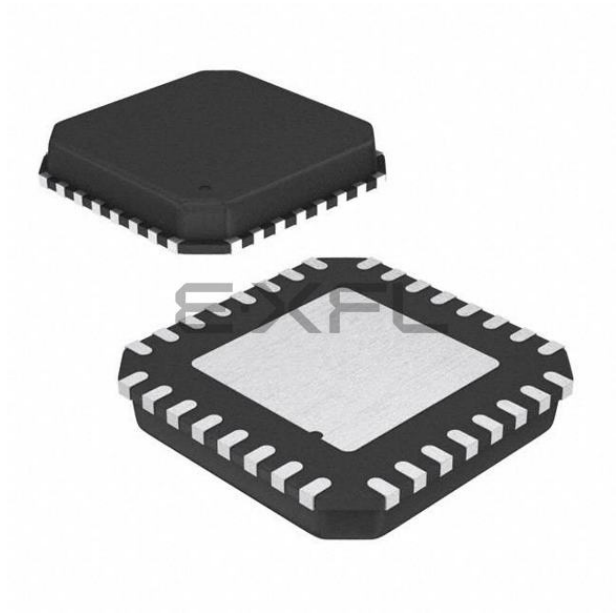


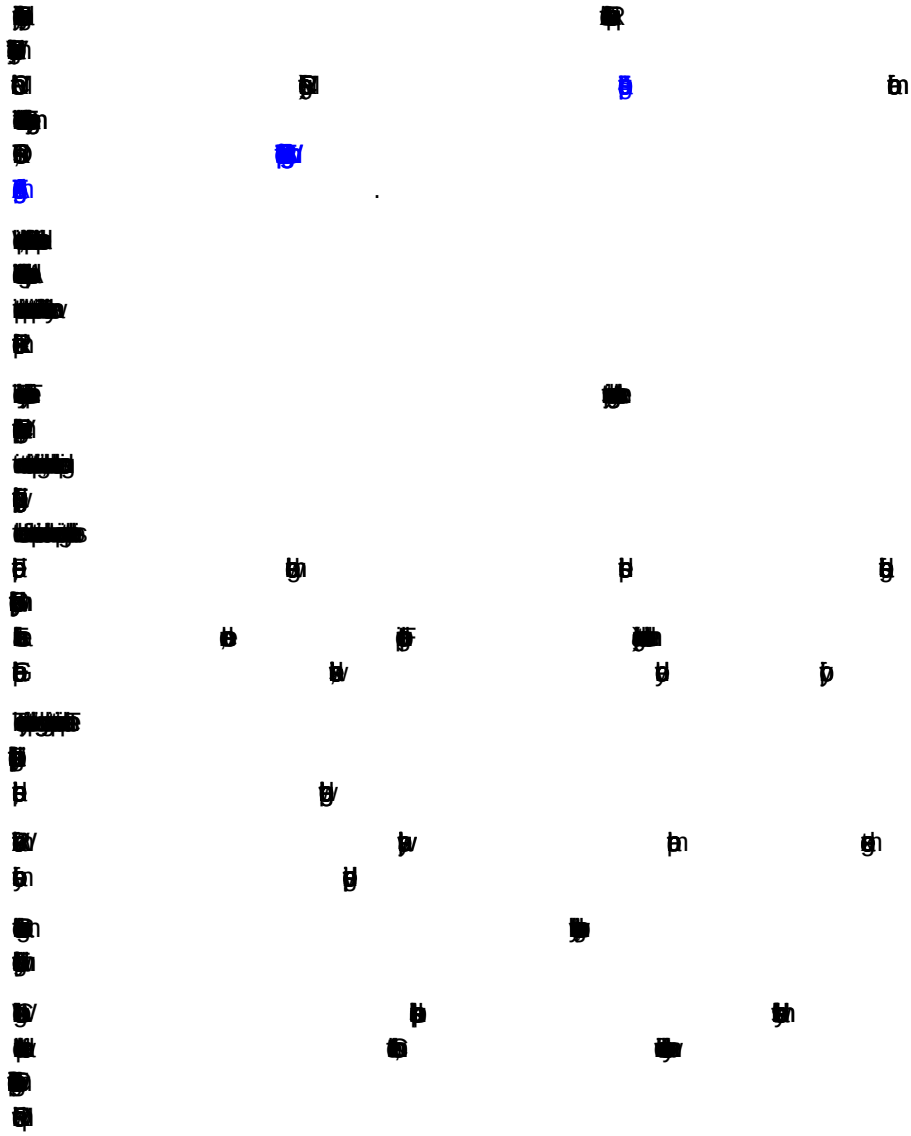
Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

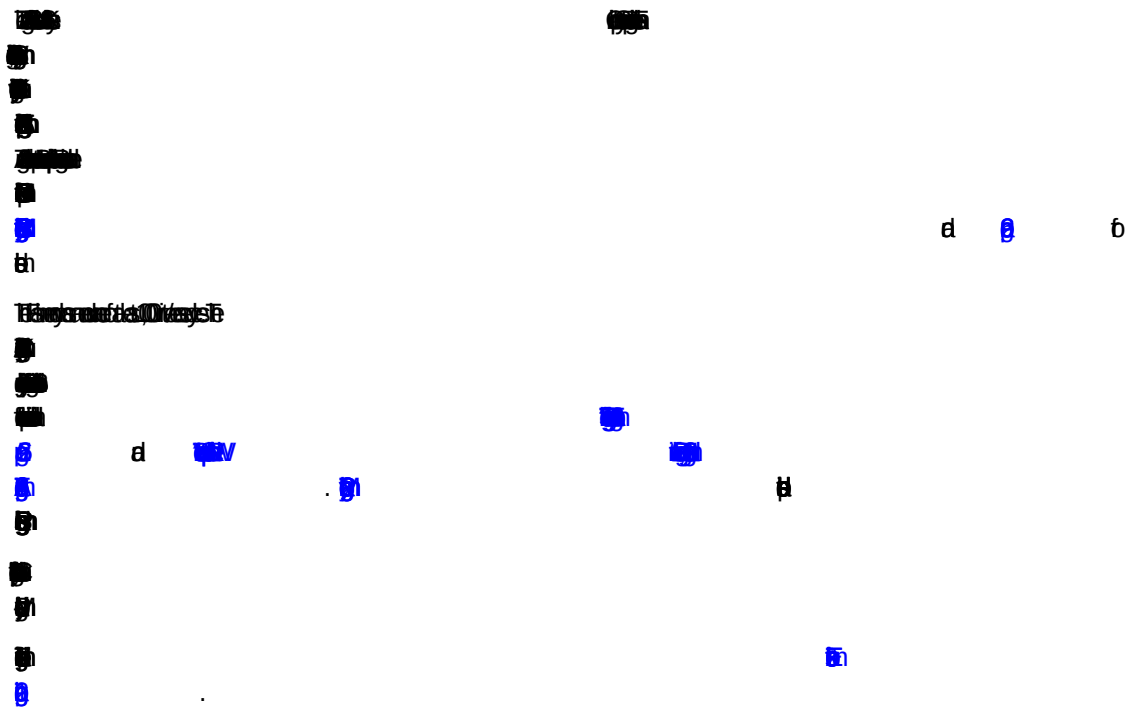
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	10MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega168v-10mi



5. AVR ATmega48/88/168 Memories

5.1 In-System Reprogrammable Flash Program Memory



- **Bit 0 – EERE: EEPROM Read Enable**

Table 5-2.

Symbol	Number of Calibrated RC Oscillator Cycles	Typ Programming Time

[illegible]

[illegible]

[illegible]

```
EEPROM_read:
  ; Wait for completion of previous write
  sbic   EECR,EEPE
  rjmp   EEPROM_read
  ; Set up address (r18:r17) in address register
  out    EEARH, r18
  out    EEARL, r17
  ; Start eeprom read by writing EERE
  sbi    EECR,EERE
  ; Read data from Data Register
  in     r16,EEDR
  ret
```

[illegible]

```
unsigned char      EEPROM_read( unsigned int      uiAddress)
{
  /* Wait for completion of previous write */
  while(EECR & (1<<EEPE))
    ;
  /* Set up address register */
  EEAR = uiAddress;
  /*    Start eeprom read by writing EERE */
  EECR |= (1<<EERE);
  /* Return data from Data Register */
  return EEDR;
}
```

5.3.5 Preventing EEPROM Corruption

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

6.1.4 Asynchronous Timer Clock – clk_{ASY}

6.1.5 ADC Clock – clk_{ADC}

6.2 Clock Sources

Table 6-1.

Device Clocking Option	CKSEL3..0
	0
	0

6.2.1 Default Clock Source

6.2.2 Clock Startup Sequence

Table 6-9.

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	SUT1..0
			0
		1	

6.6.1 Oscillator Calibration Register – OSCCAL

Bit	7	6	5	4	3	2	1	0	
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value									

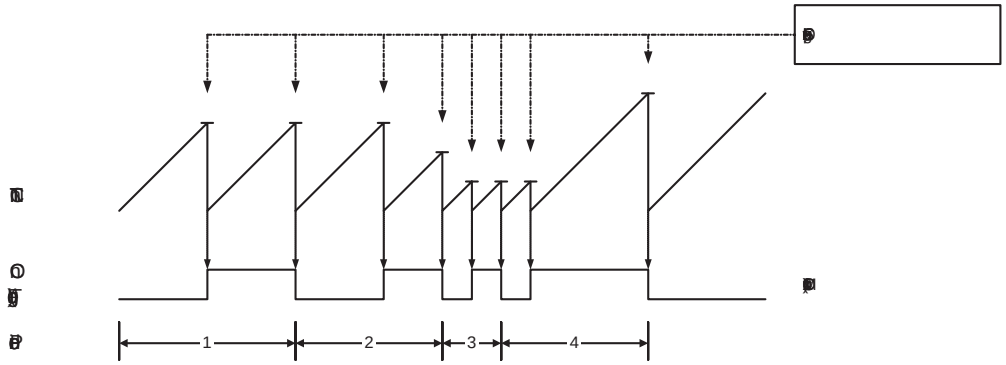
- **Bits 7..0 – CAL7..0: Oscillator Calibration Value**

Table 9-5.

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0		
1	1		
0	0		
0	1		

```
Address   Labels  Code                    Comments
0x0000            jmp     RESET           ; Reset Handler
0x0002            jmp     EXT_INT0        ; IRQ0 Handler
0x0004            jmp     EXT_INT1        ; IRQ1 Handler
0x0006            jmp     PCINT0          ; PCINT0 Handler
0x0008            jmp     PCINT1          ; PCINT1 Handler
0x000A            jmp     PCINT2          ; PCINT2 Handler
0x000C            jmp     WDT             ; Watchdog Timer Handler
0x000E            jmp     TIM2_COMPA      ; Timer2 Compare A Handler
0x0010            jmp     TIM2_COMPB      ; Timer2 Compare B Handler
0x0012            jmp     TIM2_OVF        ; Timer2 Overflow Handler
0x0014            jmp     TIM1_CAPT       ; Timer1 Capture Handler
0x0016            jmp     TIM1_COMPA      ; Timer1 Compare A Handler
0x0018            jmp     TIM1_COMPB      ; Timer1 Compare B Handler
0x001A            jmp     TIM1_OVF        ; Timer1 Overflow Handler
0x001C            jmp     TIM0_COMPA      ; Timer0 Compare A Handler
0x001E            jmp     TIM0_COMPB      ; Timer0 Compare B Handler
0x0020            jmp     TIM0_OVF        ; Timer0 Overflow Handler
0x0022            jmp     SPI_STC         ; SPI Transfer Complete Handler
0x0024            jmp     USART_RXC       ; USART, RX Complete Handler
0x0026            jmp     USART_UDRE      ; USART, UDR Empty Handler
0x0028            jmp     USART_TXC       ; USART, TX Complete Handler
0x002A            jmp     ADC             ; ADC Conversion Complete Handler
0x002C            jmp     EE_RDY          ; EEPROM Ready Handler
0x002E            jmp     ANA_COMP        ; Analog Comparator Handler
0x0030            jmp     TWI             ; 2-wire Serial Interface Handler
0x0032            jmp     SPM_RDY         ; Store Program Memory Ready Handler
```

Figure 12-5.



$$f_{OCnx} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

12.6.3 Fast PWM Mode

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
[illegible]		–	–	–	–	–	[illegible]	[illegible]		
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	[illegible]	[illegible]	[illegible]	
[illegible]		–	–	[illegible]	–	–	[illegible]	[illegible]	[illegible]	
[illegible]		–	–	–	–	–	[illegible]	[illegible]		
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]	[illegible]	[illegible]		5	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]				
[illegible]	[illegible]	P		[illegible]	[illegible]	[illegible]	[illegible]	8		
[illegible]		–	[illegible]	[illegible]	[illegible]	[illegible]				
[illegible]		–	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]		–	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]			
[illegible]	[illegible]	[illegible]		5	[illegible]	[illegible]	[illegible]	[illegible]		
[illegible]	[illegible]	[illegible]		5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]				
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	
[illegible]		–	–	–	–	–	–	–	–	

[illegible] 1. [illegible]

2. [illegible]

3. [illegible]

4. [illegible]

5. [illegible]

[illegible]

29. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
D R		R	3 R	1	
D R		R	3 R	1	
W R		R	3 R	2	
B R		R	3 R	1	
B R		R	3 R	1	
B R		R	3 R	1	
B R		R	3 R	1	
W R		R	3 R	2	
N R		R	3" R -R	1	
N R		R	3 R -"K	1	
O R		R	3 R	1	
O R		R	3" R	1	
O R		R	3 R #R	1	
M R		R	3 R 4R	1	
S R		R	3 0 4R	1	
B R		R	3 R	1	
B R		R	3 R -	1	
N R		R	3 R	1	
D R		R	3 R 41	1	
S R		R	3 R -R	1	
C R		R	3 R #R	1	
B R		R	3	1	
M R		R	3 R	2	
M R		R	3 R	2	
M R		R	3 R	2	
M R		R	3 R		2
M R		R	3 R		2
M R		R	3 R		2
BRANCH INSTRUCTIONS					
M k			"3	2	
M			3 Z		2
M	k		"3 k		3
C k			3	3	
C			3 Z		3
C	k		3 k		4
R			3	4	
R			3	4	
B R		R	"3	2	
C R		R	4R	1	
C R		R	4R 4C	1	
C R		R	4K	1	
B R		R	3	2	
B R		R	3	2	
B R		R	3	2	
B R		R	3	2	
B k		R	3	2	
B k		R	3	2	
Q k		R	3	2	
B k		R	3	2	
B k		R	3	2	
B k		R	3	2	
B k		R	3	2	
B k		R	3	2	
D k		R	3	2	
M k		R	3	2	
B k		R	3	2	
B k		N	# 3	2	
R k		N	# 3	2	
B k		R	3	2	
B k		R	3	2	
B k		R	3	2	
B k		R	3	2	
B k		R	3	2	
B k		R	3	2	