

TM57 Series

TM57PA10(A) DEMO CODE FOR TM57PA10(A) BASIC FUNCTIONS SAMPLE

Application Note

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. tenx does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. tenx products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

AMENDMENT HISTORY

Version	Date	Description
V1.0	May, 2011	New release

CONTENTS

AMENDMENT HISTORY2

PRODUCT NAME4

TITLE4

APPLICATION NOTE4

1. TM57PA10_ADC Application Sample5

2. TM57PA10_PWMA Application Sample.....6

3. TM57PA10_PWMB Application Sample.....7

4. TM57PA10_timer0 Application Sample8

5. TM57PA10_Timer1 Application Sample9

6. TM57PA10_wakeup Application Sample10

7. TM57PA10_WDT Application Sample11

PRODUCT NAME

TM57 series IC

TITLE

TM57PA10_ADC Application Sample

TM57PA10_pwmA Application Sample

TM57PA10_pwmB Application Sample

TM57PA10_timer0 Application Sample

TM57PA10_timer1 Application Sample

TM57PA10_wakeup Application Sample

TM57PA10_WDT Application Sample

APPLICATION NOTE

EV2793 supports TM57PA10 and TM57PE11 ICs. In using TM57PE11 simulation, please avoid using TM57PA10 unique resources; it can still be used in EV board, however, these resources don't exist in the real IC.

The difference between TM57PA10 and TM57PA10A is in FIRC (Fast Internal RC clock), the FIRC in TM57PA10 is only 4 MHz, and however there are two types of FIRC selection in TM57PA10A, i.e. 4 MHz and 12 MHz, the default is 4 MHz.

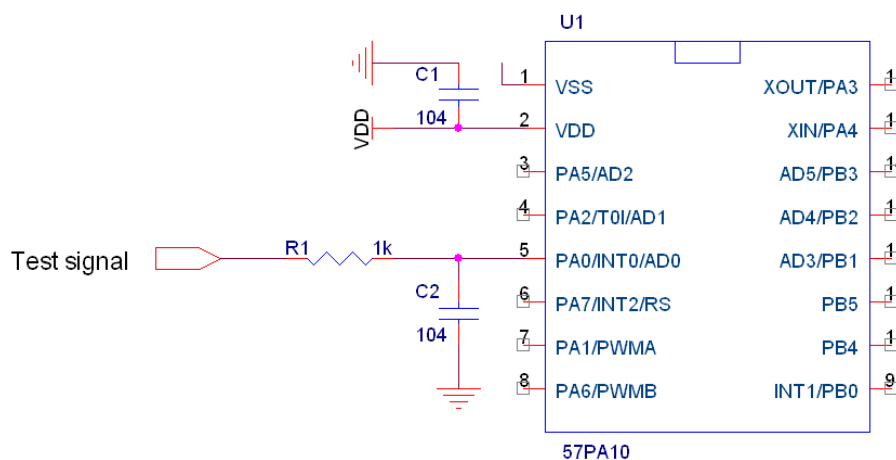
1. TM57PA10_ADC Application Sample

1-1. Details of the DEMO program, please refer to TM57PA10_ADC.asm.

1-2. Sample Description

- 1) 12-bit ADC (Analog Digital Converter) of TM57PA10 consists of a 6-channel input analog multiplexer, register controller, clock generator, 12-bit successive approaching ADC and output data register. When using ADC, please be noted to connect ADC port with a resistance, and connect a 104 capacitor to GND, to protect AD port and guarantee ADC conversion precision.
- 2) AD conversion step:
 1. Set ADC port as input port, disable pull high, set I/O port data register to 1.
 2. Select ADC conversion channel (select channel 0 in procedure, F-plane 11h.2~0).
 3. Set I/O needed to ADC type (in setting I/O port to AD type, please do not set unused I/O port to ADC type, R-plane 12h.5~0 , AD0~AD5).
 4. Set ADC conversion clock rate (the max ADC conversion clock rate must be not more than 2 MHz, in the DEMO program, it is using 128 frequency divisions (when user selecting conversion clock rate, it is suggested to select the slowest conversion rate which is acceptable for the application, it is advantageous to improve AD conversion precision. It takes 50 AD clock period for one AD conversion, R-plane oc.6~4).
 5. Switch on ADC conversion (ADCSTART is set to 1, F-plane 11h.3).
 6. Wait for ADC conversion finishes (When ADC conversion is finish, ADCSTART will be switched automatically to 0. When ADCSTART=0 is detected, means that ADC conversion is finish, F-plane 11h.3).
 7. Save the result (every time ADC conversion is finish, it is necessary to save the result to the proper location, otherwise, ADC data register will be overwritten on the next conversion, F-plane 10h.7~0, 11h.7~4).

1-3.Circuit Diagram



2. TM57PA10_PWMA Application Sample

2-1. Details of DEMO program, please refer to TM57PA10_PWMA.asm.

2-2. Sample description

- 1) PWMA is 8-bit period non-adjustable PWM generator, the frequency sources must be from FOSC, and the period time is 256 clocks period. The PWM output signal resets to low level whenever the 8-bit base counter matches the 8-bit MSB of PWM duty register (PWMDUTY). When the base counter rolls over, the 2-bit LSB of PWM duty register decides whether to set the PWM output signal high immediately or set it high after one clock cycle delay. Based on 4 PWM period, when the LSB 2-bit value of PWM duty register is 0, 1, 2, 3, the 4 PWM period high level will increase 0, 1, 2, 3 clock period, respectively; therefore, the PWM duty resolution can be expanded to 1024 clocks period.
- 2) PWMA output configuration steps:
 1. Set PA1 port as input, and disable pull high.
 2. Set PWMA duty value (set the duty value of 0-7 bits in PWMADUTYH F-plane 0ch.7~0 and 6-7 bits in PWMADUTYL F-plane 0dh.7~6).
 3. Set PWMA to be output from PA1 (PWMAE =1, at this moment, PEM outputs from PA1, therefore, if PWMA is to be closed, then, set PWMAE=0, R-plane 0bh.6).
- 3) The formula for the Period is as follows:

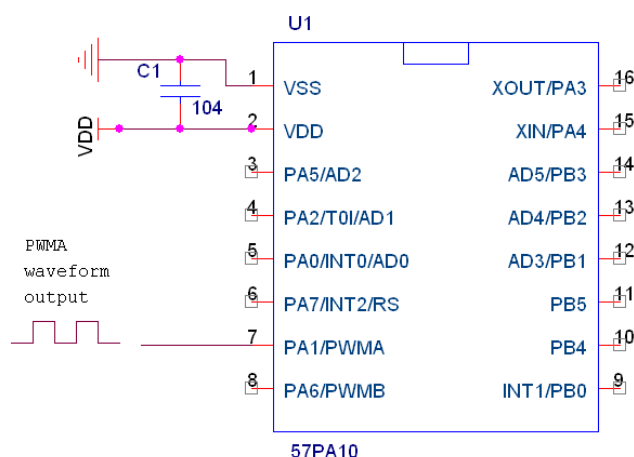
$$1/F_{osc} * 256 = \text{PWM period}$$

Using procedure setting value as sample: $1/4M * 256 = 0.25 \mu s * 256 = 64 \mu s$
- 4) The formula for the Duty is as follows:

$$\text{PWMADUTYH}/256 + (\text{PWMADUTYL})/1024 = \text{duty}$$

Example: PWMADUTYH=128, PWMADUTYL=2, therefore
Duty=128/256+2/1024=514/1024

2-3. Circuit Diagram



3. TM57PA10_PWMB Application Sample

3-1. Details of DEMO program, please refer to TM57PA10_WMB.asm

3-2. Sample description

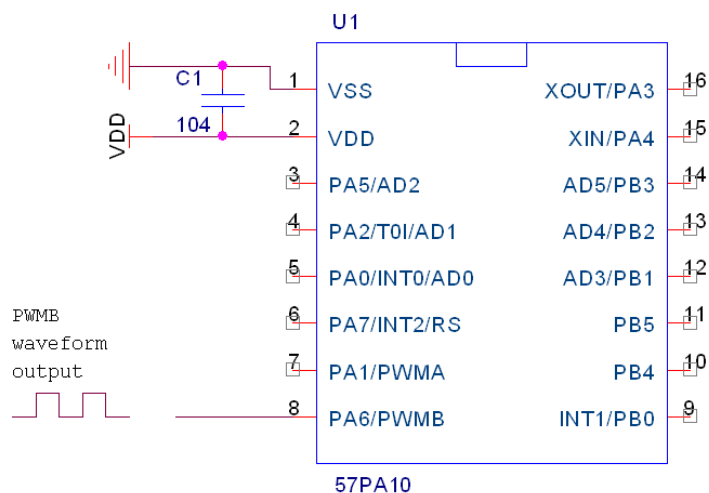
- 1) PWMB and PWMA theoretically is the same, the period time in 8-bit non-adjustable PWM generator is 256 clock periods, which frequency source must be FOSC.
- 2) PWMB output configuration steps:
 1. Set PA6 port as input, and disable pull high.
 2. Set PWMB duty value (set the duty value of 0-7 bits in PWMBDUTYH F-plane 0eh.7~0 and 6-7 bits in PWMBDUTYL F-plane 0fh.7~6).
 3. Set PWMB to be output from PA6 (PWMBE =1, at this moment, PEM outputs from PA6, therefore, if PWMB is to be closed, then, set PWMBE=0, R-plane 0bh.5).
- 3) The formula for the Period is as follows:

$$1/F_{osc} * 256 = \text{PWM period}$$

Using procedure setting value as sample: $1/4M * 256 = 0.25 \mu s * 256 = 64 \mu s$
- 4) The formula for the Duty is as follows:

$$\text{PWMA_DUTY_H}/256 + (\text{PWMA_DUTY_L})/1024 = \text{duty}$$

3-3. Circuit Diagram



4. TM57PA10_timer0 Application Sample

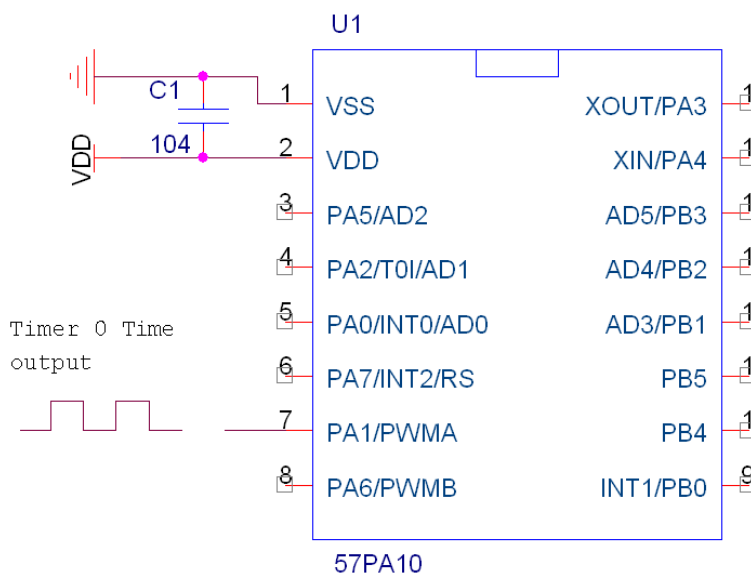
4-1. Details of DEMO program, please refer to TM57PA10_timer0.asm.

4-2. Sample description

- 1) TIMER0 is an 8-bit non-auto reload periodic counter, every time interrupt occurs, TIMER0 initial value must be reconfigured. The counter clock source selection is SELT0I; SELT0I=0, means the clock source is fosc/2, while SELT0I =1 means the clock source is input from T0I pin.
- 2) Configuration steps:
 1. Set TIMER0 initial value (F-plane 01h.7~0)
 2. Set Timer0 frequency division (bit0-3 in TM0PSC, R-plane 02h.3~0)
 3. Set Timer clock source (SELT0I, R-plane 02h.4)
 4. Enable counter interrupt (when TM0IE = 1, the timer starts countdown to interrupt, F-plane 08h.4)
- 3) Counter time calculation:

$$1/fosc * 2 * \text{Timer0 Pre-Scale} * (256 - \text{TIMER0}) = \text{time}$$
 or
$$1/Ft0i * \text{Timer0 Pre-Scale} * (256 - \text{TIMER0}) = \text{time}$$
 (Ft0i is input frequency for T0I pin)
 Using this sample in DEMO program, the configuration value is: $1/4M * 2 * 8 * (256 - 131) = 500 \mu s$.

4-3. Circuit Diagram



5. TM57PA10_Timer1 Application Sample

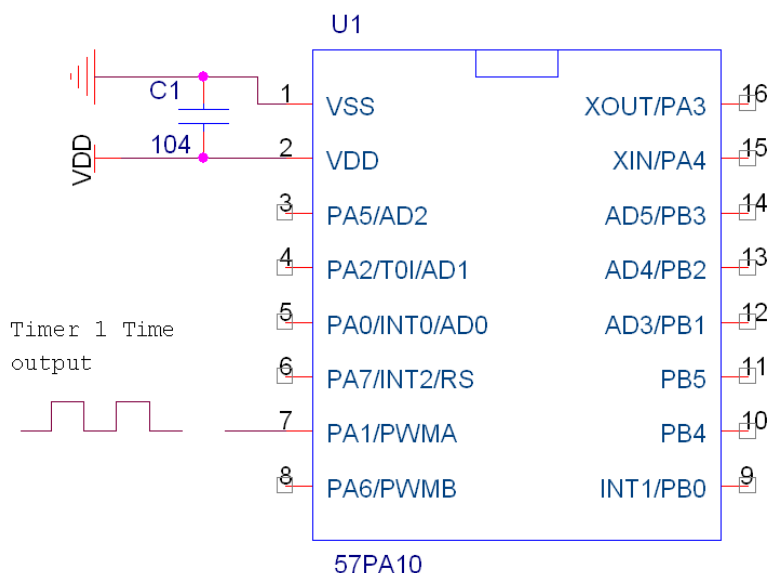
5-1. Details of DEMO program, please refer to TM57PA10_timer1.asm

5-2. Sample description

- 1) TIMER1 is in F-Plane of the 8-bit counter. Same as other F-Plane registers, it can also read and write. Besides, in clock division rate set before, TIMER1 periodically increases and auto loads a new "Counter value" (TM1RELD). TIMER1 counter value increasing is decided by R-Plane "Timer1 Pre-Scale" (TM1PSC) register. After TIMER1 overflows, TIMER1 will generate interrupt trigger signal (TM1I).
- 2) TIMER1 configuration steps:
 1. Set TIMER1 initial value, and set auto-reload register value (F-plane 0ah.7~0, R-plane 0dh 7~0)
 2. Set TIMER1 frequency division (bit0-3 in TM0PSC, R-plane 0ch.3~0)
 3. Switch on the timer interrupt TM1IE=1, the timer will start countdown for the interrupt (F-plane 08h.5)
- 3) TIMER1 clock source must be from FOSC/2, TIMER1 time calculation is as follows:

$$1/fosc * 2 * \text{Timer1 Pre-Scale} * (256 - \text{TIMER0}) = \text{time}$$
 Using this sample in DEMO program, the configuration value is: $1/4M * 2 * 8(256 - 131) = 500 \mu s$

5-3. Circuit Diagram



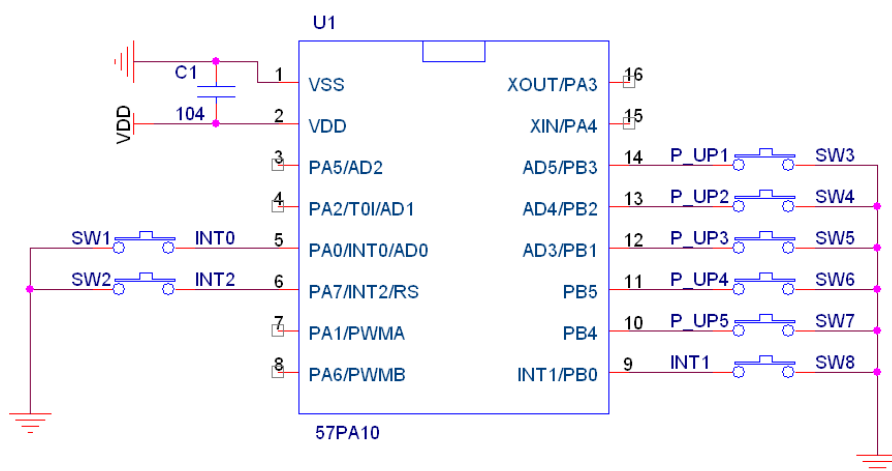
6. TM57PA10_wakeup Application Sample

6-1. Details of DEMO program, please refer to TM57PA10_wakeup.asm

6-2. Sample description

- 1) After the subroutine executes Sleep command to enter sleep mode, TM57PA10 can use several ways to wake up. Such as the interrupt, i.e. via 3 external interrupt ports, PB1-PB5 I/O port level variation and watch dog timer overflow.
- 2) External interrupt wakeup configuration steps:
 1. At first, set INT0(PA0), INT1(PB0), INT2(PA7) ports as inputs and enable pull high, set the I/O port data register to 1.
 2. Enable the related external interrupt. (F-plane 08h.2~0)
 3. When entering sleep mode, interrupt port level from high to low condition will cause interrupt, at this moment, subroutine will enter interrupt, and sleep mode will be waken up.
- 3) Using PB port to wake up:
 1. Set PB1-PB5 ports as input, and enable pull high, set the port data register to 1.
 2. Enable I/O port wake up function (i.e. set PBWKUP to 1, R-plane 0bh.2)
 3. After enters sleep mode, PB1-PB5 port level from high to low will wake up the sleep mode, subroutine will continue operating.
- 4) WKT wakeup function steps:
 1. At first, from the config selection, select Disable WDT Reset, Enable WKT timer.
 2. Select Watch Dog timer overflow time (0-1 bit of WKTPSC, R-plane 0bh.1~0)
 3. Enable watch dog interrupt (when watch dog timer overflows, subroutine will be interrupted, sleep mode will be waken up, F-plane 08.3)

6-3. Circuit Diagram



57PA10 External
interrupt and port of
wakeup

Note: Before IC enters sleep mode, please make sure I/O with wake up function cannot be set as FLOATING (input and without pull high resistance), otherwise, I/O FLOATING status will affect electric current consumption and IC may not enter SLEEP status because of the unstable I/O condition.

7. TM57PA10_ WDT Application Sample

7-1. Details of DEMO program, please refer to TM57PA10_ WDT.asm

Watchdog circuit consists of an internal independent RC oscillator circuit and a timer. Regarding to power save reason, this independent RC oscillator circuit can be switched OFF, as shown as below table, and once the oscillator is in suspend condition, timer overflow will not happen, it means, the further reset (WDT) or interrupt (WKT) will not happen.

Mode	WDTE	WKTIE	Watch Dog Circuit
Operating	0	0	Stop
	0	1	Operating
	1	0	
	1	1	
Sleep	0	0	Stop
	0	1	Operating
	1	0	Stop
	1	1	Operating

Note: WDTE is the bit6 of CONFIG

WKTIE is enable bit of WKT interrupt (f_plane, 08h.3)

7-2. Sample Description

- 1) This DEMO program is mainly to show how to use WDT in SLEEP mode to finish RESET function, the key point of the program is before entering SLEEP mode, set WDTE, WKTIE to '1', to make sure when enabling watch dog circuit, the oscillator circuit in SLEEP mode also keeps operating, so that when watch dog timer overflows, reset will occur.
- 2) Please note that WKTIE needs to be set to 0 if user hopes to spend the least power consumption in SLEEP mode when the watch dog oscillator circuit is closed.