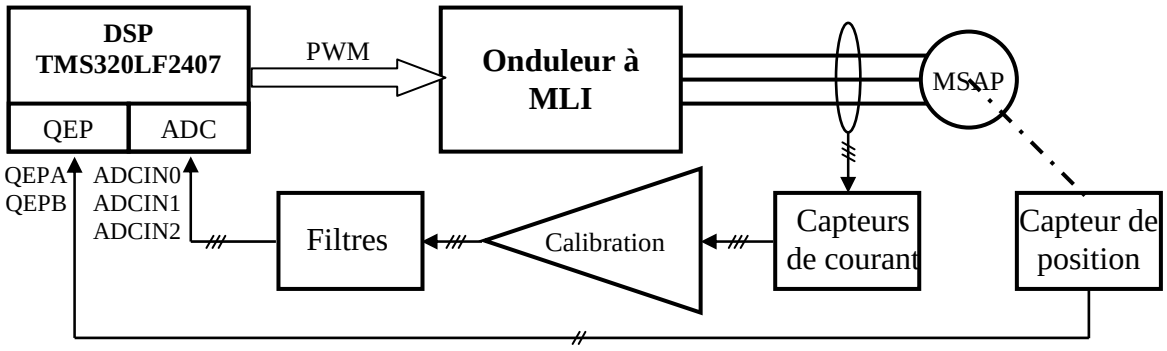
Médian MC43

Exercice 1 : Programmation d'un variateur industriel

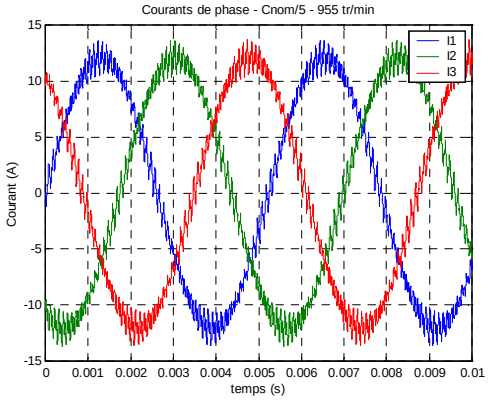
Partie 1 : Mesure des signaux analogiques (7 points)

Un variateur industriel permettant le contrôle vectoriel des machines synchrones à aimants permanents (MSAP) comprend un DSP TMS320LF2407A cadencé à 40MHz, un onduleur de tension à MLI et l'électronique de mesure des grandeurs de contre-réaction. Le rôle du DSP est de mesurer les 3 courants de phase du moteur et la position angulaire du rotor du moteur, de calculer les algorithmes de contrôle vectoriel et de délivrer les signaux PWM adéquats à l'onduleur de tension. Seule la mesure des 3 courants de phase est traitée dans cet exercice.



On définit les paramètres suivants :
- Fréquence d'échantillonnage : Fe
- Fréquence PWM : Fpwm.

Avec Fe = Fpwm = 10kHz



Les courants à mesurer, représentés ci-contre, ont les caractéristiques suivantes :
- Fréquence fondamental : 0 < F1 < 1kHz
- Plage fréquentielle des harmoniques dues à la PWM : Fh > (Fpwm-4×F1)
- amplitude crête des courants : 0 < Ic < 80A

On utilise des filtres anti-repliement analogiques sur chaque voie analogique. Les séquences de conversion analogique numérique des voies ADCIN0, 1 et 2 sont déclenchées par le timer3 et la lecture des résultats de conversion s'effectue par la fonction d'interruption liée à INT1 et déclenchée par le module ADC.

Question 1 (1 point) :
Déterminer la fréquence de coupure des filtres anti-repliement pour préserver au maximum les transitoires de courant, et représenter l'enveloppe spectrale des signaux à la sortie des filtres.

Question 2 (1 point) :
Le circuit de calibration permet d'adapter l'étendue de mesure des courants à la plage de tension (0, 3.3V) du module ADC 10 bits. Exprimer N le résultat de conversion en fonction du courant.

Question 3 (1 point) :
Déterminer le temps pris par une séquence de conversion. Déterminer le pourcentage de la période d'échantillonnage qu'il représente.

Question 4 (1.5 point) :
Ecrire en langage C la fonction d'initialisation du Timer 3 : ***void init_Timer3(void)***.

Question 5 (1.5 point) :
Ecrire en langage C la fonction d'initialisation du module ADC : ***void init_ADC(void)*** pour permettre son fonctionnement en mode cascadé et start/stop.

Question 6 (1 point) :
Indiquer (en assembleur) comment initialiser les vecteurs d'interruption pour associer l'interruption INT1 du DSP à la fonction : ***interrupt void interADC(void)***.

Partie 2 : Transmission de données par liaison série (6 points)

La liaison SCI du DSP est utilisée pour communiquer avec un micro-ordinateur. Le constructeur fournit un logiciel de communication PC non étudié ici permettant le paramétrage du variateur et l'affichage de données. On étudie dans cette partie la programmation de la liaison SCI du DSP.

Question 1 (1 point) :
Les broches SCIRX et SCITX du DSP sont connectées à un driver/receiver de ligne RS232. Rappeler quelles sont les limites des niveaux électriques de la ligne, correspondant à la transmission des '0' et des '1'. Donner également la longueur maximale de la ligne et la vitesse de transmission maximale pour cette longueur de ligne.

Question 2 (2 points) :
Le module SCI est configuré dans le mode suivant : 115200 bauds, données de 8 bits, pas de bit de parité, 1 bit de STOP, RXINT activée en priorité faible (SCIRX Priority = 1). Ecrire en langage C la fonction d'initialisation du module SCI ***void init_SCI(void)***.

Question 3 (1 point) :
Ecrire en langage C la fonction d'émission d'un octet ***void putchar(unsigned int a)***.

Question 4 (1 point) :
Déterminer la vitesse de transmission utile maximale (bande passante) en octets par seconde.

Question 5 (1 point) :
Indiquer comment initialiser les vecteurs d'interruption pour associer l'interruption INT5 du DSP via RXINT à la fonction gérant le remplissage du buffer circulaire : ***interrupt void interSCI(void)***.

Exercice 2 : Horloge temps réel I2C PCF8563 (7 points)

Question 1 (1 point) :
Représenter schématiquement le principe de connexion de plusieurs composant I2C au bus I2C.

Question 2 (0.5 point) :
Décrire le principe de la dominance de bit et préciser comment elle est définie pour le bus I2C.

Question 3(0.5 point) :
Indiquer quelles sont les adresses en lecture et en écriture du **PCF8563**.

Question 4 (2 points) :
Représenter la trame I2C (à l'image de celles représentées dans la documentation) qui permet de lire l'heure (secondes, minutes, heures) et la date (jour du mois, jour de la semaine, mois, siècle et année).

Question 5 (1 point) :
En négligeant les temps de traitement du processeur maître, évaluer la durée de lecture de la date et de l'heure lorsque le bus I2C est cadencé à sa vitesse maximale.

Question 6 (1 point) :
En considérant que chaque accès au contrôleur I2C (start, stop, lecture, écriture) requière 2 cycles d'un processeur cadencé à 40MHz, évaluer le temps de traitement CPU utile associé à la lecture de la date.

Question 7 (1 point) :
Indiquer quel est le principal avantage de l'utilisation des interruptions d'un contrôleur I2C.

Annexes

Registre de configuration du DSP : SCSR1

15	14	13	12	11	10	9	8
Reserved	CLKSRC	LPM1	LPM0	CLK PS2	CLK PS1	CLK PS0	Reserved
R-0	RW-0	RW-0	RW-0	RW-1	RW-1	RW-1	R-0
7	6	5	4	3	2	1	0
ADC CLKEN	SCI CLKEN	SPI CLKEN	CAN CLKEN	EVB CLKEN	EVA CLKEN	Reserved	ILLADR
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	R-0	RC-0

Note: R = Read access, W = Write access, C = Clear, -0 = value after reset

Entrées/sorties numériques multiplexées

PIN FUNCTION SELECTED		MUX CONTROL REGISTER (name.bit #)	MUX CONTROL VALUE AT RESET (MCRx.n)	I/O PORT DATA AND DIRECTION‡		
(MCRx.n = 1) Primary Function	(MCRX.N = 0) I/O			REGISTER	DATA BIT NO.§	DIR BIT NO.¶
				PORT A		
SCITXD	***IOPA0***	MCRA.0	0	PADATDIR	0	8
SCIRXD	***IOPA1***	MCRA.1	0	PADATDIR	1	9
XINT1	***IOPA2***	MCRA.2	0	PADATDIR	2	10
CAP1/QEP1	***IOPA3***	MCRA.3	0	PADATDIR	3	11
CAP2/QEP2	***IOPA4***	MCRA.4	0	PADATDIR	4	12
CAP3	***IOPA5***	MCRA.5	0	PADATDIR	5	13
PWM1	***IOPA6***	MCRA.6	0	PADATDIR	6	14
PWM2	***IOPA7***	MCRA.7	0	PADATDIR	7	15

Module SCI

Address	Register Mnemonic	Bit Number 7	6	5	4	3	2	1	0	Register Name
7050h	SCICCR	STOP BITS	EVEN/ ODD PARITY	PARITY ENABLE	LOOP-BACK ENA	ADDR/ IDLE MODE	SCI CHAR2	SCI CHAR1	SCI CHAR0	Commu-nication control
7051h	SCICTL1	Reserved	RX ERR INT ENA	SW RESET	Reserved	TXWAKE	SLEEP	TXENA	RXENA	SCI control register1
7052h	SCIHBAUD	BAUD15 (MSB)	BAUD14	BAUD13	BAUD12	BAUD11	BAUD10	BAUD9	BAUD8	Baud rate (MSbyte)
7053h	SCILBAUD	BAUD7	BAUD6	BAUD5	BAUD4	BAUD3	BAUD2	BAUD1	BAUD0 (LSB)	Baud rate (LSbyte)
7054h	SCICTL2	TXRDY	TX EMPTY	Reserved				RX/BK INT ENA	TX INT ENA	SCI control register 2
7055h	SCIRXST	RX ERROR	RXRDY	BRKDT	FE	OE	PE	RXWAKE	Reserved	Receiver status
7057h	SCIRXBUF	RXDT7	RXDT6	RXDT5	RXDT4	RXDT3	RXDT2	RXDT1	RXDT0	Receiver data buffer
7058h	—	Reserved								—
7059h	SCITXBUF	TXDT7	TXDT7	TXDT5	TXDT4	TXDT3	TXDT2	TXDT1	TXDT0	Transmitter data buffer
705Fh	SCIPRI	Reserved	SCITX PRIORITY	SCIRX PRIORITY	SCI SOFT	SCI FREE	Reserved			Priority control

$$BRR = \frac{CLKOUT}{\text{SCI Asynchronous Baud} \times 8} - 1$$

ADC Control Register 1 (ADCTRL1)

15	14	13	12	11	10	9	8
Reserved	RESET	SOFT	FREE	ACQ PS3	ACQ PS2	ACQ PS1	ACQ PS0
	RS-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
7	6	5	4	3	2	1	0
CPS	CONT RUN	INT PRI	SEQ CASC	CAL ENA	BRG ENA	HI/LO	STEST ENA
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Note: R = Read access, W = Write access, S = Set only, -0 = value after reset

Bit 15 **Reserved**

Bit 14 **RESET**. ADC module software reset

This bit causes a master reset on the entire ADC module. All register bits and sequencer state machines are reset to the initial state as occurs when the device reset pin is pulled low (or after a power-on reset).

0 No effect

1 Resets entire ADC module (bit is then set back to 0 by ADC logic)

Note: Using the RESET Bit in the ADCTRL1 Register

The ADC module is reset during a system reset. If an ADC module reset is desired at any other time, you can do so by writing a 1 to this bit. After a NOP, you can then write the appropriate values to the ADCTRL1 register bits:

```
SPLK #01xxxxxxxxxxxxxxb,ADCTRL1; Resets the ADC (RESET = 1)
NOP                            ; Provides the required delay
                               ; between writes to ADCTRL1
SPLK #00xxxxxxxxxxxxxxb,ADCTRL1; Takes the ADC out of Reset
                               ; (RESET = 0)
```

Note that the second SPLK is not required if the default configuration is sufficient.

Bits 13, 12 **SOFT and FREE**. Soft and Free bits

These bits determine what occurs when an emulation-suspend occurs (due to the debugger hitting a breakpoint, for example). In free-run mode, the peripheral can continue with whatever it is doing. In stop mode, the peripheral can either stop immediately or stop when the current operation (i.e., the current conversion) is complete.

Soft	Free	
0	0	Immediate stop on suspend
1	0	Complete current conversion before stopping
X	1	Free run, continue operation regardless of suspend

Bits 11–8 **ACQ PS3 – ACQ PS0**. Acquisition time window – prescale bits 3–0

These bits define the ADC clock prescale factor applied to the acquisition portion of the conversion. The prescale values are defined in Table 7–3 and Table 7–4.

#	ACQ PS3	ACQ PS2	ACQ PS1	ACQ PS0	Pre-scaler (div by)	Acquisition Time Window	Source Z (CPS=0) (Ω)	Source Z (CPS=1) (Ω)
0	0	0	0	0	1	2 x T_{clk}	67	385
1	0	0	0	1	2	4 x T_{clk}	385	1020
2	0	0	1	0	3	6 x T_{clk}	702	1655
3	0	0	1	1	4	8 x T_{clk}	1020	2290
4	0	1	0	0	5	10 x T_{clk}	1337	2925
5	0	1	0	1	6	12 x T_{clk}	1655	3560
6	0	1	1	0	7	14 x T_{clk}	1972	4194
7	0	1	1	1	8	16 x T_{clk}	2290	4829
8	1	0	0	0	9	18 x T_{clk}	2607	5464
9	1	0	0	1	10	20 x T_{clk}	2925	6099
A	1	0	1	0	11	22 x T_{clk}	3242	6734
B	1	0	1	1	12	24 x T_{clk}	3560	7369
C	1	1	0	0	13	26 x T_{clk}	3877	8004
D	1	1	0	1	14	28 x T_{clk}	4194	8639
E	1	1	1	0	15	30 x T_{clk}	4512	9274
F	1	1	1	1	16	32 x T_{clk}	4829	9909

Notes: 1) Period of T_{clk} is dependent on the "Conversion Clock Prescale" bit (Bit 7); i.e.,
CPS = 0: T_{clk} = 1/CLK (example, for CLK = 30 MHz, T_{clk} = 33 ns)
CPS = 1: T_{clk} = 2 × (1/CLK) (example, for CLK = 30 MHz, T_{clk} = 66 ns)
2) Source impedance Z is a design estimate only.

Bit 7 **CPS**. Conversion clock prescale

This bit defines the ADC conversion logic clock prescale

0 F_{clk} = CLK/1

1 F_{clk} = CLK/2

CLK = CPU clock frequency

Bit 6 **CONT RUN**. Continuous run

This bit determines whether the sequencer operates in continuous conversion mode or start-stop mode. This bit can be written while a current conversion sequence is active. This bit will take effect at the end of the current conversion sequence; i.e., software can set/clear this bit until EOS has occurred, for valid action to be taken. In the continuous conversion mode, there is no need to reset the sequencer; however, the sequencer must be reset in the start-stop mode to put the converter in state CONV00.

0 Start-stop mode. Sequencer stops after reaching EOS. This is used for multiple time-sequenced triggers.

1 Continuous conversion mode. After reaching EOS, the sequencer starts all over again from state CONV00 (for SEQ1 and cascaded) or CONV08 (for SEQ2).

Bit 5 **INT PRI**. ADC interrupt request priority

0 High priority

1 Low priority

Bit 4 **SEQ CASC**. Cascaded sequencer operation

This bit determines whether SEQ1 and SEQ2 operate as two 8-state sequencers or as a single 16-state sequencer (SEQ).

0 Dual-sequencer mode. SEQ1 and SEQ2 operate as two 8-state sequencers.

1 Cascaded mode. SEQ1 and SEQ2 operate as a single 16-state sequencer (SEQ).

Bit 3 **CAL ENA**. Offset calibration enable

When set to 1, CAL ENA disables the input channel multiplexer, and connects the calibration reference selected by the bits HI/LO and BRG ENA to the ADC core inputs. The calibration conversion can then be started by setting bit 14 of ADCTRL2 register (STRT CAL) to 1. Note that CAL ENA should be set to 1 first before the STRT CAL bit can be used.

Note: This bit should not be set to 1 if STEST ENA = 1

0 Calibration mode disabled

1 Calibration mode enabled

Bit 2 **BRG ENA**. Bridge enable

Together with the HI/LO bit, BRG ENA allows a reference voltage to be converted in calibration mode. See the description of the HI/LO bit for reference voltage selections during calibration.

0 Full reference voltage is applied to the ADC input

1 A reference midpoint voltage is applied to the ADC input

Bit 1 **HI/LO**. V_{REFHI}/V_{REFLO} selection

When the fail self-test mode is enabled (STEST ENA = 1), HI/LO defines the test voltage to be connected. In calibration mode, HI/LO defines the reference source polarity; see Table 7–5. In normal operating mode, HI/LO has no effect.

0 V_{REFLO} is used as precharge value at ADC input

1 V_{REFHI} is used as precharge value at ADC input

Table 7–5. Reference Voltage Bit Selection

BRG ENA	HI/LO	CAL ENA = 1 Reference voltage (V)	STEST ENA = 1 Reference voltage (V)
0	0	V_{REFLO}	V_{REFLO}
0	1	V_{REFHI}	V_{REFHI}
1	0	$\lvert(V_{REFHI} - V_{REFLO}) / 2\rvert$	V_{REFLO}
1	1	$\lvert(V_{REFLO} - V_{REFHI}) / 2\rvert$	V_{REFHI}

Bit 0 **STEST ENA**. Self-test function enable

0 Self-test mode disabled

1 Self-test mode enabled

ADC Control Register 2 (ADCTRL2)

15	14	13	12	11	10	9	8
EVB SOC SEQ	RST SEQ1/ STRT CAL	SOC SEQ1	SEQ1 BSY	INT ENA SEQ1 (Mode 1)	INT ENA SEQ1 (Mode 0)	INT FLAG SEQ1	EVA SOC SEQ1
RW-0	RS-0	RW-0	R-0	RW-0	RW-0	RC-0	RW-0
7	6	5	4	3	2	1	0
EXT SOC SEQ1	RST SEQ2	SOC SEQ2	SEQ2 BSY	INT ENA SEQ2 (Mode 1)	INT ENA SEQ2 (Mode 0)	INT FLAG SEQ2	EVB SOC SEQ2
RW-0	RS-0	RW-0	R-0	RW-0	RW-0	RC-0	RW-0

Note: R = Read access, W = Write access, S = Set only, C = Clear, -0 = value after reset

Bit 15 **EVB SOC SEQ**. EVB SOC enable for cascaded sequencer
(Note: This bit is active only in cascaded mode.)

0 No action

1 Setting this bit allows the cascaded sequencer to be started by an Event Manager B signal. The Event Manager can be programmed to start a conversion on various events. See chapter 6, Event Manager (EV), for details.

Bit 14 **RST SEQ1 / STRT CAL**. Reset Sequencer1/Start Calibration

Case: Calibration Disabled (Bit 3 of ADCTRL1) = 0

Writing a 1 to this bit will reset the sequencer immediately to an initial "pretriggered" state, i.e., waiting for a trigger at CONV00. A currently active conversion sequence will be aborted.

0 No action

1 Immediately reset sequencer to state CONV00

Case: Calibration Enabled (Bit 3 of ADCTRL1) = 1

Writing a 1 to this bit will begin the converter calibration process.

0 No action

1 Immediately start calibration process

Bit 13 **SOC SEQ1**. Start-of-conversion (SOC) trigger for Sequencer 1 (SEQ1). This bit can be set by the following triggers:

- ❏ S/W – Software writing a 1 to this bit
- ❏ EVA – Event Manager A
- ❏ EVB – Event Manager B (only in cascaded mode)
- ❏ EXT – External pin (i.e., the ADCSOC pin)

When a trigger occurs, there are three possibilities:

Case 1: SEQ1 idle and SOC bit clear
SEQ1 starts immediately (under arbiter control). This bit is set and cleared, allowing for any "pending" trigger requests.

Case 2: SEQ1 busy and SOC bit clear
Bit is set signifying a trigger request is pending. When SEQ1 finally starts after completing current conversion, this bit will be cleared.

Case 3: SEQ1 busy and SOC bit set
Any trigger occurring in this case will be ignored (lost).

0 Clears a pending SOC trigger.
Note: If the sequencer has already started, this bit will automatically be cleared, and hence, writing a zero will have no effect; i.e., an already started sequencer cannot be stopped by clearing this bit.

1 Software trigger – Start SEQ1 from currently stopped position (i.e., Idle mode)

Note:

The RST SEQ1 (ADCTRL2.14) and the SOC SEQ1 (ADCTRL2.13) bits should not be set in the same instruction. This will reset the sequencer, but will not start the sequence. The correct sequence of operation is to set the RST SEQ1 bit first, and the SOC SEQ1 bit in the following instruction. This ensures that the sequencer is reset and a new sequence started. This sequence applies to the RST SEQ2 (ADCTRL2.6) and SOC SEQ2 (ADCTRL2.5) bits also.

Bit 12 **SEQ1 BSY**. SEQ1 Busy

This bit is set to a 1 while the ADC autoconversion sequence is in progress. It is cleared when the conversion sequence is complete.

0 Sequencer is Idle (i.e., waiting for trigger)

1 Conversion sequence is in progress

After a start-of-sequence (SOC) is initiated, four NOPs need to be executed before polling the SEQ1 or SEQ2 BSY bit.

Example code:

```
ADC_LOOP1:
       LDP    #ADCTRL1>>7

       SPLK   #0100000000000000b,ADCTRL2   ;Reset for SEQ1

       SPLK   #0010000000000000b,ADCTRL2   ;SOC for SEQ1

       NOP                                 ;Wait for Busy
                                           ;bit to set.
       NOP
       NOP
       NOP

CHK_EOS1:
       BIT    ADCTRL2, 3          ; Wait for SEQ1 Busy bit
                                  ; to clear

       BCND   CHK_EOS1, TC        ; If TC=1, keep looping.
```

A better approach would be to check the INT FLAG SEQn bit for end-of-sequence. This does not require NOPs, as the bit should already be cleared prior to starting a sequenced conversion. To reiterate, the NOPs are required only when polling the SEQn BUSY bit; interrupt-driven conversions do not have this requirement.

Bits 11–10 **INT ENA SEQ1**. Interrupt-mode-enable control for SEQ1

Bit 11	Bit 10	Operation Description
0	0	Interrupt is Disabled
0	1	Interrupt **Mode 1** Interrupt requested immediately when INT FLAG SEQ1 flag is set
1	0	Interrupt **Mode 2** Interrupt requested only if INT FLAG SEQ1 flag is already set. If clear†, INT FLAG SEQ1 flag is set and INT request is suppressed. (This mode allows Interrupt requests to be generated for every other EOS.)
1	1	Reserved

† This means that the last completed sequence is the first of the two sequences needed to assert an interrupt.

Bit 9 **INT FLAG SEQ1.** ADC interrupt flag bit for SEQ1

This bit indicates whether an interrupt event has occurred or not. This bit must be cleared by the user writing a 1 to it.

0 No interrupt event

1 An interrupt event has occurred.

Bit 8 **EVA SOC SEQ1.** Event Manager A SOC mask bit for SEQ1

0 SEQ1 cannot be started by EVA trigger.

1 Allows SEQ1/SEQ to be started by Event Manager A trigger. The Event Manager can be programmed to start a conversion on various events. See chapter 6, *Event Manager (EV)*, for details.

Bit 7 **EXT SOC SEQ1.** External signal start-of-conversion bit for SEQ1

0 No action

1 Setting this bit enables an ADC autoconversion sequence to be started by a signal from the ADCSOC device pin.

Bit 6 **RST SEQ2.** Reset SEQ2

0 No action

1 Immediately resets SEQ2 to an initial "pretriggered" state, i.e., waiting for a trigger at CONV08. A currently active conversion sequence will be aborted.

Bit 5 **SOC SEQ2.** Start-of-conversion trigger for Sequencer 2 (SEQ2)
(Only applicable in dual-sequencer mode; ignored in cascaded mode.)

This bit can be set by the following triggers:

- ❏ S/W – Software writing of 1 to this bit
- ❏ EVB – Event Manager B

When a trigger occurs, there are three possibilities:

Case 1: SEQ2 idle and SOC bit clear
SEQ2 starts immediately (under arbiter control) and the bit is cleared, allowing for any pending trigger requests.

Case 2: SEQ2 busy and SOC bit clear
Bit is set signifying a trigger request is pending. When SEQ2 finally starts after completing current conversion, this bit will be cleared.

Case 3: SEQ2 busy and SOC bit set
Any trigger occurring in this case will be ignored (lost).

0 Clears a Pending SOC trigger.
Note: If the sequencer has already started, this bit will automatically be cleared, and hence, writing a zero will have no effect; i.e., an already started sequencer cannot be stopped by clearing this bit.

1 Software trigger – Start SEQ2 from currently stopped position (i.e., Idle mode)

Bit 4 **SEQ2 BSY.** SEQ2 Busy

This bit is set to a 1 while the ADC autoconversion sequence is in progress. It is cleared when the conversion sequence is complete.

0 Sequencer is idle (i.e., waiting for trigger).

1 Conversion sequence is in progress.

Bits 3–2 **INT ENA SEQ2.** Interrupt-mode-enable control for SEQ2

Bit 3	Bit 2	Operation Description
0	0	Interrupt is Disabled
0	1	Interrupt **Mode 1** Interrupt requested immediate on INT FLAG SEQ2 flag set
1	0	Interrupt **Mode 2** Interrupt requested only if INT FLAG SEQ2 flag is already set. If clear†, INT FLAG SEQ2 flag is set and INT request is suppressed. (This mode allows Interrupt requests to be generated for every other EOS)
1	1	Reserved

† This means that the last completed sequence is the first of the two sequences needed to assert an interrupt.

Bit 1 **INT FLAG SEQ2.** ADC interrupt flag bit for SEQ2

This bit indicates whether an interrupt event has occurred or not. This bit must be cleared by the user writing a 1 to it.

0 No interrupt event.

1 An interrupt event has occurred.

Bit 0 **EVB SOC SEQ2.** Event Manager B SOC mask bit for SEQ2

0 SEQ2 cannot be started by EVB trigger.

1 Allows SEQ2 to be started by Event Manager B trigger. The Event Manager can be programmed to start a conversion on various events. See chapter 6, *Event Manager (EV)*, for details.

Timers du DSP

Registre de contrôle général des timers GPTCONA

15	14	13	12–11	10–9	8–7
Reserved	T2STAT	T1STAT	Reserved	T2TOADC	T1TOADC
RW-0	R-1	R-1	RW-0	RW-0	RW-0

6	5–4	3–2	1–0
TCOMPOE	Reserved	T2PIN	T1PIN
RW-0	RW-0	RW-0	RW-0

Note: R = Read access, W = Write access, -n = value after reset

Bit 15 **Reserved**. Reads return zero; writes have no effect.

Bit 14 **T2STAT**. GP timer 2 Status. Read only.
- 0 Counting downward
- 1 Counting upward

Bit 13 **T1STAT**. GP timer 1 Status. Read only.
- 0 Counting downward
- 1 Counting upward

Bits 12–11 **Reserved**. Reads return zero; writes have no effect.

Bits 10–9 **T2TOADC**. Start ADC with timer 2 event.
- 00 No event starts ADC
- 01 Setting of underflow interrupt flag starts ADC
- 10 Setting of period interrupt flag starts ADC
- 11 Setting of compare interrupt flag starts ADC

Bits 8–7 **T1TOADC**. Start ADC with timer 1 event.
- 00 No event starts ADC
- 01 Setting of underflow interrupt flag starts ADC
- 10 Setting of period interrupt flag starts ADC
- 11 Setting of compare interrupt flag starts ADC

Bit 6 **TCOMPOE**. Compare output enable. If $\overline{\text{PDPINTx}}$ is active this bit is set to zero.
- 0 Disable all GP timer compare outputs (all compare outputs are put in the high-impedance state)
- 1 Enable all GP timer compare outputs

Bits 5–4 **Reserved**. Reads return zero; writes have no effect.

Bits 3–2 **T2PIN**. Polarity of GP timer 2 compare output.
- 00 Forced low
- 01 Active low
- 10 Active high
- 11 Forced high

Bits 1–0 **T1PIN**. Polarity of GP timer 1 compare output.
- 00 Forced low
- 01 Active low
- 10 Active high
- 11 Forced high

Registre de contrôle individuel des timers : TxCON (x=1, 2, 3 ou 4)

15	14	13	12	11	10	9	8
Free	Soft	Reserved	TMODE1	TMODE0	TPS2	TPS1	TPS0
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

7	6	5	4	3	2	1	0
T2SWT1/ T4SWT3†	TENABLE	TCLKS1	TCLKS0	TCLD1	TCLD0	TECMPR	SELT1PR/ SELT3PR†
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Note: R = Read access, W = Write access, -0 = value after reset
† Reserved in T1CON andT3CON

Bits 15–14 **Free, Soft.** Emulation control bits.
- 00 Stop immediately on emulation suspend
- 01 Stop after current timer period is complete on emulation suspend
- 10 Operation is not affected by emulation suspend
- 11 Operation is not affected by emulation suspend

Bit 13 **Reserved**. Reads return zero, writes have no effect.

Bits 12–11 **TMODE1–TMODE0.** Count Mode Selection.
- 00 Stop/Hold
- 01 Continuous-Up/-Down Count Mode
- 10 Continuous-Up Count Mode
- 11 Directional-Up/-Down Count Mode

Bits 10–8 **TPS2–TPS0.** Input Clock Prescaler.

000	x/1	100	x/16
001	x/2	101	x/32
010	x/4	110	x/64
011	x/8	111	x/128

x = device (CPU) clock frequency

Bit 7 **T2SWT1.** In the case of EVA, this bit is T2SWT1. (GP timer 2 start with GP timer 1.) Start GP timer 2 with GP timer 1's timer enable bit. This bit is reserved in T1CON.
T4SWT3. In the case of EVB, this bit is T4SWT3. (GP timer 4 start with GP timer 3.) Start GP timer 4 with GP timer 3's timer enable bit. This bit is reserved in T3CON.
- 0 Use own TENABLE bit
- 1 Use TENABLE bit of T1CON (in case of EVA) or T3CON (in case of EVB) to enable and disable operation ignoring own TENABLE bit

Bit 6 **TENABLE**. Timer enable.
- 0 Disable timer operation (the timer is put in hold and the prescaler counter is reset)
- 1 Enable timer operations

Bits 5–4 **TCLKS1, TCLKS0.** Clock Source Select.

5	4	Source
0	0	Internal
0	1	External
1	0	Reserved
1	1	QEP Circuit† (in case of Timer 2/Timer 4)
		Reserved (in case of Timer 1/Timer 3)

† This option is valid only if SELT1PR = 0

Bits 3–2 **TCLD1, TCLD0**. Timer Compare Register Reload Condition.
- 00 When counter is 0
- 01 When counter value is 0 or equals period register value
- 10 Immediately
- 11 Reserved

Bit 1 **TECMPR**. Timer compare enable.
- 0 Disable timer compare operation
- 1 Enable timer compare operation

Bit 0 **SELT1PR**. In the case of EVA, this bit is SELT1PR (Period register select). When set to1 in T2CON, the period register of Timer 1 is chosen for Timer 2 also, ignoring the period register of Timer 2. This bit is a reserved bit in T1CON.
SELT3PR. In the case of EVB, this bit is SELT3PR (Period register select). When set to1 in T4CON, the period register of Timer 3 is chosen for Timer 4 also, ignoring the period register of Timer 4. This bit is a reserved bit in T3CON.
- 0 Use own period register
- 1 Use T1PR (in case of EVA) or T3PR (in case of EVB) as period register ignoring own period register

PCF8563 : Horloge temps réel

(Extrait du datasheet)

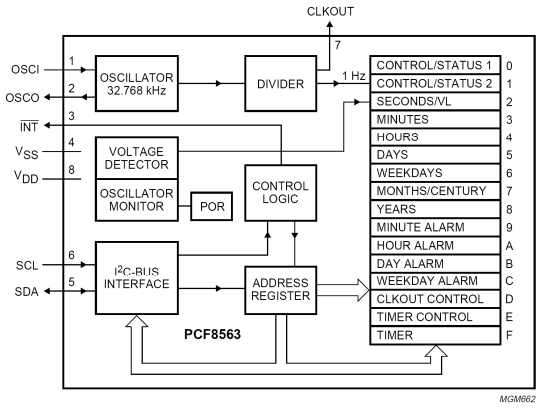
Functional description

The PCF8563 contains sixteen 8-bit registers with an auto-incrementing address register, an on-chip 32.768 kHz oscillator with one integrated capacitor, a frequency divider which provides the source clock for the Real Time Clock/calender (RTC), a programmable clock output, a timer, an alarm, a voltage-low detector and a 400 kHz I²C-bus interface.

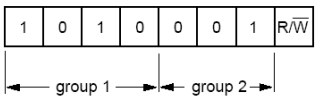
All 16 registers are designed as addressable 8-bit parallel registers although not all bits are implemented. The first two registers (memory address 00H and 01H) are used as control and/or status registers. The memory addresses 02H through 08H are used as counters for the clock function (seconds up to years counters). Address locations 09H through 0CH contain alarm registers which define the conditions for an alarm. Address 0DH controls the CLKOUT output frequency. 0EH and 0FH are the timer control and timer registers, respectively.

The seconds, minutes, hours, days, weekdays, months, years as well as the minute alarm, hour alarm, day alarm and weekday alarm registers are all coded in BCD format.

When one of the RTC registers is read the contents of all counters are frozen. Therefore, faulty reading of the clock/calendar during a carry condition is prevented.



Slave address.



Clock/calendar read/write cycles

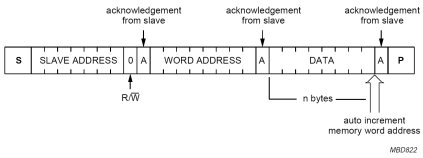


Fig 13. Master transmits to slave receiver (write mode).

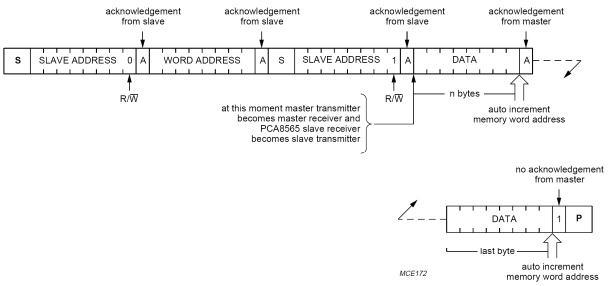


Fig 14. Master reads after setting word address (write word address; read data).

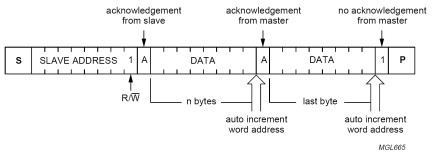


Fig 15. Master reads slave immediately after first byte (read mode).

Register organization

Table 4: Binary formatted registers overview

Bit positions labelled as x are not implemented. Bit positions labelled with 0 should always be written with logic 0; if read they could be either logic 0 or logic 1.

Address	Register name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00H	control/status 1	TEST1	0	STOP	0	TESTC	0	0	0
01H	control/status 2	0	0	0	TI/TP	AF	TF	AIE	TIE
0DH	CLKOUT control	FE	x	x	x	x	x	FD1	FD0
0EH	timer control	TE	x	x	x	x	x	TD1	TD0
0FH	timer	<timer countdown value>							

Table 5: BCD formatted registers overview

Bit positions labelled as x are not implemented.

Address	Register name	BCD format tens nibble				BCD format units nibble			
		Bit 7 2^3	Bit 6 2^2	Bit 5 2^1	Bit 4 2^0	Bit 3 2^3	Bit 2 2^2	Bit 1 2^1	Bit 0 2^0
02H	seconds	VL	<seconds 00 to 59 coded in BCD>						
03H	minutes	x	<minutes 00 to 59 coded in BCD>						
04H	hours	x	x	<hours 00 to 23 coded in BCD>					
05H	days	x	x	<days 01 to 31 coded in BCD>					
06H	weekdays	x	x	x	x	x	<weekdays 0 to 6>		
07H	months/century	C	x	x	<months 01 to 12 coded in BCD>				
08H	years	<years 00 to 99 coded in BCD>							
09H	minute alarm	AE	<minute alarm 00 to 59 coded in BCD>						
0AH	hour alarm	AE	x	<hour alarm 00 to 23 coded in BCD>					
0BH	day alarm	AE	x	<day alarm 01 to 31 coded in BCD>					
0CH	weekday alarm	AE	x	x	x	x	<weekday alarm 0 to 6>		

Table 9: Seconds/VL (address 02H) bits description

Bit	Symbol	Value	Description
7	VL	0	clock integrity is guaranteed
		1	integrity of the clock information is no longer guaranteed
6 to 0	seconds	00 to 59	this register holds the current seconds coded in BCD format; example: seconds register contains x101 1001 = 59 seconds

Table 14: Weekday assignments

Day	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sunday	x	x	x	x	x	0	0	0
Monday	x	x	x	x	x	0	0	1
Tuesday	x	x	x	x	x	0	1	0
Wednesday	x	x	x	x	x	0	1	1
Thursday	x	x	x	x	x	1	0	0
Friday	x	x	x	x	x	1	0	1
Saturday	x	x	x	x	x	1	1	0

Table 15: Months/century (address 07H) bits description

Bit	Symbol	Value	Description
7	century[1]		this bit is toggled when the years register overflows from 99 to 00
		0	indicates the century is 20xx
		1	indicates the century is 19xx
4 to 0	month	01 to 12	this register holds the current month coded in BCD format, see Table 16