit position
5153  .equ TC0_WGMODE_gm = 0x07 ; Waveform generation mode group mask
5154  .equ TC0_WGMODE_gp = 0 ; Waveform generation mode group position
5155  .equ TC0_WGMODE0_bm = (1<<0) ; Waveform generation mode bit 0 mask
5156  .equ TC0_WGMODE0_bp = 0 ; Waveform generation mode bit 0 position
5157  .equ TC0_WGMODE1_bm = (1<<1) ; Waveform generation mode bit 1 mask
5158  .equ TC0_WGMODE1_bp = 1 ; Waveform generation mode bit 1 position
5159  .equ TC0_WGMODE2_bm = (1<<2) ; Waveform generation mode bit 2 mask
5160  .equ TC0_WGMODE2_bp = 2 ; Waveform generation mode bit 2 position
5161
5162  ; TC0_CTRLC masks
5163  .equ TC0_CMPD_bm = 0x08 ; Compare D Output Value bit mask
5164  .equ TC0_CMPD_bp = 3 ; Compare D Output Value bit position
5165  .equ TC0_CMPC_bm = 0x04 ; Compare C Output Value bit mask
5166  .equ TC0_CMPC_bp = 2 ; Compare C Output Value bit position
5167  .equ TC0_CMPB_bm = 0x02 ; Compare B Output Value bit mask
5168  .equ TC0_CMPB_bp = 1 ; Compare B Output Value bit position
5169  .equ TC0_CMPA_bm = 0x01 ; Compare A Output Value bit mask
5170  .equ TC0_CMPA_bp = 0 ; Compare A Output Value bit position
5171
5172  ; TC0_CTRLD masks
5173  .equ TC0_EVACT_gm = 0xE0 ; Event Action group mask
5174  .equ TC0_EVACT_gp = 5 ; Event Action group position
5175  .equ TC0_EVACT0_bm = (1<<5) ; Event Action bit 0 mask
5176  .equ TC0_EVACT0_bp = 5 ; Event Action bit 0 position
5177  .equ TC0_EVACT1_bm = (1<<6) ; Event Action bit 1 mask
5178  .equ TC0_EVACT1_bp = 6 ; Event Action bit 1 position
5179  .equ TC0_EVACT2_bm = (1<<7) ; Event Action bit 2 mask
5180  .equ TC0_EVACT2_bp = 7 ; Event Action bit 2 position
5181  .equ TC0_EVDLY_bm = 0x10 ; Event Delay bit mask
5182  .equ TC0_EVDLY_bp = 4 ; Event Delay bit position
5183  .equ TC0_EVSEL_gm = 0x0F ; Event Source Select group mask
5184  .equ TC0_EVSEL_gp = 0 ; Event Source Select group position
5185  .equ TC0_EVSEL0_bm = (1<<0) ; Event Source Select bit 0 mask
5186  .equ TC0_EVSEL0_bp = 0 ; Event Source Select bit 0 position
5187  .equ TC0_EVSEL1_bm = (1<<1) ; Event Source Select bit 1 mask
5188  .equ TC0_EVSEL1_bp = 1 ; Event Source Select bit 1 position
5189  .equ TC0_EVSEL2_bm = (1<<2) ; Event Source Select bit 2 mask
5190  .equ TC0_EVSEL2_bp = 2 ; Event Source Select bit 2 position
5191  .equ TC0_EVSEL3_bm = (1<<3) ; Event Source Select bit 3 mask
5192  .equ TC0_EVSEL3_bp = 3 ; Event Source Select bit 3 position
5193
5194  ; TC0_CTRLE masks
5195  .equ TC0_BYTEM_gm = 0x03 ; Byte Mode group mask
5196  .equ TC0_BYTEM_gp = 0 ; Byte Mode group position
5197  .equ TC0_BYTEM0_bm = (1<<0) ; Byte Mode bit 0 mask
5198  .equ TC0_BYTEM0_bp = 0 ; Byte Mode bit 0 position
5199  .equ TC0_BYTEM1_bm = (1<<1) ; Byte Mode bit 1 mask
5200  .equ TC0_BYTEM1_bp = 1 ; Byte Mode bit 1 position
5201
5202  ; TC0_INTCTRLA masks
5203  .equ TC0_ERRINTLVL_gm = 0x0C ; Error Interrupt Level group mask
5204  .equ TC0_ERRINTLVL_gp = 2 ; Error Interrupt Level group position
```

```
.equ TC0_ERRINTLVL0_bm = (1<<2) ; Error Interrupt Level bit 0 mask
.equ TC0_ERRINTLVL0_bp = 2 ; Error Interrupt Level bit 0 position
.equ TC0_ERRINTLVL1_bm = (1<<3) ; Error Interrupt Level bit 1 mask
.equ TC0_ERRINTLVL1_bp = 3 ; Error Interrupt Level bit 1 position
.equ TC0_OVFINTLVL_gm = 0x03 ; Overflow interrupt level group mask
.equ TC0_OVFINTLVL_gp = 0 ; Overflow interrupt level group position
.equ TC0_OVFINTLVL0_bm = (1<<0) ; Overflow interrupt level bit 0 mask
.equ TC0_OVFINTLVL0_bp = 0 ; Overflow interrupt level bit 0 position
.equ TC0_OVFINTLVL1_bm = (1<<1) ; Overflow interrupt level bit 1 mask
.equ TC0_OVFINTLVL1_bp = 1 ; Overflow interrupt level bit 1 position

; TC0_INTCTRLB masks
.equ TC0_CCDINTLVL_gm = 0xC0 ; Compare or Capture D Interrupt Level group mask
.equ TC0_CCDINTLVL_gp = 6 ; Compare or Capture D Interrupt Level group position
.equ TC0_CCDINTLVL0_bm = (1<<6) ; Compare or Capture D Interrupt Level bit 0 mask
.equ TC0_CCDINTLVL0_bp = 6 ; Compare or Capture D Interrupt Level bit 0 position
.equ TC0_CCDINTLVL1_bm = (1<<7) ; Compare or Capture D Interrupt Level bit 1 mask
.equ TC0_CCDINTLVL1_bp = 7 ; Compare or Capture D Interrupt Level bit 1 position
.equ TC0_CCCINTLVL_gm = 0x30 ; Compare or Capture C Interrupt Level group mask
.equ TC0_CCCINTLVL_gp = 4 ; Compare or Capture C Interrupt Level group position
.equ TC0_CCCINTLVL0_bm = (1<<4) ; Compare or Capture C Interrupt Level bit 0 mask
.equ TC0_CCCINTLVL0_bp = 4 ; Compare or Capture C Interrupt Level bit 0 position
.equ TC0_CCCINTLVL1_bm = (1<<5) ; Compare or Capture C Interrupt Level bit 1 mask
.equ TC0_CCCINTLVL1_bp = 5 ; Compare or Capture C Interrupt Level bit 1 position
.equ TC0_CCBINTLVL_gm = 0x0C ; Compare or Capture B Interrupt Level group mask
.equ TC0_CCBINTLVL_gp = 2 ; Compare or Capture B Interrupt Level group position
.equ TC0_CCBINTLVL0_bm = (1<<2) ; Compare or Capture B Interrupt Level bit 0 mask
.equ TC0_CCBINTLVL0_bp = 2 ; Compare or Capture B Interrupt Level bit 0 position
.equ TC0_CCBINTLVL1_bm = (1<<3) ; Compare or Capture B Interrupt Level bit 1 mask
.equ TC0_CCBINTLVL1_bp = 3 ; Compare or Capture B Interrupt Level bit 1 position
.equ TC0_CCAINTLVL_gm = 0x03 ; Compare or Capture A Interrupt Level group mask
.equ TC0_CCAINTLVL_gp = 0 ; Compare or Capture A Interrupt Level group position
.equ TC0_CCAINTLVL0_bm = (1<<0) ; Compare or Capture A Interrupt Level bit 0 mask
.equ TC0_CCAINTLVL0_bp = 0 ; Compare or Capture A Interrupt Level bit 0 position
.equ TC0_CCAINTLVL1_bm = (1<<1) ; Compare or Capture A Interrupt Level bit 1 mask
.equ TC0_CCAINTLVL1_bp = 1 ; Compare or Capture A Interrupt Level bit 1 position

; TC0_CTRLFCLR masks
.equ TC0_CMD_gm = 0x0C ; Command group mask
.equ TC0_CMD_gp = 2 ; Command group position
.equ TC0_CMD0_bm = (1<<2) ; Command bit 0 mask
.equ TC0_CMD0_bp = 2 ; Command bit 0 position
.equ TC0_CMD1_bm = (1<<3) ; Command bit 1 mask
.equ TC0_CMD1_bp = 3 ; Command bit 1 position
.equ TC0_LUPD_bm = 0x02 ; Lock Update bit mask
.equ TC0_LUPD_bp = 1 ; Lock Update bit position
.equ TC0_DIR_bm = 0x01 ; Direction bit mask
.equ TC0_DIR_bp = 0 ; Direction bit position

; TC0_CTRLFSET masks
; Masks for CMD aready defined
; Masks for LUPD aready defined
; Masks for DIR aready defined

; TC0_CTRLGCLR masks
.equ TC0_CCDBV_bm = 0x10 ; Compare or Capture D Buffer Valid bit mask
.equ TC0_CCDBV_bp = 4 ; Compare or Capture D Buffer Valid bit position
.equ TC0_CCCBV_bm = 0x08 ; Compare or Capture C Buffer Valid bit mask
.equ TC0_CCCBV_bp = 3 ; Compare or Capture C Buffer Valid bit position
.equ TC0_CCBBV_bm = 0x04 ; Compare or Capture B Buffer Valid bit mask
.equ TC0_CCBBV_bp = 2 ; Compare or Capture B Buffer Valid bit position
.equ TC0_CCABV_bm = 0x02 ; Compare or Capture A Buffer Valid bit mask
.equ TC0_CCABV_bp = 1 ; Compare or Capture A Buffer Valid bit position
.equ TC0_PERBV_bm = 0x01 ; Period Buffer Valid bit mask
.equ TC0_PERBV_bp = 0 ; Period Buffer Valid bit position
```

```
; TC0_CTRLGSET masks
; Masks for CCDBV aready defined
; Masks for CCCBV aready defined
; Masks for CCBBV aready defined
; Masks for CCABV aready defined
; Masks for PERBV aready defined

; TC0_INTFLAGS masks
.equ TC0_CCDIF_bm = 0x80 ; Compare or Capture D Interrupt Flag bit mask
.equ TC0_CCDIF_bp = 7 ; Compare or Capture D Interrupt Flag bit position
.equ TC0_CCCIF_bm = 0x40 ; Compare or Capture C Interrupt Flag bit mask
.equ TC0_CCCIF_bp = 6 ; Compare or Capture C Interrupt Flag bit position
.equ TC0_CCBIF_bm = 0x20 ; Compare or Capture B Interrupt Flag bit mask
.equ TC0_CCBIF_bp = 5 ; Compare or Capture B Interrupt Flag bit position
.equ TC0_CCAIF_bm = 0x10 ; Compare or Capture A Interrupt Flag bit mask
.equ TC0_CCAIF_bp = 4 ; Compare or Capture A Interrupt Flag bit position
.equ TC0_ERRIF_bm = 0x02 ; Error Interrupt Flag bit mask
.equ TC0_ERRIF_bp = 1 ; Error Interrupt Flag bit position
.equ TC0_OVFIF_bm = 0x01 ; Overflow Interrupt Flag bit mask
.equ TC0_OVFIF_bp = 0 ; Overflow Interrupt Flag bit position

; TC1_CTRLA masks
.equ TC1_CLKSEL_gm = 0x0F ; Clock Selection group mask
.equ TC1_CLKSEL_gp = 0 ; Clock Selection group position
.equ TC1_CLKSEL0_bm = (1<<0) ; Clock Selection bit 0 mask
.equ TC1_CLKSEL0_bp = 0 ; Clock Selection bit 0 position
.equ TC1_CLKSEL1_bm = (1<<1) ; Clock Selection bit 1 mask
.equ TC1_CLKSEL1_bp = 1 ; Clock Selection bit 1 position
.equ TC1_CLKSEL2_bm = (1<<2) ; Clock Selection bit 2 mask
.equ TC1_CLKSEL2_bp = 2 ; Clock Selection bit 2 position
.equ TC1_CLKSEL3_bm = (1<<3) ; Clock Selection bit 3 mask
.equ TC1_CLKSEL3_bp = 3 ; Clock Selection bit 3 position

; TC1_CTRLB masks
.equ TC1_CCBEN_bm = 0x20 ; Compare or Capture B Enable bit mask
.equ TC1_CCBEN_bp = 5 ; Compare or Capture B Enable bit position
.equ TC1_CCAEN_bm = 0x10 ; Compare or Capture A Enable bit mask
.equ TC1_CCAEN_bp = 4 ; Compare or Capture A Enable bit position
.equ TC1_WGMODE_gm = 0x07 ; Waveform generation mode group mask
.equ TC1_WGMODE_gp = 0 ; Waveform generation mode group position
.equ TC1_WGMODE0_bm = (1<<0) ; Waveform generation mode bit 0 mask
.equ TC1_WGMODE0_bp = 0 ; Waveform generation mode bit 0 position
.equ TC1_WGMODE1_bm = (1<<1) ; Waveform generation mode bit 1 mask
.equ TC1_WGMODE1_bp = 1 ; Waveform generation mode bit 1 position
.equ TC1_WGMODE2_bm = (1<<2) ; Waveform generation mode bit 2 mask
.equ TC1_WGMODE2_bp = 2 ; Waveform generation mode bit 2 position

; TC1_CTRLC masks
.equ TC1_CMPB_bm = 0x02 ; Compare B Output Value bit mask
.equ TC1_CMPB_bp = 1 ; Compare B Output Value bit position
.equ TC1_CMPA_bm = 0x01 ; Compare A Output Value bit mask
.equ TC1_CMPA_bp = 0 ; Compare A Output Value bit position

; TC1_CTRLD masks
.equ TC1_EVACT_gm = 0xE0 ; Event Action group mask
.equ TC1_EVACT_gp = 5 ; Event Action group position
.equ TC1_EVACT0_bm = (1<<5) ; Event Action bit 0 mask
.equ TC1_EVACT0_bp = 5 ; Event Action bit 0 position
.equ TC1_EVACT1_bm = (1<<6) ; Event Action bit 1 mask
.equ TC1_EVACT1_bp = 6 ; Event Action bit 1 position
.equ TC1_EVACT2_bm = (1<<7) ; Event Action bit 2 mask
.equ TC1_EVACT2_bp = 7 ; Event Action bit 2 position
.equ TC1_EVDLY_bm = 0x10 ; Event Delay bit mask
.equ TC1_EVDLY_bp = 4 ; Event Delay bit position
.equ TC1_EVSEL_gm = 0x0F ; Event Source Select group mask
.equ TC1_EVSEL_gp = 0 ; Event Source Select group position
```

```
5337  .equ TC1_EVSEL0_bm = (1<<0) ; Event Source Select bit 0 mask
5338  .equ TC1_EVSEL0_bp = 0 ; Event Source Select bit 0 position
5339  .equ TC1_EVSEL1_bm = (1<<1) ; Event Source Select bit 1 mask
5340  .equ TC1_EVSEL1_bp = 1 ; Event Source Select bit 1 position
5341  .equ TC1_EVSEL2_bm = (1<<2) ; Event Source Select bit 2 mask
5342  .equ TC1_EVSEL2_bp = 2 ; Event Source Select bit 2 position
5343  .equ TC1_EVSEL3_bm = (1<<3) ; Event Source Select bit 3 mask
5344  .equ TC1_EVSEL3_bp = 3 ; Event Source Select bit 3 position
5345
5346  ; TC1_CTRLE masks
5347  .equ TC1_BYTEM_bm = 0x01 ; Byte Mode bit mask
5348  .equ TC1_BYTEM_bp = 0 ; Byte Mode bit position
5349
5350  ; TC1_INTCTRLA masks
5351  .equ TC1_ERRINTLVL_gm = 0x0C ; Error Interrupt Level group mask
5352  .equ TC1_ERRINTLVL_gp = 2 ; Error Interrupt Level group position
5353  .equ TC1_ERRINTLVL0_bm = (1<<2) ; Error Interrupt Level bit 0 mask
5354  .equ TC1_ERRINTLVL0_bp = 2 ; Error Interrupt Level bit 0 position
5355  .equ TC1_ERRINTLVL1_bm = (1<<3) ; Error Interrupt Level bit 1 mask
5356  .equ TC1_ERRINTLVL1_bp = 3 ; Error Interrupt Level bit 1 position
5357  .equ TC1_OVFINTLVL_gm = 0x03 ; Overflow interrupt level group mask
5358  .equ TC1_OVFINTLVL_gp = 0 ; Overflow interrupt level group position
5359  .equ TC1_OVFINTLVL0_bm = (1<<0) ; Overflow interrupt level bit 0 mask
5360  .equ TC1_OVFINTLVL0_bp = 0 ; Overflow interrupt level bit 0 position
5361  .equ TC1_OVFINTLVL1_bm = (1<<1) ; Overflow interrupt level bit 1 mask
5362  .equ TC1_OVFINTLVL1_bp = 1 ; Overflow interrupt level bit 1 position
5363
5364  ; TC1_INTCTRLB masks
5365  .equ TC1_CCBINTLVL_gm = 0x0C ; Compare or Capture B Interrupt Level group mask
5366  .equ TC1_CCBINTLVL_gp = 2 ; Compare or Capture B Interrupt Level group position
5367  .equ TC1_CCBINTLVL0_bm = (1<<2) ; Compare or Capture B Interrupt Level bit 0 mask
5368  .equ TC1_CCBINTLVL0_bp = 2 ; Compare or Capture B Interrupt Level bit 0 position
5369  .equ TC1_CCBINTLVL1_bm = (1<<3) ; Compare or Capture B Interrupt Level bit 1 mask
5370  .equ TC1_CCBINTLVL1_bp = 3 ; Compare or Capture B Interrupt Level bit 1 position
5371  .equ TC1_CCAINTLVL_gm = 0x03 ; Compare or Capture A Interrupt Level group mask
5372  .equ TC1_CCAINTLVL_gp = 0 ; Compare or Capture A Interrupt Level group position
5373  .equ TC1_CCAINTLVL0_bm = (1<<0) ; Compare or Capture A Interrupt Level bit 0 mask
5374  .equ TC1_CCAINTLVL0_bp = 0 ; Compare or Capture A Interrupt Level bit 0 position
5375  .equ TC1_CCAINTLVL1_bm = (1<<1) ; Compare or Capture A Interrupt Level bit 1 mask
5376  .equ TC1_CCAINTLVL1_bp = 1 ; Compare or Capture A Interrupt Level bit 1 position
5377
5378  ; TC1_CTRLFCLR masks
5379  .equ TC1_CMD_gm = 0x0C ; Command group mask
5380  .equ TC1_CMD_gp = 2 ; Command group position
5381  .equ TC1_CMD0_bm = (1<<2) ; Command bit 0 mask
5382  .equ TC1_CMD0_bp = 2 ; Command bit 0 position
5383  .equ TC1_CMD1_bm = (1<<3) ; Command bit 1 mask
5384  .equ TC1_CMD1_bp = 3 ; Command bit 1 position
5385  .equ TC1_LUPD_bm = 0x02 ; Lock Update bit mask
5386  .equ TC1_LUPD_bp = 1 ; Lock Update bit position
5387  .equ TC1_DIR_bm = 0x01 ; Direction bit mask
5388  .equ TC1_DIR_bp = 0 ; Direction bit position
5389
5390  ; TC1_CTRLFSET masks
5391  ; Masks for CMD aready defined
5392  ; Masks for LUPD aready defined
5393  ; Masks for DIR aready defined
5394
5395  ; TC1_CTRLGCLR masks
5396  .equ TC1_CCBBV_bm = 0x04 ; Compare or Capture B Buffer Valid bit mask
5397  .equ TC1_CCBBV_bp = 2 ; Compare or Capture B Buffer Valid bit position
5398  .equ TC1_CCABV_bm = 0x02 ; Compare or Capture A Buffer Valid bit mask
5399  .equ TC1_CCABV_bp = 1 ; Compare or Capture A Buffer Valid bit position
5400  .equ TC1_PERBV_bm = 0x01 ; Period Buffer Valid bit mask
5401  .equ TC1_PERBV_bp = 0 ; Period Buffer Valid bit position
5402
```

```
5403  ; TC1_CTRLGSET masks
5404  ; Masks for CCBBV aready defined
5405  ; Masks for CCABV aready defined
5406  ; Masks for PERBV aready defined
5407
5408  ; TC1_INTFLAGS masks
5409  .equ TC1_CCBIF_bm = 0x20 ; Compare or Capture B Interrupt Flag bit mask
5410  .equ TC1_CCBIF_bp = 5 ; Compare or Capture B Interrupt Flag bit position
5411  .equ TC1_CCAIF_bm = 0x10 ; Compare or Capture A Interrupt Flag bit mask
5412  .equ TC1_CCAIF_bp = 4 ; Compare or Capture A Interrupt Flag bit position
5413  .equ TC1_ERRIF_bm = 0x02 ; Error Interrupt Flag bit mask
5414  .equ TC1_ERRIF_bp = 1 ; Error Interrupt Flag bit position
5415  .equ TC1_OVFIF_bm = 0x01 ; Overflow Interrupt Flag bit mask
5416  .equ TC1_OVFIF_bp = 0 ; Overflow Interrupt Flag bit position
5417
5418  ; Clock Selection
5419  .equ TC_CLKSEL_OFF_gc = (0x00<<0) ; Timer Off
5420  .equ TC_CLKSEL_DIV1_gc = (0x01<<0) ; System Clock
5421  .equ TC_CLKSEL_DIV2_gc = (0x02<<0) ; System Clock / 2
5422  .equ TC_CLKSEL_DIV4_gc = (0x03<<0) ; System Clock / 4
5423  .equ TC_CLKSEL_DIV8_gc = (0x04<<0) ; System Clock / 8
5424  .equ TC_CLKSEL_DIV64_gc = (0x05<<0) ; System Clock / 64
5425  .equ TC_CLKSEL_DIV256_gc = (0x06<<0) ; System Clock / 256
5426  .equ TC_CLKSEL_DIV1024_gc = (0x07<<0) ; System Clock / 1024
5427  .equ TC_CLKSEL_EVCH0_gc = (0x08<<0) ; Event Channel 0
5428  .equ TC_CLKSEL_EVCH1_gc = (0x09<<0) ; Event Channel 1
5429  .equ TC_CLKSEL_EVCH2_gc = (0x0A<<0) ; Event Channel 2
5430  .equ TC_CLKSEL_EVCH3_gc = (0x0B<<0) ; Event Channel 3
5431  .equ TC_CLKSEL_EVCH4_gc = (0x0C<<0) ; Event Channel 4
5432  .equ TC_CLKSEL_EVCH5_gc = (0x0D<<0) ; Event Channel 5
5433  .equ TC_CLKSEL_EVCH6_gc = (0x0E<<0) ; Event Channel 6
5434  .equ TC_CLKSEL_EVCH7_gc = (0x0F<<0) ; Event Channel 7
5435
5436  ; Waveform Generation Mode
5437  .equ TC_WGMODE_NORMAL_gc = (0x00<<0) ; Normal Mode
5438  .equ TC_WGMODE_FRQ_gc = (0x01<<0) ; Frequency Generation Mode
5439  .equ TC_WGMODE_SINGLESLOPE_gc = (0x03<<0) ; Single Slope
5440  .equ TC_WGMODE_SS_gc = (0x03<<0) ; Single Slope
5441  .equ TC_WGMODE_DSTOP_gc = (0x05<<0) ; Dual Slope, Update on TOP
5442  .equ TC_WGMODE_DS_T_gc = (0x05<<0) ; Dual Slope, Update on TOP
5443  .equ TC_WGMODE_DSBOTH_gc = (0x06<<0) ; Dual Slope, Update on both TOP and BOTTOM
5444  .equ TC_WGMODE_DS_TB_gc = (0x06<<0) ; Dual Slope, Update on both TOP and BOTTOM
5445  .equ TC_WGMODE_DSBOTTOM_gc = (0x07<<0) ; Dual Slope, Update on BOTTOM
5446  .equ TC_WGMODE_DS_B_gc = (0x07<<0) ; Dual Slope, Update on BOTTOM
5447
5448  ; Byte Mode
5449  .equ TC_BYTEM_NORMAL_gc = (0x00<<0) ; 16-bit mode
5450  .equ TC_BYTEM_BYTEMODE_gc = (0x01<<0) ; Timer/Counter operating in byte mode only
5451  .equ TC_BYTEM_SPLITMODE_gc = (0x02<<0) ; Timer/Counter split into two 8-bit Counters
      (TC2)
5452
5453  ; Event Action
5454  .equ TC_EVACT_OFF_gc = (0x00<<5) ; No Event Action
5455  .equ TC_EVACT_CAPT_gc = (0x01<<5) ; Input Capture
5456  .equ TC_EVACT_UPDOWN_gc = (0x02<<5) ; Externally Controlled Up/Down Count
5457  .equ TC_EVACT_QDEC_gc = (0x03<<5) ; Quadrature Decode
5458  .equ TC_EVACT_RESTART_gc = (0x04<<5) ; Restart
5459  .equ TC_EVACT_FRQ_gc = (0x05<<5) ; Frequency Capture
5460  .equ TC_EVACT_PW_gc = (0x06<<5) ; Pulse-width Capture
5461
5462  ; Event Selection
5463  .equ TC_EVSEL_OFF_gc = (0x00<<0) ; No Event Source
5464  .equ TC_EVSEL_CH0_gc = (0x08<<0) ; Event Channel 0
5465  .equ TC_EVSEL_CH1_gc = (0x09<<0) ; Event Channel 1
5466  .equ TC_EVSEL_CH2_gc = (0x0A<<0) ; Event Channel 2
5467  .equ TC_EVSEL_CH3_gc = (0x0B<<0) ; Event Channel 3
```

```
5468   .equ TC_EVSEL_CH4_gc = (0x0C<<0) ; Event Channel 4
5469   .equ TC_EVSEL_CH5_gc = (0x0D<<0) ; Event Channel 5
5470   .equ TC_EVSEL_CH6_gc = (0x0E<<0) ; Event Channel 6
5471   .equ TC_EVSEL_CH7_gc = (0x0F<<0) ; Event Channel 7
5472
5473   ; Error Interrupt Level
5474   .equ TC_ERRINTLVL_OFF_gc = (0x00<<2) ; Interrupt Disabled
5475   .equ TC_ERRINTLVL_LO_gc = (0x01<<2) ; Low Level
5476   .equ TC_ERRINTLVL_MED_gc = (0x02<<2) ; Medium Level
5477   .equ TC_ERRINTLVL_HI_gc = (0x03<<2) ; High Level
5478
5479   ; Overflow Interrupt Level
5480   .equ TC_OVFINTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
5481   .equ TC_OVFINTLVL_LO_gc = (0x01<<0) ; Low Level
5482   .equ TC_OVFINTLVL_MED_gc = (0x02<<0) ; Medium Level
5483   .equ TC_OVFINTLVL_HI_gc = (0x03<<0) ; High Level
5484
5485   ; Compare or Capture D Interrupt Level
5486   .equ TC_CCDINTLVL_OFF_gc = (0x00<<6) ; Interrupt Disabled
5487   .equ TC_CCDINTLVL_LO_gc = (0x01<<6) ; Low Level
5488   .equ TC_CCDINTLVL_MED_gc = (0x02<<6) ; Medium Level
5489   .equ TC_CCDINTLVL_HI_gc = (0x03<<6) ; High Level
5490
5491   ; Compare or Capture C Interrupt Level
5492   .equ TC_CCCINTLVL_OFF_gc = (0x00<<4) ; Interrupt Disabled
5493   .equ TC_CCCINTLVL_LO_gc = (0x01<<4) ; Low Level
5494   .equ TC_CCCINTLVL_MED_gc = (0x02<<4) ; Medium Level
5495   .equ TC_CCCINTLVL_HI_gc = (0x03<<4) ; High Level
5496
5497   ; Compare or Capture B Interrupt Level
5498   .equ TC_CCBINTLVL_OFF_gc = (0x00<<2) ; Interrupt Disabled
5499   .equ TC_CCBINTLVL_LO_gc = (0x01<<2) ; Low Level
5500   .equ TC_CCBINTLVL_MED_gc = (0x02<<2) ; Medium Level
5501   .equ TC_CCBINTLVL_HI_gc = (0x03<<2) ; High Level
5502
5503   ; Compare or Capture A Interrupt Level
5504   .equ TC_CCAINTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
5505   .equ TC_CCAINTLVL_LO_gc = (0x01<<0) ; Low Level
5506   .equ TC_CCAINTLVL_MED_gc = (0x02<<0) ; Medium Level
5507   .equ TC_CCAINTLVL_HI_gc = (0x03<<0) ; High Level
5508
5509   ; Timer/Counter Command
5510   .equ TC_CMD_NONE_gc = (0x00<<2) ; No Command
5511   .equ TC_CMD_UPDATE_gc = (0x01<<2) ; Force Update
5512   .equ TC_CMD_RESTART_gc = (0x02<<2) ; Force Restart
5513   .equ TC_CMD_RESET_gc = (0x03<<2) ; Force Hard Reset
5514
5515
5516   ;**************************************************************************
5517   ;** TC2 - 16-bit Timer/Counter type 2
5518   ;**************************************************************************/
5519
5520   ; TC2_CTRLA masks
5521   .equ TC2_CLKSEL_gm = 0x0F ; Clock Selection group mask
5522   .equ TC2_CLKSEL_gp = 0 ; Clock Selection group position
5523   .equ TC2_CLKSEL0_bm = (1<<0) ; Clock Selection bit 0 mask
5524   .equ TC2_CLKSEL0_bp = 0 ; Clock Selection bit 0 position
5525   .equ TC2_CLKSEL1_bm = (1<<1) ; Clock Selection bit 1 mask
5526   .equ TC2_CLKSEL1_bp = 1 ; Clock Selection bit 1 position
5527   .equ TC2_CLKSEL2_bm = (1<<2) ; Clock Selection bit 2 mask
5528   .equ TC2_CLKSEL2_bp = 2 ; Clock Selection bit 2 position
5529   .equ TC2_CLKSEL3_bm = (1<<3) ; Clock Selection bit 3 mask
5530   .equ TC2_CLKSEL3_bp = 3 ; Clock Selection bit 3 position
5531
5532   ; TC2_CTRLB masks
5533   .equ TC2_HCMPDEN_bm = 0x80 ; High Byte Compare D Enable bit mask
```

```
5534  .equ TC2_HCMPDEN_bp = 7 ; High Byte Compare D Enable bit position
5535  .equ TC2_HCMPCEN_bm = 0x40 ; High Byte Compare C Enable bit mask
5536  .equ TC2_HCMPCEN_bp = 6 ; High Byte Compare C Enable bit position
5537  .equ TC2_HCMPBEN_bm = 0x20 ; High Byte Compare B Enable bit mask
5538  .equ TC2_HCMPBEN_bp = 5 ; High Byte Compare B Enable bit position
5539  .equ TC2_HCMPAEN_bm = 0x10 ; High Byte Compare A Enable bit mask
5540  .equ TC2_HCMPAEN_bp = 4 ; High Byte Compare A Enable bit position
5541  .equ TC2_LCMPDEN_bm = 0x08 ; Low Byte Compare D Enable bit mask
5542  .equ TC2_LCMPDEN_bp = 3 ; Low Byte Compare D Enable bit position
5543  .equ TC2_LCMPCEN_bm = 0x04 ; Low Byte Compare C Enable bit mask
5544  .equ TC2_LCMPCEN_bp = 2 ; Low Byte Compare C Enable bit position
5545  .equ TC2_LCMPBEN_bm = 0x02 ; Low Byte Compare B Enable bit mask
5546  .equ TC2_LCMPBEN_bp = 1 ; Low Byte Compare B Enable bit position
5547  .equ TC2_LCMPAEN_bm = 0x01 ; Low Byte Compare A Enable bit mask
5548  .equ TC2_LCMPAEN_bp = 0 ; Low Byte Compare A Enable bit position
5549
5550  ; TC2_CTRLC masks
5551  .equ TC2_HCMPD_bm = 0x80 ; High Byte Compare D Output Value bit mask
5552  .equ TC2_HCMPD_bp = 7 ; High Byte Compare D Output Value bit position
5553  .equ TC2_HCMPC_bm = 0x40 ; High Byte Compare C Output Value bit mask
5554  .equ TC2_HCMPC_bp = 6 ; High Byte Compare C Output Value bit position
5555  .equ TC2_HCMPB_bm = 0x20 ; High Byte Compare B Output Value bit mask
5556  .equ TC2_HCMPB_bp = 5 ; High Byte Compare B Output Value bit position
5557  .equ TC2_HCMPA_bm = 0x10 ; High Byte Compare A Output Value bit mask
5558  .equ TC2_HCMPA_bp = 4 ; High Byte Compare A Output Value bit position
5559  .equ TC2_LCMPD_bm = 0x08 ; Low Byte Compare D Output Value bit mask
5560  .equ TC2_LCMPD_bp = 3 ; Low Byte Compare D Output Value bit position
5561  .equ TC2_LCMPC_bm = 0x04 ; Low Byte Compare C Output Value bit mask
5562  .equ TC2_LCMPC_bp = 2 ; Low Byte Compare C Output Value bit position
5563  .equ TC2_LCMPB_bm = 0x02 ; Low Byte Compare B Output Value bit mask
5564  .equ TC2_LCMPB_bp = 1 ; Low Byte Compare B Output Value bit position
5565  .equ TC2_LCMPA_bm = 0x01 ; Low Byte Compare A Output Value bit mask
5566  .equ TC2_LCMPA_bp = 0 ; Low Byte Compare A Output Value bit position
5567
5568  ; TC2_CTRLE masks
5569  .equ TC2_BYTEM_gm = 0x03 ; Byte Mode group mask
5570  .equ TC2_BYTEM_gp = 0 ; Byte Mode group position
5571  .equ TC2_BYTEM0_bm = (1<<0) ; Byte Mode bit 0 mask
5572  .equ TC2_BYTEM0_bp = 0 ; Byte Mode bit 0 position
5573  .equ TC2_BYTEM1_bm = (1<<1) ; Byte Mode bit 1 mask
5574  .equ TC2_BYTEM1_bp = 1 ; Byte Mode bit 1 position
5575
5576  ; TC2_INTCTRLA masks
5577  .equ TC2_HUNFINTLVL_gm = 0x0C ; High Byte Underflow Interrupt Level group mask
5578  .equ TC2_HUNFINTLVL_gp = 2 ; High Byte Underflow Interrupt Level group position
5579  .equ TC2_HUNFINTLVL0_bm = (1<<2) ; High Byte Underflow Interrupt Level bit 0 mask
5580  .equ TC2_HUNFINTLVL0_bp = 2 ; High Byte Underflow Interrupt Level bit 0 position
5581  .equ TC2_HUNFINTLVL1_bm = (1<<3) ; High Byte Underflow Interrupt Level bit 1 mask
5582  .equ TC2_HUNFINTLVL1_bp = 3 ; High Byte Underflow Interrupt Level bit 1 position
5583  .equ TC2_LUNFINTLVL_gm = 0x03 ; Low Byte Underflow interrupt level group mask
5584  .equ TC2_LUNFINTLVL_gp = 0 ; Low Byte Underflow interrupt level group position
5585  .equ TC2_LUNFINTLVL0_bm = (1<<0) ; Low Byte Underflow interrupt level bit 0 mask
5586  .equ TC2_LUNFINTLVL0_bp = 0 ; Low Byte Underflow interrupt level bit 0 position
5587  .equ TC2_LUNFINTLVL1_bm = (1<<1) ; Low Byte Underflow interrupt level bit 1 mask
5588  .equ TC2_LUNFINTLVL1_bp = 1 ; Low Byte Underflow interrupt level bit 1 position
5589
5590  ; TC2_INTCTRLB masks
5591  .equ TC2_LCMPDINTLVL_gm = 0xC0 ; Low Byte Compare D Interrupt Level group mask
5592  .equ TC2_LCMPDINTLVL_gp = 6 ; Low Byte Compare D Interrupt Level group position
5593  .equ TC2_LCMPDINTLVL0_bm = (1<<6) ; Low Byte Compare D Interrupt Level bit 0 mask
5594  .equ TC2_LCMPDINTLVL0_bp = 6 ; Low Byte Compare D Interrupt Level bit 0 position
5595  .equ TC2_LCMPDINTLVL1_bm = (1<<7) ; Low Byte Compare D Interrupt Level bit 1 mask
5596  .equ TC2_LCMPDINTLVL1_bp = 7 ; Low Byte Compare D Interrupt Level bit 1 position
5597  .equ TC2_LCMPCINTLVL_gm = 0x30 ; Low Byte Compare C Interrupt Level group mask
5598  .equ TC2_LCMPCINTLVL_gp = 4 ; Low Byte Compare C Interrupt Level group position
5599  .equ TC2_LCMPCINTLVL0_bm = (1<<4) ; Low Byte Compare C Interrupt Level bit 0 mask
```

```
.equ TC2_LCMPCINTLVL0_bp = 4 ; Low Byte Compare C Interrupt Level bit 0 position
.equ TC2_LCMPCINTLVL1_bm = (1<<5) ; Low Byte Compare C Interrupt Level bit 1 mask
.equ TC2_LCMPCINTLVL1_bp = 5 ; Low Byte Compare C Interrupt Level bit 1 position
.equ TC2_LCMPBINTLVL_gm = 0x0C ; Low Byte Compare B Interrupt Level group mask
.equ TC2_LCMPBINTLVL_gp = 2 ; Low Byte Compare B Interrupt Level group position
.equ TC2_LCMPBINTLVL0_bm = (1<<2) ; Low Byte Compare B Interrupt Level bit 0 mask
.equ TC2_LCMPBINTLVL0_bp = 2 ; Low Byte Compare B Interrupt Level bit 0 position
.equ TC2_LCMPBINTLVL1_bm = (1<<3) ; Low Byte Compare B Interrupt Level bit 1 mask
.equ TC2_LCMPBINTLVL1_bp = 3 ; Low Byte Compare B Interrupt Level bit 1 position
.equ TC2_LCMPAINTLVL_gm = 0x03 ; Low Byte Compare A Interrupt Level group mask
.equ TC2_LCMPAINTLVL_gp = 0 ; Low Byte Compare A Interrupt Level group position
.equ TC2_LCMPAINTLVL0_bm = (1<<0) ; Low Byte Compare A Interrupt Level bit 0 mask
.equ TC2_LCMPAINTLVL0_bp = 0 ; Low Byte Compare A Interrupt Level bit 0 position
.equ TC2_LCMPAINTLVL1_bm = (1<<1) ; Low Byte Compare A Interrupt Level bit 1 mask
.equ TC2_LCMPAINTLVL1_bp = 1 ; Low Byte Compare A Interrupt Level bit 1 position

; TC2_CTRLF masks
.equ TC2_CMD_gm = 0x0C ; Command group mask
.equ TC2_CMD_gp = 2 ; Command group position
.equ TC2_CMD0_bm = (1<<2) ; Command bit 0 mask
.equ TC2_CMD0_bp = 2 ; Command bit 0 position
.equ TC2_CMD1_bm = (1<<3) ; Command bit 1 mask
.equ TC2_CMD1_bp = 3 ; Command bit 1 position
.equ TC2_CMDEN_gm = 0x03 ; Command Enable group mask
.equ TC2_CMDEN_gp = 0 ; Command Enable group position
.equ TC2_CMDEN0_bm = (1<<0) ; Command Enable bit 0 mask
.equ TC2_CMDEN0_bp = 0 ; Command Enable bit 0 position
.equ TC2_CMDEN1_bm = (1<<1) ; Command Enable bit 1 mask
.equ TC2_CMDEN1_bp = 1 ; Command Enable bit 1 position

; TC2_INTFLAGS masks
.equ TC2_LCMPDIF_bm = 0x80 ; Low Byte Compare D Interrupt Flag bit mask
.equ TC2_LCMPDIF_bp = 7 ; Low Byte Compare D Interrupt Flag bit position
.equ TC2_LCMPCIF_bm = 0x40 ; Low Byte Compare C Interrupt Flag bit mask
.equ TC2_LCMPCIF_bp = 6 ; Low Byte Compare C Interrupt Flag bit position
.equ TC2_LCMPBIF_bm = 0x20 ; Low Byte Compare B Interrupt Flag bit mask
.equ TC2_LCMPBIF_bp = 5 ; Low Byte Compare B Interrupt Flag bit position
.equ TC2_LCMPAIF_bm = 0x10 ; Low Byte Compare A Interrupt Flag bit mask
.equ TC2_LCMPAIF_bp = 4 ; Low Byte Compare A Interrupt Flag bit position
.equ TC2_HUNFIF_bm = 0x02 ; High Byte Underflow Interrupt Flag bit mask
.equ TC2_HUNFIF_bp = 1 ; High Byte Underflow Interrupt Flag bit position
.equ TC2_LUNFIF_bm = 0x01 ; Low Byte Underflow Interrupt Flag bit mask
.equ TC2_LUNFIF_bp = 0 ; Low Byte Underflow Interrupt Flag bit position

; Clock Selection
.equ TC2_CLKSEL_OFF_gc = (0x00<<0) ; Timer Off
.equ TC2_CLKSEL_DIV1_gc = (0x01<<0) ; System Clock
.equ TC2_CLKSEL_DIV2_gc = (0x02<<0) ; System Clock / 2
.equ TC2_CLKSEL_DIV4_gc = (0x03<<0) ; System Clock / 4
.equ TC2_CLKSEL_DIV8_gc = (0x04<<0) ; System Clock / 8
.equ TC2_CLKSEL_DIV64_gc = (0x05<<0) ; System Clock / 64
.equ TC2_CLKSEL_DIV256_gc = (0x06<<0) ; System Clock / 256
.equ TC2_CLKSEL_DIV1024_gc = (0x07<<0) ; System Clock / 1024
.equ TC2_CLKSEL_EVCH0_gc = (0x08<<0) ; Event Channel 0
.equ TC2_CLKSEL_EVCH1_gc = (0x09<<0) ; Event Channel 1
.equ TC2_CLKSEL_EVCH2_gc = (0x0A<<0) ; Event Channel 2
.equ TC2_CLKSEL_EVCH3_gc = (0x0B<<0) ; Event Channel 3
.equ TC2_CLKSEL_EVCH4_gc = (0x0C<<0) ; Event Channel 4
.equ TC2_CLKSEL_EVCH5_gc = (0x0D<<0) ; Event Channel 5
.equ TC2_CLKSEL_EVCH6_gc = (0x0E<<0) ; Event Channel 6
.equ TC2_CLKSEL_EVCH7_gc = (0x0F<<0) ; Event Channel 7

; Byte Mode
.equ TC2_BYTEM_NORMAL_gc = (0x00<<0) ; 16-bit mode
.equ TC2_BYTEM_BYTEMODE_gc = (0x01<<0) ; Timer/Counter operating in byte mode only (TC2)
.equ TC2_BYTEM_SPLITMODE_gc = (0x02<<0) ; Timer/Counter split into two 8-bit Counters
```

```
5666
5667   ; High Byte Underflow Interrupt Level
5668   .equ TC2_HUNFINTLVL_OFF_gc = (0x00<<2) ; Interrupt Disabled
5669   .equ TC2_HUNFINTLVL_LO_gc = (0x01<<2) ; Low Level
5670   .equ TC2_HUNFINTLVL_MED_gc = (0x02<<2) ; Medium Level
5671   .equ TC2_HUNFINTLVL_HI_gc = (0x03<<2) ; High Level
5672
5673   ; Low Byte Underflow Interrupt Level
5674   .equ TC2_LUNFINTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
5675   .equ TC2_LUNFINTLVL_LO_gc = (0x01<<0) ; Low Level
5676   .equ TC2_LUNFINTLVL_MED_gc = (0x02<<0) ; Medium Level
5677   .equ TC2_LUNFINTLVL_HI_gc = (0x03<<0) ; High Level
5678
5679   ; Low Byte Compare D Interrupt Level
5680   .equ TC2_LCMPDINTLVL_OFF_gc = (0x00<<6) ; Interrupt Disabled
5681   .equ TC2_LCMPDINTLVL_LO_gc = (0x01<<6) ; Low Level
5682   .equ TC2_LCMPDINTLVL_MED_gc = (0x02<<6) ; Medium Level
5683   .equ TC2_LCMPDINTLVL_HI_gc = (0x03<<6) ; High Level
5684
5685   ; Low Byte Compare C Interrupt Level
5686   .equ TC2_LCMPCINTLVL_OFF_gc = (0x00<<4) ; Interrupt Disabled
5687   .equ TC2_LCMPCINTLVL_LO_gc = (0x01<<4) ; Low Level
5688   .equ TC2_LCMPCINTLVL_MED_gc = (0x02<<4) ; Medium Level
5689   .equ TC2_LCMPCINTLVL_HI_gc = (0x03<<4) ; High Level
5690
5691   ; Low Byte Compare B Interrupt Level
5692   .equ TC2_LCMPBINTLVL_OFF_gc = (0x00<<2) ; Interrupt Disabled
5693   .equ TC2_LCMPBINTLVL_LO_gc = (0x01<<2) ; Low Level
5694   .equ TC2_LCMPBINTLVL_MED_gc = (0x02<<2) ; Medium Level
5695   .equ TC2_LCMPBINTLVL_HI_gc = (0x03<<2) ; High Level
5696
5697   ; Low Byte Compare A Interrupt Level
5698   .equ TC2_LCMPAINTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
5699   .equ TC2_LCMPAINTLVL_LO_gc = (0x01<<0) ; Low Level
5700   .equ TC2_LCMPAINTLVL_MED_gc = (0x02<<0) ; Medium Level
5701   .equ TC2_LCMPAINTLVL_HI_gc = (0x03<<0) ; High Level
5702
5703   ; Timer/Counter Command
5704   .equ TC2_CMD_NONE_gc = (0x00<<2) ; No Command
5705   .equ TC2_CMD_RESTART_gc = (0x02<<2) ; Force Restart
5706   .equ TC2_CMD_RESET_gc = (0x03<<2) ; Force Hard Reset
5707
5708   ; Timer/Counter Command
5709   .equ TC2_CMDEN_LOW_gc = (0x01<<0) ; Low Byte Timer/Counter
5710   .equ TC2_CMDEN_HIGH_gc = (0x02<<0) ; High Byte Timer/Counter
5711   .equ TC2_CMDEN_BOTH_gc = (0x03<<0) ; Both Low Byte and High Byte Timer/Counters
5712
5713
5714   ;**************************************************************************
5715   ;** AWEX - Timer/Counter Advanced Waveform Extension
5716   ;**************************************************************************/
5717
5718   ; AWEX_CTRL masks
5719   .equ AWEX_PGM_bm = 0x20 ; Pattern Generation Mode bit mask
5720   .equ AWEX_PGM_bp = 5 ; Pattern Generation Mode bit position
5721   .equ AWEX_CWCM_bm = 0x10 ; Common Waveform Channel Mode bit mask
5722   .equ AWEX_CWCM_bp = 4 ; Common Waveform Channel Mode bit position
5723   .equ AWEX_DTICCDEN_bm = 0x08 ; Dead Time Insertion Compare Channel D Enable bit mask
5724   .equ AWEX_DTICCDEN_bp = 3 ; Dead Time Insertion Compare Channel D Enable bit position
5725   .equ AWEX_DTICCCEN_bm = 0x04 ; Dead Time Insertion Compare Channel C Enable bit mask
5726   .equ AWEX_DTICCCEN_bp = 2 ; Dead Time Insertion Compare Channel C Enable bit position
5727   .equ AWEX_DTICCBEN_bm = 0x02 ; Dead Time Insertion Compare Channel B Enable bit mask
5728   .equ AWEX_DTICCBEN_bp = 1 ; Dead Time Insertion Compare Channel B Enable bit position
5729   .equ AWEX_DTICCAEN_bm = 0x01 ; Dead Time Insertion Compare Channel A Enable bit mask
5730   .equ AWEX_DTICCAEN_bp = 0 ; Dead Time Insertion Compare Channel A Enable bit position
5731
```

```
; AWEX_FDCTRL masks
.equ AWEX_FDDBD_bm = 0x10 ; Fault Detect on Disable Break Disable bit mask
.equ AWEX_FDDBD_bp = 4 ; Fault Detect on Disable Break Disable bit position
.equ AWEX_FDMODE_bm = 0x04 ; Fault Detect Mode bit mask
.equ AWEX_FDMODE_bp = 2 ; Fault Detect Mode bit position
.equ AWEX_FDACT_gm = 0x03 ; Fault Detect Action group mask
.equ AWEX_FDACT_gp = 0 ; Fault Detect Action group position
.equ AWEX_FDACT0_bm = (1<<0) ; Fault Detect Action bit 0 mask
.equ AWEX_FDACT0_bp = 0 ; Fault Detect Action bit 0 position
.equ AWEX_FDACT1_bm = (1<<1) ; Fault Detect Action bit 1 mask
.equ AWEX_FDACT1_bp = 1 ; Fault Detect Action bit 1 position

; AWEX_STATUS masks
.equ AWEX_FDF_bm = 0x04 ; Fault Detect Flag bit mask
.equ AWEX_FDF_bp = 2 ; Fault Detect Flag bit position
.equ AWEX_DTHSBUFV_bm = 0x02 ; Dead Time High Side Buffer Valid bit mask
.equ AWEX_DTHSBUFV_bp = 1 ; Dead Time High Side Buffer Valid bit position
.equ AWEX_DTLSBUFV_bm = 0x01 ; Dead Time Low Side Buffer Valid bit mask
.equ AWEX_DTLSBUFV_bp = 0 ; Dead Time Low Side Buffer Valid bit position

; AWEX_STATUSSET masks
; Masks for FDF aready defined
; Masks for DTHSBUFV aready defined
; Masks for DTLSBUFV aready defined

; Fault Detect Action
.equ AWEX_FDACT_NONE_gc = (0x00<<0) ; No Fault Protection
.equ AWEX_FDACT_CLEAROE_gc = (0x01<<0) ; Clear Output Enable Bits
.equ AWEX_FDACT_CLEARDIR_gc = (0x03<<0) ; Clear I/O Port Direction Bits


;**************************************************************************
;** HIRES - Timer/Counter High-Resolution Extension
;**************************************************************************/

; HIRES_CTRLA masks
.equ HIRES_HREN_gm = 0x03 ; High Resolution Enable group mask
.equ HIRES_HREN_gp = 0 ; High Resolution Enable group position
.equ HIRES_HREN0_bm = (1<<0) ; High Resolution Enable bit 0 mask
.equ HIRES_HREN0_bp = 0 ; High Resolution Enable bit 0 position
.equ HIRES_HREN1_bm = (1<<1) ; High Resolution Enable bit 1 mask
.equ HIRES_HREN1_bp = 1 ; High Resolution Enable bit 1 position

; High Resolution Enable
.equ HIRES_HREN_NONE_gc = (0x00<<0) ; No Fault Protection
.equ HIRES_HREN_TC0_gc = (0x01<<0) ; Enable High Resolution on Timer/Counter 0
.equ HIRES_HREN_TC1_gc = (0x02<<0) ; Enable High Resolution on Timer/Counter 1
.equ HIRES_HREN_BOTH_gc = (0x03<<0) ; Enable High Resolution both Timer/Counters


;**************************************************************************
;** USART - Universal Asynchronous Receiver-Transmitter
;**************************************************************************/

; USART_STATUS masks
.equ USART_RXCIF_bm = 0x80 ; Receive Interrupt Flag bit mask
.equ USART_RXCIF_bp = 7 ; Receive Interrupt Flag bit position
.equ USART_TXCIF_bm = 0x40 ; Transmit Interrupt Flag bit mask
.equ USART_TXCIF_bp = 6 ; Transmit Interrupt Flag bit position
.equ USART_DREIF_bm = 0x20 ; Data Register Empty Flag bit mask
.equ USART_DREIF_bp = 5 ; Data Register Empty Flag bit position
.equ USART_FERR_bm = 0x10 ; Frame Error bit mask
.equ USART_FERR_bp = 4 ; Frame Error bit position
.equ USART_BUFOVF_bm = 0x08 ; Buffer Overflow bit mask
.equ USART_BUFOVF_bp = 3 ; Buffer Overflow bit position
.equ USART_PERR_bm = 0x04 ; Parity Error bit mask
```

```
.equ USART_PERR_bp = 2 ; Parity Error bit position
.equ USART_RXB8_bm = 0x01 ; Receive Bit 8 bit mask
.equ USART_RXB8_bp = 0 ; Receive Bit 8 bit position

; USART_CTRLA masks
.equ USART_RXCINTLVL_gm = 0x30 ; Receive Interrupt Level group mask
.equ USART_RXCINTLVL_gp = 4 ; Receive Interrupt Level group position
.equ USART_RXCINTLVL0_bm = (1<<4) ; Receive Interrupt Level bit 0 mask
.equ USART_RXCINTLVL0_bp = 4 ; Receive Interrupt Level bit 0 position
.equ USART_RXCINTLVL1_bm = (1<<5) ; Receive Interrupt Level bit 1 mask
.equ USART_RXCINTLVL1_bp = 5 ; Receive Interrupt Level bit 1 position
.equ USART_TXCINTLVL_gm = 0x0C ; Transmit Interrupt Level group mask
.equ USART_TXCINTLVL_gp = 2 ; Transmit Interrupt Level group position
.equ USART_TXCINTLVL0_bm = (1<<2) ; Transmit Interrupt Level bit 0 mask
.equ USART_TXCINTLVL0_bp = 2 ; Transmit Interrupt Level bit 0 position
.equ USART_TXCINTLVL1_bm = (1<<3) ; Transmit Interrupt Level bit 1 mask
.equ USART_TXCINTLVL1_bp = 3 ; Transmit Interrupt Level bit 1 position
.equ USART_DREINTLVL_gm = 0x03 ; Data Register Empty Interrupt Level group mask
.equ USART_DREINTLVL_gp = 0 ; Data Register Empty Interrupt Level group position
.equ USART_DREINTLVL0_bm = (1<<0) ; Data Register Empty Interrupt Level bit 0 mask
.equ USART_DREINTLVL0_bp = 0 ; Data Register Empty Interrupt Level bit 0 position
.equ USART_DREINTLVL1_bm = (1<<1) ; Data Register Empty Interrupt Level bit 1 mask
.equ USART_DREINTLVL1_bp = 1 ; Data Register Empty Interrupt Level bit 1 position

; USART_CTRLB masks
.equ USART_RXEN_bm = 0x10 ; Receiver Enable bit mask
.equ USART_RXEN_bp = 4 ; Receiver Enable bit position
.equ USART_TXEN_bm = 0x08 ; Transmitter Enable bit mask
.equ USART_TXEN_bp = 3 ; Transmitter Enable bit position
.equ USART_CLK2X_bm = 0x04 ; Double transmission speed bit mask
.equ USART_CLK2X_bp = 2 ; Double transmission speed bit position
.equ USART_MPCM_bm = 0x02 ; Multi-processor Communication Mode bit mask
.equ USART_MPCM_bp = 1 ; Multi-processor Communication Mode bit position
.equ USART_TXB8_bm = 0x01 ; Transmit bit 8 bit mask
.equ USART_TXB8_bp = 0 ; Transmit bit 8 bit position

; USART_CTRLC masks
.equ USART_CMODE_gm = 0xC0 ; Communication Mode group mask
.equ USART_CMODE_gp = 6 ; Communication Mode group position
.equ USART_CMODE0_bm = (1<<6) ; Communication Mode bit 0 mask
.equ USART_CMODE0_bp = 6 ; Communication Mode bit 0 position
.equ USART_CMODE1_bm = (1<<7) ; Communication Mode bit 1 mask
.equ USART_CMODE1_bp = 7 ; Communication Mode bit 1 position
.equ USART_PMODE_gm = 0x30 ; Parity Mode group mask
.equ USART_PMODE_gp = 4 ; Parity Mode group position
.equ USART_PMODE0_bm = (1<<4) ; Parity Mode bit 0 mask
.equ USART_PMODE0_bp = 4 ; Parity Mode bit 0 position
.equ USART_PMODE1_bm = (1<<5) ; Parity Mode bit 1 mask
.equ USART_PMODE1_bp = 5 ; Parity Mode bit 1 position
.equ USART_SBMODE_bm = 0x08 ; Stop Bit Mode bit mask
.equ USART_SBMODE_bp = 3 ; Stop Bit Mode bit position
.equ USART_CHSIZE_gm = 0x07 ; Character Size group mask
.equ USART_CHSIZE_gp = 0 ; Character Size group position
.equ USART_CHSIZE0_bm = (1<<0) ; Character Size bit 0 mask
.equ USART_CHSIZE0_bp = 0 ; Character Size bit 0 position
.equ USART_CHSIZE1_bm = (1<<1) ; Character Size bit 1 mask
.equ USART_CHSIZE1_bp = 1 ; Character Size bit 1 position
.equ USART_CHSIZE2_bm = (1<<2) ; Character Size bit 2 mask
.equ USART_CHSIZE2_bp = 2 ; Character Size bit 2 position

; USART_BAUDCTRLA masks
.equ USART_BSEL_gm = 0xFF ; Baud Rate Selection Bits [7:0] group mask
.equ USART_BSEL_gp = 0 ; Baud Rate Selection Bits [7:0] group position
.equ USART_BSEL0_bm = (1<<0) ; Baud Rate Selection Bits [7:0] bit 0 mask
.equ USART_BSEL0_bp = 0 ; Baud Rate Selection Bits [7:0] bit 0 position
.equ USART_BSEL1_bm = (1<<1) ; Baud Rate Selection Bits [7:0] bit 1 mask
```

```
5864  .equ USART_BSEL1_bp = 1 ; Baud Rate Selection Bits [7:0] bit 1 position
5865  .equ USART_BSEL2_bm = (1<<2) ; Baud Rate Selection Bits [7:0] bit 2 mask
5866  .equ USART_BSEL2_bp = 2 ; Baud Rate Selection Bits [7:0] bit 2 position
5867  .equ USART_BSEL3_bm = (1<<3) ; Baud Rate Selection Bits [7:0] bit 3 mask
5868  .equ USART_BSEL3_bp = 3 ; Baud Rate Selection Bits [7:0] bit 3 position
5869  .equ USART_BSEL4_bm = (1<<4) ; Baud Rate Selection Bits [7:0] bit 4 mask
5870  .equ USART_BSEL4_bp = 4 ; Baud Rate Selection Bits [7:0] bit 4 position
5871  .equ USART_BSEL5_bm = (1<<5) ; Baud Rate Selection Bits [7:0] bit 5 mask
5872  .equ USART_BSEL5_bp = 5 ; Baud Rate Selection Bits [7:0] bit 5 position
5873  .equ USART_BSEL6_bm = (1<<6) ; Baud Rate Selection Bits [7:0] bit 6 mask
5874  .equ USART_BSEL6_bp = 6 ; Baud Rate Selection Bits [7:0] bit 6 position
5875  .equ USART_BSEL7_bm = (1<<7) ; Baud Rate Selection Bits [7:0] bit 7 mask
5876  .equ USART_BSEL7_bp = 7 ; Baud Rate Selection Bits [7:0] bit 7 position
5877
5878  ; USART_BAUDCTRLB masks
5879  .equ USART_BSCALE_gm = 0xF0 ; Baud Rate Scale group mask
5880  .equ USART_BSCALE_gp = 4 ; Baud Rate Scale group position
5881  .equ USART_BSCALE0_bm = (1<<4) ; Baud Rate Scale bit 0 mask
5882  .equ USART_BSCALE0_bp = 4 ; Baud Rate Scale bit 0 position
5883  .equ USART_BSCALE1_bm = (1<<5) ; Baud Rate Scale bit 1 mask
5884  .equ USART_BSCALE1_bp = 5 ; Baud Rate Scale bit 1 position
5885  .equ USART_BSCALE2_bm = (1<<6) ; Baud Rate Scale bit 2 mask
5886  .equ USART_BSCALE2_bp = 6 ; Baud Rate Scale bit 2 position
5887  .equ USART_BSCALE3_bm = (1<<7) ; Baud Rate Scale bit 3 mask
5888  .equ USART_BSCALE3_bp = 7 ; Baud Rate Scale bit 3 position
5889  ; Masks for BSEL aready defined
5890
5891  ; Receive Complete Interrupt level
5892  .equ USART_RXCINTLVL_OFF_gc = (0x00<<4) ; Interrupt Disabled
5893  .equ USART_RXCINTLVL_LO_gc = (0x01<<4) ; Low Level
5894  .equ USART_RXCINTLVL_MED_gc = (0x02<<4) ; Medium Level
5895  .equ USART_RXCINTLVL_HI_gc = (0x03<<4) ; High Level
5896
5897  ; Transmit Complete Interrupt level
5898  .equ USART_TXCINTLVL_OFF_gc = (0x00<<2) ; Interrupt Disabled
5899  .equ USART_TXCINTLVL_LO_gc = (0x01<<2) ; Low Level
5900  .equ USART_TXCINTLVL_MED_gc = (0x02<<2) ; Medium Level
5901  .equ USART_TXCINTLVL_HI_gc = (0x03<<2) ; High Level
5902
5903  ; Data Register Empty Interrupt level
5904  .equ USART_DREINTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
5905  .equ USART_DREINTLVL_LO_gc = (0x01<<0) ; Low Level
5906  .equ USART_DREINTLVL_MED_gc = (0x02<<0) ; Medium Level
5907  .equ USART_DREINTLVL_HI_gc = (0x03<<0) ; High Level
5908
5909  ; Character Size
5910  .equ USART_CHSIZE_5BIT_gc = (0x00<<0) ; Character size: 5 bit
5911  .equ USART_CHSIZE_6BIT_gc = (0x01<<0) ; Character size: 6 bit
5912  .equ USART_CHSIZE_7BIT_gc = (0x02<<0) ; Character size: 7 bit
5913  .equ USART_CHSIZE_8BIT_gc = (0x03<<0) ; Character size: 8 bit
5914  .equ USART_CHSIZE_9BIT_gc = (0x07<<0) ; Character size: 9 bit
5915
5916  ; Communication Mode
5917  .equ USART_CMODE_ASYNCHRONOUS_gc = (0x00<<6) ; Asynchronous Mode
5918  .equ USART_CMODE_SYNCHRONOUS_gc = (0x01<<6) ; Synchronous Mode
5919  .equ USART_CMODE_IRDA_gc = (0x02<<6) ; IrDA Mode
5920  .equ USART_CMODE_MSPI_gc = (0x03<<6) ; Master SPI Mode
5921
5922  ; Parity Mode
5923  .equ USART_PMODE_DISABLED_gc = (0x00<<4) ; No Parity
5924  .equ USART_PMODE_EVEN_gc = (0x02<<4) ; Even Parity
5925  .equ USART_PMODE_ODD_gc = (0x03<<4) ; Odd Parity
5926
5927
5928  ;**************************************************************************
5929  ;** SPI - Serial Peripheral Interface
```

```
;**************************************************************************/

; SPI_CTRL masks
.equ SPI_CLK2X_bm = 0x80 ; Enable Double Speed bit mask
.equ SPI_CLK2X_bp = 7 ; Enable Double Speed bit position
.equ SPI_ENABLE_bm = 0x40 ; Enable Module bit mask
.equ SPI_ENABLE_bp = 6 ; Enable Module bit position
.equ SPI_DORD_bm = 0x20 ; Data Order Setting bit mask
.equ SPI_DORD_bp = 5 ; Data Order Setting bit position
.equ SPI_MASTER_bm = 0x10 ; Master Operation Enable bit mask
.equ SPI_MASTER_bp = 4 ; Master Operation Enable bit position
.equ SPI_MODE_gm = 0x0C ; SPI Mode group mask
.equ SPI_MODE_gp = 2 ; SPI Mode group position
.equ SPI_MODE0_bm = (1<<2) ; SPI Mode bit 0 mask
.equ SPI_MODE0_bp = 2 ; SPI Mode bit 0 position
.equ SPI_MODE1_bm = (1<<3) ; SPI Mode bit 1 mask
.equ SPI_MODE1_bp = 3 ; SPI Mode bit 1 position
.equ SPI_PRESCALER_gm = 0x03 ; Prescaler group mask
.equ SPI_PRESCALER_gp = 0 ; Prescaler group position
.equ SPI_PRESCALER0_bm = (1<<0) ; Prescaler bit 0 mask
.equ SPI_PRESCALER0_bp = 0 ; Prescaler bit 0 position
.equ SPI_PRESCALER1_bm = (1<<1) ; Prescaler bit 1 mask
.equ SPI_PRESCALER1_bp = 1 ; Prescaler bit 1 position

; SPI_INTCTRL masks
.equ SPI_INTLVL_gm = 0x03 ; Interrupt level group mask
.equ SPI_INTLVL_gp = 0 ; Interrupt level group position
.equ SPI_INTLVL0_bm = (1<<0) ; Interrupt level bit 0 mask
.equ SPI_INTLVL0_bp = 0 ; Interrupt level bit 0 position
.equ SPI_INTLVL1_bm = (1<<1) ; Interrupt level bit 1 mask
.equ SPI_INTLVL1_bp = 1 ; Interrupt level bit 1 position

; SPI_STATUS masks
.equ SPI_IF_bm = 0x80 ; Interrupt Flag bit mask
.equ SPI_IF_bp = 7 ; Interrupt Flag bit position
.equ SPI_WRCOL_bm = 0x40 ; Write Collision bit mask
.equ SPI_WRCOL_bp = 6 ; Write Collision bit position

; SPI Mode
.equ SPI_MODE_0_gc = (0x00<<2) ; SPI Mode 0
.equ SPI_MODE_1_gc = (0x01<<2) ; SPI Mode 1
.equ SPI_MODE_2_gc = (0x02<<2) ; SPI Mode 2
.equ SPI_MODE_3_gc = (0x03<<2) ; SPI Mode 3

; Prescaler setting
.equ SPI_PRESCALER_DIV4_gc = (0x00<<0) ; System Clock / 4
.equ SPI_PRESCALER_DIV16_gc = (0x01<<0) ; System Clock / 16
.equ SPI_PRESCALER_DIV64_gc = (0x02<<0) ; System Clock / 64
.equ SPI_PRESCALER_DIV128_gc = (0x03<<0) ; System Clock / 128

; Interrupt level
.equ SPI_INTLVL_OFF_gc = (0x00<<0) ; Interrupt Disabled
.equ SPI_INTLVL_LO_gc = (0x01<<0) ; Low Level
.equ SPI_INTLVL_MED_gc = (0x02<<0) ; Medium Level
.equ SPI_INTLVL_HI_gc = (0x03<<0) ; High Level


;**************************************************************************
;** IRCOM - IR Communication Module
;**************************************************************************/

; IRCOM_CTRL masks
.equ IRCOM_EVSEL_gm = 0x0F ; Event Channel Select group mask
.equ IRCOM_EVSEL_gp = 0 ; Event Channel Select group position
.equ IRCOM_EVSEL0_bm = (1<<0) ; Event Channel Select bit 0 mask
.equ IRCOM_EVSEL0_bp = 0 ; Event Channel Select bit 0 position
```

```
5996   .equ IRCOM_EVSEL1_bm = (1<<1) ; Event Channel Select bit 1 mask
5997   .equ IRCOM_EVSEL1_bp = 1 ; Event Channel Select bit 1 position
5998   .equ IRCOM_EVSEL2_bm = (1<<2) ; Event Channel Select bit 2 mask
5999   .equ IRCOM_EVSEL2_bp = 2 ; Event Channel Select bit 2 position
6000   .equ IRCOM_EVSEL3_bm = (1<<3) ; Event Channel Select bit 3 mask
6001   .equ IRCOM_EVSEL3_bp = 3 ; Event Channel Select bit 3 position
6002
6003   ; Event channel selection
6004   .equ IRDA_EVSEL_OFF_gc = (0x00<<0) ; No Event Source
6005   .equ IRDA_EVSEL_0_gc = (0x08<<0) ; Event Channel 0
6006   .equ IRDA_EVSEL_1_gc = (0x09<<0) ; Event Channel 1
6007   .equ IRDA_EVSEL_2_gc = (0x0A<<0) ; Event Channel 2
6008   .equ IRDA_EVSEL_3_gc = (0x0B<<0) ; Event Channel 3
6009   .equ IRDA_EVSEL_4_gc = (0x0C<<0) ; Event Channel 4
6010   .equ IRDA_EVSEL_5_gc = (0x0D<<0) ; Event Channel 5
6011   .equ IRDA_EVSEL_6_gc = (0x0E<<0) ; Event Channel 6
6012   .equ IRDA_EVSEL_7_gc = (0x0F<<0) ; Event Channel 7
6013
6014
6015   ;**************************************************************************
6016   ;** FUSE - Fuses and Lockbits
6017   ;*************************************************************************/
6018
6019   ; NVM_LOCKBITS_LOCKBITS masks
6020   .equ NVM_LOCKBITS_BLBB_gm = 0xC0 ; Boot Lock Bits - Boot Section group mask
6021   .equ NVM_LOCKBITS_BLBB_gp = 6 ; Boot Lock Bits - Boot Section group position
6022   .equ NVM_LOCKBITS_BLBB0_bm = (1<<6) ; Boot Lock Bits - Boot Section bit 0 mask
6023   .equ NVM_LOCKBITS_BLBB0_bp = 6 ; Boot Lock Bits - Boot Section bit 0 position
6024   .equ NVM_LOCKBITS_BLBB1_bm = (1<<7) ; Boot Lock Bits - Boot Section bit 1 mask
6025   .equ NVM_LOCKBITS_BLBB1_bp = 7 ; Boot Lock Bits - Boot Section bit 1 position
6026   .equ NVM_LOCKBITS_BLBA_gm = 0x30 ; Boot Lock Bits - Application Section group mask
6027   .equ NVM_LOCKBITS_BLBA_gp = 4 ; Boot Lock Bits - Application Section group position
6028   .equ NVM_LOCKBITS_BLBA0_bm = (1<<4) ; Boot Lock Bits - Application Section bit 0 mask
6029   .equ NVM_LOCKBITS_BLBA0_bp = 4 ; Boot Lock Bits - Application Section bit 0 position
6030   .equ NVM_LOCKBITS_BLBA1_bm = (1<<5) ; Boot Lock Bits - Application Section bit 1 mask
6031   .equ NVM_LOCKBITS_BLBA1_bp = 5 ; Boot Lock Bits - Application Section bit 1 position
6032   .equ NVM_LOCKBITS_BLBAT_gm = 0x0C ; Boot Lock Bits - Application Table group mask
6033   .equ NVM_LOCKBITS_BLBAT_gp = 2 ; Boot Lock Bits - Application Table group position
6034   .equ NVM_LOCKBITS_BLBAT0_bm = (1<<2) ; Boot Lock Bits - Application Table bit 0 mask
6035   .equ NVM_LOCKBITS_BLBAT0_bp = 2 ; Boot Lock Bits - Application Table bit 0 position
6036   .equ NVM_LOCKBITS_BLBAT1_bm = (1<<3) ; Boot Lock Bits - Application Table bit 1 mask
6037   .equ NVM_LOCKBITS_BLBAT1_bp = 3 ; Boot Lock Bits - Application Table bit 1 position
6038   .equ NVM_LOCKBITS_LB_gm = 0x03 ; Lock Bits group mask
6039   .equ NVM_LOCKBITS_LB_gp = 0 ; Lock Bits group position
6040   .equ NVM_LOCKBITS_LB0_bm = (1<<0) ; Lock Bits bit 0 mask
6041   .equ NVM_LOCKBITS_LB0_bp = 0 ; Lock Bits bit 0 position
6042   .equ NVM_LOCKBITS_LB1_bm = (1<<1) ; Lock Bits bit 1 mask
6043   .equ NVM_LOCKBITS_LB1_bp = 1 ; Lock Bits bit 1 position
6044
6045   ; NVM_FUSES_FUSEBYTE0 masks
6046   .equ NVM_FUSES_JTAGUSERID_gm = 0xFF ; JTAG User ID group mask
6047   .equ NVM_FUSES_JTAGUSERID_gp = 0 ; JTAG User ID group position
6048   .equ NVM_FUSES_JTAGUSERID0_bm = (1<<0) ; JTAG User ID bit 0 mask
6049   .equ NVM_FUSES_JTAGUSERID0_bp = 0 ; JTAG User ID bit 0 position
6050   .equ NVM_FUSES_JTAGUSERID1_bm = (1<<1) ; JTAG User ID bit 1 mask
6051   .equ NVM_FUSES_JTAGUSERID1_bp = 1 ; JTAG User ID bit 1 position
6052   .equ NVM_FUSES_JTAGUSERID2_bm = (1<<2) ; JTAG User ID bit 2 mask
6053   .equ NVM_FUSES_JTAGUSERID2_bp = 2 ; JTAG User ID bit 2 position
6054   .equ NVM_FUSES_JTAGUSERID3_bm = (1<<3) ; JTAG User ID bit 3 mask
6055   .equ NVM_FUSES_JTAGUSERID3_bp = 3 ; JTAG User ID bit 3 position
6056   .equ NVM_FUSES_JTAGUSERID4_bm = (1<<4) ; JTAG User ID bit 4 mask
6057   .equ NVM_FUSES_JTAGUSERID4_bp = 4 ; JTAG User ID bit 4 position
6058   .equ NVM_FUSES_JTAGUSERID5_bm = (1<<5) ; JTAG User ID bit 5 mask
6059   .equ NVM_FUSES_JTAGUSERID5_bp = 5 ; JTAG User ID bit 5 position
6060   .equ NVM_FUSES_JTAGUSERID6_bm = (1<<6) ; JTAG User ID bit 6 mask
6061   .equ NVM_FUSES_JTAGUSERID6_bp = 6 ; JTAG User ID bit 6 position
```

```
.equ NVM_FUSES_JTAGUSERID7_bm = (1<<7) ; JTAG User ID bit 7 mask
.equ NVM_FUSES_JTAGUSERID7_bp = 7 ; JTAG User ID bit 7 position

; NVM_FUSES_FUSEBYTE1 masks
.equ NVM_FUSES_WDWP_gm = 0xF0 ; Watchdog Window Timeout Period group mask
.equ NVM_FUSES_WDWP_gp = 4 ; Watchdog Window Timeout Period group position
.equ NVM_FUSES_WDWP0_bm = (1<<4) ; Watchdog Window Timeout Period bit 0 mask
.equ NVM_FUSES_WDWP0_bp = 4 ; Watchdog Window Timeout Period bit 0 position
.equ NVM_FUSES_WDWP1_bm = (1<<5) ; Watchdog Window Timeout Period bit 1 mask
.equ NVM_FUSES_WDWP1_bp = 5 ; Watchdog Window Timeout Period bit 1 position
.equ NVM_FUSES_WDWP2_bm = (1<<6) ; Watchdog Window Timeout Period bit 2 mask
.equ NVM_FUSES_WDWP2_bp = 6 ; Watchdog Window Timeout Period bit 2 position
.equ NVM_FUSES_WDWP3_bm = (1<<7) ; Watchdog Window Timeout Period bit 3 mask
.equ NVM_FUSES_WDWP3_bp = 7 ; Watchdog Window Timeout Period bit 3 position
.equ NVM_FUSES_WDP_gm = 0x0F ; Watchdog Timeout Period group mask
.equ NVM_FUSES_WDP_gp = 0 ; Watchdog Timeout Period group position
.equ NVM_FUSES_WDP0_bm = (1<<0) ; Watchdog Timeout Period bit 0 mask
.equ NVM_FUSES_WDP0_bp = 0 ; Watchdog Timeout Period bit 0 position
.equ NVM_FUSES_WDP1_bm = (1<<1) ; Watchdog Timeout Period bit 1 mask
.equ NVM_FUSES_WDP1_bp = 1 ; Watchdog Timeout Period bit 1 position
.equ NVM_FUSES_WDP2_bm = (1<<2) ; Watchdog Timeout Period bit 2 mask
.equ NVM_FUSES_WDP2_bp = 2 ; Watchdog Timeout Period bit 2 position
.equ NVM_FUSES_WDP3_bm = (1<<3) ; Watchdog Timeout Period bit 3 mask
.equ NVM_FUSES_WDP3_bp = 3 ; Watchdog Timeout Period bit 3 position

; NVM_FUSES_FUSEBYTE2 masks
.equ NVM_FUSES_BOOTRST_bm = 0x40 ; Boot Loader Section Reset Vector bit mask
.equ NVM_FUSES_BOOTRST_bp = 6 ; Boot Loader Section Reset Vector bit position
.equ NVM_FUSES_TOSCSEL_bm = 0x20 ; Timer Oscillator pin location bit mask
.equ NVM_FUSES_TOSCSEL_bp = 5 ; Timer Oscillator pin location bit position
.equ NVM_FUSES_BODPD_gm = 0x03 ; BOD Operation in Power-Down Mode group mask
.equ NVM_FUSES_BODPD_gp = 0 ; BOD Operation in Power-Down Mode group position
.equ NVM_FUSES_BODPD0_bm = (1<<0) ; BOD Operation in Power-Down Mode bit 0 mask
.equ NVM_FUSES_BODPD0_bp = 0 ; BOD Operation in Power-Down Mode bit 0 position
.equ NVM_FUSES_BODPD1_bm = (1<<1) ; BOD Operation in Power-Down Mode bit 1 mask
.equ NVM_FUSES_BODPD1_bp = 1 ; BOD Operation in Power-Down Mode bit 1 position

; NVM_FUSES_FUSEBYTE4 masks
.equ NVM_FUSES_RSTDISBL_bm = 0x10 ; External Reset Disable bit mask
.equ NVM_FUSES_RSTDISBL_bp = 4 ; External Reset Disable bit position
.equ NVM_FUSES_SUT_gm = 0x0C ; Start-up Time group mask
.equ NVM_FUSES_SUT_gp = 2 ; Start-up Time group position
.equ NVM_FUSES_SUT0_bm = (1<<2) ; Start-up Time bit 0 mask
.equ NVM_FUSES_SUT0_bp = 2 ; Start-up Time bit 0 position
.equ NVM_FUSES_SUT1_bm = (1<<3) ; Start-up Time bit 1 mask
.equ NVM_FUSES_SUT1_bp = 3 ; Start-up Time bit 1 position
.equ NVM_FUSES_WDLOCK_bm = 0x02 ; Watchdog Timer Lock bit mask
.equ NVM_FUSES_WDLOCK_bp = 1 ; Watchdog Timer Lock bit position
.equ NVM_FUSES_JTAGEN_bm = 0x01 ; JTAG Interface Enable bit mask
.equ NVM_FUSES_JTAGEN_bp = 0 ; JTAG Interface Enable bit position

; NVM_FUSES_FUSEBYTE5 masks
.equ NVM_FUSES_BODACT_gm = 0x30 ; BOD Operation in Active Mode group mask
.equ NVM_FUSES_BODACT_gp = 4 ; BOD Operation in Active Mode group position
.equ NVM_FUSES_BODACT0_bm = (1<<4) ; BOD Operation in Active Mode bit 0 mask
.equ NVM_FUSES_BODACT0_bp = 4 ; BOD Operation in Active Mode bit 0 position
.equ NVM_FUSES_BODACT1_bm = (1<<5) ; BOD Operation in Active Mode bit 1 mask
.equ NVM_FUSES_BODACT1_bp = 5 ; BOD Operation in Active Mode bit 1 position
.equ NVM_FUSES_EESAVE_bm = 0x08 ; Preserve EEPROM Through Chip Erase bit mask
.equ NVM_FUSES_EESAVE_bp = 3 ; Preserve EEPROM Through Chip Erase bit position
.equ NVM_FUSES_BODLVL_gm = 0x07 ; Brown Out Detection Voltage Level group mask
.equ NVM_FUSES_BODLVL_gp = 0 ; Brown Out Detection Voltage Level group position
.equ NVM_FUSES_BODLVL0_bm = (1<<0) ; Brown Out Detection Voltage Level bit 0 mask
.equ NVM_FUSES_BODLVL0_bp = 0 ; Brown Out Detection Voltage Level bit 0 position
.equ NVM_FUSES_BODLVL1_bm = (1<<1) ; Brown Out Detection Voltage Level bit 1 mask
.equ NVM_FUSES_BODLVL1_bp = 1 ; Brown Out Detection Voltage Level bit 1 position
```

```
6128  .equ NVM_FUSES_BODLVL2_bm = (1<<2) ; Brown Out Detection Voltage Level bit 2 mask
6129  .equ NVM_FUSES_BODLVL2_bp = 2 ; Brown Out Detection Voltage Level bit 2 position
6130
6131  ; Boot lock bits - boot setcion
6132  .equ FUSE_BLBB_RWLOCK_gc = (0x00<<6) ; Read and write not allowed
6133  .equ FUSE_BLBB_RLOCK_gc = (0x01<<6) ; Read not allowed
6134  .equ FUSE_BLBB_WLOCK_gc = (0x02<<6) ; Write not allowed
6135  .equ FUSE_BLBB_NOLOCK_gc = (0x03<<6) ; No locks
6136
6137  ; Boot lock bits - application section
6138  .equ FUSE_BLBA_RWLOCK_gc = (0x00<<4) ; Read and write not allowed
6139  .equ FUSE_BLBA_RLOCK_gc = (0x01<<4) ; Read not allowed
6140  .equ FUSE_BLBA_WLOCK_gc = (0x02<<4) ; Write not allowed
6141  .equ FUSE_BLBA_NOLOCK_gc = (0x03<<4) ; No locks
6142
6143  ; Boot lock bits - application table section
6144  .equ FUSE_BLBAT_RWLOCK_gc = (0x00<<2) ; Read and write not allowed
6145  .equ FUSE_BLBAT_RLOCK_gc = (0x01<<2) ; Read not allowed
6146  .equ FUSE_BLBAT_WLOCK_gc = (0x02<<2) ; Write not allowed
6147  .equ FUSE_BLBAT_NOLOCK_gc = (0x03<<2) ; No locks
6148
6149  ; Lock bits
6150  .equ FUSE_LB_RWLOCK_gc = (0x00<<0) ; Read and write not allowed
6151  .equ FUSE_LB_WLOCK_gc = (0x02<<0) ; Write not allowed
6152  .equ FUSE_LB_NOLOCK_gc = (0x03<<0) ; No locks
6153
6154  ; Boot Loader Section Reset Vector
6155  .equ BOOTRST_BOOTLDR_gc = (0x00<<6) ; Boot Loader Reset
6156  .equ BOOTRST_APPLICATION_gc = (0x01<<6) ; Application Reset
6157
6158  ; Timer Oscillator pin location
6159  .equ TOSCSEL_ALTERNATE_gc = (0x00<<5) ; TOSC1 / TOSC2 on separate pins
6160  .equ TOSCSEL_XTAL_gc = (0x01<<5) ; TOSC1 / TOSC2 shared with XTAL1 / XTAL2
6161
6162  ; BOD operation
6163  .equ BOD_SAMPLED_gc = (0x01<<0) ; BOD enabled in sampled mode
6164  .equ BOD_CONTINUOUS_gc = (0x02<<0) ; BOD enabled continuously
6165  .equ BOD_DISABLED_gc = (0x03<<0) ; BOD Disabled
6166
6167  ; Watchdog (Window) Timeout Period
6168  .equ WD_8CLK_gc = (0x00<<4) ; 8 cycles (8ms @ 3.3V)
6169  .equ WD_16CLK_gc = (0x01<<4) ; 16 cycles (16ms @ 3.3V)
6170  .equ WD_32CLK_gc = (0x02<<4) ; 32 cycles (32ms @ 3.3V)
6171  .equ WD_64CLK_gc = (0x03<<4) ; 64 cycles (64ms @ 3.3V)
6172  .equ WD_128CLK_gc = (0x04<<4) ; 128 cycles (0.125s @ 3.3V)
6173  .equ WD_256CLK_gc = (0x05<<4) ; 256 cycles (0.25s @ 3.3V)
6174  .equ WD_512CLK_gc = (0x06<<4) ; 512 cycles (0.5s @ 3.3V)
6175  .equ WD_1KCLK_gc = (0x07<<4) ; 1K cycles (1s @ 3.3V)
6176  .equ WD_2KCLK_gc = (0x08<<4) ; 2K cycles (2s @ 3.3V)
6177  .equ WD_4KCLK_gc = (0x09<<4) ; 4K cycles (4s @ 3.3V)
6178  .equ WD_8KCLK_gc = (0x0A<<4) ; 8K cycles (8s @ 3.3V)
6179
6180  ; Start-up Time
6181  .equ SUT_0MS_gc = (0x03<<2) ; 0 ms
6182  .equ SUT_4MS_gc = (0x01<<2) ; 4 ms
6183  .equ SUT_64MS_gc = (0x00<<2) ; 64 ms
6184
6185  ; Brown Out Detection Voltage Level
6186  .equ BODLVL_1V6_gc = (0x07<<0) ; 1.6 V
6187  .equ BODLVL_1V8_gc = (0x06<<0) ; 1.8 V
6188  .equ BODLVL_2V0_gc = (0x05<<0) ; 2.0 V
6189  .equ BODLVL_2V2_gc = (0x04<<0) ; 2.2 V
6190  .equ BODLVL_2V4_gc = (0x03<<0) ; 2.4 V
6191  .equ BODLVL_2V6_gc = (0x02<<0) ; 2.6 V
6192  .equ BODLVL_2V8_gc = (0x01<<0) ; 2.8 V
6193  .equ BODLVL_3V0_gc = (0x00<<0) ; 3.0 V
```

```
6194
6195
6196   ;***************************************************************************
6197   ;** SIGROW - Signature Row
6198   ;***************************************************************************/
6199
6200
6201
6202
6203   ; ***** CPU REGISTER DEFINITIONS ***************************************
6204   .def    XH  = r27
6205   .def    XL  = r26
6206   .def    YH  = r29
6207   .def    YL  = r28
6208   .def    ZH  = r31
6209   .def    ZL  = r30
6210
6211
6212   ; ***** DATA MEMORY DECLARATIONS ***************************************
6213
6214
6215   #define PROGMEM_START 0x00000
6216   #define PROGMEM_SIZE 0x22000
6217   #define PROGMEM_END (0x00000 + 0x22000 - 1)
6218
6219   #define APP_SECTION_START 0x00000
6220   #define APP_SECTION_SIZE 0x20000
6221   #define APP_SECTION_END (0x00000 + 0x20000 - 1)
6222
6223   #define APPTABLE_SECTION_START 0x1E000
6224   #define APPTABLE_SECTION_SIZE 0x2000
6225   #define APPTABLE_SECTION_END (0x1E000 + 0x2000 - 1)
6226
6227   #define BOOT_SECTION_START 0x20000
6228   #define BOOT_SECTION_SIZE 0x2000
6229   #define BOOT_SECTION_END (0x20000 + 0x2000 - 1)
6230
6231   #define EEPROM_START 0x00000
6232   #define EEPROM_SIZE 0x0800
6233   #define EEPROM_END (0x00000 + 0x0800 - 1)
6234
6235   #define FUSE_START 0x0000
6236   #define FUSE_SIZE 0x0006
6237   #define FUSE_END (0x0000 + 0x0006 - 1)
6238
6239   #define LOCKBIT_START 0x0000
6240   #define LOCKBIT_SIZE 0x0001
6241   #define LOCKBIT_END (0x0000 + 0x0001 - 1)
6242
6243   #define SIGNATURES_START 0x0000
6244   #define SIGNATURES_SIZE 0x0003
6245   #define SIGNATURES_END (0x0000 + 0x0003 - 1)
6246
6247   #define USER_SIGNATURES_START 0x0000
6248   #define USER_SIGNATURES_SIZE 0x0200
6249   #define USER_SIGNATURES_END (0x0000 + 0x0200 - 1)
6250
6251   #define PROD_SIGNATURES_START 0x0000
6252   #define PROD_SIGNATURES_SIZE 0x0040
6253   #define PROD_SIGNATURES_END (0x0000 + 0x0040 - 1)
6254
6255   #define DATAMEM_START 0x0000
6256   #define DATAMEM_SIZE 0x1000000
6257   #define DATAMEM_END (0x0000 + 0x1000000 - 1)
6258
6259   #define IO_START 0x0000
```

```
6260  #define IO_SIZE 0x001000
6261  #define IO_END (0x0000 + 0x001000 - 1)
6262
6263  #define MAPPED_EEPROM_START 0x1000
6264  #define MAPPED_EEPROM_SIZE 0x000800
6265  #define MAPPED_EEPROM_END (0x1000 + 0x000800 - 1)
6266
6267  #define INTERNAL_SRAM_START 0x2000
6268  #define INTERNAL_SRAM_SIZE 0x002000
6269  #define INTERNAL_SRAM_END (0x2000 + 0x002000 - 1)
6270
6271  #define EXTERNAL_SRAM_START 0x4000
6272  #define EXTERNAL_SRAM_SIZE 0xFFC000
6273  #define EXTERNAL_SRAM_END (0x4000 + 0xFFC000 - 1)
6274
6275
6276  ; Legacy definitions
6277  .equ    FLASHEND    = (PROGMEM_END / 2)  ; Note: Word address
6278  .equ    IOEND       = IO_END
6279  .equ    SRAM_START  = INTERNAL_SRAM_START
6280  .equ    SRAM_SIZE   = INTERNAL_SRAM_SIZE
6281  .equ    RAMEND      = INTERNAL_SRAM_END
6282  .equ    XRAMEND     = EXTERNAL_SRAM_END
6283  .equ    E2END       = EEPROM_END
6284  .equ    EEPROMEND   = EEPROM_END
6285
6286
6287  ; Definitions used by the assembler
6288  #pragma AVRPART MEMORY PROG_FLASH 0x22000
6289  #pragma AVRPART MEMORY EEPROM 0x0800
6290  #pragma AVRPART MEMORY INT_SRAM SIZE 0x002000
6291  #pragma AVRPART MEMORY INT_SRAM START_ADDR 0x2000
6292
6293
6294
6295  ; ***** INTERRUPT VECTORS, ABSOLUTE ADDRESSES ****************************
6296
6297  ; OSC interrupt vectors
6298  .equ OSC_OSCF_vect = 2 ; Oscillator Failure Interrupt (NMI)
6299
6300  ; PORTC interrupt vectors
6301  .equ PORTC_INT0_vect = 4 ; External Interrupt 0
6302  .equ PORTC_INT1_vect = 6 ; External Interrupt 1
6303
6304  ; PORTR interrupt vectors
6305  .equ PORTR_INT0_vect = 8 ; External Interrupt 0
6306  .equ PORTR_INT1_vect = 10 ; External Interrupt 1
6307
6308  ; DMA interrupt vectors
6309  .equ DMA_CH0_vect = 12 ; Channel 0 Interrupt
6310  .equ DMA_CH1_vect = 14 ; Channel 1 Interrupt
6311  .equ DMA_CH2_vect = 16 ; Channel 2 Interrupt
6312  .equ DMA_CH3_vect = 18 ; Channel 3 Interrupt
6313
6314  ; RTC interrupt vectors
6315  .equ RTC_OVF_vect = 20 ; Overflow Interrupt
6316  .equ RTC_COMP_vect = 22 ; Compare Interrupt
6317
6318  ; TWIC interrupt vectors
6319  .equ TWIC_TWIS_vect = 24 ; TWI Slave Interrupt
6320  .equ TWIC_TWIM_vect = 26 ; TWI Master Interrupt
6321
6322  ; TCC0 interrupt vectors
6323  .equ TCC0_OVF_vect = 28 ; Overflow Interrupt
6324  .equ TCC0_ERR_vect = 30 ; Error Interrupt
6325  .equ TCC0_CCA_vect = 32 ; Compare or Capture A Interrupt
```

```
6326  .equ TCC0_CCB_vect = 34 ; Compare or Capture B Interrupt
6327  .equ TCC0_CCC_vect = 36 ; Compare or Capture C Interrupt
6328  .equ TCC0_CCD_vect = 38 ; Compare or Capture D Interrupt
6329
6330  ; TCC2 interrupt vectors
6331  .equ TCC2_LUNF_vect = 28 ; Low Byte Underflow Interrupt
6332  .equ TCC2_HUNF_vect = 30 ; High Byte Underflow Interrupt
6333  .equ TCC2_LCMPA_vect = 32 ; Low Byte Compare A Interrupt
6334  .equ TCC2_LCMPB_vect = 34 ; Low Byte Compare B Interrupt
6335  .equ TCC2_LCMPC_vect = 36 ; Low Byte Compare C Interrupt
6336  .equ TCC2_LCMPD_vect = 38 ; Low Byte Compare D Interrupt
6337
6338  ; TCC1 interrupt vectors
6339  .equ TCC1_OVF_vect = 40 ; Overflow Interrupt
6340  .equ TCC1_ERR_vect = 42 ; Error Interrupt
6341  .equ TCC1_CCA_vect = 44 ; Compare or Capture A Interrupt
6342  .equ TCC1_CCB_vect = 46 ; Compare or Capture B Interrupt
6343
6344  ; SPIC interrupt vectors
6345  .equ SPIC_INT_vect = 48 ; SPI Interrupt
6346
6347  ; USARTC0 interrupt vectors
6348  .equ USARTC0_RXC_vect = 50 ; Reception Complete Interrupt
6349  .equ USARTC0_DRE_vect = 52 ; Data Register Empty Interrupt
6350  .equ USARTC0_TXC_vect = 54 ; Transmission Complete Interrupt
6351
6352  ; USARTC1 interrupt vectors
6353  .equ USARTC1_RXC_vect = 56 ; Reception Complete Interrupt
6354  .equ USARTC1_DRE_vect = 58 ; Data Register Empty Interrupt
6355  .equ USARTC1_TXC_vect = 60 ; Transmission Complete Interrupt
6356
6357  ; AES interrupt vectors
6358  .equ AES_INT_vect = 62 ; AES Interrupt
6359
6360  ; NVM interrupt vectors
6361  .equ NVM_EE_vect = 64 ; EE Interrupt
6362  .equ NVM_SPM_vect = 66 ; SPM Interrupt
6363
6364  ; PORTB interrupt vectors
6365  .equ PORTB_INT0_vect = 68 ; External Interrupt 0
6366  .equ PORTB_INT1_vect = 70 ; External Interrupt 1
6367
6368  ; ACB interrupt vectors
6369  .equ ACB_AC0_vect = 72 ; AC0 Interrupt
6370  .equ ACB_AC1_vect = 74 ; AC1 Interrupt
6371  .equ ACB_ACW_vect = 76 ; ACW Window Mode Interrupt
6372
6373  ; ADCB interrupt vectors
6374  .equ ADCB_CH0_vect = 78 ; Interrupt 0
6375  .equ ADCB_CH1_vect = 80 ; Interrupt 1
6376  .equ ADCB_CH2_vect = 82 ; Interrupt 2
6377  .equ ADCB_CH3_vect = 84 ; Interrupt 3
6378
6379  ; PORTE interrupt vectors
6380  .equ PORTE_INT0_vect = 86 ; External Interrupt 0
6381  .equ PORTE_INT1_vect = 88 ; External Interrupt 1
6382
6383  ; TWIE interrupt vectors
6384  .equ TWIE_TWIS_vect = 90 ; TWI Slave Interrupt
6385  .equ TWIE_TWIM_vect = 92 ; TWI Master Interrupt
6386
6387  ; TCE0 interrupt vectors
6388  .equ TCE0_OVF_vect = 94 ; Overflow Interrupt
6389  .equ TCE0_ERR_vect = 96 ; Error Interrupt
6390  .equ TCE0_CCA_vect = 98 ; Compare or Capture A Interrupt
6391  .equ TCE0_CCB_vect = 100 ; Compare or Capture B Interrupt
```

```
6392  .equ TCE0_CCC_vect = 102 ; Compare or Capture C Interrupt
6393  .equ TCE0_CCD_vect = 104 ; Compare or Capture D Interrupt
6394
6395  ; TCE2 interrupt vectors
6396  .equ TCE2_LUNF_vect = 94 ; Low Byte Underflow Interrupt
6397  .equ TCE2_HUNF_vect = 96 ; High Byte Underflow Interrupt
6398  .equ TCE2_LCMPA_vect = 98 ; Low Byte Compare A Interrupt
6399  .equ TCE2_LCMPB_vect = 100 ; Low Byte Compare B Interrupt
6400  .equ TCE2_LCMPC_vect = 102 ; Low Byte Compare C Interrupt
6401  .equ TCE2_LCMPD_vect = 104 ; Low Byte Compare D Interrupt
6402
6403  ; TCE1 interrupt vectors
6404  .equ TCE1_OVF_vect = 106 ; Overflow Interrupt
6405  .equ TCE1_ERR_vect = 108 ; Error Interrupt
6406  .equ TCE1_CCA_vect = 110 ; Compare or Capture A Interrupt
6407  .equ TCE1_CCB_vect = 112 ; Compare or Capture B Interrupt
6408
6409  ; SPIE interrupt vectors
6410  .equ SPIE_INT_vect = 114 ; SPI Interrupt
6411
6412  ; USARTE0 interrupt vectors
6413  .equ USARTE0_RXC_vect = 116 ; Reception Complete Interrupt
6414  .equ USARTE0_DRE_vect = 118 ; Data Register Empty Interrupt
6415  .equ USARTE0_TXC_vect = 120 ; Transmission Complete Interrupt
6416
6417  ; USARTE1 interrupt vectors
6418  .equ USARTE1_RXC_vect = 122 ; Reception Complete Interrupt
6419  .equ USARTE1_DRE_vect = 124 ; Data Register Empty Interrupt
6420  .equ USARTE1_TXC_vect = 126 ; Transmission Complete Interrupt
6421
6422  ; PORTD interrupt vectors
6423  .equ PORTD_INT0_vect = 128 ; External Interrupt 0
6424  .equ PORTD_INT1_vect = 130 ; External Interrupt 1
6425
6426  ; PORTA interrupt vectors
6427  .equ PORTA_INT0_vect = 132 ; External Interrupt 0
6428  .equ PORTA_INT1_vect = 134 ; External Interrupt 1
6429
6430  ; ACA interrupt vectors
6431  .equ ACA_AC0_vect = 136 ; AC0 Interrupt
6432  .equ ACA_AC1_vect = 138 ; AC1 Interrupt
6433  .equ ACA_ACW_vect = 140 ; ACW Window Mode Interrupt
6434
6435  ; ADCA interrupt vectors
6436  .equ ADCA_CH0_vect = 142 ; Interrupt 0
6437  .equ ADCA_CH1_vect = 144 ; Interrupt 1
6438  .equ ADCA_CH2_vect = 146 ; Interrupt 2
6439  .equ ADCA_CH3_vect = 148 ; Interrupt 3
6440
6441  ; TWID interrupt vectors
6442  .equ TWID_TWIS_vect = 150 ; TWI Slave Interrupt
6443  .equ TWID_TWIM_vect = 152 ; TWI Master Interrupt
6444
6445  ; TCD0 interrupt vectors
6446  .equ TCD0_OVF_vect = 154 ; Overflow Interrupt
6447  .equ TCD0_ERR_vect = 156 ; Error Interrupt
6448  .equ TCD0_CCA_vect = 158 ; Compare or Capture A Interrupt
6449  .equ TCD0_CCB_vect = 160 ; Compare or Capture B Interrupt
6450  .equ TCD0_CCC_vect = 162 ; Compare or Capture C Interrupt
6451  .equ TCD0_CCD_vect = 164 ; Compare or Capture D Interrupt
6452
6453  ; TCD2 interrupt vectors
6454  .equ TCD2_LUNF_vect = 154 ; Low Byte Underflow Interrupt
6455  .equ TCD2_HUNF_vect = 156 ; High Byte Underflow Interrupt
6456  .equ TCD2_LCMPA_vect = 158 ; Low Byte Compare A Interrupt
6457  .equ TCD2_LCMPB_vect = 160 ; Low Byte Compare B Interrupt
```

```
6458  .equ TCD2_LCMPC_vect = 162 ; Low Byte Compare C Interrupt
6459  .equ TCD2_LCMPD_vect = 164 ; Low Byte Compare D Interrupt
6460
6461  ; TCD1 interrupt vectors
6462  .equ TCD1_OVF_vect = 166 ; Overflow Interrupt
6463  .equ TCD1_ERR_vect = 168 ; Error Interrupt
6464  .equ TCD1_CCA_vect = 170 ; Compare or Capture A Interrupt
6465  .equ TCD1_CCB_vect = 172 ; Compare or Capture B Interrupt
6466
6467  ; SPID interrupt vectors
6468  .equ SPID_INT_vect = 174 ; SPI Interrupt
6469
6470  ; USARTD0 interrupt vectors
6471  .equ USARTD0_RXC_vect = 176 ; Reception Complete Interrupt
6472  .equ USARTD0_DRE_vect = 178 ; Data Register Empty Interrupt
6473  .equ USARTD0_TXC_vect = 180 ; Transmission Complete Interrupt
6474
6475  ; USARTD1 interrupt vectors
6476  .equ USARTD1_RXC_vect = 182 ; Reception Complete Interrupt
6477  .equ USARTD1_DRE_vect = 184 ; Data Register Empty Interrupt
6478  .equ USARTD1_TXC_vect = 186 ; Transmission Complete Interrupt
6479
6480  ; PORTQ interrupt vectors
6481  .equ PORTQ_INT0_vect = 188 ; External Interrupt 0
6482  .equ PORTQ_INT1_vect = 190 ; External Interrupt 1
6483
6484  ; PORTH interrupt vectors
6485  .equ PORTH_INT0_vect = 192 ; External Interrupt 0
6486  .equ PORTH_INT1_vect = 194 ; External Interrupt 1
6487
6488  ; PORTJ interrupt vectors
6489  .equ PORTJ_INT0_vect = 196 ; External Interrupt 0
6490  .equ PORTJ_INT1_vect = 198 ; External Interrupt 1
6491
6492  ; PORTK interrupt vectors
6493  .equ PORTK_INT0_vect = 200 ; External Interrupt 0
6494  .equ PORTK_INT1_vect = 202 ; External Interrupt 1
6495
6496  ; PORTF interrupt vectors
6497  .equ PORTF_INT0_vect = 208 ; External Interrupt 0
6498  .equ PORTF_INT1_vect = 210 ; External Interrupt 1
6499
6500  ; TWIF interrupt vectors
6501  .equ TWIF_TWIS_vect = 212 ; TWI Slave Interrupt
6502  .equ TWIF_TWIM_vect = 214 ; TWI Master Interrupt
6503
6504  ; TCF0 interrupt vectors
6505  .equ TCF0_OVF_vect = 216 ; Overflow Interrupt
6506  .equ TCF0_ERR_vect = 218 ; Error Interrupt
6507  .equ TCF0_CCA_vect = 220 ; Compare or Capture A Interrupt
6508  .equ TCF0_CCB_vect = 222 ; Compare or Capture B Interrupt
6509  .equ TCF0_CCC_vect = 224 ; Compare or Capture C Interrupt
6510  .equ TCF0_CCD_vect = 226 ; Compare or Capture D Interrupt
6511
6512  ; TCF2 interrupt vectors
6513  .equ TCF2_LUNF_vect = 216 ; Low Byte Underflow Interrupt
6514  .equ TCF2_HUNF_vect = 218 ; High Byte Underflow Interrupt
6515  .equ TCF2_LCMPA_vect = 220 ; Low Byte Compare A Interrupt
6516  .equ TCF2_LCMPB_vect = 222 ; Low Byte Compare B Interrupt
6517  .equ TCF2_LCMPC_vect = 224 ; Low Byte Compare C Interrupt
6518  .equ TCF2_LCMPD_vect = 226 ; Low Byte Compare D Interrupt
6519
6520  ; TCF1 interrupt vectors
6521  .equ TCF1_OVF_vect = 228 ; Overflow Interrupt
6522  .equ TCF1_ERR_vect = 230 ; Error Interrupt
6523  .equ TCF1_CCA_vect = 232 ; Compare or Capture A Interrupt
```

```
6524  .equ TCF1_CCB_vect = 234 ; Compare or Capture B Interrupt
6525
6526  ; SPIF interrupt vectors
6527  .equ SPIF_INT_vect = 236 ; SPI Interrupt
6528
6529  ; USARTF0 interrupt vectors
6530  .equ USARTF0_RXC_vect = 238 ; Reception Complete Interrupt
6531  .equ USARTF0_DRE_vect = 240 ; Data Register Empty Interrupt
6532  .equ USARTF0_TXC_vect = 242 ; Transmission Complete Interrupt
6533
6534  ; USARTF1 interrupt vectors
6535  .equ USARTF1_RXC_vect = 244 ; Reception Complete Interrupt
6536  .equ USARTF1_DRE_vect = 246 ; Data Register Empty Interrupt
6537  .equ USARTF1_TXC_vect = 248 ; Transmission Complete Interrupt
6538
6539  ; USB interrupt vectors
6540  .equ USB_BUSEVENT_vect = 250 ; SOF, suspend, resume, reset bus event interrupts, crc,
     underflow, overflow and stall error interrupts
6541  .equ USB_TRNCOMPL_vect = 252 ; Transaction complete interrupt
6542
6543
6544
6545  ; ***** INTERRUPT VECTORS, MODULE BASES **********************************
6546
6547  .equ OSC_vbase = 2
6548  .equ PORTC_vbase = 4
6549  .equ PORTR_vbase = 8
6550  .equ DMA_vbase = 12
6551  .equ RTC_vbase = 20
6552  .equ TWIC_vbase = 24
6553  .equ TCC0_vbase = 28
6554  .equ TCC2_vbase = 28
6555  .equ TCC1_vbase = 40
6556  .equ SPIC_vbase = 48
6557  .equ USARTC0_vbase = 50
6558  .equ USARTC1_vbase = 56
6559  .equ AES_vbase = 62
6560  .equ NVM_vbase = 64
6561  .equ PORTB_vbase = 68
6562  .equ ACB_vbase = 72
6563  .equ ADCB_vbase = 78
6564  .equ PORTE_vbase = 86
6565  .equ TWIE_vbase = 90
6566  .equ TCE0_vbase = 94
6567  .equ TCE2_vbase = 94
6568  .equ TCE1_vbase = 106
6569  .equ SPIE_vbase = 114
6570  .equ USARTE0_vbase = 116
6571  .equ USARTE1_vbase = 122
6572  .equ PORTD_vbase = 128
6573  .equ PORTA_vbase = 132
6574  .equ ACA_vbase = 136
6575  .equ ADCA_vbase = 142
6576  .equ TWID_vbase = 150
6577  .equ TCD0_vbase = 154
6578  .equ TCD2_vbase = 154
6579  .equ TCD1_vbase = 166
6580  .equ SPID_vbase = 174
6581  .equ USARTD0_vbase = 176
6582  .equ USARTD1_vbase = 182
6583  .equ PORTQ_vbase = 188
6584  .equ PORTH_vbase = 192
6585  .equ PORTJ_vbase = 196
6586  .equ PORTK_vbase = 200
6587  .equ PORTF_vbase = 208
6588  .equ TWIF_vbase = 212
```

```
6589   .equ TCF0_vbase = 216
6590   .equ TCF2_vbase = 216
6591   .equ TCF1_vbase = 228
6592   .equ SPIF_vbase = 236
6593   .equ USARTF0_vbase = 238
6594   .equ USARTF1_vbase = 244
6595   .equ USB_vbase = 250
6596
6597
6598   ; ***** INTERRUPT VECTORS, VECTOR OFFSETS *********************************
6599
6600   ; OSC interrupt vector offsets
6601
6602   .equ OSC_OSCF_voffset = 0
6603
6604   ; AES interrupt vector offsets
6605
6606   .equ AES_INT_voffset = 0
6607
6608   ; DMA interrupt vector offsets
6609
6610   .equ DMA_CH0_voffset = 0
6611   .equ DMA_CH1_voffset = 2
6612   .equ DMA_CH2_voffset = 4
6613   .equ DMA_CH3_voffset = 6
6614
6615   ; NVM interrupt vector offsets
6616
6617   .equ NVM_EE_voffset = 0
6618   .equ NVM_SPM_voffset = 2
6619
6620   ; ADC interrupt vector offsets
6621
6622   .equ ADC_CH0_voffset = 0
6623   .equ ADC_CH1_voffset = 2
6624   .equ ADC_CH2_voffset = 4
6625   .equ ADC_CH3_voffset = 6
6626
6627   ; AC interrupt vector offsets
6628
6629   .equ AC_AC0_voffset = 0
6630   .equ AC_AC1_voffset = 2
6631   .equ AC_ACW_voffset = 4
6632
6633   ; RTC interrupt vector offsets
6634
6635   .equ RTC_OVF_voffset = 0
6636   .equ RTC_COMP_voffset = 2
6637
6638   ; TWI interrupt vector offsets
6639
6640   .equ TWI_TWIS_voffset = 0
6641   .equ TWI_TWIM_voffset = 2
6642
6643   ; USB interrupt vector offsets
6644
6645   .equ USB_BUSEVENT_voffset = 0
6646   .equ USB_TRNCOMPL_voffset = 2
6647
6648   ; PORT interrupt vector offsets
6649
6650   .equ PORT_INT0_voffset = 0
6651   .equ PORT_INT1_voffset = 2
6652
6653   ; TC0 interrupt vector offsets
6654
```

```
6655  .equ TC0_OVF_voffset = 0
6656  .equ TC0_ERR_voffset = 2
6657  .equ TC0_CCA_voffset = 4
6658  .equ TC0_CCB_voffset = 6
6659  .equ TC0_CCC_voffset = 8
6660  .equ TC0_CCD_voffset = 10
6661
6662  ; TC1 interrupt vector offsets
6663
6664  .equ TC1_OVF_voffset = 0
6665  .equ TC1_ERR_voffset = 2
6666  .equ TC1_CCA_voffset = 4
6667  .equ TC1_CCB_voffset = 6
6668
6669  ; TC2 interrupt vector offsets
6670
6671  .equ TC2_LUNF_voffset = 0
6672  .equ TC2_HUNF_voffset = 2
6673  .equ TC2_LCMPA_voffset = 4
6674  .equ TC2_LCMPB_voffset = 6
6675  .equ TC2_LCMPC_voffset = 8
6676  .equ TC2_LCMPD_voffset = 10
6677
6678  ; USART interrupt vector offsets
6679
6680  .equ USART_RXC_voffset = 0
6681  .equ USART_DRE_voffset = 2
6682  .equ USART_TXC_voffset = 4
6683
6684  ; SPI interrupt vector offsets
6685
6686  .equ SPI_INT_voffset = 0
6687
6688
6689
6690  .equ INT_VECTORS_SIZE = 254 ; size in words
6691
6692
6693  #endif  /* _ATxmega128A1UDEF_INC_ */
6694
6695  ; ***** END OF FILE ******************************************************
6696
6697
6698
6699
```