**LAB4: TIMERS/COUNTER**

1. **Objectives:**

- Provide an overview of the timers of 8051

- Write a C program to generate a time delay using timer.

1. I**ntroduction**

As you already know, the microcontroller oscillator uses quartz crystal for its operation. As the frequency of this oscillator is precisely defined and very stable, pulses it generates are always of the same width, which makes them ideal for time measurement. Such crystals are also used in quartz watches. In order to measure time between two events it is sufficient to count up pulses coming from this oscillator. That is exactly what the timer does. If the timer is properly programmed, the value stored in its register will be incremented (or decremented) with each coming pulse, i.e. once per each machine cycle. A single machine-cycle instruction lasts for 12 quartz oscillator periods, which means that by embedding quartz with oscillator frequency of 12MHz, a number stored in the timer register will be changed million times per second, i.e. each microsecond.

The 8051 microcontroller has 2 timers/counters called T0 and T1. As their names suggest, their main purpose is to measure time and count external events. Besides, they can be used for generating clock pulses to be used in serial communication, so called Baud Rate.

**Timer T0**

As seen in figure below, the timer T0 consists of two registers – TH0 and TL0 representing a low and a high byte of one 16-digit binary number.

**A black and white image of a timer

Description automatically generated**

Accordingly, if the content of the timer T0 is equal to 0 (T0=0) then both registers it consists of will contain 0. If the timer contains for example number 1000 (decimal), then the TH0 register (high byte) will contain the number 3, while the TL0 register (low byte) will contain decimal number 232.

**A close-up of a number

Description automatically generated**

Formula used to calculate values in these two registers is very simple:

TH0 × 256 + TL0 = T

Matching the previous example it would be as follows:

3 × 256 + 232 = 1000

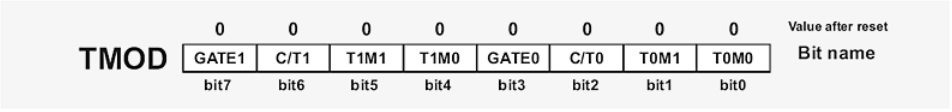
**A close-up of a number

Description automatically generated**

Since the timer T0 is virtually 16-bit register, the largest value it can store is 65 535. In case of exceeding this value, the timer will be automatically cleared and counting starts from 0. This condition is called an overflow. Two registers TMOD and TCON are closely connected to this timer and control its operation.

**TMOD Register (Timer Mode)**

The TMOD register selects the operational mode of the timers T0 and T1. As seen in figure below, the low 4 bits (bit0 - bit3) refer to the timer 0, while the high 4 bits (bit4 - bit7) refer to the timer 1. There are 4 operational modes and each of them is described herein.

****

Bits of this register have the following function:

* **GATE1** enables and disables Timer 1 by means of a signal brought to the INT1 pin

(P3.3):

* + 1 - Timer 1 operates only if the INT1 bit is set.
  + 0 - Timer 1 operates regardless of the logic state of the INT1 bit.
* **C/T1** selects pulses to be counted up by the timer/counter 1:
  + 1 - Timer counts pulses brought to the T1 pin (P3.5).
  + 0 - Timer counts pulses from internal oscillator.
* **T1M1, T1M0** These two bits select the operational mode of the Timer 1.

A table with numbers and symbols

Description automatically generated

* **GATE0** enables and disables Timer 1 using a signal brought to the INT0 pin (P3.2):
  + 1 - Timer 0 operates only if the INT0 bit is set.
  + 0 - Timer 0 operates regardless of the logic state of the INT0 bit.
* **C/T0** selects pulses to be counted up by the timer/counter 0:
  + 1 - Timer counts pulses brought to the T0 pin (P3.4).
  + 0 - Timer counts pulses from internal oscillator.
* **T0M1, T0M0** These two bits select the optional mode of the Timer 0.

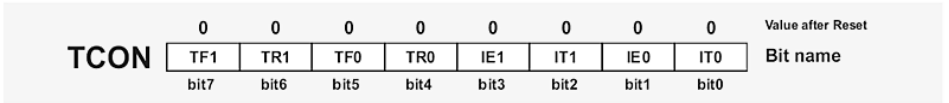
A table with numbers and letters

Description automatically generated

**Timer Control (TCON) Register**

TCON register is also one of the registers whose bits are directly in control of timer

operation.

Only 4 bits of this register are used for this purpose, while rest of them is used for interrupt control to be discussed later.

* TF1 bit is automatically set on the Timer 1 overflow.
* TR1 bit enables the Timer 1.
  + 1 - Timer 1 is enabled.
  + 0 - Timer 1 is disabled.
* TF0 bit is automatically set on the Timer 0 overflow.
* TR0 bit enables the timer 0.
  + 1 - Timer 0 is enabled.
  + 0 - Timer 0 is disabled.

**Timer 1**

Timer 1 is identical to timer 0, except for mode 3 which is a hold-count mode. It means that they have the same function, their operation is controlled by the same registers TMOD and TCON and both of them can operate in one out of 4 different modes.**A screenshot of a computer program

Description automatically generated**

3. **Lab exercise**

3.1 Write a C program to toggle the P1.5 bit continuously every 50ms. Use Timer 0, mode 1 (16bit) to make a delay function.

3.2. Write a C program to toggle all bits of P2 continuously every 500ms. Use Timer 1, mode 1 to make a delay time.

3.3. Write a C program to toggle of bit P2.7 with a frequency of 2500 Hz. Use Timer 1, mode 2.

Name: Student Code: Class: Lab:

1. Circuit
2. Algorithm flowchart
3. Code and explanation
4. Summary