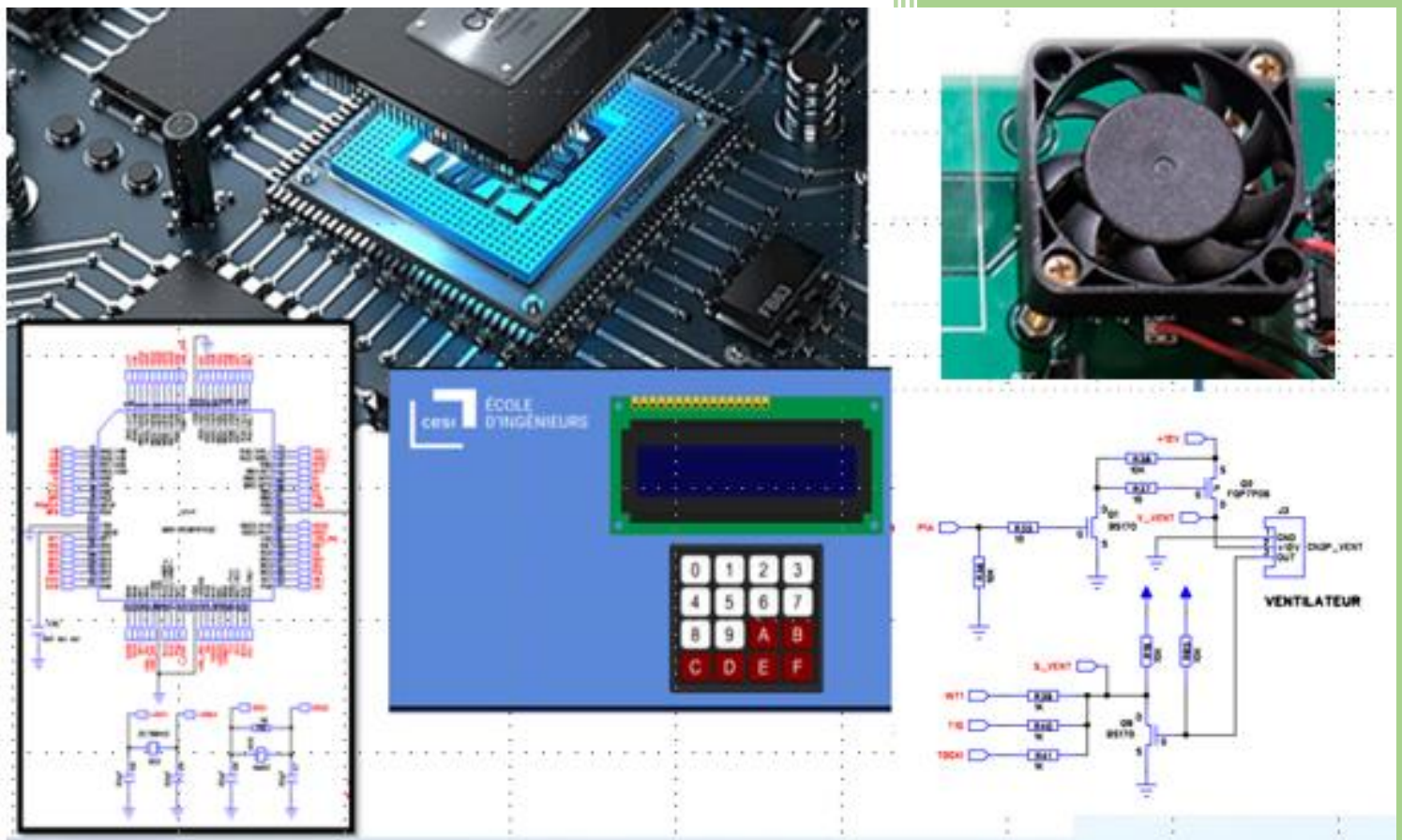




CAMPUS
D'ENSEIGNEMENT SUPÉRIEUR
ET DE FORMATION PROFESSIONNELLE

2022

TP5 FISA 26 Microcontrôleur



Nicolas ANTINI 2021:
FISA 25 Langage C bas niveau pour Microcontrôleur.

ANTINI Nicolas

CESI

01/01/2022

TP N°5 MICROCONTROLEUR PIC 18F87K22

Mise en œuvre des TIMERS.




Objectifs du TP :

Mesure de la durée d'un signal

Mesure de sa fréquence.


Gestion d'une sortie PWM

Expérimentation n°1:

-  Alimenter la turbine qui génère un signal (impulsions en retour).
-  Mesurer la durée de l'état haut de l'entrée S_VENT broche RB1.
-  Afficher sur l'afficheur 2 lignes 16 caractères.

__/__/20__ h__
S_VENT=0000 ms

Expérimentation n°2:

-  Mesurer la fréquence du signal S_VENT broche RB1 en utilisant le TIMER 0 en mode compteur.

(Entrée T0CKI). Afficher sur l'afficheur 2 lignes 16 caractères.

__/__/20__ h__
SVENT=0000 Hz



❏ **Expérimentation n°3:**

- ❏ Générer un signal PWM 10 KHz(rapport cyclique variable), BP1 augmente le rapport cyclique, BP2 le réduit.

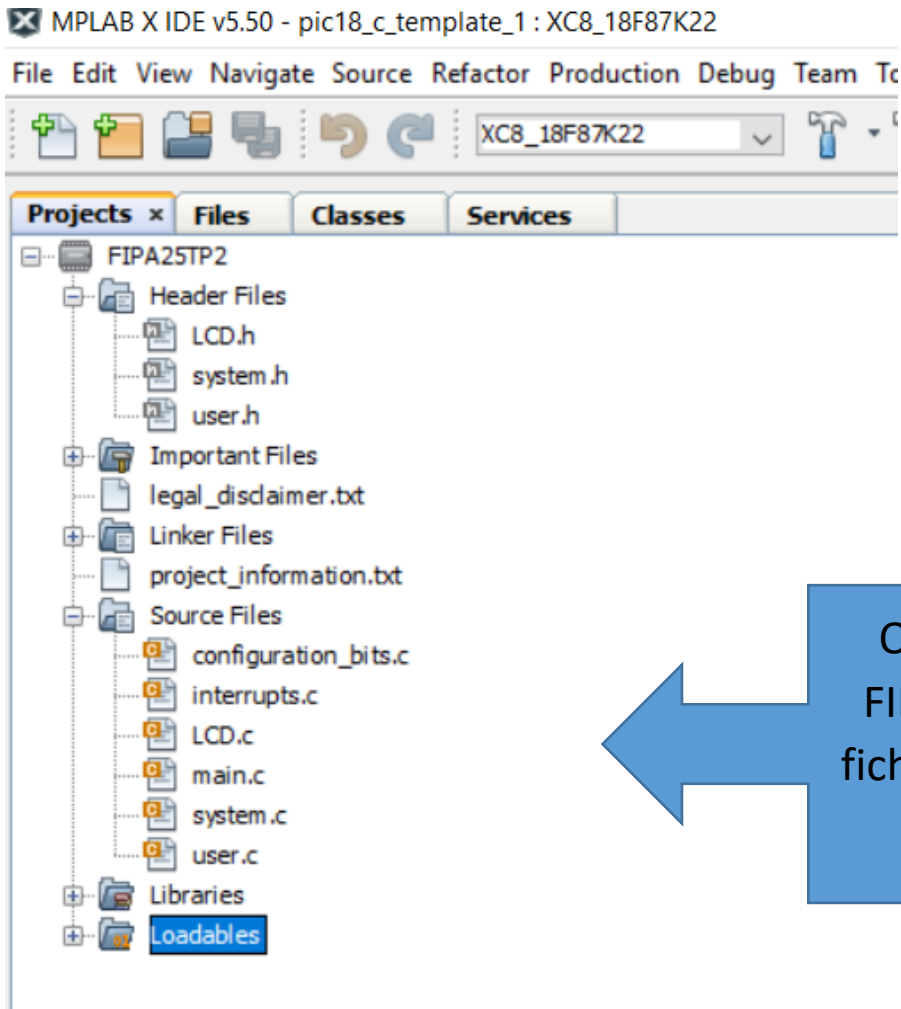
__/__/20__ h__
RCY= 000 %

❏ **Expérimentation N°4 :**

- ❏ Gérer l'horloge temps réel pour inscrire sur la ligne 1 de l'afficheur : La date (11/10/2021) et l'heure (10h16). L'utilisation des touches '+' et – permet de balayer les 3 types de mesures S_VENT en ms, SVENT en Fréquence, RCY...

19/11/2021 15h21
RCY= 000 %

Créer le projet FIPA25TP5 à partir du TP4.



On intègre à
FIPA25TP5 les
fichiers Timer et
RTC

user.h regroupe les entêtes de mes fonctions.

Ex : void Init(void) ;

system.h regroupe les entêtes des fonctions spécifiques au système.



Annexes :

PIC18F87K22 : Présentation détaillée du fonctionnement des TIMERS 0, 1, 2 et 3.

1) TIMER 0

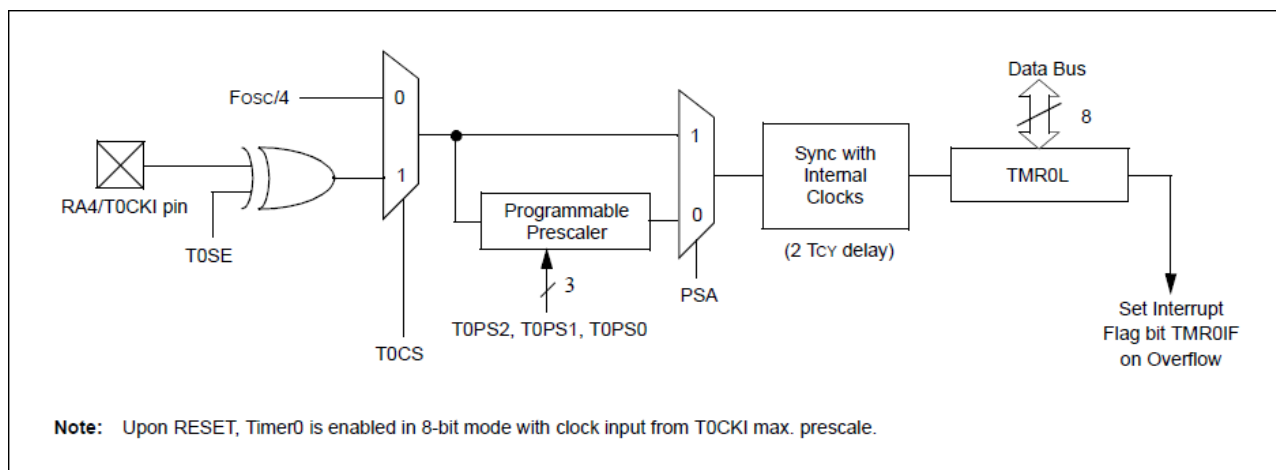
Le timer 0 est implémenté identiquement sur l'ensemble des familles :

Il présente les caractéristiques suivantes :

- X Timer 8 ou 16 bits. Le mode de fonctionnement (8 ou 16 bits) est sélectionnable à l'aide du bit **T08BIT** du registre **TOCON** (**Mode 8 bits : T08BIT = 1, Mode 16 bits : T08BIT = 0**).
- X Registres de comptage **TMR0L** et **TMR0H** en accessibles en lecture/écriture.
- X Prédiviseur 8 bits programmable à l'aide des bits **T0PS[2..0]** et **PSA** du registre **TOCON**.
- X L'horloge peut être interne (timer) ou externe (compteur). Dans ce dernier cas, le front de comptage est configurable. Ces configurations sont à faire dans le registre **TOCON** :
 - ✓ T0CS = 0 : Timer (horloge interne)
 - ✓ T0CS = 1 : Compteur (horloge externe sur RA4/T0CKI)

T0SE = 1 : Incréméntation sur front descendant
T0SE = 0 : Incréméntation sur front montant
- X Une interruptions de type **overflow (T0IF)** peut être déclenchée. Elle est déclenchée par le passage de **0xFF à 0x00 en mode 8 bits** ou par le passage de **0xFFFF à 0x0000 en mode 16 bits**.

Schéma bloc du Timer 0 en mode 8 bits



Registres associés au timer 0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TMR0L	Timer0 Register, Low Byte								50
TMR0H	Timer0 Register, High Byte								50
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	50
TRISA	RA7 ⁽¹⁾	RA6 ⁽¹⁾	RA5	RA4	RA3	RA2	RA1	RA0	52

Legend: Shaded cells are not used by Timer0.

Note 1: PORTA<7:6> and their direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as '0'.

Rappel : Compteur (horloge externe sur RA4/T0CKI).



2) Timer 1

Caractéristiques du timer TMR1 :

- X Timer **16 bits**. Le registre de comptage est accessible par les deux registres 8 bits **TMR1H et TMR1L**.
- X Registres de comptage accessibles en **lecture/écriture** (R/W).
- X Horloge **interne (mode timer) ou externe (mode compteur)**.
- X Interruption de type **Overflow (TMR1IF)** au passage de 0xFFFF à 0x0000.
- X Remise à 0 possible à partir d'un module CCP.

Schéma bloc du Timer 1

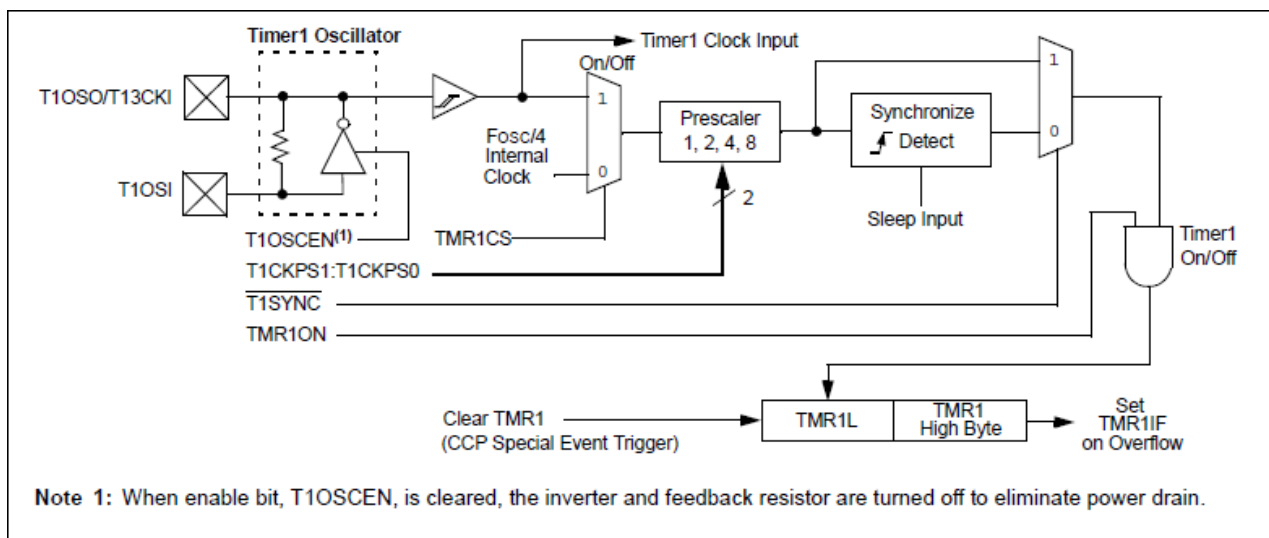
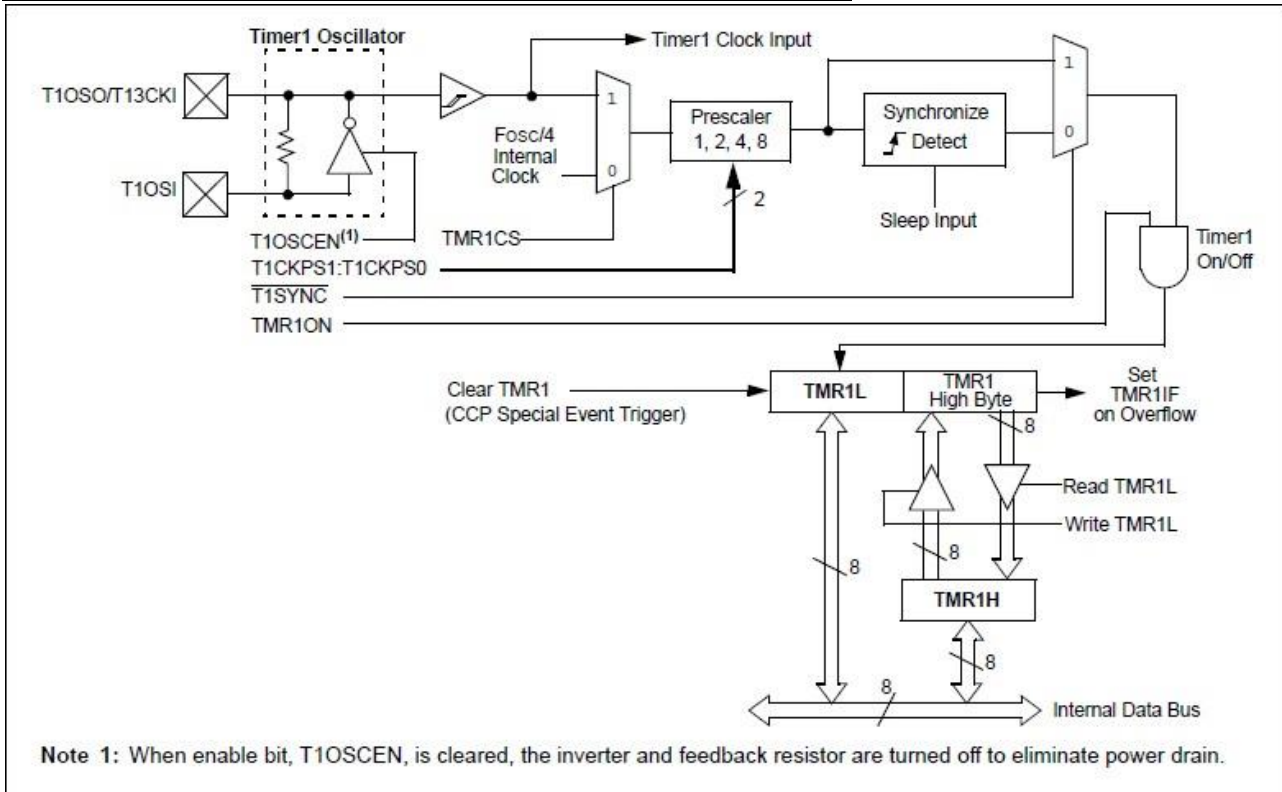


Schéma bloc du Timer 1 sur des opérations R/W 16 bits



Fonctionnement du TIMER1

Ce timer peut fonctionner selon les trois modes suivants :

- X Timer (sur l'horloge interne)
- X Compteur synchrone
- X Compteur asynchrone

Le mode de fonctionnement est choisi grâce au bit **TMR1CS** (Registre **T1CON**).

- X **TMR1CS = 0** : Mode timer, TMR1 est incrémenté à chaque cycle d'instruction
- X **TMR1CS = 1** : Mode compteur, TMR1 est incrémenté chaque **front montant sur l'entrée RC0/T1CKI** ou par l'oscillateur externe si celui-ci est utilisé.

Si l'oscillateur externe est utilisé (**T1OSCEN = 1**), les lignes **RC1/T1OSI** et **RC0/T1OSO** deviennent des entrées.

Le timer 1 peut être remis à 0 par un événement généré par le module CCP (voir fiche modules CCP).

Remarques sur l'accès aux registres de comptage

Lorsque le **bit RD16 est positionné à 1**, l'accès 16bits est activé : une lecture de TMR1L génère la sauvegarde immédiate de TMR1H : il est donc inutile de vérifier s'il n'y a pas eu de changement de TMR1H entre la lecture du poids faible et celle du poids fort. Cette sécurité est aussi active en écriture.

L'interruption associée au timer 1 est **TMR1IF**, le bit de masquage est **TMR1IE** et la configuration de la priorité de l'IT se fait grâce à **TMR1IP**.



Il ne faut pas oublier d'activer les interruptions périphériques en plaçant **PEIE à 1**.

Le registre T1CON

	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN \bar{C}	TMR1CS	TMR1ON
	bit 7							bit 0
bit 7	RD16: 16-bit Read/Write Mode Enable bit 1 = Enables register read/write of Timer1 in one 16-bit operation 0 = Enables register read/write of Timer1 in two 8-bit operations							
bit 6	T1RUN: Timer1 System Clock Status bit 1 = Device clock is derived from Timer1 oscillator 0 = Device clock is derived from another source							
bit 5-4	T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits 11 = 1:8 Prescale value 10 = 1:4 Prescale value 01 = 1:2 Prescale value 00 = 1:1 Prescale value							
bit 3	T1OSCEN: Timer1 Oscillator Enable bit 1 = Timer1 oscillator is enabled 0 = Timer1 oscillator is shut off The oscillator inverter and feedback resistor are turned off to eliminate power drain.							
bit 2	T1SYN\bar{C}: Timer1 External Clock Input Synchronization Select bit <u>When TMR1CS = 1:</u> 1 = Do not synchronize external clock input 0 = Synchronize external clock input <u>When TMR1CS = 0:</u> This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.							
bit 1	TMR1CS: Timer1 Clock Source Select bit 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge) 0 = Internal clock (Fosc/4)							
bit 0	TMR1ON: Timer1 On bit 1 = Enables Timer1 0 = Stops Timer1							





Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TMR1L	Timer1 Register, Low Byte								50
TMR1H	Timer1 Register, High Byte								50
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	50

Legend: Shaded cells are not used by the Timer1 module.

Note 1: These bits are unimplemented on 28-pin devices; always maintain these bits clear.



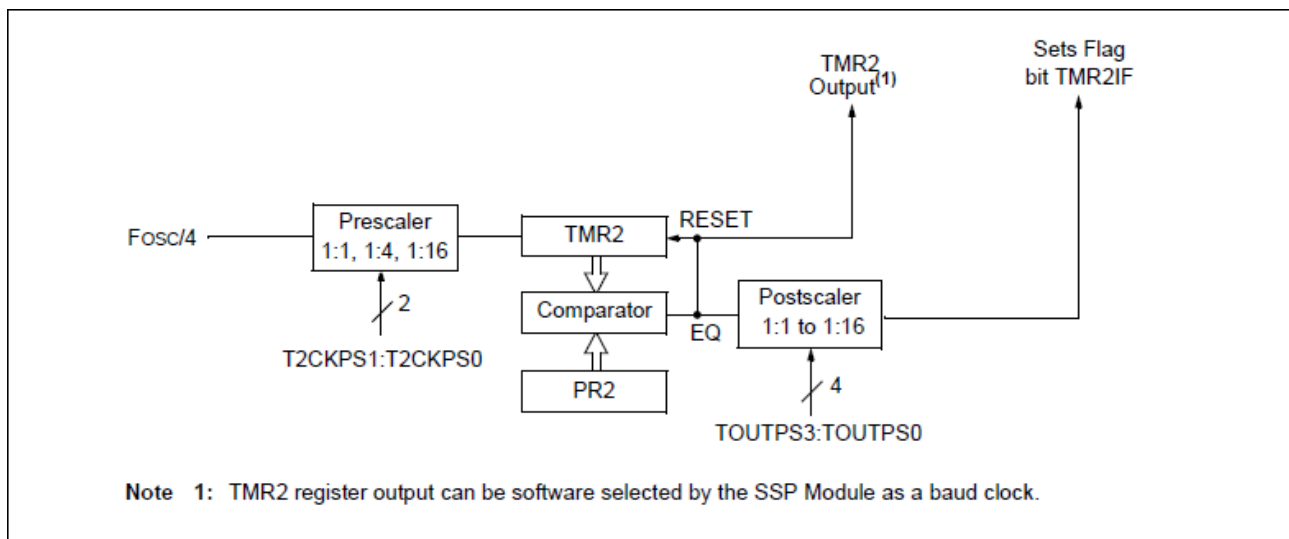
3) Le timer 2

Là encore, ce timer est implémenté à l'identique sur les PIC 18Fxx2 et 18Fxx20.

Ses principales caractéristiques sont les suivantes :

- X Timer **8 bits** : registre **TMR2**.
- X Associé à un « **Period register** » **8 bits** : **PR2**.
- X Registres TMR2 et PR2 accessibles en lecture / Ecriture.
- X Prédiviseur programmable : divisions par 1, 4 ou 16.
- X Postdiviseur programmable : facteur de division compris entre 1 et 16.
- X Interruption (**TMR2IF**) déclenchée lorsque **TMR2 = PR2**.

Schéma bloc du Timer 2



Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TMR2	Timer2 Register								50
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	50
PR2	Timer2 Period Register								50

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

Note 1: These bits are unimplemented on 28-pin devices; always maintain these bits clear.



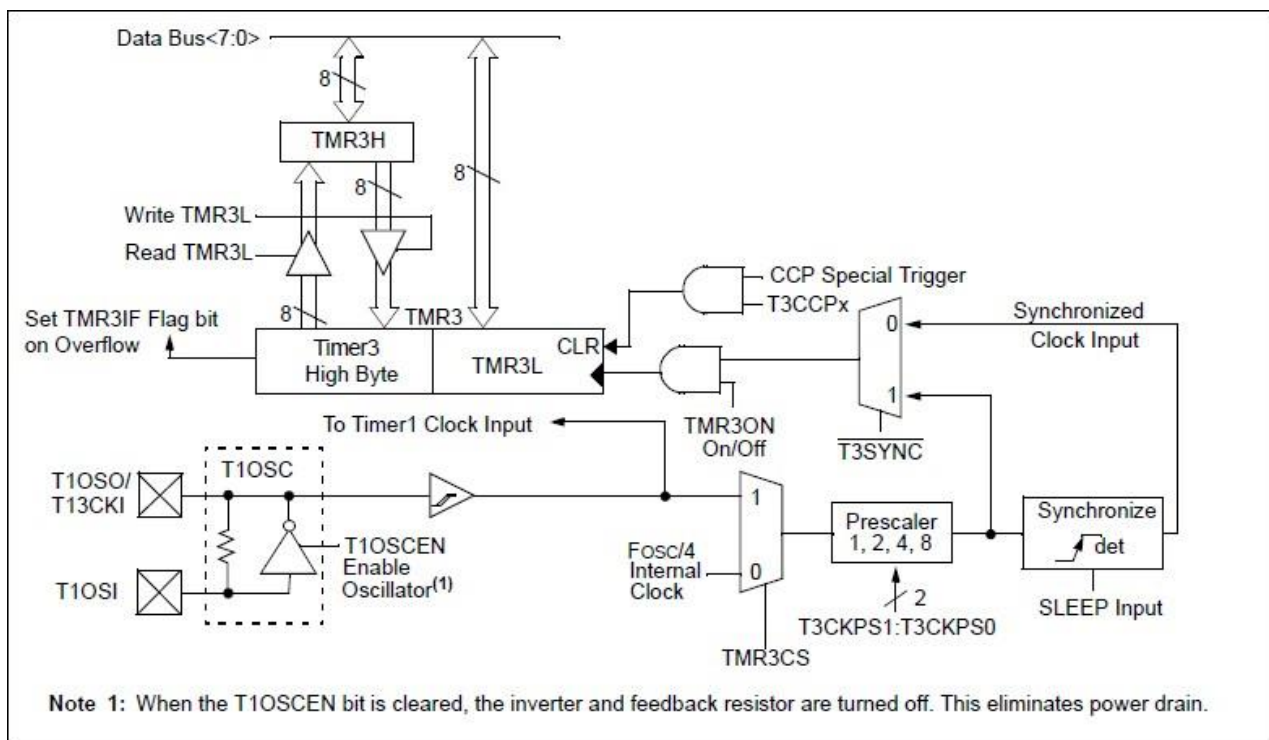
Le registre T2CON

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6-3 **T2OUTPS3:T2OUTPS0:** Timer2 Output Postscale Select bits
 0000 = 1:1 Postscale
 0001 = 1:2 Postscale
 .
 .
 .
 1111 = 1:16 Postscale
- bit 2 **TMR2ON:** Timer2 On bit
 1 = Timer2 is on
 0 = Timer2 is off
- bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits
 00 = Prescaler is 1
 01 = Prescaler is 4
 1x = Prescaler is 16

4) Timer 3

Le timer 3 est quasiment identique au timer1 : **T3CON** : Voir **T1CON**. Ils partagent un éventuel oscillateur externe.



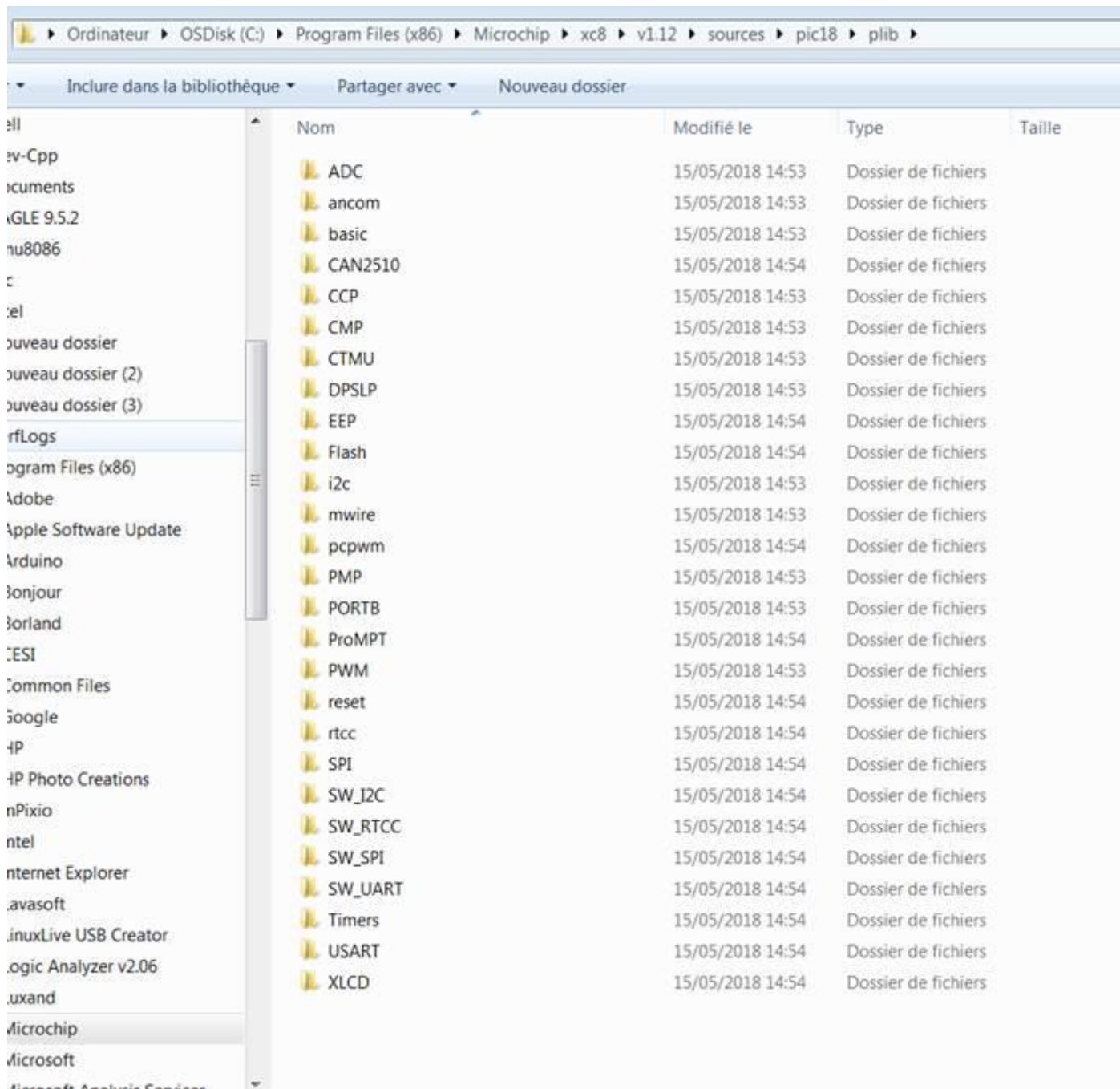
Gestion de l'Horloge Temps Réel :

Utilisation des Librairies RTCC et TIME.

Dans votre TP vous devez utiliser les directives suivantes:



















#include<Rtcc.H>

#include <Time.H>



Vous disposez du coup des fonctions constructeur pour la gestion du RTCC.

Dans `rtcc.c`

 <code>RtccInitClock.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccReadAlarmDate.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccReadAlarmTime.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccReadAlarmTimeDate.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccReadDate.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccReadTime.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccReadTimeDate.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccSetAlarmRpt.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccSetAlarmRptCount.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccSetCalibration.c</code>	03/12/2012 03:56	Fichier C	3 Ko
 <code>RtccSetChimeEnbl.c</code>	03/12/2012 03:56	Fichier C	2 Ko
 <code>RtccWriteAlarmDate.c</code>	03/12/2012 03:56	Fichier C	3 Ko
 <code>RtccWriteAlarmTime.c</code>	03/12/2012 03:56	Fichier C	3 Ko
 <code>RtccWriteAlarmTimeDate.c</code>	03/12/2012 03:56	Fichier C	3 Ko
 <code>RtccWriteDate.c</code>	03/12/2012 03:56	Fichier C	4 Ko
 <code>RtccWriteTime.c</code>	03/12/2012 03:56	Fichier C	4 Ko
 <code>RtccWriteTimeDate.c</code>	03/12/2012 03:56	Fichier C	4 Ko
 <code>RtccWrOn.c</code>	03/12/2012 03:56	Fichier C	2 Ko

Dans **`RtccReadTimeDate`** vous pouvez instantanément récupérer les 2 infos.

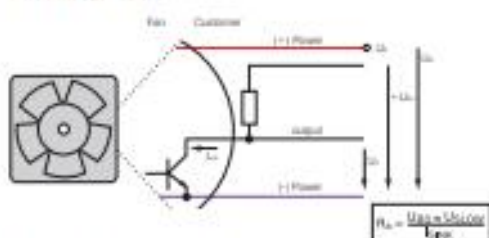
Ne pas oublier d'utiliser `RtccInitClock()` ;

Sensor signal /2 "tacho"



- Speed-proportional rectangular pulse for external speed monitoring of fan motor
- 2 pulses per revolution / 6 pulses per revolution with TURBOFANS.
- Open-Collector signal output
- Extremely wide operating voltage range (5 ... 60 V)
- Easy adaptation to user interface
- Connection via separate lead
- The sensor signal also serves as a major comparison variable for setting and maintaining the desired speed for interactive or controlled cooling with one or several interconnected fans.

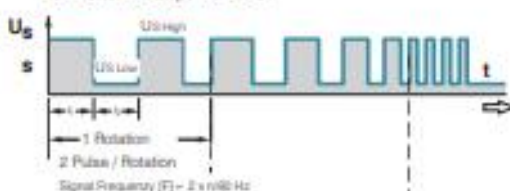
Electrical connection



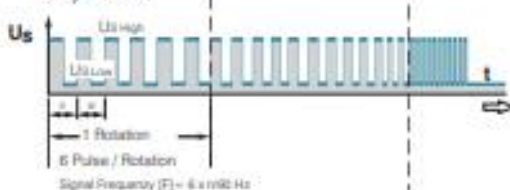
All voltages measured to ground.
External load resistance R_L / U_s / U_{sH} required.

Signal output voltage

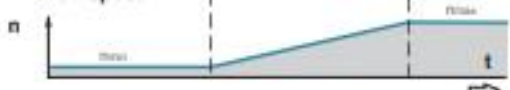
all models except TD-Fans



only TD-Fans



Fan speed



Signal data

Type	Sensor signal U_{sH}	Condition: I_{sH}	Sensor signal U_{sL}	Condition: I_{sL}	Sensor operating voltage U_s	Peak sink current I_{sHmax}
250	≤ 0.4	≤ 2	30	0	≤ 30	2
400 F	≤ 0.4	1	30	0	≤ 30	≤ 2
400	≤ 0.4	1	30	0	≤ 30	≤ 2
412 J	≤ 0.4	2	30	0	≤ 30	≤ 4
414 J	≤ 0.4	2	30	0	≤ 30	≤ 4
500 F	≤ 0.4	1	30	0	≤ 30	≤ 2
600 F	≤ 0.4	1	30	0	≤ 30	≤ 2
620	≤ 0.4	2	30	0	≤ 30	≤ 4
600 N	≤ 0.4	2	30	0	≤ 30	≤ 4
800 J	≤ 0.4	2	30	0	≤ 30	≤ 4
700 F	≤ 0.4	2	30	0	≤ 30	≤ 4
8400 N	≤ 0.4	2	28	0	≤ 28	≤ 4
8300	≤ 0.4	2	30	0	≤ 30	≤ 4
8200 J	≤ 0.4	2	30	0	≤ 30	≤ 4
3400 N	≤ 0.4	2	28	0	≤ 28	≤ 4
3330	≤ 0.4	2	30	0	≤ 30	≤ 4
3200 J	≤ 0.4	2	30	0	≤ 30	≤ 4
4400 F	≤ 0.4	2	30	0	≤ 30	≤ 4
4300 N	≤ 0.4	2	30	0	≤ 30	≤ 4
4300	≤ 0.4	2	30	0	≤ 30	≤ 4
4400	≤ 0.4	2	30	0	≤ 30	≤ 4
4212	≤ 0.4	2	30	0	≤ 30	≤ 4
4214	≤ 0.4	2	30	0	4-30	≤ 4
4218	≤ 0.4	2	30	0	4-30	≤ 4
4100 N	≤ 0.4	2	30	0	4-30	≤ 4
DW 4100	≤ 0.4	2	30	0	≤ 30	≤ 4
5200 N	≤ 0.4	2	30	0	4-30	≤ 4
DW 5200	≤ 0.4	2	30	0	≤ 30	≤ 4
5112 N	≤ 0.4	2	15	0	≤ 5	≤ 20
5114 N	≤ 0.4	2	80	0	≤ 60	≤ 20
5118 N	≤ 0.4	2	80	0	≤ 60	≤ 20
7112 N	≤ 0.4	2	80	0	≤ 60	≤ 20
7114 N	≤ 0.4	2	30	0	≤ 30	≤ 20
7118 N	≤ 0.4	2	80	0	≤ 60	≤ 20

Signal data	Sensor signal Us _{in}	Conditions Is	Sensor signal Us _{in}	Conditions Is	Sensor operating voltage Us	Power sink current Is _{max}
Type	V DC	mA	V DC	mA	V DC	mA
6224 N	≤ 0.4	8	30	0	≤ 30	≤ 20
6248 N	≤ 0.4	8	60	0	≤ 30	≤ 20
DN 6200	≤ 0.4	2	30	0	≤ 60	≤ 20
6400	≤ 0.4	2	60	0	≤ 60	≤ 20
RL 48	≤ 0.4	2	28	0	4–30	≤ 4
RL 65	≤ 0.4	2	30	0	≤ 30	≤ 4
RL 90 N	≤ 0.4	2	30	0	≤ 30	≤ 4
RLF 100	≤ 0.4	2	30	0	≤ 30	≤ 4
RG 90 N	≤ 0.4	2	30	0	≤ 30	≤ 4
RG 125 N	≤ 0.4	2	30	0	≤ 30	≤ 4
RG 160 N	≤ 0.4	2	30	0	≤ 30	≤ 20
REF 100	≤ 0.4	2	30	0	≤ 30	≤ 4

Attention:

With these fan options, deviations in regard to temperature range, voltage range and power consumption are possible compared with standard fan data.