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, WDT
Number of I/O	17
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f021ash020sc00tr

Table 3. Pin Characteristics (20- and 28-pin Devices) (Continued)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt-Trigger Input	Open Drain Output	5 V Tolerance
PC[7:0]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	PC[7:3] unless pullups enabled
RESET/PD0	I/O	I/O (defaults to RESET)	Low (in Reset mode)	Yes (PD0 only)	Programmable for PD0; always on for RESET	Yes	Programmable for PD0; always on for RESET	Yes, unless pullups enabled
VDD	N/A	N/A	N/A	N/A			N/A	N/A
VSS	N/A	N/A	N/A	N/A			N/A	N/A

> ▶ **Note:** *PB6 and PB7 are available only in those devices without ADC.*

Table 4. Pin Characteristics (8-Pin Devices)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt-Trigger Input	Open Drain Output	5 V Tolerance
PA0/DBG	I/O	I (but can change during reset if key sequence detected)	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
PA1	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
RESET/PA2	I/O	I/O (defaults to RESET)	Low (in Reset mode)	Yes	Programmable for PA2; always on for RESET	Yes	Programmable for PA2; always on for RESET	Yes, unless pull-ups enabled
PA[5:3]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
V_{DD}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
V_{SS}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

PIN[7:0]—Port Input Data
Sampled data from the corresponding port pin input.
0 = Input data is logical 0 (Low).
1 = Input data is logical 1 (High).

Port A–D Output Data Register

The Port A–D Output Data register (Table 28) controls the output data to the pins.

Table 28. Port A–D Output Data Register (P*x*OUT)

BITS	7	6	5	4	3	2	1	0
FIELD	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
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	FD3H, FD7H, FDBH, FDFH							

POUT[7:0]—Port Output Data
These bits contain the data to be driven to the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation.
0 = Drive a logical 0 (Low).
1= Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding Port Output Control register bit to 1.

LED Drive Enable Register

The LED Drive Enable register (Table 29) activates the controlled current drive. The Port C pin must first be enabled by setting the Alternate Function register to select the LED function.

Table 29. LED Drive Enable (LEDEN)

BITS	7	6	5	4	3	2	1	0
FIELD	LEDEN[7:0]							
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	F82H							

Interrupt Controller

The interrupt controller on the Z8 Encore! XP F082A Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 possible interrupt sources with 18 unique interrupt vectors:
    – Twelve GPIO port pin interrupt sources (two interrupt vectors are shared).
    – Eight on-chip peripheral interrupt sources (two interrupt vectors are shared).
- Flexible GPIO interrupts:
    – Eight selectable rising and falling edge GPIO interrupts.
    – Four dual-edge interrupts.
- Three levels of individually programmable interrupt priority.
- Watchdog Timer and LVD can be configured to generate an interrupt.
- Supports vectored as well as polled interrupts

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the 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.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt controller has no effect on operation. For more information on interrupt servicing by the eZ8 CPU, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com.

Interrupt Vector Listing

Table 32 on page 56 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

> **Note:** *Some port interrupts are not available on the 8- and 20-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.*

Table 39. IRQ1 Enable and Priority Encoding

IRQ1ENH[*x*]	IRQ1ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Medium
1	1	Level 3	High

where *x* indicates the register bits from 0–7.

Table 40. IRQ1 Enable High Bit Register (IRQ1ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VENH	PA6CENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
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	FC4H							

PA7VENH—Port A Bit[7] or LVD Interrupt Request Enable High Bit
PA6CENH—Port A Bit[7] or Comparator Interrupt Request Enable High Bit
PA*x*ENH—Port A Bit[*x*] Interrupt Request Enable High Bit

See Shared Interrupt Select (IRQSS) register for selection of either the LVD or the comparator as the interrupt source.

Table 41. IRQ1 Enable Low Bit Register (IRQ1ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
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	FC5H							

PA7VENL—Port A Bit[7] or LVD Interrupt Request Enable Low Bit
PA6CENL—Port A Bit[6] or Comparator Interrupt Request Enable Low Bit
PA*x*ENL—Port A Bit[*x*] Interrupt Request Enable Low Bit

Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state for one system clock cycle (from Low to High or from High to Low) upon timer Reload. If it is appropriate to have the Timer Output make a state change at a One-Shot time-out (rather than a single cycle pulse), first set the TPOL bit in the Timer Control Register to the start value before enabling ONE-SHOT mode. After starting the timer, set TPOL to the opposite bit value.

Follow the steps below for configuring a timer for ONE-SHOT mode and initiating the count:

1. Write to the Timer Control register to:
   - Disable the timer
   - Configure the timer for ONE-SHOT mode.
   - Set the prescale value.
   - Set the initial output level (High or Low) if using the Timer Output alternate function.

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 Reload value.

4. If appropriate, enable the timer interrupt 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 ONE-SHOT mode, the system clock always provides the timer input. The timer period is given by the following equation:

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

CONTINUOUS Mode

In CONTINUOUS mode, 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. 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. 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.

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

1. Write to the Timer Control register to:
   - Disable the timer
   - Configure the timer for CONTINUOUS mode.

- 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.

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

1. Write to the Timer Control register to:
   – Disable the timer.
   – Configure the timer for COUNTER mode.
   – Select either the rising edge or falling edge of the Timer Input signal for the count. This selection also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function is not required 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`. 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 appropriate, 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 register to enable the timer.

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

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

COMPARATOR COUNTER Mode

In COMPARATOR COUNTER mode, the timer counts input transitions from the analog comparator output. The `TPOL` bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the comparator output signal. In COMPARATOR COUNTER mode, the prescaler is disabled.

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

After 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.

Timer Control Register Definitions

Timer 0–1 Control Registers

Time 0–1 Control Register 0

The Timer Control Register 0 (TxCTL0) and Timer Control Register 1 (TxCTL1) determine the timer operating mode (Table 48). It also includes a programmable PWM dead-band delay, two bits to configure timer interrupt definition, and a status bit to identify if the most recent timer interrupt is caused by an input capture event.

Table 48. Timer 0–1 Control Register 0 (TxCTL0)

BITS	7	6	5	4	3	2	1	0
FIELD	TMODEHI	TICONFIG		Reserved	PWMD			INPCAP
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
ADDR	F06H, F0EH							

TMODEHI—Timer Mode High Bit
This bit along with the TMODE field in TxCTL1 register determines the operating mode of the timer. This is the most significant bit of the Timer mode selection value. See the TxCTL1 register description for details of the full timer mode decoding.

TICONFIG—Timer Interrupt Configuration
This field configures timer interrupt definition.

  0x = Timer Interrupt occurs on all defined Reload, Compare and Input Events
  10 = Timer Interrupt only on defined Input Capture/Deassertion Events
  11 = Timer Interrupt only on defined Reload/Compare Events

Reserved—Must be 0.

PWMD—PWM Delay value
This field is a programmable delay to control the number of system clock cycles delay before the Timer Output and the Timer Output Complement are forced to their active state.

  000 = No delay
  001 = 2 cycles delay
  010 = 4 cycles delay
  011 = 8 cycles delay
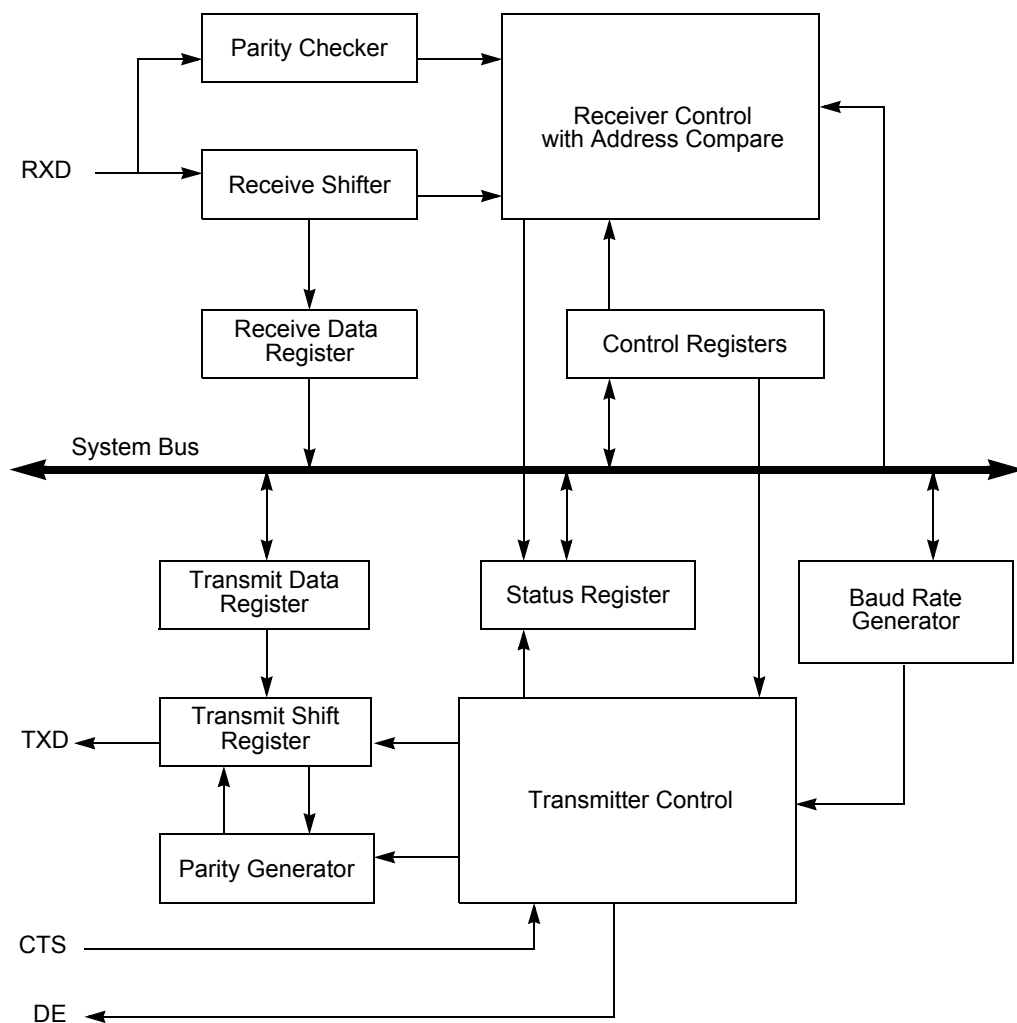 100 = 16 cycles delay
  101 = 32 cycles delay

Figure 10. UART Block Diagram

Operation

Data Format

The UART always transmits and receives data in an 8-bit data format, least-significant bit first. An even or odd parity bit can be added to the data stream. Each character begins with an active Low START bit and ends with either 1 or 2 active High STOP bits. Figure 11 and Figure 12 display the asynchronous data format employed by the UART without parity and with parity, respectively.

The UART is now configured for interrupt-driven data transmission. Because the UART Transmit Data register is empty, an interrupt is generated immediately. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Write the UART Control 1 register to select the multiprocessor bit for the byte to be transmitted:

2. Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.

3. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Transmit Shift register and transmits the data.

4. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.

5. Execute the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data register to again become empty.

Receiving Data using the Polled Method

Follow the steps below to configure the UART for polled data reception:

1. Write to the UART Baud Rate High and Low Byte registers to set an acceptable baud rate for the incoming data stream.

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 to enable MULTIPROCESSOR mode functions, if appropriate.

4. Write to the UART Control 0 register to:
   – Set the receive enable bit (REN) to enable the UART for data reception
   – Enable parity, if appropriate and if Multiprocessor mode is not enabled, and select either even or odd parity.

5. Check the RDA bit in the UART Status 0 register to determine if the Receive Data register contains a valid data byte (indicated by a 1). If RDA is set to 1 to indicate available data, continue to Step 5. If the Receive Data register is empty (indicated by a 0), continue to monitor the RDA bit awaiting reception of the valid data.

6. Read data from the UART Receive Data register. If operating in MULTIPROCESSOR (9-bit) mode, further actions may be required depending on the MULTIPROCESSOR mode bits MPMD[1:0].

7. Return to Step 4 to receive additional data.

Flash Program Memory Address 0001H

Table 87. Flash Options Bits at Program Memory Address 0001H

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved			XTLDIS	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 0001H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Reserved—Must be 1.

XTLDIS—State of Crystal Oscillator at Reset.

> **Note:** *This bit only enables the crystal oscillator. Its selection as system clock must be done manually.*
> *0 = Crystal oscillator is enabled during reset, resulting in longer reset timing*
> *1 = Crystal oscillator is disabled during reset, resulting in shorter reset timing*

> ⚠ **Warning:** *Programming the XTLDIS bit to zero on 8-pin versions of this device prevents any further communication via the debug pin. This is due to the fact that the XIN and DBG functions are shared on pin 2 of this package. Do not program this bit to zero on 8-pin devices unless no further debugging or Flash programming is required.*

Trim Bit Address Space

All available Trim bit addresses and their functions are listed in Table 88 through Table 92.

Trim Bit Address 0000H

Table 88. Trim Options Bits at Address 0000H

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	Information Page Memory 0020H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Reserved—Altering this register may result in incorrect device operation.

Trim Bit Address 0004H

Table 92. Trim Option Bits at 0004H

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	Information Page Memory 0024H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Reserved—Altering this register may result in incorrect device operation.

Zilog Calibration Data

ADC Calibration Data

Table 93. ADC Calibration Bits

BITS	7	6	5	4	3	2	1	0
FIELD	ADC_CAL							
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	Information Page Memory 0060H–007DH							
Note: U = Unchanged by Reset. R/W = Read/Write.								

ADC_CAL—Analog-to-Digital Converter Calibration Values
Contains factory calibrated values for ADC gain and offset compensation. Each of the ten supported modes has one byte of offset calibration and two bytes of gain calibration. These values are read by the software to compensate ADC measurements as described in Software Compensation Procedure Using Factory Calibration Data on page 126. The location of each calibration byte is provided in Table 94 on page 162.

- **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 0). If the device is not in DEBUG mode or if the Flash 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]
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

- **Write Program Memory (0AH)**—The Write Program Memory command writes data to Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Program Memory (0BH)**—The Read Program Memory command reads data from Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1–65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

- **Write Data Memory (0CH)**—The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
```

Internal Precision Oscillator

The internal precision oscillator (IPO) is designed for use without external components. You can either manually trim the oscillator for a non-standard frequency or use the automatic factory-trimmed version to achieve a 5.53 MHz frequency. IPO features include:

- On-chip RC oscillator that does not require external components

- Output frequency of either 5.53 MHz or 32.8 kHz (contains both a fast and a slow mode)

- Trimmed through Flash option bits with user override

- Elimination of crystals or ceramic resonators in applications where very high timing accuracy is not required

Operation

An 8-bit trimming register, incorporated into the design, compensates for absolute variation of oscillator frequency. Once trimmed the oscillator frequency is stable and does not require subsequent calibration. Trimming is performed during manufacturing and is not necessary for you to repeat unless a frequency other than 5.53 MHz (fast mode) or 32.8 kHz (slow mode) is required. This trimming is done at +30 ºC and a supply voltage of 3.3 V, so accuracy of this operating point is optimal.

If not used, the IPO can be disabled by the Oscillator Control register (see Oscillator Control Register Definitions on page 190).

By default, the oscillator frequency is set by the factory trim value stored in the write-protected Flash information page. However, the user code can override these trim values as described in Trim Bit Address Space on page 158.

Select one of two frequencies for the oscillator: 5.53 MHz and 32.8 kHz, using the OSCSEL bits in the Oscillator Control on page 187.

Table 119. CPU Control Instructions (Continued)

Mnemonic	Operands	Instruction
SCF	—	Set Carry Flag
SRP	src	Set Register Pointer
STOP	—	STOP Mode
WDT	—	Watchdog Timer Refresh

Table 120. Load Instructions

Mnemonic	Operands	Instruction
CLR	dst	Clear
LD	dst, src	Load
LDC	dst, src	Load Constant to/from Program Memory
LDCI	dst, src	Load Constant to/from Program Memory and Auto-Increment Addresses
LDE	dst, src	Load External Data to/from Data Memory
LDEI	dst, src	Load External Data to/from Data Memory and Auto-Increment Addresses
LDWX	dst, src	Load Word using Extended Addressing
LDX	dst, src	Load using Extended Addressing
LEA	dst, X(src)	Load Effective Address
POP	dst	Pop
POPX	dst	Pop using Extended Addressing
PUSH	src	Push
PUSHX	src	Push using Extended Addressing

Table 121. Logical Instructions

Mnemonic	Operands	Instruction
AND	dst, src	Logical AND
ANDX	dst, src	Logical AND using Extended Addressing
COM	dst	Complement
OR	dst, src	Logical OR

Table 124. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
SUBX dst, src	dst ← dst – src	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29							4	3
SWAP dst	dst[7:4] ↔ dst[3:0]	R		F0	X	*	*	X	–	–	2	2
		IR		F1							2	3
TCM dst, src	(NOT dst) AND src	r	r	62	–	*	*	0	–	–	2	3
		r	Ir	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	–	*	*	0	–	–	4	3
		ER	IM	69							4	3
TM dst, src	dst AND src	r	r	72	–	*	*	0	–	–	2	3
		r	Ir	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	–	*	*	0	–	–	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	–	–	–	–	–	–	2	6
WDT				5F	–	–	–	–	–	–	1	2
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

Table 135. Analog-to-Digital Converter Electrical Characteristics and Timing (Continued)

Symbol	Parameter	V_{DD} = 3.0 V to 3.6 V T_A = 0 °C to +70 °C (unless otherwise stated)				
		Minimum	Typical	Maximum	Units	Conditions
	Single-Shot Conversion Time	–	5129	–	System clock cycles	All measurements but temperature sensor
			10258			Temperature sensor measurement
	Continuous Conversion Time	–	256	–	System clock cycles	All measurements but temperature sensor
			512			Temperature sensor measurement
	Signal Input Bandwidth	–	10		kHz	As defined by -3 dB point
R_S	Analog Source Impedance[4]	–	–	10	kΩ	In unbuffered mode
				500	kΩ	In buffered modes
Zin	Input Impedance	–	150		kΩ	In unbuffered mode at 20 MHz[5]
		10	–		MΩ	In buffered modes
Vin	Input Voltage Range	0		V_{DD}	V	Unbuffered Mode
		0.3		V_{DD}-1.1	V	Buffered Modes

▶ **Note:** *These values define the range over which the ADC performs within spec; exceeding these values does not cause damage or instability;* see DC Characteristics on page 222 *for absolute pin voltage limits*

Notes
1. Analog source impedance affects the ADC offset voltage (because of pin leakage) and input settling time.
2. Devices are factory calibrated at V_{DD} = 3.3 V and T_A = +30 °C, so the ADC is maximally accurate under these conditions.
3. LSBs are defined assuming 10-bit resolution.
4. This is the maximum recommended resistance seen by the ADC input pin.
5. The input impedance is inversely proportional to the system clock frequency.

Table 138. Temperature Sensor Electrical Characteristics

| Symbol | Parameter | V$_{DD}$ = 2.7 V to 3.6 V | | | Units | Conditions |
		Minimum	Typical	Maximum		
T$_{AERR}$	Temperature Error		±0.5	±2	°C	Over the range +20 °C to +30 °C (as measured by ADC)[1]
			±1	±5	°C	Over the range +0 °C to +70 °C (as measured by ADC)
			±2	±7	°C	Over the range +0 °C to +105 °C (as measured by ADC)
			±7		°C	Over the range -40 °C to +105 °C (as measured by ADC)
T$_{AERR}$	Temperature Error		TBD		°C	Over the range -40 °C to +105 °C (as measured by comparator)
t$_{WAKE}$	Wakeup Time		80	100	µs	Time required for Temperature Sensor to stabilize after enabling

[1]Devices are factory calibrated at for maximal accuracy between +20 °C and +30 °C, so the sensor is maximally accurate in that range. User re-calibration for a different temperature range is possible and increases accuracy near the new calibration point.

General Purpose I/O Port Input Data Sample Timing

Figure 34 displays timing of the GPIO Port input sampling. The input value on a GPIO Port pin is sampled on the rising edge of the system clock. The Port value is available to the eZ8 CPU on the second rising clock edge following the change of the Port value.

UART Timing

Figure 37 and Table 142 provide timing information for UART pins for the case where CTS is used for flow control. The CTS to DE assertion delay (T1) assumes the transmit data register has been loaded with data prior to CTS assertion.
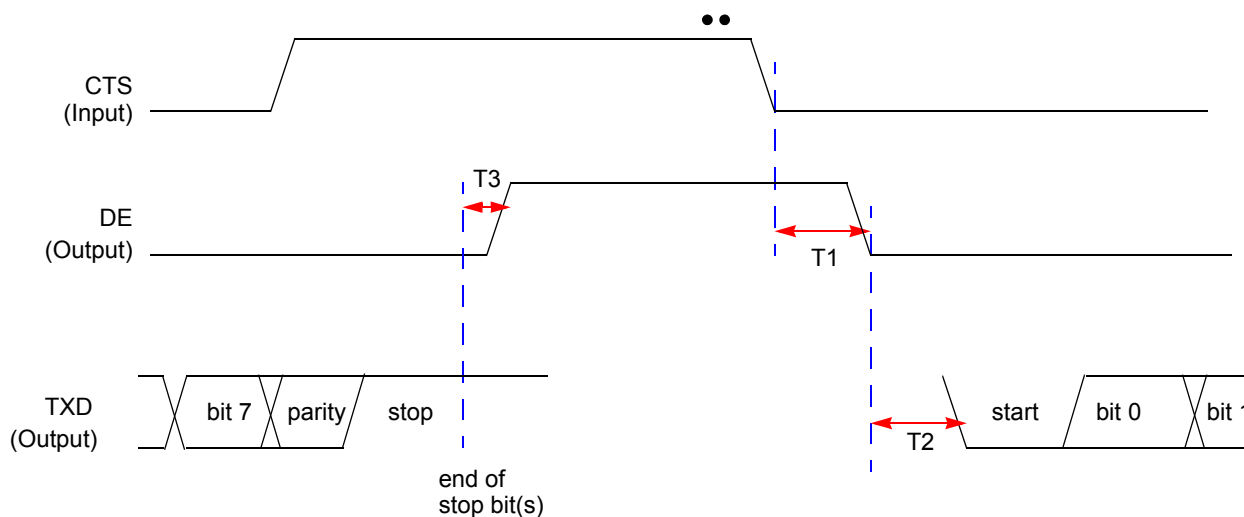

Figure 37. UART Timing With CTS

Table 142. UART Timing With CTS

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
UART			
T_1	CTS Fall to DE output delay	2 * XIN period	2 * XIN period + 1 bit time
T_2	DE assertion to TXD falling edge (start bit) delay	± 5	
T_3	End of Stop Bit(s) to DE deassertion delay	± 5	

Figure 42 displays the 20-pin Plastic Dual Inline Package (PDIP) available for the Z8 Encore! XP F082A Series devices.

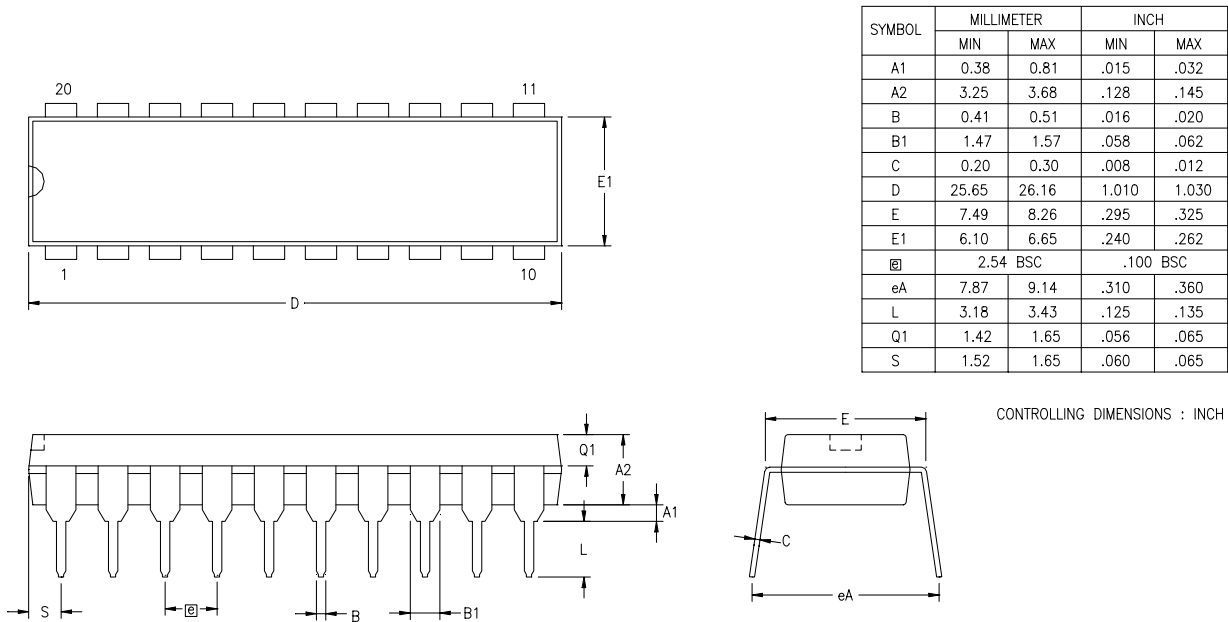| SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.38	0.81	.015	.032
A2	3.25	3.68	.128	.145
B	0.41	0.51	.016	.020
B1	1.47	1.57	.058	.062
C	0.20	0.30	.008	.012
D	25.65	26.16	1.010	1.030
E	7.49	8.26	.295	.325
E1	6.10	6.65	.240	.262
℮	2.54 BSC		.100 BSC	
eA	7.87	9.14	.310	.360
L	3.18	3.43	.125	.135
Q1	1.42	1.65	.056	.065
S	1.52	1.65	.060	.065

CONTROLLING DIMENSIONS : INCH

Figure 42. 20-Pin Plastic Dual Inline Package (PDIP)