ANALOG DEVICES

AN-644
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106 • Tel: 781/329-4700 • Fax: 781/326-8703 • www.analog.com

Frequency Measurement Using Timer 2 on a MicroConverter®

By Eamon Neary

INTRODUCTION

This application note describes the implementation of a single pin measurement of frequencies up to 500 kHz using the Timer 2 counter input pin. The code accompanying this document implements a frequency acquisition system that can be combined with a voltage measurement via the ADC to track both the frequency and voltage of the input signal. The measurement is displayed on an HD44780 compatible LCD screen. The display.c code accompanying this application note implements the routines necessary to output to the LCD display.

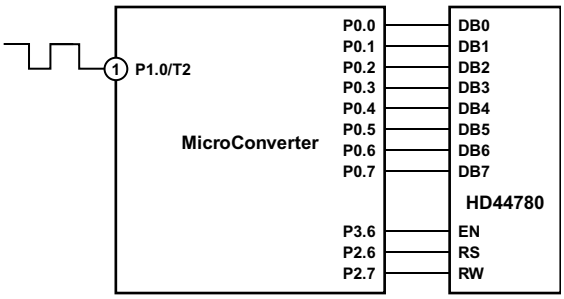


Figure 1. Setup of LCD Display

MEASURING THE FREQUENCY

Since frequency is the number of cycles of a given waveform recorded over 1 second, both the number of cycles and the time taken for that number of cycles have to be measured. By measuring the number of cycles over 1 second, the frequency of the waveform is determined.

In order to measure the number of cycles, a record of the number of times a 1-to-0 transition occurs on the waveform should be maintained. This is done using the Timer 2 counter input pin, which increments the Timer 2 registers on a 1-to-0 transition.

There are two ways to measure a second. The first involves setting up Timer 0 with reload values such that it overflows when 10 ms has elapsed. By counting 100 of these overflows, a 1 second interval can be measured. See freq.c for an example.

The second method is to use the time interval counter, which can be set up to interrupt the core after 1 second has elapsed. The advantage here is that the core is interrupted less frequently than a segmented count using Timer 0, and thus is free to carry out tasks between these interruptions. See tic.c for an example.

Since the Timer 2 registers can hold only a 16-bit result, the maximum measurable frequency is 65535 Hz. In order to extend this frequency, a record is kept of the number of times Timer 2 overflows during the 1 second interval. This is then factored into the calculation of the actual frequency during this interval.

Implemented in both methods is a calibration function. If a 100 kHz square wave is input to P1.0 and the INT0 button is pressed, the software will calibrate the frequency measurement to compensate for errors that interrupt latency introduces at the higher frequencies; however, the error is only significant above 10 kHz, so the calibration routine ignores the gain error below this frequency.

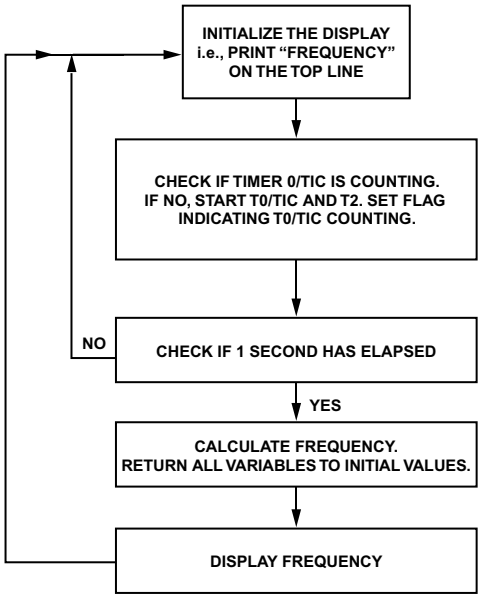
PROGRAM EXAMPLE

The following C code example illustrates the concept outlined earlier. The resulting frequency is output to an LCD display. Though this code is written in C, the concept of communicating with the LCD is explained in the AN-645 (uC014) application note. This code is written for use with the ADuC816, ADuC824, and ADuC834 MicroConverters, though similar principles can be used for other MicroConverters.

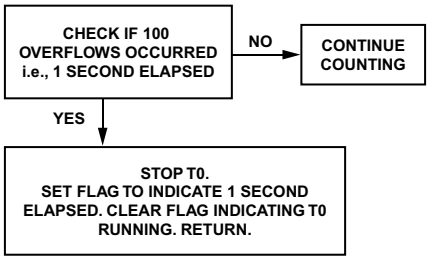
This C code example contains three files: freq.c, tic.c, and display.c. Freq.c implements the T0 time base, while tic.c uses the time interval counter to measure the second. Both freq.c and tic.c call the LCD display routines in display.c. Most standard 8052 C compilers will be able to compile these files.

PROGRAM FLOW

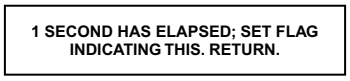
Frequency Measurement Routine Flowchart



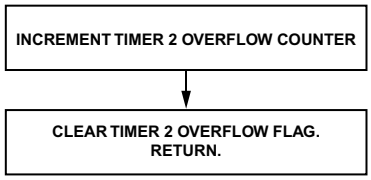
Interrupt Service Routine for Timer 0 Flowchart



Time Interval Counter Interrupt Service Routine



Timer 2 Interrupt Service Routine Flowchart



E03676–0–4/03(0)