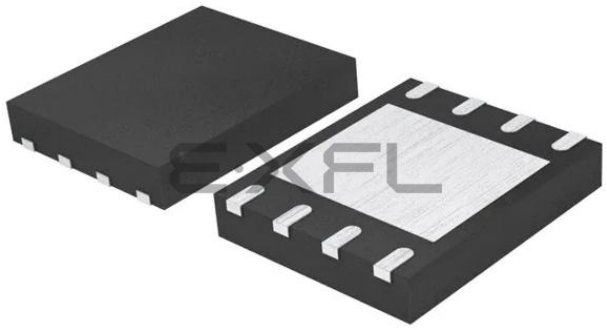


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	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	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	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°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/z8f041aqb020sg

operate in ONE-SHOT, CONTINUOUS, GATED, CAPTURE, CAPTURE RESTART, COMPARE, CAPTURE and COMPARE, PWM SINGLE OUTPUT and PWM DUAL OUTPUT modes.

General-Purpose Input/Output

The Product Line MCUs feature 6 to 25 port pins (Ports A–D) for general- purpose input/output (GPIO). The number of GPIO pins available is a function of package and each pin is individually programmable. 5 V tolerant input pins are available on all I/Os on 8-pin devices and most I/Os on other package types.

Direct LED Drive

The 20- and 28-pin devices support controlled current sinking output pins capable of driving LEDs without the need for a current limiting resistor. These LED drivers are independently programmable to four different intensity levels.

Flash Controller

The Flash Controller programs and erases Flash memory. The Flash Controller supports several protection mechanisms against accidental program and erasure, plus factory serialization and read protection.

Non-Volatile Data Storage

The nonvolatile data storage (NVDS) uses a hybrid hardware/software scheme to implement a byte programmable data memory and is capable of over 100,000 write cycles.

Note: Devices with 8KB of Flash memory do not include the NVDS feature.

Interrupt Controller

The Z8 Encore! XP F082A Series products support up to 20 interrupts. These interrupts consist of 8 internal peripheral interrupts and 12 general-purpose I/O pin interrupt sources. The interrupts have three levels of programmable interrupt priority.

Reset Controller

The Z8 Encore! XP F082A Series products can be reset using the $\overline{\text{RESET}}$ pin, Power-On Reset, Watchdog Timer (WDT) time-out, STOP Mode exit, or Voltage Brown-Out (VBO)

Table 2. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
Analog		
ANA[7:0]	I	Analog Port. These signals are used as inputs to the analog-to-digital converter (ADC).
VREF	I/O	Analog-to-digital converter reference voltage input, or buffered output for internal reference.
Low-Power Operational Amplifier (LPO)		
AMPINP/AMPINN	I	LPO inputs. If enabled, these pins drive the positive and negative amplifier inputs respectively.
AMPOUT	O	LPO output. If enabled, this pin is driven by the on-chip LPO.
Oscillators		
XIN	I	External Crystal Input. This is the input pin to the crystal oscillator. A crystal can be connected between it and the X_{OUT} pin to form the oscillator. In addition, this pin is used with external RC networks or external clock drivers to provide the system clock.
X_{OUT}	O	External Crystal Output. This pin is the output of the crystal oscillator. A crystal can be connected between it and the **XIN** pin to form the oscillator.
Clock Input		
CLKIN	I	Clock Input Signal. This pin may be used to input a TTL-level signal to be used as the system clock.
LED Drivers		
LED	O	Direct LED drive capability. All port C pins have the capability to drive an LED without any other external components. These pins have programmable drive strengths set by the GPIO block.
On-Chip Debugger		
DBG	I/O	Debug. This signal is the control and data input and output to and from the On-Chip Debugger. **Caution:** The DBG pin is open-drain and requires a pull-up resistor to ensure proper operation.

Notes:
1. PB6 and PB7 are only available in 28-pin packages without ADC. In 28-pin packages with ADC, they are replaced by AV_{DD} and AV_{SS}.
2. The AV_{DD} and AV_{SS} signals are available only in 28-pin packages with ADC. They are replaced by PB6 and PB7 on 28-pin packages without ADC.

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
FD9	Port C Control	PCCTL	00	46
FDA	Port C Input Data	PCIN	XX	46
FDB	Port C Output Data	PCOUT	00	46
GPIO Port D				
FDC	Port D Address	PDADDR	00	44
FDD	Port D Control	PDCTL	00	46
FDE	Reserved	—	XX	
FDF	Port D Output Data	PDOUT	00	46
FE0–FEF	Reserved	—	XX	
Watchdog Timer (WDT)				
FF0	Reset Status (Read-only)	RSTSTAT	X0	29
	Watchdog Timer Control (Write-only)	WDTCTL	N/A	96
FF1	Watchdog Timer Reload Upper Byte	WDTU	00	97
FF2	Watchdog Timer Reload High Byte	WDTH	04	97
FF3	Watchdog Timer Reload Low Byte	WDTL	00	98
FF4–FF5	Reserved	—	XX	
Trim Bit Control				
FF6	Trim Bit Address	TRMADR	00	161
FF7	Trim Bit Data	TRMDR	00	162
Flash Memory Controller				
FF8	Flash Control	FCTL	00	155
FF8	Flash Status	FSTAT	00	155
FF9	Flash Page Select	FPS	00	156
	Flash Sector Protect	FPROT	00	157
FFA	Flash Programming Frequency High Byte	FFREQH	00	158
FFB	Flash Programming Frequency Low Byte	FFREQL	00	158
eZ8 CPU				
FFC	Flags	—	XX	See footnote 2.
FFD	Register Pointer	RP	XX	
FFE	Stack Pointer High Byte	SPH	XX	
FFF	Stack Pointer Low Byte	SPL	XX	

Notes:
1. XX = Undefined.
2. Refer to the eZ8 CPU Core User Manual (UM0128).

Port A–D Pull-up Enable Subregisters

The Port A–D Pull-up Enable Subregister, shown in Table 26, is accessed through the Port A–D Control Register by writing 06H to the Port A–D Address Register. Setting the bits in the Port A–D Pull-up Enable subregisters enables a weak internal resistive pull-up on the specified port pins.

Table 26. Port A–D Pull-Up Enable Subregisters (PxPUE)

Bit	7	6	5	4	3	2	1	0
Field	PPUE7	PPUE6	PPUE5	PPUE4	PPUE3	PPUE2	PPUE1	PPUE0
RESET	00H (Ports A-C); 01H (Port D); 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
Address	If 06H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] PPUEx	**Port Pull-up Enabled** 0 = The weak pull-up on the port pin is disabled. 1 = The weak pull-up on the port pin is enabled.

Note: x indicates the specific GPIO port pin number (7–0).

Port A–D Alternate Function Set 1 Subregisters

The Port A–D Alternate Function Set1 Subregister, shown in Table 27, is accessed through the Port A–D Control Register by writing 07H to the Port A–D Address Register. The Alternate Function Set 1 subregisters selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register are defined in the GPIO Alternate Functions section on page 37.

Note: Alternate function selection on port pins must also be enabled as described in the Port A–D Alternate Function Subregisters section on page 47.

– Set the Capture edge (rising or falling) for the Timer Input

2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
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.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

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

WDT Reset in Normal Operation

If configured to generate a Reset when a time-out occurs, the Watchdog Timer forces the device into the System Reset state. The WDT status bit in the Reset Status (RSTSTAT) Register is set to 1. For more information about system reset, see the Reset, Stop Mode Recovery and Low Voltage Detection chapter on page 22.

WDT Reset in STOP Mode

If configured to generate a Reset when a time-out occurs and the device is in STOP Mode, the Watchdog Timer initiates a Stop Mode Recovery. Both the WDT status bit and the STOP bit in the Reset Status (RSTSTAT) Register are set to 1 following WDT time-out in STOP Mode.

Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watchdog Timer (WDTCTL) Control Register address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH 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. Observe the following steps to unlock the Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) for write access.

1. Write 55H to the Watchdog Timer Control Register (WDTCTL).
2. Write AAH to the Watchdog Timer Control Register (WDTCTL).
3. Write the Watchdog Timer Reload Upper Byte Register (WDTU) with the appropriate time-out value.
4. Write the Watchdog Timer Reload High Byte Register (WDTH) with the appropriate time-out value.
5. Write the Watchdog Timer Reload Low Byte Register (WDTL) with the appropriate time-out value.

All three Watchdog Timer Reload registers must be written in the order just listed. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes can occur unless the sequence is restarted. The value in the Watchdog Timer Reload registers is loaded into the counter when the Watchdog Timer is first enabled and every time a WDT instruction is executed.

Watchdog Timer Calibration

Due to its extremely low operating current, the Watchdog Timer oscillator is somewhat inaccurate. This variation can be corrected using the calibration data stored in the Flash Information Page; see Tables 100 and 101 on page 173 for details. Loading these values

- Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin

6. Check the TDRE bit in the UART Status 0 Register to determine if the Transmit Data Register is empty (indicated by a 1). If empty, continue to Step 7. If the Transmit Data Register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data Register becomes available to receive new data.
7. Write the UART Control 1 Register to select the outgoing address bit.
8. Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
9. 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.
10. Make any changes to the Multiprocessor Bit Transmitter (MPBT) value, if appropriate and MULTIPROCESSOR Mode is enabled.
11. To transmit additional bytes, return to Step 5.

Transmitting Data using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data Register to accept new data for transmission. Observe the following steps to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the acceptable priority.
5. Write to the UART Control 1 Register to enable MULTIPROCESSOR (9-bit) Mode functions, if MULTIPROCESSOR Mode is appropriate.
6. Set the MULTIPROCESSOR Mode Select (MPEN) to Enable MULTIPROCESSOR Mode.
7. Write to the UART Control 0 Register to:
   - Set the transmit enable bit (TEN) to enable the UART for data transmission
   - Enable parity, if appropriate and if MULTIPROCESSOR Mode is not enabled and select either even or odd parity

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Checks the UART Status 0 Register to determine the source of the interrupt - error, break, or received data.
2. Reads the data from the UART Receive Data Register if the interrupt was because of data available. If operating in MULTIPROCESSOR (9-bit) Mode, further actions may be required depending on the MULTIPROCESSOR Mode bits MPMD[1:0].
3. Clears the UART Receiver interrupt in the applicable Interrupt Request Register.
4. Executes the IRET instruction to return from the interrupt-service routine and await more data.

Clear To Send ($\overline{\text{CTS}}$) Operation

The CTS pin, if enabled by the CTSE bit of the UART Control 0 Register, performs flow control on the outgoing transmit datastream. The Clear To Send ($\overline{\text{CTS}}$) input pin is sampled one system clock before beginning any new character transmission. To delay transmission of the next data character, an external receiver must deassert $\overline{\text{CTS}}$ at least one system clock cycle before a new data transmission begins. For multiple character transmissions, this action is typically performed during Stop Bit transmission. If $\overline{\text{CTS}}$ deasserts in the middle of a character transmission, the current character is sent completely.

MULTIPROCESSOR (9-bit) Mode

The UART features a MULTIPROCESSOR (9-bit) Mode that uses an extra (9th) bit for selective communication when a number of processors share a common UART bus. In MULTIPROCESSOR Mode (also referred to as *9-bit Mode*), the multiprocessor bit (MP) is transmitted immediately following the 8-bits of data and immediately preceding the Stop bit(s) as displayed in Figure 13. The character format is:

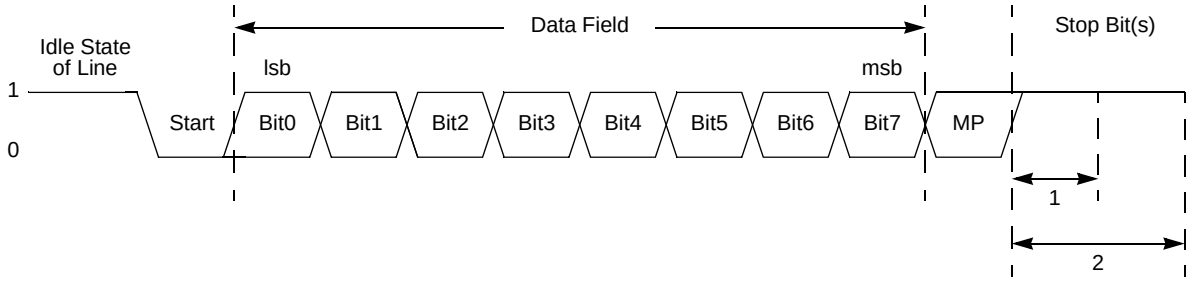


Figure 13. UART Asynchronous MULTIPROCESSOR Mode Data Format

Bit	Description (Continued)
[2] TDRE	**TDRE—Transmitter Data Register Empty** This bit indicates that the UART Transmit Data Register is empty and ready for additional data. Writing to the UART Transmit Data Register resets this bit. 0 = Do not write to the UART Transmit Data Register. 1 = The UART Transmit Data Register is ready to receive an additional byte to be transmitted.
[1] TXE	**Transmitter Empty** This bit indicates that the Transmit Shift Register is empty and character transmission is finished. 0 = Data is currently transmitting. 1 = Transmission is complete.
[0] CTS	**$\overline{\text{CTS}}$ Signal** When this bit is read it returns the level of the $\overline{\text{CTS}}$ signal. This signal is active Low.

UART Status 1 Register

This register contains multiprocessor control and status bits.

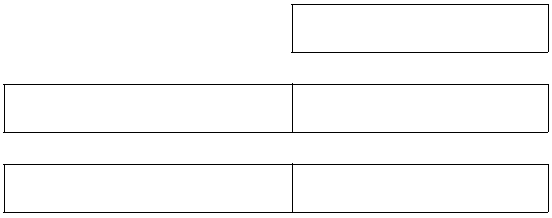
Table 66. UART Status 1 Register (U0STAT1)

Bit	7	6	5	4	3	2	1	0
Field	Reserved						NEWFRM	MPRX
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R
Address	F44H							

Bit	Description
[7:2]	**Reserved** These bits are reserved and must be programmed to 000000.
[1] NEWFRM	**New Frame** A status bit denoting the start of a new frame. Reading the UART Receive Data Register resets this bit to 0. 0 = The current byte is not the first data byte of a new frame. 1 = The current byte is the first data byte of a new frame.
[0] MPRX	**Multiprocessor Receive** Returns the value of the most recent multiprocessor bit received. Reading from the UART Receive Data Register resets this bit to 0.

UART Transmit Data Register

Data bytes written to the UART Transmit Data (UxTXD) Register, shown in Table 67, are shifted out on the TXDx pin. The Write-only UART Transmit Data Register shares a Register File address with the read-only UART Receive Data Register.



Compensation Steps:

1. Correct for Offset:

ADC MSB	ADC LSB

–

Offset MSB	Offset LSB

=

#1 MSB	#1 LSB

2. Compute the absolute value of the offset-corrected ADC value *if negative*; the gain correction factor is computed assuming positive numbers, with sign restoration afterward.

#2 MSB	#2 LSB

Also compute the absolute value of the gain correction word, if negative.

AGain MSB	AGain LSB

3. Multiply by the Gain Correction Word. If operating in DIFFERENTIAL Mode, there are two gain correction values: one for positive ADC values, another for negative ADC values. Use the appropriate Gain Correction Word based on the sign computed by byte #2.

#2 MSB	#2 LSB

*

AGain MSB	AGain LSB

=

Output Data

The output format of the corrected ADC value is shown below.

MSB								LSB							
s	v	b	a	9	8	7	6	5	4	3	2	1	0	–	–

The overflow bit in the corrected output indicates that the computed value was greater than the maximum logical value (+1023) or less than the minimum logical value (–1024). Unlike the hardware overflow bit, this is not a simple binary flag. For a normal (nonoverflow) sample, the sign and the overflow bit match. If the sign bit and overflow bit do not match, a computational overflow has occurred.

Input Buffer Stage

Many applications require the measurement of an input voltage source with a high output impedance. This ADC provides a buffered input for such situations. The drawback of the buffered input is a limitation of the input range. When using unity gain buffered mode, the input signal must be prevented from coming too close to either V_{SS} or V_{DD}. See Table 139 on page 236 for details.

This condition applies only to the input voltage level (with respect to ground) of each differential input signal. The actual differential input voltage magnitude may be less than 300 mV.

The input range of the unbuffered ADC swings from V_{SS} to V_{DD}. Input signals smaller than 300 mV must use the unbuffered input mode. If these signals do not contain low output impedances, they might require off-chip buffering.

Signals outside the allowable input range can be used without instability or device damage. Any ADC readings made outside the input range are subject to greater inaccuracy than specified.

ADC Control Register Definitions

This section defines the features of the following ADC Control registers.

ADC Control Register 0 (ADCCTL0): see page 134

ADC Control/Status Register 1 (ADCCTL1): see page 136

ADC Data High Byte Register (ADCD_H): see page 137

ADC Data Low Byte Register (ADCD_L): see page 137

Figure 22 displays a basic Flash Controller flow. The following subsections provide details about the various operations displayed in Figure 22.

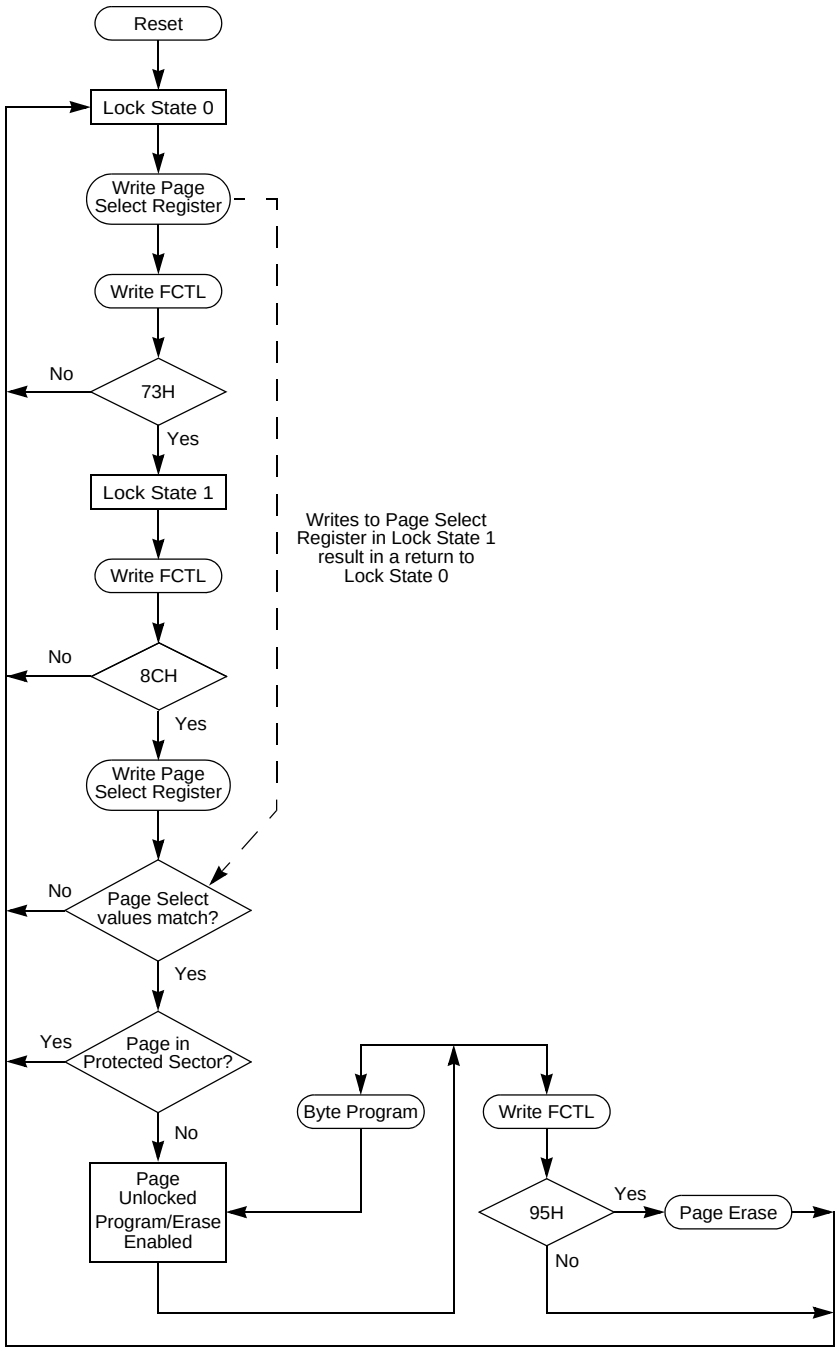


Figure 22. Flash Controller Operation Flow Chart

Flash Controller Behavior in DEBUG Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored.
- The Flash Sector Protect Register is ignored for programming and erase operations.
- Programming operations are not limited to the page selected in the Page Select Register.
- Bits in the Flash Sector Protect Register can be written to one or zero.
- The second write of the Page Select Register to unlock the Flash Controller is not necessary.
- The Page Select Register can be written when the Flash Controller is unlocked.
- The Mass Erase command is enabled through the Flash Control Register.

! Caution: For security reasons, the Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the flash controller must go through the unlock sequence again to select another page.

Flash Control Register Definitions

This section defines the features of the following Flash Control registers.

Flash Control Register: see page 153

Flash Status Register: see page 155

Flash Page Select Register: see page 156

Flash Sector Protect Register: see page 157

Flash Frequency High and Low Byte Registers: see page 157

Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FCTL) Register before programming or erasing the Flash memory. Writing the sequence 73H 8CH, sequentially, to the Flash Control Register unlocks the Flash Controller. When the Flash Controller is unlocked, the Flash memory can be enabled for Mass Erase or Page Erase by writing the appropriate enable command to the FCTL. Page Erase applies only to the active page selected in Flash Page Select Register. Mass Erase is enabled only through the On-Chip

! **Caution:** The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device. Also, Flash programming and erasure is not supported for system clock frequencies below 20 kHz or above 20 MHz.

Table 84. Flash Frequency High Byte Register (FFREQH)

Bit	7	6	5	4	3	2	1	0
Field	FFREQH							
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
Address	FFAH							

Bit	Description
[7:0] FFREQH	**Flash Frequency High Byte** High byte of the 16-bit Flash Frequency value.

Table 85. Flash Frequency Low Byte Register (FFREQL)

Bit	7	6	5	4	3	2	1	0
Field	FFREQL							
RESET	0							
R/W	R/W							
Address	FFBH							

Bit	Description
[7:0] FFREQL	**Flash Frequency Low Byte** Low byte of the 16-bit Flash Frequency value.

Flash Program Memory Address 0001H

Table 89. Flash Options Bits at Program Memory Address 0001H

Bit	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
Address	Program Memory 0001H							

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

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 111.
[4] XTLDIS	**State of the Crystal Oscillator at Reset** This bit only enables the crystal oscillator. Its selection as a system clock must be performed 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. **Caution:** Programming the XTLDIS bit to zero on 8-pin versions of this device prevents any further communication via the debug pin 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 further debugging or Flash programming is not required.
[3:0]	**Reserved** These bits are reserved and must be programmed to 1111.

Randomized Lot Identifier

Table 104. Lot Identification Number (RAND_LOT)

Bit	7	6	5	4	3	2	1	0
Field	RAND_LOT							
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
Address	Interspersed throughout Information Page Memory							

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

Bit	Description
[7] RAND_LOT	**Randomized Lot ID** The randomized lot ID is a 32-byte binary value that changes for each production lot. See Table 105.

Table 105. Randomized Lot ID Locations

Info Page Address	Memory Address	Usage
3C	FE3C	Randomized Lot ID Byte 31 (most significant).
3D	FE3D	Randomized Lot ID Byte 30.
3E	FE3E	Randomized Lot ID Byte 29.
3F	FE3F	Randomized Lot ID Byte 28.
58	FE58	Randomized Lot ID Byte 27.
59	FE59	Randomized Lot ID Byte 26.
5A	FE5A	Randomized Lot ID Byte 25.
5B	FE5B	Randomized Lot ID Byte 24.
5C	FE5C	Randomized Lot ID Byte 23.
5D	FE5D	Randomized Lot ID Byte 22.
5E	FE5E	Randomized Lot ID Byte 21.
5F	FE5F	Randomized Lot ID Byte 20.
61	FE61	Randomized Lot ID Byte 19.
62	FE62	Randomized Lot ID Byte 18.
64	FE64	Randomized Lot ID Byte 17.
65	FE65	Randomized Lot ID Byte 16.
67	FE67	Randomized Lot ID Byte 15.
68	FE68	Randomized Lot ID Byte 14.

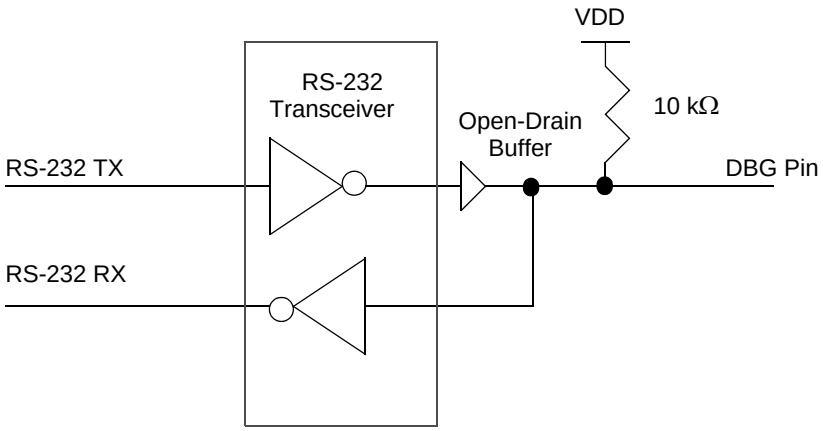


Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #2 of 2

DEBUG Mode

The operating characteristics of the devices in DEBUG Mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions
- The system clock operates unless in STOP Mode
- All enabled on-chip peripherals operate unless in STOP Mode
- Automatically exits HALT Mode
- Constantly refreshes the Watchdog Timer, if enabled

Entering DEBUG Mode

The operating characteristics of the devices entering DEBUG Mode are:

- The device enters DEBUG Mode after the eZ8 CPU executes a BRK (Breakpoint) instruction
- If the DBG pin is held Low during the final clock cycle of system reset, the part enters DEBUG Mode immediately (20-/28-pin products only)

Note: Holding the DBG pin Low for an additional 5000 (minimum) clock cycles after reset (making sure to account for any specified frequency error if using an internal oscillator) prevents a false interpretation of an Autobaud sequence (see the OCD Auto-Baud Detector/Generator section on page 183).

Assembly Language Source Program Example

```
JP START          ; Everything after the semicolon is a comment.
START:            ; A label called 'START'. The first instruction (JP START) in this
                  ; example causes program execution to jump to the point within the
                  ; program where the START label occurs.
LD R4, R7         ; A Load (LD) instruction with two operands. The first operand,
                  ; Working Register R4, is the destination. The second operand,
                  ; Working Register R7, is the source. The contents of R7 is
                  ; written into R4.
LD 234H, #%01     ; Another Load (LD) instruction with two operands.
                  ; The first operand, Extended Mode Register Address 234H,
                  ; identifies the destination. The second operand, Immediate Data
                  ; value 01H, is the source. The value 01H is written into the
                  ; Register at address 234H.
```

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as 'destination, source'. After assembly, the object code usually has the operands in the order 'source, destination', but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed if manual program coding is preferred or if you intend to implement your own assembler.

Example 1. If the contents of registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 116. Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H,	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2. In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0–R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 117. Assembly Language Syntax Example 2

Assembly Language Code	ADD	43H,	R8	(ADD dst, src)
Object Code	04	E8	43	(OPC src, dst)

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 4 KB Flash, 10-Bit Analog-to-Digital Converter											
Standard Temperature: 0°C to 70°C											
Z8F042APB020SG	4 KB	1KB	128 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F042AQB020SG	4 KB	1KB	128 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F042ASB020SG	4 KB	1KB	128 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F042ASH020SG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F042AHH020SG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F042APH020SG	4 KB	1KB	128 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F042ASJ020SG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F042AHJ020SG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F042APJ020SG	4 KB	1KB	128 B	23	20	2	8	1	1	1	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F042APB020EG	4 KB	1KB	128 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F042AQB020EG	4 KB	1KB	128 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F042ASB020EG	4 KB	1KB	128 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F042ASH020EG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F042AHH020EG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F042APH020EG	4 KB	1KB	128 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F042ASJ020EG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F042AHJ020EG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F042APJ020EG	4 KB	1KB	128 B	23	20	2	8	1	1	1	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 4 KB Flash											
Standard Temperature: 0°C to 70°C											
Z8F041APB020SG	4 KB	1KB	128 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F041AQB020SG	4 KB	1KB	128 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F041ASB020SG	4 KB	1KB	128 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F041ASH020SG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F041AHH020SG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F041APH020SG	4 KB	1KB	128 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F041ASJ020SG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F041AHJ020SG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F041APJ020SG	4 KB	1KB	128 B	25	19	2	0	1	1	0	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F041APB020EG	4 KB	1KB	128 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F041AQB020EG	4 KB	1KB	128 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F041ASB020EG	4 KB	1KB	128 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F041ASH020EG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F041AHH020EG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F041APH020EG	4 KB	1KB	128 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F041ASJ020EG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F041AHJ020EG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F041APJ020EG	4 KB	1KB	128 B	25	19	2	0	1	1	0	PDIP 28-pin package