GETTING STARTED WITH THE TMS320F24x PROCESSOR

Tutorial 9: Using the General Purpose Timers

Introduction

In a DSP based system, there is often the need to carry out actions at specific times. A typical requirement is to measure an analog input at regular intervals and generate a corresponding output. For these applications, timers can be used.

Generating Time Delays

In Tutorial 3 we saw to use time delays to control the amount of time the XF LED is on and the amount of time the XF LED is off.

Example 9-1.

```
        |                       |
start:  | SETC XF               | ; Turn on XF LED.
        | LACC #7FFFh           | ; Load accumulator with 7FFFh.
loop1:  | SUB #1                | ; Decrement accumulator.
        | BCND loop1, NEQ       | ; Test if accumulator is zero.
        | CLRC XF               | ; First loop done. Turn off XF
        |                       | ; LED.
        | LACC #7FFFh           | ; Load accumulator with 7FFFh.
loop2:  | SUB #1                | ; Decrement accumulator.
        | BCND loop2, NEQ       | ; Test if accumulator is zero.
        | B start               | ; Second loop done. Go round
        |                       | ; again.
```

There is a major problem with the code in Example 9-1. When the program is running, all resources are dedicated to turning the XF LED on and off. There is no time for any other processing!

What we need is some form of clock that runs independent of our program and that prompts us when it is time to change the state of the XF LED. For this purpose, the Event Manager (abbreviated to EV) is provided. The TMS320LF2407 has two Event Managers, referred to as EVA and EVB.

The Event Manager (EV)

The event manager provides a whole range of timer and counter related functions. For the moment, the two components we are most interested in are General Purpose Timer 1 and General Purpose Timer 2. These are can be used to provide pulse-width modulation (PWM) outputs and to control the ADCs.

A TMS320F24x timer behaves rather like a central heating timer. The clock of a central heating system is running continually 24 hours a day. If we decide to have the heating to come on and go off at pre-determined times, we simply set these times. The clock runs and when turn-on time is reached, the heating comes on. When the turn-off time is reached, the heating goes off. The important thing is that we do not have to monitor the time – it is all done automatically.

General Purpose Timer 1 (abbreviated to GP T1) and General Purpose Timer 2 (GP T2) work in a similar way. Once a timer is set, it runs automatically, controlling events, independent of the operation of our program.

Configuring General Purpose Timer 1

We shall configure General Purpose Timer 1 (GP T1) in what is known as "continuous up-counting mode". This means that the timer counts 0,1, 2, 3, 4, 5 etc. until it reaches a predetermined value when it continues counting again from zero.

General purpose Timer 1 (GP T1) can also be used to directly generate an output on input / output port pin IOPB4. This pin can be used to drive external devices and also allows us to monitor the operation of the timer.

Generating a 100 kHz Signal

In the Example 9-2, the timer continually produces an output of 100 kHz on port pin IOPB4 of the TMS320F243 DSK. This can be measured using an oscilloscope or a frequency meter.

Example 9-2.

```
        .setsect ".text",   8800h

start:  LDP #225            ; Page for OCRA
        SPLK #1001h, 10h    ; PWM output on IOPB4
loop:   LDP #0E8h           ; Page 232 for timers.
        SPLK #8142h, 4h     ; Internal clock, GP T1 off.
        SPLK #6ah, 0h       ; Timer compare enabled. Timer
                            ; enabled.
        SPLK #63h, 3h       ; 99 + 1 generates 100 kHz.
        SPLK #32h, 2h       ; 50% duty cycle.
        SPLK #0FFFEh, 1h    ; Timer on value before overflow
```

	SPLK #9142h, 4h	; GP T1 on.
loop:	B *loop*	; Loop forever.

The output frequency is derived from the crystal on the circuit board. If the TMS320LF2407 DSK were used, then the output frequency on IOPB4 would be different.

Rather than using "magic numbers", we can use the `.set` directive to use the names of registers instead. Labels must go in the first column.

Example 9-3.

	.setsect ".text",	8800h
DP_EV	.set 232	; Data page for Event Manager
DP_DGCR	.set 225	; Data page for digital ; control registers
OCRA	.set 10h	; Output Control register
GPTCON	.set 0h	; General-Purpose Timer ; Control Register
T1CNT	.set 1h	; GP Timer 1 Count Register
T1CMP	.set 2h	; GP Timer 1 Compare Register
T1PR	.set 3h	; GP Timer 1 Period Register
T1CON	.set 4h	; GP Timer 1 Control Register
start:	LDP #DP_DGCR	; Data Page for OCRA.
	SPLK #1000h, OCRA	; T1PWM output on pin IOPC4.
	LDP #DP_EV	; Data page for timers.
	SPLK #8142h, T1CON	; Internal clock, GP T1 off.
	SPLK #42h, GPTCON	; Active high output on ; IOPC4.
	SPLK #63h, T1PR	; 99 + 1 generates 100 kHz.
	SPLK #32h, T1CMP	; 50% duty cycle.
	SPLK #0FFFEh, T1CNT	; Timer count value before ; counter starts (-2).
	SPLK #9142h, T1CON	; GP T1 on.
loop:	B *loop*	; Loop forever.

Examples 9-2 and 9-3 show how to set up General Purpose Timer 1 (GPT1). We shall now look at the operation in more detail.

Configuring OCRA

Certain input / output pins of the TMS320F24x have dual functionality. For example, IOPB4 can be used as either general-purpose input / output pin or as a dedicated output pin for GP Timer 1. Control of this functionality is by setting the appropriate

values in I/O Multiplexed Control Register A (OCRA). The TMS320LF2407 databook uses the slightly different abbreviation of MCRA.

Table 9-1. Fields of OCRA Controlling ADC and Timers.

Bits	Port Pin	Primary Function	Secondary Function
Bit 13	IOPB5	1 = T2PWM	0 = GP I/O
Bit 12	IOPB4	1 = T1PWM	0 = GP I/O
Bit 1	IOPA1	1 = ADC Channel 1	0 = GP I/O
Bit 0	IOPA0	1 = ADC Channel 0	0 = GP I/O

At power-up, the TMS320F243 DSK has OCRA configured so that the PWM outputs T1PWM and T2PWM are turned off. To turn generate timer outputs on IOPB4 and IOPB5, we must set bits 12 and 13 of OCRA.

Configuring T1CON

T1CON determines how the timer operates and at what speed. The bits of the General Purpose Timer 1 Control Register T1CON we are interested in are:

Table 9-2. Fields of T1CON.

Bits	Values
Bits 15-14	10 = Free run emulation.
Bits 13-11	000 = Timer stopped / held. 010 = Count up.
Bits 10-8	Pre-scaler = 001. Use CPU clock frequency / 2.
Bit 7	Reserved. Not used by GP T1.
Bits 6	1 = Timer enabled (but not necessarily running).
Bits 5-4	00 = Internal clock.
Bits 3-2	00 = Timer compare (active) register reload when count = 0.
Bit 1	1 = Timer compare enable.
Bit 0	Reserved. Not used by GP T1.

The pre-scaler is a frequency divider, which reduces the clock speed of the timer. Setting the pre-scaler to “001” means that the CPU clock divided by 2 is used as the clock source. One clock period on the TMS320F243 DSK is 2 *x* 50 ns = 100 ns, corresponding to a clock frequency of 10 MHz.

If we were to set the pre-scaler to “111”, the clock period would be 128 *x* 50 ns = 6.4 μs.

Figure 9-1. Configuration of T1CON

Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

We shall use the value 8142h to configure T1CON. This means GP Timer 1 is stopped, internal clock with pre-scaler of 2, timer enabled and timer compare occurs when the count is zero.

Controlling the Output Period

The Timer 1 Period Register (T1PR) controls how many counts there are before the counter returns to zero. We can use a value between 1 and FFFFh. In Example 9-2 we have used a value of 99 for T1PR. Hence the counter will count 0, 1, 2, 3, … 97, 98, 99, 0, 1, 2 etc.

This means that there will be 99 + 1 = 100 counts.

Using the TMS320F243 DSK, each count takes 100 ns. Therefore, the total period is 100 *x* 100 ns = 10 µs. This corresponds to an output frequency of 100 kHz on IOPB4.

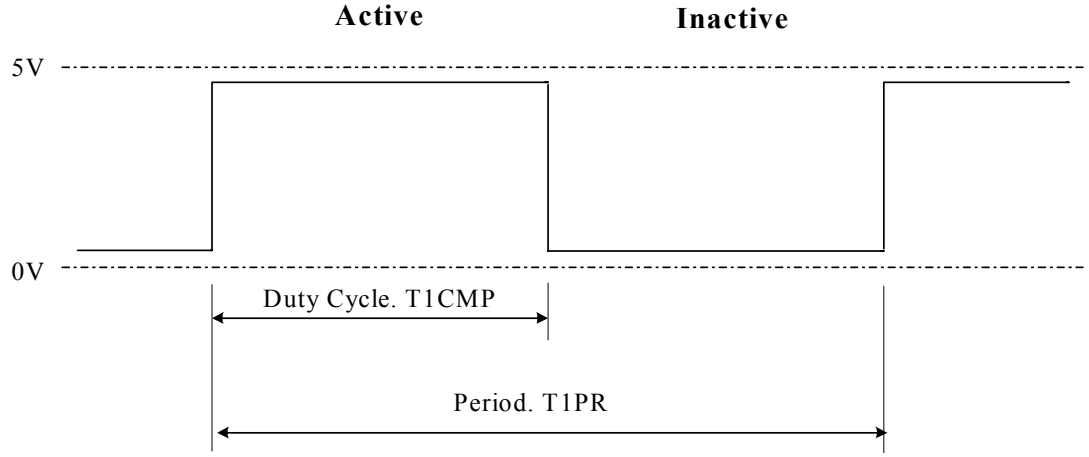
To generate an output of 1 MHz, we would use T1PR = 10 - 1 = 9.

Similarly, to generate an output of 10 kHz, we would use T1PR of 1000 - 1 = 999.

Setting the Output Duty Cycle

The Timer 1 Compare Register (T1CMP) determines the *duty-cycle*. The output is shown in diagrammatic form in Figure 9-1.

Figure 9-2. Duty Cycle



By *duty cycle* we mean the percentage of time the output is active. In Figure 9-1, the duty cycle is 50%. This means that the output is active for 50% of the time and inactive for 50% of the time. The *duty cycle* is controlled by T1CMP

The *period* is controlled by T1PR. By period we mean the time between repeating sequences.

Normally the output period T1PR would be fixed and the duty-cycle (determined from the value of T1CMP) is altered. This maintains the same output frequency at all times, but alters the amount of time the output is high or low.

When the counter reaches this pre-set value, the output pin IOPB4 is made to change state. The duty-cycle must be greater than zero and less than the period i.e. T1CMP must always be less than T1PR.

Action When Compare Occurs

When the count reaches the value in T1CMP, an action can be made to occur. For example, it is possible to start an analog-to-digital conversion. We can also set up the timers to control the output on pins IOPB4 and IOPB5.

The action when a compare occurs is determined by the General Purpose Control Register (GPTCON). The TMS320LF2407 uses the abbreviation GPTCONA instead.

Table 9-3. Fields of GPTCON.

Bits	Values
Bit 15	Reserved.
Bit 14	Timer 2 status. Read only. 1 = Counting upwards.
Bit 13	Timer 1 status. Read only. 1 = Counting upwards.
Bits 12-11	Reserved.
Bits 10-9	00 = No timer 2 event starts ADC
Bits 8-7	00 = No timer 1 event starts ADC
Bits 6	1 = Enable all GP Timer compare outputs
Bits 5-4	Reserved.
Bit 3-2	GP Timer 2 compare output.
Bits 1-0	GP Timer 1 compare output. 10 = Active high

We shall use the value 0042h, that is GP Timer compare outputs enabled and active high when a compare occurs.

Figure 9-3. Configuration of GPTCON.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

The important bit to configure is bit 6, which enables the output pins on all the timers.

Configuring T1CNT

As part of the initialization in Example 9-3 we have set the value of T1CNT to FFFEh (–2 decimal). Why have we chosen this value?

Suppose we are using T1PR = 100. This means that T1CNT takes values in the range 0 to 99. However, when the timer is being configured, T1CNT can take a value of anywhere between 0 and 65535.

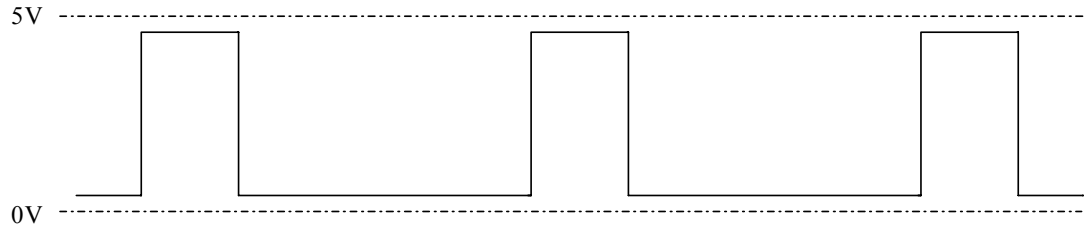
Suppose the timer starts with a value of 100. In order to get to zero, it must first count up to 65535, then overflow to zero. This will mean that the TMS320F243 system can take up to 32.76 µs before the outputs work as expected. Even worse, with the timer pre-scaler set to its maximum value, this time increases to somewhere in the region of 0.4 seconds.

If we configure T1CNT to FFFEh (-2 decimal) before we start running GP T1, then it is guaranteed to reach 0000 in a short time.

Altering the Duty Cycle

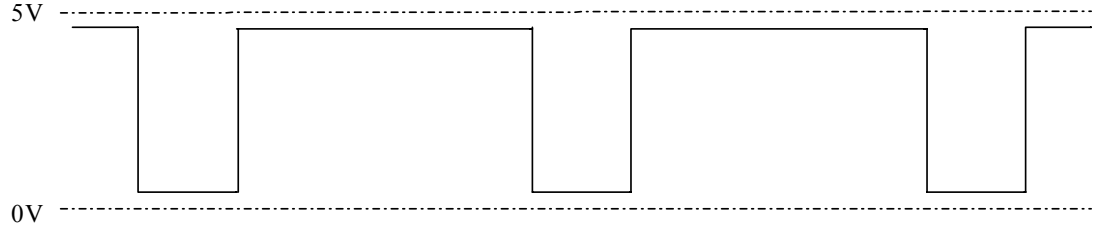
In Examples 9-2 and 9-3 we have used a fixed output period of 10 µs (100 kHz) and a duty cycle of 50%; that is, the output is active for 50% of the time and inactive for 50% of the time. We can use other duty cycles. Figure 9-4 shows a duty cycle of 25%, that is the output is active 25% of the time and inactive 75% of the time.

Figure 9-4. Duty Cycle of 25%



In a similar way, we can have a duty cycle of 75%.

Figure 9-5. Duty Cycle of 75%.



In Figure 9-5, the output is active for 75% of the time and inactive for 25% of the time.

Using General Purpose Timer 1 as a Digital-to-Analog Converter

We can take the input from ADC channel 1 and use it to control the duty cycle of the output on IOPB4. For this we shall take the value from the ADC and use it to set T1CMP.

The analog-to-digital converter produces a result in the range 0 to 1023. To make life easy for ourselves, we shall also use a time period of T1PR = 1023. This means that the input range maps directly to the output range.

Because the result of the analog-to-digital conversion is right justified, we must first shift the result 6 places to the right (divide by 2^6). We could use the instruction `SFR` (shift accumulator right) six times.

Another way to do this is to shift the result 10 places to the left and then use the high word of the accumulator. This is a multiply by 2^{10} then a divide by 2^{16}, which produces the net result of a divide by 2^6.

Example 9-4.

```
                .setsect ".text",       8800h

DP_DIGCR        .set 225                ; Data Page for Digital
                                        ; control
OCRA            .set 10h                ; Output control register

DP_ADC          .set  224               ; Data Page for ADC
ADCTRL1         .set  32h               ; ADC Control Register 1
ADCTRL2         .set  34h               ; ADC Control Register 2
ADCFIFO1        .set  36h               ; ADC measured value

DP_EV           .set  232               ; Data Page for Event
                                        ; Manager
GPTCON          .set  0h                ; GP Timer Control Register
T1PR            .set  3h                ; Timer 1 period register
T1CMP           .set  2h                ; Timer 1 compare register
T1CNT           .set  1h                ; Timer 1 count

start:          LDP #DP_DIGCR           ; Data Page for Digital
                                        ; control
                SPLK #1000h, OCRA       ; Configure T1PWM on IOPC4.

                LDP #DP_ADC             ; Data Page for ADC.
                SPLK #0C01h, ADCTRL1    ; Channel 0. Continuous
                                        ; conversion. Start
                                        ; conversion.
```

```
        SPLK #0005h, ADCTRL2    ; Conversion time.
        
        LDP #DP_EV              ; Data Page for timers.
        SPLK #8142h, T1CON      ; GP T1 off.
        SPLK #41h, GPTCON       ; PWM active low.
        SPLK #03FE, T1PR        ; Output period = 1023.
        SPLK #0h, T1CMP         ; Start with output off.
        SPLK #0FFFEh, T1CNT     ; Set count to negative.
        SPLK #9142h, T1CON      ; Enable GP Timer 1.
        
        CLRC SXM                ; Ensure ADCFIFO1 is taken
                                ; as a positive value.
        
loop:   LDP #DP_ADC             ; Page 224 ADC
        LACC ADCFIF01, 10       ; Load accumulator with
                                ; result of ADC conversion
                                ; but shifted 10 places to
                                ; the left.
        LDP #DP_TIMER           ; Change to timer page.
        SACH T1CMP              ; Store new value as duty
                                ; cycle.
        B loop                  ; Go round again.
```

Using General Purpose Timer 2

So far, we have exclusively used General Purpose Timer 1 (GP T1). We also have General Purpose Timer 2 (GP T2) at our disposal. This works in a similar way to GP T1.

This time we shall set up the timer so that the timer goes low when the compare occurs. The output frequency on IOPB5 will be 50 kHz.

Example 9-5.

```
        
        .setsect ".text",       8800h
        
DP_DIGCR .set 225               ; For digital control
OCRA    .set 10h                ; Output control register
        
DP_ADC  .set  224               ; Data page for ADC
ADCTRL1 .set  32h               ; ADC Control Register 1
ADCTRL2 .set  34h               ; ADC Control Register 2
ADCFIFO1 .set  36h              ; ADC measured value
        
DP_EV   .set 232                ; Data Page Event Manager.
GPTCON  .set 0h                 ; General-Purpose Timer
                                ; Control Register.
T2PR    .set 7h                 ; GP Timer 2 Period Register
T2CMP   .set 6h                 ; GP Timer 2 Compare Register
```

```
T2CNT     .set 5h                ; GP Timer 2 Count Register
T2CON     .set 8h                ; GP Timer 2 Control Register

start:    LDP #DP_DIGCR          ; Data Page for digital
          SPLK #2001h, OCRA      ; T2PWM on IOPC5.

          LDP #DP_EV             ; Data Page for timers
          SPLK #8142h, T2CON     ; Disable GP Timer 2.
          SPLK #44h, GTPCON      ; T2 active high
          SPLK #0C7h, T2PR       ; 199 + 1 = 200 counts
          SPLK #64h, T2CMP       ; 50% duty factor
          SPLK #0FFFEh, T2CNT    ; Count to negative value.
          SPLK #9142h, T2CON     ; Start GP Timer 2

loop:     B loop                 ; Go round again
```

When using the TMS320F243 DSK, OCRA bit 0 must always be 1, otherwise it jams the debugger.

Using the Timers to Control Software

We can use the General Purpose timers to control software. Flags are available in the Event Interrupt Flag Register (EV1FRA) at address 742Fh.

When general-purpose timer 1 overflows, the flag TIOFINT (bit 10 of EV1FRA) is set to '1'. This we can test using software.

The method of going around a loop and checking if a flag is set is known as *polling*. We continually poll the flag looking for a change of status.

Using GP Timer 1 to Control ADCs of the TMS320F243

From the DSP perspective, an important usage of GP T1 and GP T2 is to start an analog-to-digital conversion. In this mode of operation, the ADCs are not used in continuous conversion mode. The ADCs remain off until a specific timer event occurs.

Example 9-6 shows how timers can be used to control the ADCs of the TMS320F243. On the TMS320LF2407, the ADCs work in a different way.

Example 9-6.

```

          .setsect ".text",      8800h

DP_ADC    .set 224               ; Data page for ADC.
ADCTRL1   .set 32h               ; ADC Control Register 1.
ADCTRL2   .set 34h               ; ADC Control Register 2.
```

```
ADCFIFO1    .set 36h                ; Result of A-D conversion.

DP_DIGCR    .set 225                ; Digital I/O Control
                                    ; Registers.
OCRA        .set 10h                ; Output Control Register A

DP_EV       .set 232                ; Event Manager.
GPTCON      .set 4h                 ; General Purpose Timer
                                    ; Control Register.
T1PR        .set 3h                 ; GP Timer 1 Period Register
T1CMP       .set 2h                 ; GP Timer 1 Compare
                                    ; Register.
T1CNT       .set 1h                 ; GP Timer 1 Counter
                                    ; Register.
EVIMRA      .set 2Ch                ; EV Interrupt Mask Register
                                    ; A.
EVIFRA      .set 2Fh                ; EV Interrupt Flag Register
                                    ; A.

start:      LDP #DIG_CR             ; Digital I/O Control
                                    ; Register.
            SPLK #1001h, OCRA       ; T1PWM on IOPB4.

            LDP #DP_ADC             ; Data Page for ADC.
            SPLK #0800h, ADCTRL1    ; Channel 0. ADC1 Enabled.
                                    ; Single conversion.
            SPLK #0407h, ADCTRL2    ; Synchronize with GPT1.

            LDP #DP_EV              ; Data Page for GP Timers.
            SPLK #8142h, T1CON      ; Turn of GP Timer 1.
            SPLK #0141h, GPTCON     ; T1PINT starts ADC.
            SPLK #03FEh, T1PR       ; Max output ADC = 3FFh.
            SPLK #0200h, T1CMP      ; 50% duty cycle.
            SPLK #9142h, T1CON      ; Start GP Timer 1.
            SPLK #0FFFFh, EVIFRA    ; Clear any pending
                                    ; interrupts.
            SPLK #0080h, EVIMRA     ; Enable T1 period interrupt
            CLRC SXM                ; For later shift.

loop:       LDP #DP_ADC             ; Data page for ADC.
            LACC ADCTL1             ; Test if an ADC interrupt
            AND #0100h              ; event has occurred.
            BCND skip, EQ           ; Skip if no ADC event
            SPLK #0100h, ADCTRL1    ; Reset ADCINTFLAG.
            LACC ADCFIFO1, 10       ; Read ADC and shift right
            LDP #DP_EV              ; Data page for timers
            SACH T1CMP              ; ADC value into T1CMP

skip:       LDP #DP_EV              ; Data page for timers.
            LACC EVIFRA             ; Load flags register.
            AND #0080h              ; Test for T1PINT interrupt
```

	BCND *loop*, EQ	; Branch if no timer event.
	SPLK #0080h, EVIFRA	; Reset T1PINT.
	B *loop*	; Go round again.

When resetting EVIFRA flags, writing the value 0 to a particular bit has no effect.

When T1PINT flag is set in EVIFRA, it must be reset, otherwise further ADC conversions do not occur.

TMS320F243 DSK EXPERIMENTS

The following experiments have been designed to run on the TMS320F243 DSK. If another TMS320F24x DSK/EVM is being used then the clocks will most likely run at a different speed.

Equipment Required

TMS320F243 DSK
Frequency meter or oscilloscope
Analog meter or oscilloscope
Linear Potentiometer for Experiment 9-5.

Experiment 9-1.

Objective: To Generate a 100 kHz Output on IOPB4

Using a text editor, enter the code in Example 9-3, save it, assemble it and run it on the debugger. Measure the output between IOPB4 (Connector P2, Pin 15) and Ground (Connector P2, Pin 19 or Pin 20) using a frequency meter or an oscilloscope. The output should be 100 kHz.

To show that the code entered actually controls the output pin, change the line `SPLK #63h, T1PR` to `SPLK #0C7h, T1PR` . Assemble and run it on the debugger. The output frequency measured between IOP4 (Connector P2, Pin 15) and ground (Connector P2, Pin 19 or Pin 20) should now be 50 kHz. Measure this frequency using a frequency meter or an oscilloscope.

Experiment 9-2.

Objective: To Generate a 10 kHz output on IOPB4

Load the code in Example 9-3 into a text editor. Change the line `SPLK #63h, T1PR` to `SPLK #999, T1PR`. This makes the value T1PR to 999 decimal.

Save the file, assemble it and run it on the debugger. The output measured between IOPB4 (Connector P2, Pin 15) and Ground (Connector P2, Pin 19 or Pin 20) should now be 10 kHz.

Experiment 9-3.

Objective: To Generate a 1 MHz output on IOPB4

Load the code in Example 9-3 into a text editor. Change the line `SPLK #63h, T1PR` to `SPLK #9, T1PR`. .

Save the file, assemble it and run it on the debugger. The output measured between IOPB4 (Connector P2, Pin 15) and Ground (Connector P2, Pin 19 or Pin 20) should now be 1 MHz.

Experiment 9-4.

Objective: To Generate a PWM waveform using GPT2.

Enter the code in Example 9-5 into a text editor, save it, assemble it and run it on the debugger.

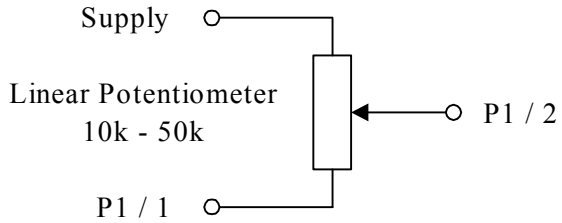
Monitor the output between IOPB5 (Connector P2, Pin 16) and Ground (Connector P2, Pin 19 or Pin 20). It should be 50 kHz.

Experiment 9-5.

Objective: To Adjust the Duty Cycle from an ADC Input.

Enter the code in Example 9-6 into a text editor, save it, assemble it and it on the debugger. Connect an input potentiometer as shown in Figure 9-6.

Figure 9-6. Configuration of Input Potentiometer for ADC.



For the TMS320F243 DSK, the supply can be up to 5V. However, if a 3.3V device is being used such as the TMS320LZ2407, then the maximum allowed supply is 3.3V.

Monitor the output on IOPB4 (Connector P2, Pin 15) and Ground (Connector P2, Pin 19 or Pin 20) using an analogue meter. As the potentiometer is turned from one end of its travel, the meter reading should change between 0V and 5V. Note: some potentiometers do not go fully from one end of their range to the other, so that the extremes may not reach 0V or 5V.

Design Problem 9-1.

Objective: To Generate the slowest output on IOPB4.

Modify the code in Example 9-3 to use generate the lowest frequency possible on IOPB4. It will be necessary to alter both the pre-scaler and the period. Assemble the code then load it and run it on the debugger.

Measure the output between IOPB4 (Connector P2, Pin 15) and Ground (Connector P2, Pin 19 or Pin 20) using a frequency meter or an oscilloscope. What is the frequency?

Self-Test Questions..... *Click Here To View* Answers!

1.	What is the problem in using time delays for controlling the time between events?
2.	What is the *Event Manager* of the TMS320F24x?
3.	What is meant by the term GP T1? a) General Purpose Timer 1 b) Generate Product Term 1 c) Greater Product Than 1 d) Group Product Time 1
4.	What is meant by the term "continuous up-counting mode"?
5.	At power up, the PWM outputs of the TMS320F24x controlled by GPT1 and GPT2 are turned off. How do we turn them on?
6.	What is a *pre-scaler*?
7.	What is the maximum value we can use for T1PR?
8.	What is the minimum value we can use for T1PR?
9.	What is meant by the term *period* when applied to a timer?
10.	What is meant by the *term duty* cycle?
11.	Why is it important to setup T1CNT to –1 or –2 when setting up GP T1?
12.	What happens if we make T1CMP greater than T1PR?
13.	The instruction `LACC` allows us to shift left only. How do we use this instruction to perform a shift to the right?
14.	What is meant by the term *polling*?

References

TMS320F/C24x DSP Controllers. CPU and Instruction Set. Reference number SPRU160

TMS320F/C240 DSP Controllers. Peripheral Library and Specific Devices. Reference Number SPRU161.

TMS320F243, TMS320F241 DSP Controllers. Reference Number SPRS064.

CLICK TO VIEW
Tutorials
1 2 3 4 5 6 7 8 9 10

Click here to view..... Route Map