



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	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	6
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-QFN (5x6)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f042aqb020ec

Overview

Zilog's Z8 Encore![®] MCU family of products are the first in a line of Zilog[®] microcontroller products based upon the 8-bit eZ8 CPU. Zilog's Z8 Encore! XP[®] F082A Series products expand upon Zilog's extensive line of 8-bit microcontrollers. The Flash in-circuit programming capability allows for faster development time and program changes in the field. The new eZ8 CPU is upward compatible with existing Z8[®] instructions. The rich peripheral set of the Z8 Encore! XP F082A Series makes it suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Features

The key features of Z8 Encore! XP F082A Series products include:

- 20 MHz eZ8 CPU
- 1 KB, 2 KB, 4 KB, or 8 KB Flash memory with in-circuit programming capability
- 256 B, 512 B, or 1 KB register RAM
- Up to 128 B non-volatile data storage (NVDS)
- Internal precision oscillator trimmed to $\pm 1\%$ accuracy
- External crystal oscillator, operating up to 20 MHz
- Optional 8-channel, 10-bit analog-to-digital converter (ADC)
- Optional on-chip temperature sensor
- On-chip analog comparator
- Optional on-chip low-power operational amplifier (LPO)
- Full-duplex UART
- The UART baud rate generator (BRG) can be configured and used as a basic 16-bit timer
- Infrared Data Association (IrDA)-compliant infrared encoder/decoders, integrated with UART
- Two enhanced 16-bit timers with capture, compare, and PWM capability
- Watchdog Timer (WDT) with dedicated internal RC oscillator
- Up to 20 vectored interrupts
- 6 to 25 I/O pins depending upon package

Address Space

The eZ8 CPU can access the following three distinct address spaces:

1. The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral, and general-purpose I/O port control registers.
2. The Program Memory contains addresses for all memory locations having executable code and/or data.
3. The Data Memory contains addresses for all memory locations that contain data only.

These three address spaces are covered briefly in the following subsections. For more information on eZ8 CPU and its address space, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com.

Register File

The Register File address space in the Z8 Encore![®] MCU is 4 KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as sources are read, and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4 KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256 B control register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The Z8 Encore! XP[®] F082A Series devices contain 256 B to 1 KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

Program Memory

The eZ8 CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP F082A Series devices contain 1 KB to 8 KB of on-chip Flash memory in the Program Memory address space, depending on the device. Reading from Program Memory

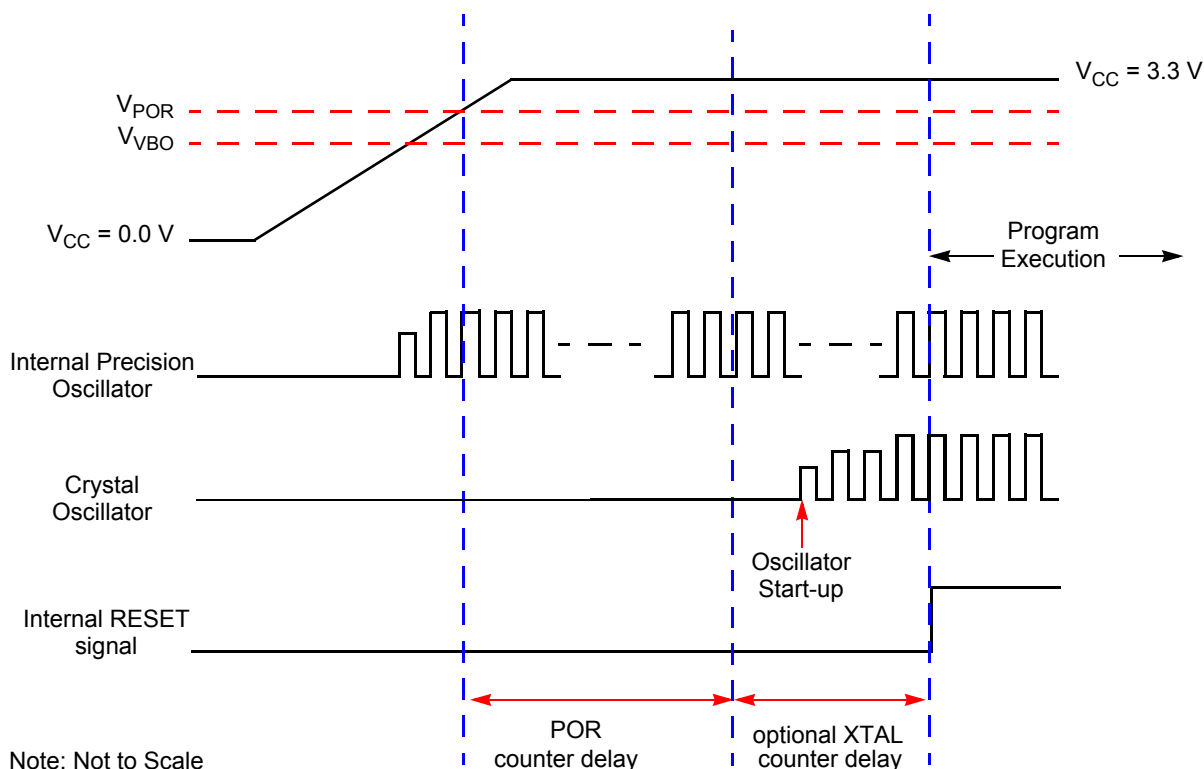


Figure 5. Power-On Reset Operation

Voltage Brownout Reset

The devices in the Z8 Encore! XP F082A Series provide low Voltage Brownout (VBO) protection. The VBO circuit senses when the supply voltage drops to an unsafe level (below the VBO threshold voltage) and forces the device into the Reset state. While the supply voltage remains below the Power-On Reset voltage threshold (V_{POR}), the VBO block holds the device in the Reset.

After the supply voltage again exceeds the Power-On Reset voltage threshold, the device progresses through a full System Reset sequence, as described in the Power-On Reset section. Following Power-On Reset, the POR status bit in the Reset Status (RSTSTAT) register is set to 1. [Figure 6](#) displays Voltage Brownout operation. See [Electrical Characteristics](#) on page 221 for the VBO and POR threshold voltages (V_{VBO} and V_{POR}).

The Voltage Brownout circuit can be either enabled or disabled during STOP mode. Operation during STOP mode is set by the VBO_AO Flash Option Bit. See [Flash Option Bits](#) for information about configuring VBO_AO.

General-Purpose Input/Output

The Z8 Encore! XP[®] F082A Series products support a maximum of 25 port pins (Ports A–D) for general-purpose input/output (GPIO) operations. Each port contains control and data registers. The GPIO control registers determine data direction, open-drain, output drive current, programmable pull-ups, Stop Mode Recovery functionality, and alternate pin functions. Each port pin is individually programmable. In addition, the Port C pins are capable of direct LED drive at programmable drive strengths.

GPIO Port Availability By Device

[Table 13](#) lists the port pins available with each device and package type.

Table 13. Port Availability by Device and Package Type

Devices	Package	ADC	Port A	Port B	Port C	Port D	Total I/O
Z8F082ASB, Z8F082APB, Z8F082AQB Z8F042ASB, Z8F042APB, Z8F042AQB Z8F022ASB, Z8F022APB, Z8F022AQB Z8F012ASB, Z8F012APB, Z8F012AQB	8-pin	Yes	[5:0]	No	No	No	6
Z8F081ASB, Z8F081APB, Z8F081AQB Z8F041ASB, Z8F041APB, Z8F041AQB Z8F021ASB, Z8F021APB, Z8F021AQB Z8F011ASB, Z8F011APB, Z8F011AQB	8-pin	No	[5:0]	No	No	No	6
Z8F082APH, Z8F082AHH, Z8F082ASH Z8F042APH, Z8F042AHH, Z8F042ASH Z8F022APH, Z8F022AHH, Z8F022ASH Z8F012APH, Z8F012AHH, Z8F012ASH	20-pin	Yes	[7:0]	[3:0]	[3:0]	[0]	17
Z8F081APH, Z8F081AHH, Z8F081ASH Z8F041APH, Z8F041AHH, Z8F041ASH Z8F021APH, Z8F021AHH, Z8F021ASH Z8F011APH, Z8F011AHH, Z8F011ASH	20-pin	No	[7:0]	[3:0]	[3:0]	[0]	17
Z8F082APJ, Z8F082ASJ, Z8F082AHJ Z8F042APJ, Z8F042ASJ, Z8F042AHJ Z8F022APJ, Z8F022ASJ, Z8F022AHJ Z8F012APJ, Z8F012ASJ, Z8F012AHJ	28-pin	Yes	[7:0]	[5:0]	[7:0]	[0]	23
Z8F081APJ, Z8F081ASJ, Z8F081AHJ Z8F041APJ, Z8F041ASJ, Z8F041AHJ Z8F021APJ, Z8F021ASJ, Z8F021AHJ Z8F011APJ, Z8F011ASJ, Z8F011AHJ	28-pin	No	[7:0]	[7:0]	[7:0]	[0]	25

PAFS1[7:0]—Port Alternate Function Set 1

0 = Port Alternate Function selected as defined in [Table 14](#) and [Table 15](#) on page 44.

1 = Port Alternate Function selected as defined in [Table 14](#) and [Table 15](#) on page 44.

Port A–D Alternate Function Set 2 Sub-Registers

The Port A–D Alternate Function Set 2 sub-register ([Table 26](#)) is accessed through the Port A–D Control register by writing 08H to the Port A–D Address register. The Alternate Function Set 2 sub-registers selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register is defined in [Table 15](#).

► **Note:** *Alternate function selection on port pins must also be enabled as described in [Port A–D Alternate Function Sub-Registers](#) on page 47.*

Table 26. Port A–D Alternate Function Set 2 Sub-Registers (PxAFS2)

BITS	7	6	5	4	3	2	1	0
FIELD	PAFS27	PAFS26	PAFS25	PAFS24	PAFS23	PAFS22	PAFS21	PAFS20
RESET	00H (all ports of 20/28 pin devices); 04H (Port A of 8-pin device)							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	If 08H in Port A–D Address Register, accessible through the Port A–D Control Register							

PAFS2[7:0]—Port Alternate Function Set 2

0 = Port Alternate Function selected as defined in [Table 15](#).

1 = Port Alternate Function selected as defined in [Table 15](#).

Port A–C Input Data Registers

Reading from the Port A–C Input Data registers ([Table 27](#)) returns the sampled values from the corresponding port pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8- and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

Table 27. Port A–C Input Data Registers (PxIN)

BITS	7	6	5	4	3	2	1	0
FIELD	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	FD2H, FD6H, FDAH							
X = Undefined.								

- Set the prescale value.
 - If using the Timer Output alternate function, set the initial output level (High or Low).
2. Write to the Timer High and Low Byte registers to set the starting count value (usually 0001H). This action only affects the first pass in CONTINUOUS mode. After the first timer Reload in CONTINUOUS mode, counting always begins at the reset value of 0001H.
 3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
 4. Enable the timer interrupt (if appropriate) and set the timer interrupt priority by writing to the relevant interrupt registers.
 5. Configure the associated GPIO port pin (if using the Timer Output function) for the Timer Output alternate function.
 6. Write to the Timer Control register to enable the timer and initiate counting.

In CONTINUOUS mode, the system clock always provides the timer input. The timer period is given by the following equation:

$$\text{CONTINUOUS Mode Time-Out Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT mode equation to determine the first time-out period.

COUNTER Mode

In COUNTER mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO Port pin Timer Input alternate function. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the Timer Input signal. In COUNTER mode, the prescaler is disabled.



Caution: *The input frequency of the Timer Input signal must not exceed one-fourth the system clock frequency. Further, the high or low state of the input signal pulse must be no less than twice the system clock period. A shorter pulse may not be captured.*

Upon reaching the Reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer Reload.

If the TPOL bit in the Timer Control register is set to 0, the Timer Output signal begins as a Low (0) and 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.

Follow the steps below for configuring a timer for PWM SINGLE OUTPUT mode and initiating the PWM operation:

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for PWM SINGLE OUTPUT mode.
 - Set the prescale value.
 - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the Timer Output alternate function.
7. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT mode equation to determine the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

PWM DUAL OUTPUT Mode

In PWM DUAL OUTPUT mode, the timer outputs a Pulse-Width Modulated (PWM) output signal pair (basic PWM signal and its complement) through two GPIO Port pins. 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 register is set to 1, the Timer Output signal begins as a High (1) and 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 register is set to 0, the Timer Output signal begins as a Low (0) and 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.

The timer also generates a second PWM output signal Timer Output Complement. The Timer Output Complement is the complement of the Timer Output PWM signal. A programmable deadband delay can be configured to time delay (0 to 128 system clock cycles) PWM output transitions on these two pins from a low to a high (inactive to active). This ensures a time gap between the deassertion of one PWM output to the assertion of its complement.

Follow the steps below for configuring a timer for PWM DUAL OUTPUT mode and initiating the PWM operation:

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for PWM DUAL OUTPUT mode by writing the TMODE bits in the TxCTL1 register and the TMODEHI bit in TxCTL0 register.
 - Set the prescale value.
 - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the PWM Control register to set the PWM dead band delay value. The deadband delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM high and low byte registers). It must also be less than the

duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).

5. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
8. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the ONE-SHOT mode equation determines the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

CAPTURE Mode

In CAPTURE mode, the current timer count value is recorded when the appropriate external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the timer continues counting. The INPCAP bit in TxCTL0 register is set to indicate the timer interrupt is because of an input capture event.

The timer continues counting up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt and continues counting. The INPCAP bit in TxCTL0 register clears indicating the timer interrupt is not because of an input capture event.

Follow the steps below for configuring a timer for COMPARE mode and initiating the count:

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for COMPARE mode.
 - Set the prescale value.
 - Set the initial logic level (High or Low) for the Timer Output alternate function, if appropriate.
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the Compare value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
6. Write to the Timer Control register to enable the timer and initiate counting.

In COMPARE mode, the system clock always provides the timer input. The Compare time can be calculated by the following equation:

$$\text{COMPARE Mode Time (s)} = \frac{(\text{Compare Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

GATED Mode

In GATED mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is deasserted or a timer reload occurs. To determine if a Timer Input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. When 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 (assuming the Timer Input signal remains asserted). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reset.

Follow the steps below for configuring a timer for GATED mode and initiating the count:

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for GATED mode.
 - Set the prescale value.

4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt are generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL0 register.
5. Configure the associated GPIO port pin for the Timer Input alternate function.
6. Write to the Timer Control register to enable the timer.
7. Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge.

In CAPTURE/COMPARE mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte register is read, the contents of the Timer Low Byte register are placed in a holding register. A subsequent read from the Timer Low Byte register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte register returns the actual value in the counter.

Timer Pin Signal Operation

Timer Output is a GPIO Port pin alternate function. The Timer Output is toggled every time the counter is reloaded.

The Timer Input can be used as a selectable counting source. It shares the same pin as the complementary timer output. When selected by the GPIO Alternate Function Registers, this pin functions as a timer input in all modes except for the DUAL PWM OUTPUT mode. For this mode, there is no timer input available.

Watchdog Timer Reload Registers results in a one-second timeout at room temperature and 3.3 V supply voltage.

Timeouts other than one second may be obtained by scaling the calibration values up or down as required.

► **Note:** *The Watchdog Timer accuracy still degrades as temperature and supply voltage vary. See [Table 133](#) on page 230 for details.*

Watchdog Timer Control Register Definitions

Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) register is a write-only control register. Writing the 55H, AAH unlock sequence to the WDTCTL register address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTL, and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL register address produce no effect on the bits in the WDTCTL register. The locking mechanism prevents spurious writes to the Reload registers.

This register address is shared with the read-only Reset Status register.

Table 57. Watchdog Timer Control Register (WDTCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	WDTUNLK							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
ADDR	FF0H							
X = Undefined.								

WDTUNLK—Watchdog Timer Unlock

The software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer reload registers.

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTL, and WDTL) registers ([Table 58](#) through [Table 60](#)) 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 appropriate Reload Value. Reading from these registers returns the current Watchdog Timer count value.

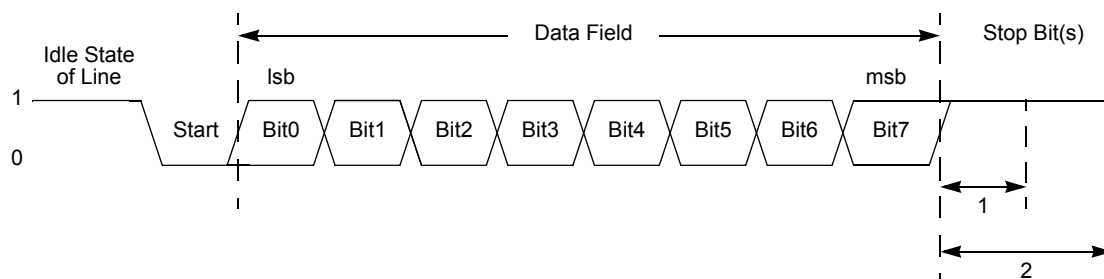


Figure 11. UART Asynchronous Data Format without Parity

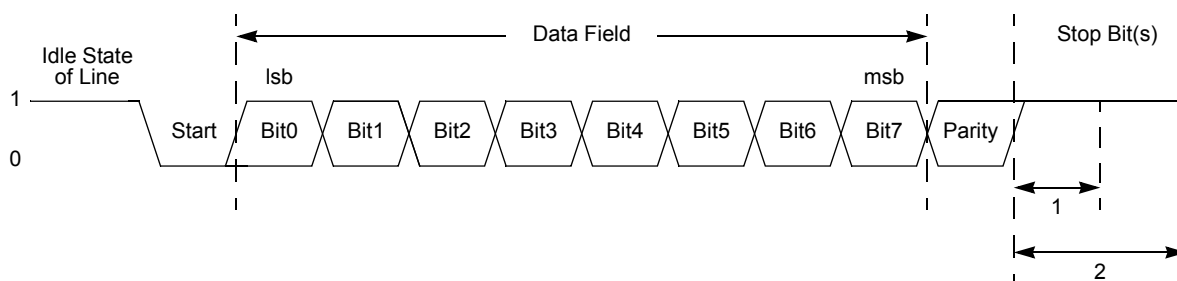


Figure 12. UART Asynchronous Data Format with Parity

Transmitting Data using the Polled Method

Follow the steps below to transmit data using the polled method of operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the required baud rate.
2. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
3. Write to the UART Control 1 register, if MULTIPROCESSOR mode is appropriate, to enable MULTIPROCESSOR (9-bit) mode functions.
4. Set the Multiprocessor Mode Select (MPEN) bit to enable MULTIPROCESSOR mode.
5. Write to the UART Control 0 register to:
 - Set the transmit enable bit (TEN) to enable the UART for data transmission.
 - Set the parity enable bit (PEN), if parity is appropriate and MULTIPROCESSOR mode is not enabled, and select either even or odd parity (PSEL).
 - Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.

3. Write to the [ADC Control Register 0](#) 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 from the available analog input sources (different input pins available depending on the device).
 - Set `CONT` to 1 to select continuous conversion.
 - If the internal `VREF` must be output to a pin, set the `REFEXT` bit to 1. The internal voltage reference must be enabled in this case.
 - Write the `REFSELL` bit of the pair `{REFSELH, REFSELL}` to select the internal voltage reference level or to disable the internal reference. The `REFSELH` bit is contained in [ADC Control/Status Register 1](#).
 - Set `CEN` to 1 to start the conversions.
4. 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.
5. The ADC writes a new data result every 256 system clock cycles. For each completed conversion, the ADC control logic performs the following operations:
 - Writes the 13-bit two's complement result to `{ADCD_H[7:0], ADCD_L[7:3]}`.
 - Sends an interrupt request to the Interrupt Controller denoting conversion complete.
6. To disable continuous conversion, clear the `CONT` bit in the ADC Control Register to 0.

Interrupts

The ADC is able to interrupt the CPU when a conversion has been completed. When the ADC is disabled, no new interrupts are asserted; however, an interrupt pending when the ADC is disabled is not cleared.

Calibration and Compensation

The Z8 Encore! XP[®] F082A Series ADC is factory calibrated for offset error and gain error, with the compensation data stored in Flash memory. Alternatively, you can perform your own calibration, storing the values into Flash themselves. Thirdly, the user code can perform a manual offset calibration during DIFFERENTIAL mode operation.

Low Power Operational Amplifier

Overview

The LPO is a general-purpose low power operational amplifier. Each of the three ports of the amplifier is accessible from the package pins. The LPO contains only one pin configuration: ANA0 is the output/feedback node, ANA1 is the inverting input and ANA2 is the non-inverting input.

Operation

To use the LPO, it must be enabled in the [Power Control Register 0 \(PWRCTL0\)](#). The default state of the LPO is OFF. To use the LPO, the LPO bit must be cleared, turning it ON ([Power Control Register 0 \(PWRCTL0\)](#) on page 35). When making normal ADC measurements on ANA0 (measurements not involving the LPO output), the LPO bit must be OFF. Turning the LPO bit ON interferes with normal ADC measurements.



Warning: *The LPO bit enables the amplifier even in STOP mode. If the amplifier is not required in STOP mode, disable it. Failing to perform this results in STOP mode currents higher than necessary.*

As with other ADC measurements, any pins used for analog purposes must be configured as such in the GPIO registers (see [Port A–D Alternate Function Sub-Registers](#) on page 47).

LPO output measurements are made on ANA0, as selected by the ANAIN[3:0] bits of [ADC Control Register 0](#). It is also possible to make single-ended measurements on ANA1 and ANA2 while the amplifier is enabled, which is often useful for determining offset conditions. Differential measurements between ANA0 and ANA2 may be useful for noise cancellation purposes.

If the LPO output is routed to the ADC, then the BUFFMODE[2:0] bits of [ADC Control/Status Register 1](#) must also be configured for unity-gain buffered operation. Sampling the LPO in an unbuffered mode is not recommended.

When either input is overdriven, the amplifier output saturates at the positive or negative supply voltage. No instability results.

Assuming a compensated ADC measurement, the following equation defines the relationship between the ADC reading and the die temperature:

$$T = (25/128) \times (ADC - TSCAL[11:2]) + 30$$

where, T is the temperature in C; ADC is the 10-bit compensated ADC value; and TSCAL is the temperature sensor calibration value, ignoring the two least significant bits of the 12-bit value.

See [Temperature Sensor Calibration Data](#) on page 164 for the location of TSCAL.

Calibration

The temperature sensor undergoes calibration during the manufacturing process and is maximally accurate at 30 °C. Accuracy decreases as measured temperatures move further from the calibration point.

Packaging

Figure 39 displays the 8-pin Plastic Dual Inline Package (PDIP) available for Z8 Encore! XP[®] F082A Series devices.

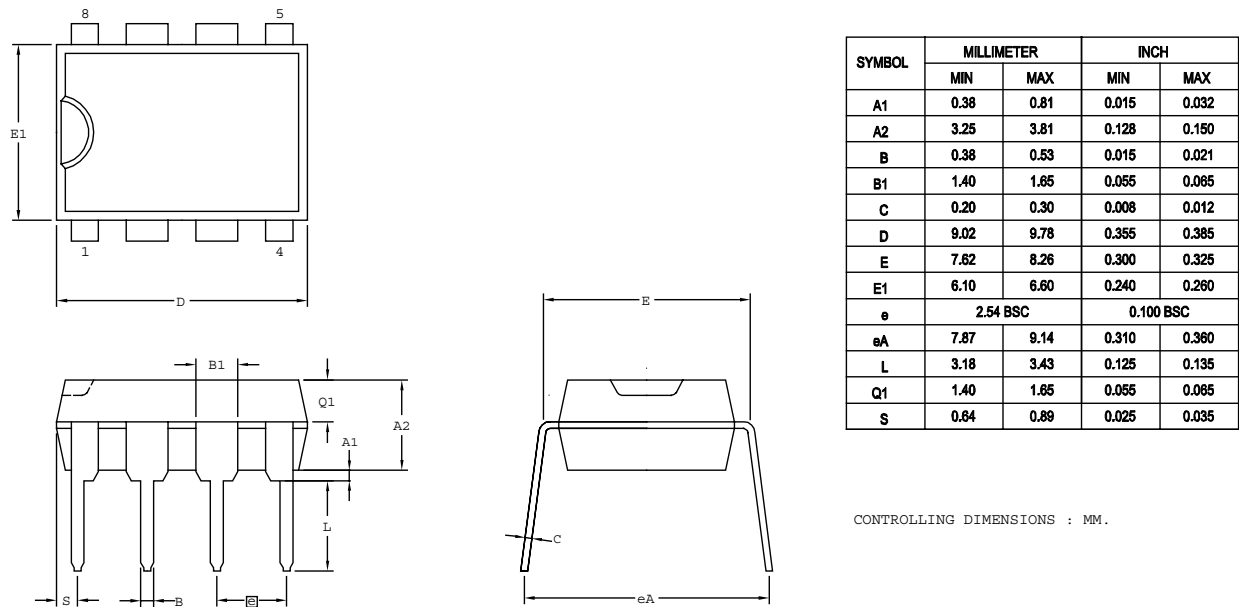


Figure 39. 8-Pin Plastic Dual Inline Package (PDIP)



Part Number Suffix Designations

Z8 F 04 2A S H 020 S C

Environmental Flow

C = Standard Plastic Packaging Compound
G = Green Plastic Packaging Compound

Temperature Range

S = Standard, 0 °C to 70 °C
E = Extended, -40 °C to +105 °C

Speed

020 = 20 MHz

Pin Count

B = 8
H = 20
J = 28

Package

H = SSOP
P = PDIP
Q = QFN
S = SOIC

Device Type

2A = Contains Advanced Analog Peripherals
1A = Does Not Contain Advanced Analog Peripherals

Memory Size

08 = 8 KB Flash, 1 KB RAM, 0 B NVDS
04 = 4 KB Flash, 1 KB RAM, 128 B NVDS
02 = 2 KB Flash, 512 B RAM, 64 B NVDS
01 = 1 KB Flash, 256 B RAM, 16 B NVDS

Memory Type

F = Flash

Device Family

Z8 = Zilog's 8-Bit Microcontroller

- timing 237
- OCD commands
 - execute instruction (12H) 183
 - read data memory (0DH) 183
 - read OCD control register (05H) 181
 - read OCD revision (00H) 180
 - read OCD status register (02H) 180
 - read program counter (07H) 181
 - read program memory (0BH) 182
 - read program memory CRC (0EH) 183
 - read register (09H) 182
 - read runtime counter (03H) 180
 - step instruction (10H) 183
 - stuff instruction (11H) 183
 - write data memory (0CH) 182
 - write OCD control register (04H) 181
 - write program counter (06H) 181
 - write program memory (0AH) 182
 - write register (08H) 181
- on-chip debugger (OCD) 173
- on-chip debugger signals 12
- on-chip oscillator 193
- ONE-SHOT mode 84
- opcode map
 - abbreviations 217
 - cell description 216
 - first 218
 - second after 1FH 219
- Operational Description 23, 33, 37, 55, 69, 91, 97, 117, 121, 134, 135, 139, 141, 153, 169, 173, 187, 193, 197
- OR 205
- ordering information 251
- ORX 206
- oscillator signals 12

P

- p 201
- packaging
 - 20-pin PDIP 244, 245
 - 20-pin SSOP 246, 249
 - 28-pin PDIP 247
 - 28-pin SOIC 248

- 8-pin PDIP 241
- 8-pin SOIC 242
- PDIP 248, 249
- part selection guide 2
- PC 202
- PDIP 248, 249
- peripheral AC and DC electrical characteristics 229
- pin characteristics 13
- Pin Descriptions 9
- polarity 201
- POP 205
- pop using extended addressing 205
- POPX 205
- port availability, device 37
- port input timing (GPIO) 235
- port output timing, GPIO 236
- power supply signals 13
- power-down, automatic (ADC) 122
- Power-on and Voltage Brownout electrical characteristics and timing 229
- Power-On Reset (POR) 25
- program control instructions 206
- program counter 202
- program memory 15
- PUSH 205
- push using extended addressing 205
- PUSHX 205
- PWM mode 85
- PxADDR register 46
- PxCTL register 47

R

- R 201
- r 201
- RA
 - register address 201
- RCF 204
- receive
 - IrDA data 119
- receiving UART data-interrupt-driven method 102
- receiving UART data-pollled method 101