# Timers

## Block Guide

Version #: **0.2**
Last revision date: **OUT 25, 2010**

# *Version Control*

**Table 1 – Version control**

| Revision | Date | Author | Comments | Revisors |
|---|---|---|---|---|
| 0.1 | 04/aug/2010 | Inácio Mendonça/Eloi Magalhães | Initial Version - Gramatically Revised | Felipe Prado Yonehara |

## Summary

# *Timer / Counter*

## *Preface*

The EMC08 has two 24-bit Timer/Counter registers: Timer 0 and Timer 1. The EMC08 has these two plus one more: Timer 2. The two first can be configured to operate either as timers or event counters and the third is a specific automotive aplication.

The Timer 0 and Timer 1 are 24bit counter register incremented every machine cycle. Thus, one can think of it as counting machine cycles.

These Timers 0 and 1 have two operating modes which are selected by bit-pairs (M1, MO) in TMOD. Refers to REGS section for TMOD/TCON registers for more details.

## *Conventions*

**Table 2 – Conventions**

| Convention | Description |
|---|---|
| Data memory | The memory for data storage |
| Program memory | The memory that contains a set of instructions |

## *1.1 Bibliography*

- Statement of the Work – S.O.W
  Version: 1.0
  Revision date: July 27, 2010

- Intel data sheet:
  MCS@51 MICROCONTROLLER
  FAMILY USER'S MANUAL

## *1.2 Acronyms and abbreviations*

**Table 3 – Acronyms and Abbreviations**

| Term | Meaning |
|---|---|
| ACG | Angle Clock Generator |
| ACR | Angle Counter Register |
| CLK | Clock |
| DFP | Digital Filter Clock Period |
| DFSEL | Digital Filter Selector |
| ECU | Engine Control Unit |
| ED | Edge Detector |
| EDSEL | Edge Digital Selector |
| FI | Fuel Injection |
| I/O | Input / Output |

| | |
|---|---|
| IC | Ignition Coil |
| MC | Machine Cycle |
| PDCF | Prescaler Digital Clock Filter |
| PDF | Programmable Digital Filter |
| PHT | Digital Flywheel Tooth Sensor |
| REGS | Registers |
| ROM | Read Only Memory |
| RPM | Rotations Per Minute |
| SYNC | Synchronizer |
| TACP | Timer Angle Clock Period |

## *1.3 Glossary*

Terms and definitions for Timer / Counter:

**Table 4 – Glossary**

| Term | Definition |
|---|---|
| Crankshaft | It is a part of the engine where the piston is connected. |
| Down-Counter | Binary counter declining. |
| Edge | Transition signal voltage. |
| Ignition Coil | Coil responsible for elevating the level of tension to activate the sails of the engine. |
| Microcode | Part of a program implemented in memory or in hardware. |
| Register | Data storage device. |
| Up-Counter | Binary counter growing. |

## *1.4 Introduction*

The microcontroller EMC08 has 3 timers / counters, calls of TIMER (timer 0, timer 1 and timer 2), being two on general purpose (timer 0 and 1) and an on specific purpose (timer 2).

The timers 0 and 1 can assume the timer function or counter depending on the configurations attributed to the same by the application (software).

The timer 2 has his functionality focused for the section automotive being used as an angle counter in a jagged wheel in the which lacks a tooth that through his occurrence allows to synchronize the counting previously stored in a register with the current counting obtained by a turn of the jagged wheel, being then that validated result and stored in a register, and through these data stored in the register makes possible the counting of turns in the motor of the vehicle (RPM) and it provides to the system (CPU) to evaluate the automobile is accelerating or slowing down so that of ownership of those registrations to increase or to reduce the flow of injection of combustible mixture and the speed of the ignition, that system FlyWheel is called.

Figure 1 shows the block diagram (top level) of the timers, they are prepared in all inputs and outputs, the arrows indicate whether the signal is unidirectional or bi-directional, many bits of each. This is a representation known as black box.

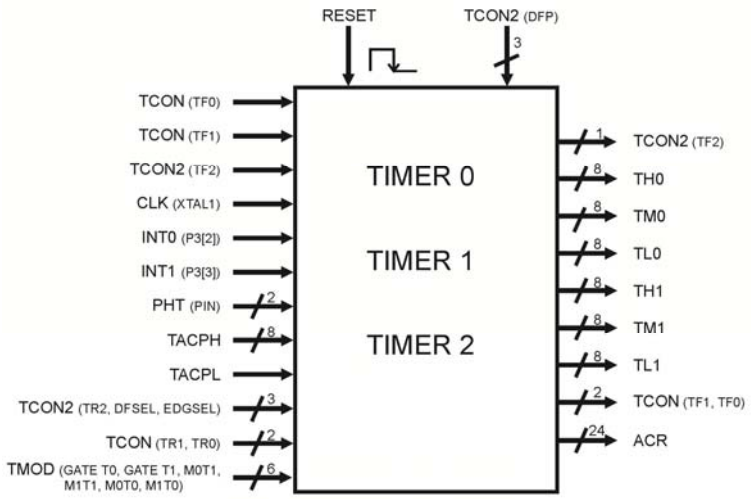



**Figure 1 – Top Level of Timers**

### *1.4.1 Overview*

Starting from the growth of the use electronics embedded on the surface mobile vehicle did necessary the creation and development of several devices for improvement of the acting and safety of those platforms, the necessity of a new device was evident. This device was called Flywheel.

He acts in the improvement of the motor acting optimizing his consumption and efficiency providing stability in his operation.

Through analogic sensor the obtained signs are converted in digital signs after have been processed, are sent for an Unit of Control of the motor (Engine Control Unit - ECU) that it makes the necessary corrections in the injection control of the fuel mixture and ignition speed.

It is a auto-adaptive system to monitor and recognizes the changes that happen in the motor and it compensates them automatically acting in the Map Base of Fuel, progress and air flow in ECU.

This document provides all the functions and settings of timers, registers and their associated memory locations.

Further, details will be discussed about the operation of timer 2, (which is a specific application automotive), all your settings and associated records.

### *1.4.2 Functional Description*

The operation of timer/counter 0 and 1 are identical, using the descriptions for both below worked.

There are two ways to activate the timer; The timer 0 leaves in low level the bit **GATETx** inside **TMOD** register and set a bit **TRx** from **TCON** register**.** After this, it will now operates as a counter updating the register values **TLx**, **TMx**, **THx** all 8-bit, and finally reaching the maximum value (FFFFFF hex), on the next cycle, an overflow will occur, setting the interrupt flag, **TFx** on **TCON** register (bits 7 and 5).

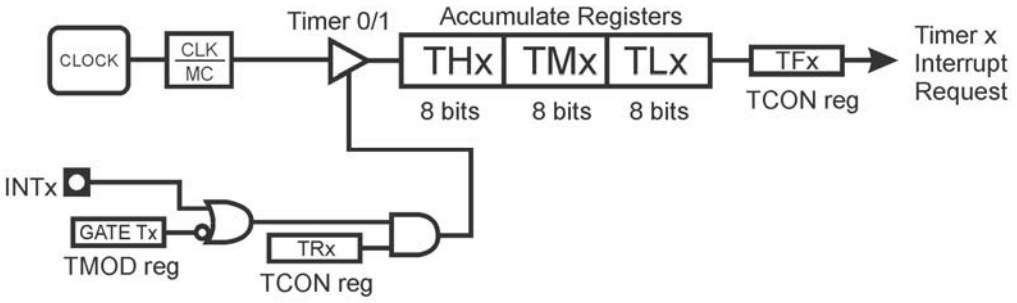



**Figure 2 - TIMER 0 or 1 Functional Circuit**

If the interrupt timers are enabled (register IE, bits: EA, ET2, ET1 e ET0), this may act in interrupts, otherwise whenever there is an overflow, the counter continues to count indefinitely until the flag TRx have been reset.

The operation of Timer 2 is specific for the application what it destine being composed of a mixer of analogical sign with digital sign, to follow it will be made a description in his operation way.

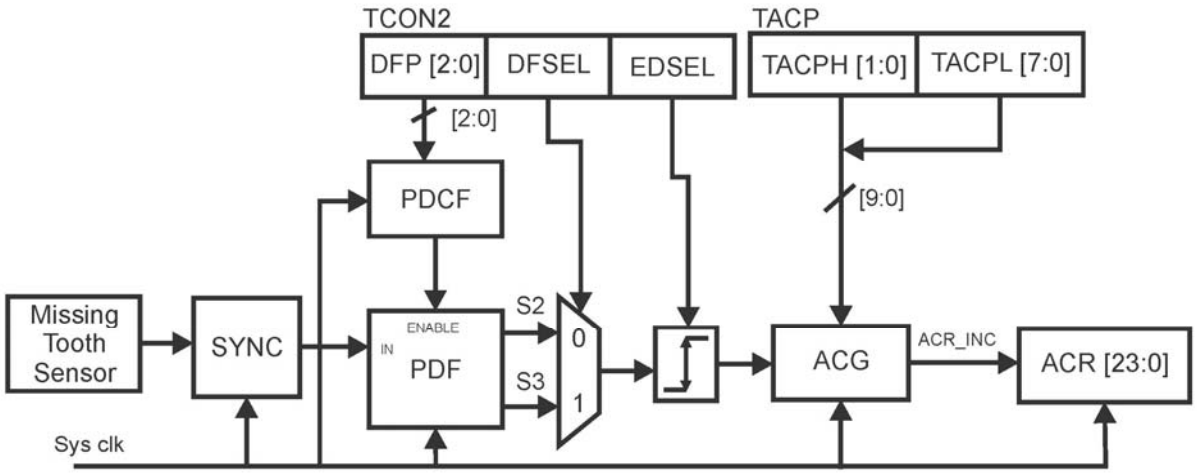



**Figure 3 - Block Diagram Timer 2**

The process of data collection done by the Timer 2 (Flywheel) is executed by sampling originating from of a sensor one analogical (inductive) in contact with the existent jagged wheel in the extremity of the tree of cranks (crankshaft) that sends pulses for a converter A/D that is processed by the Timer 2 and the pulse sequence stored in a 24 bits register (ACR) enabled and incapacitated by the sign acr_inc.

The received signal from analog sensor is submit for a synchronization through SYNC control block that has as purpose maintain the metastability of the signal sent for the filter PDF that is controlled by signal of the programmable filter PDCF, this being controlled by signal that comes DFP that to configure a frequency spliter of 3 bits with the purpose adapting the frequency of a coming signal of the analogic sensor, the frequency of the coming signal of oscillator.

The coming signal from PDCF filter, controls (enable) the PDF filter that sends the same to the next stage of the DFSEL selector, this stage is a sample selector the analog signal that is to still eliminate some instability existent happened in the system, that he does in two operation mode that defined for the application through the bit no. 1 from TCON2 register.

If the logical level of that bit is low (0) the selector DFSEL will select two samples same successive arrival of the sign of the filter PDF, otherwise, they will be rejected.

If the logical level of that bit goes 1 the selector DFSEL will select three samples same successive arrival of the sign of the filter PDF, otherwise, they will be rejected.

The sign will be given to the next stage, the border selector EDSEL that will choose the border of work of the next stage of ACG.

The stage of ACG will make a comparison with the coming signal from TACP register (10-bits) with the signal of the occurrence of the coming tooth of the sensor analogical.

The process of that comparison takes place with the entrance a die measured initial in a register TACP, after this when occurs a tooth the generator ACG will begin decrease the received data from TACP register in the end of counting coincides an occurrence from next tooth, the ACG generator update the ACR register (24-bits), when this is enabled by the signal acr_inc.

After updating the register ACR the generator ACG restarts the counting process for the next period of occurrence tooth, if the same happens before the finalization counting the obtained data is stored in the ACG generator and it will be decrement successively until the counting to conclude if on this exact moment there is an occurrence of the tooth the data is stored in ACG and starts again the countdown.

In case this counting arrives at the end and there was not the occurrence of tooth that data is stored in the ACG generator and increased each period of the angle clock to that there is the coincidence of finalization the counting with a tooth occurrence, like this the process will repeat in each period of angle clock. Sometimes confirming the counting with the tooth occurrence, other times being necessary to increase or to decrease that counting with the purpose of doing the finalization counting to coincide with the tooth occurrence.

The following illustration displays the functional diagram of Angle Clock Generator (ACG).

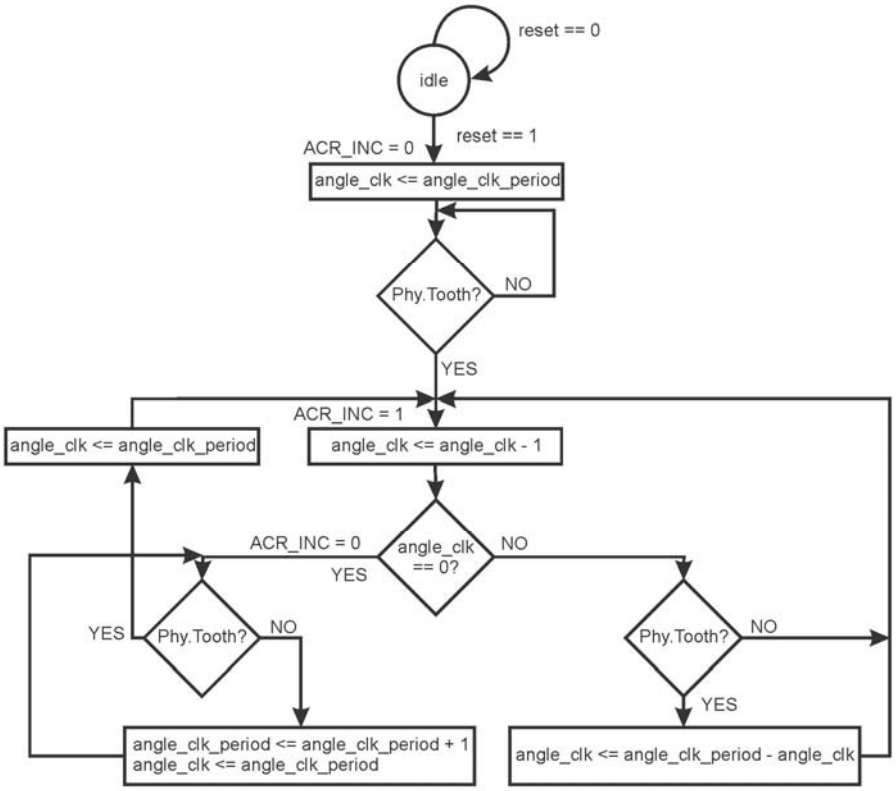



**Figure 4 - Microcode Algorithm**

### *1.4.3 Initialization Information*

Before starting the program, case you will use one of the timers, it is advisable initialize the registers **TLx**, **TMx**, **THx**, with necessary values or zero, to ensure that no one has any value due to a chain that has not been completely discharged, also known as trash.

The folowing table shows the values of registers after reset.

| Address / Offset | Register | Reset Value |
|---|---|---|
| D8h | TCON2 | x000 0001b |
| 88h | TCON | 0000 0000b |
| 89h | TMOD | 0x00 0x00b |
| 8Ch | TH0 | 0000 0000b |
| 8Eh | TM0 | 0000 0000b |
| 8Ah | TL0 | 0000 0000b |
| 8Dh | TH1 | 0000 0000b |
| 8Fh | TM1 | 0000 0000b |
| 8Bh | TL1 | 0000 0000b |
| BBh | TACPH | xxxx xx00b |
| BAh | TACPL | 0000 0000b |

### *1.4.4 Features*

#### *1.4.4.1 Timers 0 and 1*

These devices were designed only to act as timers. It can be activated externally or internally via software and its main features are:

- Clock Generator
- 24-bit Registers
- Up-Counter
- Down-Counter

#### *1.4.4.2 Timer 2*

The main characteristic of the timer 2 (Flywheel) is provide to ECU the possibility of:

- Regulate times of injection
- Regulate the ignition progress
- Control enrichment of the combustible mixture in acceleration
- Cut of fuel in the phase of I diminish of the motor
- Administration of the rotation of the motor in the slow march
- Limitation of the maximum rotation of the motor

## *1.5 Modes of operation*

### 1.5.1 Core-Related

Not applicable.

### *1.5.2 Mode 0*

Either Timer 0 and Timer 1 in Mode 0 are a 24-bit Counter. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag **TF1**. The counted input is enabled to the Timer when **TR1** = 1 and either GATE = 0 or INT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input **INT1**, to facilitate pulse width measurements.) TR1 is a control bit in the Special Function Register **TCON.GATE** is in **TMOD**.

The 24-Bit register consists of three 8 bits registers (**TH1/TM1/TL1**). Setting the run flag (TR1) won't clear the registers.

The operation mode 0 is the same for Timer 0 and Timer 1. The corresponding Timer 0 signals are **TR0/TF0/INT0/TH0/TM0/TL0**. There are two different GATE bits, one for Timer 1 (**TMOD.7**) and one for Timer 0 (**TMOD.3**).

### *1.5.3 Mode 1*

For this application, mode 1 is not enabled in microcontroller.

### *1.5.4 Mode 2*

For this application, mode 2 is not enabled in microcontroller.

### *1.5.5 Mode 3*

Either Timer 0 and Timer 1 in Mode 3 are a 24-bit Down Counter. As the count rolls over from all 0s to all 1s, it sets the Timer interrupt flag **TF1**. The counted input is enabled to the Timer when **TR1** = 1 and either **GATE** = 0 or **INT1** = 1. (Setting **GATE** = 1 allows the Timer to be controlled by external input **INT1**, to facilitate pulse width measurements.) **TR1** is a control bit in the Special Function Register **TCON.GATE** is in **TMOD**.

The 24-Bit register consists of three 8 bits registers (**TH1/TM1/TL1**). Setting the run flag (**TR1**) does not clear the registers. The operation mode 3 is the same for Timer 0 as for Timer 1. The corresponding Timer 1 signals are **TR1/TF1/INT1/TH1/TM1/TL1**. There are two different **GATE** bits, one for Timer 1 (**TMOD.7**) and Timer 0 (**TMOD.3**).

## *1.6 Signal Description*

### 1.6.1 External Signal Description

PHT Digital Flywheel Tooth sensor input to Timer 2. Digital signal genereted by analog sensor placed in to crankshaft.

### 1.6.2 Detailed Signal Descriptions

**Table 5 – Interface description**

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th><th>Reset</th></tr>
<tr><td rowspan="4">CLK</td><td rowspan="4">I</td><td colspan="2">Clock interface to work timer/counter.</td><td rowspan="4">1</td></tr>
<tr><td>State Meaning</td><td>Asserted: High Level of clock.<br>Negated: Low level of clock.</td></tr>
<tr><td>Timing</td><td>Assertion: Synchronous with external clock.<br>Negation: Synchronous with external clock.</td></tr>
<tr><td>INT0</td><td>I</td><td colspan="2">External on/off timer 0.</td><td>0</td></tr>
</table>

| Signal | I/O | Description | | Reset |
|---|---|---|---|---|
| | | **State Meaning** | Asserted: Turn on Timer 0 if GATE T0 = 0.<br>Negated: Turn off Timer 0. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| INT1 | I | External on/off timer 1. | | 0 |
| | | **State Meaning** | Asserted: Turn on Timer 1 if GATE T1 = 0.<br>Negated: Turn off Timer 1. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| PHT | I | Digital Flywheel Tooth sensor input to Timer 2. | | 0 |
| | | **State Meaning** | Asserted: Start the Process of Timer 2.<br>Negated: Stop the Process. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TACPH | I/O | Accumulator (2 bits) Msb of estimated value from Angle Clock | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TACPL | I/O | Accumulator (8 bits) Lsb of estimated value from Angle Clock Period. | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| ACR | O | Angle Clock Accumulator (3x8 bits) ACRL, ACRM, ACRL. | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |

| Signal | I/O | Description | | Reset |
|---|---|---|---|---|
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TCON | O | TCON Register to control Timers/Counter (bits TF1 and TF0) | | 0 |
| | | **State Meaning** | Asserted: Timer 0 and 1 overflow flag active.<br>Negated: Timer 0 and 1 overflow flag inactive. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TCON | I | TCON Register to control Timers/Counter Run (bits TR1 and TR0) | | 0 |
| | | **State Meaning** | Asserted: Run Timer 0 or 1.<br>Negated: Stop Timer 0 or 1. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TH1 | O | Timer 1 Accumulator Most Significant bits (8 bits). | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TM1 | O | Timer 1 Accumulator Medium Significant bits (8 bits). | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TL1 | O | Timer 1 Accumulator Low Significant bits (8 bits). | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TH0 | O | Timer 0 Accumulator Most Significant bits (8 bits). | | 0 |

| Signal | I/O | Description | | Reset |
|---|---|---|---|---|
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TM0 | O | Timer 0 Accumulator Medium Significant bits (8 bits). | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TL0 | O | Timer 0 Accumulator Low Significant bits (8 bits). | | 0 |
| | | **State Meaning** | Asserted: Not change.<br>Negated: Update value. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TMOD.3 (GATE T0) | I | Control bit to Run or Stop Timer 0. | | 0 |
| | | **State Meaning** | Asserted: if INT0 = 1 Timer 0 Run.<br>Negated: if TR0 = 1 Timer 0 Run. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TMOD.7 (GATE T1) | I | Control bit to Run or Stop Timer 1. | | 0 |
| | | **State Meaning** | Asserted: if INT1 = 1 Timer 0 Run.<br>Negated: if TR1 = 1 Timer 0 Run. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TMOD.0 (M0T0) | I | Timer 0 mode selector bit (M0). | | 0 |
| | | **State Meaning** | Asserted: Ask item 1.7.1.1<br>Negated: Ask item 1.7.1.1 | |

| Signal | I/O | Description | | Reset |
|---|---|---|---|---|
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TMOD.1<br>(M1T0) | I | Timer 0 mode selector bit (M1). | | 0 |
| | | **State Meaning** | Asserted: Ask item 1.7.1.1<br>Negated: Ask item 1.7.1.1 | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TMOD.4<br>(M0T1) | I | Timer 1 mode selector bit (M0). | | 0 |
| | | **State Meaning** | Asserted: Ask item 1.7.1.1<br>Negated: Ask item 1.7.1.1 | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TMOD.5<br>(M1T1) | I | Timer 1 mode selector bit (M1). | | 0 |
| | | **State Meaning** | Asserted: Ask item 1.7.1.1<br>Negated: Ask item 1.7.1.1 | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TCON2.5<br>(TF2) | O | Timer 2 overflow flag. | | 0 |
| | | **State Meaning** | Asserted: Overflow occur<br>Negated: Not occur overflow | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TCON2.6<br>(TR2) | I | Timer 2 run control bit. | | 0 |
| | | **State Meaning** | Asserted: Turn on timer 2.<br>Negated: Turn off timer 2. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TCON2.1 | I | Digital Filter Sampling Selection | | 0 |

| Signal | I/O | Description | | Reset |
|---|---|---|---|---|
| (DFSEL) | | **State Meaning** | Asserted: Output sampling S3 selected.<br>Negated: Output sampling S2 selected. | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |
| TCON2.0 (EDGESEL) | I | Rise-Fall Edge selection | | 1 |
| | | **State Meaning** | Asserted: Rise edge selection<br>Negated: Fall edge selection | |
| | | **Timing** | Assertion: Implemented in future.<br>Negation: Implemented in future. | |

## *1.7 Special Consideration to each block*

### *Memory map*

**Table 6 – Memory map**

| Address/Offset | Register | Access | Reset Value | Section |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

### *Memory map and register definition*

The timers have three associated registers for control and configuration, **TMOD**, **TCON** and **TCON2** that are located at the addresses specified in the statement that follows, these registers have a size of 8 bits.

### *1.7.1 Register Description*

#### *1.7.1.1 TMOD*

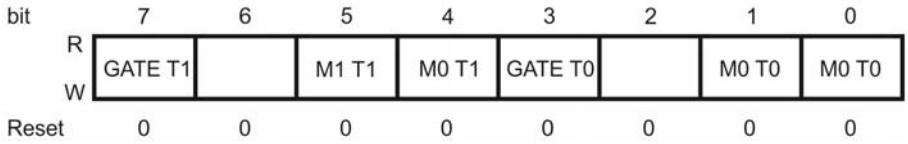



**Figure 5 - TMOD Register**

| **Field** | **Description** | | | |
|---|---|---|---|---|
| GATE T1 | When TR1 (in **TCON**) is set and **GATE** T1=1, Timer/Counter 1 will run only while **INT1** pin is high (hardware control). When **GATE** T1=0, Timer/Counter1 will run only while **TR1**=1 (software control). | | | |
| M[5:4] T1 | Timer 1 Mode Selector bit. | | | |
| | M1 | M0 | Operating Mode | Description |
| | 0 | 0 | 0 | 24-bit up Timer |
| | 0 | 1 | 1 | Reserved |
| | 1 | 0 | 2 | Reserved |
| | 1 | 1 | 3 | 24-bit down Timer |
| **Field** | **Description** | | | |
| GATE T0 | When **TR0** (in **TCON**) is set and **GATE** T0=1, Timer 0 will run only while **INT0** pin is high (hardware control). When **GATE** T0=0, Timer 0 will run only while **TR0**=1 (software control). | | | |
| M[1:0] | Timer 0 mode select bit | | | |
| | M1 | M0 | Operating Mode | Description |
| | 0 | 0 | 0 | 24-bit up Timer |
| | 0 | 1 | 1 | Reserved |
| | 1 | 0 | 2 | Reserved |
| | 1 | 1 | 3 | 24-bit down Timer |

### *1.7.1.2 TCON*

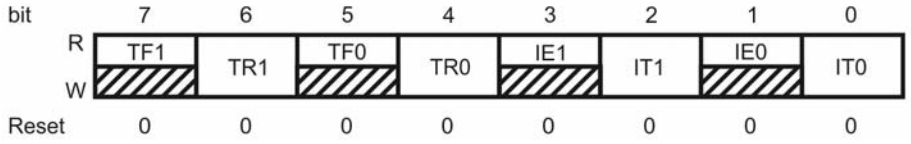



**Figure 6 - TCON Register**

| Field | Description |
|---|---|
| TF1 | Timer 1 overflow flag. Set by hardware when the Timer 1 overflows. Cleared as processor vectors to the interrupt service routine. |
| TR1 | Timer 1 run control bit. Set/cleared by software to turn Timer 1 ON/OFF. |
| TF0 | Timer 0 overflow flag. Set by hardware when the Timer 0 overflows. Cleared as processor vectors to the interrupt service routine. |
| TR0 | Timer 0 run control bit. Set/cleared by software to turn Timer 0 ON/OFF. |
| IE1 | External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed. |
| IT1 | Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt. |
| IE0 | External Interrupt 0 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed. |
| IT0 | Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt. |

### *1.7.1.3 TCON2*

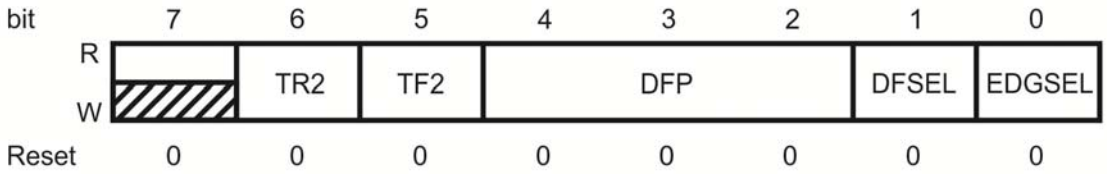
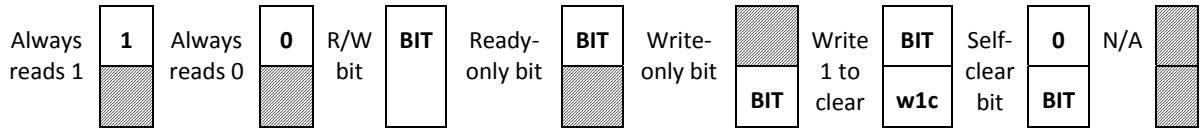



**Figure 7 - TCON2 Register**

| Field | Description |
|---|---|
| TCON2 [7] | Reserved |
| TR2 | Timer 2 run control bit. Set/cleared by software to turn Timer 2 ON/OFF . |
| TF2 | Timer 2 overflow flag. Set by hardware when the Timer 2 overflows. Cleared as processor vectors to the interrupt service routine. |
| DFP [4:2] | Digital Filter Clock Period. |
| DFSEL | Digital Filter Sampling Selection |
| | [0] DF output Sampling S2 selected |
| | [1] DF output Sampling S3 selected |
| EDGSEL | Rise-Fall Edge selection |
| | [0] Fall edge selection |
| | [1] Rise edge selection |

#### *1.7.1.4 Register Summary*




| Convention | Description |
|---|---|
| | Depending on its placement in the read or write row, indicates that the bit is not readable or not writeable. |
| FIELDNAME | Identifies the field. Its presence in the read or write row indicates that it can be read or written. |
| **Register Field Types** | |
| R | Read only. Writing this bit has no effect. |
| W | Write only. |
| R/W | Standard read/write bit. Only software can change the bit´s value (other than a hardware reset). |
| rwm | A read/write bit that may be modified by a hardware in some fashion other than by a reset. |
| w1c | Write one to clear. A status bit that can be read, and is cleared by writing a one. |
| self-clearing bit | Writing a one has some effect on the module, but it always reads as zero. |
| **Reset Values** | |
| 0 | Resets to zero. |
| 1 | Resets to one. |
| -- | Undefined at reset. |
| u | Unaffected by reset. |
| [*signal_name*] | Reset value is determined by polarity of indicated signal. |

### *Extra Information*

Not applicable. No Extra Information is needed in this version.

### *Initialization Information*

Either Timer 0 and Timer 1 in Mode 0 are a 24-bit Counter. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or INT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input INT1, to facilitate pulse width measurements.) TR1 is a control bit in the Special Function Register TCON.GATE is in TMOD.

Either Timer 0 and Timer 1 in Mode 3 are a 24-bit Down Counter. As the count rolls over from all 0s to all 1s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or INT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input INT1, to facilitate pulse width measurements.) TR1 is a control bit in the Special Function Register TCON.GATE is in TMOD.

Timer 2 is turn on / off something seting bit **TR2** in register **TCON2.**

### *Application Information*

Starting from the growth of the use electronics embedded on the surface mobile vehicle did necessary the creation and development of several devices for improvement of the acting and safety of those platforms, the necessity of a new device was evident. This device was called Flywheel.

He acts in the improvement of the motor acting optimizing his consumption and efficiency providing stability in his operation.

Through analogic sensor the obtained signs are converted in digital signs after have been processed, are sent for an Unit of Control of the motor (Engine Control Unit - ECU) that it makes the necessary corrections in the injection control of the fuel mixture and ignition speed.

It is a auto-adaptive system to monitor and recognizes the changes that happen in the motor and it compensates them automatically acting in the Map Base of Fuel, progress and air flow in ECU.

This document provides all the functions and settings of timers, registers and their associated memory locations.

Further, details will be discussed about the operation of timer 2, (which is a specific application automotive), all your settings and associated records.