What are the Limits on Pulse Width or Frequency of signals I can generate with a CTM-05/A or CTM-10 board?

Introduction

DriverLINX for the CTM series of boards can program several tasks that will generate an output signal such as: One-Shot Pulse, Variable Duty Cycle Pulse Train, Square Wave generation (50% duty cycle pulse train). Often, the limits of frequency or duration of these pulses is a point of confusion. This document will provide some background information on both the CTM board hardware and the DriverLINX implementation to explain the limits of use.

Counters and Output Signals

In general, the output of a counter will change state when an internal register has counted to its full-scale value (terminal count). Each counter on the CTM is a 16bit counter. This means, the full-scale value is 65535 ($2^{16} - 1$). The internal register can be pre-loaded with values in the range of 0 to 65535 to give the counter a "head start" toward reaching terminal count. Both the frequency of the timebase signal applied to the counter's input terminal and this initial value in the internal register determine how often the output will change state and the resulting output signal.

Below is a simplified diagram showing the output changing state for every two pulses at the input. The internal register is pre-loaded with a value of 65533 or two away from terminal count.

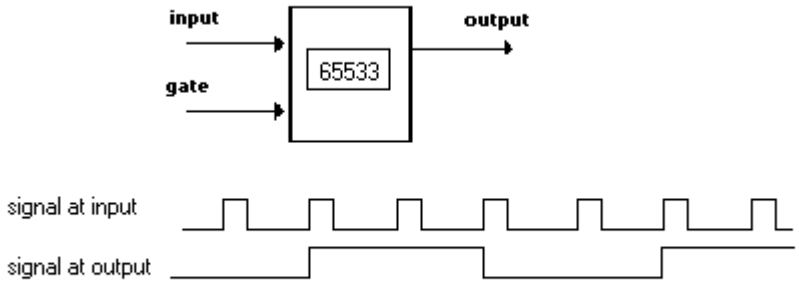

Figure 1

Let's take the case of generation of a 100 Hz square wave. If the input to the counter is a 1.0 MHz signal, the counter must be programmed with a value of 10,000 to result in a 100Hz signal at the output. Each time the counter accumulates 10,000 pulses from the input signal, the output will change state and generate the desired slower output signal.

Suppose instead, a 1 Hz square wave is desired. A single 16bit counter cannot divide the 1.0 MHz input signal sufficiently to accomplish the task. In this case, either additional counter channels must be recruited or select one of the slower internal timebase signals that are available for the CTM Series of boards.

Features and Configuration of CTM Boards

The CTM boards make use of a flexible and complex chip: AM9513A. This chip provides several choices for an internal timebase that might serve as an input to the five counters of the chip.

The CTM boards from Keithley make use of a 10MHz crystal, however, this signal is divided before it is presented to any of the counters.

This 10MHz signal is reduced to either a 1.0 MHz or a 5.0 MHz signal. The setting in the DriverLINX Configuration Panel controls which one of these is in use by the board/driver. Figure 2 below shows the 1.0 MHz selection. Use the drop down to select the 5.0 MHz setting instead. A reboot after exiting the DriverLINX Configuration Panel is required for a change of setting to take effect.

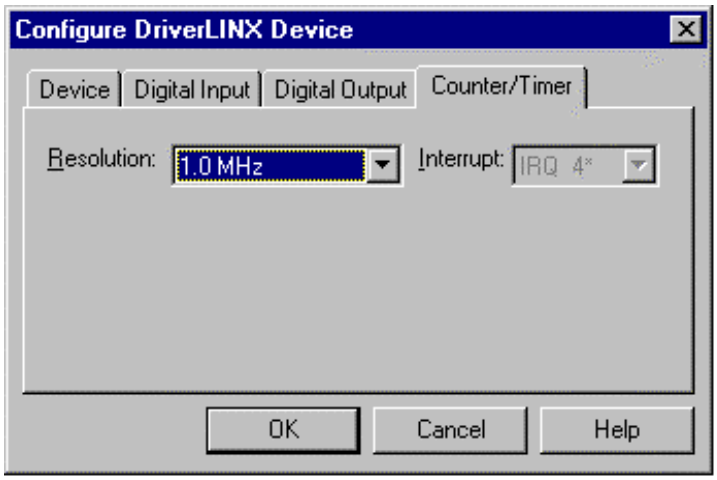

Figure 2

The CTM boards (AM9513A chip) also feature a programmable divider that can further reduce the 1.0 MHz or 5.0 MHz signal before it is applied to the input of a counter. This programmable divider makes available 4 additional timebase frequencies derived from the core 1.0 MHz or 5.0 MHz selection. Appendix D of the *CTM-10 and CTM-05/A User's Guide* contains more information on this Crystal Oscillator Scaler (MM15 of the AM9513A chip's Master Mode Register).

The divider can make use of BCD or Binary division (divide by 10 or by 16). Figure 3 is a screen shot of the dialog presented when the 'Special' button of the DriverLINX Configuration Panel is pushed. The radio buttons control whether Binary or BCD division will be used to further reduce the selected timebase resolution from the 'Counter/Timer' tab.

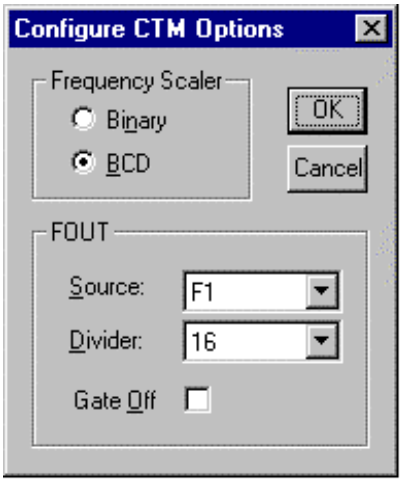

Figure 3

Applying this Information

Below is a table that outlines the input frequencies that can be available to the input of a counter depending on the DriverLINX configuration:

Freq Scaler	rateClock	@ 1MHz resolution	@ 5MHz resolution
BCD	Internal1	1,000,000 Hz	5,000,000 Hz
BCD	Internal2	100,000 Hz	500,000 Hz
BCD	Internal3	10,000 Hz	50,000 Hz
BCD	Internal4	1000 Hz	5000 Hz
BCD	Internal5	100 Hz	500 Hz
Binary	Internal1	1,000,000 Hz	5,000,000 Hz
Binary	Internal2	62,500 Hz	312,500 Hz
Binary	Internal3	3906 Hz	19,531 Hz
Binary	Internal4	244 Hz	1220 Hz
Binary	Internal5	15 Hz	76 Hz

The rateClock column in the table refers to DriverLINX programming syntax for selecting the signal that will be the input to the counter. When using the ActiveX API of DriverLINX (VB or Delphi typically), the .Evt_Tim_rateClock is the property; when using the DLL API (C/C++ typically), the timing.u.rateEvent.clock member of the SR structure is used.

Let's return to the 1Hz square wave objective from the fourth paragraph. In the earlier discussion, we were constrained by the 1MHz timebase. From the information above, the DriverLINX configuration provides many means by which to have a slower timebase available. For the sake of discussion, let's assume DriverLINX is configured to use the 1MHz resolution and BCD frequency scaling. By selecting the Internal5 timebase in the code and programming the counter with a value of 100, a 1Hz output signal would result.