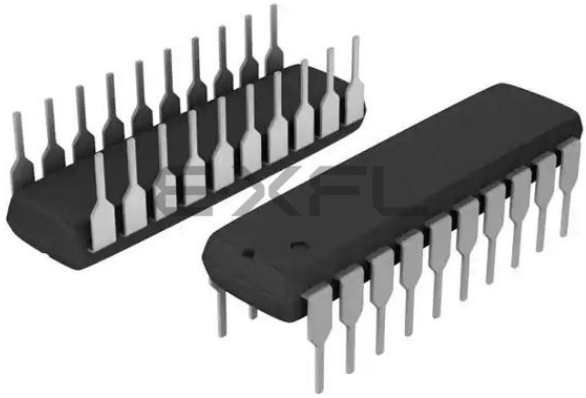


Welcome to E-XFL.COM

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	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	16 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f011aph020sc

General Purpose I/O Port Output Timing 236
On-Chip Debugger Timing 237
UART Timing 238

Packaging 241

Ordering Information 251

Index 261

Customer Support 271

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 **XOUT** pin to form the oscillator. In addition, this pin is used with external RC networks or external clock drivers to provide the system clock.
XOUT	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.*
Reset		
$\overline{\text{RESET}}$	I/O	RESET. Generates a Reset when asserted (driven Low). Also serves as a reset indicator; the Z8 Encore! XP forces this pin low when in reset. This pin is open-drain and features an enabled internal pull-up resistor.

addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 5 describes the Program Memory Maps for the Z8 Encore! XP F082A Series products.

Table 5. Z8 Encore! XP F082A Series Program Memory Maps

Program Memory Address (Hex)	Function
Z8F082A and Z8F081A Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–0039	Reserved
003A–003D	Oscillator Fail Trap Vectors
003E–1FFF	Program Memory
Z8F042A and Z8F041A Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–0039	Reserved
003A–003D	Oscillator Fail Trap Vectors
003E–0FFF	Program Memory

Port A–D Address Registers

The Port A–D Address registers select the GPIO Port functionality accessible through the Port A–D Control registers. The Port A–D Address and Control registers combine to provide access to all GPIO Port controls (Table 17).

Table 17. Port A–D GPIO Address Registers (P*x*ADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	PADDR[7:0]							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FD0H, FD4H, FD8H, FDCH							

PADDR[7:0]—Port Address
The Port Address selects one of the sub-registers accessible through the Port Control register.

PADDR[7:0]	Port Control sub-register accessible using the Port A–D Control Registers
00H	No function. Provides some protection against accidental Port reconfiguration.
01H	Data Direction.
02H	Alternate Function.
03H	Output Control (Open-Drain).
04H	High Drive Enable.
05H	Stop Mode Recovery Source Enable.
06H	Pull-up Enable.
07H	Alternate Function Set 1.
08H	Alternate Function Set 2.
09H–FFH	No function.

Port A–D Control Registers

The Port A–D Control registers set the GPIO port operation. The value in the corresponding Port A–D Address register determines which sub-register is read from or written to by a Port A–D Control register transaction (Table 18).

U0RXI—UART 0 Receiver Interrupt Request
0 = No interrupt request is pending for the UART 0 receiver.
1 = An interrupt request from the UART 0 receiver is awaiting service.

U0TXI—UART 0 Transmitter Interrupt Request
0 = No interrupt request is pending for the UART 0 transmitter.
1 = An interrupt request from the UART 0 transmitter is awaiting service.

ADCI—ADC Interrupt Request
0 = No interrupt request is pending for the analog-to-digital Converter.
1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register (Table 34) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 1 register to determine if any interrupt requests are pending.

Table 34. Interrupt Request 1 Register (IRQ1)

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VI	PA6CI	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
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	FC3H							

PA*7VI*—Port A Pin 7 or LVD Interrupt Request
0 = No interrupt request is pending for GPIO Port A or LVD.
1 = An interrupt request from GPIO Port A or LVD.

PA6CI—Port A Pin 6 or Comparator Interrupt Request
0 = No interrupt request is pending for GPIO Port A or Comparator.
1 = An interrupt request from GPIO Port A or Comparator.

PA*x*I—Port A Pin *x* Interrupt Request
0 = No interrupt request is pending for GPIO Port A pin *x*.
1 = An interrupt request from GPIO Port A pin *x* is awaiting service.

where *x* indicates the specific GPIO Port pin number (0–5).

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) register (Table 35) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 2 register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 2 Register (IRQ2)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved				PC3I	PC2I	PC1I	PC0I
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	FC6H							

Reserved—Must be 0.

PC*x*I—Port C Pin *x* Interrupt Request
0 = No interrupt request is pending for GPIO Port C pin *x*.
1 = An interrupt request from GPIO Port C pin *x* is awaiting service.

where *x* indicates the specific GPIO Port C pin number (0–3).

IRQ0 Enable High and Low Bit Registers

Table 36 describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers (Table 37 and Table 38) form a priority encoded enabling for interrupts in the Interrupt Request 0 register.

Table 36. IRQ0 Enable and Priority Encoding

IRQ0ENH[*x*]	IRQ0ENL[*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.

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. Follow the steps below 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.
   - Set or clear CTSE to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.
8. Execute an EI instruction to enable interrupts.

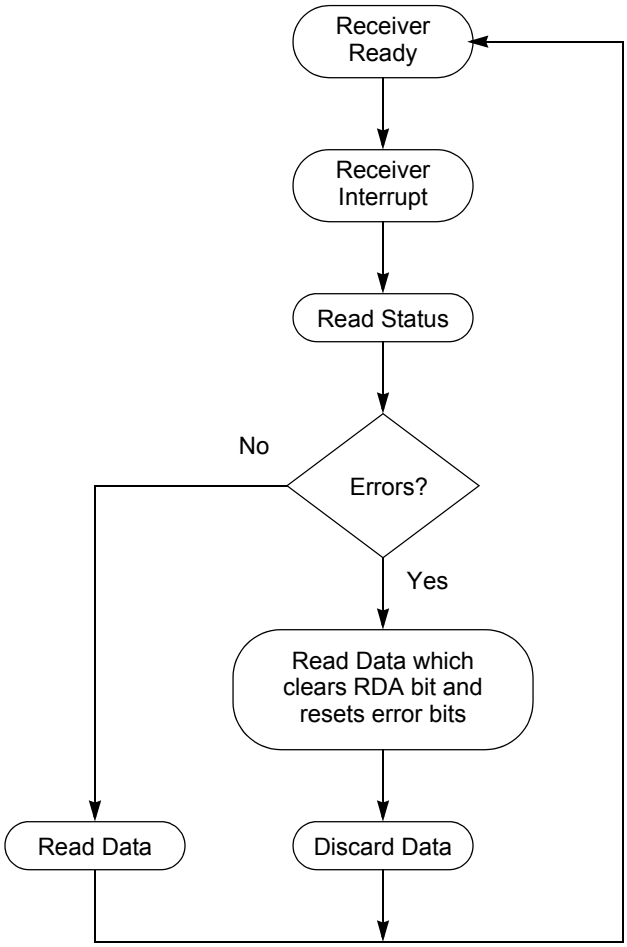


Figure 15. UART Receiver Interrupt Service Routine Flow

Baud Rate Generator Interrupts

If the baud rate generator (BRG) interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This condition allows the Baud Rate Generator to function as an additional counter if the UART functionality is not employed.

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UART Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value

(BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. The UART data rate is calculated using the following equation:

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

When the UART is disabled, the Baud Rate Generator functions as a basic 16-bit timer with interrupt on time-out. Follow the steps below to configure the Baud Rate Generator as a timer with interrupt on time-out:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 register to 0.
2. Load the acceptable 16-bit count value into the UART Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BRGCTL bit in the UART Control 1 register to 1.

When configured as a general purpose timer, the interrupt interval is calculated using the following equation:

$$\text{Interrupt Interval(s)} = \text{System Clock Period (s)} \times \text{BRG[15:0]}$$

UART Control Register Definitions

The UART control registers support the UART and the associated Infrared Encoder/Decoders. For more information on infrared operation, see Infrared Encoder/Decoder on page 117.

UART Control 0 and Control 1 Registers

The UART Control 0 (UxCTL0) and Control 1 (UxCTL1) registers (Table 61 and Table 62) configure the properties of the UART's transmit and receive operations. The UART Control registers must not be written while the UART is enabled.

Table 61. UART Control 0 Register (U0CTL0)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
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	F42H							

TEN—Transmit Enable
This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal

Infrared Encoder/Decoder

The Z8 Encore! XP® F082A Series products contain a fully-functional, high-performance UART to Infrared Encoder/Decoder (Endec). The Infrared Endec is integrated with an on-chip UART to allow easy communication between the Z8 Encore! and IrDA Physical Layer Specification, Version 1.3-compliant infrared transceivers. Infrared communication provides secure, reliable, low-cost, point-to-point communication between PCs, PDAs, cell phones, printers, and other infrared enabled devices.

Architecture

Figure 16 displays the architecture of the Infrared Endec.

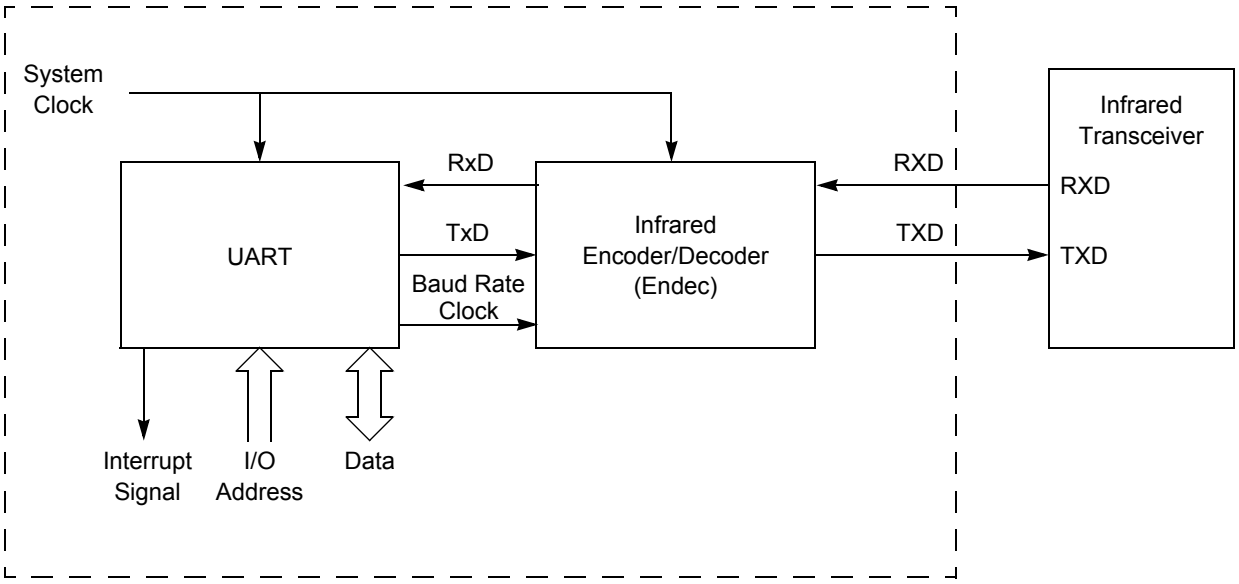


Figure 16. Infrared Data Communication System Block Diagram

Operation

When the Infrared Endec is enabled, the transmit data from the associated on-chip UART is encoded as digital signals in accordance with the IrDA standard and output to the infrared transceiver through the TXD pin. Likewise, data received from the infrared transceiver is passed to the Infrared Endec through the RXD pin, decoded by the Infrared

Operation

The Flash Controller programs and erases Flash memory. The Flash Controller provides the proper Flash controls and timing for Byte Programming, Page Erase, and Mass Erase of Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels.

The Flow Chart in Figure 22 displays basic Flash Controller operation. The following subsections provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select, Page Erase, and Mass Erase) displayed in Figure 22.

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.

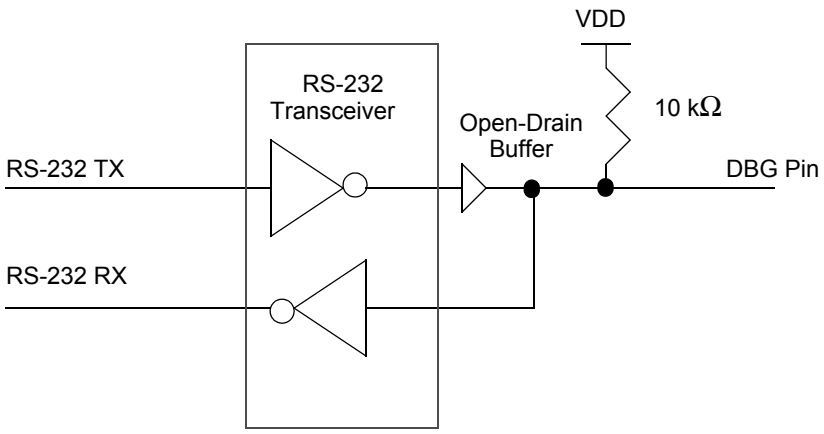


Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (2)

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* OCD Auto-Baud Detector/ Generator *on page 176).*

Table 111. Transconductance Values for Low, Medium, and High Gain Operating Modes

Mode	Crystal Frequency Range	Function	Transconductance (mA/V) Use this range for calculations		
Low Gain*	32 kHz–1 MHz	Low Power/Frequency Applications	0.02	0.04	0.09
Medium Gain*	0.5 MHz–10 MHz	Medium Power/Frequency Applications	0.84	1.7	3.1
High Gain*	8 MHz–20 MHz	High Power/Frequency Applications	1.1	2.3	4.2

Note: *Printed circuit board layout must not add more than 4 pF of stray capacitance to either XIN or XOUT pins. if no Oscillation occurs, reduce the values of the capacitors C1 and C2 to decrease the loading.

Oscillator Operation with an External RC Network

Figure 28 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

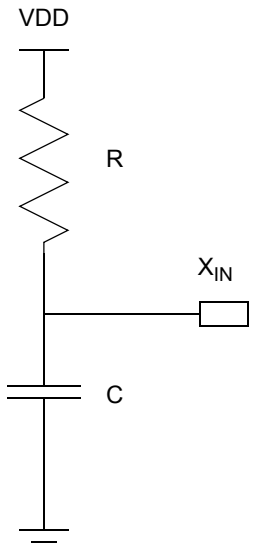
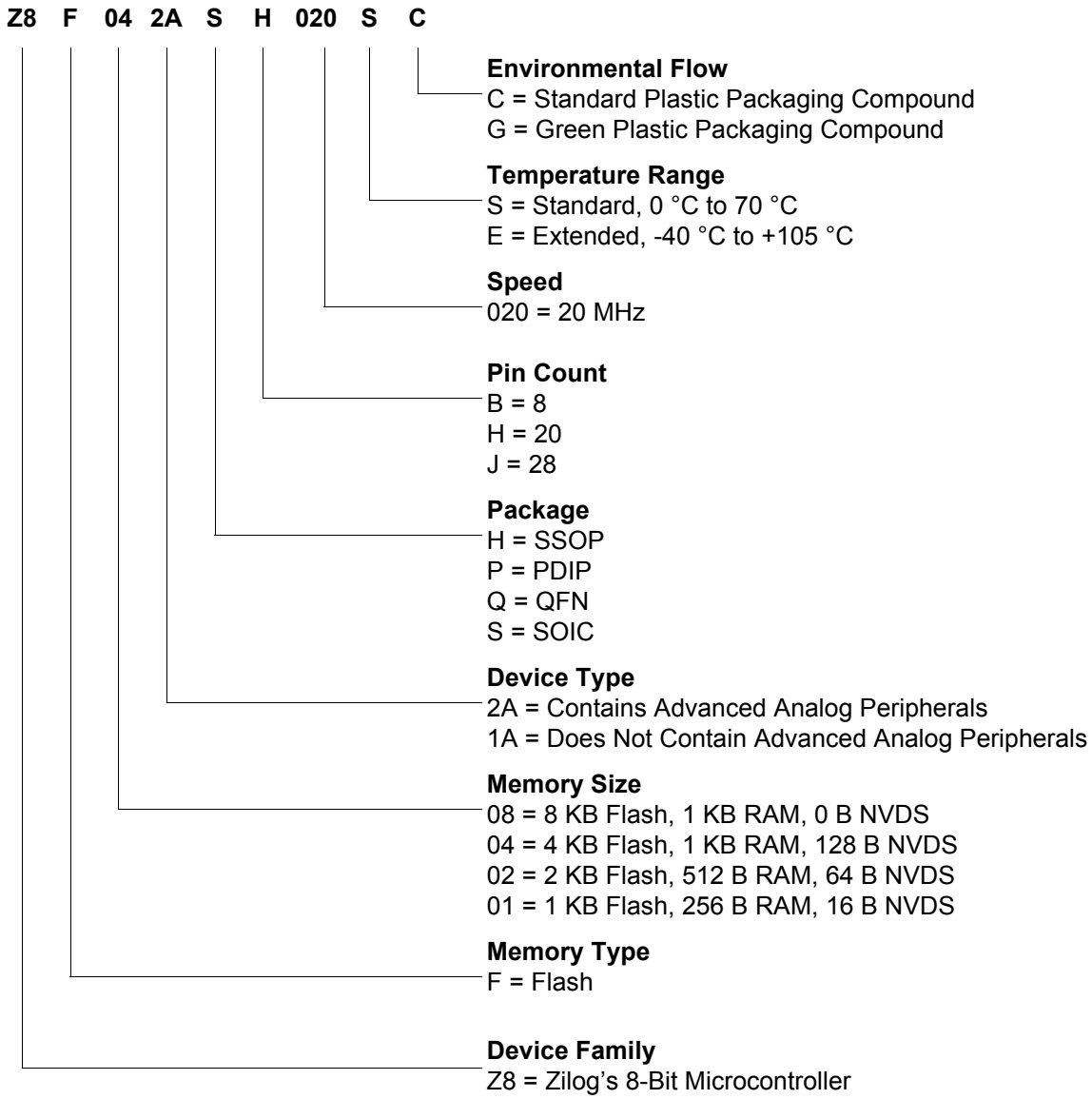


Figure 28. Connecting the On-Chip Oscillator to an External RC Network

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

$$\text{Oscillator Frequency (kHz)} = \frac{1\times10^{6}}{(0.4\times R\times C)+(4\times C)}$$

Part Number Suffix Designations



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-polled method 101

subtract with carry - extended addressing 203
SUBX 203
SWAP 207
swap nibbles 207
symbols, additional 202

T

TCM 204
TCMX 204
Technical Support 271
test complement under mask 204
test complement under mask - extended addressing 204
test under mask 204
test under mask - extended addressing 204
timer signals 11
timers 69
  architecture 69
  block diagram 70
  CAPTURE mode 77, 78, 85, 86
  CAPTURE/COMPARE mode 81, 85
  COMPARE mode 79, 85
  CONTINUOUS mode 71, 84
  COUNTER mode 72, 73
  COUNTER modes 84
  GATED mode 80, 85
  ONE-SHOT mode 70, 84
  operating mode 70
  PWM mode 74, 76, 85
  reading the timer count values 82
  reload high and low byte registers 87
  timer control register definitions 83
  timer output signal operation 82
timers 0-3
  control registers 83, 84
  high and low byte registers 87, 88
TM 204
TMX 204
transmit
  IrDA data 118
transmitting UART data-polled method 99
transmitting UART dat-interrupt-driven method 100
TRAP 206

U

UART 7
  architecture 97
  baud rate generator 107
  baud rates table 115
  control register definitions 108
  controller signals 11
  data format 98
  interrupts 105
  multiprocessor mode 103
  receiving data using interrupt-driven method 102
  receiving data using the polled method 101
  transmitting data usin the interrupt-driven method 100
  transmitting data using the polled method 99
  x baud rate high and low registers 114
  x control 0 and control 1 registers 108
  x status 0 and status 1 registers 111, 112
UxBRH register 114
UxBRL register 114
UxCTL0 register 108, 114
UxCTL1 register 109
UxRXD register 113
UxSTAT0 register 111
UxSTAT1 register 112
UxTXD register 113

V

vector 201
Voltage Brownout reset (VBR) 26

W

Watchdog Timer
  approximate time-out delay 91
  approximate time-out delays 135
  CNTL 26
  control register 94, 136, 190