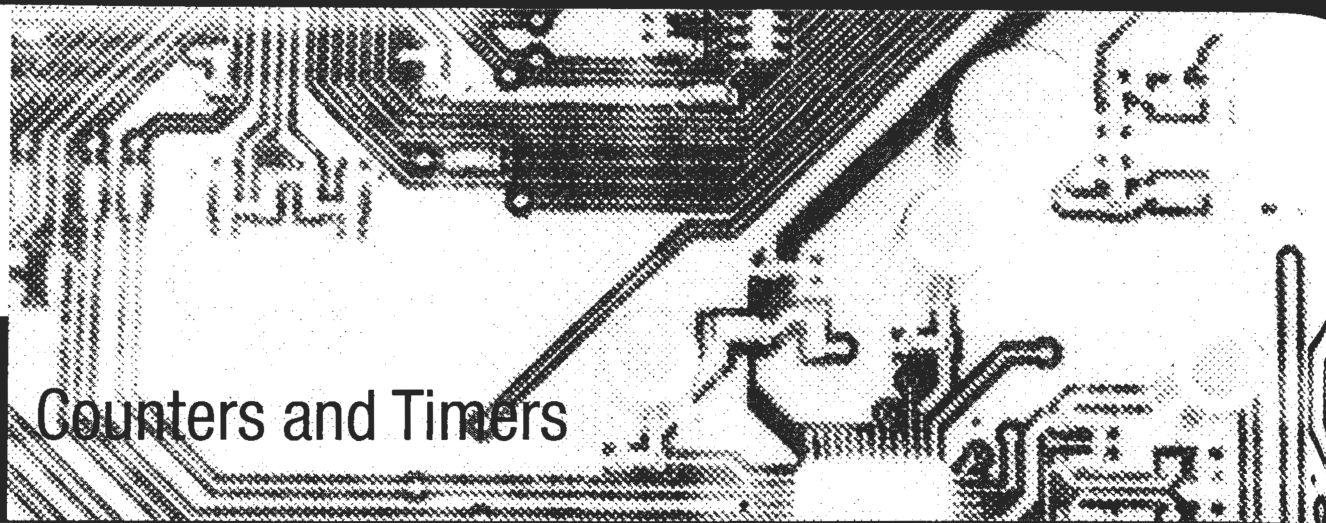


14 Counters and Timers

Objectives

Many embedded applications require a timer to generate waveforms of a specific frequency, to time external events, to count events, and to generate interrupts at specific intervals. In this chapter we will look at the basic operation of the timer circuits found in modern microcontrollers.

14.1 Introduction

Designers of embedded systems often refer to "real-time" events, or real-time control. Real time does not mean time in hours, minutes, and seconds. Instead, the term usually refers to generating time intervals to create waveforms of a specific frequency for driving stepper motors, for example, or for generating interrupts to acquire data from an external source such as an analog-to-digital converter. The timer module in your microcontroller can do the following functions:

- Generate accurate timing signals and waveforms.
- Measure time intervals.
- Generate interrupts at specific intervals.
- Capture and count external events.

14.2 The Timer/Counter

The Timer Overflow

The heart of the timer module is actually a counter, as shown in Figure 14-1. A programmable (in some microcontrollers but not in others) divider or prescaler reduces the bus clock

TMR0 APPLICATIONS 887

1. TIMING TO CAUSE INTERRUPTS EVERY <u>1 ms</u> (TMR0 8 bits)

Resolution $\frac{1\,ms}{2^8} = 0.003906\ sec \times 10^{-3}\,s$

$= 3.906\ \mu s$

$F_{OSC} = 4\ MHz$

$f_{input} = 4\,MHz/4 = 1\,MHz \quad \Delta T_{in} = 10^{-6}\,sec$

RANGE:

$10^{-3}\,sec = \Delta T_{imer} \times 2^8$

So $\Delta T_{Time} = \frac{10^{-3}\,sec}{2^8} = 3.906\,\mu s$ as before

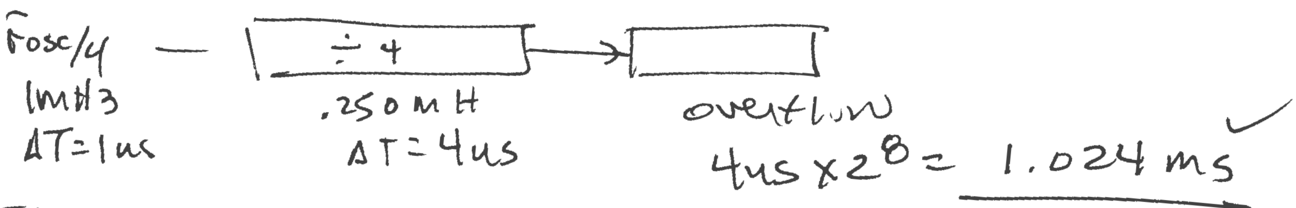
CHECK $T_{overflow} = 3.906 \times 10^{-6}\,s \times 2^8 = 1.000\,msec$

CALCULATE PRESCALAR

$\Delta T_{Timer} = \frac{2^{(BITS+1)}}{F_{OSC}/4} = 2^{(BITS+1)} \cdot 10^{-6}\,sec$

$2^{(BITS+1)} = 3.906\,\mu s \times 10^6 = 3.9 \approx 4$

Choose <u>BITS = 1</u> ie <u>$2^2 = 4$</u> AND CHECK PSA = 001_2



OVERFLOW T0IF (INTCON(2)) = 1 FLAG

INTERRUPT IF T0IE (INTCON(5)) = 1 <u>AND</u> GIE = 1

2. EXTERNAL TIMING OR COUNTING

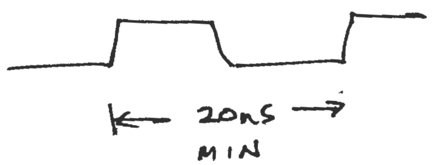
USE THE RA4/TOCKI PIN6 ON CHIP

SET TOCS (OPTION_REG(5)) = 1

TOSE (OPTION_REG(4)) = 1 ↓ 0 ↑

INPUT TRANSITION

PEATMAN P101-102



$\frac{1}{20\times10^{-9}} = 5\times10^{-2}\times10^{9}$

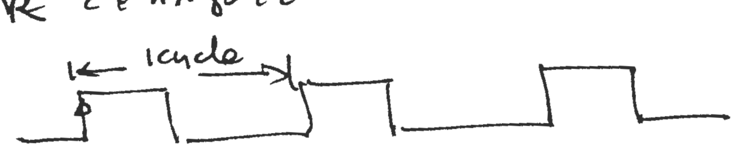
$= 50\times10^{6}$ Hz

EXAMPLE

50 MHz INPUT; PRESCALE BY 256

$f_{input} = \frac{50\times10^{6}}{256} \approx$ 200 kHz OK

CAN BE USED TO COUNT FREQUENCY
FOR EXAMPLE



COUNT N TRANSITIONS in T seconds

a) TMR0 IS INCREMENTED ON EACH
TRANSITION RISING

b) ANOTHER TIMER IS USED
FOR EXAMPLE 10 TRANSITIONS / ms
MEANS 10 × 1000 = 10,000 cycles/sec
= 10 kHz.

TIMER Ø

11/15/20

Peat p102

INTERRUPT Every 2048us

$F_{OSC}/4 = 1\,MHz \quad T_{OSC4} = 1\,us$

Prescale

$$T_{OSC\phi} \cdot \text{Prescale} \times \underset{(\text{Range})}{2^8} = 2048\,us$$

So $\text{prescale} = \dfrac{2048\,us}{1\,us \times 2^8} = 8$ ✓

IN GENERAL

1. MINIMUM (RESOLUTION)

   $T_{OSC} \times \text{PRESCALE}$

2. MAXIMUM TIME FOR N-BIT REGISTER

   $(T_{OSC} \times \text{PRESCALE}) \times 2^N$

TIMER Ø 8 bits

TIMER 1 16 bits

TIMER Ø   OPTION_REG (5) = 0   INTERNAL CLOCK Fosc/4
TOCS = 0

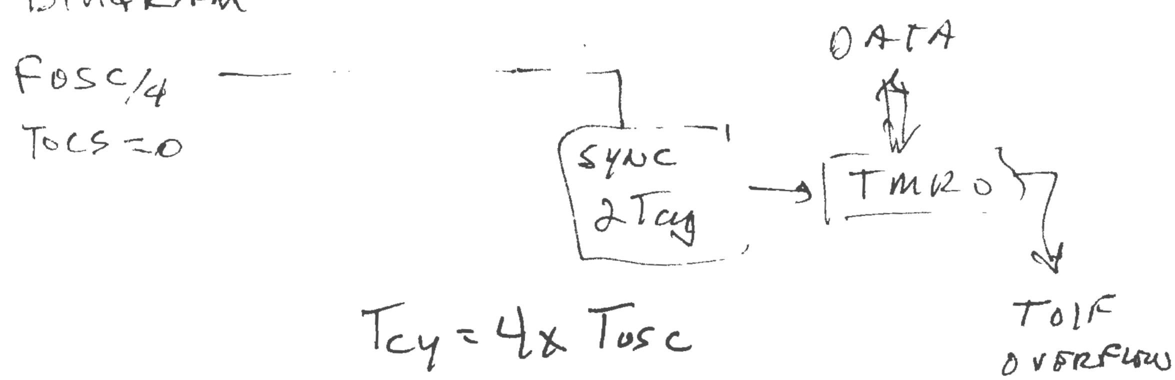
Fosc → ~~Fosc~~/4 EXTERNAL 20 MHz MAX → 5 MHz = Fosc

INTERNAL Fosc
P62 DATA SHEET

SELECT IN OSCCON REGISTER

	Tosc
8 MHz	125 ns
4 MHz	250 ns
2 MHz	500 ns
1 MHz	1 us
500 kHz	2 us
250 kHz	4 us
125 kHz	8 us
31 kHz	32.3 us

P73 DIAGRAM



$T_{cy} = 4 \times T_{osc}$

So Resolution 125 ns (minimum at 8 MHz)

Range $2^8 \times 125 \times 10^{-9}$ sec = 32000 ns
= 32 us

Others Range = $2^k \times 32$ us   k = 1, 2, 3, . . .

For 31 KHz Range = $2^8 \times 32.3$ us = 8.268 ms

APPLICATIONS

MEASURE TIME WITH TIMER Ø.

OPTION REGISTER

SOURCE: TOCS (BIT 5) = 0 — Fosc/4 (INTERNAL)

PRESCALE: PSA (BIT 3) = 0 — TIMER Ø

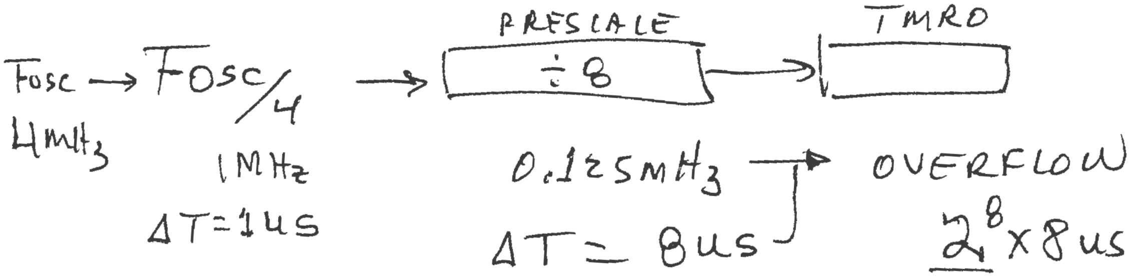
PRESCALER: PS <2:0> — VALUE IS $2^{[BITS+1]}$

EXAMPLE: — PEATMAN P 100

$F_{osc} = 4\,MHz$

$TMR0_{CLK} = 1\,MHz$ ($\Delta T = 1\,us$) BEFORE PRESCALE

PRESCALE (010_2) = 8 (DIVIDES FREQUENCY)



$2^8 \times 8\,us = \underline{2.048\,MS}$

IF $F_{TIMER} = \dfrac{F_{OSC}/4}{2^{(BITS+1)}}$; $\Delta T_{Timer} = \dfrac{2^{(BITS+1)}}{F_{OSC}/4} = \dfrac{PRESCALE}{F_{INPUT}}$

SO RESOLUTION $\Delta T_{imer} = 8 \times \dfrac{1}{1\,MHz} = 8 \times 1\,us = 8\,us$

RANGE $\Delta T_{imer} \times 2^8 = 8\,us \times 2^8$

```
%    CAdy Chapter 14
%      Timer Range 16 bits
format long
Bits16=16
fcyc=[40,8,4,.1,.032768]*10^6    % Mhz clock
prescale=1                        % Prescalers are usually 1,2,4,8,...256
%  For the PIC24F  Prescale = 1, 8, 64, 256 ONLY
%
fprintf('Time in Nanoseconds')
TimerResolution = (1./fcyc)*prescale*(10^9)  % NanoSeconds between counts
TimerRange=(1./fcyc)*prescale*(2^Bits16)      % Seconds to timeout
%
% Example 14-1 Cady
% 40 MHz        8MHZ,       4MHz,        100 kHz,     32.768 kHz
%  Resolution
%   25 ns        125 ns       250. ns       10 us       30.518 us
%  TimerRange
%   1.63840 ms   8.192 ms    16.384 ms      0.65536 sec    2.00 sec
%
%      24F   16 and 32 bit timers possible Fmax= 40 MHz
prescale24F=[1 8 64 256]    % Prescaler value (Multiply period=divide fcyc
fcyc24= 40*10^6             %Hz
fprintf('Time in Nanoseconds')
timeResolution24F=(1./fcyc24)*prescale24F*(10^9)  % Resolution in nanosec.
fprintf('Time Range in Milliseconds')
timeRange24F16=timeResolution24F*(2^16)*(10^-6)
fprintf('Time in Seconds')
timeRange24F32=(1./fcyc24)*prescale24F*(2^32)
% PRE=              1          8          64          256     % Scaling
% Resolution (ns)   25        200        1600        6400
% 16bit Range(ms) 1.6384    13.1072    104.8576     419.430400
% 32bit Range(s)  107.374   858.9934  6871.9476     27487.7906944
```