

```

;***** THIS IS A MACHINE GENERATED FILE - DO NOT EDIT *****
;***** Created: 2009-11-03 14:40 ***** Source: ATmega328P.xml *****
;*****
;* 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        : "m328Pdef.inc"
;* Title            : Register/Bit Definitions for the ATmega328P
;* Date             : 2009-11-03
;* Version          : 2.35
;* Support E-mail   : avr@atmel.com
;* Target MCU       : ATmega328P
;*
;* 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
;*
;* The Register names are represented by their hexadecimal address.
;*
;* The Register Bit names are represented by their bit number (0-7).
;*
;* Please observe the difference in using the bit names with instructions
;* such as "sbr"/"cbr" (set/clear bit in register) and "sbrs"/"sbrc"
;* (skip if bit in register set/cleared). The following example illustrates
;* this:
;*
;* in    r16,PORTB           ;read PORTB latch
;* sbr   r16,(1<<PB6)+(1<<PB5) ;set PB6 and PB5 (use masks, not bit#)
;* out   PORTB,r16          ;output to PORTB
;*
;* in    r16,TIFR           ;read the Timer Interrupt Flag Register
;* sbrc  r16,TOV0           ;test the overflow flag (use bit#)
;* rjmp  TOV0_is_set        ;jump if set
;* ...                      ;otherwise do something else
;*****

#ifndef _M328PDEF_INC_
#define _M328PDEF_INC_

#pragma partinc 0

; ***** SPECIFY DEVICE *****
.device ATmega328P
#pragma AVRPART ADMIN PART_NAME ATmega328P
.equ    SIGNATURE_000    = 0x1e
.equ    SIGNATURE_001    = 0x95
.equ    SIGNATURE_002    = 0x0f

#pragma AVRPART CORE CORE_VERSION V2E

```

```

; ***** I/O REGISTER DEFINITIONS *****
; NOTE:
; Definitions marked "MEMORY MAPPED" are extended I/O ports
; and cannot be used with IN/OUT instructions
.equ    UDR0      = 0xc6 ; MEMORY MAPPED
.equ    UBRR0L    = 0xc4 ; MEMORY MAPPED
.equ    UBRR0H    = 0xc5 ; MEMORY MAPPED
.equ    UCSR0C    = 0xc2 ; MEMORY MAPPED
.equ    UCSR0B    = 0xc1 ; MEMORY MAPPED
.equ    UCSR0A    = 0xc0 ; MEMORY MAPPED
.equ    TWAMR     = 0xbd ; MEMORY MAPPED
.equ    TWCR      = 0xbc ; MEMORY MAPPED
.equ    TWDR      = 0xbb ; MEMORY MAPPED
.equ    TWAR      = 0xba ; MEMORY MAPPED
.equ    TWSR      = 0xb9 ; MEMORY MAPPED
.equ    TWBR      = 0xb8 ; MEMORY MAPPED
.equ    ASSR      = 0xb6 ; MEMORY MAPPED
.equ    OCR2B     = 0xb4 ; MEMORY MAPPED
.equ    OCR2A     = 0xb3 ; MEMORY MAPPED
.equ    TCNT2     = 0xb2 ; MEMORY MAPPED
.equ    TCCR2B    = 0xb1 ; MEMORY MAPPED
.equ    TCCR2A    = 0xb0 ; MEMORY MAPPED
.equ    OCR1BL    = 0x8a ; MEMORY MAPPED
.equ    OCR1BH    = 0x8b ; MEMORY MAPPED
.equ    OCR1AL    = 0x88 ; MEMORY MAPPED
.equ    OCR1AH    = 0x89 ; MEMORY MAPPED
.equ    ICR1L     = 0x86 ; MEMORY MAPPED
.equ    ICR1H     = 0x87 ; MEMORY MAPPED
.equ    TCNT1L    = 0x84 ; MEMORY MAPPED
.equ    TCNT1H    = 0x85 ; MEMORY MAPPED
.equ    TCCR1C    = 0x82 ; MEMORY MAPPED
.equ    TCCR1B    = 0x81 ; MEMORY MAPPED
.equ    TCCR1A    = 0x80 ; MEMORY MAPPED
.equ    DIDR1     = 0x7f ; MEMORY MAPPED
.equ    DIDR0     = 0x7e ; MEMORY MAPPED
.equ    ADMUX     = 0x7c ; MEMORY MAPPED
.equ    ADCSRB    = 0x7b ; MEMORY MAPPED
.equ    ADCSRA    = 0x7a ; MEMORY MAPPED
.equ    ADCH      = 0x79 ; MEMORY MAPPED
.equ    ADCL      = 0x78 ; MEMORY MAPPED
.equ    TIMSK2    = 0x70 ; MEMORY MAPPED
.equ    TIMSK1    = 0x6f ; MEMORY MAPPED
.equ    TIMSK0    = 0x6e ; MEMORY MAPPED
.equ    PCMSK1    = 0x6c ; MEMORY MAPPED
.equ    PCMSK2    = 0x6d ; MEMORY MAPPED
.equ    PCMSK0    = 0x6b ; MEMORY MAPPED
.equ    EICRA     = 0x69 ; MEMORY MAPPED
.equ    PCICR     = 0x68 ; MEMORY MAPPED
.equ    OSCCAL    = 0x66 ; MEMORY MAPPED
.equ    PRR       = 0x64 ; MEMORY MAPPED
.equ    CLKPR     = 0x61 ; MEMORY MAPPED
.equ    WDTCSR    = 0x60 ; MEMORY MAPPED
.equ    SREG      = 0x3f
.equ    SPL       = 0x3d
.equ    SPH       = 0x3e

```

```

.equ    SPMCSR    = 0x37
.equ    MCUCR     = 0x35
.equ    MCUSR     = 0x34
.equ    SMCR      = 0x33
.equ    ACSR      = 0x30
.equ    SPDR      = 0x2e
.equ    SPSR      = 0x2d
.equ    SPCR      = 0x2c
.equ    GPIOR2    = 0x2b
.equ    GPIOR1    = 0x2a
.equ    OCR0B     = 0x28
.equ    OCR0A     = 0x27
.equ    TCNT0     = 0x26
.equ    TCCR0B    = 0x25
.equ    TCCR0A    = 0x24
.equ    GTCCR     = 0x23
.equ    EEARH     = 0x22
.equ    EEARL     = 0x21
.equ    EEDR      = 0x20
.equ    EECR      = 0x1f
.equ    GPIOR0    = 0x1e
.equ    EIMSK     = 0x1d
.equ    EIFR      = 0x1c
.equ    PCIFR     = 0x1b
.equ    TIFR2     = 0x17
.equ    TIFR1     = 0x16
.equ    TIFR0     = 0x15
.equ    PORTD     = 0x0b
.equ    DDRD      = 0x0a
.equ    PIND      = 0x09
.equ    PORTC     = 0x08
.equ    DDRC      = 0x07
.equ    PINC      = 0x06
.equ    PORTB     = 0x05
.equ    DDRB      = 0x04
.equ    PINB      = 0x03

```

```
; ***** BIT DEFINITIONS *****
```

```
; ***** USART0 *****
```

```
; UDR0 - USART I/O Data Register
```

```

.equ    UDR0_0    = 0 ; USART I/O Data Register bit 0
.equ    UDR0_1    = 1 ; USART I/O Data Register bit 1
.equ    UDR0_2    = 2 ; USART I/O Data Register bit 2
.equ    UDR0_3    = 3 ; USART I/O Data Register bit 3
.equ    UDR0_4    = 4 ; USART I/O Data Register bit 4
.equ    UDR0_5    = 5 ; USART I/O Data Register bit 5
.equ    UDR0_6    = 6 ; USART I/O Data Register bit 6
.equ    UDR0_7    = 7 ; USART I/O Data Register bit 7

```

```
; UCSR0A - USART Control and Status Register A
```

```

.equ    MPCM0     = 0 ; Multi-processor Communication Mode
.equ    U2X0      = 1 ; Double the USART transmission speed
.equ    UPE0      = 2 ; Parity Error

```

```

.equ    DOR0    = 3 ; Data overRun
.equ    FE0     = 4 ; Framing Error
.equ    UDRE0   = 5 ; USART Data Register Empty
.equ    TXC0    = 6 ; USART Transmitt Complete
.equ    RXC0    = 7 ; USART Receive Complete

; UCSR0B - USART Control and Status Register B
.equ    TXB80   = 0 ; Transmit Data Bit 8
.equ    RXB80   = 1 ; Receive Data Bit 8
.equ    UCSZ02  = 2 ; Character Size
.equ    TXEN0   = 3 ; Transmitter Enable
.equ    RXEN0   = 4 ; Receiver Enable
.equ    UDRIE0  = 5 ; USART Data register Empty Interrupt Enable
.equ    TXCIE0  = 6 ; TX Complete Interrupt Enable
.equ    RXCIE0  = 7 ; RX Complete Interrupt Enable

; UCSR0C - USART Control and Status Register C
.equ    UCPOL0  = 0 ; Clock Polarity
.equ    UCSZ00  = 1 ; Character Size
.equ    UCPHA0  = UCSZ00 ; For compatibility
.equ    UCSZ01  = 2 ; Character Size
.equ    UDORD0  = UCSZ01 ; For compatibility
.equ    USBS0   = 3 ; Stop Bit Select
.equ    UPM00   = 4 ; Parity Mode Bit 0
.equ    UPM01   = 5 ; Parity Mode Bit 1
.equ    UMSEL00 = 6 ; USART Mode Select
.equ    UMSEL0  = UMSEL00 ; For compatibility
.equ    UMSEL01 = 7 ; USART Mode Select
.equ    UMSEL1  = UMSEL01 ; For compatibility

; UBRR0H - USART Baud Rate Register High Byte
.equ    UBRR8   = 0 ; USART Baud Rate Register bit 8
.equ    UBRR9   = 1 ; USART Baud Rate Register bit 9
.equ    UBRR10  = 2 ; USART Baud Rate Register bit 10
.equ    UBRR11  = 3 ; USART Baud Rate Register bit 11

; UBRR0L - USART Baud Rate Register Low Byte
.equ    _UBRR0  = 0 ; USART Baud Rate Register bit 0
.equ    _UBRR1  = 1 ; USART Baud Rate Register bit 1
.equ    UBRR2   = 2 ; USART Baud Rate Register bit 2
.equ    UBRR3   = 3 ; USART Baud Rate Register bit 3
.equ    UBRR4   = 4 ; USART Baud Rate Register bit 4
.equ    UBRR5   = 5 ; USART Baud Rate Register bit 5
.equ    UBRR6   = 6 ; USART Baud Rate Register bit 6
.equ    UBRR7   = 7 ; USART Baud Rate Register bit 7

; ***** TWI *****
; TWAMR - TWI (Slave) Address Mask Register
.equ    TWAM0   = 1 ;
.equ    TWAMR0  = TWAM0 ; For compatibility
.equ    TWAM1   = 2 ;
.equ    TWAMR1  = TWAM1 ; For compatibility
.equ    TWAM2   = 3 ;
.equ    TWAMR2  = TWAM2 ; For compatibility

```

```
.equ TWAM3 = 4 ;
.equ TWAMR3 = TWAM3 ; For compatibility
.equ TWAM4 = 5 ;
.equ TWAMR4 = TWAM4 ; For compatibility
.equ TWAM5 = 6 ;
.equ TWAMR5 = TWAM5 ; For compatibility
.equ TWAM6 = 7 ;
.equ TWAMR6 = TWAM6 ; For compatibility

; TWBR - TWI Bit Rate register
.equ TWBR0 = 0 ;
.equ TWBR1 = 1 ;
.equ TWBR2 = 2 ;
.equ TWBR3 = 3 ;
.equ TWBR4 = 4 ;
.equ TWBR5 = 5 ;
.equ TWBR6 = 6 ;
.equ TWBR7 = 7 ;

; TWCR - TWI Control Register
.equ TWIE = 0 ; TWI Interrupt Enable
.equ TWEN = 2 ; TWI Enable Bit
.equ TWWC = 3 ; TWI Write Collision Flag
.equ TWSTO = 4 ; TWI Stop Condition Bit
.equ TWSTA = 5 ; TWI Start Condition Bit
.equ TWEA = 6 ; TWI Enable Acknowledge Bit
.equ TWINT = 7 ; TWI Interrupt Flag

; TWSR - TWI Status Register
.equ TWPS0 = 0 ; TWI Prescaler
.equ TWPS1 = 1 ; TWI Prescaler
.equ TWS3 = 3 ; TWI Status
.equ TWS4 = 4 ; TWI Status
.equ TWS5 = 5 ; TWI Status
.equ TWS6 = 6 ; TWI Status
.equ TWS7 = 7 ; TWI Status

; TWDR - TWI Data register
.equ TWD0 = 0 ; TWI Data Register Bit 0
.equ TWD1 = 1 ; TWI Data Register Bit 1
.equ TWD2 = 2 ; TWI Data Register Bit 2
.equ TWD3 = 3 ; TWI Data Register Bit 3
.equ TWD4 = 4 ; TWI Data Register Bit 4
.equ TWD5 = 5 ; TWI Data Register Bit 5
.equ TWD6 = 6 ; TWI Data Register Bit 6
.equ TWD7 = 7 ; TWI Data Register Bit 7

; TWAR - TWI (Slave) Address register
.equ TWGCE = 0 ; TWI General Call Recognition Enable Bit
.equ TWA0 = 1 ; TWI (Slave) Address register Bit 0
.equ TWA1 = 2 ; TWI (Slave) Address register Bit 1
.equ TWA2 = 3 ; TWI (Slave) Address register Bit 2
.equ TWA3 = 4 ; TWI (Slave) Address register Bit 3
.equ TWA4 = 5 ; TWI (Slave) Address register Bit 4
.equ TWA5 = 6 ; TWI (Slave) Address register Bit 5
```

```
.equ    TWA6      = 7 ; TWI (Slave) Address register Bit 6

; ***** TIMER_COUNTER_1 *****
; TIMSK1 - Timer/Counter Interrupt Mask Register
.equ    TOIE1     = 0 ; Timer/Counter1 Overflow Interrupt Enable
.equ    OCIE1A    = 1 ; Timer/Counter1 Output CompareA Match Interrupt Enable
.equ    OCIE1B    = 2 ; Timer/Counter1 Output CompareB Match Interrupt Enable
.equ    ICIE1     = 5 ; Timer/Counter1 Input Capture Interrupt Enable

; TIFR1 - Timer/Counter Interrupt Flag register
.equ    TOV1      = 0 ; Timer/Counter1 Overflow Flag
.equ    OCF1A     = 1 ; Output Compare Flag 1A
.equ    OCF1B     = 2 ; Output Compare Flag 1B
.equ    ICF1      = 5 ; Input Capture Flag 1

; TCCR1A - Timer/Counter1 Control Register A
.equ    WGM10     = 0 ; Waveform Generation Mode
.equ    WGM11     = 1 ; Waveform Generation Mode
.equ    COM1B0    = 4 ; Compare Output Mode 1B, bit 0
.equ    COM1B1    = 5 ; Compare Output Mode 1B, bit 1
.equ    COM1A0    = 6 ; Compare Output Mode 1A, bit 0
.equ    COM1A1    = 7 ; Compare Output Mode 1A, bit 1

; TCCR1B - Timer/Counter1 Control Register B
.equ    CS10      = 0 ; Prescaler source of Timer/Counter 1
.equ    CS11      = 1 ; Prescaler source of Timer/Counter 1
.equ    CS12      = 2 ; Prescaler source of Timer/Counter 1
.equ    WGM12     = 3 ; Waveform Generation Mode
.equ    WGM13     = 4 ; Waveform Generation Mode
.equ    ICES1     = 6 ; Input Capture 1 Edge Select
.equ    ICNC1     = 7 ; Input Capture 1 Noise Canceler

; TCCR1C - Timer/Counter1 Control Register C
.equ    FOC1B     = 6 ;
.equ    FOC1A     = 7 ;

; GTCCR - General Timer/Counter Control Register
.equ    PSRSYNC   = 0 ; Prescaler Reset Timer/Counter1 and Timer/Counter0
.equ    TSM       = 7 ; Timer/Counter Synchronization Mode

; ***** TIMER_COUNTER_2 *****
; TIMSK2 - Timer/Counter Interrupt Mask register
.equ    TOIE2     = 0 ; Timer/Counter2 Overflow Interrupt Enable
.equ    TOIE2A    = TOIE2 ; For compatibility
.equ    OCIE2A    = 1 ; Timer/Counter2 Output Compare Match A Interrupt Enable
.equ    OCIE2B    = 2 ; Timer/Counter2 Output Compare Match B Interrupt Enable

; TIFR2 - Timer/Counter Interrupt Flag Register
.equ    TOV2      = 0 ; Timer/Counter2 Overflow Flag
.equ    OCF2A     = 1 ; Output Compare Flag 2A
.equ    OCF2B     = 2 ; Output Compare Flag 2B

; TCCR2A - Timer/Counter2 Control Register A
```

```

.equ    WGM20    = 0 ; Waveform Genration Mode
.equ    WGM21    = 1 ; Waveform Genration Mode
.equ    COM2B0   = 4 ; Compare Output Mode bit 0
.equ    COM2B1   = 5 ; Compare Output Mode bit 1
.equ    COM2A0   = 6 ; Compare Output Mode bit 1
.equ    COM2A1   = 7 ; Compare Output Mode bit 1

; TCCR2B - Timer/Counter2 Control Register B
.equ    CS20     = 0 ; Clock Select bit 0
.equ    CS21     = 1 ; Clock Select bit 1
.equ    CS22     = 2 ; Clock Select bit 2
.equ    WGM22    = 3 ; Waveform Generation Mode
.equ    FOC2B    = 6 ; Force Output Compare B
.equ    FOC2A    = 7 ; Force Output Compare A

; TCNT2 - Timer/Counter2
.equ    TCNT2_0  = 0 ; Timer/Counter 2 bit 0
.equ    TCNT2_1  = 1 ; Timer/Counter 2 bit 1
.equ    TCNT2_2  = 2 ; Timer/Counter 2 bit 2
.equ    TCNT2_3  = 3 ; Timer/Counter 2 bit 3
.equ    TCNT2_4  = 4 ; Timer/Counter 2 bit 4
.equ    TCNT2_5  = 5 ; Timer/Counter 2 bit 5
.equ    TCNT2_6  = 6 ; Timer/Counter 2 bit 6
.equ    TCNT2_7  = 7 ; Timer/Counter 2 bit 7

; OCR2A - Timer/Counter2 Output Compare Register A
.equ    OCR2A_0  = 0 ; Timer/Counter2 Output Compare Register Bit 0
.equ    OCR2A_1  = 1 ; Timer/Counter2 Output Compare Register Bit 1
.equ    OCR2A_2  = 2 ; Timer/Counter2 Output Compare Register Bit 2
.equ    OCR2A_3  = 3 ; Timer/Counter2 Output Compare Register Bit 3
.equ    OCR2A_4  = 4 ; Timer/Counter2 Output Compare Register Bit 4
.equ    OCR2A_5  = 5 ; Timer/Counter2 Output Compare Register Bit 5
.equ    OCR2A_6  = 6 ; Timer/Counter2 Output Compare Register Bit 6
.equ    OCR2A_7  = 7 ; Timer/Counter2 Output Compare Register Bit 7

; OCR2B - Timer/Counter2 Output Compare Register B
.equ    OCR2B_0  = 0 ; Timer/Counter2 Output Compare Register Bit 0
.equ    OCR2B_1  = 1 ; Timer/Counter2 Output Compare Register Bit 1
.equ    OCR2B_2  = 2 ; Timer/Counter2 Output Compare Register Bit 2
.equ    OCR2B_3  = 3 ; Timer/Counter2 Output Compare Register Bit 3
.equ    OCR2B_4  = 4 ; Timer/Counter2 Output Compare Register Bit 4
.equ    OCR2B_5  = 5 ; Timer/Counter2 Output Compare Register Bit 5
.equ    OCR2B_6  = 6 ; Timer/Counter2 Output Compare Register Bit 6
.equ    OCR2B_7  = 7 ; Timer/Counter2 Output Compare Register Bit 7

; ASSR - Asynchronous Status Register
.equ    TCR2BUB  = 0 ; Timer/Counter Control Register2 Update Busy
.equ    TCR2AUB  = 1 ; Timer/Counter Control Register2 Update Busy
.equ    OCR2BUB  = 2 ; Output Compare Register 2 Update Busy
.equ    OCR2AUB  = 3 ; Output Compare Register2 Update Busy
.equ    TCN2UB   = 4 ; Timer/Counter2 Update Busy
.equ    AS2      = 5 ; Asynchronous Timer/Counter2
.equ    EXCLK    = 6 ; Enable External Clock Input

; GTCCR - General Timer Counter Control register

```

```

.equ    PSRASY    = 1 ; Prescaler Reset Timer/Counter2
.equ    PSR2      = PSRASY    ; For compatibility
;.equ    TSM      = 7 ; Timer/Counter Synchronization Mode

; ***** AD_CONVERTER *****
; ADMUX - The ADC multiplexer Selection Register
.equ    MUX0      = 0 ; Analog Channel and Gain Selection Bits
.equ    MUX1      = 1 ; Analog Channel and Gain Selection Bits
.equ    MUX2      = 2 ; Analog Channel and Gain Selection Bits
.equ    MUX3      = 3 ; Analog Channel and Gain Selection Bits
.equ    ADLAR     = 5 ; Left Adjust Result
.equ    REFS0     = 6 ; Reference Selection Bit 0
.equ    REFS1     = 7 ; Reference Selection Bit 1

; ADCSRA - The ADC Control and Status register A
.equ    ADPS0     = 0 ; ADC Prescaler Select Bits
.equ    ADPS1     = 1 ; ADC Prescaler Select Bits
.equ    ADPS2     = 2 ; ADC Prescaler Select Bits
.equ    ADIE      = 3 ; ADC Interrupt Enable
.equ    ADIF      = 4 ; ADC Interrupt Flag
.equ    ADATE     = 5 ; ADC Auto Trigger Enable
.equ    ADSC      = 6 ; ADC Start Conversion
.equ    ADEN      = 7 ; ADC Enable

; ADCSRB - The ADC Control and Status register B
.equ    ADTS0     = 0 ; ADC Auto Trigger Source bit 0
.equ    ADTS1     = 1 ; ADC Auto Trigger Source bit 1
.equ    ADTS2     = 2 ; ADC Auto Trigger Source bit 2
.equ    ACME      = 6 ;

; ADCH - ADC Data Register High Byte
.equ    ADCH0     = 0 ; ADC Data Register High Byte Bit 0
.equ    ADCH1     = 1 ; ADC Data Register High Byte Bit 1
.equ    ADCH2     = 2 ; ADC Data Register High Byte Bit 2
.equ    ADCH3     = 3 ; ADC Data Register High Byte Bit 3
.equ    ADCH4     = 4 ; ADC Data Register High Byte Bit 4
.equ    ADCH5     = 5 ; ADC Data Register High Byte Bit 5
.equ    ADCH6     = 6 ; ADC Data Register High Byte Bit 6
.equ    ADCH7     = 7 ; ADC Data Register High Byte Bit 7

; ADCL - ADC Data Register Low Byte
.equ    ADCL0     = 0 ; ADC Data Register Low Byte Bit 0
.equ    ADCL1     = 1 ; ADC Data Register Low Byte Bit 1
.equ    ADCL2     = 2 ; ADC Data Register Low Byte Bit 2
.equ    ADCL3     = 3 ; ADC Data Register Low Byte Bit 3
.equ    ADCL4     = 4 ; ADC Data Register Low Byte Bit 4
.equ    ADCL5     = 5 ; ADC Data Register Low Byte Bit 5
.equ    ADCL6     = 6 ; ADC Data Register Low Byte Bit 6
.equ    ADCL7     = 7 ; ADC Data Register Low Byte Bit 7

; DIDR0 - Digital Input Disable Register
.equ    ADC0D     = 0 ;
.equ    ADC1D     = 1 ;
.equ    ADC2D     = 2 ;

```



```

.equ    ADC3D    = 3 ;
.equ    ADC4D    = 4 ;
.equ    ADC5D    = 5 ;

; ***** ANALOG_COMPARATOR *****
; ACSR - Analog Comparator Control And Status Register
.equ    ACIS0    = 0 ; Analog Comparator Interrupt Mode Select bit 0
.equ    ACIS1    = 1 ; Analog Comparator Interrupt Mode Select bit 1
.equ    ACIC     = 2 ; Analog Comparator Input Capture Enable
.equ    ACIE     = 3 ; Analog Comparator Interrupt Enable
.equ    ACI      = 4 ; Analog Comparator Interrupt Flag
.equ    ACO      = 5 ; Analog Compare Output
.equ    ACBG     = 6 ; Analog Comparator Bandgap Select
.equ    ACD      = 7 ; Analog Comparator Disable

; DIDR1 - Digital Input Disable Register 1
.equ    AIN0D    = 0 ; AIN0 Digital Input Disable
.equ    AIN1D    = 1 ; AIN1 Digital Input Disable

; ***** PORTB *****
; PORTB - Port B Data Register
.equ    PORTB0   = 0 ; Port B Data Register bit 0
.equ    PB0      = 0 ; For compatibility
.equ    PORTB1   = 1 ; Port B Data Register bit 1
.equ    PB1      = 1 ; For compatibility
.equ    PORTB2   = 2 ; Port B Data Register bit 2
.equ    PB2      = 2 ; For compatibility
.equ    PORTB3   = 3 ; Port B Data Register bit 3
.equ    PB3      = 3 ; For compatibility
.equ    PORTB4   = 4 ; Port B Data Register bit 4
.equ    PB4      = 4 ; For compatibility
.equ    PORTB5   = 5 ; Port B Data Register bit 5
.equ    PB5      = 5 ; For compatibility
.equ    PORTB6   = 6 ; Port B Data Register bit 6
.equ    PB6      = 6 ; For compatibility
.equ    PORTB7   = 7 ; Port B Data Register bit 7
.equ    PB7      = 7 ; For compatibility

; DDRB - Port B Data Direction Register
.equ    DDB0     = 0 ; Port B Data Direction Register bit 0
.equ    DDB1     = 1 ; Port B Data Direction Register bit 1
.equ    DDB2     = 2 ; Port B Data Direction Register bit 2
.equ    DDB3     = 3 ; Port B Data Direction Register bit 3
.equ    DDB4     = 4 ; Port B Data Direction Register bit 4
.equ    DDB5     = 5 ; Port B Data Direction Register bit 5
.equ    DDB6     = 6 ; Port B Data Direction Register bit 6
.equ    DDB7     = 7 ; Port B Data Direction Register bit 7

; PINB - Port B Input Pins
.equ    PINB0    = 0 ; Port B Input Pins bit 0
.equ    PINB1    = 1 ; Port B Input Pins bit 1
.equ    PINB2    = 2 ; Port B Input Pins bit 2
.equ    PINB3    = 3 ; Port B Input Pins bit 3

```

```
.equ    PINB4    = 4 ; Port B Input Pins bit 4
.equ    PINB5    = 5 ; Port B Input Pins bit 5
.equ    PINB6    = 6 ; Port B Input Pins bit 6
.equ    PINB7    = 7 ; Port B Input Pins bit 7

; ***** PORTC *****
; PORTC - Port C Data Register
.equ    PORTC0   = 0 ; Port C Data Register bit 0
.equ    PC0      = 0 ; For compatibility
.equ    PORTC1   = 1 ; Port C Data Register bit 1
.equ    PC1      = 1 ; For compatibility
.equ    PORTC2   = 2 ; Port C Data Register bit 2
.equ    PC2      = 2 ; For compatibility
.equ    PORTC3   = 3 ; Port C Data Register bit 3
.equ    PC3      = 3 ; For compatibility
.equ    PORTC4   = 4 ; Port C Data Register bit 4
.equ    PC4      = 4 ; For compatibility
.equ    PORTC5   = 5 ; Port C Data Register bit 5
.equ    PC5      = 5 ; For compatibility
.equ    PORTC6   = 6 ; Port C Data Register bit 6
.equ    PC6      = 6 ; For compatibility

; DDRC - Port C Data Direction Register
.equ    DDC0     = 0 ; Port C Data Direction Register bit 0
.equ    DDC1     = 1 ; Port C Data Direction Register bit 1
.equ    DDC2     = 2 ; Port C Data Direction Register bit 2
.equ    DDC3     = 3 ; Port C Data Direction Register bit 3
.equ    DDC4     = 4 ; Port C Data Direction Register bit 4
.equ    DDC5     = 5 ; Port C Data Direction Register bit 5
.equ    DDC6     = 6 ; Port C Data Direction Register bit 6

; PINC - Port C Input Pins
.equ    PINC0    = 0 ; Port C Input Pins bit 0
.equ    PINC1    = 1 ; Port C Input Pins bit 1
.equ    PINC2    = 2 ; Port C Input Pins bit 2
.equ    PINC3    = 3 ; Port C Input Pins bit 3
.equ    PINC4    = 4 ; Port C Input Pins bit 4
.equ    PINC5    = 5 ; Port C Input Pins bit 5
.equ    PINC6    = 6 ; Port C Input Pins bit 6

; ***** PORTD *****
; PORTD - Port D Data Register
.equ    PORTD0   = 0 ; Port D Data Register bit 0
.equ    PD0      = 0 ; For compatibility
.equ    PORTD1   = 1 ; Port D Data Register bit 1
.equ    PD1      = 1 ; For compatibility
.equ    PORTD2   = 2 ; Port D Data Register bit 2
.equ    PD2      = 2 ; For compatibility
.equ    PORTD3   = 3 ; Port D Data Register bit 3
.equ    PD3      = 3 ; For compatibility
.equ    PORTD4   = 4 ; Port D Data Register bit 4
.equ    PD4      = 4 ; For compatibility
.equ    PORTD5   = 5 ; Port D Data Register bit 5
```

```
.equ PD5 = 5 ; For compatibility
.equ PORTD6 = 6 ; Port D Data Register bit 6
.equ PD6 = 6 ; For compatibility
.equ PORTD7 = 7 ; Port D Data Register bit 7
.equ PD7 = 7 ; For compatibility

; DDRD - Port D Data Direction Register
.equ DDD0 = 0 ; Port D Data Direction Register bit 0
.equ DDD1 = 1 ; Port D Data Direction Register bit 1
.equ DDD2 = 2 ; Port D Data Direction Register bit 2
.equ DDD3 = 3 ; Port D Data Direction Register bit 3
.equ DDD4 = 4 ; Port D Data Direction Register bit 4
.equ DDD5 = 5 ; Port D Data Direction Register bit 5
.equ DDD6 = 6 ; Port D Data Direction Register bit 6
.equ DDD7 = 7 ; Port D Data Direction Register bit 7

; PIND - Port D Input Pins
.equ PIND0 = 0 ; Port D Input Pins bit 0
.equ PIND1 = 1 ; Port D Input Pins bit 1
.equ PIND2 = 2 ; Port D Input Pins bit 2
.equ PIND3 = 3 ; Port D Input Pins bit 3
.equ PIND4 = 4 ; Port D Input Pins bit 4
.equ PIND5 = 5 ; Port D Input Pins bit 5
.equ PIND6 = 6 ; Port D Input Pins bit 6
.equ PIND7 = 7 ; Port D Input Pins bit 7

; ***** TIMER_COUNTER_0 *****
; TIMSK0 - Timer/Counter0 Interrupt Mask Register
.equ TOIE0 = 0 ; Timer/Counter0 Overflow Interrupt Enable
.equ OCIE0A = 1 ; Timer/Counter0 Output Compare Match A Interrupt Enable
.equ OCIE0B = 2 ; Timer/Counter0 Output Compare Match B Interrupt Enable

; TIFR0 - Timer/Counter0 Interrupt Flag register
.equ TOV0 = 0 ; Timer/Counter0 Overflow Flag
.equ OCF0A = 1 ; Timer/Counter0 Output Compare Flag 0A
.equ OCF0B = 2 ; Timer/Counter0 Output Compare Flag 0B

; TCCR0A - Timer/Counter Control Register A
.equ WGM00 = 0 ; Waveform Generation Mode
.equ WGM01 = 1 ; Waveform Generation Mode
.equ COM0B0 = 4 ; Compare Output Mode, Fast PwM
.equ COM0B1 = 5 ; Compare Output Mode, Fast PwM
.equ COM0A0 = 6 ; Compare Output Mode, Phase Correct PWM Mode
.equ COM0A1 = 7 ; Compare Output Mode, Phase Correct PWM Mode

; TCCR0B - Timer/Counter Control Register B
.equ CS00 = 0 ; Clock Select
.equ CS01 = 1 ; Clock Select
.equ CS02 = 2 ; Clock Select
.equ WGM02 = 3 ;
.equ FOC0B = 6 ; Force Output Compare B
.equ FOC0A = 7 ; Force Output Compare A

; TCNT0 - Timer/Counter0
```

```

.equ    TCNT0_0 = 0 ;
.equ    TCNT0_1 = 1 ;
.equ    TCNT0_2 = 2 ;
.equ    TCNT0_3 = 3 ;
.equ    TCNT0_4 = 4 ;
.equ    TCNT0_5 = 5 ;
.equ    TCNT0_6 = 6 ;
.equ    TCNT0_7 = 7 ;

; OCR0A - Timer/Counter0 Output Compare Register
.equ    OCR0A_0 = 0 ;
.equ    OCR0A_1 = 1 ;
.equ    OCR0A_2 = 2 ;
.equ    OCR0A_3 = 3 ;
.equ    OCR0A_4 = 4 ;
.equ    OCR0A_5 = 5 ;
.equ    OCR0A_6 = 6 ;
.equ    OCR0A_7 = 7 ;

; OCR0B - Timer/Counter0 Output Compare Register
.equ    OCR0B_0 = 0 ;
.equ    OCR0B_1 = 1 ;
.equ    OCR0B_2 = 2 ;
.equ    OCR0B_3 = 3 ;
.equ    OCR0B_4 = 4 ;
.equ    OCR0B_5 = 5 ;
.equ    OCR0B_6 = 6 ;
.equ    OCR0B_7 = 7 ;

; GTCCR - General Timer/Counter Control Register
;.equ    PSRSYNC = 0 ; Prescaler Reset Timer/Counter1 and Timer/Counter0
.equ    PSR10    = PSRSYNC    ; For compatibility
;.equ    TSM = 7 ; Timer/Counter Synchronization Mode

; ***** EXTERNAL_INTERRUPT *****
; EICRA - External Interrupt Control Register
.equ    ISC00    = 0 ; External Interrupt Sense Control 0 Bit 0
.equ    ISC01    = 1 ; External Interrupt Sense Control 0 Bit 1
.equ    ISC10    = 2 ; External Interrupt Sense Control 1 Bit 0
.equ    ISC11    = 3 ; External Interrupt Sense Control 1 Bit 1

; EIMSK - External Interrupt Mask Register
.equ    INT0     = 0 ; External Interrupt Request 0 Enable
.equ    INT1     = 1 ; External Interrupt Request 1 Enable

; EIFR - External Interrupt Flag Register
.equ    INTF0    = 0 ; External Interrupt Flag 0
.equ    INTF1    = 1 ; External Interrupt Flag 1

; PCICR - Pin Change Interrupt Control Register
.equ    PCIE0    = 0 ; Pin Change Interrupt Enable 0
.equ    PCIE1    = 1 ; Pin Change Interrupt Enable 1
.equ    PCIE2    = 2 ; Pin Change Interrupt Enable 2

```

```

; PCMSK2 - Pin Change Mask Register 2
.equ    PCINT16 = 0 ; Pin Change Enable Mask 16
.equ    PCINT17 = 1 ; Pin Change Enable Mask 17
.equ    PCINT18 = 2 ; Pin Change Enable Mask 18
.equ    PCINT19 = 3 ; Pin Change Enable Mask 19
.equ    PCINT20 = 4 ; Pin Change Enable Mask 20
.equ    PCINT21 = 5 ; Pin Change Enable Mask 21
.equ    PCINT22 = 6 ; Pin Change Enable Mask 22
.equ    PCINT23 = 7 ; Pin Change Enable Mask 23

; PCMSK1 - Pin Change Mask Register 1
.equ    PCINT8  = 0 ; Pin Change Enable Mask 8
.equ    PCINT9  = 1 ; Pin Change Enable Mask 9
.equ    PCINT10 = 2 ; Pin Change Enable Mask 10
.equ    PCINT11 = 3 ; Pin Change Enable Mask 11
.equ    PCINT12 = 4 ; Pin Change Enable Mask 12
.equ    PCINT13 = 5 ; Pin Change Enable Mask 13
.equ    PCINT14 = 6 ; Pin Change Enable Mask 14

; PCMSK0 - Pin Change Mask Register 0
.equ    PCINT0  = 0 ; Pin Change Enable Mask 0
.equ    PCINT1  = 1 ; Pin Change Enable Mask 1
.equ    PCINT2  = 2 ; Pin Change Enable Mask 2
.equ    PCINT3  = 3 ; Pin Change Enable Mask 3
.equ    PCINT4  = 4 ; Pin Change Enable Mask 4
.equ    PCINT5  = 5 ; Pin Change Enable Mask 5
.equ    PCINT6  = 6 ; Pin Change Enable Mask 6
.equ    PCINT7  = 7 ; Pin Change Enable Mask 7

; PCIFR - Pin Change Interrupt Flag Register
.equ    PCIF0   = 0 ; Pin Change Interrupt Flag 0
.equ    PCIF1   = 1 ; Pin Change Interrupt Flag 1
.equ    PCIF2   = 2 ; Pin Change Interrupt Flag 2

; ***** SPI *****
; SPDR - SPI Data Register
.equ    SPDR0   = 0 ; SPI Data Register bit 0
.equ    SPDR1   = 1 ; SPI Data Register bit 1
.equ    SPDR2   = 2 ; SPI Data Register bit 2
.equ    SPDR3   = 3 ; SPI Data Register bit 3
.equ    SPDR4   = 4 ; SPI Data Register bit 4
.equ    SPDR5   = 5 ; SPI Data Register bit 5
.equ    SPDR6   = 6 ; SPI Data Register bit 6
.equ    SPDR7   = 7 ; SPI Data Register bit 7

; SPSR - SPI Status Register
.equ    SPI2X   = 0 ; Double SPI Speed Bit
.equ    WCOL    = 6 ; Write Collision Flag
.equ    SPIF    = 7 ; SPI Interrupt Flag

; SPCR - SPI Control Register
.equ    SPR0    = 0 ; SPI Clock Rate Select 0
.equ    SPR1    = 1 ; SPI Clock Rate Select 1
.equ    CPHA    = 2 ; Clock Phase

```

```

.equ    CPOL      = 3 ; Clock polarity
.equ    MSTR      = 4 ; Master/Slave Select
.equ    DORD      = 5 ; Data Order
.equ    SPE = 6 ; SPI Enable
.equ    SPIE      = 7 ; SPI Interrupt Enable

; ***** WATCHDOG *****
; WDTCSR - Watchdog Timer Control Register
.equ    WDP0      = 0 ; Watch Dog Timer Prescaler bit 0
.equ    WDP1      = 1 ; Watch Dog Timer Prescaler bit 1
.equ    WDP2      = 2 ; Watch Dog Timer Prescaler bit 2
.equ    WDE = 3 ; Watch Dog Enable
.equ    WDCE      = 4 ; Watchdog Change Enable
.equ    WDP3      = 5 ; Watchdog Timer Prescaler Bit 3
.equ    WDIE      = 6 ; Watchdog Timeout Interrupt Enable
.equ    WDIF      = 7 ; Watchdog Timeout Interrupt Flag

; ***** CPU *****
; SREG - Status Register
.equ    SREG_C    = 0 ; Carry Flag
.equ    SREG_Z    = 1 ; Zero Flag
.equ    SREG_N    = 2 ; Negative Flag
.equ    SREG_V    = 3 ; Two's Complement Overflow Flag
.equ    SREG_S    = 4 ; Sign Bit
.equ    SREG_H    = 5 ; Half Carry Flag
.equ    SREG_T    = 6 ; Bit Copy Storage
.equ    SREG_I    = 7 ; Global Interrupt Enable

; OSCCAL - Oscillator Calibration Value
.equ    CAL0      = 0 ; Oscillator Calibration Value Bit0
.equ    CAL1      = 1 ; Oscillator Calibration Value Bit1
.equ    CAL2      = 2 ; Oscillator Calibration Value Bit2
.equ    CAL3      = 3 ; Oscillator Calibration Value Bit3
.equ    CAL4      = 4 ; Oscillator Calibration Value Bit4
.equ    CAL5      = 5 ; Oscillator Calibration Value Bit5
.equ    CAL6      = 6 ; Oscillator Calibration Value Bit6
.equ    CAL7      = 7 ; Oscillator Calibration Value Bit7

; CLKPR - Clock Prescale Register
.equ    CLKPS0    = 0 ; Clock Prescaler Select Bit 0
.equ    CLKPS1    = 1 ; Clock Prescaler Select Bit 1
.equ    CLKPS2    = 2 ; Clock Prescaler Select Bit 2
.equ    CLKPS3    = 3 ; Clock Prescaler Select Bit 3
.equ    CLKPCE    = 7 ; Clock Prescaler Change Enable

; SPMCSR - Store Program Memory Control and Status Register
.equ    SELFPRGEN = 0 ; Self Programming Enable
.equ    PGERS     = 1 ; Page Erase
.equ    PGWRT     = 2 ; Page Write
.equ    BLBSET    = 3 ; Boot Lock Bit Set
.equ    RWWSRE    = 4 ; Read-While-Write section read enable
.equ    RWWSB     = 6 ; Read-While-Write Section Busy
.equ    SPMIE     = 7 ; SPM Interrupt Enable

```

```
; MCUCR - MCU Control Register
.equ    IVCE    = 0 ;
.equ    IVSEL   = 1 ;
.equ    PUD    = 4 ;
.equ    BODSE   = 5 ; BOD Sleep Enable
.equ    BODS    = 6 ; BOD Sleep

; MCUSR - MCU Status Register
.equ    PORF    = 0 ; Power-on reset flag
.equ    EXTRF   = 1 ; External Reset Flag
.equ    EXTREF  = EXTRF ; For compatibility
.equ    BORF    = 2 ; Brown-out Reset Flag
.equ    WDRF    = 3 ; Watchdog Reset Flag

; SMCR - Sleep Mode Control Register
.equ    SE      = 0 ; Sleep Enable
.equ    SM0     = 1 ; Sleep Mode Select Bit 0
.equ    SM1     = 2 ; Sleep Mode Select Bit 1
.equ    SM2     = 3 ; Sleep Mode Select Bit 2

; GPIOR2 - General Purpose I/O Register 2
.equ    GPIOR20 = 0 ;
.equ    GPIOR21 = 1 ;
.equ    GPIOR22 = 2 ;
.equ    GPIOR23 = 3 ;
.equ    GPIOR24 = 4 ;
.equ    GPIOR25 = 5 ;
.equ    GPIOR26 = 6 ;
.equ    GPIOR27 = 7 ;

; GPIOR1 - General Purpose I/O Register 1
.equ    GPIOR10 = 0 ;
.equ    GPIOR11 = 1 ;
.equ    GPIOR12 = 2 ;
.equ    GPIOR13 = 3 ;
.equ    GPIOR14 = 4 ;
.equ    GPIOR15 = 5 ;
.equ    GPIOR16 = 6 ;
.equ    GPIOR17 = 7 ;

; GPIOR0 - General Purpose I/O Register 0
.equ    GPIOR00 = 0 ;
.equ    GPIOR01 = 1 ;
.equ    GPIOR02 = 2 ;
.equ    GPIOR03 = 3 ;
.equ    GPIOR04 = 4 ;
.equ    GPIOR05 = 5 ;
.equ    GPIOR06 = 6 ;
.equ    GPIOR07 = 7 ;

; PRR - Power Reduction Register
.equ    PRADC   = 0 ; Power Reduction ADC
.equ    PRUSART0 = 1 ; Power Reduction USART
.equ    PRSPI   = 2 ; Power Reduction Serial Peripheral Interface
```

```

.equ    PRTIM1    = 3 ; Power Reduction Timer/Counter1
.equ    PRTIM0    = 5 ; Power Reduction Timer/Counter0
.equ    PRTIM2    = 6 ; Power Reduction Timer/Counter2
.equ    PRTWI     = 7 ; Power Reduction TWI

; ***** EEPROM *****
; EEARL - EEPROM Address Register Low Byte
.equ    EEAR0     = 0 ; EEPROM Read/Write Access Bit 0
.equ    EEAR1     = 1 ; EEPROM Read/Write Access Bit 1
.equ    EEAR2     = 2 ; EEPROM Read/Write Access Bit 2
.equ    EEAR3     = 3 ; EEPROM Read/Write Access Bit 3
.equ    EEAR4     = 4 ; EEPROM Read/Write Access Bit 4
.equ    EEAR5     = 5 ; EEPROM Read/Write Access Bit 5
.equ    EEAR6     = 6 ; EEPROM Read/Write Access Bit 6
.equ    EEAR7     = 7 ; EEPROM Read/Write Access Bit 7

; EEARH - EEPROM Address Register High Byte
.equ    EEAR8     = 0 ; EEPROM Read/Write Access Bit 8
.equ    EEAR9     = 1 ; EEPROM Read/Write Access Bit 9

; EEDR - EEPROM Data Register
.equ    EEDR0     = 0 ; EEPROM Data Register bit 0
.equ    EEDR1     = 1 ; EEPROM Data Register bit 1
.equ    EEDR2     = 2 ; EEPROM Data Register bit 2
.equ    EEDR3     = 3 ; EEPROM Data Register bit 3
.equ    EEDR4     = 4 ; EEPROM Data Register bit 4
.equ    EEDR5     = 5 ; EEPROM Data Register bit 5
.equ    EEDR6     = 6 ; EEPROM Data Register bit 6
.equ    EEDR7     = 7 ; EEPROM Data Register bit 7

; EECR - EEPROM Control Register
.equ    EERE      = 0 ; EEPROM Read Enable
.equ    EEPE      = 1 ; EEPROM Write Enable
.equ    EEMPE     = 2 ; EEPROM Master Write Enable
.equ    EERIE     = 3 ; EEPROM Ready Interrupt Enable
.equ    EEPM0     = 4 ; EEPROM Programming Mode Bit 0
.equ    EEPM1     = 5 ; EEPROM Programming Mode Bit 1

; ***** LOCKSBITS *****
.equ    LB1       = 0 ; Lock bit
.equ    LB2       = 1 ; Lock bit
.equ    BLB01     = 2 ; Boot Lock bit
.equ    BLB02     = 3 ; Boot Lock bit
.equ    BLB11     = 4 ; Boot lock bit
.equ    BLB12     = 5 ; Boot lock bit

; ***** FUSES *****
; LOW fuse bits
.equ    CKSEL0    = 0 ; Select Clock Source
.equ    CKSEL1    = 1 ; Select Clock Source
.equ    CKSEL2    = 2 ; Select Clock Source

```



```

.equ    CKSEL3    = 3 ; Select Clock Source
.equ    SUT0      = 4 ; Select start-up time
.equ    SUT1      = 5 ; Select start-up time
.equ    CKOUT     = 6 ; Clock output
.equ    CKDIV8    = 7 ; Divide clock by 8

; HIGH fuse bits
.equ    BOTRST   = 0 ; Select reset vector
.equ    BOOTSZ0  = 1 ; Select boot size
.equ    BOOTSZ1  = 2 ; Select boot size
.equ    EESAVE   = 3 ; EEPROM memory is preserved through chip erase
.equ    WDTON    = 4 ; Watchdog Timer Always On
.equ    SPIEN    = 5 ; Enable Serial programming and Data Downloading
.equ    DWEN     = 6 ; debugWIRE Enable
.equ    RSTDISBL = 7 ; External reset disable

; EXTENDED fuse bits
.equ    BODLEVEL0 = 0 ; Brown-out Detector trigger level
.equ    BODLEVEL1 = 1 ; Brown-out Detector trigger level
.equ    BODLEVEL2 = 2 ; Brown-out Detector trigger level

; ***** CPU REGISTER DEFINITIONS *****
.def    XH    = r27
.def    XL    = r26
.def    YH    = r29
.def    YL    = r28
.def    ZH    = r31
.def    ZL    = r30

; ***** DATA MEMORY DECLARATIONS *****
.equ    FLASHEND    = 0x3fff ; Note: Word address
.equ    IOEND       = 0x00ff
.equ    SRAM_START  = 0x0100
.equ    SRAM_SIZE   = 2048
.equ    RAMEND      = 0x08ff
.equ    XRAMEND     = 0x0000
.equ    E2END       = 0x03ff
.equ    EEPROMEND   = 0x03ff
.equ    EEADRBITS   = 10
#pragma AVRPART MEMORY PROG_FLASH 32768
#pragma AVRPART MEMORY EEPROM 1024
#pragma AVRPART MEMORY INT_SRAM SIZE 2048
#pragma AVRPART MEMORY INT_SRAM START_ADDR 0x100

; ***** BOOTLOADER DECLARATIONS *****
.equ    NRWW_START_ADDR = 0x3800
.equ    NRWW_STOP_ADDR  = 0x3fff
.equ    RWW_START_ADDR  = 0x0
.equ    RWW_STOP_ADDR   = 0x37ff

```

```

.equ    PAGESIZE      = 64
.equ    FIRSTBOOTSTART = 0x3f00
.equ    SECONDBOOTSTART = 0x3e00
.equ    THIRDBOOTSTART = 0x3c00
.equ    FOURTHBOOTSTART = 0x3800
.equ    SMALLBOOTSTART = FIRSTBOOTSTART
.equ    LARGEBOOTSTART = FOURTHBOOTSTART

; ***** INTERRUPT VECTORS *****
.equ    INT0addr      = 0x0002    ; External Interrupt Request 0
.equ    INT1addr      = 0x0004    ; External Interrupt Request 1
.equ    PCI0addr      = 0x0006    ; Pin Change Interrupt Request 0
.equ    PCI1addr      = 0x0008    ; Pin Change Interrupt Request 0
.equ    PCI2addr      = 0x000a    ; Pin Change Interrupt Request 1
.equ    WDTaddr       = 0x000c    ; Watchdog Time-out Interrupt
.equ    OC2Aaddr      = 0x000e    ; Timer/Counter2 Compare Match A
.equ    OC2Baddr      = 0x0010    ; Timer/Counter2 Compare Match A
.equ    OVF2addr      = 0x0012    ; Timer/Counter2 Overflow
.equ    ICPL1addr     = 0x0014    ; Timer/Counter1 Capture Event
.equ    OC1Aaddr      = 0x0016    ; Timer/Counter1 Compare Match A
.equ    OC1Baddr      = 0x0018    ; Timer/Counter1 Compare Match B
.equ    OVF1addr      = 0x001a    ; Timer/Counter1 Overflow
.equ    OC0Aaddr      = 0x001c    ; TimerCounter0 Compare Match A
.equ    OC0Baddr      = 0x001e    ; TimerCounter0 Compare Match B
.equ    OVF0addr      = 0x0020    ; Timer/Couner0 Overflow
.equ    SPIaddr       = 0x0022    ; SPI Serial Transfer Complete
.equ    URXCaddr      = 0x0024    ; USART Rx Complete
.equ    UDREaddr      = 0x0026    ; USART, Data Register Empty
.equ    UTXCaddr      = 0x0028    ; USART Tx Complete
.equ    ADCCaddr      = 0x002a    ; ADC Conversion Complete
.equ    ERDYaddr      = 0x002c    ; EEPROM Ready
.equ    ACIaddr       = 0x002e    ; Analog Comparator
.equ    TWIaddr       = 0x0030    ; Two-wire Serial Interface
.equ    SPMRaddr      = 0x0032    ; Store Program Memory Read

.equ    INT_VECTORS_SIZE = 52    ; size in words

#endif /* _M328PDEF_INC_ */

; ***** END OF FILE *****

```