SGS-THOMSON MICROELECTRONICS

# APPLICATION NOTE

## Input Capture with ST62 Auto-reload Timer

**J.NICOLAI**

### INTRODUCTION

This note presents how to use the Auto-reload timer (ARTimer) of the ST62 to measure durations or frequencies of an input signal.

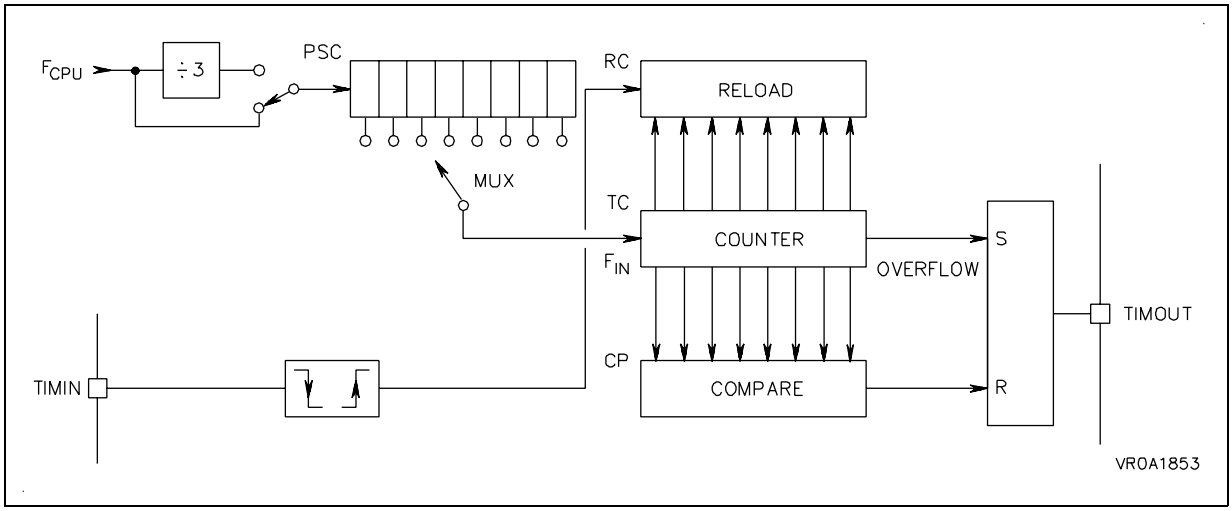
### Auto-reload timer description

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**



### Capture Mode With Reset

This can be used to measure time durations or frequencies.This mode is used to measure the time elapsed between two edges of an external signal: two rising edges, two falling edges, or one rising and one falling if the configuration of the timer is modified after the first edge detected.

The minimum duration of one signal to measure depends on the CPU clock and on the required precision. With a 8MHz Quartz crystal, a signal of 8µs duration can be measured with a resolution of 1/64.

**Example:**

Let's measure the time elapsed between two rising edges on TIMIN:

The prescaler ratio must be programmed according to the expected duration to measure. In this example it is programmed to: prescaler ratio = 4, clock source = CPU clock. (in case the duration expected can be longer than the 255 count span of the TC count register, the overflow interrupt can be used to increment a RAM variable, used as the most significant part of the result).

In the program example, waiting for the edge on TIMIN is made by a software loop (jrr 2, SC0, wt_edge), but it is also possible to use the external event interrupt to read the result immediately after the edge without loading the rest of the program. It is also possible to read the result directly from RC without waiting or using the interrupt, if the duration is required at an asynchronous time, not just after the edge.

**Table 1. Prescaler Programming Ratio**

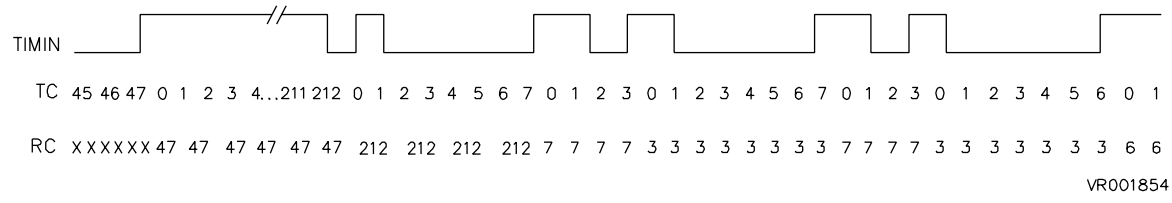
| Bit 0 Reg. SC1 | PS2 | PS1 | PS0 | PRESCALER Ratio |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 2 |
| 0 | 0 | 1 | 0 | 4 |
| 0 | 0 | 1 | 1 | 8 |
| 0 | 1 | 0 | 0 | 16 |
| 0 | 1 | 0 | 1 | 32 |
| 0 | 1 | 1 | 0 | 64 |
| 0 | 1 | 1 | 1 | 128 |
| 1 | 0 | 0 | 0 | 3 |
| 1 | 0 | 0 | 1 | 6 |
| 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 |

**Program example**

```
;**************** accumulator ***************************
A  .def 0FFh,0FFh,0FFh ;Accumulator
;**************** RAM variable ***************************
period .def 084h,0FFh,0FFh ;elapsed time
;**************** 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 SC1,01000100b       ;clock source = CPU,
                                   ;prescaler ratio = 4,
                                   ;pull-up on TIMIN disabled,
                                   ;capture on rising edge of TIMIN
           ldi MC, 11000010b       ;capture with reset, ints. disabled
                                   ;PWMOUT disabled, start timer
; .
; . normal program
; .
;now we want to know the elapsed time between last and
;next edge on TIMIN:
           res 2, SC0              ;clear external event flag
wt_edge
           jrr 2,SC0,wt_edge       ;wait for edge on TIMIN, then
           ld A,RC                 ;read elapsed time from reload/capt
           ld period,A             ;register into variable “period”
```

The following figure shows the evolution of the contents of TC and RC during capture operation: a rising edge on TIMIN results in copying the latest value of TC into RC, and clearing TC. Afterwards, TC increments at the selected clock rate until the next capture or until it reaches 255. In the latter case, it wraps to 0. In capture mode, TC always counts from 0 to 255, it is never loaded from the RC register.

**Figure 2. TC and RC Evolution**



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.