

```

;***** THIS IS A MACHINE GENERATED FILE - DO NOT EDIT ****
;***** Created: 2011-02-09 12:03 ***** Source: ATmega2560.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      : "m2560def.inc"
;* Title          : Register/Bit Definitions for the ATmega2560
;* Date           : 2011-02-09
;* Version         : 2.35
;* Support E-mail : avr@atmel.com
;* Target MCU     : ATmega2560
;*
;* 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 _M2560DEF_INC_
#define _M2560DEF_INC_


#pragma partinc 0

; ***** SPECIFY DEVICE ****
.device ATmega2560
#pragma AVRPART ADMIN PART_NAME ATmega2560
.equ SIGNATURE_000 = 0x1e
.equ SIGNATURE_001 = 0x98
.equ SIGNATURE_002 = 0x01

#pragma AVRPART CORE CORE_VERSION V3

; ***** I/O REGISTER DEFINITIONS ****
; NOTE:
; Definitions marked "MEMORY MAPPED"are extended I/O ports

```

```
; and cannot be used with IN/OUT instructions
.equ UDR3      = 0x136      ; MEMORY MAPPED
.equ UBRR3L    = 0x134      ; MEMORY MAPPED
.equ UBRR3H    = 0x135      ; MEMORY MAPPED
.equ UCSR3C    = 0x132      ; MEMORY MAPPED
.equ UCSR3B    = 0x131      ; MEMORY MAPPED
.equ UCSR3A    = 0x130      ; MEMORY MAPPED
.equ OCR5CL   = 0x12c      ; MEMORY MAPPED
.equ OCR5CH   = 0x12d      ; MEMORY MAPPED
.equ OCR5BL   = 0x12a      ; MEMORY MAPPED
.equ OCR5BH   = 0x12b      ; MEMORY MAPPED
.equ OCR5AL   = 0x128      ; MEMORY MAPPED
.equ OCR5AH   = 0x129      ; MEMORY MAPPED
.equ ICR5H    = 0x127      ; MEMORY MAPPED
.equ ICR5L    = 0x126      ; MEMORY MAPPED
.equ TCNT5L   = 0x124      ; MEMORY MAPPED
.equ TCNT5H   = 0x125      ; MEMORY MAPPED
.equ TCCR5C   = 0x122      ; MEMORY MAPPED
.equ TCCR5B   = 0x121      ; MEMORY MAPPED
.equ TCCR5A   = 0x120      ; MEMORY MAPPED
.equ PORTL    = 0x10b      ; MEMORY MAPPED
.equ DDRL     = 0x10a      ; MEMORY MAPPED
.equ PINL     = 0x109      ; MEMORY MAPPED
.equ PORTK    = 0x108      ; MEMORY MAPPED
.equ DDRK     = 0x107      ; MEMORY MAPPED
.equ PINK     = 0x106      ; MEMORY MAPPED
.equ PORTJ    = 0x105      ; MEMORY MAPPED
.equ DDRJ     = 0x104      ; MEMORY MAPPED
.equ PINJ     = 0x103      ; MEMORY MAPPED
.equ PORTH    = 0x102      ; MEMORY MAPPED
.equ DDRH     = 0x101      ; MEMORY MAPPED
.equ PINH     = 0x100      ; MEMORY MAPPED
.equ UDR2     = 0xd6 ; MEMORY MAPPED
.equ UBRR2L   = 0xd4 ; MEMORY MAPPED
.equ UBRR2H   = 0xd5 ; MEMORY MAPPED
.equ UCSR2C   = 0xd2 ; MEMORY MAPPED
.equ UCSR2B   = 0xd1 ; MEMORY MAPPED
.equ UCSR2A   = 0xd0 ; MEMORY MAPPED
.equ UDR1     = 0xce ; MEMORY MAPPED
.equ UBRR1L   = 0xcc ; MEMORY MAPPED
.equ UBRR1H   = 0xcd ; MEMORY MAPPED
.equ UCSR1C   = 0xca ; MEMORY MAPPED
.equ UCSR1B   = 0xc9 ; MEMORY MAPPED
.equ UCSR1A   = 0xc8 ; MEMORY MAPPED
.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 OCR4CL = 0xac ; MEMORY MAPPED
.equ OCR4CH = 0xad ; MEMORY MAPPED
.equ OCR4BL = 0xaa ; MEMORY MAPPED
.equ OCR4BH = 0xab ; MEMORY MAPPED
.equ OCR4AL = 0xa8 ; MEMORY MAPPED
.equ OCR4AH = 0xa9 ; MEMORY MAPPED
.equ ICR4L = 0xa6 ; MEMORY MAPPED
.equ ICR4H = 0xa7 ; MEMORY MAPPED
.equ TCNT4L = 0xa4 ; MEMORY MAPPED
.equ TCNT4H = 0xa5 ; MEMORY MAPPED
.equ TCCR4C = 0xa2 ; MEMORY MAPPED
.equ TCCR4B = 0xa1 ; MEMORY MAPPED
.equ TCCR4A = 0xa0 ; MEMORY MAPPED
.equ OCR3CL = 0x9c ; MEMORY MAPPED
.equ OCR3CH = 0x9d ; MEMORY MAPPED
.equ OCR3BL = 0x9a ; MEMORY MAPPED
.equ OCR3BH = 0x9b ; MEMORY MAPPED
.equ OCR3AL = 0x98 ; MEMORY MAPPED
.equ OCR3AH = 0x99 ; MEMORY MAPPED
.equ ICR3L = 0x96 ; MEMORY MAPPED
.equ ICR3H = 0x97 ; MEMORY MAPPED
.equ TCNT3L = 0x94 ; MEMORY MAPPED
.equ TCNT3H = 0x95 ; MEMORY MAPPED
.equ TCCR3C = 0x92 ; MEMORY MAPPED
.equ TCCR3B = 0x91 ; MEMORY MAPPED
.equ TCCR3A = 0x90 ; MEMORY MAPPED
.equ OCR1CL = 0x8c ; MEMORY MAPPED
.equ OCR1CH = 0x8d ; 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 DIDR2 = 0x7d ; 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 XMCRB = 0x75 ; MEMORY MAPPED
.equ XMCRA = 0x74 ; MEMORY MAPPED
.equ TIMSK5 = 0x73 ; MEMORY MAPPED
.equ TIMSK4 = 0x72 ; MEMORY MAPPED
.equ TIMSK3 = 0x71 ; MEMORY MAPPED
.equ TIMSK2 = 0x70 ; MEMORY MAPPED
.equ TIMSK1 = 0x6f ; MEMORY MAPPED
```

```
.equ  TIMSK0 = 0x6e ; MEMORY MAPPED
.equ  PCMSK2 = 0x6d ; MEMORY MAPPED
.equ  PCMSK1 = 0x6c ; MEMORY MAPPED
.equ  PCMSK0 = 0x6b ; MEMORY MAPPED
.equ  EICRB = 0x6a ; MEMORY MAPPED
.equ  EICRA = 0x69 ; MEMORY MAPPED
.equ  PCICR = 0x68 ; MEMORY MAPPED
.equ  OSCCAL = 0x66 ; MEMORY MAPPED
.equ  PRR1 = 0x65 ; MEMORY MAPPED
.equ  PRR0 = 0x64 ; MEMORY MAPPED
.equ  CLKPR = 0x61 ; MEMORY MAPPED
.equ  WDTCSR = 0x60 ; MEMORY MAPPED
.equ  SREG = 0x3f
.equ  SPL = 0x3d
.equ  SPH = 0x3e
.equ  EIND = 0x3c
.equ  RAMPZ = 0x3b
.equ  SPMCSR = 0x37
.equ  MCUCR = 0x35
.equ  MCUSR = 0x34
.equ  SMCR = 0x33
.equ  OCDR = 0x31
.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  TIFR5 = 0x1a
.equ  TIFR4 = 0x19
.equ  TIFR3 = 0x18
.equ  TIFR2 = 0x17
.equ  TIFR1 = 0x16
.equ  TIFR0 = 0x15
.equ  PORTG = 0x14
.equ  DDRG = 0x13
.equ  PING = 0x12
.equ  PORTF = 0x11
.equ  DDRF = 0x10
.equ  PINF = 0x0f
.equ  PORTE = 0x0e
.equ  DDRE = 0x0d
.equ  PINE = 0x0c
.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
.equ PORTA = 0x02
.equ DDRA = 0x01
.equ PINA = 0x00

; ***** BIT DEFINITIONS *****
; ***** ANALOG_COMPARATOR *****
; ADCSRB - ADC Control and Status Register B
.equ ACME = 6 ; Analog Comparator Multiplexer Enable

; 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

; ***** 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  
  

; ***** 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 SPRO = 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

; ***** PORTA *****
; PORTA - Port A Data Register
.equ PORTA0 = 0 ; Port A Data Register bit 0
.equ PA0 = 0 ; For compatibility
.equ PORTA1 = 1 ; Port A Data Register bit 1
.equ PA1 = 1 ; For compatibility
.equ PORTA2 = 2 ; Port A Data Register bit 2
.equ PA2 = 2 ; For compatibility
.equ PORTA3 = 3 ; Port A Data Register bit 3
.equ PA3 = 3 ; For compatibility
.equ PORTA4 = 4 ; Port A Data Register bit 4
.equ PA4 = 4 ; For compatibility
.equ PORTA5 = 5 ; Port A Data Register bit 5
.equ PA5 = 5 ; For compatibility
.equ PORTA6 = 6 ; Port A Data Register bit 6
.equ PA6 = 6 ; For compatibility
.equ PORTA7 = 7 ; Port A Data Register bit 7
.equ PA7 = 7 ; For compatibility

; DDRA - Port A Data Direction Register
.equ DDA0 = 0 ; Data Direction Register, Port A, bit 0
.equ DDA1 = 1 ; Data Direction Register, Port A, bit 1
.equ DDA2 = 2 ; Data Direction Register, Port A, bit 2
.equ DDA3 = 3 ; Data Direction Register, Port A, bit 3
.equ DDA4 = 4 ; Data Direction Register, Port A, bit 4
.equ DDA5 = 5 ; Data Direction Register, Port A, bit 5
.equ DDA6 = 6 ; Data Direction Register, Port A, bit 6
.equ DDA7 = 7 ; Data Direction Register, Port A, bit 7

; PINA - Port A Input Pins
.equ PINA0 = 0 ; Input Pins, Port A bit 0
.equ PINA1 = 1 ; Input Pins, Port A bit 1
.equ PINA2 = 2 ; Input Pins, Port A bit 2
.equ PINA3 = 3 ; Input Pins, Port A bit 3
.equ PINA4 = 4 ; Input Pins, Port A bit 4
.equ PINA5 = 5 ; Input Pins, Port A bit 5
.equ PINA6 = 6 ; Input Pins, Port A bit 6
.equ PINA7 = 7 ; Input Pins, Port A bit 7

; ***** 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
.equ PORTC7 = 7 ; Port C Data Register bit 7
.equ PC7 = 7 ; 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
.equ DDC7 = 7 ; Port C Data Direction Register bit 7

; 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
.equ PINC7 = 7 ; Port C Input Pins bit 7

; ***** 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

```

```

; ***** PORTE *****
; PORTE - Data Register, Port E
.equ  PORTE0 = 0      ;
.equ  PE0    = 0      ; For compatibility
.equ  PORTE1 = 1      ;
.equ  PE1    = 1      ; For compatibility
.equ  PORTE2 = 2      ;
.equ  PE2    = 2      ; For compatibility
.equ  PORTE3 = 3      ;
.equ  PE3    = 3      ; For compatibility
.equ  PORTE4 = 4      ;
.equ  PE4    = 4      ; For compatibility
.equ  PORTE5 = 5      ;
.equ  PE5    = 5      ; For compatibility
.equ  PORTE6 = 6      ;
.equ  PE6    = 6      ; For compatibility
.equ  PORTE7 = 7      ;
.equ  PE7    = 7      ; For compatibility

; DDRE - Data Direction Register, Port E
.equ  DDE0  = 0      ;
.equ  DDE1  = 1      ;
.equ  DDE2  = 2      ;
.equ  DDE3  = 3      ;
.equ  DDE4  = 4      ;
.equ  DDE5  = 5      ;
.equ  DDE6  = 6      ;
.equ  DDE7  = 7      ;

; PINE - Input Pins, Port E
.equ  PINE0 = 0      ;
.equ  PINE1 = 1      ;
.equ  PINE2 = 2      ;
.equ  PINE3 = 3      ;
.equ  PINE4 = 4      ;
.equ  PINE5 = 5      ;
.equ  PINE6 = 6      ;
.equ  PINE7 = 7      ;

; ***** PORTF *****
; PORTF - Data Register, Port F
.equ  PORTF0 = 0      ;
.equ  PF0   = 0      ; For compatibility
.equ  PORTF1 = 1      ;
.equ  PF1   = 1      ; For compatibility
.equ  PORTF2 = 2      ;
.equ  PF2   = 2      ; For compatibility
.equ  PORTF3 = 3      ;
.equ  PF3   = 3      ; For compatibility
.equ  PORTF4 = 4      ;
.equ  PF4   = 4      ; For compatibility
.equ  PORTF5 = 5      ;
.equ  PF5   = 5      ; For compatibility
.equ  PORTF6 = 6      ;
.equ  PF6   = 6      ; For compatibility
.equ  PORTF7 = 7      ;

```

```

.equ PF7 = 7 ; For compatibility

; DDRF - Data Direction Register, Port F
.equ DDF0 = 0 ;
.equ DDF1 = 1 ;
.equ DDF2 = 2 ;
.equ DDF3 = 3 ;
.equ DDF4 = 4 ;
.equ DDF5 = 5 ;
.equ DDF6 = 6 ;
.equ DDF7 = 7 ;

; PINF - Input Pins, Port F
.equ PINF0 = 0 ;
.equ PINF1 = 1 ;
.equ PINF2 = 2 ;
.equ PINF3 = 3 ;
.equ PINF4 = 4 ;
.equ PINF5 = 5 ;
.equ PINF6 = 6 ;
.equ PINF7 = 7 ;

; ***** PORTG *****
; PORTG - Data Register, Port G
.equ PORTG0 = 0 ;
.equ PG0 = 0 ; For compatibility
.equ PORTG1 = 1 ;
.equ PG1 = 1 ; For compatibility
.equ PORTG2 = 2 ;
.equ PG2 = 2 ; For compatibility
.equ PORTG3 = 3 ;
.equ PG3 = 3 ; For compatibility
.equ PORTG4 = 4 ;
.equ PG4 = 4 ; For compatibility
.equ PORTG5 = 5 ;
.equ PG5 = 5 ; For compatibility

; DDRG - Data Direction Register, Port G
.equ DDG0 = 0 ;
.equ DDG1 = 1 ;
.equ DDG2 = 2 ;
.equ DDG3 = 3 ;
.equ DDG4 = 4 ;
.equ DDG5 = 5 ;

; PING - Input Pins, Port G
.equ PING0 = 0 ;
.equ PING1 = 1 ;
.equ PING2 = 2 ;
.equ PING3 = 3 ;
.equ PING4 = 4 ;
.equ PING5 = 5 ;

; ***** PORTH *****
; PORTH - PORT H Data Register
.equ PORTH0 = 0 ; PORT H Data Register bit 0

```

```

.equ PH0      = 0      ; For compatibility
.equ PPORTH1 = 1      ; PORT H Data Register bit 1
.equ PH1      = 1      ; For compatibility
.equ PPORTH2 = 2      ; PORT H Data Register bit 2
.equ PH2      = 2      ; For compatibility
.equ PPORTH3 = 3      ; PORT H Data Register bit 3
.equ PH3      = 3      ; For compatibility
.equ PPORTH4 = 4      ; PORT H Data Register bit 4
.equ PH4      = 4      ; For compatibility
.equ PPORTH5 = 5      ; PORT H Data Register bit 5
.equ PH5      = 5      ; For compatibility
.equ PPORTH6 = 6      ; PORT H Data Register bit 6
.equ PH6      = 6      ; For compatibility
.equ PPORTH7 = 7      ; PORT H Data Register bit 7
.equ PH7      = 7      ; For compatibility

; DDRH - PORT H Data Direction Register
.equ DDH0      = 0      ; PORT H Data Direction Register bit 0
.equ DDH1      = 1      ; PORT H Data Direction Register bit 1
.equ DDH2      = 2      ; PORT H Data Direction Register bit 2
.equ DDH3      = 3      ; PORT H Data Direction Register bit 3
.equ DDH4      = 4      ; PORT H Data Direction Register bit 4
.equ DDH5      = 5      ; PORT H Data Direction Register bit 5
.equ DDH6      = 6      ; PORT H Data Direction Register bit 6
.equ DDH7      = 7      ; PORT H Data Direction Register bit 7

; PINH - PORT H Input Pins
.equ PINH0     = 0      ; PORT H Input Pins bit 0
.equ PINH1     = 1      ; PORT H Input Pins bit 1
.equ PINH2     = 2      ; PORT H Input Pins bit 2
.equ PINH3     = 3      ; PORT H Input Pins bit 3
.equ PINH4     = 4      ; PORT H Input Pins bit 4
.equ PINH5     = 5      ; PORT H Input Pins bit 5
.equ PINH6     = 6      ; PORT H Input Pins bit 6
.equ PINH7     = 7      ; PORT H Input Pins bit 7

; ***** PORTJ *****
; PORTJ - PORT J Data Register
.equ PORTJ0    = 0      ; PORT J Data Register bit 0
.equ PJ0       = 0      ; For compatibility
.equ PORTJ1    = 1      ; PORT J Data Register bit 1
.equ PJ1       = 1      ; For compatibility
.equ PORTJ2    = 2      ; PORT J Data Register bit 2
.equ PJ2       = 2      ; For compatibility
.equ PORTJ3    = 3      ; PORT J Data Register bit 3
.equ PJ3       = 3      ; For compatibility
.equ PORTJ4    = 4      ; PORT J Data Register bit 4
.equ PJ4       = 4      ; For compatibility
.equ PORTJ5    = 5      ; PORT J Data Register bit 5
.equ PJ5       = 5      ; For compatibility
.equ PORTJ6    = 6      ; PORT J Data Register bit 6
.equ PJ6       = 6      ; For compatibility
.equ PORTJ7    = 7      ; PORT J Data Register bit 7
.equ PJ7       = 7      ; For compatibility

; DDRJ - PORT J Data Direction Register
.equ DDJ0      = 0      ; PORT J Data Direction Register bit 0

```

```

.equ DDJ1 = 1 ; PORT J Data Direction Register bit 1
.equ DDJ2 = 2 ; PORT J Data Direction Register bit 2
.equ DDJ3 = 3 ; PORT J Data Direction Register bit 3
.equ DDJ4 = 4 ; PORT J Data Direction Register bit 4
.equ DDJ5 = 5 ; PORT J Data Direction Register bit 5
.equ DDJ6 = 6 ; PORT J Data Direction Register bit 6
.equ DDJ7 = 7 ; PORT J Data Direction Register bit 7

; PINJ - PORT J Input Pins
.equ PINJ0 = 0 ; PORT J Input Pins bit 0
.equ PINJ1 = 1 ; PORT J Input Pins bit 1
.equ PINJ2 = 2 ; PORT J Input Pins bit 2
.equ PINJ3 = 3 ; PORT J Input Pins bit 3
.equ PINJ4 = 4 ; PORT J Input Pins bit 4
.equ PINJ5 = 5 ; PORT J Input Pins bit 5
.equ PINJ6 = 6 ; PORT J Input Pins bit 6
.equ PINJ7 = 7 ; PORT J Input Pins bit 7

; ***** PORTK *****
; PORTK - PORT K Data Register
.equ PORTK0 = 0 ; PORT K Data Register bit 0
.equ PK0 = 0 ; For compatibility
.equ PORTK1 = 1 ; PORT K Data Register bit 1
.equ PK1 = 1 ; For compatibility
.equ PORTK2 = 2 ; PORT K Data Register bit 2
.equ PK2 = 2 ; For compatibility
.equ PORTK3 = 3 ; PORT K Data Register bit 3
.equ PK3 = 3 ; For compatibility
.equ PORTK4 = 4 ; PORT K Data Register bit 4
.equ PK4 = 4 ; For compatibility
.equ PORTK5 = 5 ; PORT K Data Register bit 5
.equ PK5 = 5 ; For compatibility
.equ PORTK6 = 6 ; PORT K Data Register bit 6
.equ PK6 = 6 ; For compatibility
.equ PORTK7 = 7 ; PORT K Data Register bit 7
.equ PK7 = 7 ; For compatibility

; DDRK - PORT K Data Direction Register
.equ DDK0 = 0 ; PORT K Data Direction Register bit 0
.equ DDK1 = 1 ; PORT K Data Direction Register bit 1
.equ DDK2 = 2 ; PORT K Data Direction Register bit 2
.equ DDK3 = 3 ; PORT K Data Direction Register bit 3
.equ DDK4 = 4 ; PORT K Data Direction Register bit 4
.equ DDK5 = 5 ; PORT K Data Direction Register bit 5
.equ DDK6 = 6 ; PORT K Data Direction Register bit 6
.equ DDK7 = 7 ; PORT K Data Direction Register bit 7

; PINK - PORT K Input Pins
.equ PINK0 = 0 ; PORT K Input Pins bit 0
.equ PINK1 = 1 ; PORT K Input Pins bit 1
.equ PINK2 = 2 ; PORT K Input Pins bit 2
.equ PINK3 = 3 ; PORT K Input Pins bit 3
.equ PINK4 = 4 ; PORT K Input Pins bit 4
.equ PINK5 = 5 ; PORT K Input Pins bit 5
.equ PINK6 = 6 ; PORT K Input Pins bit 6
.equ PINK7 = 7 ; PORT K Input Pins bit 7

```

```

; ***** PORTL *****
; PORTL - PORT L Data Register
.equ PORTL0 = 0 ; PORT L Data Register bit 0
.equ PL0 = 0 ; For compatibility
.equ PORTL1 = 1 ; PORT L Data Register bit 1
.equ PL1 = 1 ; For compatibility
.equ PORTL2 = 2 ; PORT L Data Register bit 2
.equ PL2 = 2 ; For compatibility
.equ PORTL3 = 3 ; PORT L Data Register bit 3
.equ PL3 = 3 ; For compatibility
.equ PORTL4 = 4 ; PORT L Data Register bit 4
.equ PL4 = 4 ; For compatibility
.equ PORTL5 = 5 ; PORT L Data Register bit 5
.equ PL5 = 5 ; For compatibility
.equ PORTL6 = 6 ; PORT L Data Register bit 6
.equ PL6 = 6 ; For compatibility
.equ PORTL7 = 7 ; PORT L Data Register bit 7
.equ PL7 = 7 ; For compatibility

; DDRL - PORT L Data Direction Register
.equ DDL0 = 0 ; PORT L Data Direction Register bit 0
.equ DDL1 = 1 ; PORT L Data Direction Register bit 1
.equ DDL2 = 2 ; PORT L Data Direction Register bit 2
.equ DDL3 = 3 ; PORT L Data Direction Register bit 3
.equ DDL4 = 4 ; PORT L Data Direction Register bit 4
.equ DDL5 = 5 ; PORT L Data Direction Register bit 5
.equ DDL6 = 6 ; PORT L Data Direction Register bit 6
.equ DDL7 = 7 ; PORT L Data Direction Register bit 7

; PINL - PORT L Input Pins
.equ PINL0 = 0 ; PORT L Input Pins bit 0
.equ PINL1 = 1 ; PORT L Input Pins bit 1
.equ PINL2 = 2 ; PORT L Input Pins bit 2
.equ PINL3 = 3 ; PORT L Input Pins bit 3
.equ PINL4 = 4 ; PORT L Input Pins bit 4
.equ PINL5 = 5 ; PORT L Input Pins bit 5
.equ PINL6 = 6 ; PORT L Input Pins bit 6
.equ PINL7 = 7 ; PORT L 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

; ***** 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 Generation Mode
.equ WGM21 = 1 ; Waveform Generation 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

; ***** 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

; ***** USART1 *****
; UDR1 - USART I/O Data Register
.equ UDR1_0  = 0      ; USART I/O Data Register bit 0
.equ UDR1_1  = 1      ; USART I/O Data Register bit 1
.equ UDR1_2  = 2      ; USART I/O Data Register bit 2
.equ UDR1_3  = 3      ; USART I/O Data Register bit 3
.equ UDR1_4  = 4      ; USART I/O Data Register bit 4
.equ UDR1_5  = 5      ; USART I/O Data Register bit 5
.equ UDR1_6  = 6      ; USART I/O Data Register bit 6
.equ UDR1_7  = 7      ; USART I/O Data Register bit 7

; UCSR1A - USART Control and Status Register A
.equ MPCM1   = 0      ; Multi-processor Communication Mode
.equ U2X1    = 1      ; Double the USART transmission speed
.equ UPE1    = 2      ; Parity Error
.equ DOR1    = 3      ; Data overRun
.equ FE1     = 4      ; Framing Error
.equ UDRE1   = 5      ; USART Data Register Empty
.equ TXC1    = 6      ; USART Transmitt Complete
.equ RXC1    = 7      ; USART Receive Complete

; UCSR1B - USART Control and Status Register B
.equ TXB81   = 0      ; Transmit Data Bit 8
.equ RXB81   = 1      ; Receive Data Bit 8
.equ UCSZ12  = 2      ; Character Size
.equ TXEN1   = 3      ; Transmitter Enable
.equ RXEN1   = 4      ; Receiver Enable
.equ UDRIE1  = 5      ; USART Data register Empty Interrupt Enable
.equ TXCIE1  = 6      ; TX Complete Interrupt Enable
.equ RXCIE1  = 7      ; RX Complete Interrupt Enable

; UCSR1C - USART Control and Status Register C
.equ UCPOL1  = 0      ; Clock Polarity
.equ UCSZ10  = 1      ; Character Size
.equ UCPHA1  = UCSZ10  ; For compatibility
.equ UCSZ11  = 2      ; Character Size
.equ UDORD1  = UCSZ11  ; For compatibility
.equ USBS1   = 3      ; Stop Bit Select
.equ UPM10   = 4      ; Parity Mode Bit 0
.equ UPM11   = 5      ; Parity Mode Bit 1
.equ UMSEL10  = 6      ; USART Mode Select
.equ UMSEL11  = 7      ; USART Mode Select

```

```

; UBRR1H - USART Baud Rate Register High Byte
.equ UBRR_8 = 0 ; USART Baud Rate Register bit 8
.equ UBRR_9 = 1 ; USART Baud Rate Register bit 9
.equ UBRR_10 = 2 ; USART Baud Rate Register bit 10
.equ UBRR_11 = 3 ; USART Baud Rate Register bit 11

; UBRR1L - USART Baud Rate Register Low Byte
.equ UBRR_0 = 0 ; USART Baud Rate Register bit 0
.equ UBRR_1 = 1 ; USART Baud Rate Register bit 1
.equ UBRR_2 = 2 ; USART Baud Rate Register bit 2
.equ UBRR_3 = 3 ; USART Baud Rate Register bit 3
.equ UBRR_4 = 4 ; USART Baud Rate Register bit 4
.equ UBRR_5 = 5 ; USART Baud Rate Register bit 5
.equ UBRR_6 = 6 ; USART Baud Rate Register bit 6
.equ UBRR_7 = 7 ; USART Baud Rate Register bit 7

; ***** EEPROM *****
; EEARH - EEPROM Address Register Low Byte
.equ EEAR8 = 0 ; EEPROM Read/Write Access Bit 8
.equ EEAR9 = 1 ; EEPROM Read/Write Access Bit 9
.equ EEAR10 = 2 ; EEPROM Read/Write Access Bit 10
.equ EEAR11 = 3 ; EEPROM Read/Write Access Bit 11

; 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

; 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 EEEPE = 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

; ***** TIMER_COUNTER_5 *****
; TIMSK5 - Timer/Counter5 Interrupt Mask Register
.equ TOIE5 = 0 ; Timer/Counter5 Overflow Interrupt Enable
.equ OCIE5A = 1 ; Timer/Counter5 Output Compare A Match Interrupt Enable
.equ OCIE5B = 2 ; Timer/Counter5 Output Compare B Match Interrupt Enable

```

```

.equ OCIE5C = 3      ; Timer/Counter5 Output Compare C Match Interrupt Enable
.equ ICIE5  = 5      ; Timer/Counter5 Input Capture Interrupt Enable

; TIFR5 - Timer/Counter5 Interrupt Flag register
.equ TOV5  = 0      ; Timer/Counter5 Overflow Flag
.equ OCF5A = 1      ; Output Compare Flag 5A
.equ OCF5B = 2      ; Output Compare Flag 5B
.equ OCF5C = 3      ; Output Compare Flag 5C
.equ ICF5  = 5      ; Input Capture Flag 5

; TCCR5A - Timer/Counter5 Control Register A
.equ WGM50 = 0      ; Waveform Generation Mode
.equ WGM51 = 1      ; Waveform Generation Mode
.equ COM5C0 = 2      ; Compare Output Mode 5C, bit 0
.equ COM5C1 = 3      ; Compare Output Mode 5C, bit 1
.equ COM5B0 = 4      ; Compare Output Mode 5B, bit 0
.equ COM5B1 = 5      ; Compare Output Mode 5B, bit 1
.equ COM5A0 = 6      ; Compare Output Mode 5A, bit 0
.equ COM5A1 = 7      ; Compare Output Mode 1A, bit 1

; TCCR5B - Timer/Counter5 Control Register B
.equ CS50  = 0      ; Prescaler source of Timer/Counter 5
.equ CS51  = 1      ; Prescaler source of Timer/Counter 5
.equ CS52  = 2      ; Prescaler source of Timer/Counter 5
.equ WGM52 = 3      ; Waveform Generation Mode
.equ WGM53 = 4      ; Waveform Generation Mode
.equ ICES5 = 6      ; Input Capture 5 Edge Select
.equ ICNC5 = 7      ; Input Capture 5 Noise Canceler

; TCCR5C - Timer/Counter 5 Control Register C
.equ FOC5C = 5      ; Force Output Compare 5C
.equ FOC5B = 6      ; Force Output Compare 5B
.equ FOC5A = 7      ; Force Output Compare 5A

; ICR5H - Timer/Counter5 Input Capture Register High Byte
.equ ICR5H0 = 0      ; Timer/Counter5 Input Capture Register High Byte bit 0
.equ ICR5H1 = 1      ; Timer/Counter5 Input Capture Register High Byte bit 1
.equ ICR5H2 = 2      ; Timer/Counter5 Input Capture Register High Byte bit 2
.equ ICR5H3 = 3      ; Timer/Counter5 Input Capture Register High Byte bit 3
.equ ICR5H4 = 4      ; Timer/Counter5 Input Capture Register High Byte bit 4
.equ ICR5H5 = 5      ; Timer/Counter5 Input Capture Register High Byte bit 5
.equ ICR5H6 = 6      ; Timer/Counter5 Input Capture Register High Byte bit 6
.equ ICR5H7 = 7      ; Timer/Counter5 Input Capture Register High Byte bit 7

; ICR5L - Timer/Counter5 Input Capture Register Low Byte
.equ ICR5L0 = 0      ; Timer/Counter5 Input Capture Register Low Byte bit 0
.equ ICR5L1 = 1      ; Timer/Counter5 Input Capture Register Low Byte bit 1
.equ ICR5L2 = 2      ; Timer/Counter5 Input Capture Register Low Byte bit 2
.equ ICR5L3 = 3      ; Timer/Counter5 Input Capture Register Low Byte bit 3
.equ ICR5L4 = 4      ; Timer/Counter5 Input Capture Register Low Byte bit 4
.equ ICR5L5 = 5      ; Timer/Counter5 Input Capture Register Low Byte bit 5
.equ ICR5L6 = 6      ; Timer/Counter5 Input Capture Register Low Byte bit 6
.equ ICR5L7 = 7      ; Timer/Counter5 Input Capture Register Low Byte bit 7

; ***** TIMER_COUNTER_4 *****
; TIMSK4 - Timer/Counter4 Interrupt Mask Register
.equ TOIE4 = 0      ; Timer/Counter4 Overflow Interrupt Enable

```

```

.equ OCIE4A = 1 ; Timer/Counter4 Output Compare A Match Interrupt Enable
.equ OCIE4B = 2 ; Timer/Counter4 Output Compare B Match Interrupt Enable
.equ OCIE4C = 3 ; Timer/Counter4 Output Compare C Match Interrupt Enable
.equ ICIE4 = 5 ; Timer/Counter4 Input Capture Interrupt Enable

; TIFR4 - Timer/Counter4 Interrupt Flag register
.equ TOV4 = 0 ; Timer/Counter4 Overflow Flag
.equ OCF4A = 1 ; Output Compare Flag 4A
.equ OCF4B = 2 ; Output Compare Flag 4B
.equ OCF4C = 3 ; Output Compare Flag 4C
.equ ICF4 = 5 ; Input Capture Flag 4

; TCCR4A - Timer/Counter4 Control Register A
.equ WGM40 = 0 ; Waveform Generation Mode
.equ WGM41 = 1 ; Waveform Generation Mode
.equ COM4C0 = 2 ; Compare Output Mode 4C, bit 0
.equ COM4C1 = 3 ; Compare Output Mode 4C, bit 1
.equ COM4B0 = 4 ; Compare Output Mode 4B, bit 0
.equ COM4B1 = 5 ; Compare Output Mode 4B, bit 1
.equ COM4A0 = 6 ; Compare Output Mode 4A, bit 0
.equ COM4A1 = 7 ; Compare Output Mode 1A, bit 1

; TCCR4B - Timer/Counter4 Control Register B
.equ CS40 = 0 ; Prescaler source of Timer/Counter 4
.equ CS41 = 1 ; Prescaler source of Timer/Counter 4
.equ CS42 = 2 ; Prescaler source of Timer/Counter 4
.equ WGM42 = 3 ; Waveform Generation Mode
.equ WGM43 = 4 ; Waveform Generation Mode
.equ ICES4 = 6 ; Input Capture 4 Edge Select
.equ ICNC4 = 7 ; Input Capture 4 Noise Canceler

; TCCR4C - Timer/Counter 4 Control Register C
.equ FOC4C = 5 ; Force Output Compare 4C
.equ FOC4B = 6 ; Force Output Compare 4B
.equ FOC4A = 7 ; Force Output Compare 4A

; ***** TIMER_COUNTER_3 *****
; TIMSK3 - Timer/Counter3 Interrupt Mask Register
.equ TOIE3 = 0 ; Timer/Counter3 Overflow Interrupt Enable
.equ OCIE3A = 1 ; Timer/Counter3 Output Compare A Match Interrupt Enable
.equ OCIE3B = 2 ; Timer/Counter3 Output Compare B Match Interrupt Enable
.equ OCIE3C = 3 ; Timer/Counter3 Output Compare C Match Interrupt Enable
.equ ICIE3 = 5 ; Timer/Counter3 Input Capture Interrupt Enable

; TIFR3 - Timer/Counter3 Interrupt Flag register
.equ TOV3 = 0 ; Timer/Counter3 Overflow Flag
.equ OCF3A = 1 ; Output Compare Flag 3A
.equ OCF3B = 2 ; Output Compare Flag 3B
.equ OCF3C = 3 ; Output Compare Flag 3C
.equ ICF3 = 5 ; Input Capture Flag 3

; TCCR3A - Timer/Counter3 Control Register A
.equ WGM30 = 0 ; Waveform Generation Mode
.equ WGM31 = 1 ; Waveform Generation Mode
.equ COM3C0 = 2 ; Compare Output Mode 3C, bit 0
.equ COM3C1 = 3 ; Compare Output Mode 3C, bit 1
.equ COM3B0 = 4 ; Compare Output Mode 3B, bit 0

```

```

.equ COM3B1 = 5 ; Compare Output Mode 3B, bit 1
.equ COM3A0 = 6 ; Compare Output Mode 3A, bit 0
.equ COM3A1 = 7 ; Compare Output Mode 1A, bit 1

; TCCR3B - Timer/Counter3 Control Register B
.equ CS30 = 0 ; Prescaler source of Timer/Counter 3
.equ CS31 = 1 ; Prescaler source of Timer/Counter 3
.equ CS32 = 2 ; Prescaler source of Timer/Counter 3
.equ WGM32 = 3 ; Waveform Generation Mode
.equ WGM33 = 4 ; Waveform Generation Mode
.equ ICES3 = 6 ; Input Capture 3 Edge Select
.equ ICNC3 = 7 ; Input Capture 3 Noise Canceler

; TCCR3C - Timer/Counter 3 Control Register C
.equ FOC3C = 5 ; Force Output Compare 3C
.equ FOC3B = 6 ; Force Output Compare 3B
.equ FOC3A = 7 ; Force Output Compare 3A

; ***** TIMER_COUNTER_1 *****
; TIMSK1 - Timer/Counter1 Interrupt Mask Register
.equ TOIE1 = 0 ; Timer/Counter1 Overflow Interrupt Enable
.equ OCIE1A = 1 ; Timer/Counter1 Output Compare A Match Interrupt Enable
.equ OCIE1B = 2 ; Timer/Counter1 Output Compare B Match Interrupt Enable
.equ OCIE1C = 3 ; Timer/Counter1 Output Compare C Match Interrupt Enable
.equ ICIE1 = 5 ; Timer/Counter1 Input Capture Interrupt Enable

; TIFR1 - Timer/Counter1 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 OCF1C = 3 ; Output Compare Flag 1C
.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 COM1C0 = 2 ; Compare Output Mode 1C, bit 0
.equ COM1C1 = 3 ; Compare Output Mode 1C, bit 1
.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/Counter 1 Control Register C
.equ FOC1C = 5 ; Force Output Compare 1C
.equ FOC1B = 6 ; Force Output Compare 1B
.equ FOC1A = 7 ; Force Output Compare 1A

```

```

; ***** JTAG *****
; OCDR - On-Chip Debug Related Register in I/O Memory
.equ OCDR0 = 0 ; On-Chip Debug Register Bit 0
.equ OCDR1 = 1 ; On-Chip Debug Register Bit 1
.equ OCDR2 = 2 ; On-Chip Debug Register Bit 2
.equ OCDR3 = 3 ; On-Chip Debug Register Bit 3
.equ OCDR4 = 4 ; On-Chip Debug Register Bit 4
.equ OCDR5 = 5 ; On-Chip Debug Register Bit 5
.equ OCDR6 = 6 ; On-Chip Debug Register Bit 6
.equ OCDR7 = 7 ; On-Chip Debug Register Bit 7
.equ IDRD = OCDR7 ; For compatibility

; MCUCR - MCU Control Register
.equ JTD = 7 ; JTAG Interface Disable

; MCUSR - MCU Status Register
.equ JTRF = 4 ; JTAG Reset Flag

; ***** EXTERNAL_INTERRUPT *****
; EICRA - External Interrupt Control Register A
.equ ISC00 = 0 ; External Interrupt Sense Control Bit
.equ ISC01 = 1 ; External Interrupt Sense Control Bit
.equ ISC10 = 2 ; External Interrupt Sense Control Bit
.equ ISC11 = 3 ; External Interrupt Sense Control Bit
.equ ISC20 = 4 ; External Interrupt Sense Control Bit
.equ ISC21 = 5 ; External Interrupt Sense Control Bit
.equ ISC30 = 6 ; External Interrupt Sense Control Bit
.equ ISC31 = 7 ; External Interrupt Sense Control Bit

; EICRB - External Interrupt Control Register B
.equ ISC40 = 0 ; External Interrupt 7-4 Sense Control Bit
.equ ISC41 = 1 ; External Interrupt 7-4 Sense Control Bit
.equ ISC50 = 2 ; External Interrupt 7-4 Sense Control Bit
.equ ISC51 = 3 ; External Interrupt 7-4 Sense Control Bit
.equ ISC60 = 4 ; External Interrupt 7-4 Sense Control Bit
.equ ISC61 = 5 ; External Interrupt 7-4 Sense Control Bit
.equ ISC70 = 6 ; External Interrupt 7-4 Sense Control Bit
.equ ISC71 = 7 ; External Interrupt 7-4 Sense Control Bit

; EIMSK - External Interrupt Mask Register
.equ INT0 = 0 ; External Interrupt Request 0 Enable
.equ INT1 = 1 ; External Interrupt Request 1 Enable
.equ INT2 = 2 ; External Interrupt Request 2 Enable
.equ INT3 = 3 ; External Interrupt Request 3 Enable
.equ INT4 = 4 ; External Interrupt Request 4 Enable
.equ INT5 = 5 ; External Interrupt Request 5 Enable
.equ INT6 = 6 ; External Interrupt Request 6 Enable
.equ INT7 = 7 ; External Interrupt Request 7 Enable

; EIFR - External Interrupt Flag Register
.equ INTF0 = 0 ; External Interrupt Flag 0
.equ INTF1 = 1 ; External Interrupt Flag 1
.equ INTF2 = 2 ; External Interrupt Flag 2
.equ INTF3 = 3 ; External Interrupt Flag 3
.equ INTF4 = 4 ; External Interrupt Flag 4
.equ INTF5 = 5 ; External Interrupt Flag 5

```

```

.equ INTF6 = 6 ; External Interrupt Flag 6
.equ INTF7 = 7 ; External Interrupt Flag 7

; 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

; 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

; 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
.equ PCINT15 = 7 ; Pin Change Enable Mask 15

; 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

; ***** 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

; MCUCR - MCU Control Register
.equ IVCE = 0 ; Interrupt Vector Change Enable
.equ IVSEL = 1 ; Interrupt Vector Select

```

```

.equ PUD = 4 ; Pull-up disable
;.equ JTD = 7 ; JTAG Interface Disable

; MCUSR - MCU Status Register
.equ PORF = 0 ; Power-on reset flag
.equ EXTRF = 1 ; External Reset Flag
.equ BORF = 2 ; Brown-out Reset Flag
.equ WDRF = 3 ; Watchdog Reset Flag
;.equ JTRF = 4 ; JTAG Reset Flag

; XMCRA - External Memory Control Register A
.equ SRW00 = 0 ; Wait state select bit lower page
.equ SRW01 = 1 ; Wait state select bit lower page
.equ SRW10 = 2 ; Wait state select bit upper page
.equ SRW11 = 3 ; Wait state select bit upper page
.equ SRL0 = 4 ; Wait state page limit
.equ SRL1 = 5 ; Wait state page limit
.equ SRL2 = 6 ; Wait state page limit
.equ SRE = 7 ; External SRAM Enable

; XMCRB - External Memory Control Register B
.equ XMM0 = 0 ; External Memory High Mask
.equ XMM1 = 1 ; External Memory High Mask
.equ XMM2 = 2 ; External Memory High Mask
.equ XMBK = 7 ; External Memory Bus Keeper 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 -
.equ CLKPS0 = 0 ;
.equ CLKPS1 = 1 ;
.equ CLKPS2 = 2 ;
.equ CLKPS3 = 3 ;
.equ CLKPCE = 7 ;

; 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

; RAMPZ - RAM Page Z Select Register
.equ RAMPZ0 = 0 ; RAM Page Z Select Register Bit 0
.equ RAMPZ1 = 1 ; RAM Page Z Select Register Bit 1

; EIND - Extended Indirect Register
.equ EIND0 = 0 ; Bit 0

; GPIOR2 - General Purpose IO Register 2
.equ GPIOR20 = 0 ; General Purpose IO Register 2 bit 0

```

```

.equ GPIO21      = 1      ; General Purpose IO Register 2 bit 1
.equ GPIO22      = 2      ; General Purpose IO Register 2 bit 2
.equ GPIO23      = 3      ; General Purpose IO Register 2 bit 3
.equ GPIO24      = 4      ; General Purpose IO Register 2 bit 4
.equ GPIO25      = 5      ; General Purpose IO Register 2 bit 5
.equ GPIO26      = 6      ; General Purpose IO Register 2 bit 6
.equ GPIO27      = 7      ; General Purpose IO Register 2 bit 7

; GPIO1 - General Purpose IO Register 1
.equ GPIO10      = 0      ; General Purpose IO Register 1 bit 0
.equ GPIO11      = 1      ; General Purpose IO Register 1 bit 1
.equ GPIO12      = 2      ; General Purpose IO Register 1 bit 2
.equ GPIO13      = 3      ; General Purpose IO Register 1 bit 3
.equ GPIO14      = 4      ; General Purpose IO Register 1 bit 4
.equ GPIO15      = 5      ; General Purpose IO Register 1 bit 5
.equ GPIO16      = 6      ; General Purpose IO Register 1 bit 6
.equ GPIO17      = 7      ; General Purpose IO Register 1 bit 7

; GPIO0 - General Purpose IO Register 0
.equ GPIO00      = 0      ; General Purpose IO Register 0 bit 0
.equ GPIO01      = 1      ; General Purpose IO Register 0 bit 1
.equ GPIO02      = 2      ; General Purpose IO Register 0 bit 2
.equ GPIO03      = 3      ; General Purpose IO Register 0 bit 3
.equ GPIO04      = 4      ; General Purpose IO Register 0 bit 4
.equ GPIO05      = 5      ; General Purpose IO Register 0 bit 5
.equ GPIO06      = 6      ; General Purpose IO Register 0 bit 6
.equ GPIO07      = 7      ; General Purpose IO Register 0 bit 7

; PRR1 - Power Reduction Register1
.equ PRUSART1    = 0      ; Power Reduction USART1
.equ PRUSART2    = 1      ; Power Reduction USART2
.equ PRUSART3    = 2      ; Power Reduction USART3
.equ PRTIM3 = 3      ; Power Reduction Timer/Counter3
.equ PRTIM4 = 4      ; Power Reduction Timer/Counter4
.equ PRTIM5 = 5      ; Power Reduction Timer/Counter5

; PRR0 - Power Reduction Register0
.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

; ***** 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 MUX4 = 4      ; 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 MUX5 = 3 ; Analog Channel and Gain Selection Bits
;.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 ;
.equ ADC6D = 6 ;
.equ ADC7D = 7 ;

; DIDR2 - Digital Input Disable Register
.equ ADC8D = 0 ;
.equ ADC9D = 1 ;
.equ ADC10D = 2 ;
.equ ADC11D = 3 ;
.equ ADC12D = 4 ;
.equ ADC13D = 5 ;
.equ ADC14D = 6 ;
.equ ADC15D = 7 ;

; ***** BOOT_LOAD *****

```

```

; SPMCSR - Store Program Memory Control Register
.equ SPMEN = 0 ; Store Program Memory Enable
.equ PGERS = 1 ; Page Erase
.equ PGWRT = 2 ; Page Write
.equ BLBSET = 3 ; Boot Lock Bit Set
.equ RWWWSRE = 4 ; Read While Write section read enable
.equ SIGRD = 5 ; Signature Row Read
.equ RWWSB = 6 ; Read While Write Section Busy
.equ SPMIE = 7 ; SPM Interrupt Enable

; ***** USART2 *****
; UDR2 - USART I/O Data Register
.equ UDR2_0 = 0 ; USART I/O Data Register bit 0
.equ UDR2_1 = 1 ; USART I/O Data Register bit 1
.equ UDR2_2 = 2 ; USART I/O Data Register bit 2
.equ UDR2_3 = 3 ; USART I/O Data Register bit 3
.equ UDR2_4 = 4 ; USART I/O Data Register bit 4
.equ UDR2_5 = 5 ; USART I/O Data Register bit 5
.equ UDR2_6 = 6 ; USART I/O Data Register bit 6
.equ UDR2_7 = 7 ; USART I/O Data Register bit 7

; UCSR2A - USART Control and Status Register A
.equ MPCM2 = 0 ; Multi-processor Communication Mode
.equ U2X2 = 1 ; Double the USART transmission speed
.equ UPE2 = 2 ; Parity Error
.equ DOR2 = 3 ; Data overRun
.equ FE2 = 4 ; Framing Error
.equ UDRE2 = 5 ; USART Data Register Empty
.equ TXC2 = 6 ; USART Transmitt Complete
.equ RXC2 = 7 ; USART Receive Complete

; UCSR2B - USART Control and Status Register B
.equ TXB82 = 0 ; Transmit Data Bit 8
.equ RXB82 = 1 ; Receive Data Bit 8
.equ UCSZ22 = 2 ; Character Size
.equ TXEN2 = 3 ; Transmitter Enable
.equ RXEN2 = 4 ; Receiver Enable
.equ UDRIE2 = 5 ; USART Data register Empty Interrupt Enable
.equ TXCIE2 = 6 ; TX Complete Interrupt Enable
.equ RXCIE2 = 7 ; RX Complete Interrupt Enable

; UCSR2C - USART Control and Status Register C
.equ UCPOL2 = 0 ; Clock Polarity
.equ UCSZ20 = 1 ; Character Size
.equ UCSZ21 = 2 ; Character Size
.equ USBS2 = 3 ; Stop Bit Select
.equ UPM20 = 4 ; Parity Mode Bit 0
.equ UPM21 = 5 ; Parity Mode Bit 1
.equ UMSEL20 = 6 ; USART Mode Select
.equ UMSEL21 = 7 ; USART Mode Select

; UBRR2H - 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

```

```

; UBRR2L - 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

; ***** USART3 *****
; UDR3 - USART I/O Data Register
.equ UDR3_0 = 0 ; USART I/O Data Register bit 0
.equ UDR3_1 = 1 ; USART I/O Data Register bit 1
.equ UDR3_2 = 2 ; USART I/O Data Register bit 2
.equ UDR3_3 = 3 ; USART I/O Data Register bit 3
.equ UDR3_4 = 4 ; USART I/O Data Register bit 4
.equ UDR3_5 = 5 ; USART I/O Data Register bit 5
.equ UDR3_6 = 6 ; USART I/O Data Register bit 6
.equ UDR3_7 = 7 ; USART I/O Data Register bit 7

; UCSRA - USART Control and Status Register A
.equ MPCM3 = 0 ; Multi-processor Communication Mode
.equ U2X3 = 1 ; Double the USART transmission speed
.equ UPE3 = 2 ; Parity Error
.equ DOR3 = 3 ; Data overRun
.equ FE3 = 4 ; Framing Error
.equ UDRE3 = 5 ; USART Data Register Empty
.equ TXC3 = 6 ; USART Transmitt Complete
.equ RXC3 = 7 ; USART Receive Complete

; UCSRB - USART Control and Status Register B
.equ TXB83 = 0 ; Transmit Data Bit 8
.equ RXB83 = 1 ; Receive Data Bit 8
.equ UCSZ32 = 2 ; Character Size
.equ TXEN3 = 3 ; Transmitter Enable
.equ RXEN3 = 4 ; Receiver Enable
.equ UDRIE3 = 5 ; USART Data register Empty Interrupt Enable
.equ TXCIE3 = 6 ; TX Complete Interrupt Enable
.equ RXCIE3 = 7 ; RX Complete Interrupt Enable

; UCSRC - USART Control and Status Register C
.equ UCPOL3 = 0 ; Clock Polarity
.equ UCSZ30 = 1 ; Character Size
.equ UCSZ31 = 2 ; Character Size
.equ USBS3 = 3 ; Stop Bit Select
.equ UPM30 = 4 ; Parity Mode Bit 0
.equ UPM31 = 5 ; Parity Mode Bit 1
.equ UMSEL30 = 6 ; USART Mode Select
.equ UMSEL31 = 7 ; USART Mode Select

; UBRR3H - 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

```

```

; UBRR3L - 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

; ***** 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 BOOTRST = 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 JTAGEN = 6 ; Enable JTAG
.equ OCDEN = 7 ; Enable OCD

; 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      = 0x1fffff ; Note: Word address
.equ IOEND        = 0x01ff
.equ SRAM_START    = 0x0200
.equ SRAM_SIZE     = 8192
.equ RAMEND       = 0x21ff
.equ XRAMEND      = 0xfffff
.equ E2END        = 0x0ffff
.equ EEPROMEND    = 0x0ffff
.equ EEADRBITS    = 12

#pragma AVRPART MEMORY PROG_FLASH 262144
#pragma AVRPART MEMORY EEPROM 4096
#pragma AVRPART MEMORY INT_SRAM SIZE 8192
#pragma AVRPART MEMORY INT_SRAM START_ADDR 0x200

; ***** BOOTLOADER DECLARATIONS *****
.equ NRWW_START_ADDR      = 0x1f000
.equ NRWW_STOP_ADDR        = 0x1fffff
.equ RWW_START_ADDR        = 0x0
.equ RWW_STOP_ADDR = 0x1effff
.equ PAGESIZE      = 128
.equ FIRSTBOOTSTART      = 0x1fe00
.equ SECONDBOOTSTART     = 0x1fc00
.equ THIRDBOOTSTART      = 0x1f800
.equ FOURTHBOOTSTART     = 0x1f000
.equ SMALLBOOTSTART      = FIRSTBOOTSTART
.equ LARGEBOOTSTART      = FOURTHBOOTSTART

; ***** INTERRUPT VECTORS *****
.equ INT0addr      = 0x0002 ; External Interrupt Request 0
.equ INT1addr      = 0x0004 ; External Interrupt Request 1
.equ INT2addr      = 0x0006 ; External Interrupt Request 2
.equ INT3addr      = 0x0008 ; External Interrupt Request 3
.equ INT4addr      = 0x000a ; External Interrupt Request 4
.equ INT5addr      = 0x000c ; External Interrupt Request 5
.equ INT6addr      = 0x000e ; External Interrupt Request 6
.equ INT7addr      = 0x0010 ; External Interrupt Request 7
.equ PCI0addr      = 0x0012 ; Pin Change Interrupt Request 0
.equ PCI1addr      = 0x0014 ; Pin Change Interrupt Request 1
.equ PCI2addr      = 0x0016 ; Pin Change Interrupt Request 2
.equ WDTaddr       = 0x0018 ; Watchdog Time-out Interrupt
.equ OC2Aaddr      = 0x001a ; Timer/Counter2 Compare Match A
.equ OC2Baddr      = 0x001c ; Timer/Counter2 Compare Match B
.equ OVF2addr      = 0x001e ; Timer/Counter2 Overflow
.equ ICP1addr      = 0x0020 ; Timer/Counter1 Capture Event
.equ OC1Aaddr      = 0x0022 ; Timer/Counter1 Compare Match A
.equ OC1Baddr      = 0x0024 ; Timer/Counter1 Compare Match B
.equ OC1Caddr      = 0x0026 ; Timer/Counter1 Compare Match C
.equ OVF1addr      = 0x0028 ; Timer/Counter1 Overflow
.equ OC0Aaddr      = 0x002a ; Timer/Counter0 Compare Match A
.equ OC0Baddr      = 0x002c ; Timer/Counter0 Compare Match B
.equ OVF0addr      = 0x002e ; Timer/Counter0 Overflow
.equ SPIaddr       = 0x0030 ; SPI Serial Transfer Complete

```

```

.equ URXC0addr = 0x0032 ; USART0, Rx Complete
.equ UDRE0addr = 0x0034 ; USART0 Data register Empty
.equ UTXC0addr = 0x0036 ; USART0, Tx Complete
.equ ACIaddr = 0x0038 ; Analog Comparator
.equ ADCCaddr = 0x003a ; ADC Conversion Complete
.equ ERDYaddr = 0x003c ; EEPROM Ready
.equ ICP3addr = 0x003e ; Timer/Counter3 Capture Event
.equ OC3Aaddr = 0x0040 ; Timer/Counter3 Compare Match A
.equ OC3Baddr = 0x0042 ; Timer/Counter3 Compare Match B
.equ OC3Caddr = 0x0044 ; Timer/Counter3 Compare Match C
.equ OVFBaddr = 0x0046 ; Timer/Counter3 Overflow
.equ URXC1addr = 0x0048 ; USART1, Rx Complete
.equ UDRE1addr = 0x004a ; USART1 Data register Empty
.equ UTXC1addr = 0x004c ; USART1, Tx Complete
.equ TWIaddr = 0x004e ; 2-wire Serial Interface
.equ SPMRaddr = 0x0050 ; Store Program Memory Read
.equ ICP4addr = 0x0052 ; Timer/Counter4 Capture Event
.equ OC4Aaddr = 0x0054 ; Timer/Counter4 Compare Match A
.equ OC4Baddr = 0x0056 ; Timer/Counter4 Compare Match B
.equ OC4Caddr = 0x0058 ; Timer/Counter4 Compare Match C
.equ OVFBaddr = 0x005a ; Timer/Counter4 Overflow
.equ ICP5addr = 0x005c ; Timer/Counter5 Capture Event
.equ OC5Aaddr = 0x005e ; Timer/Counter5 Compare Match A
.equ OC5Baddr = 0x0060 ; Timer/Counter5 Compare Match B
.equ OC5Caddr = 0x0062 ; Timer/Counter5 Compare Match C
.equ OVFBaddr = 0x0064 ; Timer/Counter5 Overflow
.equ URXC2addr = 0x0066 ; USART2, Rx Complete
.equ UDRE2addr = 0x0068 ; USART2 Data register Empty
.equ UTXC2addr = 0x006a ; USART2, Tx Complete
.equ URXC3addr = 0x006c ; USART3, Rx Complete
.equ UDRE3addr = 0x006e ; USART3 Data register Empty
.equ UTXC3addr = 0x0070 ; USART3, Tx Complete

.equ INT_VECTORS_SIZE = 114 ; size in words

#endif /* _M2560DEF_INC_ */

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

```