SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

PLL Generation using ST62 Auto-reload Timer

J.NICOLAI

INTRODUCTION

This note describes how to generate a digital signal locked in phase and frequency (PLL) with a calibrated delay starting from an active edge on the Auto-reload timer input pin.

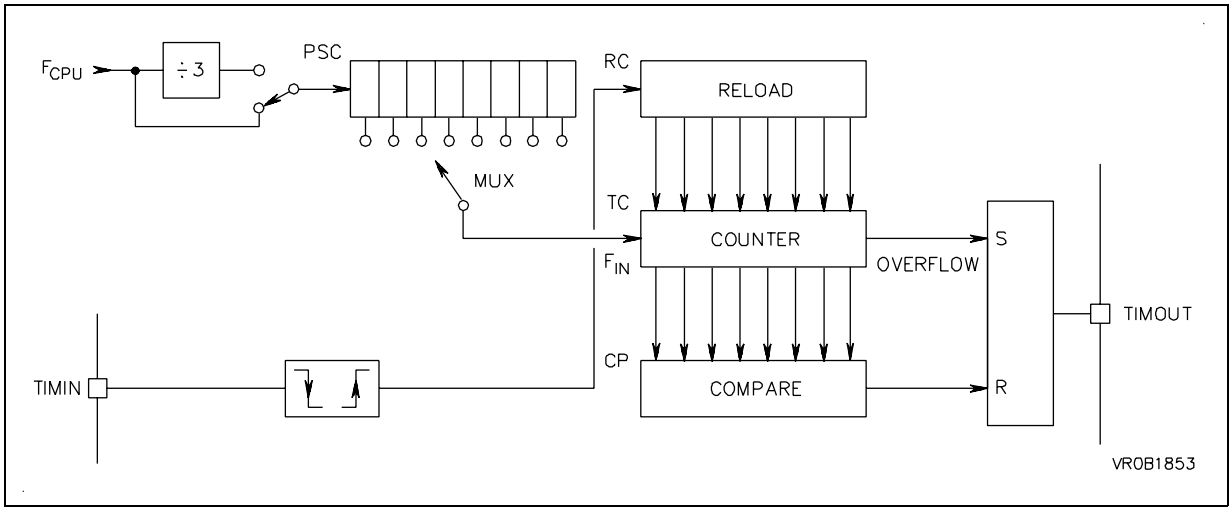
Auto-reload timer description

This timer is an 8 bit timer/counter with prescaler. It includes auto-reload PWM, capture and compare capability with one input and one output pins. It can be controlled by thefollowing registers (8 bit):

- Mode Control Register (MC)
- Status registers (SC0, SC1)
- Load register (LR)
- Incremental counter register (TC)
- Compare register (CP)
- Reload/Capture register (RC)

It can also wake-up the MCU from wait mode and exit from stop mode if an external event is present on the input pin. The prescaler ratio can be programmed to choose the timer input frequency F_{IN} (see Table 1).

Figure 1. Auto-reload Timer Block Diagram



Example:

The TIMIN input receives a 15 kHz digital signal. We want to generate a phase-locked 15 kHz digital signal with a falling edge delayed 19µs from the input rising edge, and a duty cycle of 75%. The CPU quartz frequency is 8 MHz.

The Figure 2 shows the TIMOUT signal generated in “load on external edge” mode, given the above TIMIN signal: on TIMIN rising edge, the TC count register is loaded with the value contained in RC register (160 in this example). The timer will resume counting from value 160. When the compare value (210 in this example) is reached, the TIMOUT signal goes down. The timer keeps counting until the overflow (255) is reached. At this point, signal TIMOUT rises again. The timer keeps counting from 0 until next active edge on TIMIN. At this time, TC is loaded again with the RC value (160) and so on...

The delay from TIMIN edge to TIMOUT falling edge is given by CP-RC (multiplied by the TC register clock period Fin). The low level duration on TIMOUT is given by 255 - CP. The remaining of the TIMOUT period is variable, and will adjust to the TIMIN period: small variations of TIMIN period will be absorbed by a variation of T_{VAR}. The following rule must be respected in order to get the proper output signal:

The variable time Tvar must stay smaller than CP, (otherwise the falling edge on TIMOUT occurs before rising edge on TIMIN) and larger than 0 (otherwise the rising edge on TIMOUT never occurs).

In other words, the period of input TIMIN (TOT in terms of TC clock cycles) must meet the following requirement:

255 - RC < TOT < 255 - RC + CP

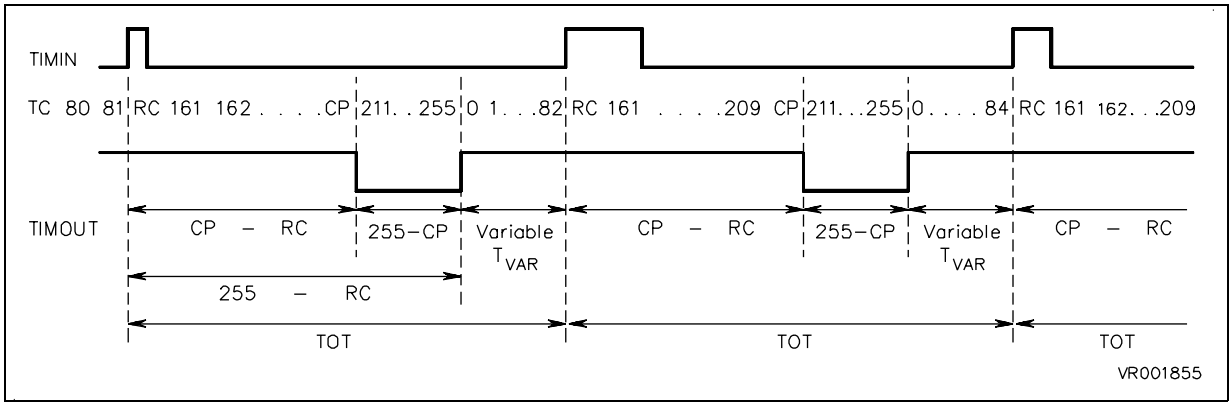
Coming back to our example, let’s calculate the timer settings:

The input period is Tin = 1/15 kHz = 66.7µs

Table 1. Prescaler Programming Ratio

Bit 0 Reg. SC1	PS2	PS1	PS0	PRESCALER Ratio
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	3
1	0	0	1	6
1	0	1	0	12
1	0	1	1	24
1	1	0	0	48
1	1	0	1	96
1	1	1	0	192
1	1	1	1	384

Figure 2. TIMOUT signal



Note: All numbers are decimal

Calculation of the prescaler ratio:

We want the best possible resolution, e.g. the smaller possible prescaler ratio: we would like the TC counter to count up to the highest number: 255 (for best resolution). In this case 66.7µs / 255 = 0.26µs is the smallest TC clock period that we can use. With a prescaler ratio of 1, the TC clock period is 1/8MHz = 0.125µs. So we need a prescaler ratio of 3, giving a TC clock period of 3/8 MHz = 0.375µs. This is the elementary incrementing time of the TC counter, which gives the resolution of the phase shift and of the low level on TIMOUT (the TC counter is incremented with a clock F_{IN} = 1/0.375µs = 2667kHz).

The desired delay is 19µs: CP - RC = 19µs /0.375µs = 50.67.

The TIMIN period is:

TOT = 66.7µs / 0.375 µs = 177.78.

The duty cycle is: 0.25 = (255 - CP)/TOT

Rounding the decimals, this gives:

RC = 160; CP = 210; TOT = 178

The variable time T_{VAR} is: T_{VAR} = TOT - 255 + RC = 83

The condition is met: Tvar is positive and smaller than CP: (255 - 160) < 178 < (255 - 160 + 210)

The TIMOUT signal will remain correct and stable as long as this condition is met, even if the input frequency varies:

The input frequency limits for output correctness are:

(255 - 160) x 0.375µs < T_{IN} < (255 - 160 + 210) x 0.375µs

The output signal will remain locked in phase and frequency as long as the input signal TIMIN is in the range:

8.7kHz < TIMIN frequency < 28kHz

Of course, it is also possible to modify by software the 19µs delay and the 75% duty cycle, for example by measuring repetitively an error voltage with the A/D converter and calculating the modified delay or duty cycle.

Program example

```
;**************** A-R Timer Register Set ******************
RC         .def 0D9h,0FFh,0FFh     ;reload/capture register
CP         .def 0DAh,0FFh,0FFh     ;compare register
MC         .def 0D5h,0FFh,0FFh     ;mode control register
SC0        .def 0D6h,0FFh,0FFh     ;status/control register 0
SC1        .def 0D7h,0FFh,0FFh     ;status/control register 1
LR         .def 0DBh,0FFh,0FFh     ;load register
;============================================================
           ldi CP, 210             ;compare register = 210d
           ldi RC, 160             ;reload register = 160d
           ldi SC1,00000101b       ;clock source= CPU clock divided by 3
                                   ;rising TIMIN edge active
                                   ;pull-up disabled, prescaler ratio = 1
           ldi MC, 11100011b       ;load on TIMIN mode,interrupts disabled
                                   ;PWMOUT enabled, start timer
```

THE SOFTWARE INCLUDED IN THIS NOTE IS FOR GUIDANCE ONLY. SGS-THOMSON SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM USE OF THE SOFTWARE.