Table R.1. TM4C GPIO registers

Address	Name
$400F.E108	SYSCTL_RCGC2_R
$4000.4000	PORTA base address
$4000.5000	PORTB base address
$4000.6000	PORTC base address
$4000.7000	PORTD base address
$4002.4000	PORTE base address
$4002.5000	PORTF base address
base+$3FC	GPIO_PORTx_DATA_R
base+$400	GPIO_PORTx_DIR_R
base+$420	GPIO_PORTx_AFSEL_R
base+$510	GPIO_PORTx_PUR_R
base+$51C	GPIO_PORTx_DEN_R
base+$524	GPIO_PORTx_CR_R
base+$528	GPIO_PORTx_AMSEL_R

Address	7	6	5	4	3	2	1	0	Name
$400F.E108	-	-	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	SYSCTL_RCGC2_R
$4000.4000									PORTA base address
$4000.5000									PORTB base address
$4000.6000									PORTC base address
$4000.7000									PORTD base address
$4002.4000									PORTE base address
$4002.5000									PORTF base address
base+$3FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTx_DATA_R
base+$400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTx_DIR_R
base+$420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTx_AFSEL_R
base+$510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTx_PUR_R
base+$51C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTx_DEN_R
base+$524	CR	CR	CR	CR	CR	CR	CR	CR	GPIO_PORTx_CR_R
base+$528	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	GPIO_PORTx_AMSEL_R

	31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0	
base+$52C	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTx_PCTL_R
base+$520	LOCK (32 bits)								GPIO_PORTx_LOCK_R

- **GPIO_PORTx_DATA_R**: GPIO data register
- **GPIO_PORTx_DIR_R**: GPIO data direction register (0: input; 1: output)
- **GPIO_PORTx_AFSEL_R**: GPIO mode control select register. If a bit is clear, the pin is used as a GPIO and is controlled by the GPIO registers. Setting a bit in this register configures the corresponding GPIO line to be controlled by an associated peripheral.
- **GPIO_PORTx_PUR_R**: GPIO pull-up control register (0: weak pull-up resistor disabled; 1: weak pull-up resistor enabled)

- **GPIO_PORTx_DEN_R:** GPIO digital enable register (0: The digital functions for the corresponding pin are disabled; 1: The digital functions for the corresponding pin are enabled.)
- **GPIO_PORTx_CR_R:** GPIO commit register (0: The corresponding GPIOAFSEL, GPIOPUR, GPIOPDR, or GPIODEN bits cannot be written; 1: The corresponding GPIOAFSEL, GPIOPUR, GPIOPDR, or GPIODEN bits can be written.)
- **GPIO_PORTx_AMSEL_R:** GPIO analog mode select register (0: The analog function for the corresponding pin is disabled; 1: The analog function for the corresponding pin is enabled.)

Table R. 2. Address Offset used to specify individual data port bits for bit-specific addressing

If we wish to access bit	*Constant*
7	0x0200
6	0x0100
5	0x0080
4	0x0040
3	0x0020
2	0x0010
1	0x0008
0	0x0004

Table R.3. Register information for configuring edge-triggered interrupts

Register Name	Description	Detail
GPIO_IS_R	GPIO Interrupt Sense	IS 0: edge-sensitive; 1: level-sensitive
GPIO_IBE_R	GPIO Interrupt Both Edges	IBE 0: Interrupt generation is controlled by the GPIO_IEV_R register 1: Both edges on the pin triggers an interrupt
GPIO_IEV_R	GPIO Interrupt Event	IEV 0: falling edge triggers an interrupt 1: rising edge triggers an interrupt
GPIO_IM_R	GPIO Interrupt Mask	IME 0: interrupt from the pin is masked (disarmed) 1: interrupt from the pin is armed

Table R.4 and R.5. NVIC registers (the **interrupt number** of an interrupt source determines which register to configure)

Table R.4. NVIC_ENx_R: NVIC device specific interrupt enable register

0x100	EN0	R/W	0x0000.0000	Interrupt 0-31 Set Enable
0x104	EN1	R/W	0x0000.0000	Interrupt 32-63 Set Enable
0x108	EN2	R/W	0x0000.0000	Interrupt 64-95 Set Enable
0x10C	EN3	R/W	0x0000.0000	Interrupt 96-127 Set Enable
0x110	EN4	R/W	0x0000.0000	Interrupt 128-138 Set Enable

Table R. 5. NVIC_PRIx_R: NVIC device specific interrupt priority register

0x404	PRI1	R/W	0x0000.0000	Interrupt 4-7 Priority
0x408	PRI2	R/W	0x0000.0000	Interrupt 8-11 Priority
0x40C	PRI3	R/W	0x0000.0000	Interrupt 12-15 Priority
0x410	PRI4	R/W	0x0000.0000	Interrupt 16-19 Priority
0x414	PRI5	R/W	0x0000.0000	Interrupt 20-23 Priority
0x418	PRI6	R/W	0x0000.0000	Interrupt 24-27 Priority
0x41C	PRI7	R/W	0x0000.0000	Interrupt 28-31 Priority
0x420	PRI8	R/W	0x0000.0000	Interrupt 32-35 Priority
0x424	PRI9	R/W	0x0000.0000	Interrupt 36-39 Priority
0x428	PRI10	R/W	0x0000.0000	Interrupt 40-43 Priority
0x42C	PRI11	R/W	0x0000.0000	Interrupt 44-47 Priority
0x430	PRI12	R/W	0x0000.0000	Interrupt 48-51 Priority
0x434	PRI13	R/W	0x0000.0000	Interrupt 52-55 Priority
0x438	PRI14	R/W	0x0000.0000	Interrupt 56-59 Priority
0x43C	PRI15	R/W	0x0000.0000	Interrupt 60-63 Priority
0x440	PRI16	R/W	0x0000.0000	Interrupt 64-67 Priority
0x444	PRI17	R/W	0x0000.0000	Interrupt 68-71 Priority
0x448	PRI18	R/W	0x0000.0000	Interrupt 72-75 Priority

Interrupt Context Switch procedure

1. The execution of the main program is suspended
    1. the current instruction is finished,
    2. suspend execution and push 8 registers (R0-R3, R12, LR, PC, PSR) on the stack
    3. LR set to 0xFFFFFFE9 (indicates interrupt return)
    4. IPSR set to vector number
    5. sets PC to ISR address
2. The interrupt service routine (ISR) is executed
    1. clears the flag that requested the interrupt

2. performs necessary operations
3. communicates using global variables

3. The main program is resumed when ISR executes BX LR
    1. pulls the 8 registers from the stack

Table R.6. Selected interrupt vectors

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Vector Address or Offset	Description
0-15	-	0x0000.0000 - 0x0000.003C	Processor exceptions
16	0	0x0000.0040	GPIO Port A
17	1	0x0000.0044	GPIO Port B
18	2	0x0000.0048	GPIO Port C
19	3	0x0000.004C	GPIO Port D
20	4	0x0000.0050	GPIO Port E
21	5	0x0000.0054	UART0
22	6	0x0000.0058	UART1
23	7	0x0000.005C	SSI0
24	8	0x0000.0060	I^2C0
25	9	0x0000.0064	PWM0 Fault
26	10	0x0000.0068	PWM0 Generator 0
27	11	0x0000.006C	PWM0 Generator 1
28	12	0x0000.0070	PWM0 Generator 2
29	13	0x0000.0074	QEI0
30	14	0x0000.0078	ADC0 Sequence 0
31	15	0x0000.007C	ADC0 Sequence 1
32	16	0x0000.0080	ADC0 Sequence 2
33	17	0x0000.0084	ADC0 Sequence 3
34	18	0x0000.0088	Watchdog Timers 0 and 1
35	19	0x0000.008C	16/32-Bit Timer 0A
36	20	0x0000.0090	16/32-Bit Timer 0B
37	21	0x0000.0094	16/32-Bit Timer 1A
38	22	0x0000.0098	16/32-Bit Timer 1B
39	23	0x0000.009C	16/32-Bit Timer 2A
40	24	0x0000.00A0	16/32-Bit Timer 2B

UART Baud Rate Generation

- The baud rate divisor (**BRD**) is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part.
- The number formed by these two values is used by the baud-rate generator to determine the bit period.
- The UART generates an internal baud-rate reference clock at 8x or 16x the baud rate (referred to as **Baud8** and **Baud16**)
- The baud rate is configured by two registers
- UART Integer Baud-Rate Divisor (**UART_IBRD**)
- UART Fractional Baud-Rate Divisor (**UART_FBRD**)
- The baud-rate divisor (**BRD**) has the following relationship to the system clock

BRD = BRDI+BRDF = SysClk / (ClkDiv * Baud Rate)

ClkDiv is either **16** or **8**, **BRDI** is the integer part of ***BRD***, **BRDF** is the fractional part of ***BRD***

TivaWare functions

Function:
void SysCtlDelay(uint32_t ui32Count)
Parameters:
ui32Count is the number of delay loop iterations to perform.
Description:
This function provides a means of generating a delay by executing a simple 3 CPU cycle loop a given number of times.

Function:
uint32_t SysCtlClockGet(void)
Description:
This function determines the clock rate of the processor clock (TM4C123 devices)
Returns:
The actual system clock frequency in Hz

Function:
Void SysCtlClockSet(uint32_t ui32Config)
Parameters:
ui32Config is the required configuration of the device clocking.
Description:
This function configures the clocking of the TM4C123 device.