



Welcome to [E-XFL.COM](https://www.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	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6421an020ec00tr

I²C

The I²C controller makes the Z8 Encore! XP compatible with the I²C protocol. The I²C controller consists of two bidirectional bus lines, a serial data (SDA) line and a serial clock (SCL) line.

Serial Peripheral Interface

The serial peripheral interface allows the Z8 Encore! XP to exchange data between other peripheral devices such as EEPROMs, A/D converters and ISDN devices. The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface.

Timers

Up to four 16-bit reloadable timers can be used for timing/counting events or for motor control operations. These timers provide a 16-bit programmable reload counter and operate in One-Shot, Continuous, Gated, Capture, Compare, Capture and Compare, and PWM modes. Only 3 timers (Timers 0-2) are available in the 44-pin packages.

Interrupt Controller

The 64K Series products support up to 24 interrupts. These interrupts consist of 12 internal and 12 GPIO pins. The interrupts have 3 levels of programmable interrupt priority.

Reset Controller

The Z8 Encore! can be reset using the $\overline{\text{RESET}}$ pin, Power-On Reset, Watchdog Timer, STOP mode exit, or Voltage Brownout (VBO) warning signal.

On-Chip Debugger

The Z8 Encore! XP features an integrated On-Chip Debugger. The OCD provides a rich set of debugging capabilities, such as reading and writing registers, programming the Flash, setting breakpoints and executing code. A single-pin interface provides communication to the OCD.

DMA Controller

The 64K Series features three channels of DMA. Two of the channels are for register RAM to and from I/O operations. The third channel automatically controls the transfer of data from the ADC to the memory.

SPI Data

SPIDATA (F60H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

SPI Data [7:0]

SPI Control

SPICTL (F61H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- SPI Enable
0 = SPI disabled
1 = SPI enabled
- Master Mode Enabled
0 = SPI configured in Slave mode
1 = SPI configured in Master mode
- Wire-OR (open-drain) Mode
0 = SPI signals not configured for open-drain
1 = SPI signals (SCK, \overline{SS} , MISO, and MOSI) configured for open-drain
- Clock Polarity
0 = SCK idles Low
1 = SPI idles High
- Phase Select
Sets the phase relationship of the data to the clock.
- BRG Timer Interrupt Request
0 = BRG timer function is disabled
1 = BRG time-out interrupt is enabled
- Start an SPI Interrupt Request
0 = No effect
1 = Generate an SPI interrupt request
- Interrupt Request Enable
0 = SPI interrupt requests are disabled
1 = SPI interrupt requests are enabled

SPI Status

SPISTAT (F62H - Read Only)

D7 D6 D5 D4 D3 D2 D1 D0

- Slave Select
0 = If Slave, \overline{SS} pin is asserted
1 = If Slave, \overline{SS} pin is not asserted
- Transmit Status
0 = No data transmission in progress
1 = Data transmission now in progress
- Reserved
- Slave Mode Transaction Abort
0 = No slave mode transaction abort detected
1 = Slave mode transaction abort was detected
- Collision
0 = No multi-master collision detected
1 = Multi-master collision was detected
- Overrun
0 = No overrun error detected
1 = Overrun error was detected
- Interrupt Request
0 = No SPI interrupt request pending
1 = SPI interrupt request is pending

SPI Mode

SPIMODE (F63H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- Slave Select Value
If Master and SPIMODE[1] = 1:
0 = \overline{SS} pin driven Low
1 = \overline{SS} pin driven High
- Slave Select I/O
0 = \overline{SS} pin configured as an input
1 = \overline{SS} pin configured as an output (Master mode only)
- Number of Data Bits Per Character
000 = 8 bits
001 = 1 bit
010 = 2 bits
011 = 3 bits
100 = 4 bits
101 = 5 bit
110 = 6 bits

- Select either the rising edge or falling edge of the Timer Input signal for the count. This also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function does not have to be enabled
- 2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in COUNTER mode. After the first timer Reload in COUNTER mode, counting always begins at the reset value of 0001H. Generally, in COUNTER mode the Timer High and Low Byte registers must be written with the value 0001H.
- 3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
- 4. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 5. Configure the associated GPIO port pin for the Timer Input alternate function.
- 6. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
- 7. Write to the Timer Control 1 register to enable the timer.

In COUNTER mode, the number of Timer Input transitions since the timer start is given by the following equation:

$$\text{COUNTER Mode Timer Input Transitions} = \text{Current Count Value} - \text{Start Value}$$

PWM Mode

In PWM mode, the timer outputs a Pulse-Width Modulator (PWM) output signal through a GPIO Port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

If the TPOL bit in the Timer Control 1 register is set to 1, the Timer Output signal begins as a High (1) and then transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the Reload value and is reset to 0001H.

If the TPOL bit in the Timer Control 1 register is set to 0, the Timer Output signal begins as a Low (0) and then transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the Reload value and is reset to 0001H.

POR—Power-On Reset Indicator

If this bit is set to 1, a Power-On Reset event occurred. This bit is reset to 0 if a WDT time-out or Stop Mode Recovery occurs. This bit is also reset to 0 when the register is read.

STOP—Stop Mode Recovery Indicator

If this bit is set to 1, a Stop Mode Recovery occurred. If the STOP and WDT bits are both set to 1, the Stop Mode Recovery occurred due to a WDT time-out. If the STOP bit is 1 and the WDT bit is 0, the Stop Mode Recovery was not caused by a WDT time-out. This bit is reset by a Power-On Reset or a WDT time-out that occurred while not in STOP mode. Reading this register also resets this bit.

WDT—Watchdog Timer Time-Out Indicator

If this bit is set to 1, a WDT time-out occurred. A Power-On Reset resets this pin. A Stop Mode Recovery from a change in an input pin also resets this bit. Reading this register resets this bit.

EXT—External Reset Indicator

If this bit is set to 1, a Reset initiated by the external $\overline{\text{RESET}}$ pin occurred. A Power-On Reset or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit.

Reserved

These bits are reserved and must be 0.

SM—STOP Mode Configuration Indicator

0 = Watchdog Timer and its internal RC oscillator will continue to operate in STOP Mode.
1 = Watchdog Timer and its internal RC oscillator will be disabled in STOP Mode.

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTL, WDTL) registers (see [Table 49](#) on page 102 through [Table 51](#) on page 102) form the 24-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes. The 24-bit reload value is {WDTU[7:0], WDTL[7:0], WDTL[7:0]}. Writing to these registers sets the desired Reload Value. Reading from these registers returns the current Watchdog Timer count value.

REN—Receive Enable

This bit enables or disables the receiver.

0 = Receiver disabled.

1 = Receiver enabled.

CTSE—CTS Enable

0 = The $\overline{\text{CTS}}$ signal has no effect on the transmitter.

1 = The UART recognizes the $\overline{\text{CTS}}$ signal as an enable control from the transmitter.

PEN—Parity Enable

This bit enables or disables parity. Even or odd is determined by the PSEL bit. It is overridden by the MPEN bit.

0 = Parity is disabled.

1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit.

PSEL—Parity Select

0 = Even parity is transmitted and expected on all received data.

1 = Odd parity is transmitted and expected on all received data.

SBRK—Send Break

This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so ensure that the transmitter has finished sending data before setting this bit.

0 = No break is sent.

1 = The output of the transmitter is zero.

STOP—Stop Bit Select

0 = The transmitter sends one stop bit.

1 = The transmitter sends two stop bits.

LBEN—Loop Back Enable

0 = Normal operation.

1 = All transmitted data is looped back to the receiver.

Table 57. UART Control 1 Register (UxCTL1)

BITS	7	6	5	4	3	2	1	0
FIELD	MPMD[1]	MPEN	MPMD[0]	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
RESET	0							
R/W	R/W							
ADDR	F43H and F4BH							

MPMD[1:0]—MULTIPROCESSOR Mode

If MULTIPROCESSOR (9-bit) mode is enabled,

00 = The UART generates an interrupt request on all received bytes (data and address).

Operation

When the Infrared Endec is enabled, the transmit data from the associated on-chip UART is encoded as digital signals in accordance with the IrDA standard and output to the infrared transceiver via the TXD pin. Likewise, data received from the infrared transceiver is passed to the Infrared Endec via the RXD pin, decoded by the Infrared Endec, and then passed to the UART. Communication is half-duplex, which means simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART's Baud Rate Generator and supports IrDA standard baud rates from 9600 baud to 115.2 Kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the Infrared Endec. The Infrared Endec data rate is calculated using the following equation:

$$\text{Infrared Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR_TXD) that drives the infrared transceiver. Each UART/Infrared data bit is 16-clock wide. If the data to be transmitted is 1, the IR_TXD signal remains low for the full 16-clock period. If the data to be transmitted is 0, a 3-clock high pulse is output following a 7-clock low period. After the 3-clock high pulse, a 6-clock low pulse is output to complete the full 16-clock data period. Figure 20 displays IrDA data transmission.

When the Infrared Endec is enabled, the UART's TXD signal is internal to the 64K Series products while the IR_TXD signal is output through the TXD pin.

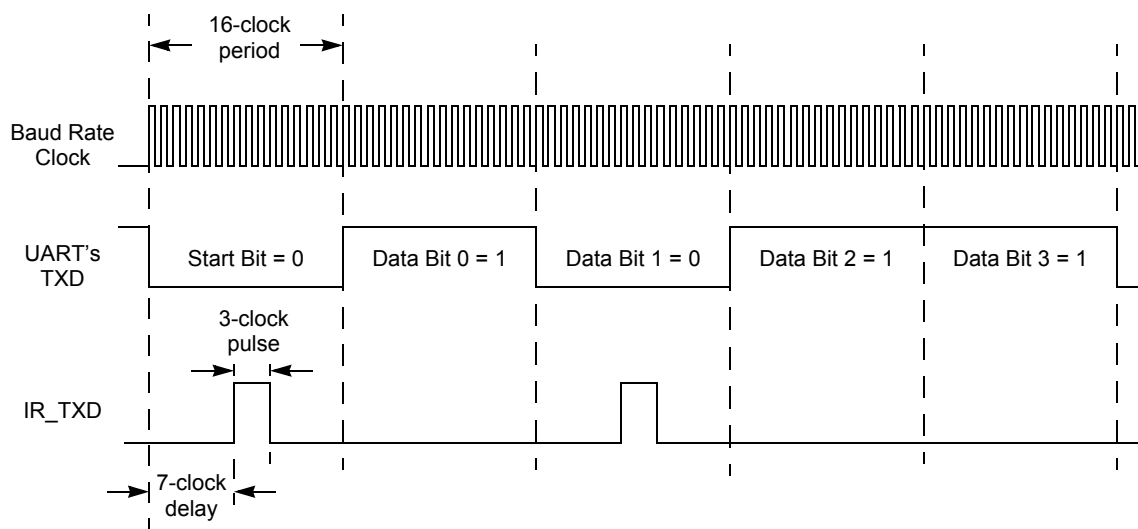


Figure 20. Infrared Data Transmission

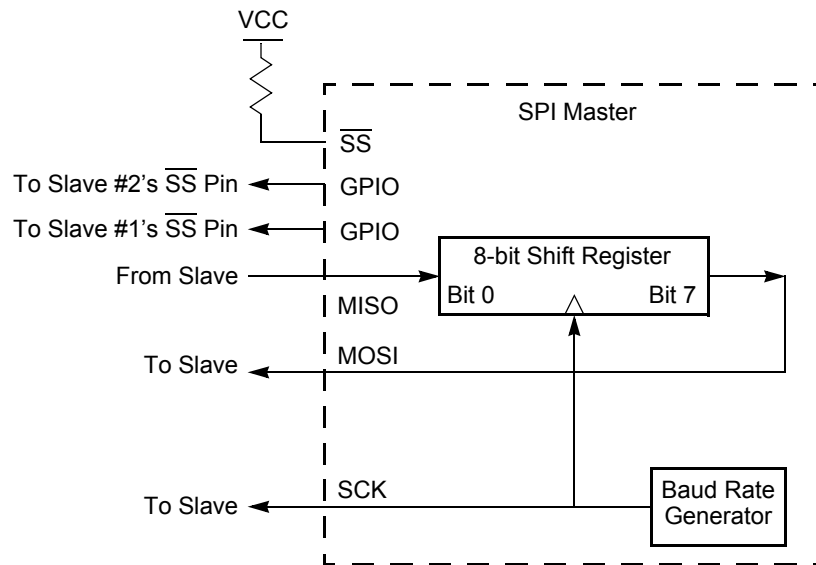


Figure 23. SPI Configured as a Master in a Single Master, Multiple Slave System

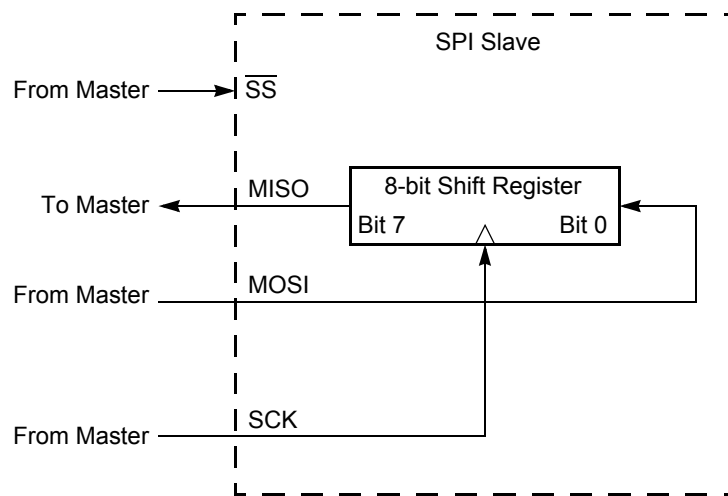


Figure 24. SPI Configured as a Slave

Operation

The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface (serial clock, transmit, receive and Slave select). The SPI block consists of a transmit/receive shift register, a Baud Rate (clock) Generator and a control unit.

Configuring DMA0 and DMA1 for Data Transfer

Follow the steps below to configure and enable DMA0 or DMA1:

1. Write to the DMA_x I/O Address register to set the Register File address identifying the on-chip peripheral control register. The upper nibble of the 12-bit address for on-chip peripheral control registers is always F_H. The full address is {F_H, DMA_x_IO[7:0]}.
2. Determine the 12-bit Start and End Register File addresses. The 12-bit Start Address is given by {DMA_x_H[3:0], DMA_START[7:0]}. The 12-bit End Address is given by {DMA_x_H[7:4], DMA_END[7:0]}.
3. Write the Start and End Register File address high nibbles to the DMA_x End/Start Address High Nibble register.
4. Write the lower byte of the Start Address to the DMA_x Start/Current Address register.
5. Write the lower byte of the End Address to the DMA_x End Address register.
6. Write to the DMA_x Control register to complete the following:
 - Select loop or single-pass mode operation
 - Select the data transfer direction (either from the Register File RAM to the on-chip peripheral control register; or from the on-chip peripheral control register to the Register File RAM)
 - Enable the DMA_x interrupt request, if desired
 - Select Word or Byte mode
 - Select the DMA_x request trigger
 - Enable the DMA_x channel

DMA_ADC Operation

DMA_ADC transfers data from the ADC to the Register File. The sequence of operations in a DMA_ADC data transfer is:

1. ADC completes conversion on the current ADC input channel and signals the DMA controller that two-bytes of ADC data are ready for transfer.
2. DMA_ADC requests control of the system bus (address and data) from the eZ8 CPU.
3. After the eZ8 CPU acknowledges the bus request, DMA_ADC transfers the two-byte ADC output value to the Register File and then returns system bus control back to the eZ8 CPU.
4. If the current ADC Analog Input is the highest numbered input to be converted:
 - DMA_ADC resets the ADC Analog Input number to 0 and initiates data conversion on ADC Analog Input 0.
 - If configured to generate an interrupt, DMA_ADC sends an interrupt request to the Interrupt Controller

Follow the steps below for setting up the ADC and initiating continuous conversion:

1. Enable the desired analog input by configuring the general-purpose I/O pins for alternate function. This disables the digital input and output driver.
2. Write to the ADC Control register to configure the ADC for continuous conversion. The bit fields in the ADC Control register may be written simultaneously:
 - Write to the ANAIN[3 : 0] field to select one of the 12 analog input sources.
 - Set CONT to 1 to select continuous conversion.
 - Write to the VREF bit to enable or disable the internal voltage reference generator.
 - Set CEN to 1 to start the conversions.
3. When the first conversion in continuous operation is complete (after 5129 system clock cycles, plus the 40 cycles for power-up, if necessary), the ADC control logic performs the following operations:
 - CEN resets to 0 to indicate the first conversion is complete. CEN remains 0 for all subsequent conversions in continuous operation.
 - An interrupt request is sent to the Interrupt Controller to indicate the conversion is complete.
4. Thereafter, the ADC writes a new 10-bit data result to {ADCD_H[7:0], ADCD_L[7:6]} every 256 system clock cycles. An interrupt request is sent to the Interrupt Controller when each conversion is complete.
5. To disable continuous conversion, clear the CONT bit in the ADC Control register to 0.

DMA Control of the ADC

The Direct Memory Access (DMA) Controller can control operation of the ADC including analog input selection and conversion enable. For more information on the DMA and configuring for ADC operations, see [Direct Memory Access Controller](#) on page 165.



Operation

OCD Interface

The On-Chip Debugger uses the DBG pin for communication with an external host. This one-pin interface is a bi-directional open-drain interface that transmits and receives data. Data transmission is half-duplex, in that transmit and receive cannot occur simultaneously. The serial data on the DBG pin is sent using the standard asynchronous data format defined in RS-232. This pin can interface the 64K Series products to the serial port of a host PC using minimal external hardware. Two different methods for connecting the DBG pin to an RS-232 interface are depicted in [Figure 37](#) and [Figure 38](#) on page 201.



Caution: *For operation of the On-Chip Debugger, **all** power pins (V_{DD} and AV_{DD}) must be supplied with power, and **all** ground pins (V_{SS} and AV_{SS}) must be properly grounded.*
The DBG pin is open-drain and must always be connected to V_{DD} through an external pull-up resistor to ensure proper operation.

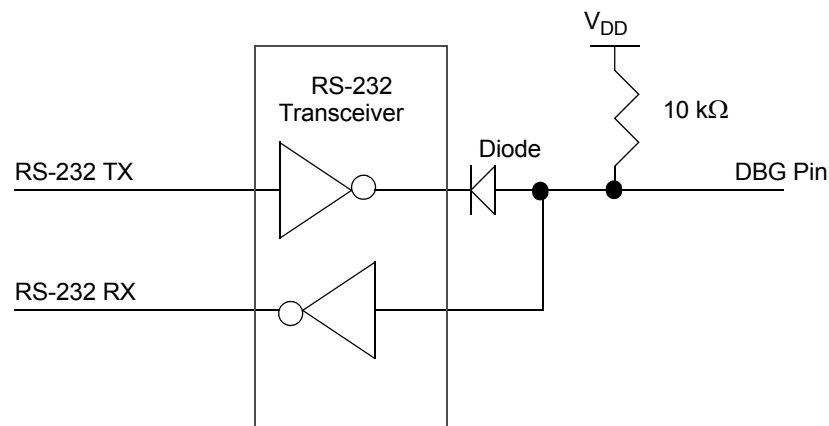


Figure 37. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (1)

- Write OCD Control Register (04H)**—The Write OCD Control Register command writes the data that follows to the OCDCTL register. When the Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of putting the device back into normal operating mode is to reset the device.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```
- Read OCD Control Register (05H)**—The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```
- Write Program Counter (06H)**—The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the Program Counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```
- Read Program Counter (07H)**—The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```
- Write Register (08H)**—The Write Register command writes data to the Register File. Data can be written 1-256 bytes at a time (256 bytes can be written by setting size to zero). If the device is not in DEBUG mode, the address and data values are discarded. If the Read Protect Option Bit is enabled, then only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0, Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```
- Read Register (09H)**—The Read Register command reads data from the Register File. Data can be read 1-256 bytes at a time (256 bytes can be read by setting size to zero). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns FFH for all the data values.

```
DBG ← 09H
DBG ← {4'h0, Register Address[11:8]}
DBG ← Register Address[7:0]
```

Oscillator Operation with an External RC Network

The External RC oscillator mode is applicable to timing insensitive applications.

Figure 41 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

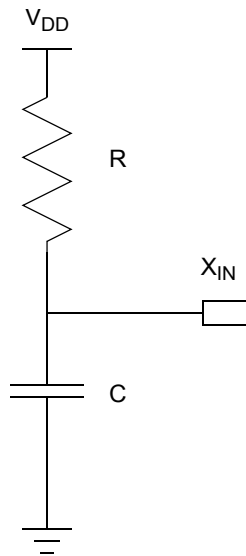


Figure 41. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45 kΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40 kΩ. The typical oscillator frequency can be estimated from the values of the resistor (R in kΩ) and capacitor (C in pF) elements using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1 \times 10^6}{(0.4 \times R \times C) + (4 \times C)}$$

Figure 42 displays the typical (3.3 V and 25 °C) oscillator frequency as a function of the capacitor (C in pF) employed in the RC network assuming a 45 kΩ external resistor. For very small values of C , the parasitic capacitance of the oscillator XIN pin and the printed circuit board should be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20 pF are recommended.

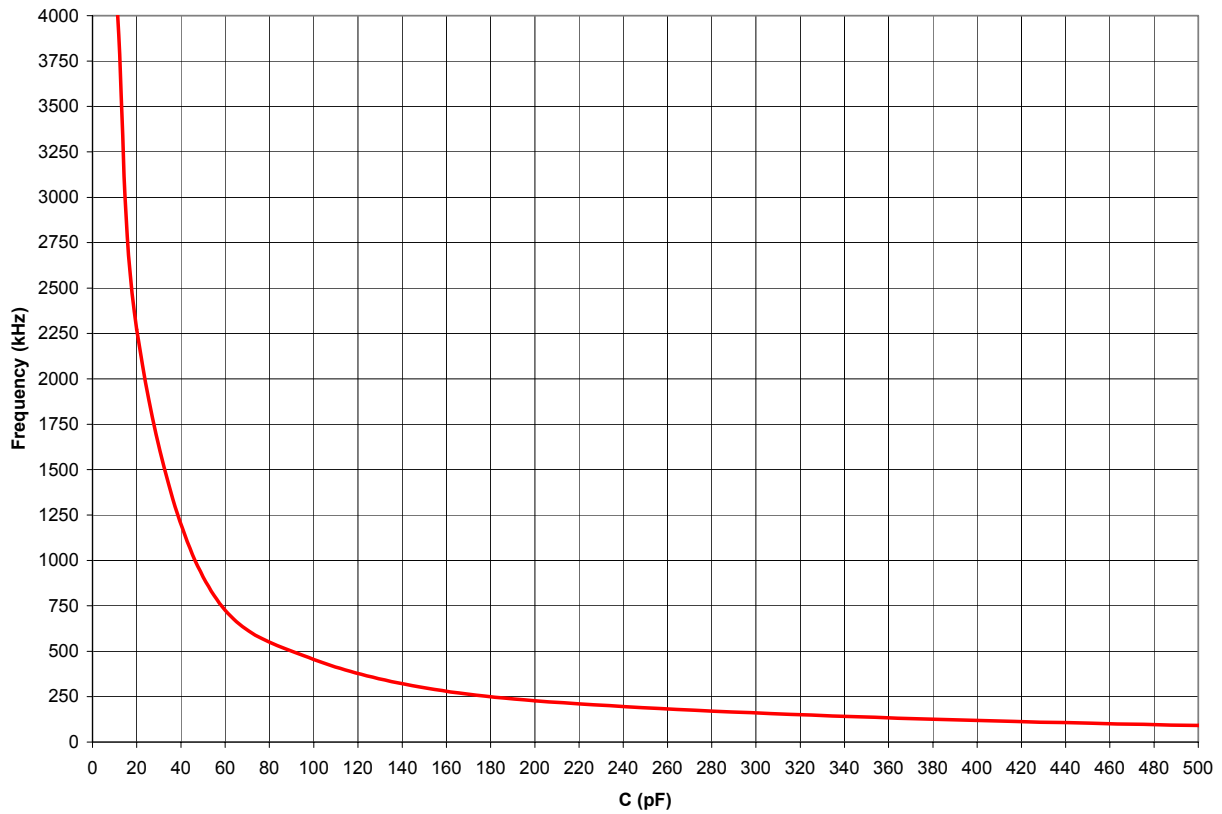


Figure 42. Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45 k Ω Resistor



Caution: *When using the external RC oscillator mode, the oscillator may stop oscillating if the power supply drops below 2.7 V, but before the power supply drops to the voltage brown-out threshold. The oscillator will resume oscillation as soon as the supply voltage exceeds 2.7 V.*

Figure 45 displays the typical current consumption in HALT mode while operating at 25 °C versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

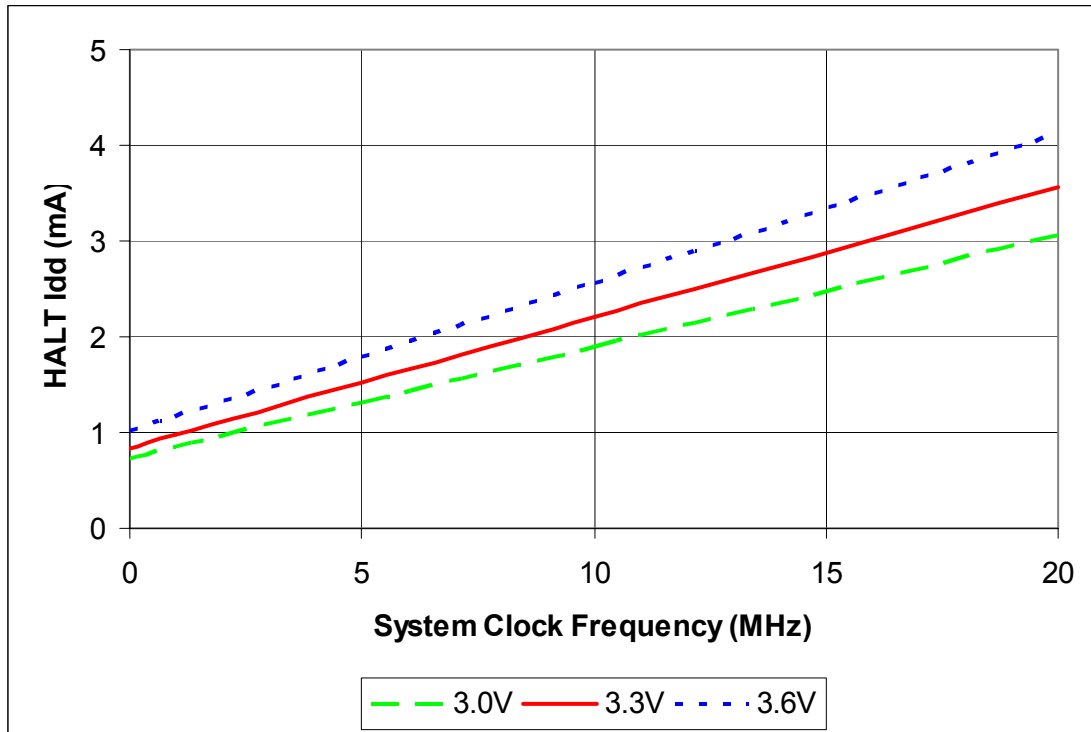


Figure 45. Typical HALT Mode Idd Versus System Clock Frequency

Table 126. Bit Manipulation Instructions (Continued)

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
CCF	—	Complement Carry Flag
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
TCM	dst, src	Test Complement Under Mask
TCMX	dst, src	Test Complement Under Mask using Extended Addressing
TM	dst, src	Test Under Mask
TMX	dst, src	Test Under Mask using Extended Addressing

Table 127. Block Transfer Instructions

Mnemonic	Operands	Instruction
LDCI	dst, src	Load Constant to/from Program Memory and Auto-Increment Addresses
LDEI	dst, src	Load External Data to/from Data Memory and Auto-Increment Addresses

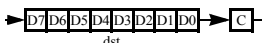
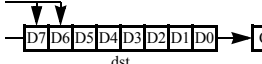

Table 128. CPU Control Instructions

Mnemonic	Operands	Instruction
ATM	—	Atomic Execution
CCF	—	Complement Carry Flag
DI	—	Disable Interrupts
EI	—	Enable Interrupts
HALT	—	HALT Mode
NOP	—	No Operation
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
SRP	src	Set Register Pointer

Table 133. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
EI	IRQCTL[7] ← 1			9F	-	-	-	-	-	-	1	2
HALT	HALT Mode			7F	-	-	-	-	-	-	1	2
INC dst	dst ← dst + 1	R		20	-	*	*	*	-	-	2	2
		IR		21							2	3
		r		0E-FE							1	2
INCW dst	dst ← dst + 1	RR		A0	-	*	*	*	-	-	2	5
		IRR		A1							2	6
IRET	FLAGS ← @SP SP ← SP + 1 PC ← @SP SP ← SP + 2 IRQCTL[7] ← 1			BF	*	*	*	*	*	*	1	5
JP dst	PC ← dst	DA		8D	-	-	-	-	-	-	3	2
		IRR		C4							2	3
JP cc, dst	if cc is true PC ← dst	DA		0D-FD	-	-	-	-	-	-	3	2
JR dst	PC ← PC + X	DA		8B	-	-	-	-	-	-	2	2
JR cc, dst	if cc is true PC ← PC + X	DA		0B-FB	-	-	-	-	-	-	2	2
LD dst, rc	dst ← src	r	IM	0C-FC	-	-	-	-	-	-	2	2
		r	X(r)	C7							3	3
		X(r)	r	D7							3	4
		r	lr	E3							2	3
		R	R	E4							3	2
		R	IR	E5							3	4
		R	IM	E6							3	2
		IR	IM	E7							3	3
		lr	r	F3							2	3
		IR	R	F5							3	3

Table 133. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
RRC dst		R		C0	*	*	*	*	-	-	2	2
		IR		C1							2	3
SBC dst, src	$dst \leftarrow dst - src - C$	r	r	32	*	*	*	*	1	*	2	3
		r	lr	33							2	4
		R	R	34							3	3
		R	IR	35							3	4
		R	IM	36							3	3
		IR	IM	37							3	4
SBCX dst, src	$dst \leftarrow dst - src - C$	ER	ER	38	*	*	*	*	1	*	4	3
		ER	IM	39							4	3
SCF	$C \leftarrow 1$			DF	1	-	-	-	-	-	1	2
SRA dst		R		D0	*	*	*	0	-	-	2	2
		IR		D1							2	3
SRL dst		R		1F C0	*	*	0	*	-	-	3	2
		IR		1F C1							3	3
SRP src	$RP \leftarrow src$		IM	01	-	-	-	-	-	-	2	2
STOP	STOP Mode			6F	-	-	-	-	-	-	1	2
SUB dst, src	$dst \leftarrow dst - src$	r	r	22	*	*	*	*	1	*	2	3
		r	lr	23							2	4
		R	R	24							3	3
		R	IR	25							3	4
		R	IM	26							3	3
		IR	IM	27							3	4
SUBX dst, src	$dst \leftarrow dst - src$	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29							4	3

Example: Part number Z8F6421AN020SC is an 8-bit microcontroller product in an LQFP package, using 44 pins, operating with a maximum 20 MHz external clock frequency over a 0 °C to +70 °C temperature range and built using the Plastic-Standard environmental flow.

- flash page select (FPS) 191
- flash status (FSTAT) 190
- GPIO port A-H address (PxADDR) 61
- GPIO port A-H alternate function sub-registers 63
- GPIO port A-H control address (PxCTL) 62
- GPIO port A-H data direction sub-registers 63
- I2C baud rate high (I2CBRH) 160, 161, 163
- I2C control (I2CCTL) 158
- I2C data (I2CDATA) 157
- I2C status 157
- I2C status (I2CSTAT) 157
- I2Cbaud rate low (I2CBRL) 161
- mode, SPI 140
- OCD control 209
- OCD status 210
- SPI baud rate high byte (SPIBRH) 142
- SPI baud rate low byte (SPIBRL) 142
- SPI control (SPICTL) 138
- SPI data (SPIDATA) 137
- SPI status (SPISTAT) 139
- status, I2C 157
- status, SPI 139
- UARTx baud rate high byte (UxBRH) 121
- UARTx baud rate low byte (UxBRL) 121
- UARTx Control 0 (UxCTL0) 117, 120
- UARTx control 1 (UxCTL1) 118
- UARTx receive data (UxRXD) 115
- UARTx status 0 (UxSTAT0) 115
- UARTx status 1 (UxSTAT1) 117
- UARTx transmit data (UxTXD) 114
- watch-dog timer control (WDTCTL) 100
- watch-dog timer reload high byte (WDTH) 102
- watch-dog timer reload low byte (WDTL) 102
- watch-dog timer reload upper byte (WDTU) 102
- register file 19
- register file address map 23
- register pair 243
- register pointer 244
- reset
 - and STOP mode characteristics 48
 - carry flag 247
 - controller 5
 - sources 48
- RET 249
- return 249
- RL 249
- RLC 249
- rotate and shift instructions 249
- rotate left 249
- rotate left through carry 249
- rotate right 249
- rotate right through carry 249
- RP 244
- RR 243, 249
- rr 243
- RRC 249
- S**
- SBC 246
- SCF 247
- SDA and SCL (IrDA) signals 145
- second opcode map after 1FH 264
- serial clock 131
- serial peripheral interface (SPI) 129
- set carry flag 247
- set register pointer 247
- shift right arithmetic 249
- shift right logical 250
- signal descriptions 14
- single-shot conversion (ADC) 177
- SIO 5
- slave data transfer formats (I2C) 151
- slave select 132
- software trap 249
- source operand 244
- SP 244
- SPI
 - architecture 129
 - baud rate generator 136
 - baud rate high and low byte register 142
 - clock phase 132
 - configured as slave 130
 - control register 137
 - control register definitions 137
 - data register 137