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	Active
Core Processor	ZNEO
Core Size	16-Bit
Speed	20MHz
Connectivity	EBI/EMI, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	76
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16f3211al20ag

Table 12 shows the External Chip Select Control Registers Low for $\overline{CS1}$(EXTSC1L). This register sets the number of wait states for Chip Select 1. Waits are only added if the chip select is enabled.

Table 12. External Chip Select Control Registers Low for $\overline{CS1}$ (EXTCS1L)

Bits	7	6	5	4	3	2	1	0
Field	RESERVED		PR1WAIT		CS1WAIT			
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FF_(E075)H							

Bit	Description
[7:6]	**Reserved** These bits are reserved and must be programmed to 00.
[5:4] PR1WAIT[2:0]	**Post Read Wait Selection** 00 = 0 wait state. 01 = 1 wait state. 10 = 2 wait states. 11 = 3 wait states.
[3:0] CS1WAIT	**Chip Select 1 Wait Selection** 0000 = 0 wait state. 0001 = 1 wait state. 0010 = 2 wait states. 0011 = 3 wait states. 0100 = 4 wait states. 0101 = 5 wait states. 0110 = 6 wait states. 0111 = 7 wait states. 1000 = 8 wait states. 1001 = 9 wait states. 1010 = 10 wait states. 1011 = 11 wait states. 1100 = 12 wait states. 1101 = 13 wait states. 1110 = 14 wait states. 1111 = 15 wait states.

Halt Mode

Execution of the ZNEO CPU's HALT instruction places the device into Halt Mode. The following list represents the operating characteristics of the ZNEO CPU in Halt Mode:

- System clock is enabled and continues to operate
- ZNEO CPU is stopped
- PC stops incrementing
- WDT's internal RC oscillator continues to operate
- If enabled, the WDT continues to operate
- All other on-chip peripherals continue to operate

The ZNEO CPU is brought out of Halt Mode by any of the following operations:

- Interrupt or System Exception
- WDT time-out (System Exception or Reset)
- Power-On Reset
- VBO reset
- External $\overline{\text{RESET}}$ pin assertion
- Instantaneous Halt Mode recovery

To minimize current in Halt Mode, all GPIO pins which are configured as inputs must be driven to one of the supply rails (V_{DD} or V_{SS}).

Peripheral-Level Power Control

On-chip peripherals in ZNEO Z16F Series parts automatically enter a low power mode after Reset and whenever the peripheral is disabled. To minimize power consumption, unused peripherals must be disabled. See the individual peripheral chapters for specific register settings to enable or disable the peripheral.

Power Control Option Bits

User programmable option bits are available in some versions of the ZNEO Z16F Series devices that enable very low power Stop Mode operation. These options include disabling the VBO protection circuits and disabling the WDT oscillator. For detailed description of the user options that affect power management, see the Option Bits chapter on page 292.

General-Purpose Input/Output

The ZNEO® Z16F Series products contain general-purpose input/output (GPIO) pins arranged as Ports A–K. Each port contains control and data registers. The GPIO control registers are used to determine data direction, open-drain, output drive current and alternate pin functions. Each port pin is individually programmable.

GPIO Port Availability by Device

Table 23 lists the port pins available by device and package pin count.

Table 23. GPIO Port Availability by Device

Device	Pin-Count	Port A	Port B	Port C	Port D	Port E	Port F	Port G	Port H	Port J	Port K
Z16F2811	100-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]	[7:0]	[7:0]
	80-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]	—	—
Z16F6411	100-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]	[7:0]	[7:0]
	80-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]	—	—
Z16F3211	100-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]	[7:0]	[7:0]
	80-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]	—	
Z16F2810	80-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]	—	—
	68-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]	—	—
	64-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]	—	—

Architecture

Figure 18 displays a simplified block diagram of a GPIO port pin. This diagram does not display the ability to accommodate alternate functions and variable port current drive strength.

Port A-K High Drive Enable Registers

Setting the bits in the Port A-K High Drive Enable registers, shown in Table 28, to 1, configures the specified port pins for high current output drive operation. The Port A-K High Drive Enable registers affect the pins directly and as a result, alternate functions are also affected.

Table 28. Port A-K High Drive Enable Registers (P*x*HDE)

Bits	7	6	5	4	3	2	1	0
Field	PHDE7	PHDE6	PHDE5	PHDE4	PHDE3	PHDE2	PHDE1	PHDE0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FF_E103, FF_E113, FF_E123, FF_E133, FF_E143, FF_E153, FF_E163, FF_E173, FF_E183, FF_E193							

Bit	Description
[7:0] PHDE[7:0]	**Port High Drive Enabled** 0 = The port pin is configured for standard output current drive. 1 = The port pin is configured for high output current drive.

Port A-K Alternate Function High and Low Registers

The Port A-K Alternate Function High and Low registers, shown in Table 29 and Table 30 on page 75, select the alternate functions for the selected pins. To determine the alternate function associated with each port pin, see the GPIO Alternate Functions section on page 67. When changing alternate functions, it is recommended to use word data mode instructions to perform simultaneous Writes to the Port Alternate Function High and Low registers.

Caution: Do not enable alternate function for GPIO port pins which do not have an associated alternate function. Failure to follow this guideline will result in undefined operation.

Interrupt Controller

The interrupt controller on the ZNEO® Z16F Series products prioritize interrupt requests from on-chip peripherals and the GPIO port pins. The features of the interrupt controller includes:

- Flexible GPIO interrupts:
  - Eight selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Software Interrupt Requests (IRQ) assertion

The IRQs allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an ISR. Usually this service routine is involved with exchange of data, status information or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

System exceptions are nonmaskable requests which allow critical system functions to suspend CPU operation in an orderly manner and force the CPU to start a service routine. Usually this service routine tries to determine how critical the exception is. When the service routine is complete, the CPU returns to the operation from which it was interrupted.

The ZNEO Z16F Series supports both vectored and polled interrupt handling. For polled interrupts, the interrupt control has no effect on operation. For more information about interrupt servicing by the ZNEO CPU, refer to the ZNEO CPU Core User Manual (UM0188), available for download at www.zilog.com.

Interrupt Vector Listing

Table 39 lists all of the interrupts available in order of priority.

IRQ0 Enable High and Low Bit Registers

The IRQ0 enable high and low bit registers, shown in Tables 48 and 49, form a priority encoded enabling for interrupts in the Interrupt Request 0 Register. Priority is generated by setting bits in each register. Table 47 describes the priority control for IRQ0.

Table 47. IRQ0 Enable and Priority Encoding

IRQ0ENH[*x*]	IRQ0ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

Note: *x* indicates the range of register bits 0 through 7.

Table 48. IRQ0 Enable High Bit Register (IRQ0ENH)

Bits	7	6	5	4	3	2	1	0
Field	T2ENH	T1ENH	T0ENH	U0RENH	U0TENH	I2CENH	SPIENH	ADCENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FF_E032h							

Bit	Description
[7] T2ENH	Timer 2 Interrupt Request Enable High Bit.
[6] T1ENH	Timer 0 Interrupt Request Enable High Bit.
[5] T0ENH	Timer 0 Interrupt Request Enable High Bit.
[4] U0RENH	UART 0 Receive Interrupt Request Enable High Bit.
[3] U0TENH	UART 0 Transmit Interrupt Request Enable High Bit.
[2] I2CENH	I^2C Interrupt Request Enable High Bit.
[1] SPIENH	SPI Interrupt Request Enable High Bit.
[0] ADCENH	ADC Interrupt Request Enable High Bit.

LIN-UART

The Local Interconnect Network Universal Asynchronous Receiver/Transmitter (LIN-UART) is a full-duplex communication channel capable of handling asynchronous data transfers in standard UART applications as well as providing LIN protocol support.

Features of the LIN-UART include:

- 8-bit asynchronous data transfer
- Selectable even and odd-parity generation and checking
- Option of one or two stop bits
- Selectable MULTIPROCESSOR (9-Bit) Mode with three configurable interrupt schemes
- Separate transmit and receive interrupts or DMA requests
- Framing, parity, overrun and break detection
- 16-bit Baud Rate Generator (BRG), which functions as a general-purpose timer with interrupt
- Driver enable output for external bus transceivers
- LIN protocol support for both Master and Slave modes:
  - Break generation and detection
  - Selectable slave autobaud
  - Check Tx versus Rx data when sending
- Configurable digital noise filter on receive data line

Architecture

The LIN-UART consists of three primary functional blocks: transmitter, receiver and BRG. The LIN-UART's transmitter and receiver function independently but use the same baud rate and data format. The basic UART operation is enhanced by the noise filter and IrDA blocks. Figure 24 displays the LIN-UART architecture.

2. The Master initiates a transfer, sending the first address byte. The Slave Mode I^2C Controller recognizes the start of a 10-bit address with a match to SLA[9:8] and detects the R/$\overline{W}$ bit = 0 (write from Master to Slave). The I^2C Controller acknowledges, indicating that it is available to accept the transaction.
3. The Master sends the second address byte. The Slave Mode I^2C Controller compares the second address byte with the value in SLA[7:0]. If there is a match, the SAM bit in the I2CISTAT Register is set = 1, causing a Slave Address Match interrupt. The RD bit is set = 0, indicating a write to the Slave. If a match occurs, the I^2C Controller acknowledges on the I^2C bus, indicating that it is available to accept the data.
4. Software responds to the Slave Address Match interrupt by reading the I2CISTAT Register which clears the SAM bit. When the RD bit = 0, no further action is required.
5. The Master notifies the Acknowledge and sends a Restart instruction, followed by the first address byte with the R/$\overline{W}$ = 1. The Slave Mode I^2C Controller recognizes the Restart followed by the first address byte with a match to SLA[9:8] and detects the R/$\overline{W}$ = 1 (Master reads from Slave). The Slave I^2C Controller sets the SAM bit in the I2CISTAT Register, which causes the Slave Address Match interrupt. The RD bit is set = 1. The Slave Mode I^2C Controller acknowledges on the bus.
6. Software responds to the interrupt by reading the I2CISTAT Register, clearing the SAM bit. Software loads the initial data byte into the I2CDATA Register and sets the TXI bit in the I2CCTL Register.
7. The Master starts the data transfer by asserting SCL Low. After the I^2C Controller has data available to transmit the SCL is released and the Master proceeds to shift the first data byte.
8. When the first bit of the first data byte is transferred, the I^2C controller sets the TDRE bit, which asserts the transmit data interrupt.
9. Software responds to the transmit data interrupt by loading the next data byte into the I2CDATA Register.
10. The I^2C Master shifts in the remainder of the data byte. The Master transmits the Acknowledge (or Not Acknowledge for the last data byte).
11. The bus cycles through steps 7–10 until the last byte has been transferred. If software has not yet loaded the next data byte when the Master brings SCL Low to transfer the most significant data bit, the Slave I^2C Controller holds SCL Low until the data register is written. When the Slave receives a Not Acknowledge, the I^2C Controller sets the NCKI bit in the I2CISTAT Register and generates the NAK interrupt.
12. Software responds to the NAK interrupt by clearing the TXI bit in the I2CCTL Register and by asserting the FLUSH bit of the I2CCTL Register.
13. When the Master has completed the acknowledge cycle of the last transfer it asserts the Stop or Restart condition on the bus.

Table 119. I2CSTATE_H (Continued)

State Encoding	State Name	State Description
0111	Master Wait	Master received a Not Acknowledge instruction, waiting for software to assert Stop or Start control bits.
1000	Slave Transmit Data	Nine substates, one for each data bit and one for the acknowledge.
1001	Slave Receive Data	Nine substates, one for each data bit and one for the acknowledge.
1010	Slave Receive Addr1	Slave Receiving first address byte (7 and 10 bit addressing) Nine substates, one for each address bit and one for the acknowledge.
1011	Slave Receive Addr2	Slave Receiving second address byte (10 bit addressing) Nine substates, one for each address bit and one for the acknowledge.
1100	Master Transmit Data	Nine substates, one for each data bit and one for the acknowledge.
1101	Master Receive Data	Nine substates, one for each data bit and one for the acknowledge.
1110	Master Transmit Addr1	Master sending first address byte (7- and 10-Bit Addressing) Nine substates, one for each address bit and one for the acknowledge.
1111	Master Transmit Addr2	Master sending second address byte (10-Bit Addressing) Nine substates, one for each address bit and one for the acknowledge.

Table 120. I2CSTATE_L

State I2CSTATE_H	Sub-State I2CSTATE_L	Sub-State Name	State Description
0000–0100	0000	—	There are no substates for these I2CSTATE_H values.
0110–0111	0000	—	There are no substates for these I2CSTATE_H values.

Bit	Description (Continued)
[4] CPISEL	**Comparator Input Select** 0 = PortB6 provides the comparator - input. 1 = PortC0 provides the comparator - input.
[3] CMPIRQ	**Comparator Interrupt Edge Select** 0 = Interrupt Request on Comparator Rising Edge. 1 = Interrupt Request on Comparator Falling Edge.
[2] CMPIV	**PWM Fault Comparator Polarity** 0 = PWM Fault is active when cp+ > cp- 1 = PWM Fault is active when cp- > cp+
[1] CMPOUT	**Comparator Output Value** 0 = Comparator output is logical 0. 1 = Comparator output is logical 1.
[0] CMPEN	**Comparator Enable** 0 = Comparator is disabled. 1 = Comparator is enabled.

Flash Status Register

The Flash Status Register, shown in Table 140, indicates the current state of the Flash Controller. This register is read at any time. The Read-only Flash Status Register shares its address with the Write-only Flash Command Register.

Table 140. Flash Status Register (FSTAT)

Bits	7	6	5	4	3	2	1	0
Field	UNLOCK	Reserved	FSTAT					
RESET	0	0	00h					
R/W	R	R	R					
Addr	FF_E060h							

Bit	Description
[7] UNLOCK	**Unlocked** This status bit is set when the Flash Controller is unlocked. 0 = Flash Controller locked. 1 = Flash Controller unlocked.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5:0] FSTAT	**Flash Controller Status** 00_0000 = Flash Controller idle. 00_1xxx = Program operation in progress. 01_0xxx = Page erase operation in progress. 10_0xxx = Mass erase operation in progress.

Bit	Description (Continued)
[3:0] REQSEL	**DMA Request Selection by Channel**
DMA0	**DMA 0 Request Select** 0000 = Continuous (i.e., Memory to Memory) 0001 = Timer 0 0010 = Timer 1 0011 = Timer 2 0100 = UART0 RXD 0101 = UART0 TXD 0110 = UART1 RXD 0111 = UART1 TXD 1000 = I2C RX 1001 = I2C TX 1010 = SPI RX 1011 = SPI TX 1100 = ADC0 1101 = Reserved 1110 = Reserved 1111 = $\overline{\text{DMA0REQ}}$ Pin
DMA1	**DMA 1 Request Select** 0000 = Continuous (i.e., Memory to Memory) 0001 = Timer 0 0010 = Timer 1 0011 = Timer 2 0100 = UART0 RXD 0101 = UART0 TXD 0110 = UART1 RXD 0111 = UART1 TXD 1000 = I2C RX 1001 = I2C TX 1010 = SPI RX 1011 = SPI TX 1100 = ADC0 1101 = Reserved 1110 = Reserved 1111 = $\overline{\text{DMA1REQ}}$ Pin

DMA List Address Register

The DMA List Address Register is written when the list mode for the DMA is used. This register contains the address of the current list the DMA is operating on. Writing the DMAxLARL Register (shown in Table 160) enables the DMA for list operation.

Table 158. DMA X List Address Register Upper DMAxLARU

Bits	7	6	5	4	3	2	1	0
Field	DMAxLARU							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFE41Dh, FFE42Dh, FFE43Dh, FFE44Dh							

In DIRECT Mode, this register is used to set a watermark interrupt. This interrupt occurs when the DMATXLN[15:8] equals 0 and DMAxTXLN[7:0] equals DMAxLARU. Note when using the watermark the DMAxLARL must not be written.

Table 159. DMA X List Address Register High (DMAxLARH)

Bits	7	6	5	4	3	2	1	0
Field	DMAxLARH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFE41Eh, FFE42Eh, FFE43Eh, FFE44Eh							

Table 160. DMA X List Address Register Low (DMAxLARL)

Bits	7	6	5	4	3	2	1	0
Field	DMAxLARL							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFE41Fh, FFE42Fh, FFE43Fh, FFE44Fh							

Writing to the DMAxLARL Register causes the DMA to enter LINKED LIST Mode.

Program Memory Address 0002h

Option Bits in this space are altered to change the chip configuration at reset.

Table 163. Options Bits at Program Memory Address 0002h

Bits	7	6	5	4	3	2	1	0
Field	Reserved							
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	Program Memory 0002h							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7:0]	**Reserved** These option bits are reserved for future use and must always be 1. This setting is the default for unprogrammed (erased) Flash.

Program Memory Address 0003h

Option bits in this space are altered to change the chip configuration at reset.

Table 164. Options Bits at Program Memory Address 0003h

Bits	7	6	5	4	3	2	1	0
Field	ROMLESS 16	LPOPT	Reserved					
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	Program Memory 0003h							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7] ROMLESS 16	**ROMLESS 16 Select** 0 = If the device is ROMLESS, the data bus is 8 bits wide and is on Port E[7:0]. 1 = If the device is ROMLESS, the data bus is 16 bits wide and is on {Port J[7:0], Port E[7:0]}

```
DBG <<- regdata[23:16]
DBG <<- regdata[15:8]
DBG <<- regdata[7:0]
DBG --> CRC[0:7]
```

Read PC. The Read Program Counter command returns the contents of the program counter.

```
DBG <-- 0000_0110
DBG --> 00h
DBG --> PC[23:16]
DBG --> PC[15:8]
DBG --> PC[7:0]
DBG --> CRC[0:7]
```

Write PC. The Write Program Counter command writes data to the program counter.

```
DBG <-- 0000_0111
DBG <-- 00h
DBG <-- PC[23:16]
DBG <-- PC[15:8]
DBG <-- PC[7:0]
DBG --> CRC[0:7]
```

Read Flags. The Read Flags command returns the contents of the CPU flags.

```
DBG <-- 0000_1000
DBG --> 00h
DBG --> flags[7:0]
DBG --> CRC[0:7]
```

Write Instruction. The Write Instruction command writes one word of Op Code to the CPU.

```
DBG <-- 0000_1001
DBG <-- opcode[15:8]
DBG <-- opcode[7:0]
DBG --> CRC[0:7]
```

Read Register. The Read Register command returns the contents of a single CPU register.

```
DBG <-- {0100,regno[3:0]}
DBG --> regdata[31:24]
DBG --> regdata[23:16]
DBG --> regdata[15:8]
DBG --> regdata[7:0]
DBG --> CRC[0:7]
```

Write Register. The Write Register command writes data to a single CPU register.

```
DBG <-- {0101,regno[3:0]}
DBG <-- regdata[31:24]
DBG <-- regdata[23:16]
DBG <-- regdata[15:8]
DBG <-- regdata[7:0]
DBG --> CRC[0:7]
```

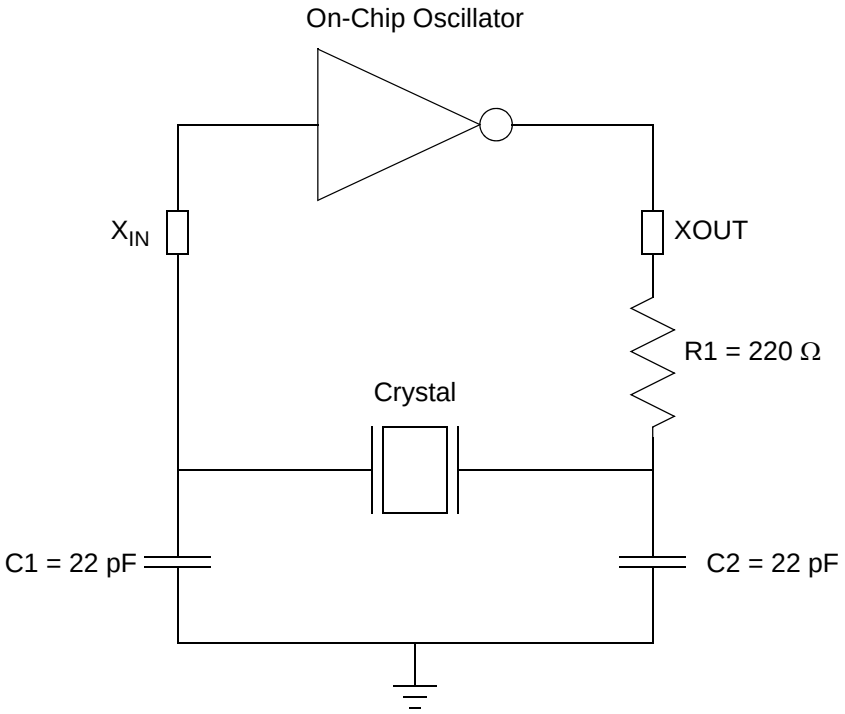

Figure 70. Recommended 20 MHz Crystal Oscillator Configuration

Table 181. Recommended Crystal Oscillator Specifications (20 MHz Operation)

Parameter	Value	Units	Comments
Frequency	20	MHz	
Resonance	Parallel		
Mode	Fundamental		
Series Resistance (R_S)	25	Ω	Maximum
Load Capacitance (C_L)	20	pF	Maximum
Shunt Capacitance (C_0)	7	pF	Maximum
Drive Level	1	mW	Maximum

Internal Precision Oscillator

The Internal Precision Oscillator (IPO) is designed for use without external components. Nominal untrimmed accuracy is approximately ±30%. You can manually trim the oscillator to achieve a ±4% frequency accuracy over the operating temperature and supply voltage range of the device.

The IPO features include:

- On-chip RC oscillator which does not require external components
- Nominal ±30% accuracy without trim or manually trim the oscillator to achieve a ± 4%
- Typical output frequency of 5.5296 MHz
- Trimming possible through Flash option bits with user override
- Eliminates crystals or ceramic resonators in applications where high timing accuracy is not required

Operation

The internal oscillator is an RC relaxation oscillator and has its sensitivity to power supply variation minimized. By using ratio tracking thresholds, the effect of power supply voltage is cancelled out. The dominant source of oscillator error is the absolute variance of chip level fabricated components, such as capacitors. An 8-bit trimming register, incorporated into the design, allows compensation of absolute variation of oscillator frequency. After it is calibrated, the oscillator frequency is relatively stable and does not require subsequent calibration.

By default, the oscillator is configured through the Flash Option bits. However, the user code overrides these trim values as described in the Option Bits chapter on page 292.

Table 186. DC Characteristics (Continued)

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
I_{TL}	Tri-State Leakage Current	–5	—	+5	µA	V_{DD} = 3.6 V
C_{PAD}	GPIO Port Pad Capacitance	—	8.0[2]	—	pF	
C_{XIN}	X_{IN} Pad Capacitance	—	8.0[2]	—	pF	
C_{XOUT}	X_{OUT} Pad Capacitance	—	9.5[2]	—	pF	
I_{PU}	Weak Pull-up Current	30	100	350	µA	V_{DD} = 2.7 V to 3.6 V
I_{CCS1}	Supply Current in Stop Mode with VBO enabled		600		µA	V_{DD} = 3.0 V; 25°C
I_{CCS2}	Supply Current in Stop Mode with VBO disabled		2		µA	V_{DD} = 3.0 V; 25°C
I_{CCS3}	Supply Current in Stop Mode with VBO disabled and WDT disabled		1		µA	V_{DD} = 3.0 V; 25°C
I_{CCA}	Active I_{DD} at 20MHz	—	18	35	mA	Typ: V_{DD}=3.0 V/30°C Max: V_{DD}=3.6 V/125°C Peripherals enabled, no loads
I_{CCH}	I_{DD} in Halt Mode at 20MHz	—	4	6	mA	Typ: V_{DD}=3.0 V/30°C Max: V_{DD}=3.6 V/125°C Peripherals off, no loads

Note:
1. This condition excludes all pins that have on-chip pull-ups enabled, when driven Low.

Table 192 provides electrical characteristics and timing information for the on-chip comparator.

Table 192. Comparator Electrical Characteristics

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
V_{COFF}	Input offset	—	5		mV	V_{DD} = 3.3 V; V_{IN} = V_{DD} ÷ 2
T_{CPROP}	Propagation delay	—	200		ns	V_{COMM} mode = 1 V V_{DIFF} = 100 mV
I_B	Input bias current			1	µA	
CMVR	Common-mode voltage range	–0.3		V_{DD}–1	V	
I_{CC}	Supply current		40		µA	V_{DD} = 3.6 V
T_{wup}	Wake up time from off state			5	µs	CINP = 0.9 V CINN= 1.0 V

Table 193 provides electrical characteristics and timing information for the on-chip operational amplifier.

Table 193. Operational Amplifier Electrical Characteristics

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
V_{OS}	Input offset		5	15	mV	V_{DD} =3.3V; V_{CM} = V_{DD} ÷ 2
TC_{VOS}	Input offset Average Drift		1		µV/C	
I_B	Input bias current		TBD		µA	
I_{OS}	Input offset current		TBD		µA	
CMVR	Common-Mode Voltage Range	–0.3		V_{DD} – 1	V	
V_{OL}	Output Low			0.1	V	I_{SINK} = 100 µA
V_{OH}	Output High	V_{DD} – 1			V	I_{SOURCE} = 100 µA
CMRR	Common-Mode Rejection Ratio		70		dB	0 < V_{CM} < 1.4V; T_A = 25ºC

Table 193. Operational Amplifier Electrical Characteristics (Continued)

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
PSRR	Power Supply Rejection Ratio		80		dB	V_{DD} = 2.7 V – 3.6 V; T_A = 25 ºC
A_{VOL}	Voltage Gain		80		dB	
SR+	Slew Rate while rising		12		V/µs	R_{LOAD} = 33 K; C_{LOAD} = 50 pF; A_{VCL} = 1, V_{IN} = 0.7 V to 1.7 V
SR-	Slew Rate while falling		16		V/µs	R_{LOAD} = 33 K; C_{LOAD} = 50 pF; A_{VCL} = 1, V_{IN} = 1.7 V to 0.7 V
GBW	Gain-Bandwidth Product	5			MHz	
FM	Phase Margin		50		degree	
I_S	Supply Current			1	mA	V_{DD} = 3.6 V; V_{OUT}= V_{DD} ÷ 2
T_{WUP}	Wake up time from off state			20	µs	