DATA SHEET

GPM6C1015A

LRC Controller with 16KB Mask ROM

Preliminary

Nov. 21, 2013

Version 0.1

Table of Contents

PAGE

LRC Controller With 16KB Mask ROM

1. GENERAL DESCRIPTION

GPM6C1015A is a special chip for learning remote control with 256 bytes built-in SRAM and 16K bytes built-in Mask ROM. It includes three Timers and up to 13 software selectable general I/Os. Additionally, it provides one frequency programmable and duty selectable Pulse Width Modulation (PWM) output for remote control. And it provides build-in capture mode timer for input signal frequency detecting by Infrared learning function. It operates over a wide voltage range of 2.0V - 3.6V @ 8MHz / 4MHz. It has a SLEEP mode for power saving. The power saving mode maintains the RAM contents, but stops the oscillator and causes all other chip functions to be inoperative. The SLEEP mode can be released by using external wakeup sources. In addition, it provides a FREEZE mode for power saves and key board locking when power-supply voltage is detected lower than V_{LVR}. In FREEZE mode, CPU and peripheral were stopped, and all I/Os maintain floating with input function disabled. The FREEZE mode releases when power-supply voltage recover from V_{LVR}. Especially, it has a very accuracy internal OSC, which can match the spec 8MHz / 4MHz ±1.5% (typ.) @ 2.0V~3.6V and can be used at most application. Meanwhile, the build-in IR transfer module can make IR control and usage easier. Using GPM6C1015A does not only share the latest technology, but also the full commitment and technical support from Generalplus.

2. FEATURES

- **CPU**
  - 151 instructions
  - 13 addressing modes
  - Up to 8MHz clock operation
- **Memories**
  - 16K bytes program Mask ROM
  - 256 bytes RAM including stack area .
- **Reset Management**
  - Enhanced reset system
  - Power On Reset (POR)
  - Low Voltage Reset (LVR)
  - Watchdog Reset (WDR)
- **Interrupt Management**
  - 8 internal interrupts
- **I/O Ports**
  - Max 13 multifunction bi-directional I/Os.
  - Each incorporates with pull-up resistor, pull-down resistor or floating input, depending on programmer's settings on the corresponding registers.
  - I/O ports with LED driving capability.
  - I/O ports with 16mA current sink.
- **Clock Management**
  - Internal oscillator: 4/8MHz±1.5% (typ.) @ 2.0V~3.6V
  - Crystal input: 4/8MHz @ 2.0V~3.6V ;
- **Power Management**
  - Two power saving modes: SLEEP, FREEZE mode
- **One Analog Peripheral**
  - LVR : Low Voltage Reset (V_{LVR}=1.85V ± 0.15V).
- **12-bit up count or 8-bit down count Timer (Timer A)**
  - Timer mode with clock source selectable
  - PWM output in carrier signal mode with duty and driver current programmable
  - PWM output in no carrier signal mode with driver current programmable
  - Capture the input signal frequency
  - Detect the signal envelop
- **12-bit up/down count Timer (Timer B)**
  - Timer mode with clock source selectable
  - Timer A's carry signal can be its clock source.
- **Watchdog Timer**
  - Frequency: 0.95Hz @4MHz(System Clock)
  - Frequency: 1.91Hz @8MHz(System Clock)
- **Key Wake up**
  - Normal Key change wake-up from SLEEP mode
  - Key scan wake-up from SLEEP mode
- **IR**
  - Build in IR TX can drive IR LED with 50mA, 100mA 150mA, 200mA current adjustable @ VDD=3.0V & V_{REM}=3.0V.
  - Build in IR RX can supply capture function with sensitivity adjustable. (2uA, 5uA, 8uA, 11uA)

Table 2-1 GPM6C1015A configuration

Part NO.	ROM Type	Voltage (V)	Speed (MHz)	ROM (Byte)	RAM (Byte)	IR Tx/Rx	CCP			CPU OSC.		IO No.	PKG
							CAP	CNT	PWM	INT	XTAL		
GPM6C1015A	Mask	2.0~3.6	4/8	16K	256	Tx/Rx	1	1	1	•	•	13	SOP16

3. BLOCK DIAGRAM

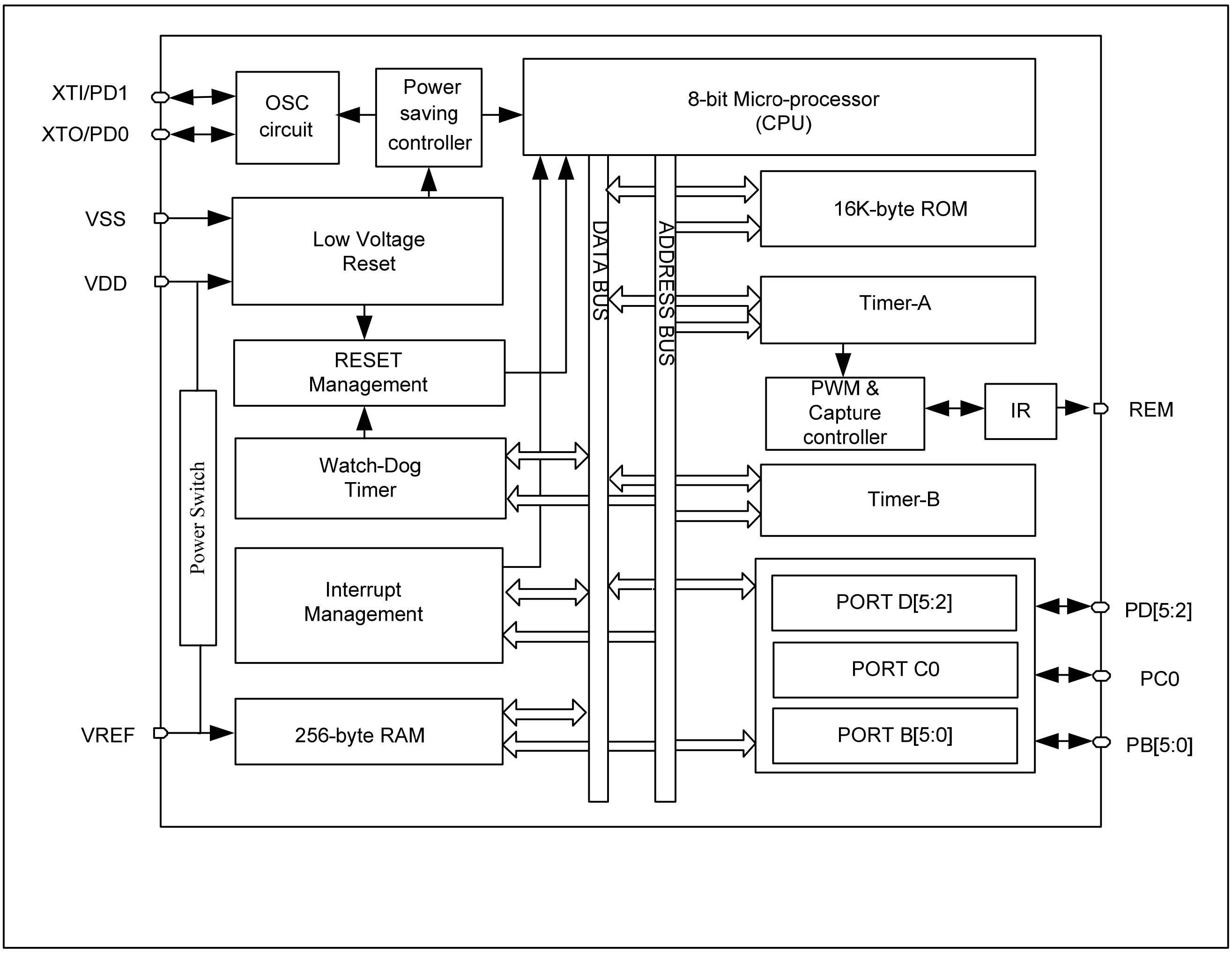


Figure 3-1 Block diagram of GPM6C1015A

4. SIGNAL DESCRIPTIONS

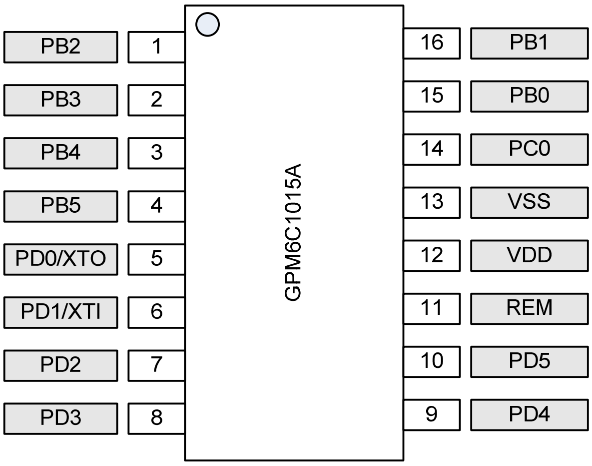
4.1. Pin Description

Type: I = Input, O = Output, S = Supply

Pin Name	GPM6P1015A SOP16	Type	Main Function	Alternate Function
PB5	4	I/O	**PortB[5:0]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Normal wakeup source, if Key is changed, chip wakeup from sleep mode. Key scan wakeup source, if key change is detected, chip wakeup from sleep mode.	
PB4	3	I/O		
PB3	2	I/O		
PB2	1	I/O		
PB1	16	I/O		
PB0	15	I/O		
PC0	14	I/O	**PortC[0]**: Bi-directional programmable Input/Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Key scan wakeup source, if key change is detected, chip wakeup from sleep mode.	
VPP /PD5	10	I/O	**VPP Power Supply**: OTP Program power supply. **PortD[5]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain NMOS output. The sink current (I_{OL}) of this I/O can reach 16 mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Key scan wakeup source, if key change is detected, chip wakes up from sleep mode.	
PD4	9	I/O	**PortD[4:2]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16 mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Key scan wakeup source, if key change is detected, chip wakes up from sleep mode.	
PD3	8	I/O		
PD2	7	I/O		
XTI / PD1	6	I/O	**Crystal Input**: It will be connected with external crystal for a crystal oscillation circuitry in crystal mode. **PortD[1]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough for driving LED.	
XTO /PD0	5	I/O	**Crystal Output**: It is connected with external crystal for a crystal oscillation circuitry in crystal mode. **PortD[0]**: Bi-directional programmable Input/ Output port. It can be configured as pull-up resistor, pull-down resistor or floating input, open-drain PMOS output, or CMOS output. The sink current (I_{OL}) of this I/O can reach 16 mA (VDD = 3.0V, V_{OL} = 0.2*VDD) enough to drive LED. Key scan wakeup source, if key change is detected, chip wakeup from sleep mode.	
REM	11	O	Remote IR signal transmit or receive pin.	
VDD	12	S	power supply	
VSS	13	S	Ground	

4.2. PIN Assignment (Top View)

SOP16 package for GPM6C1015A



5. FUNCTIONAL DESCRIPTIONS

5.1. Central Processing Unit

5.1.1. CPU Introduction

The microprocessor of GPM6C1015A is a high performance processor equipped with 6 internal registers: accumulator, program counter, X register, Y register, stack pointer, and processor status register. This CPU is a fully static CMOS design. The oscillation frequency could be varied up to 8.0MHz depending on the application.

5.1.2. CPU register

The CPU has six registers that are the Program Counter (PC), an Accumulator (A), two index registers (X, Y), the Stack Pointer (SP), and the Status register (P). The program counter consists of 16-bit register.

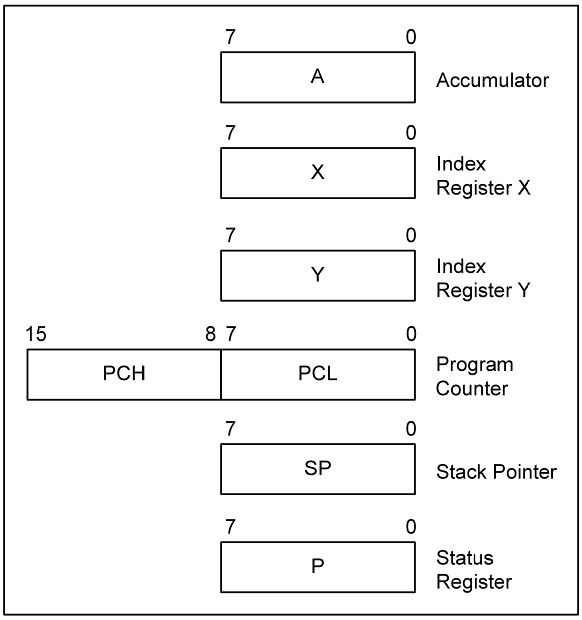


Figure 5-1 System registers

X, Y register

In address mode, X and Y registers can be used as index registers or buffer registers. These register contents are added to the specified address, which becomes the actual address. Some operations such as increment, decrement, comparison and data transfer function can be used in X and Y registers.

Accumulator

The Accumulation is the 8-bit general-purpose register, which can be operated with transfer, temporary saving, condition judgment, etc.

Stack pointer

The CPU has an 8-bit-wide register indicating the location in the stack to be accessed (push or pop) when a subroutine call or interrupt occurs.

When subroutine call is executed or an interrupt occurrence is accepted, the value of stack point is updated automatically.

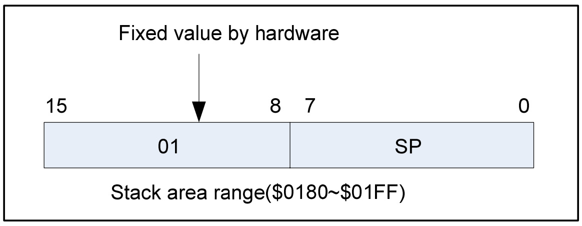


Figure 5-2 Stack point register

[Example] 5-1Initialized stack point value

```
LDX    #C_STACK_BOTTOM     ; Initial stack pointer at $1FF
TXS                        ;Transfer to stack point
```

Program counter (PC)

The program counter is a 16-bit wide register. It consists of two 8-bit registers which registers are PCH and PCL. This register indicates the address of next instruction to be executed. In Reset state, the content of program counter is stored with $FFFC.

Status register (P)

The 8-bit status register contains the interrupt mask and 6 flags representative of the result of the instruction just executed. This register can also be handled by the PHP and PLP instructions. These bits can be individually controlled by specific instructions. The detailed description is shown in following description.

Note: Not all instructions affect status register. A detailed instruction description will be discussed in 6502 instruction manual.

- **Negative flag bit**

This flag indicated the bit7 status of the result of a data or arithmetic operation. Programmer can use this bit to do some operations, e.g. branch condition or bit operation.

- **Overflow flag bit**

This flag indicates whether the overflow has occurred in arithmetic operation. When the result of an addition or subtraction is over +127 or less than –128, this overflow bit is set to ‘1’.

❑ **Decimal mode flag**

This flag indicates what mode is operated by arithmetic operation. The CPU has two operation modes, binary mode and decimal mode for arithmetic operation. Programmer can use the instruction to alternate them.

❑ **Interrupt disable flag**

This bit can enable or disable all interrupt except NMI interrupt source. If this bit is set to '1', CPU will ignore interrupt signal. On the contrary, if this bit is set to '0', CPU will accept interrupt signal.

❑ **Zero flag**

This flag indicated the result of a data or arithmetic operation. If the result is equal to zero, the zero flag is set to '1'. Contrary, this bit is set to '0' by other values.

❑ **Carry flag**

This bit is set to '1 if the result of addition operation generates a carry, or if the result of subtraction doesn't generate a borrowing. In addition, some shift instructions or rotate instructions also change this bit.

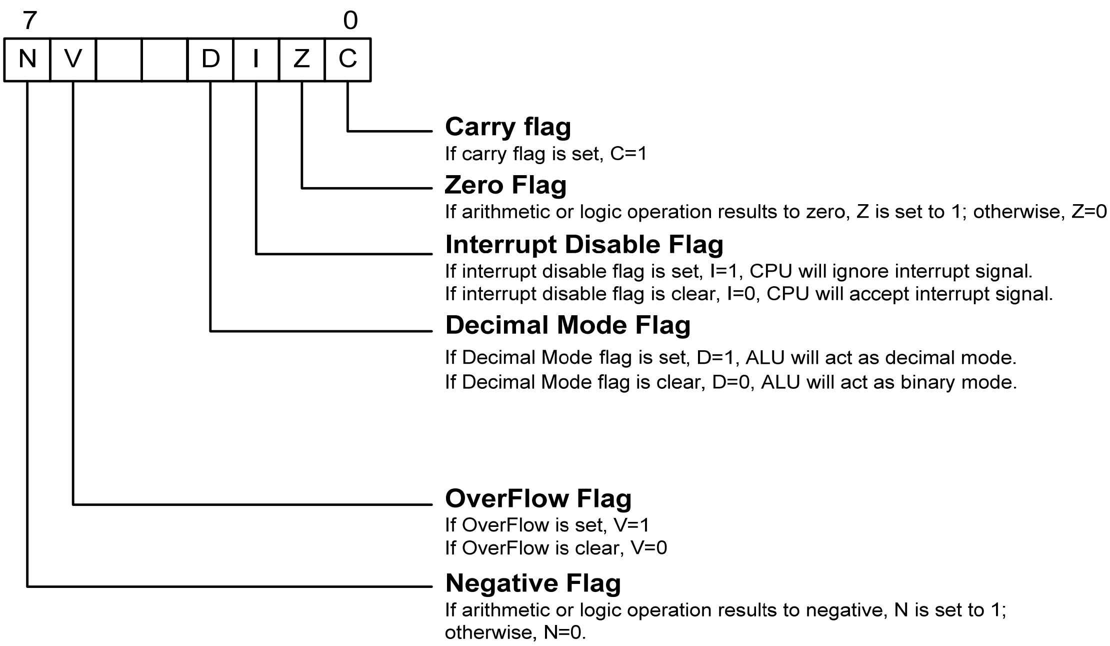


Figure 5-3 Status register

5.2. Memory Organization

5.2.1. Introduction

GPM6C1015A has separated address spaces for program memory and data memory. Program memory can be read only. GPM6C1015A contains up to 16K bytes of program memory. Data memory that contains 256 bytes of RAM including stack area can be read and written.

5.2.2. Memory Space

Memory address allocations on the GPM6C1015A are divided into several parts. The first 128 addresses are allocated for special function registers, including function control registers and I/O control registers, which allow programmer to use the first page instruction in setting this register and help for program size reduction.

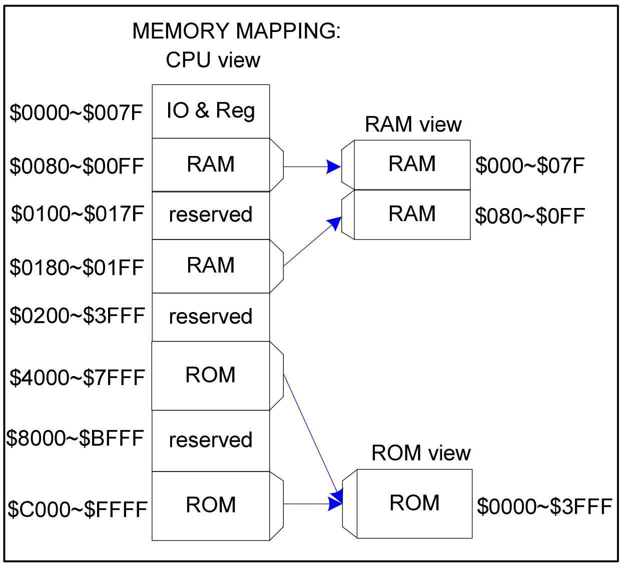


Figure 5-3 shows GPM6C1015A memory map.

GPM6P1015A 's RAM consists of 256 bytes (including Stack). In CPU view, the RAM locations are from $080 through $FF and from $180 to $1FF. They are mapped to $000~$07F and $080~$0FF respectively in RAM view.

GPM6P1015A supports 16K bytes of ROM. In CPU view, the ROM address is located on $4000 ~ $7FFF. And the ROM area, $C000~$FFFF, is always double mapped to the area $0000 to $3FFF.

The address of NMI, $\overline{\text{RESET}}$ and IRQ exception vectors are located from $FFFA to $FFFF. The exception vectors should be specified in the program to have proper operation.

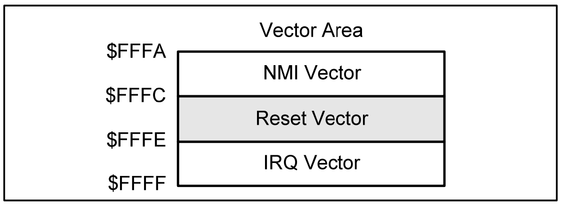


Figure 5-4 Interrupt vector area

[Example] 5-2 Interrupt vector table in software

```
VECTOR:         .SECTION
                DW      V_NMI
                DW      V_Reset
                DW      V_IRQ
```

5.2.3. Configuration Option Register

The configuration option register is used to setup the operation condition. And its CPU view address is $FFF8. It is mapped to the special reserved ROM address $7FF8.

GPM6C1015A has the following configuration options.

- Crystal resonator or internal oscillator clock source option.
- LVR enable or disable option.
- Watch dog enable or disable option.

Device Configuration Register (OPCODE0, $FFF8)

BIT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	**OPTCHK3**	**OPTCHK2**	**OPTCHK1**	**OPTCHK0**	**Reserved**	**WDTENB**	**LVRENB**	**SYSCLKS**
Access	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**
Default	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Bit [7:4] **OPTCHK [3:0]:** Configuration Option Check bits must be filled in 1011.

Bit [3] Reserved

Bit [2] **WDTENB:** disable/enable watchdog

0: WDT is enabled

1: WDT is disabled

Bit [1] **LVRENB:** disable/enable LVR

0: LVR is enabled

1: LVR is disabled

Bit [0] **SYSCLKS:** IOSC (internal) / Crystal selection

0: IOSC

1: Crystal

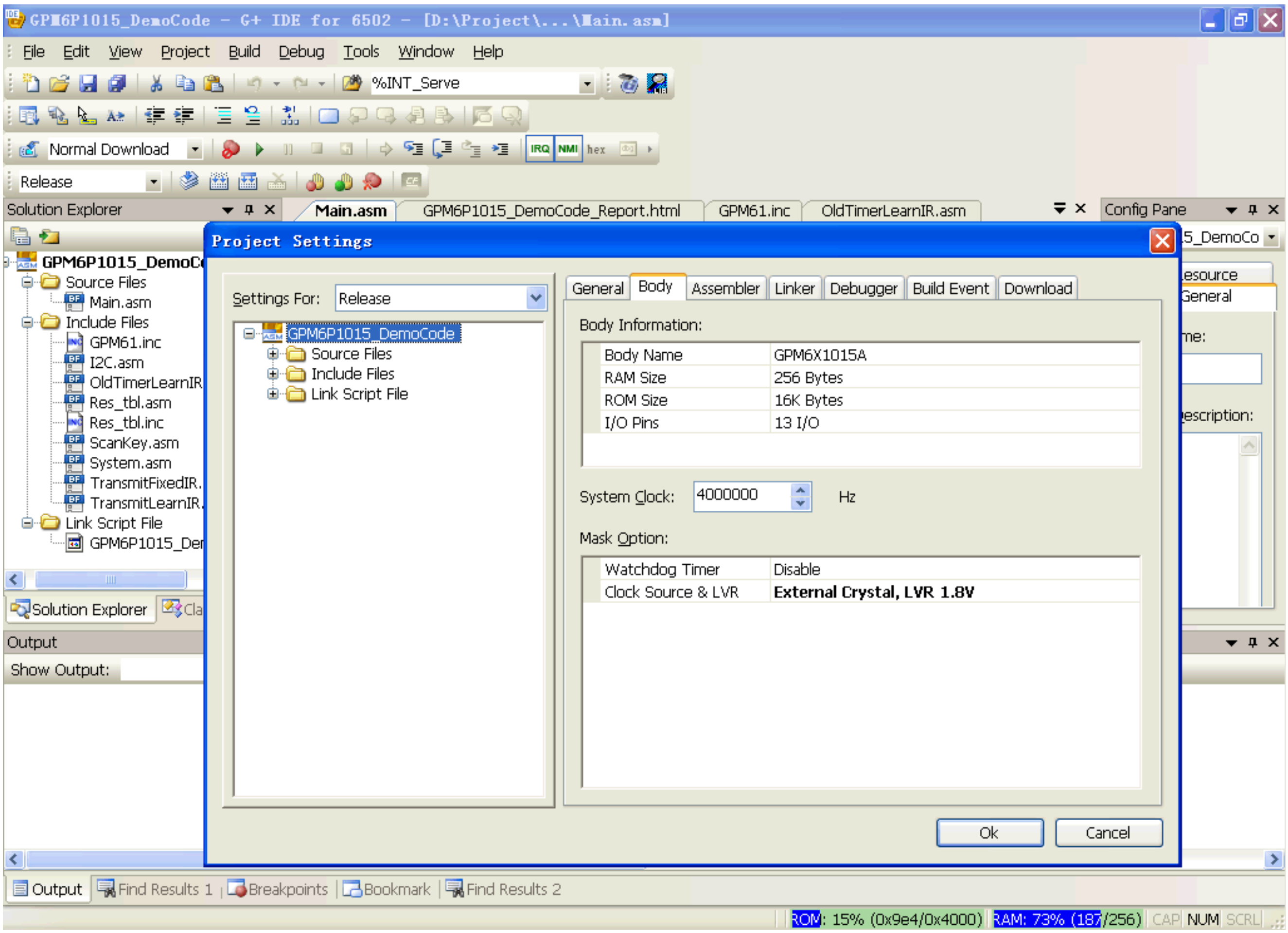


Figure 5-5 G+ IDE Body configuration

5.2.4. Special Function Registers (SFR)

GPM6C1015A device has many control registers. All of them are used by MCU and peripheral function block for controlling the needed operation of the device. Some of the control registers contain control and status bits for peripheral module such as Timer unit, Interrupt control unit, etc. Note that the reserved addresses are not implemented on the chip. Some bits in control register are read only. When writing to them, there are not any effects on the corresponding bits. The following table shows the summary of the control registers. The detailed information of each control registers are explained in each peripheral section.

GPM6C1015A Special Function Register Description

$0000~000B: I/O port

Address	Register	Reset Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$00	P_IOB_DIR	00h	R/W	R/0	R/0	PBDIR5	PBDIR4	PBDIR3	PBDIR2	PBDIR1	PBDIR0
$01	P_IOC_DIR	00h	R/W	R/0	R/0	R/0	R/0	R/0	R/0	R/0	PCDIR0
$02	P_IOD_DIR	00h	R/W	R/0	R/0	PDDIR5	PDDIR4	PDDIR3	PDDIR2	PDDIR1	PDDIR0
$04	P_IOB_ATT	00h	R/W	R/0	R/0	PBATT5	PBATT4	PBATT3	PBATT2	PBATT1	PBATT0
$05	P_IOC_ATT	00h	R/W	R/0	R/0	R/0	R/0	R/0	R/0	R/0	PCATT0
$06	P_IOD_ATT	00h	R/W	R/0	R/0	PDATT5	PDATT4	PDATT3	PDATT2	PDATT1	PDATT0
$08	P_IOB_DAT	00h	R/W	R/0	R/0	PBDAT5	PBDAT4	PBDAT3	PBDAT2	PBDAT1	PBDAT0
$09	P_IOC_DAT	00h	R/W	R/0	R/0	R/0	R/0	R/0	R/0	R/0	PCDAT0
$0A	P_IOD_DAT	00h	R/W	R/0	R/0	PDDAT5	PDDAT4	PDDAT3	PDDAT2	PDDAT1	PDDAT0
PS : R/0 read data value 0											
				$0010~$001D：INT Flag & other special register							
Address	**Register**	**Reset Value**	**R/W**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
$11	P_PWM_DRV	00h	W	R/0	R/0	R/0	R/0	PWMDRV0	*	*	*
	P_RX_SEN	00h	W	R/0	R/0	R/0	R/0	*	TMACAPS	SENSE1	SENSE0
$12	P_SYS_SLEEP	00h	W	SLPCTL7	SLPCTL6	SLPCTL5	SLPCTL4	SLPCTL3	SLPCTL2	SLPCTL1	SLPCTL0
$13	P_INT_CTRL	00h	R/W	TMADTIE	TMAOIE	CAPIE	TMBOIE	F1KIE	F4KIE	F32KIE	F2MIE
$14	P_INT_FLAG	00h	R/W	TMADTIF	TMAOIF	CAPIF	TMBOIF	F1KIF	F4KIF	F32KIF	F2MIF
$16	P_INT_FLAGC	00h	R/W	ENVDET(R)	R/0	R/0	R/0	R/0	R/0	R/0	R/0
$17	P_FUN_S	00h	R/W	TMASET	IRENB	NCDTEN	R/0	R/0	R/0	R/0	R/0
$1B	P_SC_IOB	00h	R/W	R/0	R/0	PB5SE	PB4SE	PB3SE	PB2SE	PB1SE	PB0SE
$1C	P_SC_IOC	00h	R/W	R/0	R/0	R/0	R/0	R/0	R/0	R/0	PC0SE
$1D	P_SC_IOD	00h	R/W	R/0	R/0	PD5SE	PD4SE	PD3SE	PD2SE	PD1SE	PD0SE
				$0020~0026: Timer control							
Address	**Register**	**Reset Value**	**R/W**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
$20	P_WDT_CTRL	00h	W	WDTCTL7	WDTCTL6	WDTCTL5	WDTCTL4	WDTCTL3	WDTCTL2	WDTCTL1	WDTCTL0
$21	P_TMA_CTRL	00h	R/W	TMAES	CAPEG	TMACLK1	TMACLK0	TMADUT1	TMADUT0	TMAMOD1	TMAMOD0
$22	P_TMB_CTRL	00h	R/W	TMBES	R/0	TMBCLK1	TMBCLK0	R/0	R/0	R/0	R/0
$23	P_TMA_CNTL	xxh	R	Mode 0 timer A Counter Low Byte 8-bit Pre-value for COUNTER mode.							
	P_TMA_PWML		W	Mode 0 timer A PWM carrier signal Low Byte 8-bit Period Value for PWM mode.							
	P_TMA_CAPL		R	Mode 0 timer A received carrier signal Low Byte 8-bit Period Width value for CAPTURE mode.							
	P_TMA_ENVL		W	Mode 0 timer A received carrier signal Low Byte 8-bit Period Width pre-value for ENVELOPE mode.							
$24	P_TMA_CNTH	xxh	R	R/0	R/0	R/0	R/0	Mode 0 timer A Counter High Byte 4-bit Pre-value for COUNTER mode.			
	P_TMA_PWMH		W	-	-	-	-	Mode 0 timer A PWM carrier signal High Byte 4-bit Period Value for PWM mode.			
	P_TMA_CAPH		R	R/0	R/0	R/0	R/0	Mode 0 timer A received carrier signal High Byte 4-bit Period Width value for CAPTURE mode.			
	P_TMA_ENVH		W					Mode 0 timer A received carrier signal High Byte 4-bit Period Width pre-value for ENVELOPE mode.			
$25	P_TMB_CNTL	xxh	R	Mode 0 timer B Counter Low Byte 8-bit Pre-value.							
	P_TMB_REGL		W	Mode 0 timer B Low Byte 8-bit Register.							
$26	P_TMB_CNTH	xxh	R	R/0	R/0	R/0	R/0	Mode 0 timer B Counter High Byte 4-bit Pre-value.			
	P_TMB_REGH		W		-	-	-	Mode 0 timer B High Byte 4-bit Register			

5.3. Clock Source

GPM6C1015A supports Crystal or Internal oscillator, as shown in the following diagram, Figure 5-6. They can be selected by device configuration option at address ($FFF8.0) and be set in G+ IDE as Figure 5-5.

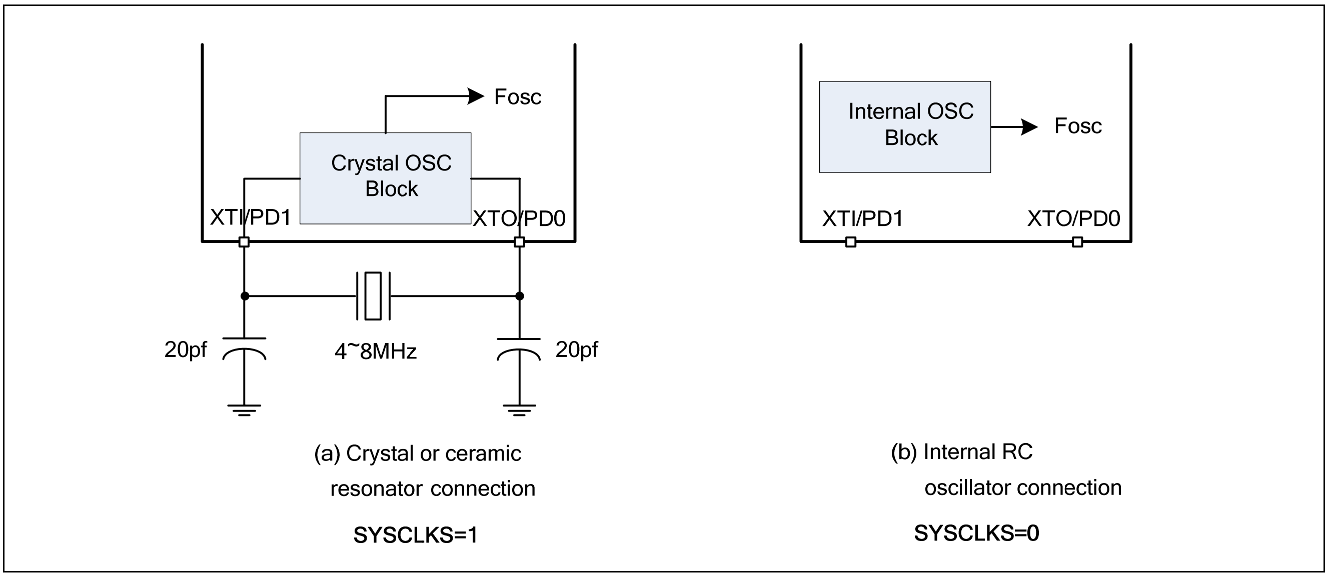


Figure 5-7 clock source

5.4. Power Saving Mode

5.4.1. Introduction

To reduce the current consumption when system does not need to be active, SLEEP mode and FREEZE mode can be utilized. These two modes are able to reduce power consumption and save power. They also feature different wakeup time. User must write corresponding value to SLEEP Control Register to enter SLEEP mode. And system will enter FREEZE mode automatically when power down. For more information about SLEEP and FREEZE modes, please see Figure 5-8. and they will be depicted in the next two sections.

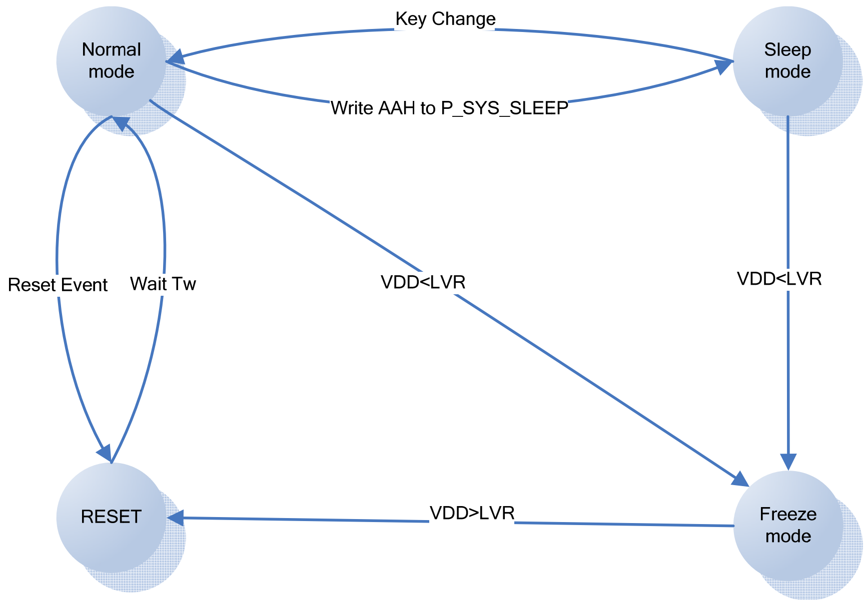


Figure 5-9 Power saving mode operation

5.4.2. SLEEP Mode

The SLEEP mode function will disable all system clocks, including the clock generation circuit. Once the system enters SLEEP mode, LVR function is disabled, RAM and I/Os will remain in their previous states until being awakened. The system will be waked up by any change on port B (M-Type Key). After the GPM6C1015A is waked up, the internal CPU will remain on previous State until Tw ≧65536 x T1 (Tw = waiting time & T1 = system clock cycle); and then continue processing the program (See Figure 5-10).

$$T1 = 1 / (F_{CPU}),\ Tw \geqq 65536 \times T1$$

To enter SLEEP mode, programmer must write #C_SYS_SLEEP ($AA) to SLEEP control register (P_SYS_SLEEP).

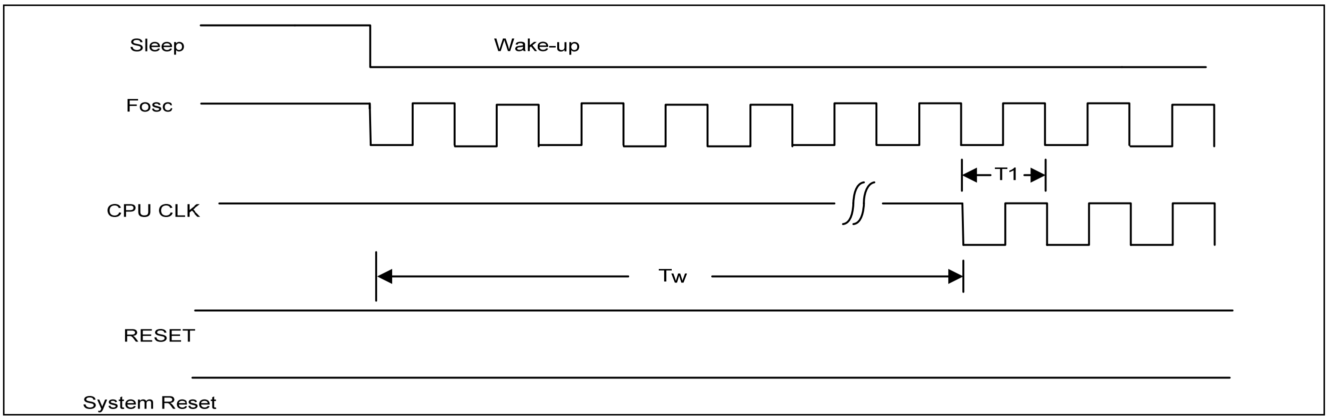


Figure 5-11 SLEEP mode

5.4.3. FREEZE mode

If the power-supply voltage drops below V_{LVR} (See Figure 5-12), Low Voltage Reset (LVR) will reset all functions into the initial operational (stable) state and system will detect whether battery is on or not. Once battery is detected removed, the system enter FREEZE mode, system clock and CPU is stopped; RAM holds its previous data; all I/Os are floating with input function disabled; The FREEZE mode would not be released by any external interrupts unless the battery is reinstalled which voltage is higher than V_{LVR}. The system watchdog action does not occur in FREEZE mode.

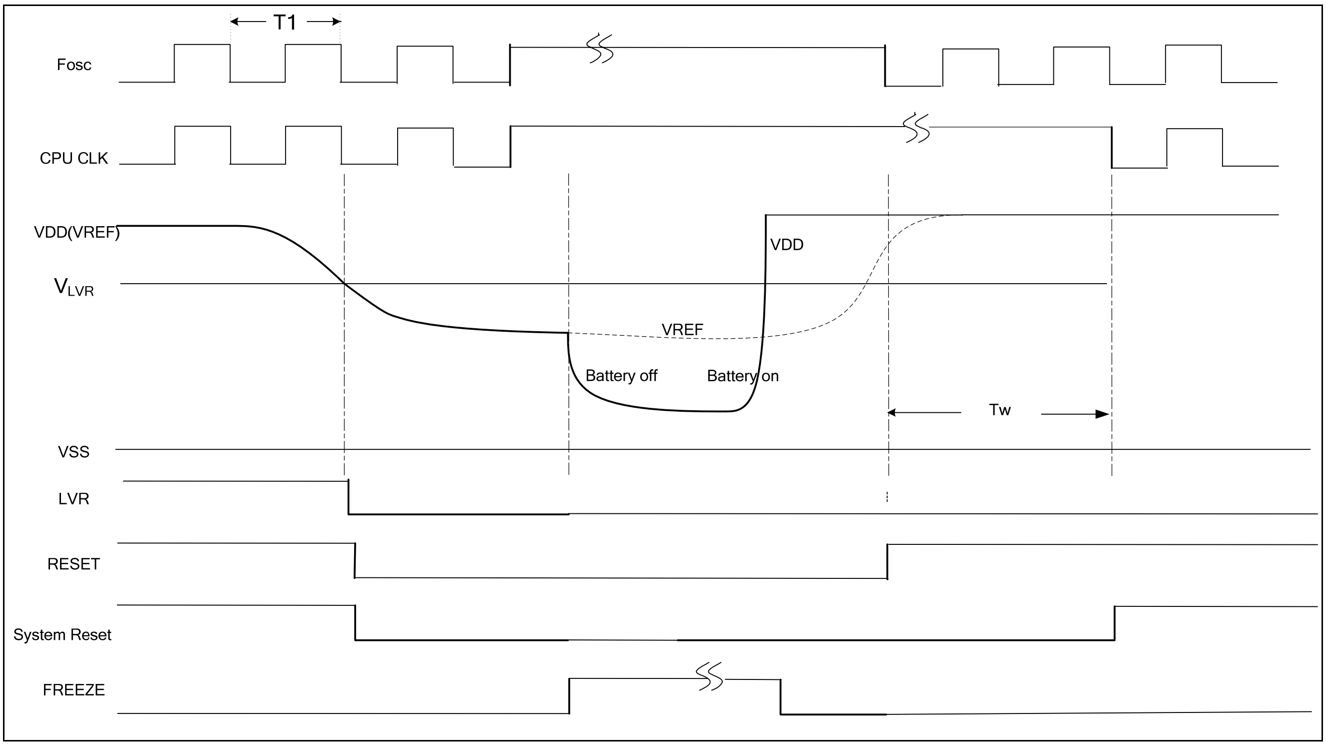


Figure 5-13 FREEZE mode

SLEEP Control Register (P_SYS_SLEEP, $0012)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	SPCTL7	SPCTL6	SPCTL5	SPCTL4	SPCTL3	SPCTL2	SPCTL1	SPCTL0
ACCESS	W	W	W	W	W	W	W	W

Bit [7:0] **SPCTL** [7:0]: Sleep mode control.
$AA : write to enter SLEEP mode (C_SYS_SLEEP)
Other data : reset system

[Example] 5-3 Let MCU enter SLEEP mode

```
    LDA     P_IOB_DAT                    ; latch PortB
    LDA     #C_SYS_SLEEP                 ; SLEEP command $AA
    STA     P_SYS_ SLEEP                 ; go to sleep mode
```

5.5. Interrupt

5.5.1. Introduction

GPM6C1015A provides eight types of interrupt sources with the same normal interrupt level. The eight types of interrupt sources are: TimerA envelope detect interrupt, TimerA capture interrupt, TimerA overflow interrupt, TimerB overflow interrupt, time Fosc/1024 interrupt, time Fosc/4096 interrupt, time Fosc/32768 interrupt, time Fosc/2097152 interrupt,

These interrupts have individual status (occurred or not) and control (enable or not) registers. In general, once an interrupt event occurs, the corresponding flag bit will be set. If the related interrupt control bit is set to enable interrupt, an interrupt request signal will be generated and then CPU executes the interrupt service routine. If the related interrupt control bit is disabled, programmer still can observe the corresponding flag bit, but no interrupt request signal will be generated. The interrupt flag bits must be cleared in the interrupt service routine to prevent program from deadlock. With any instruction, interrupts pending during the previous instruction is served.

Before entering interrupt service routine, the system saves the current PC address into bottom of the stack such as address $1FF and $1FE in Figure 5-14. And abstract the interrupt service routine first address from $FFFE and $FFFF In a corresponding way, the system abstract the return PC address from the bottom of the stack when the interrupt service is completed (See Figure 5-15).

These interrupt sources are listed as [Table] 5-1 and will be described in corresponding section.

[Table] 5-2 Interrupt source list

Source	Interrupt flag register	Interrupt control register	Source	Interrupt flag register	Interrupt control register
Envelope Detect Interrupt	TMADTF($0014.7)	TMADTE($0013.7)	Time Fosc/1024	F1KIF($0014.3)	F1KIE($0013.3)
Timer A Overflow	TMAOIF($0014.6)	TMAOIE($0013.6)	Time Fosc/4096	F4KIF($0014.2)	F4KIE($0013.2)
Capture Interrupt	CAPIF($0014.5)	CAPIE($0013.5)	Time Fosc/32768	F32KIF($0014.1)	F32KIE($0013.1)
Timer B Overflow	TMBOIF($0014.4)	TMBOIE($0013.4)	Time Fosc/2097152	F2MIF($0014.0)	F2MIE($0013.0)

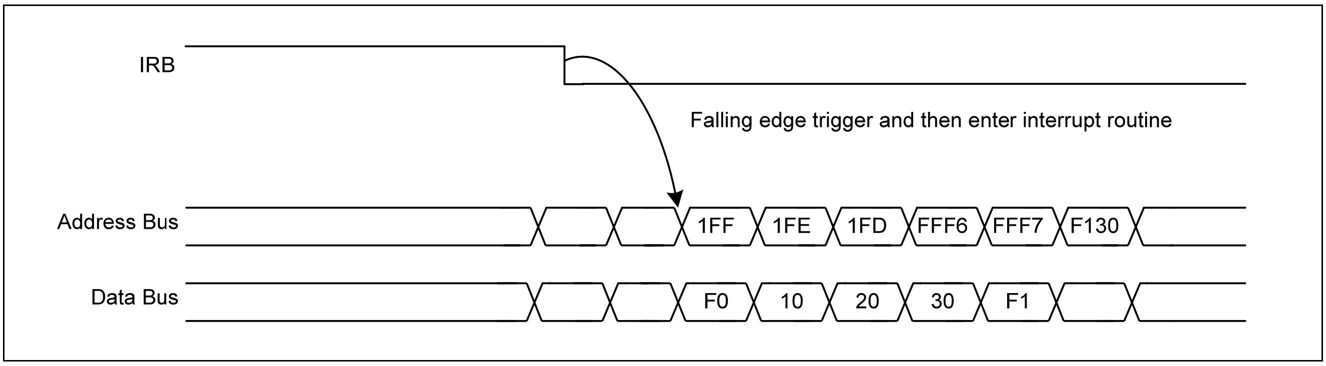


Figure 5-16 Interrupt triggered by IRB

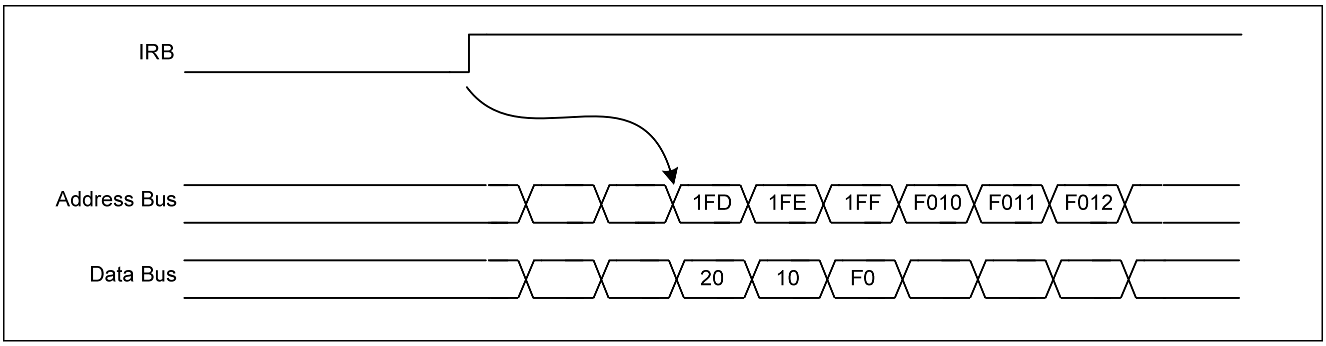


Figure 5-17 Leave interrupt routine

5.5.2. Interrupt register

Interrupt Flag Register (P_INT_FLAG, $0014)

BIT	7	6	5	4	3	2	1	0
Name	TMADTF	TMAOIF	CAPIF	TMBOIF	FD1KIF	FD4KIF	FD32KIF	FD2MIF
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

This flag is cleared by writing the corresponding bit by "1".

Bit7 **TMADTF**: Timer A envelope detect interrupt flag
0 : no event
1 : event occurs

Bit 6 **TMAOIF**: Timer A overflow interrupt flag
0 : no event
1 : event occurs

Bit 5 **CAPIF**: Timer A capture interrupt flag
0 : no event
1 : event occurs

Bit 4 **TMBOIF**: Timer B overflow interrupt flag
0 : no event
1 : event occurs

Bit 3 **FD1KIF**: Time Fosc/1024 interrupt flag
0 : no event
1 : event occurs

Bit 2 **FD4KIF**: Time Fosc/4096 interrupt flag
0 : no event
1 : event occurs

Bit 1 **FD32KIF**: Time Fosc/32768 interrupt flag
0 : no event
1 : event occurs

Bit 0 **FD2MIF**: Time Fosc/2097152 interrupt flag
0 : no event
1 : event occurs

Interrupt Control Register (P_INT_CTRL, $0013)

BIT	7	6	5	4	3	2	1	0
Name	TMADTE	TMAOIE	CAPIE	TMBOIE	FD1KIE	FD4KIE	FD32KIE	FD2MIE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit7 **TMADTE**: Timer A envelope detect interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Bit 6 **TMAOIE**: Timer A overflow interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Bit 5 **CAPIE**: Timer A capture interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Bit 4 **TMBOIE**: Timer B overflow interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Bit 3 **FD1KIE**: Time Fosc/1024 interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Bit 2 **FD4KIE**: Time Fosc/4096 interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Bit 1 **FD32KIE**: Time Fosc/32768 interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Bit 0 **FD2MIE**: Time Fosc/2097152 interrupt enable bit
0 : interrupt disable
1 : interrupt enable

Envelop detect register (P_INT_FLAGC, $0016)

BIT	7	6	5	4	3	2	1	0
Name	ENVDET	-	-	-	-	-	-	-
Access	R	-	-	-	-	-	-	-
Default	0	-	-	-	-	-	-	-

Bit 7 **ENVDET**: Envelope flag showing whether envelope exist or not.
0 : no envelope exist
1 : envelope exist

Bit [6:0] Reserved

[Example] 5-4 Enable Timer A overflow interrupt

```
;=========================;
; main loop routine
;=========================;
LDA     #C_INT_TMAOIE
STA     P_INT_CTRL                    ; enable Timer A overflow INT
CLI                                   ; enable INT
;=========================;
;IRQ interrupt service routine
;=========================;
LDA     #C_INT_TMAOIF
STA     P_INT_FLAG                    ; clear INT request flag
STA     P_INT_CTRL                    ; enable Timer A overflow INT
```

5.6. Reset Sources

There are three types of reset sources for the system: Power-On Reset (POR), Low Voltage Reset (LVR), Watchdog Timer Reset (WDR). These reset sources can be concluded as external events and internal events. The external events come from power line or external trigger event. The internal events come from the program run away. Figure 5-18 shows the affected region for each reset source.

5.6.1. Introduction

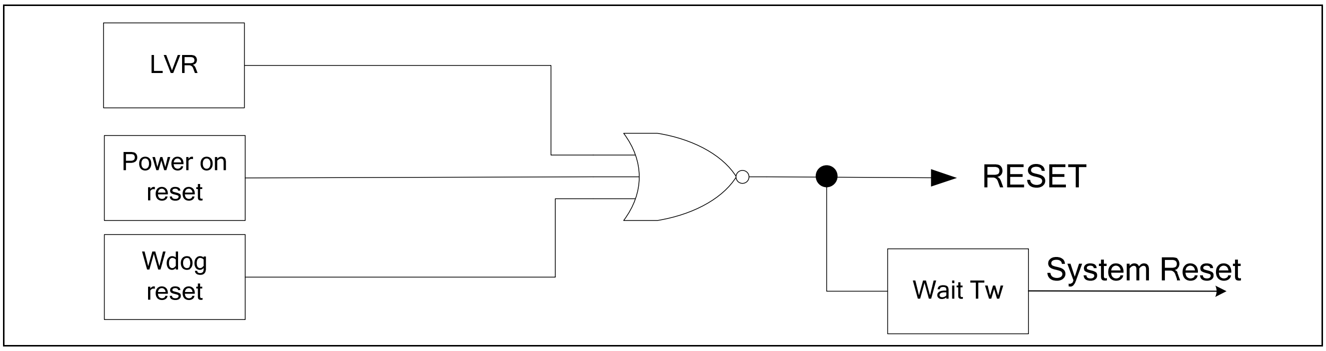


Figure 5-19 Reset sources

5.6.2. Power-On Reset (POR)

A POR is generated when VDD is rising from 0V. When VDD rises to an acceptable level (~1.45V), the power on reset circuit will start a power-on sequence. After that, the system will operate in target speed and start to activate.

5.6.3. Low Voltage Reset (LVR)

The on-chip Low Voltage Reset (LVR) circuitry forces the system returning to the initial status when the MCU voltage falls below the specific LVR trigger voltage. This function prevents MCU from working at an invalid operating voltage range.

A device configuration option bit $FFF8.1 (can be set in G+ IDE) is used to enable or disable this function. If this function is enabled, the LVR circuit will monitor power level while chip is operating. If the power is lower than the specific level for a specific period, the system will be reset to initial status.

5.6.4. Watchdog Timer Reset (WDR)

On-chip watchdog circuitry makes the device entering reset when MCU goes into an unknown state without watchdog clearing information. This function prevents the MCU from being stuck in an abnormal condition. The Watchdog Timer (WDT) can be disabled or enabled through configuration option bit $FFF8.2 (can be set in G+ IDE as Figure 5-20). The Watchdog Timer Reset will be generated by a time-out event of the WDT automatically when watchdog is enabled.

The Watchdog Timer Reset will reset the CPU and restart the program. To avoid a WDT time-out reset, user should write #C_WDT_CLR (=$AA) to P_WDT_CTRL periodically. If a reset signal is generated, it will also clear the WDT counter and restart the WDT.

Different Reset Sequences as the following figures:

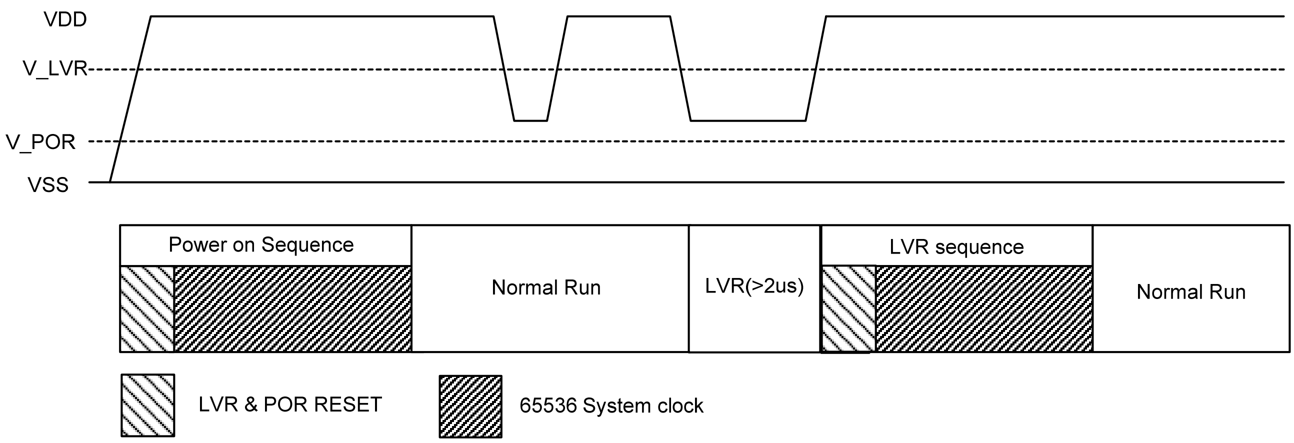


Figure 5-21 POR & LVR Reset Sequence

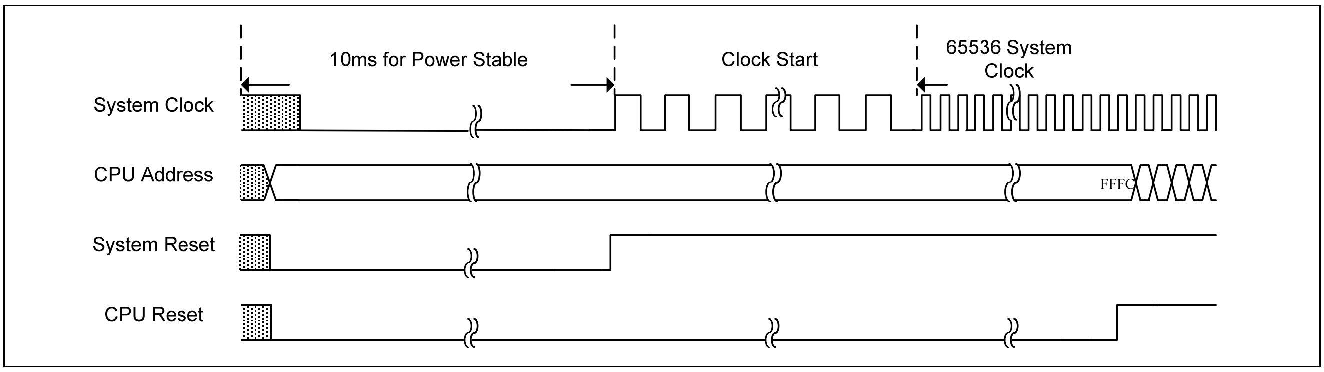


Figure 5-22 Power-On Reset Sequence

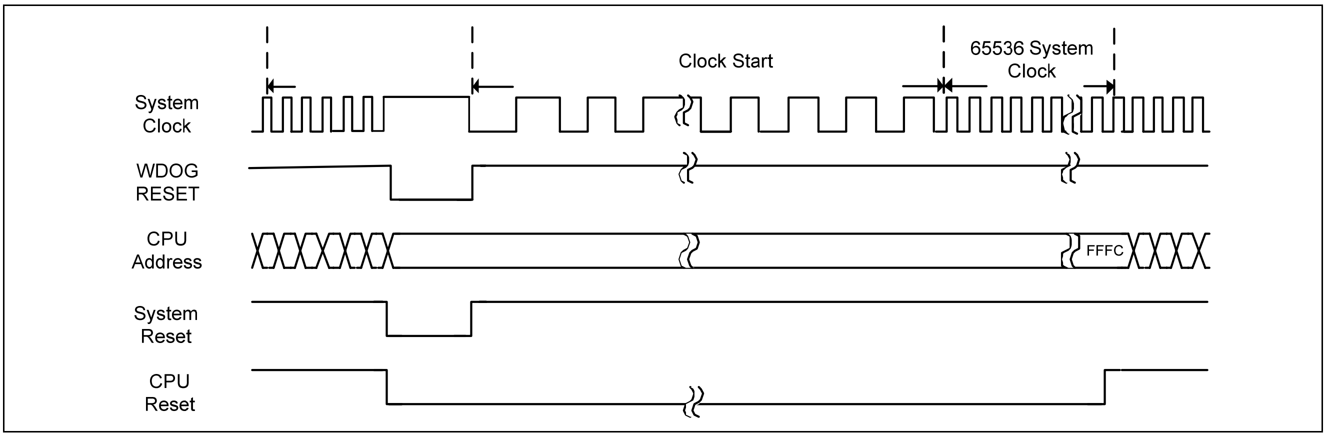


Figure 5-23 Watchdog Reset Sequence

Watchdog Control Register (P_WDT_CTRL, $0020)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	WDTCTRL7	WDTCTRL6	WDTCTRL5	WDTCTRL4	WDTCTRL3	WDTCTRL2	WDTCTRL1	WDTCTRL0
ACCESS	W	W	W	W	W	W	W	W

Bit [7:0] **WDTCTRL [7:0]:** Operation mode control register

$AA : write to clear watchdog CNT (C_WDT_CLR)

Other data : reset system

[Example] 5-5 Clear watchdog counter

```
LDA     # C_WDT_CLR          ; Clear watchdog command $AA
STA     P_WDT_ CTRL
```

5.7. I/O PORTS

5.7.1. Introduction

GPM6C1015A has three ports, Port B, Port C, and Port D.

These port pins may be multiplexed with an alternate function for the peripheral features on the device. In general, when an initial reset state occurs, all ports are used as a general purpose input port. There are three parts, data, direction and attribution registers, in these IO structures. Each corresponding bit in these ports should be given a value.
In M-Type keyboard application, Port B should be configured as input ports, and in sleep mode any change occurred in these ports will cause system wakeup.

The setting rules are as follows:

- The direction setting determines whether this pin is an input or an output.
- The data register is used to read the value on the port, which can be different when programmer sets the port to different configuration (input pull-high/pull- low).

Please refer to the **[Table] 5-4** for PB[5:0] PC[0]and PD[40]'s setting.

[Table] 5-3 I/O Configurations (for PD[5])

Attribution (P_IOX_ATT)	Direction (P_IOX_DIR)	Data (P_IOX_DAT)	Function	Description
0	0	0	Floating	Input with float
0	0	1	Pull low	Input with pull-low
0	1	0	Driving low	Output Data
0	1	1	Floating	float
1	0	0	Floating	Input with float
1	0	1	Pull high	Input with pull-high
1	1	0	Floating	float
1	1	1	Driving low	Output Data

[Table] 5-4 I/O configurations (for PB[5:0], PC[0], PD[4:0])

Attribution (P_IOX_ATT)	Direction (P_IOX_DIR)	Data (P_IOX_DAT)	Function	Description
0	0	0	Floating	Input with float
0	0	1	Pull low	Input with pull-low
0	1	0	Driving low	Output Data
0	1	1	Driving High	Output Data
1	0	0	Floating	Input with float
1	0	1	Pull high	Input with pull-high
1	1	0	Driving High	Output Data
1	1	1	Driving low	Output Data

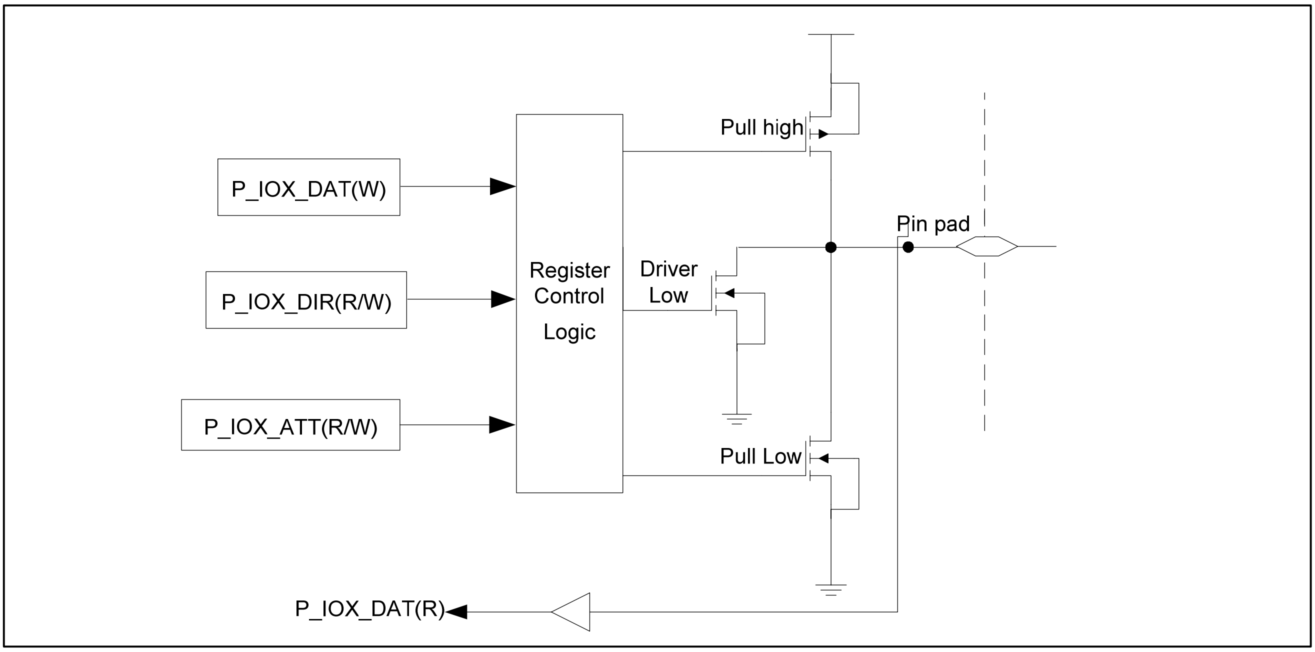


Figure 5-24 Block diagram of I/O port (PD[5])

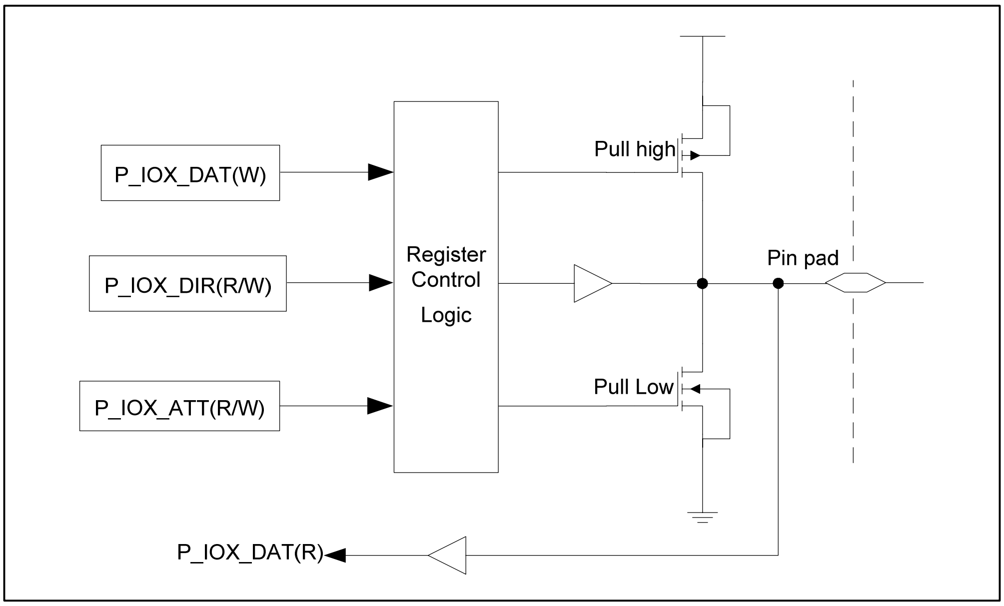


Figure 5-25 Block diagram of I/O port (PB[5:0], PC[0], PD[4:0])

5.7.2. Port B

GPM6C1015A Port B is a 6-bit programmable bi-directional port (PB[5 :0]). The Port is controlled by direction control register P_IOB_DIR, and attribution register P_IOB_ATT. Reading P_IOB_DAT will get the real IO value.

Port B Direction Register (P_IOB_DIR, $0000)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PBDIR5	PBDIR4	PBDIR3	PBDIR2	PBDIR1	PBDIR0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	-	0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PBDIR[5:0]:** Port B direction register.

0 : input

1 : output

Port B Attribution Register (P_IOB_ATT, $0004)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PBATT5	PBATT4	PBATT3	PBATT2	PBATT1	PBATT0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	-	0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PBATT:** Port B attribution register

Port B Data Register (P_IOB_ DAT, $0008)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PBDAT5	PBDAT4	PBDAT3	PBDAT2	PBDAT1	PBDAT0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	-	0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PBDAT[5:0]:** Port B Data value.

Read to get Port B value

Write to configure output high/low or configure input with pull high/low resistor.

[Example] 5-6 Set Port B [3:0] as output with low data and Port B [5:4] as input with pulling high.

```
LDA     #$0F            ; store accumulator with $0F
STA     P_IOB_DIR       ; set direction
LDA     #$30            ; store accumulator with $30
STA     P_IOB_ATT       ; set attribute
LDA     #$30            ; store accumulator with $30
STA     P_IOB_DAT       ; set Port Data
```

[Example] 5-7 Set Port B [5:0] as input with float.

```
LDA     #$00            ; store accumulator with $00
STA     P_IOB_ATT       ; set direction
STA     P_IOB_DIR       ; set attribute
STA     P_IOB_DAT       ; set Port Data
```

The Port B can be configured as scan key or not by key scan selection register.

Port B Key Scan Selection register (P_SC_IOB, $001B)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PBSE5	PBSE4	PBSE3	PBSE2	PBSE1	PBSE0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT			0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PBSE[5:0]:** Port B Key scan selection register.

0: no key scan function

1: with key scan function.

5.7.3. Port C

GPM6C1015A Port C is a 1-bit programmable bi-directional port (PC[0]). The Port is controlled by direction control register P_IOC_DIR, and attribution register P_IOC_ATT. Reading P_IOC_DAT will get the real IO value.

Port C Direction Register (P_IOC_DIR, $0002)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	-	-	-	PCDIR0
ACCESS	-	-	-	-	-	-	-	R/W
DEFAULT	-	-	-	-	-	-	-	0

Bit [7:1] **Reserve**

Bit [0] **PCDIR0:** Port C direction register.

0 : input

1 : output

Port C Attribution Register (P_IOC_ATT, $0005)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	-	-	-	PCATT0
ACCESS	-	-	-	-	-	-	-	R/W
DEFAULT	-	-	-	-	-	-	-	0

Bit [7:1] **Reserve**

Bit [0] **PCATT0:** Port C attribution register

Port C Data Register (P_IOC_ DAT, $0009)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	-	-	-	PCDAT0
ACCESS	-	-	-	-	-	-	-	R/W
DEFAULT	-	-	-	-	-	-	-	0

Bit [7:1] **Reserve**

Bit [0] **PCDAT0:** Port C Data value.

Read to get Port C value

Write to configure output high/low or configure input with pull high/low resistor.

The Port C can be configured as scan key or not by key scan selection register.

Port C Key Scan Selection register (P_SC_IOC, $001C)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	-	-	-	PCSE0
ACCESS	-	-	-	-	-	-	-	R/W
DEFAULT	-	-	-	-	-	-	-	0

Bit [7:6] **Reserve**

Bit [5:0] **PCSE0:** Port C Key scan selection register.

0: no key scan function

1: with key scan function.

5.7.4. Port D

GPM6C1015A Port D is a 6-bit programmable bi-directional port. The Port is controlled by direction control register P_IOD_DIR, and attribution register P_IOD_ATT. Reading P_IOD_DAT will get the real IO value. In addition, Port D is multiplexed with various special functions. After reset, the default setting for port D is used as general I/O ports.

[Table] 5-5 Port D function list

Port D Pin	BIT	Shared function
PD0	Bit0	Crystal output (XTO)
PD1	Bit1	Crystal input (XTI)

Port D Direction Register (P_IOD_DIR, $0002)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PDDIR5	PDDIR4	PDDIR3	PDDIR2	PDDIR1	PDDIR0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	-	0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PDDIR[5:0]:** Port D direction register.

0 : input

1 : output

Port D Attribution Register (P_IOD_ATT, $0006)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PDATT5	PDATT4	PDATT3	PDATT2	PDATT1	PDATT0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	-	0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PDATT[5:0]:** Port D attribution register

Port D Data Register (P_IOD_ DAT, $000A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PDDAT5	PDDAT4	PDDAT3	PDDAT2	PDDAT1	PDDAT0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	-	-	0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PDDAT[5:0]:** Port D Data value.

Read to get Port D value

Write to configure output high/low or configure input with pull high/low resistor.

[Example] 5-8 Set Port D[1:0] as output with low data.

```
LDA   #$03        ; store accumulator with $03
STA   P_IOD_DIR   ; set direction
LDA   #$00        ; store accumulator with $00
STA   P_IOD_ATT   ; set attribute
STA   P_IOD_DAT   ; set port data
```

The Port D can be configured as scan key or not by key scan selection register.

Port D Key Scan Selection register (P_SC_IOD, $001D)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	PDSE5	PDSE4	PDSE3	PDSE2	PDSE1	PDSE0
ACCESS	-	-	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT			0	0	0	0	0	0

Bit [7:6] **Reserve**

Bit [5:0] **PDSE[5:0]:** Port D Key scan selection register.
0: no key scan function
1: with key scan function.

5.8. Timer Module

5.8.1. Introduction

GPM6C1015A has two timers, Timer A and Timer B respectively. Timer A contains one powerful PWM function and is controlled by corresponding control registers. This function can be easily configured. GPM6C1015A Timer A also has a Capture function; it can capture the frequency of input signal. And GPM6C1015A Timer A has another function is envelope detection; it can detect envelope waveform of input signal with or without carrier signal. Each timer's function summary is shown as [Table] 5-6.

[Table] 5-6 Summary of timer function for GPM6C1015A

	Timer Counter	PWM	CAPTURE	ENVELOPE DETECT
Timer A	YES	YES	YES	YES
Timer B	YES	None	None	None

5.8.2. Mode 0 Timer A (12-bit up count timer)

When Timer A is selected as 12-bit up count timer via configuring the corresponding bits of the control register (P_TIM_SEL[7]), the Timer A is special for generating carrier signal in IR control application. The timer A's input clock is selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16), which can be configured by control register P_TMA_CTRL[5:4]. Timer A provides with two PWM modes, and the PWM signal is send to IR TX (REM) pin. The driver current of these two kinds of PWM are programmable by configuring TX PWM driving current control source register (P_PWM_DRV [3]).

GPM6C1015A 12-bit up count Timer A module has all the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$FFF to #$000
- Support PWM with carrier signal mode
- Support PWM without carrier signal mode
- Support capture mode for learning function
- Support envelope detect mode for learning function

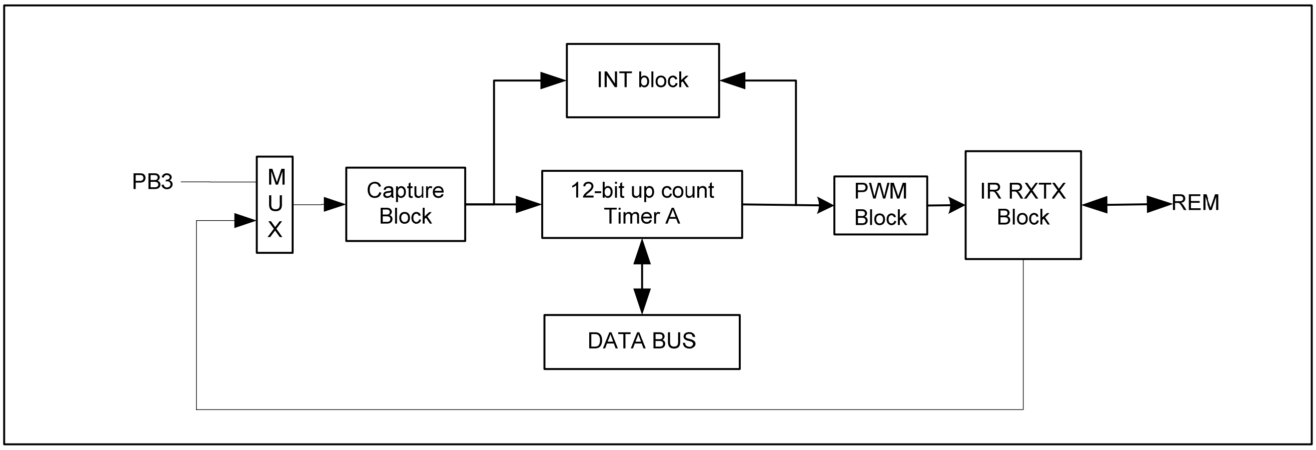


Figure 5-26 GPM6C1015A 12-bit up count Timer A block diagram

5.8.2.1. Mode 0 Timer A PWM with carrier signal mode

GPM6C1015A Timer A can be configured as PWM mode for generating carrier signal. In PWM with carrier signal mode, the 12-bit timer is an up counter with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When Timer A is started, the value of 4-bit high-byte (low-nibble) register and 8-bit low-byte register would firstly be loaded into the 12-bit counter and then the counter starts count up from the loaded value. If an overflow occurs, the value of high-byte (low-nibble) register (P_TMA_CNTH) and low-byte register (P_TMA_CNTL) would be reloaded into the counter automatically and the counter starts count up again. So the carrier signal with frequency programmable can be generated by this PWM mode via configuring these two registers. Also users can select PWM duty cycle (1/3, 1/4, 1/5, 1/2) via configuring the corresponding bits of the control register (P_TMA_CTRL[3:2]). The carrier signal's enabled or disabled bit can be controlled by two methods depended on which clock source is selected by timer B. If timer B is selected one of the first three clock sources (Fosc, Fosc/4 or Fosc/64) by P_TMB_CTRL [5:4] (TMBCLK [1:0]), the timer A's carrier signal on/ off is controlled by timer A's enable/ disable control bit (TMAES) directly.

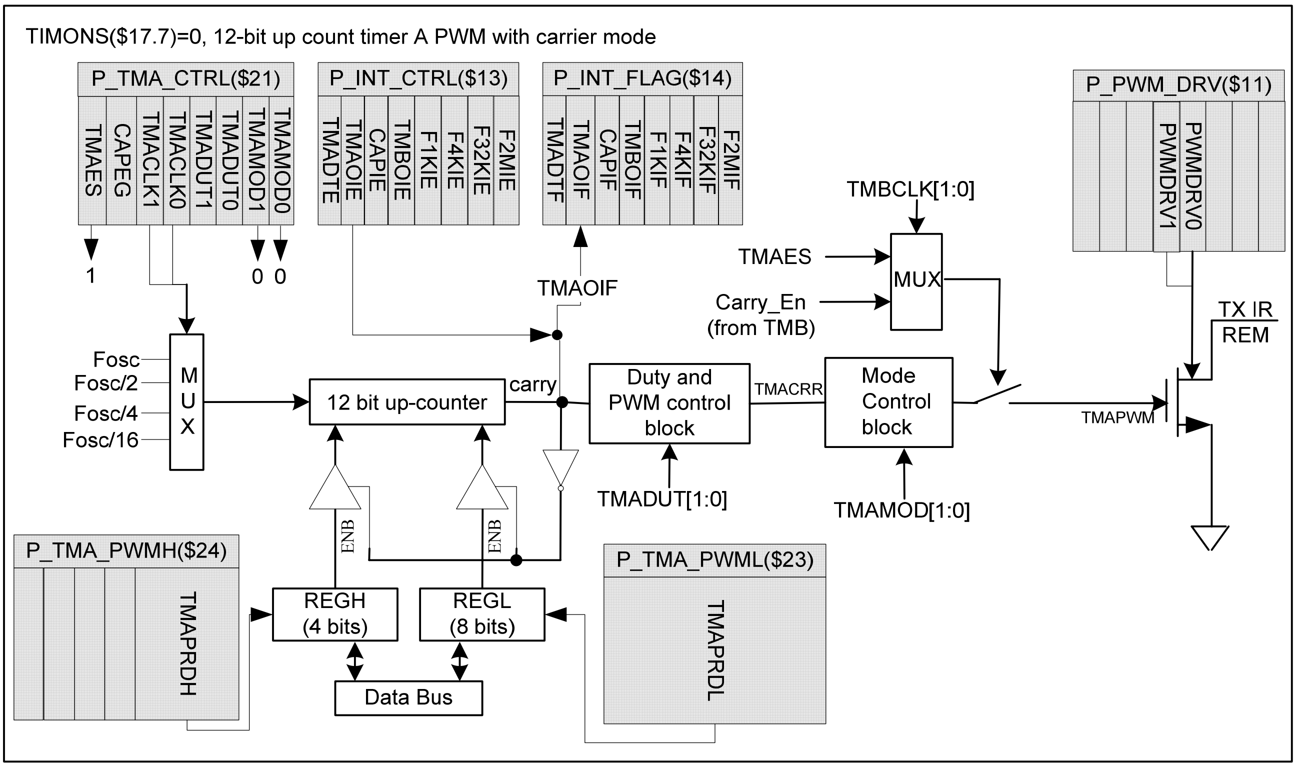


Figure 5-19 Mode 0 Timer A PWM mode diagram

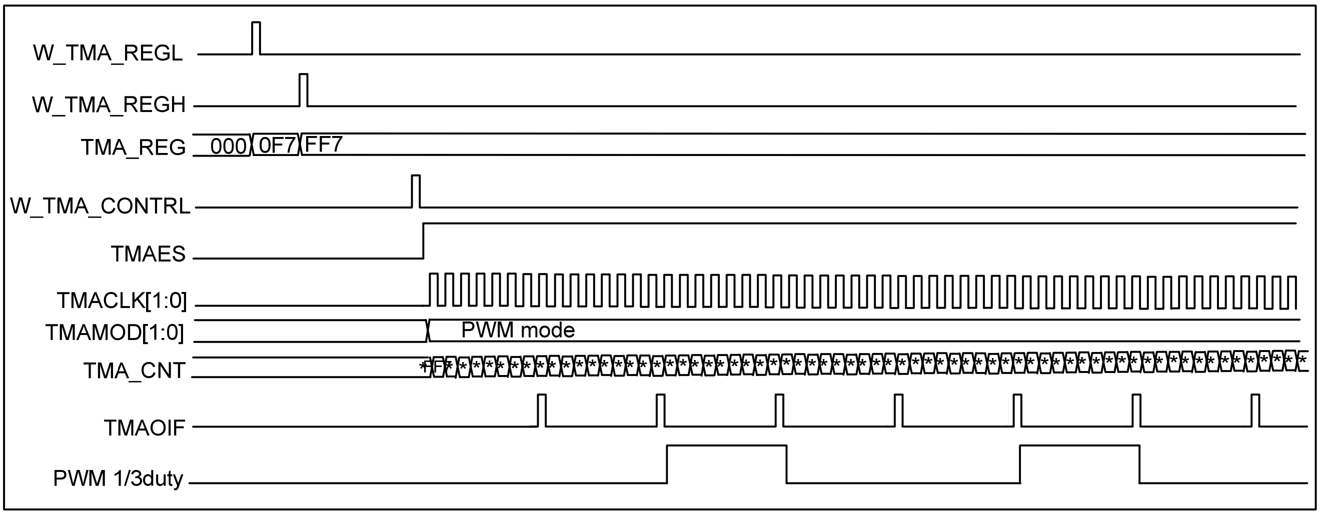


Figure 5-20 Mode 0 Timer A Normal PWM generation without envelop

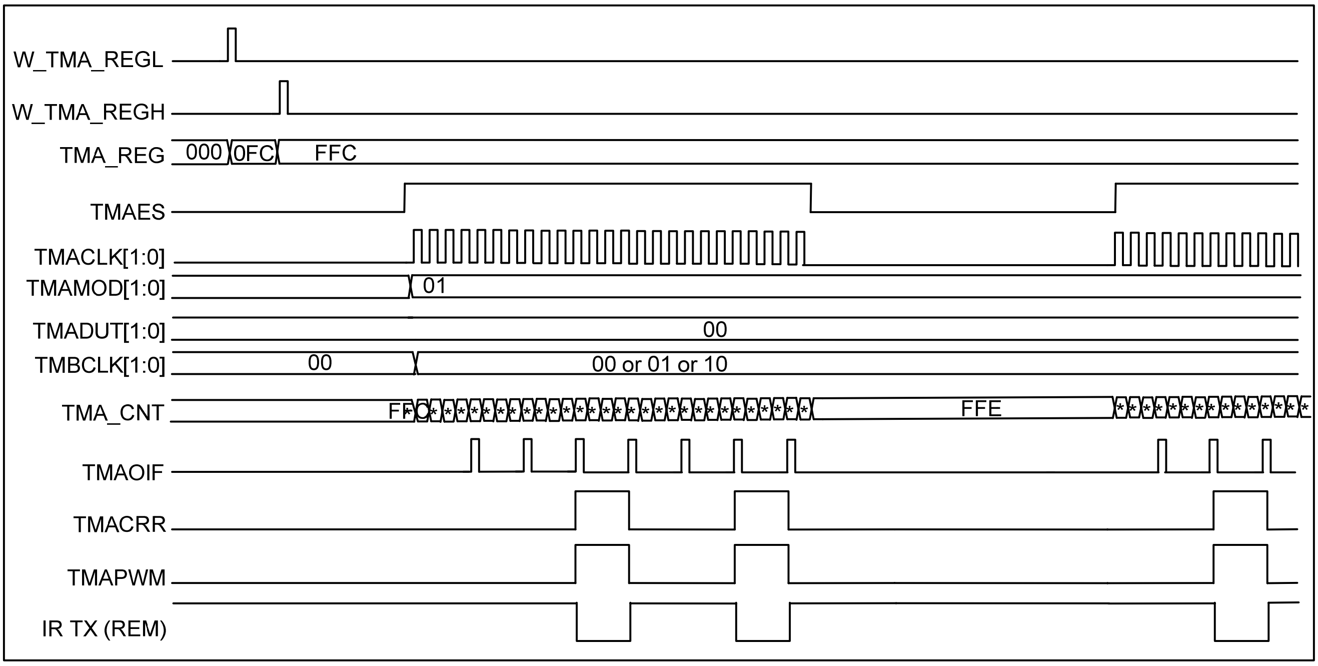
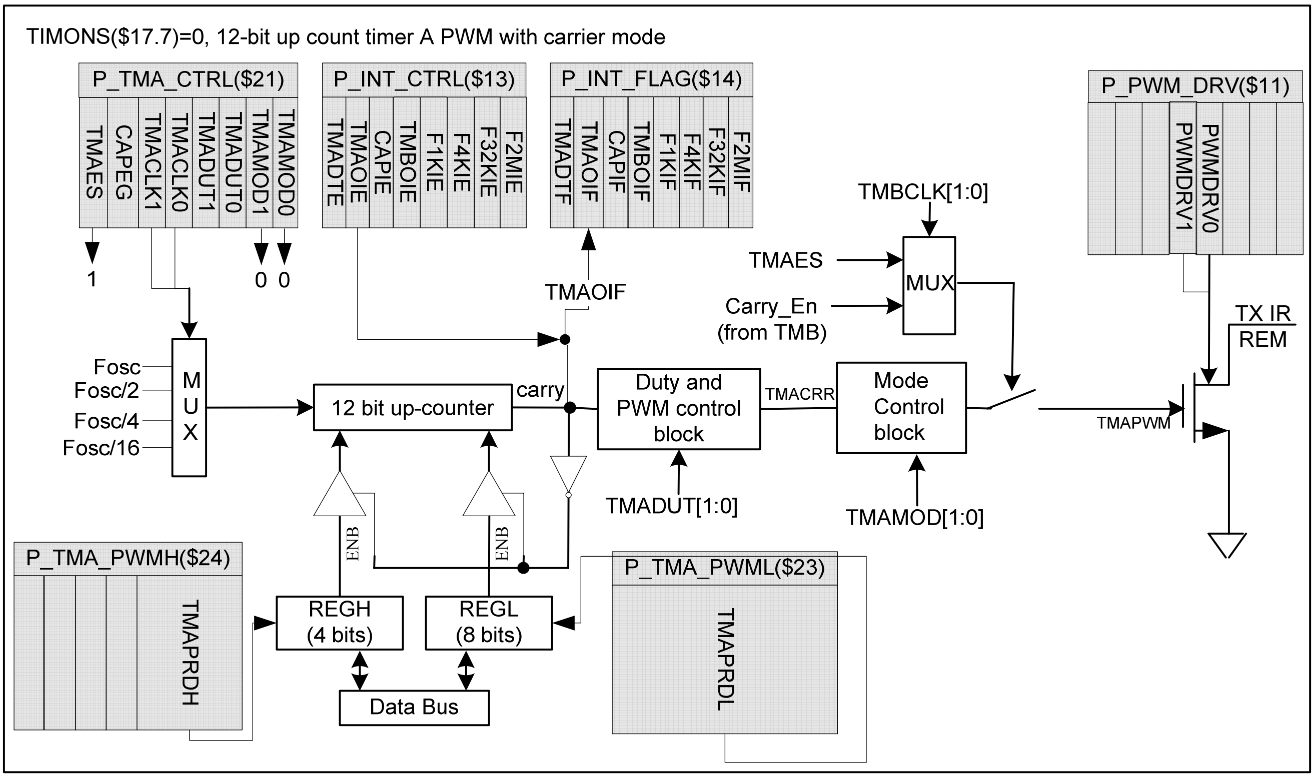


Figure 5-21 The Waveform of Mode 0 Timer A PWM with carrier signal mode (1/3 duty, on/off control by TMAES)

Another method to generate envelope PWM signal is that Timer A and Timer B must be used together. Timer A must generate carry clock at first, which is same as normal PWM generation. Then enable Timer B and select Timer A carrier signal as its input clock. And Timer B register must be written in the right data, which represents the carry number. When TMBOVF happens, another value must be written into Timer B register, which represents the no carry clock number. Envelop with carrier is on or off only when Timer B overflow events occur one by one. Then, the envelop PWM signal will be generated at REM port at last.



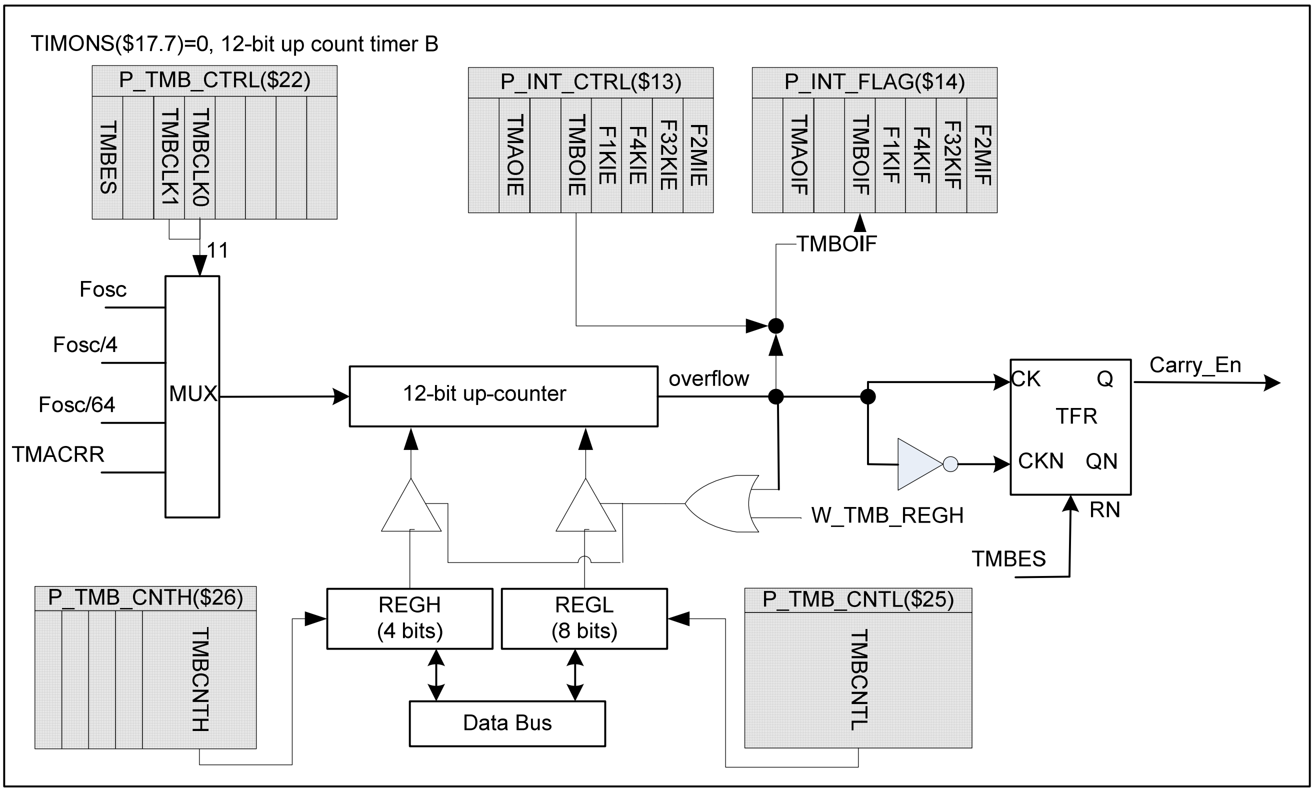


Figure 5-22 Envelope PWM Generated by Mode 0 Timer A & Mode 0 Timer B diagram

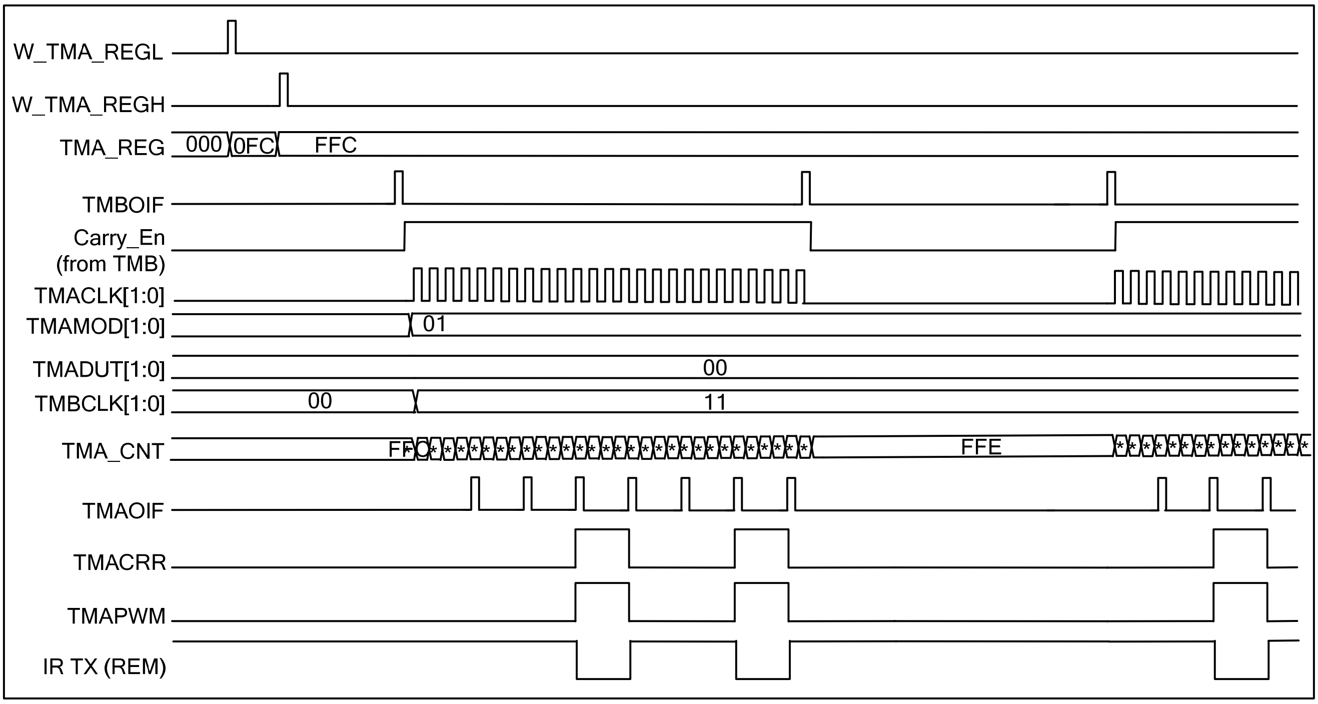


Figure 5-23 The Waveform of Mode 0 Timer A PWM with carrier signal mode (1/3 duty, on/off control by Mode 0 Timer B overflow events)

5.8.2.2. Mode 0 Timer A PWM without carrier signal mode

GPM6C1015A PWM without carrier signal mode is used to generate envelop PWM signal without carrier signal. In this mode, IR TX (REM) pin just output high or low, and is controlled by Timer A's enable or disable control bit or TimerB's overflow events in turn. The same as PWM with carrier signal mode, the 12-bit timer is an up counter with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When the Timer A is started, the value of high-byte (low-nibble) Register and low-byte Register will firstly be loaded into the 12-bit counter and then the counter starts to count up from the loaded value. If an overflow occurs, the value of high-byte (low-nibble) register and low-byte register will be reloaded into the counter automatically and the counter starts to up-count again. The internal carrier signal is generated but is not sent to IR TX pin.

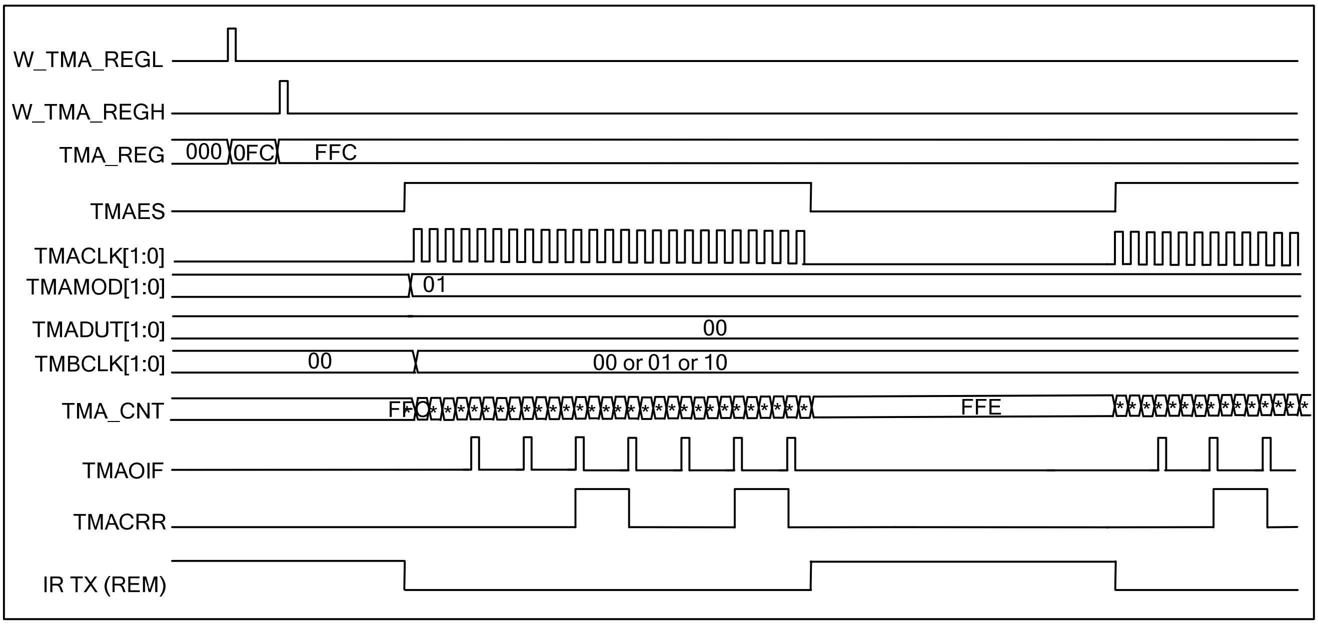


Figure 5-24 The Waveform of Mode 0 Timer A PWM without carrier signal mode (on/off control by TMAES)

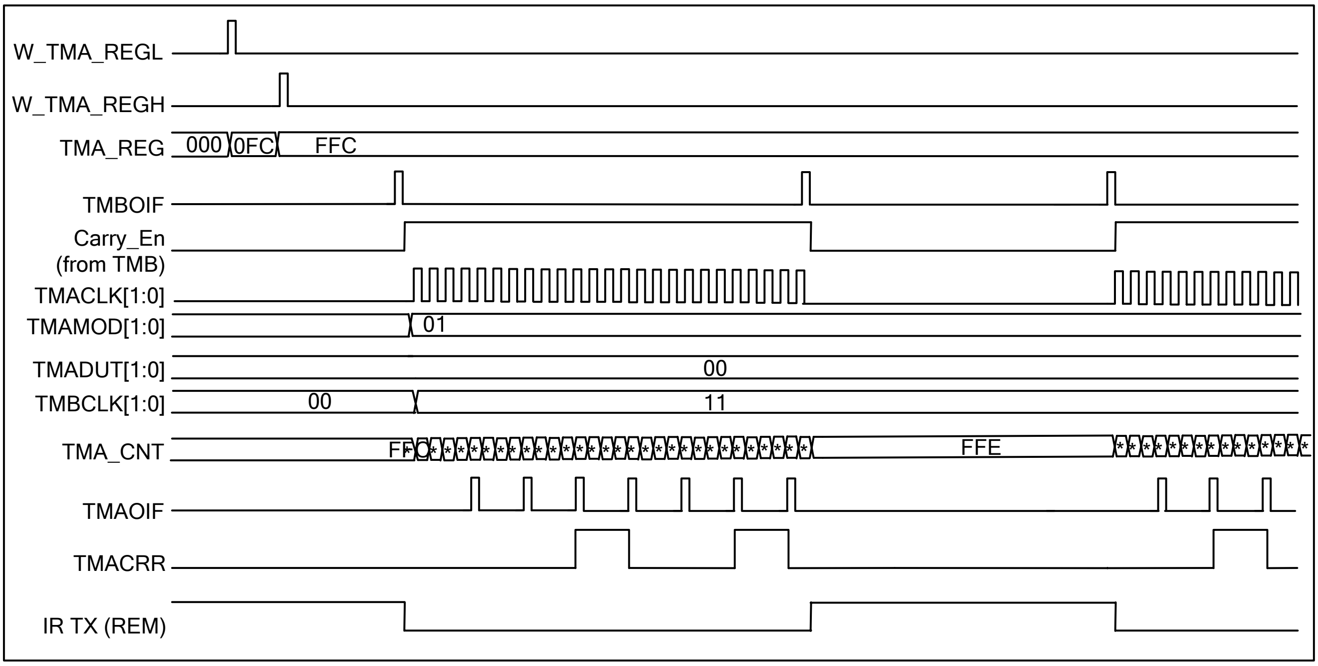


Figure 5-25 The Waveform of Mode 0 Timer A PWM without carrier signal mode (on/off control by Mode 0 Timer B overflow events)

5.8.2.3. Mode 0 Timer A Capture & Envelope detect mode

GPM6C1015A can be used in IR learning function application, Timer A should be configured as capture mode for measuring the frequency of input signal from REM pin. In capture mode, the 12-bit timer is an up counter which counts from 00H with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When rising or falling (selectable via P_TMA_CTRL) edge of RX is captured, the high-byte (low-nibble) value of the counter would be loaded into Register high and the low byte value of counter would be loaded into Register low, at the same time, it generates an interrupt (CAPIF) and then the counter is cleared to 00H. When the timer overflows, the overflow interrupt (TMAOIF) occurs. The input carrier signal cycle time is recorded in Register low (P_TMA_CAPL) and Register high (P_TMA_CAPH). Of course, if the time data that would to be record is bigger than the biggest data that these two registers can be loaded, the overflows of the timer A should be count included.

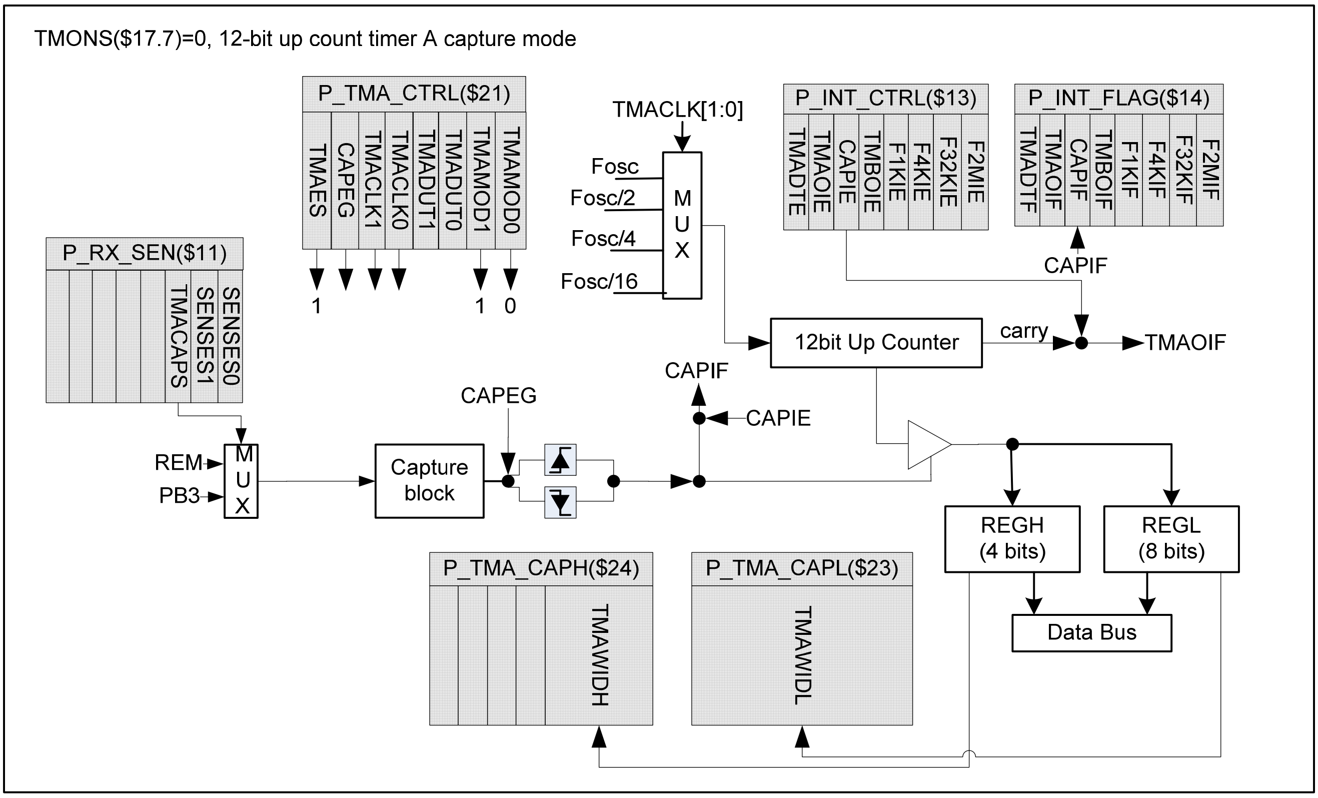


Figure 5-26 GPM6C1015A Mode 0 Timer A block diagram (Capture Mode)

After capture the carrier frequency, Timer A should be configured as envelope detect mode for measuring the envelope of input signal from REM pin.

In order to detect the envelope, enter capture mode at first, and get the carrier frequency (named F_{CRR}). Then load the value (0xFFF-1.5* F_{CRR}) to Timer A counter registers (P_TMA_ENVH & P_TMA_ENVL, $24 & $23) and enter envelope detect mode. If the first rising or falling-edge of carry wave arrives, envelope interrupt occur (TMADTIF:1) and ENVDET ($16.7) is set to '1', and the value (0xFFF-1.5* F_{CRR}) is loaded to Counter automatically, and Counter starts to count. If next rising or falling-edge arrives, the value (0xFFF-1.5* F_{CRR}) will be reloaded into the counter, and ENVDET($16.7) does not change its status (still equal '1'). However, if the next carry wave does not arrive on time (that's over 1.5* F_{CRR}), the Timer A overflow happens resulting in envelope interrupt occurring, and makes ENVDET($16.7) changed to "0". So check ENVDET bit can know whether envelope exist or not.

And if the data received is a signal without carrier signal (determined by software method used), the register NCDTEN ($17.5) should be set as 1. The signal (without carrier signal) received delivered to ENVDET($16.7) directly. Also user can check ENVDET bit to get the input signal with carrier signal.

.

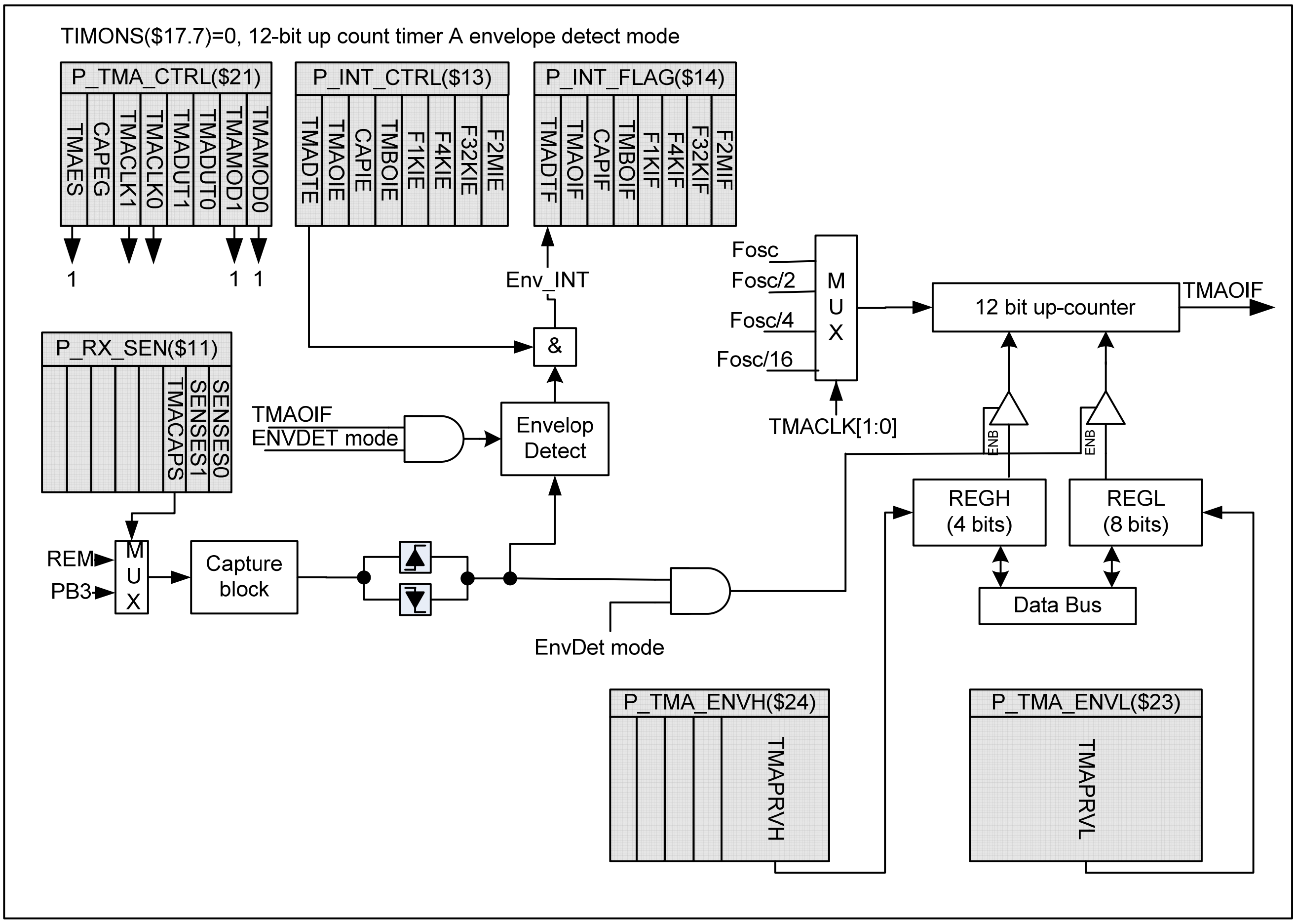


Figure 5-27 Mode 0 Timer A block diagram (Envelope detect Mode)

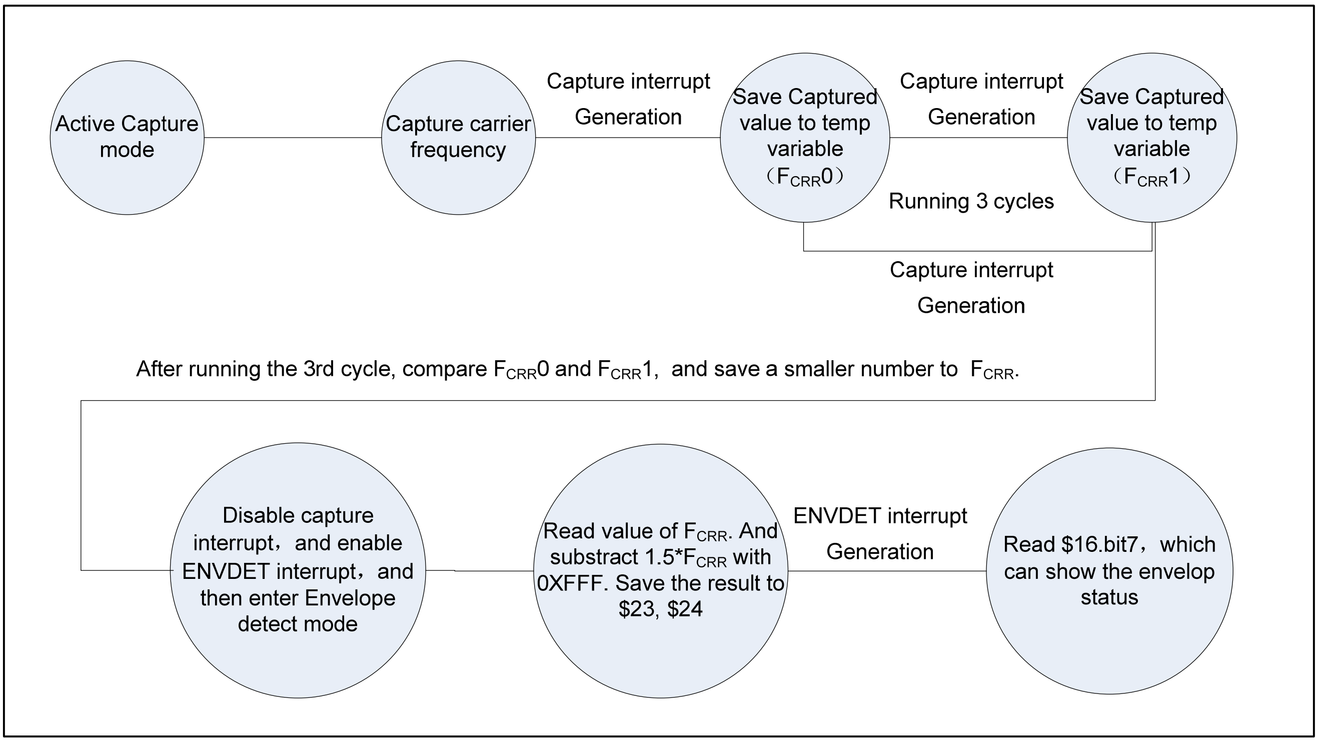


Figure 5-28 Mode 0 Timer A envelope detect flow

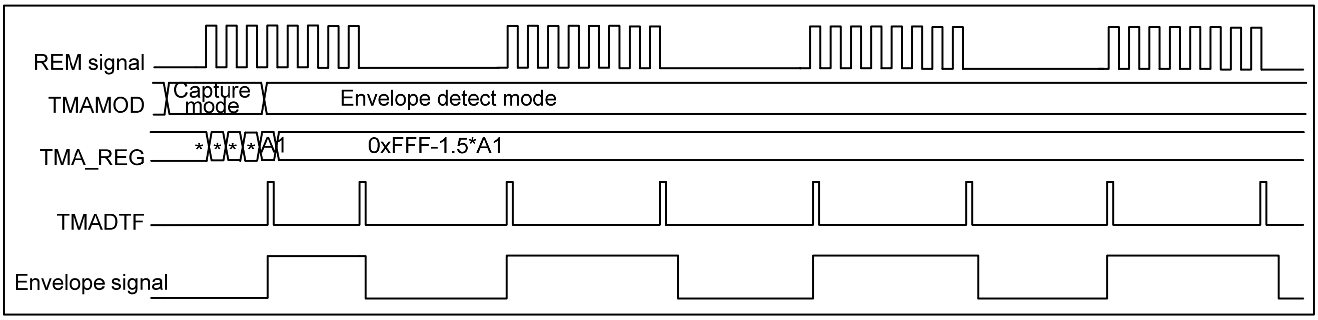


Figure 5-29 Waveform of GPM6C1015A mode 0 Timer A envelope detect

Timer Special Configure Register (P_FUN_S, $0017)

BIT	7	6	5	4	3	2	1	0
Name	TMASET	IRENB	NCDTEN	-	-	-	-	-
Access	R/W	R/W	R/W	-	-	-	-	-
Default	0	0	0	-	-	-	-	-

Bit 7 **TMONS** Timer A enable/disable control.
0, disable; (C_TMAES_DIS)
1, enable. (C_TMAES_EN)

Bit 6 **IRENB:** PWM output function enable/disable. Active when TIMONS=1
0 : PWM output function enable; (C_PWM_EN)
1 : PWM output function disable. (C_PWM_DIS)

Bit 5 **NCDTEN:** With carrier or without carrier signal envelope detect select. Active when TIMONS=1.
0 : With carrier signal; (C_ENVDT_CA)
1: Without carrier signal. (C_ ENVDT_NCA)

Bit [4:0] Reserved

Mode 0 Timer A Control Register (P_TMA_CTRL, $0021)

BIT	7	6	5	4	3	2	1	0
Name	TMAES	CAPEG	TMACLK1	TMACLK0	TMADUT1	TMADUT0	TMAMOD1	TMAMOD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit 7 **TMAES:** Timer A enable/disable control.
0 : disable; (C_TMAES_DIS)
1 : enable. (C_TMAES_EN)

Bit 6 **CAPEG:** Timer A Capture edge selection.
0 : Rising; (C_TMACAP_RISE)
1 : Falling. (C_TMACAP_FALL)

Bit [5:4] **TMACLK[1:0]:** Timer A clock source selection bits
00 : Fosc (C_TMACLK_1)
01 : Fosc/2 (C_TMACLK_2)
10 : Fosc/4 (C_TMACLK_4)
11 : Fosc/16 (C_TMACLK_16)

Bit [3:2] **TMADUT[1:0]:** Timer A PWM duty selection
00 : 1/3 (C_TMADUT_3)
01 : 1/4 (C_TMADUT_4)

10 : 1/5 (C_TMADUT_5)

11 : 1/2 (C_TMADUT_2)

Bit [1:0] **TMAMOD[1:0]:** Timer A mode setting

00 : PWM (C_TMAMOD_WTC)

01 : PWM1 (enter the mode, PWM out always high) (C_TMAMOD_WOC)

10 : Capture (C_TMAMOD_CAP)

11: Envelop detect (C_TMAMOD_ENDE)

Mode 0 Timer A Count Low Byte Register (P_TMA_CNTL, $23) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMACNTL7	TMACNTL6	TMACNTL5	TMACNTL4	TMACNTL3	TMACNTL2	TMACNTL1	TMACNTL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMACNTL**[7：0]: Timer A low byte 8-bit pre-value for the counter.

Read : Timer A Count Low Byte Value(R)

Write : Timer A Pre-Load Count Low Byte Value (W)

Mode 0 Timer A PWM Low Byte Period Register (P_TMA_PWML, $23) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMAPRDL7	TMAPRDL6	TMAPRDL5	TMAPRDL4	TMAPRDL3	TMAPRDL2	TMAPRDL1	TMAPRDL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMAPRDL**[7：0]: Timer A low byte 8-bit period value for the PWM.

Read : Timer A Count Low Byte Value(R)

Write : PWM signal carrier signal Pre-load Period Low Byte Value (W)

Mode 0 Timer A Capture Low Byte Width Register (P_TMA_CAPL, $23) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMAWIDL7	TMAWIDL6	TMAWIDL5	TMAWIDL4	TMAWIDL3	TMAWIDL2	TMAWIDL1	TMAWIDL0
Access	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMAWIDL**[7：0]: Timer A low byte 8-bit width value for the CAPTURE.

Read : Capture mode received carrier signal Period (frequency) Low Byte Value(R)

Mode 0 Timer A Envelope Low Byte Pre-value Register (P_TMA_ENVL, $23) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TMAPRVL7	TMAPRVL6	TMAPRVL5	TMAPRVL4	TMAPRVL3	TMAPRVL2	TMAPRVL1	TMAPRVL0
Access	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Bit [7:0] **TMAPRVL**[7：0]: Timer A low byte 8-bit Pre-value for the ENVELOPE DETECT.

Write : Envelope detect mode received carrier signal Pre-load Period (frequency) Low Byte Value (W)

Mode 0 Timer A Count High Byte Register (P_TMA_CNTH, $24) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	TMACNTH3	TMACNTH2	TMACNTH1	TMACNTH0
Access	-	-	-	-	R/W	R/W	R/W	R/W
Default	-	-	-	-	0	0	0	0

Bit [7:4] Reserved

Bit [3:0] **TMACNTH**[3：0]: Timer A high byte 4-bit pre-value for the counter.

Read : Timer A Count High Byte Value (R)

Write : Timer A Pre-Load Count High Byte Value (W)

Mode 0 Timer A PWM High Byte Period Register (P_TMA_PWMH, $24) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	TMAPRDH3	TMAPRDH2	TMAPRDH1	TMAPRDH0
Access	-	-	-	-	R/W	R/W	R/W	R/W
Default	-	-	-	-	0	0	0	0

Bit [7:4] Reserved

Bit [3:0] **TMAPRDH**[3：0]: Timer A high byte 4-bit period value for the PWM.

Read : Timer A Count High Byte Value(R)

Write : PWM signal carrier signal Pre-load Period High Byte Value (W)

Mode 0 Timer A Capture High Byte Width Register (P_TMA_CAPH, $24) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	TMAWIDH3	TMAWIDH2	TMAWIDH1	TMAWIDH0
Access	-	-	-	-	R	R	R	R
Default	-	-	-	-	0	0	0	0

Bit [7:4] Reserved

Bit [3:0] **TMAWIDH**[3：0]: Timer A high byte 4-bit period value for the CAPTURE.

Read : Timer A Width High Byte Value (R)

Mode 0 Timer A Envelope High Byte Width Register (P_TMA_ENVH, $24) (R/W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	TMAPRVH3	TMAPRVH2	TMAPRVH1	TMAPRVH0
Access	-	-	-	-	W	W	W	W
Default	-	-	-	-	0	0	0	0

Bit [7:4] Reserved

Bit [3:0] **TMAPRVH**[3：0]: Timer A high byte 4-bit period value for the ENVELOPE DETECT.

Write : Envelope detect mode received carrier signal Pre-load Period (frequency) High Byte Value (W)

[Example] 5-9 Set Timer A as PWM with carrier signal mode.

```
LDA   #C_TIMAB_UP
STA   P_TIM_SEL                    ;set timer as up count
LDA   #$FC                         ; Before starting timer, set Timer A counter initial value first
STA   P_TMA_PWML                   ; set low 8-bit pre-value
LDA   #$0F
STA   P_TMA_PWMH                   ;set high 4-bit pre-value
LDA   #C_TMAES_EN + #C_TMACLK _4 +#C_TMADUT_3 + #C_TMAMOD_WTC
STA   P_TMA_CTRL                   ;Set clock source Fosc/4, 1/3duty, PWM with carrier signal mode
```

5.8.2.4. PWM Carrier Signal Algorithm

The frequency of PWM carrier signal (F_{PWM}) generated by Timer A depends on three factors.

- The initial value (V_{REG}=12-bit Preload PREIOD) is filled into high-byte (low-nibble) register (P_TMA_PWMH [3:0]) and low-byte register (P_TMA_PWML [7:0])
- The duty of the carrier signal (DUT= PWM DUTY).
- The frequency of timer A clock source (F_{timer})

V_{REG} = P_TMA_PWMH[4:0]+P_TMA_PWML[7:0]
DUT = one of (1/3, 1/4, 1/5, 1/2), defined by P_TMA_CTRL[3:2]
If
F_{timer} = F_{osc}/1 or F_{osc}/2, defined by P_TMA_CTRL[5:4]
Then
V_{REG} = 4097- F_{timer} / F_{PWM} * DUT

For example, if user needs to generate 38 KHz 1/3 duty PWM carrier frequency and TIMER clock source is 4MHz/1 (system clock is 4MHz).

Condition: F_{PWM} = 38 KHz, F_{timer} =4MHz, DUT=1/3
V_{REG} = 4097 – (4M/38K)*1/3 = 4062 =FDEH

Then the result FDEH can be written into the PWM high/low register, and the 38 KHz PWM signal is generated.

V_{REG} = P_TMA_PWMH[4:0]+P_TMA_PWML[7:0]
DUT = one of (1/3, 1/4, 1/5, 1/2), defined by P_TMA_CTRL[3:2]
If
F_{timer} = F_{osc}/4, F_{osc}/16 , defined by P_TMA_CTRL[5:4]
Then
V_{REG} = 4096- F_{timer} / F_{PWM} * DUT

For example, if user need to generate 38 KHz 1/3 duty PWM carrier frequency, and system frequency is 4MHz. and F_{OSC}/4 is selected as timer clock.

Condition: F_{PWM} = 38 KHz, F_{timer} = 4MHz/4, DUT=1/3
V_{REG} = 4096 – (1M/38K)*1/3 = 4087 =FF7H

Then the result FF7H can be written into the PWM high/low register, and the 38 KHz PWM signal is generated.

[Example] 5-6 Set Timer A as PWM with carrier signal mode and the carrier frequency is 38 KHz with 1/3 duty (clock source=Fosc/1).

```
LDA     #C_TIMAB_UP
STA     P_TIM_SEL                                   ;set timer as up count and enable PWM output function
LDA     #$DE                                        ; Before starting timer, set Timer A counter initial value first
STA     P_TMA_ PWML                                  ; set low 8-bit pre-value
LDA     #$0F
STA     P_TMA_ PWMH                                  ;set high 4-bit pre-value
LDA     #C_TMAES_EN + #C_TMACLK _1 +#C_TMADUT_3 + #C_TMAMOD_WTC
STA     P_TMA_CTRL                                   ;Set clock source Fosc/1, 1/3duty, PWM with carrier signal mode
```

[Example] 5-7 Set Timer A as PWM with carrier signal mode and the carrier frequency is 38 KHz with 1/3 duty (clock source=Fosc/4).

```
LDA     #C_TIMAB_UP
STA     P_TIM_SEL                                   ;set timer as up count
LDA     #$F7                                         ; Before starting timer, set Timer A counter initial value first
STA     P_TMA_ PWML                                  ; set low 8-bit pre-value
LDA     #$0F
STA     P_TMA_ PWMH                                  ;set high 4-bit pre-value
LDA     #C_TMAES_EN + #C_TMACLK _4 +#C_TMADUT_3 + #C_TMAMOD_WTC
STA     P_TMA_CTRL                                   ;Set clock source Fosc/4, 1/3 duty, PWM with carrier signal mode
```

5.8.3. Mode 0 Timer B (12-bit up count timer)

When Timer A is selected as 12-bit up count timer via configuring the corresponding bit of register (P_TIM_SEL[7]), the Timer B is selected as 12-bit up count timer too. The Timer B is special for envelope signal generation in IR controller application. The 12-bit timer is an up counter with input clock selectable (Fosc/1, Fosc/4, Fosc/64, TMACAR) via configuring the control register P_TMB_CTRL [5:4] (TMBCLK [1:0]). And the value of low-byte register (P_TMB_CNTL) and high-byte (low-nibble) register (P_TMB_CNTH) would be reloaded into the 12-bit up counter and an interrupt (TMBOIF) would be generated whenever an overflow occurs. The interrupt frequency can be freely selected by selecting different clock source and configuring the low-byte register and high-byte (low-nibble) register with different values.

The Timer B module has the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$FFF to #$000

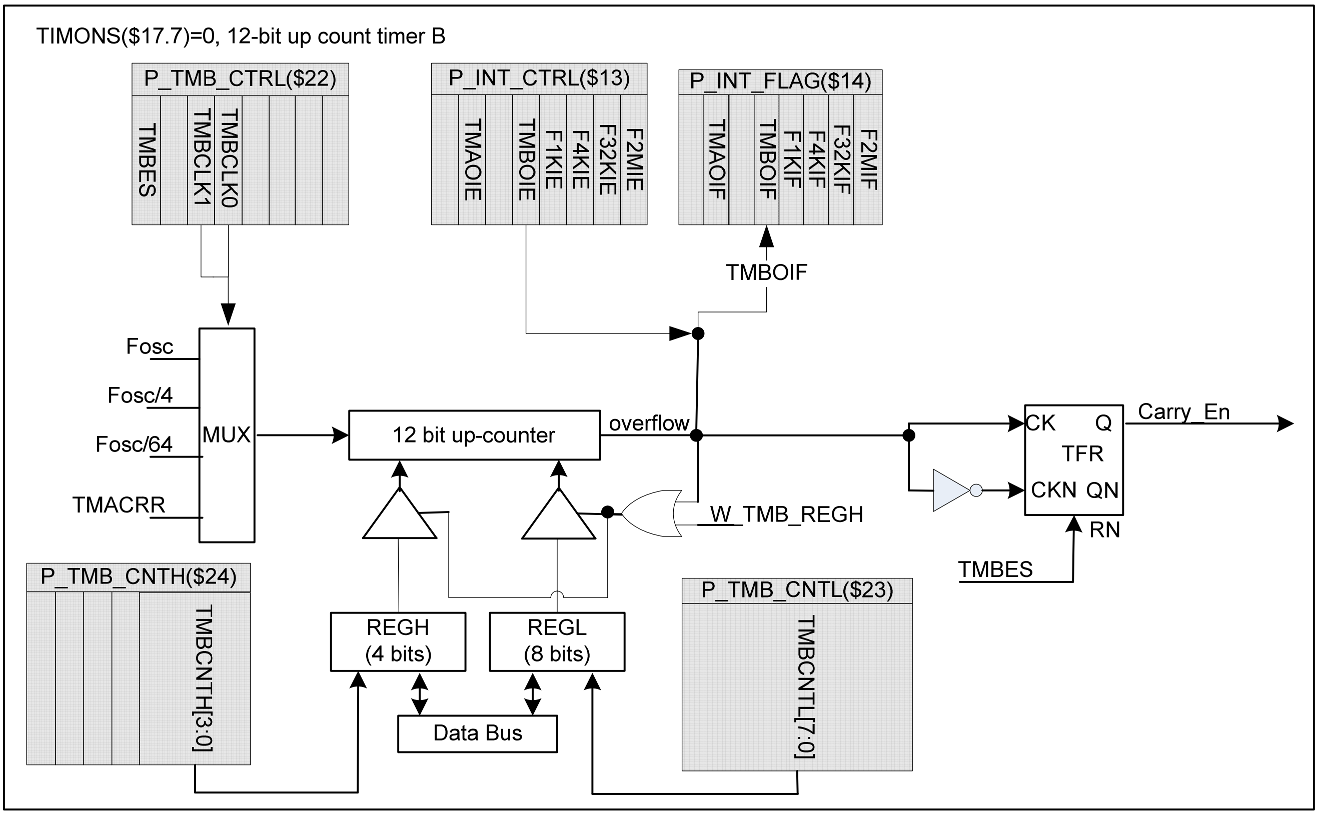


Figure 5-30 Mode 0 Timer B block diagram

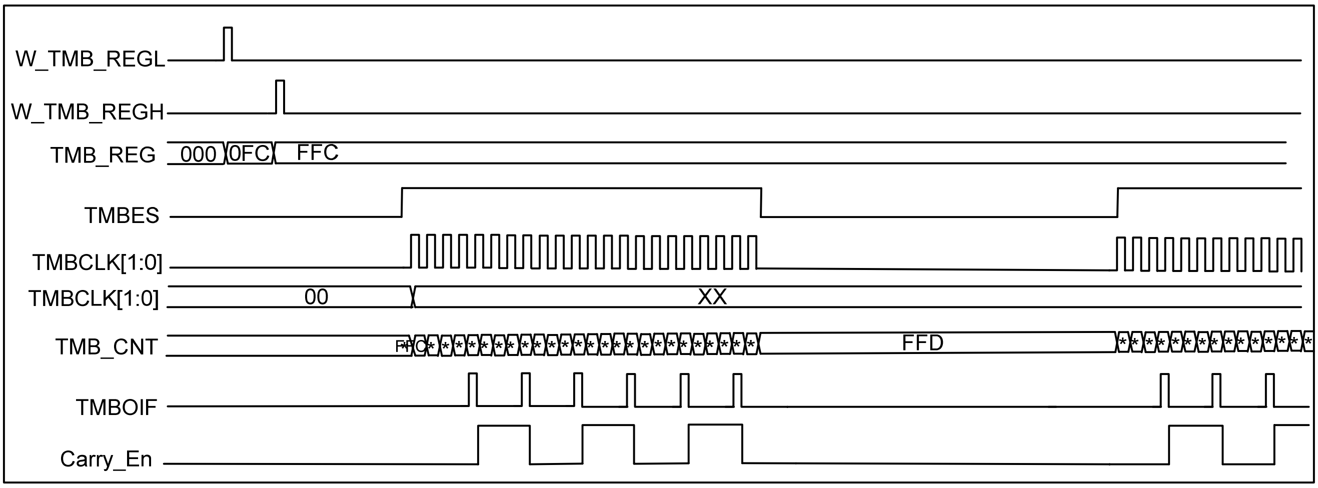


Figure 5-31 The Waveform of mode 0 Timer B

Mode 0 Timer B Control Register (P_TMB_CTRL, $0022)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBES	-	TMBCLK1	TMBCLK0	-	-	-	-
ACCESS	R/W	-	R/W	R/W	-	-	-	-
DEFAULT	0	-	0	0	-	-	-	-

Bit [7] **TMBES**: Timer B enable/disable control selected bit.

0 : disable (C_TMBES_DIS)

1 : enable (C_TMBES_EN)

Bit [6] Reserved

Bit [5:4] **TMBCLK[1：0]**: Timer B clock source selected bits

00 : Fosc (C_TMBCLK_1)

01 : Fosc/4 (C_TMBCLK_4)

10 : Fosc/64 (C_TMBCLK_64)

11 : TMACRR (C_TMBCLK_TMACRR)

Bit [3:0] **Reserved**

Mode 0 Timer B low 8-bit data Register (P_TMB_CNTL, $0025)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBCNTL7	TMBCNTL6	TMBCNTL5	TMBCNTL4	TMBCNTL3	TMBCNTL2	TMBCNTL1	TMBCNTL0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:0] **TMBCNTL**[7：0]: Timer B low byte 8-bit pre-value for the counter.

Read : Timer B Count Low Byte Value (R)

Write : Timer B Pre-Load Count Low Byte Value (W)

Mode 0 Timer B high 4-bit data Register (P_TMB_CNTH, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	TMBCNTH3	TMBCNTH2	TMBCNTH1	TMBCNTH0
ACCESS	-	-	-	-	R/W	R/W	R/W	R/W
DEFAULT	-	-	-	-	0	0	0	0

Bit [7:4] Reserved

Bit [3:0] **TMBCNTH**[3：0]: Timer B High byte 4-bit pre-value for the counter.

Read : Timer B Count High Byte Value (R)

Write : Timer B Pre-Load Count High Byte Value (W)

[Example] 5-8 Set Timer B selects timer A carrier signal as counter clock.

```
LDA   #C_TIMAB_UP
STA   P_TIM_SEL                     ; set Timer A/B as 12-bit up count timers
LDA   #$FC                          ; Before starting timer, set Timer B counter initial value first
STA   P_TMB_CNTL                    ; set low 8-bit pre-value
LDA   #$0F
STA   P_TMB_CNTH                    ; set high 4-bit pre-value
LDA   #C_TMBES_EN + #C_TMBCLK_TMACRR
STA   P_TMB_CTRL                    ;Set clock source for TMA_Carrier
```

5.9. Mode 1 Timer A (8-bit down count timer)

When Timer A is selected as 8-bit down count timer via configuring the corresponding bits of the control register (P_FUN_S[7]), the Timer A is special for generating carrier signal in IR control application. The timer A's input clock is selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16), which can be configured by control register P_TMA_CTRL[5:4]. Timer A provides with two PWM mode, and the PWM signal is send to IR TX (REM) pin. The driver current of these two kinds of PWM are programmable by configuring TX PWM driving current control source register (P_PWM_DRV [3]).

8-bit down count Timer A module has the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$00 to #$FF
- Support PWM with carrier signal mode
- Support PWM without carrier signal mode
- Support capture mode for learning function
- Support envelope detect mode for learning function

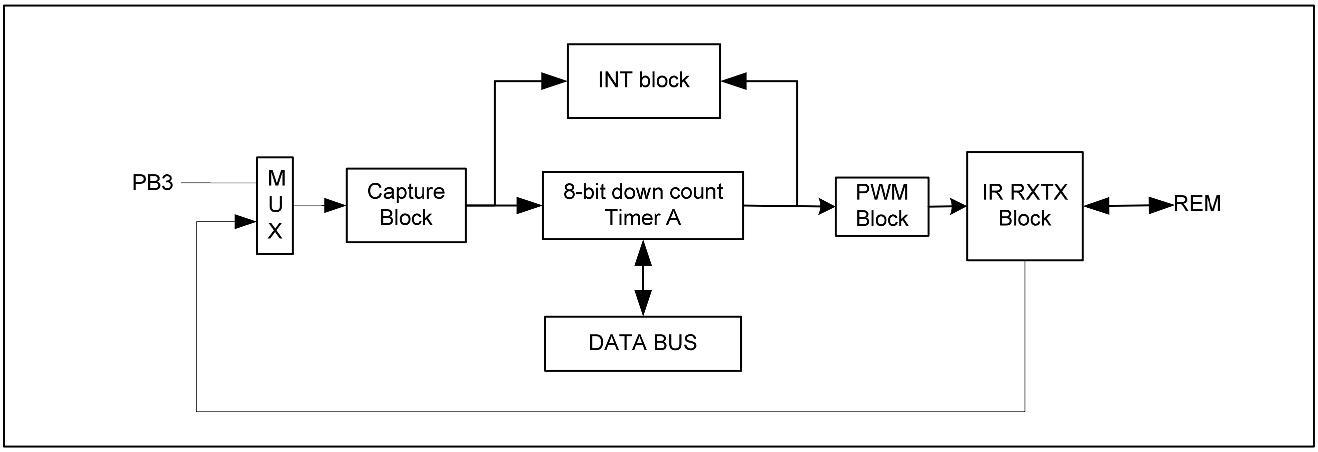


Figure 5-32 Mode 1 timer A block diagram

5.9.1. Mode 1 timer A PWM with carrier signal mode

The Timer A can be configured as PWM mode for generating carrier signal. In PWM with carrier signal mode, the 8-bit timer is an down counter with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When Timer A is started, the value of 8-bit cycle width (frequency) set register would firstly be loaded into the 8-bit counter and the value of 8-bit high pulse width (duty) set register would be loaded into the compare unit. And then the counter starts count down from the loaded value. PWM initial output low, and if the counter value is same as the value in compare unit, the PWM would switch to high. If an overflow occurs, the PWM switch to low once again, and the value of frequency register (P_TMA_PWMF, $23) and duty register (P_TMA_PWMD, $24) would be reloaded into the counter and the compare unit automatically and the counter starts count down again. So the carrier signal with frequency and duty programmable can be generated by this PWM mode via configuring these two registers. The carrier signal's enabled or disabled bit can be controlled by two methods depended on which clock source is selected by timer B. If timer B is selected one of the first three clock source (Fosc, Fosc/4 or Fosc/64) by P_TMB_CTRL [5:4] (TMBCLK [1:0]), the timer A's carrier signal on/off is controlled by timer A's enable/ disable control bit (TMAES) directly. In addition, PWM output function also can be disabled by writing 1 to register IRENB($17.6).

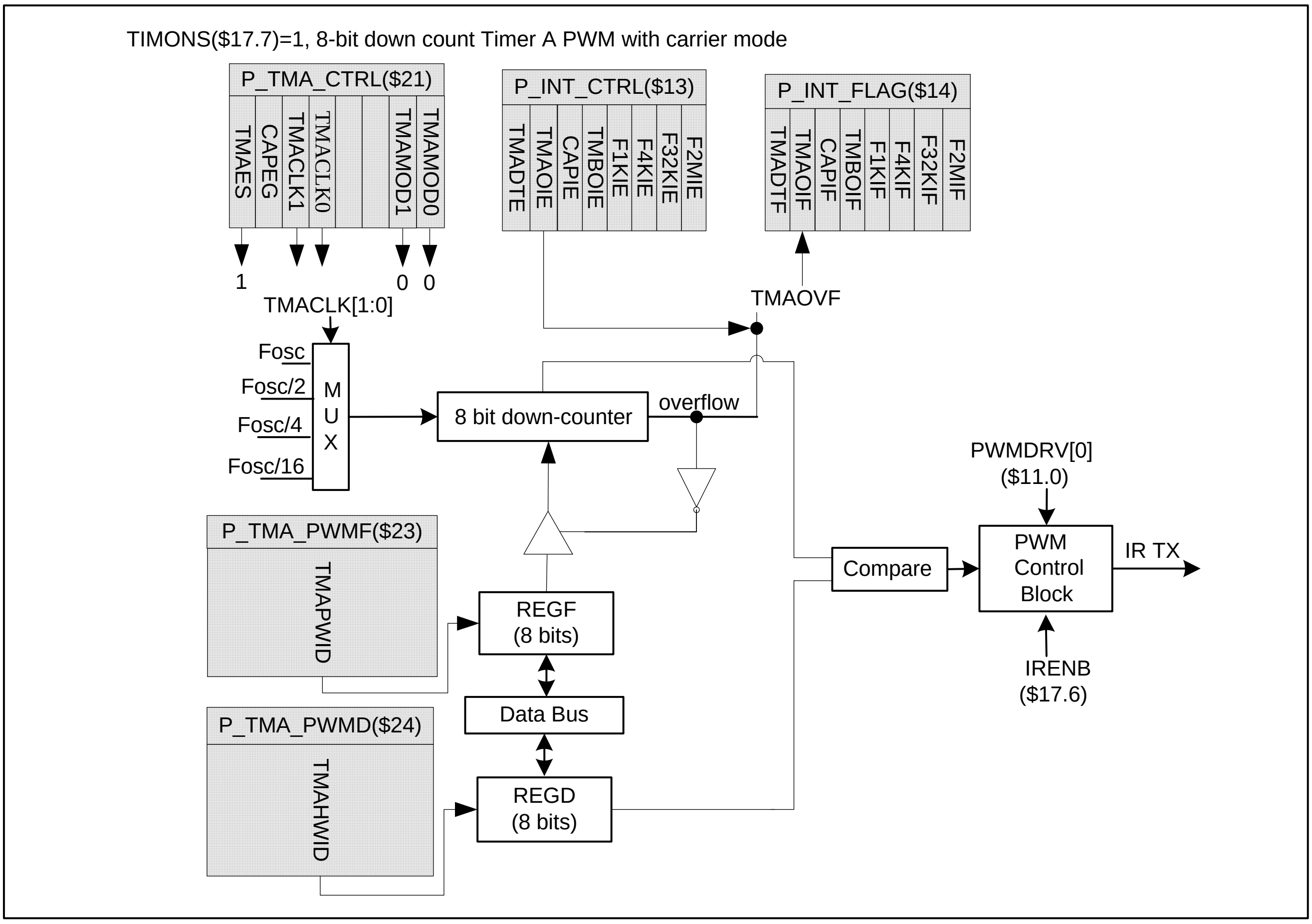


Figure 5-33 Mode 1 Timer A PWM mode diagram

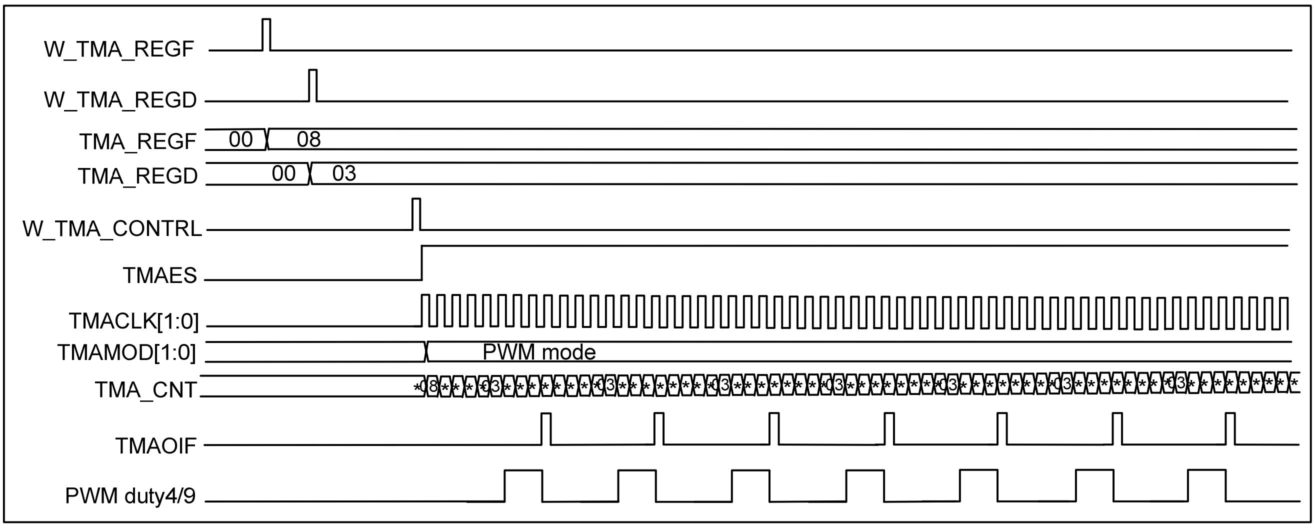


Figure 5-34 Mode 1 Timer A Normal PWM generation without envelop

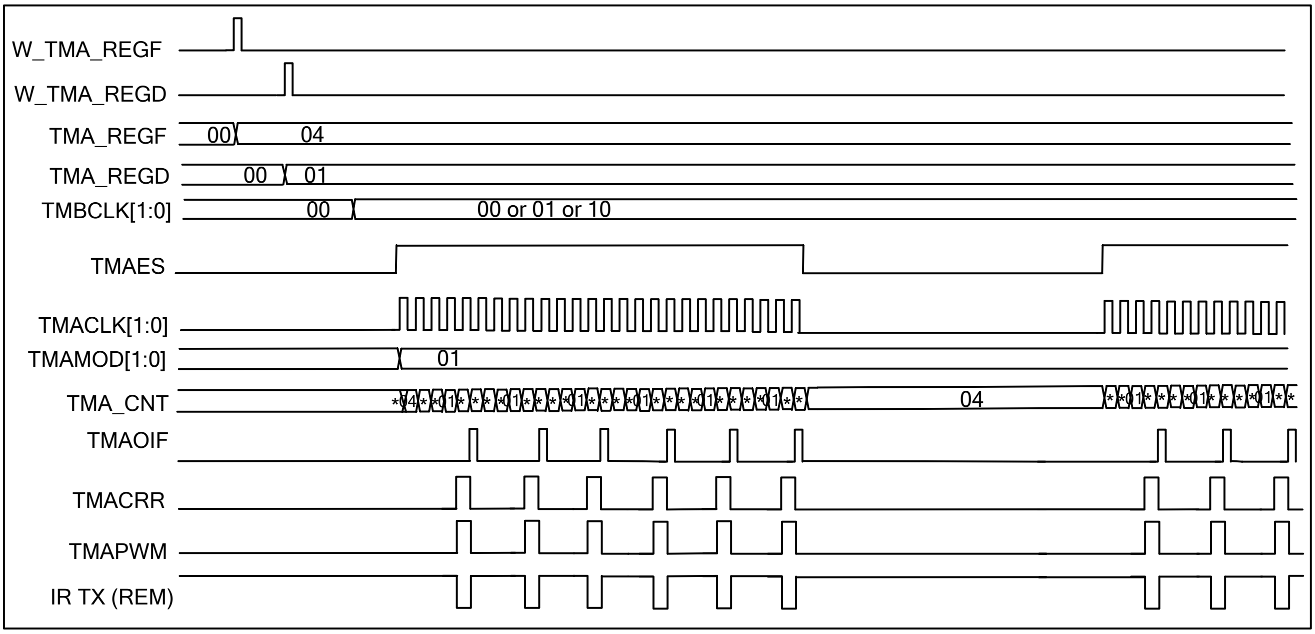
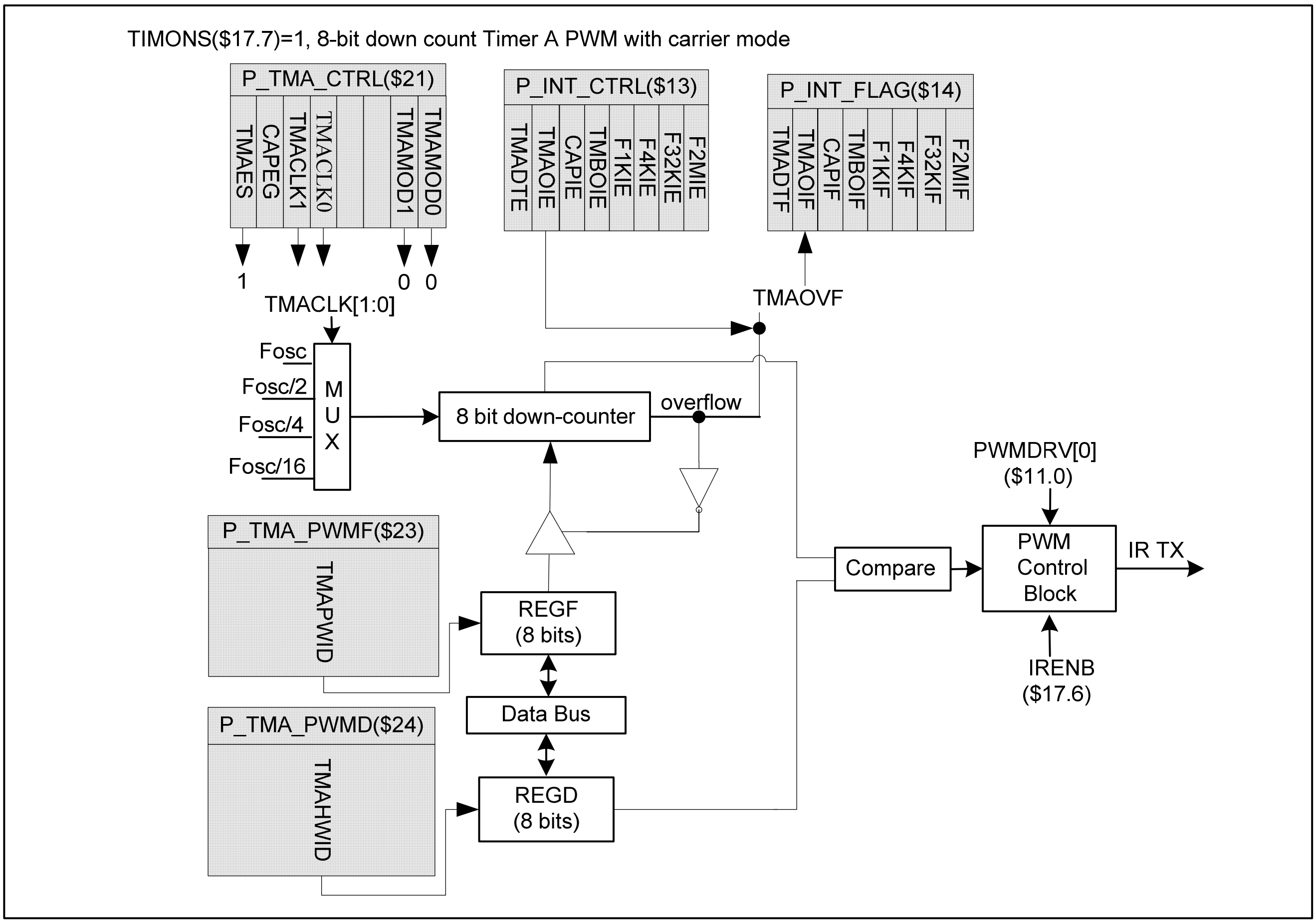


Figure 5-35 The Waveform of mode 1 timer A PWM with carrier signal mode (1/5 duty, on/off control by TMAES)

Another method to generate envelope PWM signal is that Timer A and Timer B must be used together. Timer A must generate carry clock at first, which is same as normal PWM generation. Then enable Timer B and select Timer A carrier signal as its input clock. And Timer B register must be written in the right data, which represents the carry number. When TMBOVF happen, another value must be written into Timer B register, which represents the no carry clock number. Envelop with carrier is on or off only when Timer B overflow events occur one by one. Then, the envelop PWM signal will be generated at REM port at last.



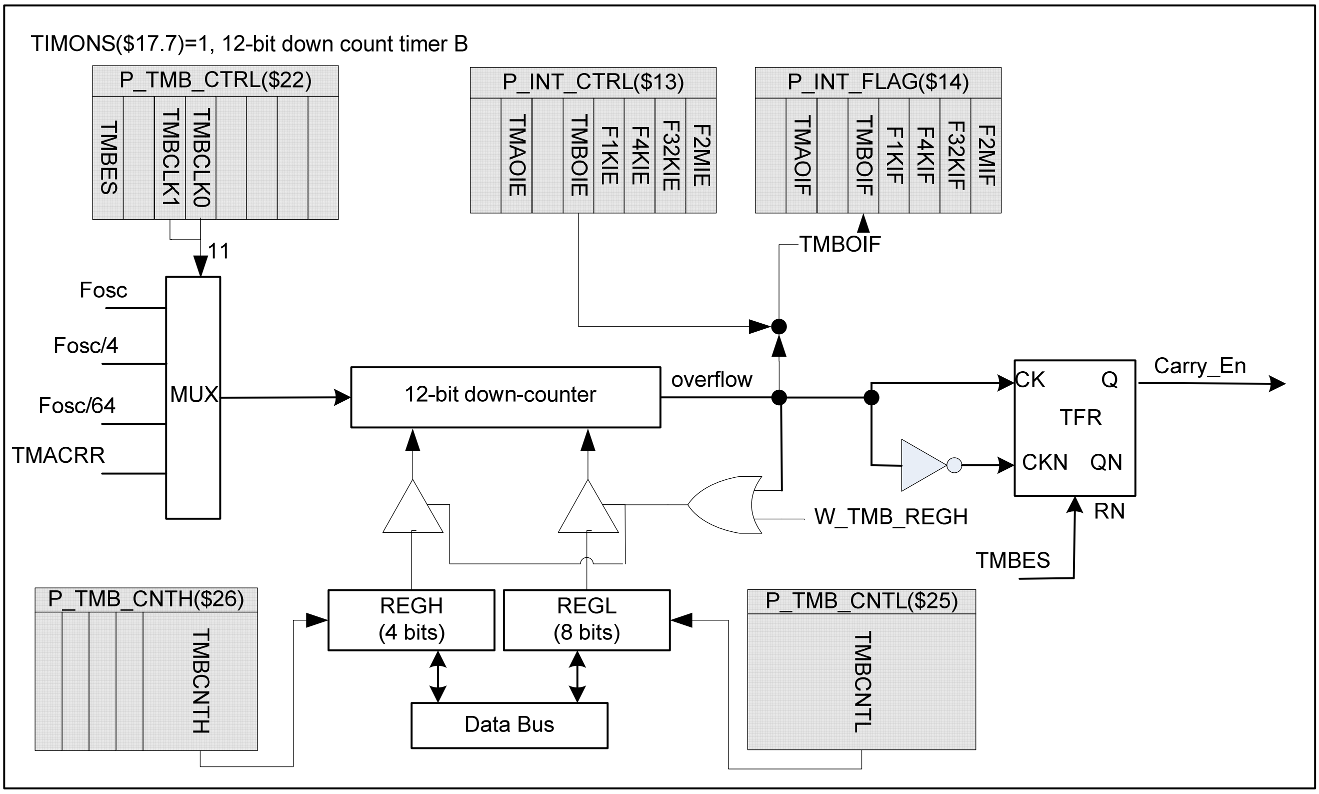


Figure 5-36 Envelope PWM Generated by mode 1 Timer A & mode 1 Timer B diagram

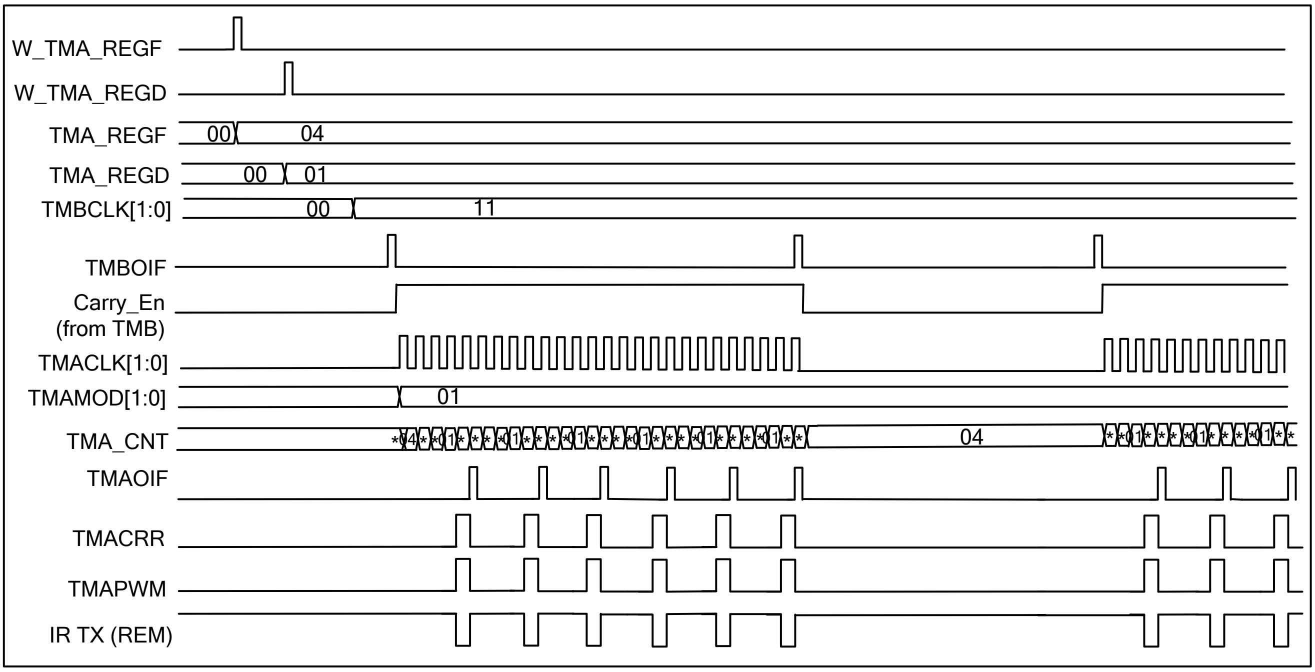


Figure 5-37 The Waveform of mode 1 timer A PWM with carrier signal mode (1/5 duty, on/off control by mode 1Timer B overflow events)

5.9.2. Mode1 PWM without carrier signal mode

PWM without carrier signal mode is used to generate envelop PWM signal without carrier signal. In this mode, IR TX (REM) pin just output high or low, and is controlled by TimerA's enable or disable control bit or Timer B's overflow events in turn. The same as PWM with carrier signal mode, the 8-bit timer is an down counter with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When the Timer A is started, the value of 8-bit pre-value Register (P_TMA_PWMF, $23) will firstly be loaded into the 8-bit counter and then the counter starts to count down from the loaded value. If an overflow occurs, the value of pre-value register will be reloaded into the counter automatically and the counter starts to count down again. The internal carrier signal is generated but is not sent to IR TX pin.

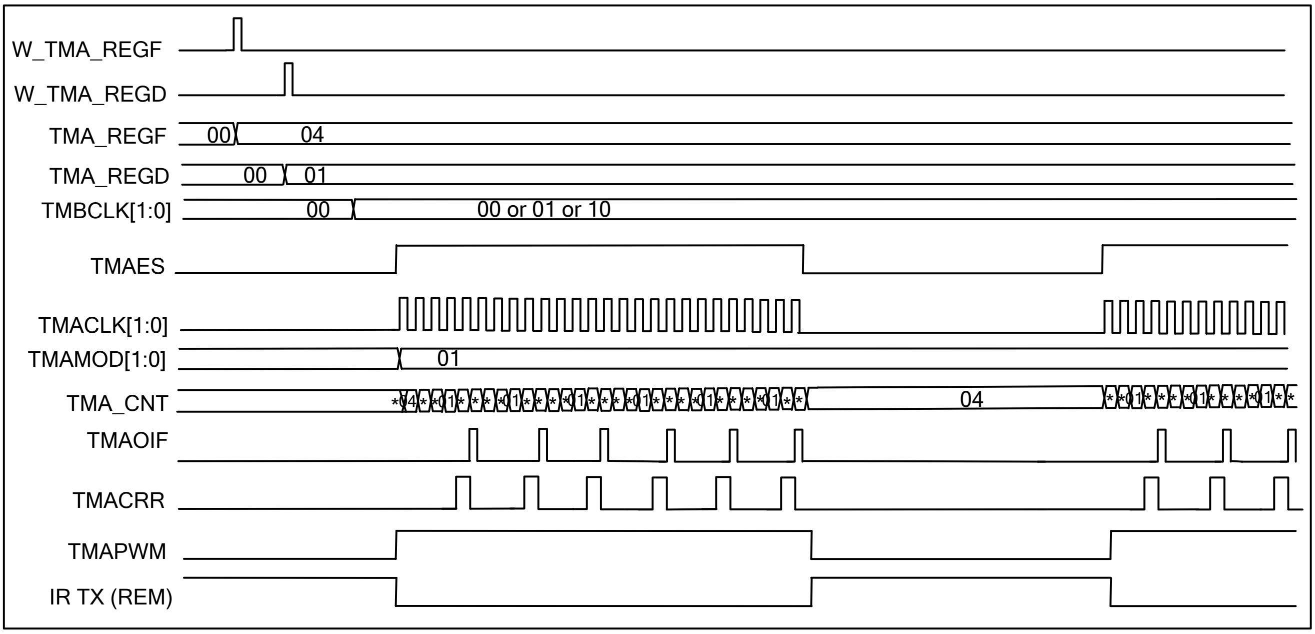


Figure 5-38 The Waveform of mode 1 timer A PWM without carrier signal mode (on/off control by TMAES)

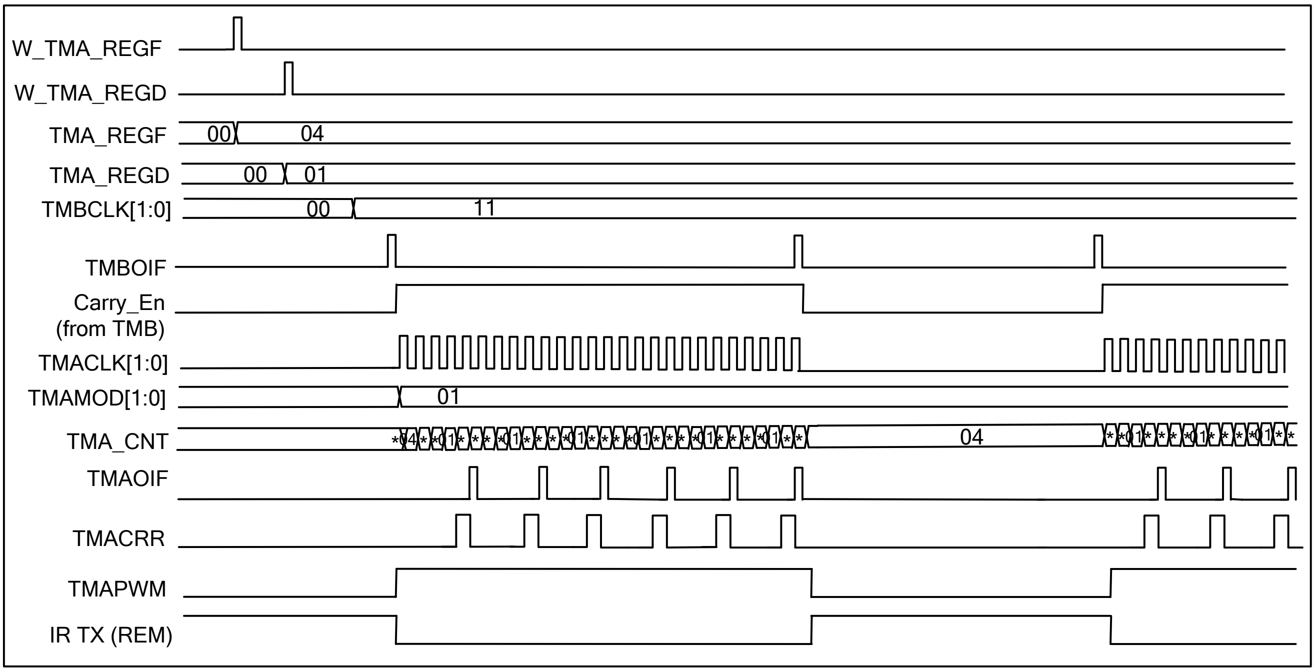


Figure 5-39 The Waveform of mode 1 timer A PWM without carrier signal mode (on/off control by mode 1 Timer B overflow events)

5.9.3. Mode1 Capture & Envelope detect mode

In IR learning function application, Timer A should be configured as capture mode for measuring the frequency of input signal from RX pin. In capture mode, the 8 bit timer is a down counter which counts from FFH with input clock selectable (Fosc/1, Fosc/2, Fosc/4, Fosc/16). When rising or falling (selectable via P_TMA_CTRL) edge of RX is captured, the value of the counter would be loaded into capture register (P_TMA_CAPF, $24), at the same time, it generates an interrupt (CAPIF) and then the counter is set to FFH. When the timer overflows, the overflow interrupt (TMAOIF) occurs. The input carrier signal cycle time is recorded in capture Register (P_TMA_CAPF). Of course, if the recorded time data is greater than the largest data can be loaded, the overflows of the timer A should be taken into account..

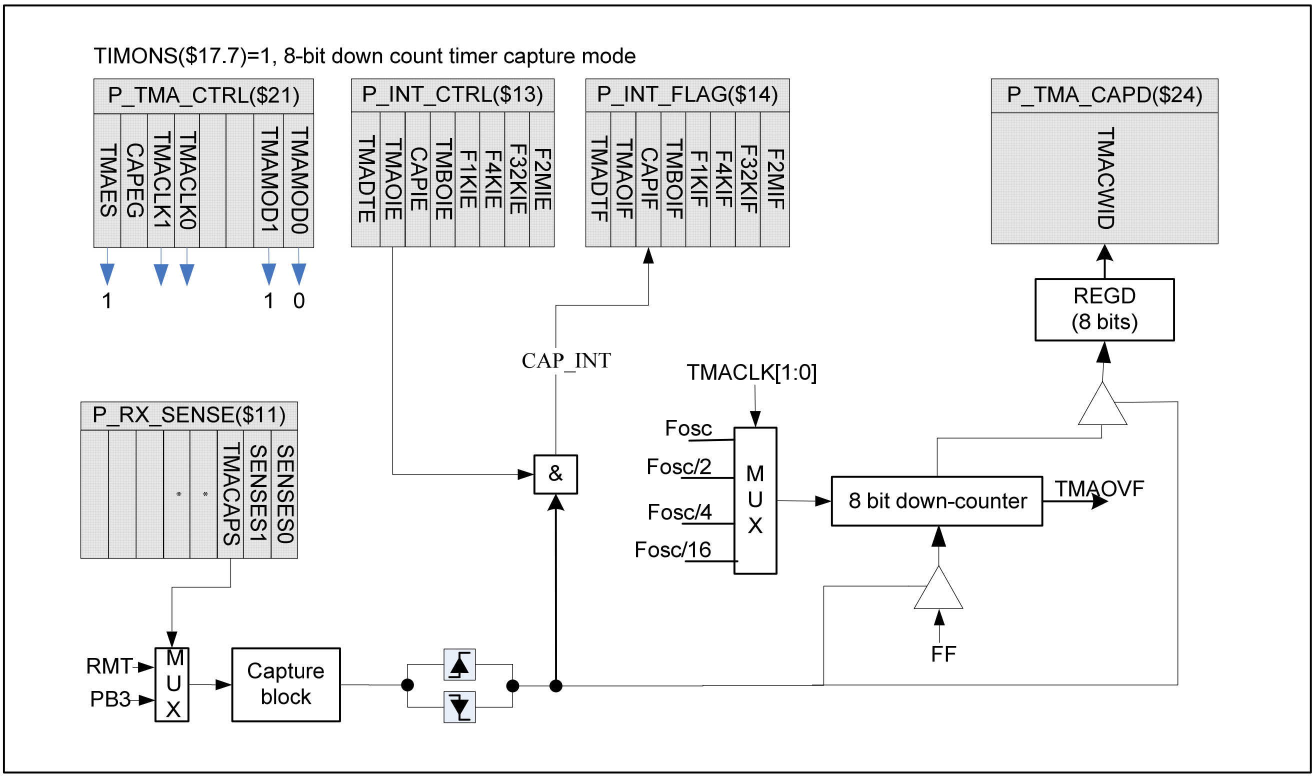


Figure 5-40 Mode 1 Timer A block diagram (Capture Mode)

When timer A is 8-bit down count timer, the envelope of input signal from RX(REM) pin can be measured just only in envelope detect mode. And users do not need get carrier frequency value in capture mode firstly. The carrier frequency value also is captured into capture register (P_TMA_CAPF, $24) in envelope detect mode.

If the data received is a signal with carrier signal, the register NCDTEN ($17.5) should be cleared 0. In order to detect the envelope, firstly get the carrier frequency (named F_{CRR}) from capture register (P_TMA_ENVD, $24). Then load the value (1.5* F_{CRR}) to Timer A counter register (P_TMA_ENVF, $23). If the first rising or falling-edge of carry wave arrive, envelope interrupt occur (TMADTF=1) and ENVDET ($16.7) is set to '1', and the value (1.5* F_{CRR}) is loaded to Counter automatically, and Counter starts to count. If next rising or falling-edge arrive, the value (1.5* F_{CRR}) will be reloaded into the counter, and ENVDET ($16.7) not changed its status (still equal '1'). However, if the next carry wave does not arrive on time (that's over 1.5* F_{CRR}), the Timer A overflow happens resulting in envelope interrupt occurring, and make ENVDET ($16.7) changed to "0". So check ENVDET bit can know whether envelope exist or not.

And if the data received is a signal without carrier signal (judged by use software method), the register NCDTEN ($17.5) should be set 1. The signal (without carrier signal) received delivered to ENVDET ($16.7) directly. Also user can check ENVDET bit to get the input signal with carrier signal.

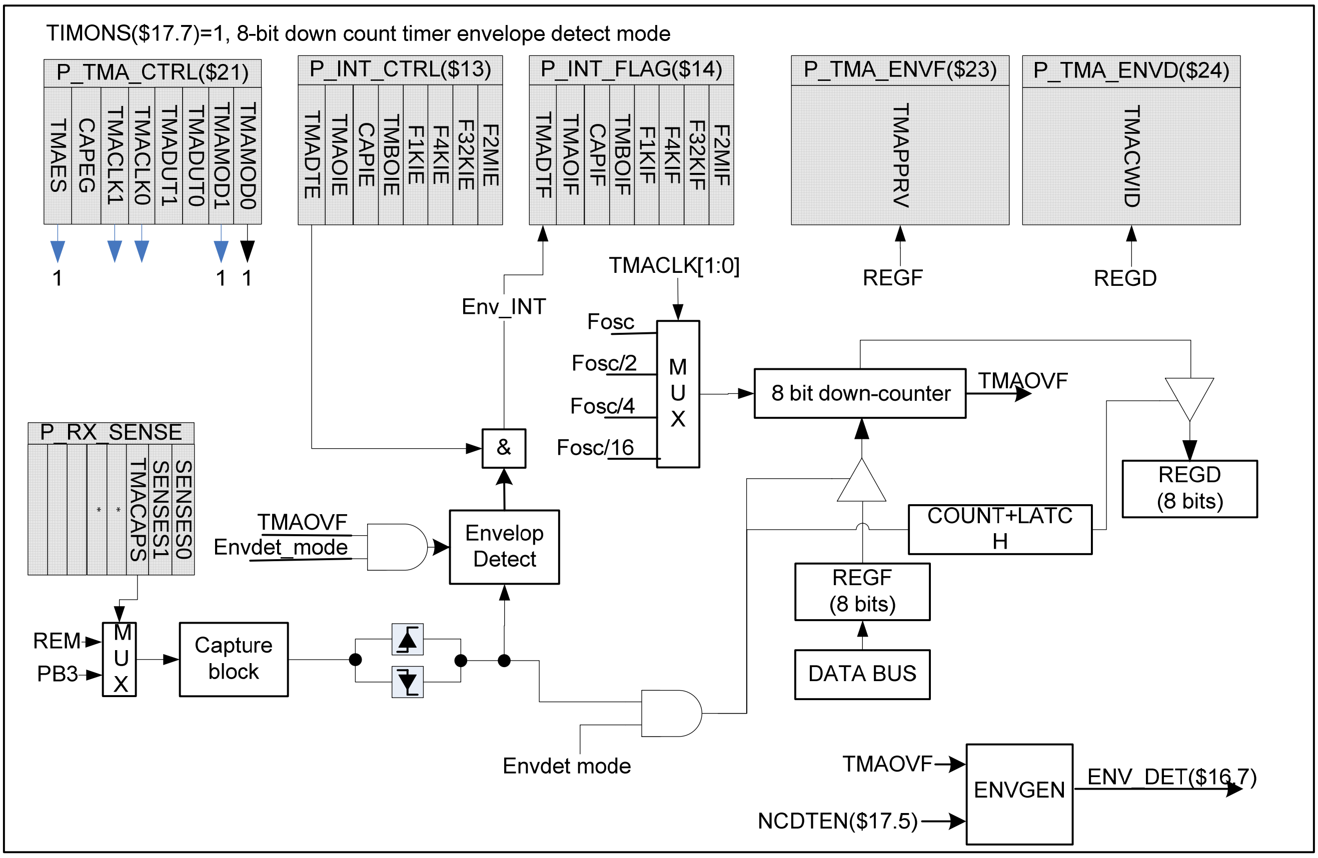


Figure 5-41 Mode 1 Timer A block diagram (Envelope detect Mode)

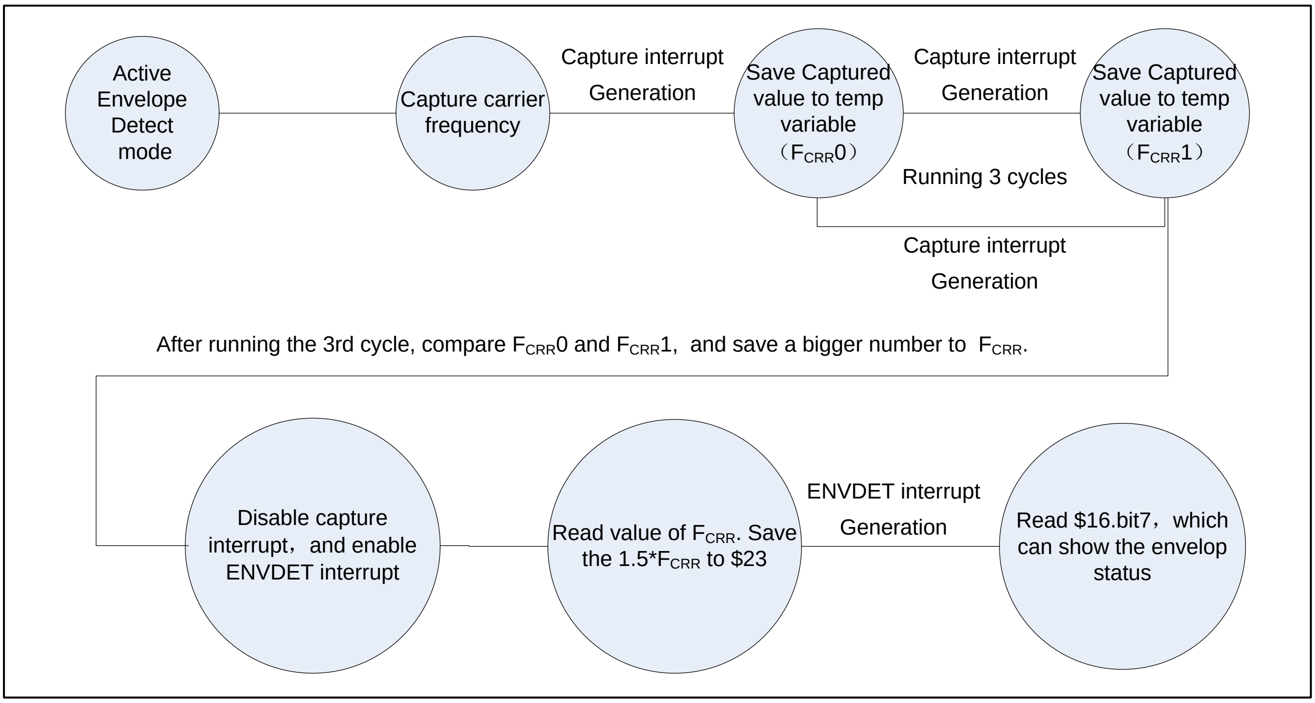


Figure 5-42 Mode 1Timer A envelope detect flow

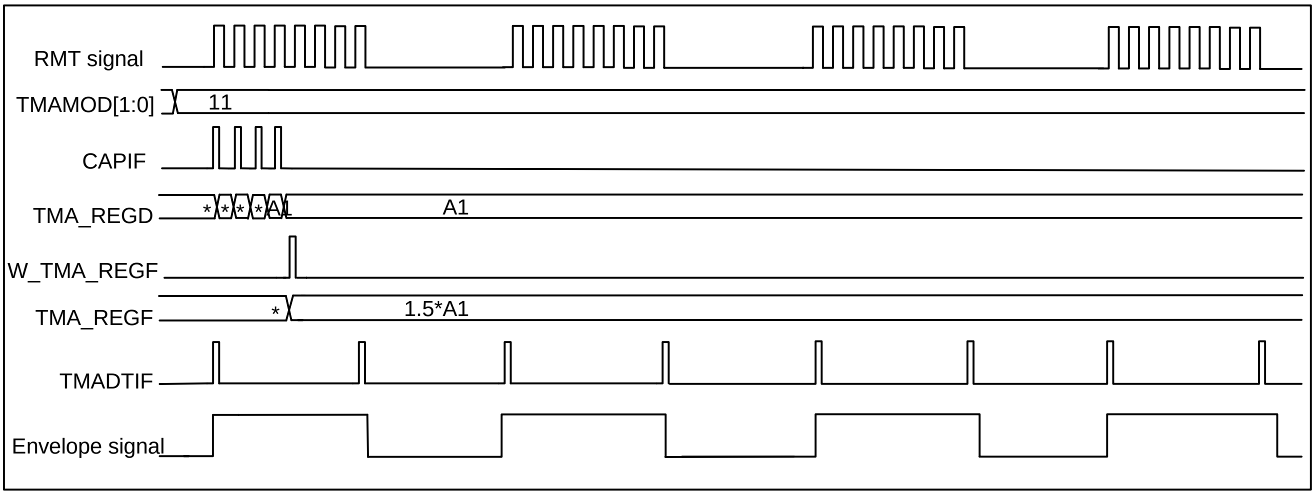


Figure 5-43 The waveform of mode 1 timer A envelope detect

Mode 1 Timer A Control Register (P_TMA_CTRL, $0021)

BIT	7	6	5	4	3	2	1	0
Name	TMAES	CAPEG	TMACLK1	TMACLK0	-	-	TMAMOD1	TMAMOD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit 7 **TMAES:** Timer A enable/disable control.
0, disable; (C_TMAES_DIS)
1, enable. (C_TMAES_EN)

Bit 6 **CAPEG:** Timer A Capture edge selection.
0, Rising; (C_TMACAP_RISE)
1, Falling. (C_TMACAP_FALL)

Bit [5:4] **TMACLK[1:0]:** Timer A clock source select bits
00 = Fosc (C_TMACLK_1)
01 = Fosc (C_TMACLK_2)
10 = Fosc/4 (C_TMACLK_4)
11 = Fosc/16 (C_TMACLK_16)

Bit [3:2] Reserved

Bit [1:0] **TMAMOD[1:0]:** Timer A mode setting
00: PWM (C_TMAMOD_WTC)
01: PWM1 (enter the mode, PWM out always high) (C_TMAMOD_WOC)
10: Capture (C_TMAMOD_CAP)
11: Envelop detect (C_TMAMOD_ENDE)

Mode 1 Timer A Count Register (P_TMA_CNTF, $23) (R)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TACNTF7	TACNTF6	TACNTF5	TACNTF4	TACNTF3	TACNTF2	TACNTF1	TACNTF0
Access	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X

Bit [7:0] **TACNTF**[7：0]: Timer A 8-bit pre-value for the counter.
Read: Timer A Count Value(R)

Mode 1 Timer A PWM Carrier Signal Period (Frequency) Register for PWM Mode(P_TMA_PWMF, $23) (W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TAPWMF7	TAPWMF6	TAPWMF5	TAPWMF4	TAPWMF3	TAPWMF2	TAPWMF1	TAPWMF0
Access	W	W	W	W	W	W	W	W
Default	X	X	X	X	X	X	X	X

Bit [7:0] **TAPWMF**[7：0]: Timer A carrier signal period(frequency) value for the PWM.
Write: Timer A Pre-Load carrier signal Period(frequency) Value (W)

Mode 1 Timer A Carrier Signal Period (Frequency) Width Pre-value Register for Envelope Detect Mode(P_TMA_ENVF, $23) (W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TAENVF7	TAENVF6	TAENVF5	TAENVF4	TAENVF3	TAENVF2	TAENVF1	TAENVF0
Access	W	W	W	W	W	W	W	W
Default	X	X	X	X	X	X	X	X

Bit [7:0] **TAENVF**[7：0]: Timer A 8-bit width pre-value period(frequency) for the carrier signal of ENVELOPE.

Write: Pre-load period(frequency) Width Value for carrier signal of ENVELOPE (W)

Mode 1 Timer A PWM Carrier Signal High Pulse (Duty) Width Register for PWM Mode (P_TMA_PWMD, $24) (W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TAPWMID7	TAPWMID6	TAPWMID5	TAPWMID4	TAPWMID3	TAPWMID2	TAPWMID1	TAPWMID0
Access	W	W	W	W	W	W	W	W
Default	X	X	X	X	X	X	X	X

Bit [7:0] **TAPWMD**[7：0]: Timer A 8-bit high pulse (duty) value for the carrier signal of PWM.

Read: Timer A high pulse (duty) Value (R)

Write: Timer A Pre-Load carrier signal high pulse (duty) Value (W)

Mode 1 Timer A Received Carrier signal Period (Frequency) Width Register for Envelope Detect Mode (P_TMA_CAPF, $24) (R)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	TACAPF 7	TACAPF 6	TACAPF 5	TACAPF 4	TACAPF 3	TACAPF 2	TACAPF 1	TACAPF0
Access	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X

Bit [7:0] **TACAPF**[7：0]: Timer A 8-bit width value period(frequency) for the carrier signal

Read: period(frequency) Width Value for carrier signal of ENVELOPE or CAPTURE mode(R)

[Example] 5-9 Set Timer A as PWM with carrier signal mode.

```
LDA   #C_TIMAB_DN + #C_PWM_EN
STA   P_FUN_S                       ;set timer as down count and enable PWM output function
LDA   #$0F                          ; Before starting timer, set Timer A counter initial value first
STA   P_TMA_PWMF                    ; set Period pre-value
LDA   #$08
STA   P_TMA_PWMD                    ;set high pulse pre-value (DUTY=($08+1)/($0F+1)=9/16)
LDA   #C_TMAES_EN + #C_TMACLK _4 + #C_TMAMOD_WTC
STA   P_TMA_CTRL                    ;Set clock source Fosc/4, PWM with carrier signal mode
```

5.9.4. Mode1 PWM Carrier Signal Algorithm

The frequency of PWM carrier signal (F_{PWM}) generated by Timer A depends on three factors.

- The initial value (V_{REGF}=8-bit Preload PREIOD) is filled into register (P_TMA_PWMF [7:0]).
- The initial value (V_{REGD}=8-bit Preload HIGH PULSE Value) is filled into register (P_TMA_PWMD [7:0]).
- The frequency of timer A clock source (F_{timer})

V_{REGF} = P_TMA_PWMF[7:0]

V_{REGD} = P_TMA_PWMD[7:0]

If

F_{timer} = $F_{osc}/1$ or $F_{osc}/2$ or $F_{osc}/4$ or $F_{osc}/16$, defined by P_TMA_CTRL[5:4]

Then

$V_{REGF} = F_{timer} / F_{PWM} - 1$

$V_{REGD} = (F_{timer} / F_{PWM}) * DUT$

For example, if user needs to generate 38 KHz 2/5 duty PWM carrier frequency and TIMER clock source is 4MHz/1 (system clock is 4MHz).

Condition: F_{PWM} = 38 KHz, F_{timer} =4MHz, DUT=2/5

$V_{REGF} = F_{timer} / F_{PWM} - 1$ = 104 =68H

$V_{REGD} = (F_{timer} / F_{PWM}) * DUT$ = 42 =2AH

Then the result 68H and 2AH can be written into the PWM Period register and High pulse register separately, and the 38 KHz PWM signal is generated.

[Example] 5-10 Set Timer A as PWM with carrier signal mode and the carrier frequency is 38 KHz with 2/5 duty (clock source=Fosc/1).

```
LDA   #C_TIMAB_DN + #C_PWM_EN
STA   P_FUN_S                  ;set timer as down count and enable PWM output function
LDA   #$68                     ; Before starting timer, set Timer A counter initial value first
STA   P_TMA_ PWMF              ; set low Period pre-value
LDA   #$2A
STA   P_TMA_ PWMD              ;set high pulse pre-value(2/5 duty)
LDA   #C_TMAES_EN + #C_TMACLK _1 + #C_TMAMOD_WTC
STA   P_TMA_CTRL               ;Set clock source Fosc/1, PWM with carrier signal mode
```

5.10. Mode 1 Timer B (12-bit down count timer)

The Timer B is special for envelope signal generation in IR controller application. The 12-bit timer is an down counter with input clock selectable (Fosc/1, Fosc/4, Fosc/64, TMACAR) via configuring the control register P_TMB_CTRL [5:4] (TMBCLK [1:0]). And the value of low-byte register (P_TMB_CNTL) and high-byte (low-nibble) register (P_TMB_CNTH) would be reloaded into the 12-bit up counter and an interrupt (TMBOIF) would be generated whenever an overflow occurs. The interrupt frequency can be freely selected by selecting different clock source and configuring the low-byte register and high-byte (low-nibble) register with different values.

The Timer B module has the following features:

- Readable and writable
- Clock source selectable
- Interrupt-on-overflow from #$000 to #$FFF

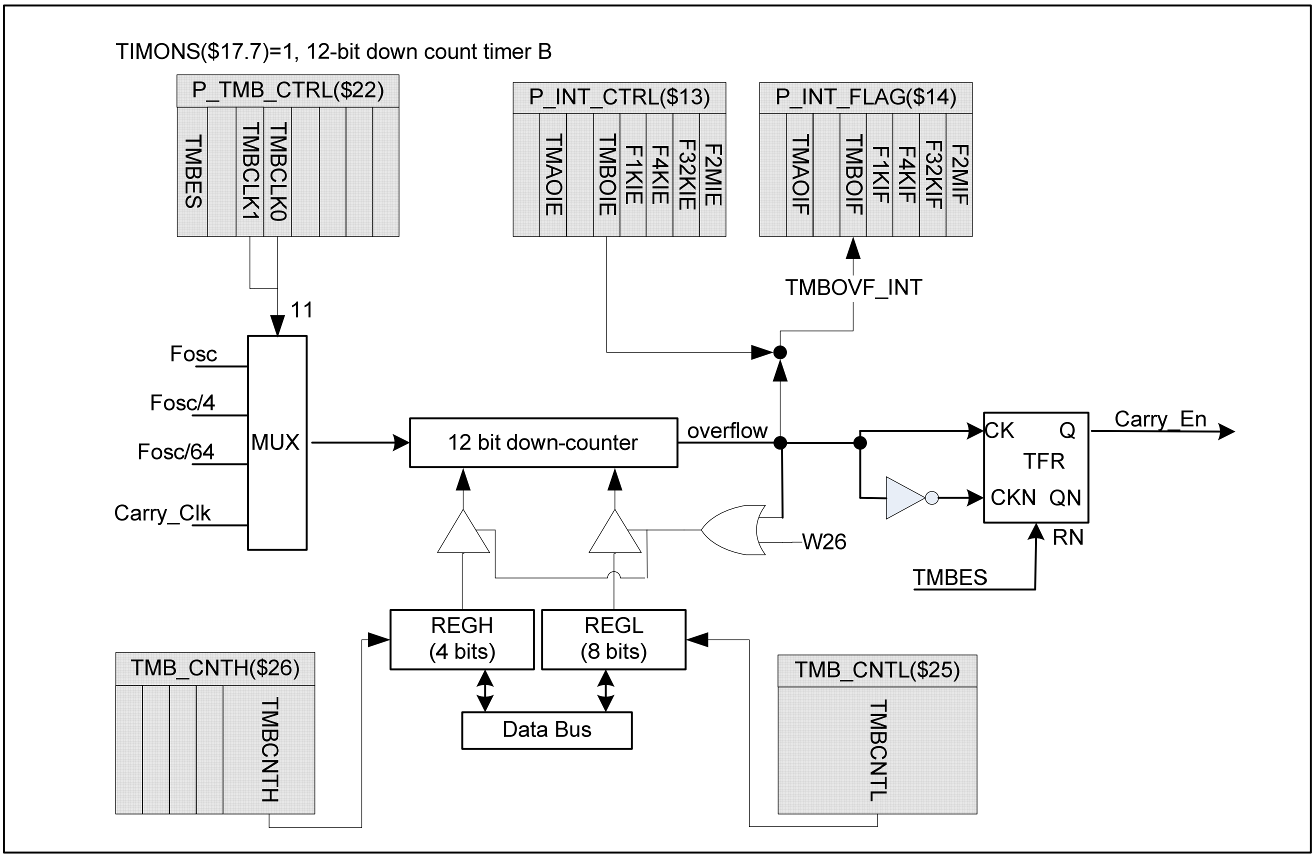


Figure 5-44 Mode 1 Timer B block diagram

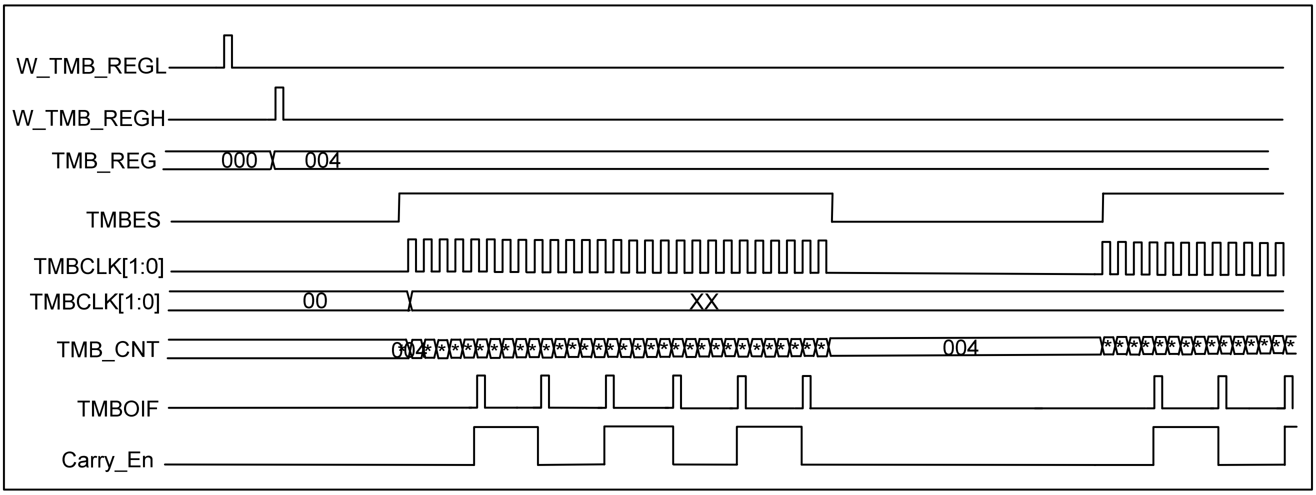


Figure 5-45 The Waveform of mode 1 Timer B

Mode 1 Timer B Control Register (P_TMB_CTRL, $0022)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TMBES	-	TMBCLK1	TMBCLK0	-	-	-	-
ACCESS	R/W	-	R/W	R/W	-	-	-	-
DEFAULT	0	-	0	0	-	-	-	-

Bit [7] **TMBES**: Timer B enable/disable control selected bit.
0 = disable (C_TMBES_DIS)
1 = enable (C_TMBES_EN)

Bit [6] Reserved

Bit [5:4] **TMBCLK[1：0]**: Timer B clock source selected bits
00 = Fosc (C_TMBCLK_1)
01 = Fosc/4 (C_TMBCLK_4)
10 = Fosc/64 (C_TMBCLK_64)
11 = TMACRR (C_TMBCLK_TMACRR)

Bit [3:0] **Reserved**

Mode 1 Timer B low 8-bit counter (P_TMB_CNTL, $0025)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TBCNTL7	TBCNTL6	TBCNTL5	TBCNTL4	TBCNTL3	TBCNTL2	TBCNTL1	TBCNTL0
ACCESS	R	R	R	R	R	R	R	R
DEFAULT	X	X	X	X	X	X	X	X

Bit [7:0] **TBCNTL**[7：0]: Timer B low byte 8-bit pre-value for the counter.
Read: Timer B Count Low Byte Value (R)

Mode 1 Timer B low 8-bit data Register (P_TMB_REGL, $0025)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	TBREGL7	TBREGL6	TBREGL5	TBREGL4	TBREGL3	TBREGL2	TBREGL1	TBREGL0
ACCESS	W	W	W	W	W	W	W	W
DEFAULT	X	X	X	X	X	X	X	X

Bit [7:0] **TBREGL**[7：0]: Timer B low byte 8-bit pre-value for the counter.
Write: Timer B Pre-Load Count Low Byte Value (W)

Mode 1 Timer B high 4-bit counter (P_TMB_CNTH, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	TMBCNTH3	TMBCNTH2	TMBCNTH1	TMBCNTH0
ACCESS	-	-	-	-	R	R	R	R
DEFAULT	-	-	-	-	X	X	X	X

Bit [7:4] Reserved

Bit [3:0] **TBCNTH**[3：0]: Timer B High byte 4-bit pre-value for the counter.

Read: Timer B Count High Byte Value (R)

Mode 1 Timer B high 4-bit data Register (P_TMB_REGH, $0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NAME	-	-	-	-	TMBCNTH3	TMBCNTH2	TMBCNTH1	TMBCNTH0
ACCESS	-	-	-	-	W	W	W	W
DEFAULT	-	-	-	-	X	X	X	X

Bit [7:4] Reserved

Bit [3:0] **TBREGH**[3：0]: Timer B High byte 4-bit pre-value for the counter.

Write: Timer B Pre-Load Count High Byte Value (W)

[Example] 5-11 Set Timer B selects timer A carrier signal as counter clock.

```
        LDA     #C_TIMAB_DN + #C_PWM_EN
        STA     P_FUN_S                         ;set timer as down count and enable PWM output function
        LDA     #$FC                            ; Before starting timer, set Timer B counter initial value first
        STA     P_TMB_CNTL                      ; set low 8-bit pre-value
        LDA     #$0F
        STA     P_TMB_CNTH                      ; set high 4-bit pre-value
        LDA     #C_TMBES_EN + #C_TMBCLK_TMACRR
        STA     P_TMB_CTRL                      ;Set clock source for TMA_Carrier
```

5.11. IR Transfer/Receiver Module

RXTX is an analog block of GPM6C1015A, which can drive LED by TX, and can translate the IR LED sense current to digital signal. RX_SEN register can control this block. User can adjust PWM output driving capability by setting value of PWMDRV0, and adjusting the sensitivity of Rx block by SENSE [1:0]. Meanwhile, by setting the value of TACAPS to '1', capture signal can be input from PB3 pin.

TMAPWM signal (as showed in Figure 5-46) controls LED driver MOS. In PWM mode, Timer A can generate PWM signal, and the PWM duty, frequency, on/off switch can be accuracy controlled by timer A. The Envelope PWM signal can be generated by timer A & Timer B. And it has been illustrated in timer instruction. RX block translates sense current to digital signal RXOUT, and RXOUT is sent to Timer A block, which can get the carrier frequency in capture mode.

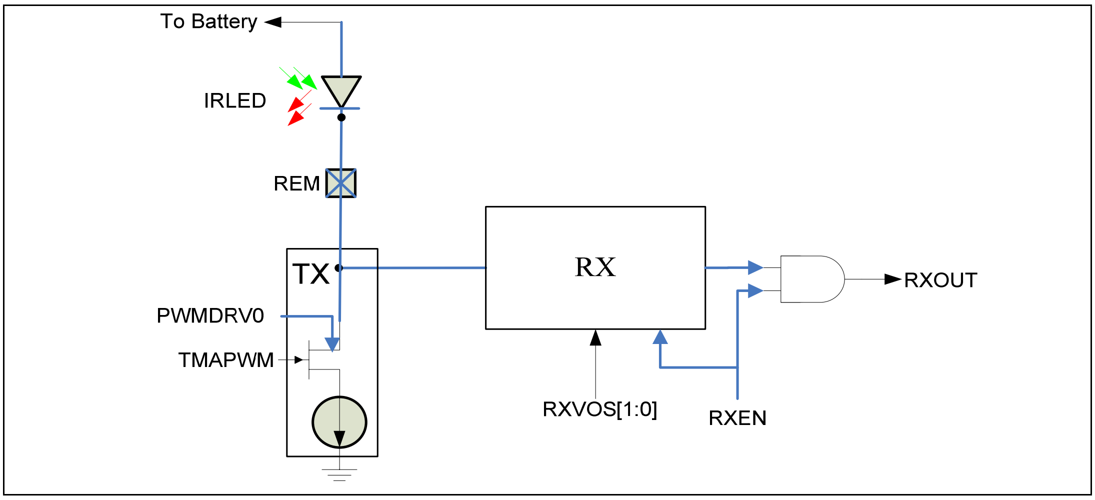


Figure 5-46 RXTX module diagram

Timer A PWM Drive Register (P_PWM_DRV, $11) (W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	PWMDRVS0	-	-	-
Access	-	-	-	-	W	-	-	-
Default	-	-	-	-	0	-	-	-

Bit [7:4] Reserved

Bit [3] **PWMDRV0 :** PWM driving current selected bits.
0 : PWM 1/2 driving current (C_PWMDRV_1)
1 : PWM 2/2 driving current (C_PWMDRV_2)

Bit [2:0] Please refer to P_RX_SEN register.

Timer A Sense Control Register (P_RX_SEN, $11) (W)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	-	TMACAPS	SENSES1	SENSES0
Access	-	-	-	-	-	W	W	W
Default	-	-	-	-	-	0	0	0

Bit [7:4] Reserved

Bit [3] Please refer to P_PWM_DRV register.

Bit [2] **TMACAPS:** Timer A capture input selected bit.
0 : REM PAD (C_RX_CAP)
1 : IOB3 (C_RX_IOB3)

Bit [1:0] **SENSES[1:0]:** RX SENSE selected bits.

00 → 01 → 10 → 11 sensitivity MAX → Min

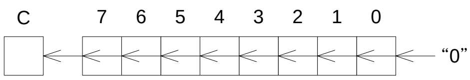
00 : RX SENSE Level 1 (C_RX_SENSE_1), sense current >=2uA

01 : RX SENSE Level 2 (C_RX_SENSE_2), sense current >=5uA

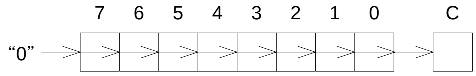
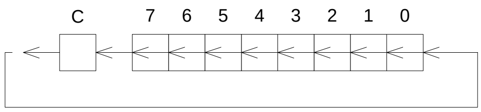
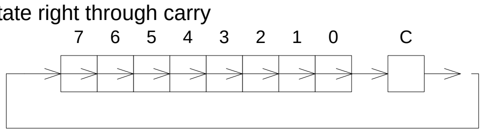
10 : RX SENSE Level 3 (C_RX_SENSE_3), sense current >=8uA

11 : RX SENSE Level 4 (C_RX_SENSE_4), sense current >=11uA

5.12. Alphabetical List of Instruction Set

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
1.	ADC #dd	69	2	2	Add to accumulator with carry.	
2.	ADC aa	65	2	3	A ← (A) + (M) + C	
3.	ADC aa, X	75	2	4		
4.	ADC aaaa	6D	3	4	If D-flag set to 1, the ADC performs decimal operation.	
5.	ADC aaaa,X	7D	3	4(A)		NV--D-ZC
6.	ADC aaaa,Y	79	3	4(A)		
7.	ADC (aa,X)	61	2	6		
8.	ADC (aa), Y	71	2	5(A)		
9.	AND #dd	29	2	2	And memory data with accumulator.	
10.	AND aa	25	2	3	A ← (A) ^ (M)	
11.	AND aa, X	35	2	4		
12.	AND aaaa	2D	3	4		N-----Z-
13.	AND aaaa,X	3D	3	4(A)		
14.	AND aaaa,Y	39	3	4(A)		
15.	AND (aa,X)	21	2	6		
16.	AND (aa), Y	31	2	5(A)		
17.	ASL A	0A	1	2	Arithmetic Shift Left	
18.	ASL aa	06	2	5		
19.	ASL aa,X	16	2	6		N-----ZC
20.	ASL aaaa	0E	3	6		
21.	ASL aaaa,X	1E	3	6(A)		
22.	BCC aa	90	2	2(C)	Branch if carry bit clear If (C) = 0, then pc ← (pc) + ??	--------
23.	BCS aa	B0	2	2(C)	Branch if carry bit set If (C) = 1, then pc ← (pc) + ??	--------
24.	BEQ aa	F0	2	2(C)	Branch if equal If (Z) = 1, then pc ← (pc) + ??	--------
25.	BIT aa	24	2	3	Test bit in memory with accumulator	NV----Z-
26.	BIT aaaa	2C	3	4	Z ← (A) ^ (M), N ← (M_7), V ← (M_6)	
27.	BMI aa	30	2	2(C)	Branch if minus If (N) = 1, then pc ← (pc) + ??	--------
28.	BNE aa	D0	2	2(C)	Branch if not equal If (Z) = 0, then pc ← (pc) + ??	--------
29.	BPL aa	10	2	2(C)	Branch if plus If (N) = 0, then pc ← (pc) + ??	--------
30.	BRK	00	1	7	Software interrupt If (B) = 1, then pc ← (pc) + 1	---B-I--
31.	BVC aa	50	2	2(C)	Branch if overflow bit clear If (V) = 0, then pc ← (pc) + ??	--------
32.	BVS aa	70	2	2(C)	Branch if overflow bit set If (V) = 1, then pc ← (pc) + ??	--------
33.	CLC	18	1	2	Clear C-flag：C ← "0"	-------0
34.	CLD	D8	1	2	Clear D-flag：D ← "0"	----0---

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
35.	CLI	58	1	2	Clear I-flag: I ← "0"	-----0--
36.	CLV	B8	1	2	Clear V-flag: V ← "0"	-0------
37.	CMP #dd	C9	2	2	Compare memory data with accumulator, (A) – (M)	N-----ZC
38.	CMP aa	C5	2	3		
39.	CMP aa, X	D5	2	4		
40.	CMP aaaa	CD	3	4		
41.	CMP aaaa,X	DD	3	4(A)		
42.	CMP aaaa,Y	D9	3	4(A)		
43.	CMP (aa,X)	C1	2	6		
44.	CMP (aa), Y	D1	2	5(A)		
45.	CPX #dd	E0	2	2	Compare memory data with X-register, (X) – (M)	N-----ZC
46.	CPX aa	E4	2	3		
47.	CPX aaaa	EC	3	4		
48.	CPY #dd	C0	2	2	Compare memory data with Y-register, (Y) – (M)	N-----ZC
49.	CPY aa	C4	2	3		
50.	CPY aaaa	CC	3	4		
51.	DEC aa	C6	2	5	Decrement M ← (M) - 1	N-----Z-
52.	DEC aa, X	D6	2	6		
53.	DEC aaaa	CE	3	6		
54.	DEC aaaa,X	DE	3	7		
55.	DEX	CA	1	2		
56.	DEY	88	1	2		
57.	EOR #dd	49	2	2	Exclusive OR A ← (A) ⊕ (M)	N-----Z-
58.	EOR aa	45	2	3		
59.	EOR aa, X	55	2	4		
60.	EOR aaaa	4D	3	4		
61.	EOR aaaa,X	5D	3	4(A)		
62.	EOR aaaa,Y	59	3	4(A)		
63.	EOR (aa,X)	41	2	6		
64.	EOR (aa), Y	51	2	5(A)		
65.	INC aa	E6	2	5	Increment M ← (M) + 1	N-----Z-
66.	INC aa, X	F6	2	6		
67.	INC aaaa	EE	3	6		
68.	INC aaaa,X	FE	3	7		
69.	INX	E8	1	2	X ← X + 1	N-----Z-
70.	INY	C8	1	2	Y ← Y + 1	N-----Z-
71.	JMP aaaa	4C	3	3	Unconditional jump Pc ← jump address	--------
72.	JMP (aaaa)	6C	3	6		
73.	JSR aaaa	20	3	6	Jump to subroutine (sp) ← (pc_H), sp ← sp – 1, (sp) ← (pc_L), sp ← sp – 1, pc ← aaaa	--------
74.	LDA #dd	A9	2	2	Load accumulator A ← (M)	N-----Z-
75.	LDA aa	A5	2	3		
76.	LDA aa, X	B5	2	4		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
77.	LDA aaaa	AD	3	4		
78.	LDA aaaa,X	BD	3	4(A)		
79.	LDA aaaa,Y	B9	3	4(A)		
80.	LDA (aa,X)	A1	2	6		
81.	LDA (aa), Y	B1	2	5(A)		
82.	LDX #dd	A2	2	2	Load X-register X ← (M)	N-----Z-
83.	LDX aa	A6	2	3		
84.	LDX aa, Y	B6	2	4		
85.	LDX aaaa	AE	3	4		
86.	LDX aaaa,Y	BE	3	4(A)		
87.	LDY #dd	A0	2	2	Load Y-register Y ← (M)	N-----Z-
88.	LDY aa	A4	2	3		
89.	LDY aa, X	B4	2	4		
90.	LDY aaaa	AC	3	4		
91.	LDY aaaa,X	BC	3	4(A)		
92.	LSR A	4A	1	2	Logical shift right 	N-----ZC
93.	LSR aa	46	2	5		
94.	LSR aa, X	56	2	6		
95.	LSR aaaa	4E	3	6		
96.	LSR aaaa,X	5E	3	6(A)		
97.	NOP	EA	1	2	No operation	--------
98.	ORA #dd	09	2	2	Logical OR A ← (A) v (M)	N-----Z-
99.	ORA aa	05	2	3		
100.	ORA aa, X	15	2	4		
101.	ORA aaaa	0D	3	4		
102.	ORA aaaa,X	1D	3	4(A)		
103.	ORA aaaa,Y	19	3	4(A)		
104.	ORA (aa,X)	01	2	6		
105.	ORA (aa), Y	11	2	5(A)		
106.	PHA	48	1	3	(sp) ← A, sp ← sp - 1	--------
107.	PHP	08	1	3	(sp) ← P status, sp ← sp -1	
108.	PLA	68	1	4	sp ← sp +1, A ← (sp)	--------
109.	PLP	28	1	4	Sp ← sp +1, P status ← (sp)	restored
110.	ROL A	2A	1	2	Rotate left through carry 	N-----ZC
111.	ROL aa	26	2	5		
112.	ROL aa, X	36	2	6		
113.	ROL aaaa	2E	3	6		
114.	ROL aaaa,X	3E	3	6(A)		
115.	ROR A	6A	1	2	Rotate right through carry 	N-----ZC
116.	ROR aa	66	2	5		
117.	ROR aa, X	76	2	6		
118.	ROR aaaa	6E	3	6		
119.	ROR aaaa,X	7E	3	6(A)		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NV-BDIZC
120.	RTI	40	1	6	Return from interrupt Sp ← sp + 1, P status ← (sp), sp ← sp + 1, pc_L← (sp), sp ← sp +1, pc_H ← (sp)	restored
121.	RTS	60	1	6	Return from subroutine Sp ← sp + 1, pc_L← (sp), sp ← sp +1, pc_H ← (sp)	--------
122.	SBC #dd	E9	2	2	Subtract with carry A ← (A) – (M) - ~(C)	NV----ZC
123.	SBC aa	E5	2	3		
124.	SBC aa, X	F5	2	4		
125.	SBC aaaa	ED	3	4		
126.	SBC aaaa,X	FD	3	4(A)		
127.	SBC aaaa,Y	F9	3	4(A)		
128.	SBC (aa,X)	E1	2	6		
129.	SBC (aa), Y	F1	2	5(A)		
130.	SEC	38	1	2	Set C-flag: C ← “1”	-------1
131.	SED	F8	1	2	Set D-flag: D ← “1”	----1---
132.	SEI	78	1	2	Set I-flag: I ← “1”	-----1--
133.	STA aa	85	2	3	Store accumulator in memory (M) ← A	--------
134.	STA aa, X	95	2	4		
135.	STA aaaa	8D	3	4		
136.	STA aaaa,X	9D	3	5		
137.	STA aaaa,Y	99	3	5		
138.	STA (aa,X)	81	2	6		
139.	STA (aa), Y	91	2	6		
140.	STX aa	86	2	3	Store X-register in memory (M) ← X	--------
141.	STX aa, Y	96	2	4		
142.	STX aaaa	8E	3	4		
143.	STY aa	84	2	3	Store Y-register in memory (M) ← Y	--------
144.	STY aa, X	94	2	4		
145.	STY aaaa	8C	3	4		
146.	TAX	AA	1	2	Transfer accumulator to X-register: X ← A	N-----Z-
147.	TAY	A8	1	2	Transfer accumulator to Y-register: Y ← A	N-----Z-
148.	TSX	BA	1	2	Transfer sp to X-register: X ← sp	N-----Z-
149.	TXA	8A	1	2	Transfer X-register to accumulator: A ← X	N-----Z-
150.	TXS	9A	1	2	Transfer X-register to sp: sp ← X	N-----Z-
151.	TYA	98	1	2	Transfer Y-register to accumulator: A ← Y	N-----Z-

Notes:

1. Cycle (A): Cycle+1 when cross a boundary.
2. Cycle(C): Cycle+1 if the branch condition is true; Cycle+2 if the branch condition is true and cross a boundary.

6. ELECTRICAL CHARACTERISTICS

6.1. Absolute Maximum Ratings

Characteristics	Symbol	Ratings
DC Supply Voltage	V_+	< 5.0V
Input Voltage Range	V_{IN}	-0.5V to V_+ + 0.5V
Operating Temperature	T_A	0℃ to +70℃
Storage Temperature	T_{STO}	-50℃ to +150℃
Average PWM MAX Driving Current	I_{REMM}	150mA
VDD Total MAX Current	I_{VDDM}	100mA
VSS Total MAX Current	I_{VSSM}	120mA

Note: Stresses beyond those given in the Absolute Maximum Rating table may cause permanent damage to the device. For normal operational conditions, see AC/DC Electrical Characteristics.

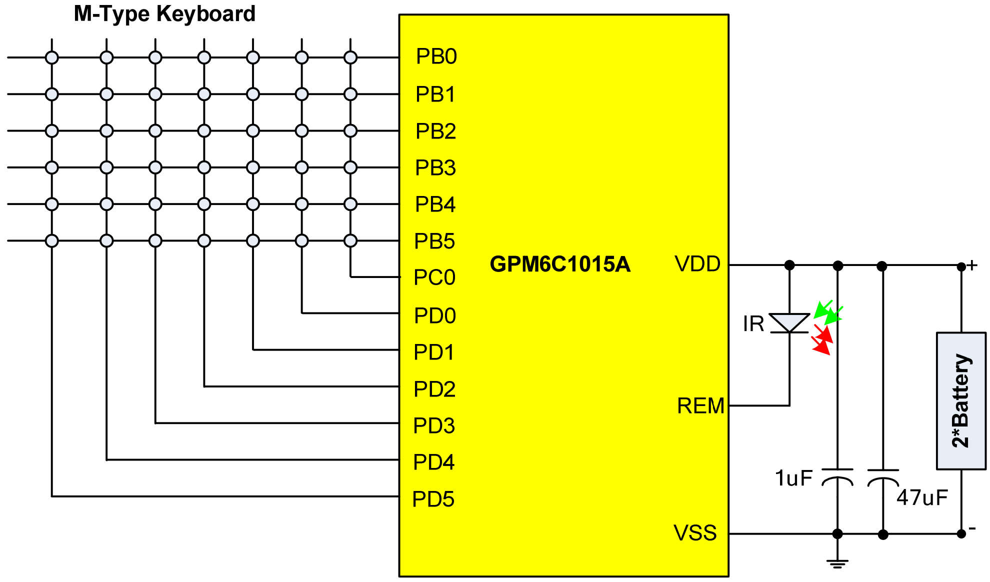
6.2. AC Characteristics (T_A = 25℃)

Characteristics	Limit			Unit	Test Condition
	Min.	Typ.	Max.		
OSC Accuracy @ Freq=4MHz					
OSC Variation	-3.0	±1.5	3.0	%	VDD = 2.0V - 3.6V, T_A =25°C

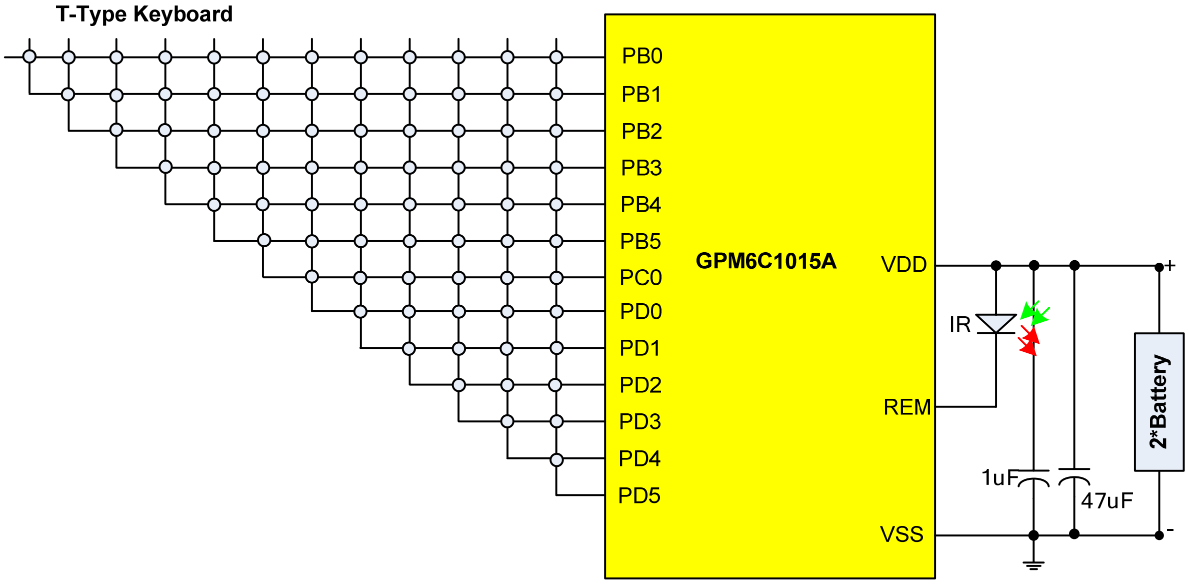
6.3. DC Characteristics (VDD = 3.0V, T_A = 25℃)

Characteristics	Symbol	Limit			Unit	Test Condition
		Min.	Typ.	Max.		
Operating Voltage	VDD	2.0	-	3.6	V	F_{CPU} = 8.0MHz, For 2-battery
Operating Current	I_{OP}	-	2.0	4.0	mA	F_{CPU} = 8.0MHz @ 3.6V, no load
M-Type key Standby Current	I_{MSTBY}	-	-	1.0	uA	VDD = 3.6V
T-Type key Standby Current	I_{tstby}	-	-	2.0	uA	VDD = 3.6V, no load
Input High Level	V_{IH}	0.7VDD	-	-	V	VDD = 3.0V
Input Low Level	V_{IL}	-	-	0.3VDD	V	VDD = 3.0V
Output High Level PB, PC, PD	V_{OH}	0.8VDD	-	-	V	VDD = 3.0V I_{OH} = -6mA
Output Low Level PB, PC, PD	V_{OL}	-	-	0.2VDD	V	VDD = 3.0V I_{OL} = 16mA
Input Pull High Resistor PA, PB, PC, PD	R_H	30	50	70	Kohm	Pull High VDD = 3.0V
Input Pull Low Resistor PA, PB, PC, PD	R_L	30	50	70	Kohm	Pull Low VDD = 3.0V
Max PWM Driving Current	I_{PWM}	200	-	-	mA	VDD = 3.0V,V_{REM} =3.0V PWMDRV0=1
LVR Active Voltage (by option)	V_{LVR}	1.7	1.85	2.0	V	

7. APPLICATION CIRCUITS



Note: Capacitor must be close to chip. (no more than 1 cm)



Note: Capacitor must be close to chip. (no more than 1 cm)

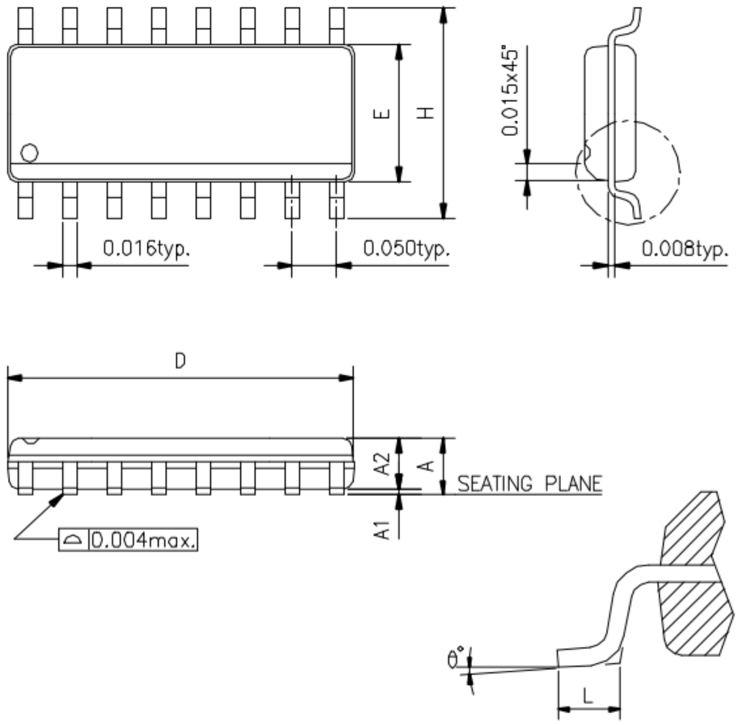
8. PACKAGE/PAD LOCATIONS

8.1. Ordering Information

Product Number	Package Type
GPM6C1015A – HS03x	Green Package

8.2. Package Information

SOP16



SYMBOLS	MIN.	MAX.
A	0.053	0.069
A1	0.004	0.010
D	0.386	0.394
E	0.150	0.157
H	0.228	0.244
L	0.016	0.050
θ°	0	8

UNIT : INCH

NOTES:
1. JEDEC OUTLINE : MS-012 AC
2. DIMENSIONS "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED .15mm (.006in) PER SIDE.
3. DIMENSIONS "E" DOES NOT INCLUDE INTER-LEAD FLASH, OR PROTRUSIONS. INTER-LEAD FLASH AND PROTRUSIONS SHALL NOT EXCEED .25mm (.010in) PER SIDE.

9. DISCLAIMER

The information appearing in this publication is believed to be accurate.

Integrated circuits sold by Generalplus Technology are covered by the warranty and patent indemnification provisions stipulated in the terms of sale only. GENERALPLUS makes no warranty, express, statutory implied or by description regarding the information in this publication or regarding the freedom of the described chip(s) from patent infringement. FURTHERMORE, GENERALPLUS MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE. GENERALPLUS reserves the right to halt production or alter the specifications and prices at any time without notice. Accordingly, the reader is cautioned to verify that the data sheets and other information in this publication are current before placing orders. Products described herein are intended for use in normal commercial applications. Applications involving unusual environmental or reliability requirements, e.g. military equipment or medical life support equipment, are specifically not recommended without additional processing by GENERALPLUS for such applications. Please note that application circuits illustrated in this document are for reference purposes only.

10. REVISION HISTORY

Date	Revision #	Description	Page
Nov 21, 2013	0.1	Preliminary version	63