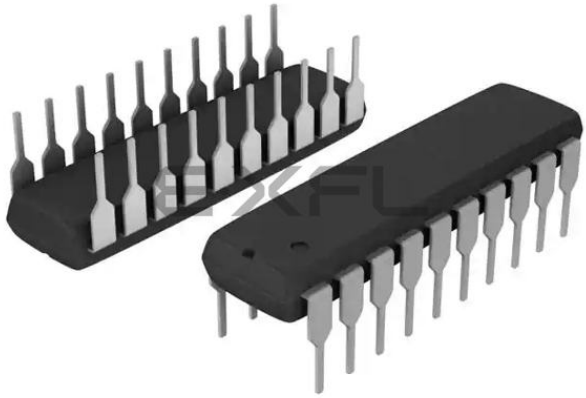

Welcome to E-XFL.COM

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	-
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
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/z8f0831ph020sg

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES or TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES or TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Chapter/Section	Description	Page No.
Dec 2012	13	GPIO	Modified GPIO Port D0 language in Shared Reset Pin section and Port Alternate Function Mapping table.	35, 36
Sep 2011	12	LED Drive Enable Register	Clarified statement surrounding the Alternate Function Register as it relates to the LED function; revised Sector Based Flash Protection description; revised Packaging chapter.	51, 115, 199
Dec 2007	11	n/a	Updated all instances of *Z8 Encore! XP F0830* to *Z8 Encore! F0830*.	All
Nov 2007	10	DC Characteristics, On-Chip Peripheral AC and DC Electrical Characteristics	Updated Tables 116 and and 122.	185, 193
Sep 2007	09	Timers, PWM SINGLE OUTPUT Mode, PWM DUAL OUTPUT Mode, Analog-to-Digital Converter, Reference Buffer.	Updated Figures 2 and 4, Table 4.	8, 9, 11, 68, 74, 75, 98, 101
Apr 2007	08	Optimizing NVDS Memory Usage for Execution Speed, On-Chip Peripheral AC and DC Electrical Characteristics	Added a note under Table 93 in Nonvolatile Data Storage chapter. Updated Table 121 and Table 122 in Electrical Characteristics chapter. Other style updates.	137, 193, 193
Dec 2006	07	General Purpose Input/Output	Added PD0 in Table 16.	38
		Overview, Interrupt Controller	Changed the number of interrupts to 17.	1,5, 53
		Nonvolatile Data Storage	Updated chapter.	136
		Oscillator Control Register Definitions, AC Characteristics, On-Chip Peripheral AC and DC Electrical Characteristics	Updated Tables 117 and 122. Added Figure 24.	156, 189, 193
		Ordering Information	Updated Part Number Suffix Designations.	205
		n/a	Removed *Preliminary* stamp from footer.	All

Part Selection Guide

Table 1 lists the basic features available for each device within the Z8 Encore! F0830 Series product line. See the Ordering Information chapter on page 200 for details.

Table 1. Z8 Encore! F0830 Series Family Part Selection Guide

Part Number	Flash (KB)	RAM (B)	NVDS (64B)	ADC
Z8F1232	12	256	No	Yes
Z8F1233	12	256	No	No
Z8F0830	8	256	Yes	Yes
Z8F0831	8	256	Yes	No
Z8F0430	4	256	Yes	Yes
Z8F0431	4	256	Yes	No
Z8F0230	2	256	Yes	Yes
Z8F0231	2	256	Yes	No
Z8F0130	1	256	Yes	Yes
Z8F0131	1	256	Yes	No

Program Memory

The eZ8 CPU supports 64KB of program memory address space. The Z8 Encore! F0830 Series devices contain 1KB to 12KB of on-chip Flash memory in the program memory address space, depending on the device. Reading from program memory addresses outside the available Flash memory address range returns FFH. Writing to these unimplemented program memory addresses produces no effect. Table 6 shows a program memory map for the Z8 Encore! F0830 Series products.

Table 6. Z8 Encore! F0830 Series Program Memory Maps

Program Memory Address (Hex)	Function
Z8F0830 and Z8F0831 Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–003D	Interrupt Vectors*
003E–1FFF	Program Memory
Z8F0430 and Z8F0431 Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–003D	Interrupt Vectors*
003E–0FFF	Program Memory
Z8F0130 and Z8F0131 Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–003D	Interrupt Vectors*
003E–03FF	Program Memory
Z8F0230 and Z8F0231 Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–003D	Interrupt Vectors*
003E–07FF	Program Memory

Note: *See Table 34 on page 54 for a list of interrupt vectors.

External Clock Setup

For systems using an external TTL drive, PB3 is the clock source for 20- and 28-pin devices. In this case, configure PB3 for Alternate function CLKIN. Write to the Oscillator Control Register (see the Oscillator Control Register Definitions section on page 154) to select the PB3 as the system clock.

Table 16. Port Alternate Function Mapping

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A[1]	PA0	T0IN/$\overline{\text{T0OUT}}$	Timer 0 input/Timer 0 output complement	N/A
		Reserved		
	PA1	T0OUT	Timer 0 output	
		Reserved		
	PA2	Reserved	Reserved	
		Reserved		
	PA3	Reserved	Reserved	
		Reserved		
	PA4	Reserved	Reserved	
		Reserved		
	PA5	Reserved	Reserved	
		Reserved		
	PA6	T1IN/$\overline{\text{T1OUT}}$	Timer 1 input/Timer 1 output complement	
		Reserved		
	PA7	T1OUT	Timer 1 output	
		Reserved		

Notes:
1. Because there is only a single alternate function for each Port A and Port D (PD0) pin, the Alternate Function Set registers are not implemented for Port A and Port D (PD0). Enabling alternate function selections (as described in the Port A–D Alternate Function Subregisters section on page 42) automatically enables the associated alternate function.
2. Because there are at most two choices of alternate functions for any Port B pin, the AFS2 Alternate Function Set Register is implemented but is not used to select the function. Additionally, alternate function selection (as described in the Port A–D Alternate Function Subregisters section on page 42) must also be enabled.
3. Because there are at most two choices of alternate functions for any Port C pin, the AFS2 Alternate Function Set Register is implemented but is not used to select the function. Additionally, alternate function selection (as described in the Port A–D Alternate Function Subregisters section on page 42) must also be enabled.

Port A–D Control Registers

The Port A–D Control registers, shown in Table 20, set the GPIO port operation. The value in the corresponding Port A–D Address Register determines which subregister is read from or written to by a Port A–D Control Register transaction.

Table 20. Port A–D Control Registers (P*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD1H, FD5H, FD9H, FDDH							

Bit	Description
[7:0] PCTL	**Port Control** The Port Control Register provides access to all subregisters that configure the GPIO port operation.

Port A–D Data Direction Subregisters

The Port A–D Data Direction Subregister, shown in Table 21, is accessed through the Port A–D Control Register by writing 01H to the Port A–D Address Register.

Table 21. Port A–D Data Direction Subregisters (PxDD)

Bit	7	6	5	4	3	2	1	0
Field	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 01H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] DDx	**Data Direction** These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction Register setting. 0 = Output. Data in the Port A–D Output Data Register is driven onto the port pin. 1 = Input. The port pin is sampled and the value written into the Port A–D Input Data Register. The output driver is tristated.

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

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register, shown in Table 35 stores the interrupt requests for both vectored and polled interrupts. When a request is sent to the Interrupt Controller, the corresponding bit in the IRQ0 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 0 Register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 0 Register (IRQ0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1I	T0I	Reserved				ADCI
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	FC0H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1I	**Timer 1 Interrupt Request** 0 = No interrupt request is pending for timer 1. 1 = An interrupt request from timer 1 is awaiting service.
[5] T0I	**Timer 0 Interrupt Request** 0 = No interrupt request is pending for timer 0. 1 = An interrupt request from timer 0 is awaiting service.
[4:1]	**Reserved** These registers are reserved and must be programmed to 0000.
[0] 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.

IRQ1 Enable High and Low Bit Registers

Table 41 describes the priority control for IRQ1. The IRQ1 Enable High and Low Bit registers, shown in Tables 42 and 43, form a priority-encoded enabling service for interrupts in the Interrupt Request 1 Register. Priority is generated by setting the bits in each register.

Table 41. IRQ1 Enable and Priority Encoding

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

Note: *x* indicates register bits in the address range 7–0.

Table 42. IRQ1 Enable High Bit Register (IRQ1ENH)

Bit	7	6	5	4	3	2	1	0
Field	PA7ENH	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
Address	FC4H							

Bit	Description
[7] PA7ENH	**Port A Bit[7] Interrupt Request Enable High Bit**
[6] PA6CENH	**Port A Bit[7] or Comparator Interrupt Request Enable High Bit**
[5:0] PA*x*ENH	**Port A Bit[*x*] Interrupt Request Enable High Bit** See the interrupt port select register for selection of either Port A or Port D as the interrupt source.

Note: *x* indicates register bits in the address range 5–0.

Interrupt Control Register

The Interrupt Control (IRQCTL) Register, shown in Table 49, contains the master enable bit for all interrupts.

Table 49. Interrupt Control Register (IRQCTL)

Bit	7	6	5	4	3	2	1	0
Field	IRQE	Reserved						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R
Address	FCFH							

Bit	Description
[7] IRQE	**Interrupt Request Enable** This bit is set to 1 by executing an Enable Interrupts (EI) or Interrupt Return (IRET) instruction or by a direct register write of 1 to this bit. It is reset to 0 by executing a `DI` instruction, eZ8 CPU acknowledgement of an interrupt request, reset, or by a direct register write of a 0 to this bit. 0 = Interrupts are disabled. 1 = Interrupts are enabled.
[6:0]	**Reserved** These registers are reserved and must be programmed to 0000000.

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

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 bit 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 bit 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 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 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: the 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) to ensure a time gap between the deassertion of one PWM output to the assertion of its complement.

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

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 the TxCTL1 Register is set to indicate the timer interrupt 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 the TxCTL1 Register clears, indicating that the timer interrupt has not occurred because of an input capture event.

Observe the following steps for configuring a timer for CAPTURE Mode and initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE Mode
   - Set the prescale value
   - 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 reload value.
4. Clear the timer PWM High and Low Byte registers to 0000H. Clearing these registers allows user software to determine if interrupts were generated either by a capture event or by a reload. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt were generated by a reload.

ADC Control Register 0

The ADC Control 0 Register, shown in Table 63, initiates an A/D conversion and provides ADC status information.

Table 63. ADC Control Register 0 (ADCCTL0)

Bit	7	6	5	4	3	2	1	0
Field	START	Reserved	REFEN	ADCEN	Reserved	ANAIN[2:0]		
RESET	0	0	0	0	0	0	0	0
R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F70h							

Bit	Description
[7] START	**ADC Start/Busy** 0 = Writing to 0 has no effect; reading a 0 indicates that the ADC is available to begin a conversion. 1 = Writing to 1 starts a conversion; reading a 1 indicates that a conversion is currently in progress.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] REFEN	**Reference Enable** 0 = Internal reference voltage is disabled allowing an external reference voltage to be used by the ADC. 1 = Internal reference voltage for the ADC is enabled. The internal reference voltage can be measured on the V_{REF} pin.
[4] ADCEN	**ADC Enable** 0 = ADC is disabled for low power operation. 1 = ADC is enabled for normal use.
[3]	**Reserved** This bit is reserved and must be programmed to 0.
[2:0] ANAIN	**Analog Input Select** 000 = ANA0 input is selected for analog to digital conversion. 001 = ANA1 input is selected for analog to digital conversion. 010 = ANA2 input is selected for analog to digital conversion. 011 = ANA3 input is selected for analog to digital conversion. 100 = ANA4 input is selected for analog to digital conversion. 101 = ANA5 input is selected for analog to digital conversion. 110 = ANA6 input is selected for analog to digital conversion. 111 = ANA7 input is selected for analog to digital conversion.

Power Failure Protection

NVDS routines employ error-checking mechanisms to ensure that any power failure will only endanger the most recently written byte. Bytes previously written to the array are not perturbed. For this protection to function, the VBO must be enabled (see the Low-Power Modes chapter on page 30) and configured for a threshold voltage of 2.4V or greater (see *the* Trim Bit Address Space *section on page 129*).

A system reset (such as a pin reset or Watchdog Timer reset) that occurs during a write operation also perturbs the byte currently being written. All other bytes in the array are unperturbed.

Optimizing NVDS Memory Usage for Execution Speed

As indicated in Table 93, the NVDS read time varies drastically; this discrepancy being a trade-off for minimizing the frequency of writes that require post-write page erases. The NVDS read time of address N is a function of the number of writes to addresses other than N since the most recent write to address N as well as the number of writes since the most recent page erase. Neglecting the effects caused by page erases and results caused by the initial condition in which the NVDS is blank, a rule of thumb to consider is that every write since the most recent page erase causes read times of unwritten addresses to increase by 0.8µs up to a maximum of 258µs.

Table 93. NVDS Read Time

Operation	Minimum Latency (µs)	Maximum Latency (µs)
Read	71	258
Write	126	136
Illegal Read	6	6
Illegal Write	7	7

Note: For every 200 writes, a maintenance operation is necessary. In this rare occurrence, the write takes up to 58ms to complete.

If NVDS read performance is critical to your software architecture, you can optimize your code for speed by using either of the two methods listed below.

1. Periodically refresh all addresses that are used; this is the more useful method. The optimal use of NVDS, in terms of speed, is to rotate the writes evenly among all addresses planned for use, thereby bringing all reads closer to the minimum read time.

Table 103. Notational Shorthand (Continued)

Notation	Description	Operand	Range
R	Register	Reg	Reg. represents a number in the range of 00H to FFH
RA	Relative Address	X	X represents an index in the range of +127 to –128 which is an offset relative to the address of the next instruction
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12 or 14
RR	Register Pair	Reg	Reg. represents an even number in the range of 00H to FEH
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to –128 range.

Table 104 contains additional symbols that are used throughout the instruction summary and instruction set description sections.

Table 104. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow, as shown in the following example.

dst ← dst + src

This example indicates that the source data is added to the destination data; the result is stored in the destination location.

eZ8 CPU Instruction Classes

eZ8 CPU instructions can be divided functionally into the following groups:

- Arithmetic
- Bit manipulation
- Block transfer
- CPU control
- Load
- Logical
- Program control
- Rotate and shift

Tables 105 through 112 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instructions can be considered as a subset of more than one category. Within these tables, the source operand is identified as *src*, the destination operand is *dst* and a condition code is *cc*.

Table 105. Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst, src	Add with Carry
ADCX	dst, src	Add with Carry using Extended Addressing
ADD	dst, src	Add
ADDX	dst, src	Add using Extended Addressing
CP	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using Extended Addressing
CPX	dst, src	Compare using Extended Addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment

Low Power Control

For more information about the Power Control Register, see the Power Control Register Definitions section on page 31.

Hex Address: F80

Table 151. Power Control Register 0 (PWRCTL0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			VBO	Reserved	Reserved	COMP	Reserved
RESET	1	0	0	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F80H							

Hex Address: F81

This address range is reserved.

LED Controller

For more information about the LED Drive registers, see the GPIO Control Register Definitions section on page 39.

Hex Address: F82

Table 152. LED Drive Enable (LEDEN)

Bit	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
Address	F82H							