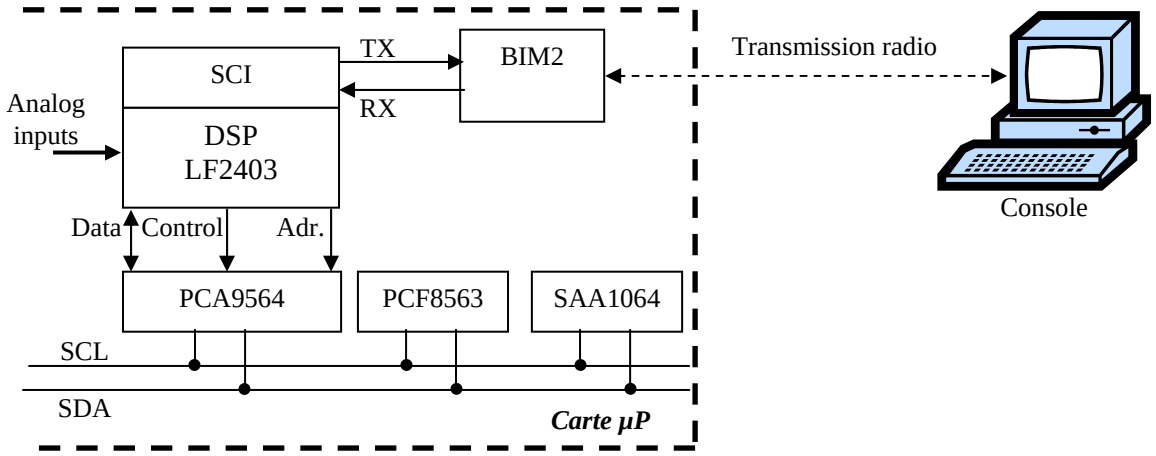
UTBM **Département GESC**
Automne 2006 **Durée : 2 heures. Fascicule DSP et fascicule de cours autorisés**

Partiel MC43

Une station météo est dotée d'une carte microcontrôleur comprenant :

- un DSP contrôleur TMS320LF2403A - 40 MHz (version allégée du TMS320LF2407A)
- un contrôleur I2C PCA9564
- une horloge temps réel I2C PCF8563 (Fmax = 400kHz)
- un pilote d'afficheur 7 segments I2C SAA1064 (Fmax = 100 kHz)
- un émetteur/récepteur radio BIM2 433MHz
- l'électronique de conditionnement des signaux de mesure analogiques et numériques



I Acquisition des données

Le DSP contrôleur réalise l'acquisition de 5 grandeurs analogiques (température, pression atmosphérique, humidité relative de l'air, ensoleillement et direction du vent) et 1 grandeur numérique (vitesse de vent via un codeur incrémental).

I.1 Représenter schématiquement la structure de la chaîne de mesure des grandeurs analogiques.
I.2 Rappeler quel est l'intérêt d'un filtre anti-repliement.
I.3 Comment doit être dimensionnée la fréquence de coupure d'un filtre anti-repliement.
I.4 Décrire le principe de fonctionnement des structures matérielles permettant de mettre en œuvre l'échantillonnage à fréquence fixe des grandeurs à mesurer.
I.5 Décrire l'organisation logicielle du programme d'exploitation des données.

II Gestion de l'horloge temps réel

L'horloge temps réel PCF8563 a pour rôle de donner la date et l'heure. Elle est utilisée ici afin d'associer à chaque mesure sont instant d'échantillonnage. Ce composant est alimenté par une pile, ce qui permet de préserver la date et l'heure lors des éventuelles coupures d'alimentation. Une documentation restreinte du PCF8563 est donnée en annexe. Le microcontrôleur accède au bus I2C par l'intermédiaire du contrôleur I2C PCA9564. On propose de mettre en œuvre les fonctions de lecture de la date.

II.1 Ecrire la fonction ***void initI2C(void)*** qui initialise le contrôleur PCA9564. (L'accès aux registres du PCA9564 s'effectue en utilisant les noms du datasheet. Exemple : I2CCON = …).
II.2 Indiquer quelles sont les adresses en lecture et en écriture du PCF8563.
II.3 Donner la trame I2C permettant de lire la date : secondes, minutes, heure, jour du mois, jour de la semaine, mois, siècle (19xx ou 20xx), année.
II.4 Donner l'organigramme de la fonction ***LIRE_DATE*** qui réalise la lecture de la date complète sur le PCF8563 en accédant aux registres du PCA9564.
II.5 Ecrire la fonction ***void LIRE_DATE(void)*** en langage C.
II.6 Evaluer la durée de lecture de la date.

III Liaison radio

La liaison radio permet de gérer une interface homme machine à l'aide d'une console ou d'un micro-ordinateur et donc entre autre, de transférer les mesures vers le micro-ordinateur. L'émetteur/récepteur radio est directement relié au SCI. Les données numériques sont envoyées à l'émetteur/récepteur configuré en émetteur, à l'aide de la broche TX du SCI tandis que les données reçues et démodulées par l'émetteur/récepteur configuré en récepteur, sont reçue par le DSP à l'aide de la broche RX du SCI. Le micro-ordinateur est doté d'un système de transmission équivalent.

Le SCI est configurée de la manière suivante : 156250 bauds, données de 8 bits, pas de parité, 1 bit de STOP.

III.1 Ecrire en langage C la fonction ***initSCI*** initialisant la configuration du module SCI.

III.2 Chaque transfert doit débuté par un entête de polarisation du BIM2 récepteur, constitué de 0 et de 1 alternativement. On utilise comme entête les 2 octets 0x55, 0x55. Représenter le chronogramme de la broche TX lors de l'envoi de l'entête par le DSP.

III.3 Ecrire en langage C la fonction ***void entete(void)*** réalisant l'émission de l'entête de polarisation.

III.4 Le protocole de communication entre la station météo et le PC est défini de la manière suivante :

- envoi de la chaîne de caractère "RTS" par la station météo
- envoi de la chaîne de caractère "CTS" par le PC après réception de "RTS"
- envoi du caractère 'STX' (début du bloc) par la station météo après réception du "CTS"
- envoi des données au format texte par la station météo, séparées par le caractère 'CR'
- envoi du caractère 'ETX' (fin du bloc) par la station météo
- envoi de la chaîne de caractère "ACK" par le PC après réception de 'ETX'

Chaque caractère est converti en code manchester avant d'être transmis grâce à la fonction ***int manchester(char c)*** non étudiée ici. Le code manchester créé un mot de 16 bits à partir d'un mot de 8 bits.

Ecrire en langage C la fonction ***void emission(int code)*** réalisant la transmission du mot de 16 bits ***code*** (déjà codé en manchester) sur la liaison série.

III.5 Ecrire en langage C la fonction ***void envoichaine(char *s)*** réalisant la transmission de la chaîne de caractère ***s*** sur la liaison série après codage en manchester, en faisant appel aux fonctions ***int manchester(char c)*** et ***void emission(int code)***.

III.6 L'entête de polarisation est envoyé avant la chaîne "RTS" et le caractère 'STX' marquant le début du bloc de données. Déterminer le temps de transmission du "RTS" en tenant compte de l'entête et du codage manchester.

III.7 Ecrire l'organigramme ***envoibloc*** traduisant l'envoi du bloc constitué de la température (T), la pression atmosphérique (P), l'humidité relative de l'air (HU), l'ensoleillement (E), la direction du vent (D), la vitesse du vent (V), les secondes (S), les minutes (MN), les heures (H), le jour du mois (JM), le jour de la semaine (JS), mois (MO), l'année (A), en faisant référence aux fonctions :

- ***void entete(void)***
- ***char *itoa(int i)*** (convertit i en chaîne de caractères et retourne cette chaîne)
- ***void envoichaine(char *s)***
- ***int manchester(char c)***
- ***void émission(int code)***

III.8 Evaluer la durée de transmission d'un bloc complet.

Parallel bus to I²C-bus controller PCA9564

FEATURES

- Parallel-bus to I²C-bus protocol converter and interface
- Both master and slave functions
- Multi-master capability
- Internal oscillator reduces external components
- Operating supply voltage 2.3 V to 3.6 V
- 5 V tolerant I/Os
- Standard and fast mode I²C capable and compatible with SMBus
- ESD protection exceeds 2000 V HEM per JESD22-A114, 200 V MM per JESD22-A115, and 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceed 100 mA.
- Packages offered: SO20, TSSOP20, HVQFN20

APPLICATIONS

- Add I²C bus port to controllers/processors that don't have one
- Add additional I²C bus ports to controllers/processors that need multiple I²C bus ports
- Higher frequency, lower voltage migration path for the PCF8584
- Converts 8 bits of parallel data to serial data stream to prevent having to run a large number of traces across the entire PC board

DESCRIPTION

The PCA9564 is an integrated circuit designed in CMOS technology that serves as an interface between most standard parallel-bus microcontrollers/microprocessors and the serial I²C-bus and allows the parallel bus system to communicate bi-directionally with the I²C-bus. The PCA9564 can operate as a master or a slave and can be a transmitter or receiver. Communication with the I²C-bus is carried out on a byte-wise basis using interrupt or polled handshake. The PCA9564 controls all the I²C-bus specific sequences, protocol, arbitration and timing with no external timing element required.

The PCA9564 is similar to the PCF8584 but operates at lower voltages and higher I²C frequencies. Other enhancements requested by design engineers have also been incorporated.

Characteristic	PCA9564	PCF8584	Comments
Voltage range	2.3-3.6 V	4.5-5.5 V	PCA9564 is 5 V tolerant
Maximum master mode I²C frequency	360 kHz	90 kHz	Faster I²C interface
Maximum slave mode I²C frequency	400 kHz	100 kHz	Faster I²C interface
Clock source	Internal	External	Less expensive and more flexible with internal oscillator
Parallel interface	Fast 50 MHz	Slow	Compatible with faster processors

While the PCF8584 supported most parallel-bus microcontrollers/microprocessors including the Intel 8049/8051, Motorola 6800/68000 and the Zilog Z80, the PCA9564 has been designed to be very similar to the Philips standard 80C51 microcontroller I²C hardware so the devices are not code compatible. Additionally, the PCA9564 does not support the bus monitor "Snoop" mode nor the long distance mode and is not foot print compatible with the PCF8584.

ORDERING INFORMATION

PACKAGES	TEMPERATURE RANGE	ORDER CODE	TOPSIDE MARK	DRAWING NUMBER
20-Pin Plastic SO	-40 to +85 °C	PCA9564D	PCA9564D	SOT163-1
20-Pin Plastic TSSOP	-40 to +85 °C	PCA9564PW	PCA9564	SOT360-1
20-Pin Plastic HVQFN	-40 to +85 °C	PCA9564BS	9564	SOT662-1

Standard packing quantities and other packaging data are available at www.philipslogic.com/packaging.

FUNCTIONAL DESCRIPTION

General

The PCA9564 acts as an interface device between standard high-speed parallel buses and the serial I²C-bus. On the I²C-bus, it can act either as master or slave. Bidirectional data transfer between the I²C-bus and the parallel-bus microcontroller is carried out on a byte-wise basis, using either an interrupt or polled handshake.

Internal Oscillator

The PCA9564 contains an internal 9 MHz oscillator which is used for all I²C timing. The oscillator requires up to 500 µs to start-up after ENSIO bit is set to "1".

Registers

The PCA9564 contains four registers which are used to configure the operation of the device as well as to send and receive serial data.

The registers are selected by setting pins A0 and A1 to the appropriate logic levels before a read or write operation is executed.

CAUTION: Do not write to I²C registers while the I²C-bus is busy and the SIO is in master or addressed slave mode.

REGISTER NAME	REGISTER FUNCTION	A1	A0	READ/ WRITE	DEFAULT
I2CSTA	Status	0	0	R	F8h
I2CTO	Time-out	0	0	W	FFh
I2CDAT	Data	0	1	R/W	00h
I2CADR	Own address	1	0	R/W	00h
I2CCON	Control	1	1	R/W	00h

The Time-out Register, I2CTO: The time-out register is used to determine the maximum time that SCL is allowed to be LOW before the I²C state machine is reset.

When the I²C interface is operating, I2CTO is loaded in the time-out counter at every SCL transition.

	7	6	5	4	3	2	1	0
I2CTO	TE	TO6	TO5	TO4	TO3	TO2	TO1	TO0
		◄ Time-out value ►						

The most significant bit of I2CTO (TE) is used as a time-out enable/disable. A "1" will enable the time-out function. The time-out period = (I2CTO[6:0] + 1) × 113.7 µs. The time-out value may vary some and is an approximate value.

The time-out register can be used in the following cases:

1. When the SIO, in the master mode, wants to send a START condition and the SCL line is held LOW by some other device. The SIO waits a time period equivalent to the time-out value for the SCL to be released. In case it is not released, the SIO concludes that there is a bus error, loads 90H in the I2CSTA register, generates an interrupt signal and releases the SCL and SDA lines. After the microcontroller reads the status register, it needs to send an external reset in order to reset the SIO.
2. In the master mode, the time-out feature starts every time the SCL goes LOW. If SCL stays LOW for a time period equal to or greater than the time-out value, the SIO concludes there is a bus error and behaves in the manner described above.
3. In case of a forced access to the I²C-bus. (See more details on page 15.)

The Address Register, I2CADR: I2CADR is not affected by the SIO hardware. The contents of this register are irrelevant when SIO is in a master mode. In the slave modes, the seven most significant bits must be loaded with the microcontroller's own slave address.

	7	6	5	4	3	2	1	0
I2CADR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	0
	◄ own slave address ►							

The most significant bit corresponds to the first bit received from the I²C-bus after a start condition. A logic 1 in I2CADR corresponds to a HIGH level on the I²C-bus, and a logic 0 corresponds to a LOW level on the bus. The least significant bit is not used but should be programmed with a '0'.

The Data Register, I2CDAT: I2CDAT contains a byte of serial data to be transmitted or a byte which has just been received. In master mode, this includes the slave address that the master wants to send out on the I²C-bus, with the most significant bit of the slave address in the SD7 bit position and the Read/Write bit in the SD0 bit position. The CPU can read from and write to this 8-bit register while it is not in the process of shifting a byte. This occurs when SIO is in a defined state and the serial interrupt flag is set. Data in I2CDAT remains stable as long as SI is set. Whenever the SIO generates an interrupt, the I2CDAT registers contain the data byte that was just transferred on the I²C-bus.

NOTE: The I2CDAT register will capture the serial address as data when addressed via the serial bus. Also, the data register will continue to capture data from the serial bus during 38H so the I2CDAT register will need to be reloaded when the bus becomes free.

	7	6	5	4	3	2	1	0
I2CDAT	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

- SD7 - SD0:

Eight bits to be transmitted or just received. A logic 1 in I2CDAT corresponds to a HIGH level on the I²C-bus, and a logic 0 corresponds to a LOW level on the bus.

The Control Register, I2CCON: The microcontroller can read from and write to this 8-bit register. Two bits are affected by the SIO hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I²C-bus.

	7	6	5	4	3	2	1	0
I2CCON	AA	ENSIO	STA	STO	SI	CR2	CR1	CR0

- ENSIO, THE SIO ENABLE BIT

ENSIO = "0": When ENSIO is "0", the SDA and SCL outputs are in a high impedance state. SDA and SCL input signals are ignored, SIO is in the "not addressed" slave state.

ENSIO = "1": When ENSIO is "1", SIO is enabled.

After the ENSIO bit is set, it takes 500 µs for the internal oscillator to start up, therefore, the PCA9564 will enter either the master or the slave mode after this time. ENSIO should not be used to temporarily release the PCA9564 from the I²C-bus since, when ENSIO is reset, the I²C-bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

In the following text, it is assumed that ENSIO = "1".

- STA, THE START FLAG

STA = “1”: When the STA bit is set to enter a master mode, the SIO hardware checks the status of the I²C-bus and generates a START condition if the bus is free. If the bus is not free, then SIO waits for a STOP condition (which will free the bus) and generates a START condition after the minimum buffer time (t_{BUF}) has elapsed.

If STA is set while SIO is already in a master mode and one or more bytes are transmitted or received, SIO transmits a repeated START condition. STA may be set at any time. STA may also be set when SIO is an addressed slave.

STA = “0”: When the STA bit is reset, no START condition or repeated START condition will be generated.

- STO, THE STOP FLAG

STO = “1”: When the STO bit is set while SIO is in a master mode, a STOP condition is transmitted to the I²C-bus. When the STOP condition is detected on the bus, the SIO hardware clears the STO flag.

If the STA and STO bits are both set, then a STOP condition is transmitted to the I²C-bus if SIO is in a master mode. SIO then transmits a START condition.

STO = “0”: When the STO bit is reset, no STOP condition will be generated.

- SI, THE SERIAL INTERRUPT FLAG

SI = “1”: When the SI flag is set, then, if the ENSIO bit is also set, a serial interrupt is requested. SI is set by hardware when one of 24 of the 25 possible SIO states is entered. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available.

While SI is set, the LOW period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A HIGH level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by writing “0” to the SI bit. The SI bit cannot be set by the user.

SI = “0”: When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

- AA, THE ASSERT ACKNOWLEDGE FLAG

AA = “1”: If the AA flag is set, an acknowledge (LOW level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- The “own slave address” has been received
- A data byte has been received while SIO is in the master receiver mode
- A data byte has been received while SIO is in the addressed slave receiver mode

AA = “0”: if the AA flag is reset, a not acknowledge (HIGH level to SDA) will be returned during the acknowledge clock pulse on SCL when:

- A data byte has been received while SIO is in the master receiver mode
- A data byte has been received while SIO is in the addressed slave receiver mode
- “Own slave address” has been received

When SIO is in the addressed slave transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 5). When SI is cleared, enters the not addressed slave receiver mode, and the SDA line remains at a HIGH level. In state C8H, the AA flag can be set again for future address recognition.

When SIO is in the not addressed slave mode, its own slave address is ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO can be temporarily released from the I²C-bus while the bus status is monitored. While SIO is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag.

- THE CLOCK RATE BITS, CR_2, CR_1, AND CR_0

Three bits determine the serial clock frequency when SIO is in master mode. The various serial rates are shown in Table 1.

The clock frequencies only take the HIGH and LOW times into consideration. The rise and fall time will cause the actual measured frequency to be lower than expected.

The frequencies shown in Table 1 are unimportant when SIO is in a slave mode. In the slave modes, SIO will automatically synchronize with any clock frequency up to 400 kHz.

Table 1. Serial Clock Rates

CR2	CR1	CR0	SERIAL CLOCK FREQUENCY (kHz)
0	0	0	330
0	0	1	288
0	1	0	217
0	1	1	146
1	0	0	88[1]
1	0	1	59
1	1	0	44
1	1	1	36

NOTE:

1. The clock frequency values are approximate and may vary with temperature, supply voltage, process, and SCL output loading. If normal mode I²C parameters must be strictly followed (SCL < 100kHz), it is recommended not to use CR[2:0] = 100 (SCL = 88kHz) since the clock frequency might be slightly higher than 100 kHz under certain temperature, voltage, and process conditions and use CR[2:0] = 101 (SCL = 59 kHz) instead.

The Status Register, I2CSTA: I2CSTA is an 8-bit read-only register. The three least significant bits are always zero. The five most significant bits contain the status code. There are 25 possible status codes. When I2CSTA contains F8H, no relevant state information is available and no serial interrupt is requested. All other I2CSTA values correspond to defined SIO states. When each of these states is entered, a serial interrupt is requested (SI = “1”).

More Information on SIO Operating Modes

The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfers in each mode of operation are shown in Figures 2-5. These figures contain the following abbreviations:

Abbreviation	Explanation
S	Start condition
SLA	7-bit slave address
R	Read bit (HIGH level at SDA)
W	Write bit (LOW level at SDA)
A	Acknowledge bit (LOW level at SDA)
$\bar{A}$	Not acknowledge bit (HIGH level at SDA)
Data	8-bit data byte
P	Stop condition

In Figures 2-5, circles are used to indicate when the serial interrupt flag is set. A serial interrupt is not generated when I2CSTA = F8H. This happens on a stop condition. The numbers in the circles show the status code held in the I2CSTA register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in I2CSTA is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Tables 2-6.

Master Transmitter Mode: In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 2). Before the master transmitter mode can be entered, I2CCON must be initialized as follows:

	7	6	5	4	3	2	1	0
I2CCON	AA	ENSIO	STA	STO	SI	CR2	CR1	CR0
	X	1	0	0	0	← bit rate →		

ENSIO must be set to logic 1 to enable SIO. If the AA bit is reset, SIO will not acknowledge its own slave address in the event of another device becoming master of the bus. In other words, if AA is reset, SIO cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit. The SIO logic will now test the I²C-bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (I2CSTA) will be 08H. This status code must be used to vector to an interrupt service routine that loads I2CDAT with the slave address and the data direction bit (SLA+W). The SI bit in I2CCON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in I2CSTA are possible. There are 18H, 20H, or 38H for the master mode and also 68H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 2. After a repeated start condition (state 10H). SIO may switch to the master receiver mode by loading I2CDAT with SLA+R).

Note that a master should never transmit its own slave address.

Master Receiver Mode: In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 3). The transfer is initialized as in the master transmitter mode. When the start condition has been transmitted, the interrupt service routine must load I2CDAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in I2CCON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in I2CSTA are possible. These are 40H, 48H, or 38H for the master mode and also 68H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 3. ENSIO is not affected by the serial transfer and are not referred to in Table 3. After a repeated start condition (state 10H), SIO may switch to the master transmitter mode by loading I2CDAT with SLA+W.

Note that a master should not transmit its own slave address.

Slave Receiver Mode: In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 4). To initiate the slave receiver mode, I2CADR and I2CCON must be loaded as follows:

	7	6	5	4	3	2	1	0
I2CADR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	0
	← own slave address →							

The upper 7 bits are the address to which SIO will respond when addressed by a master.

	7	6	5	4	3	2	1	0
I2CCON	AA	ENSIO	STA	STO	SI	CR2	CR1	CR0
	1	1	0	0	0	X	X	X

ENSIO must be set to logic 1 to enable SIO. The AA bit must be set to enable SIO to acknowledge its own slave address, STA, STO, and SI must be reset.

When I2CADR and I2CCON have been initialized, SIO waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for SIO to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (I) is set and a valid status code can be read from I2CSTA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 4. The slave receiver mode may also be entered if arbitration is lost while SIO is in the master mode (see status 68H).

If the AA bit is reset during a transfer, SIO will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO does not respond to its own slave address. However, the I²C-bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO from the I²C-bus.

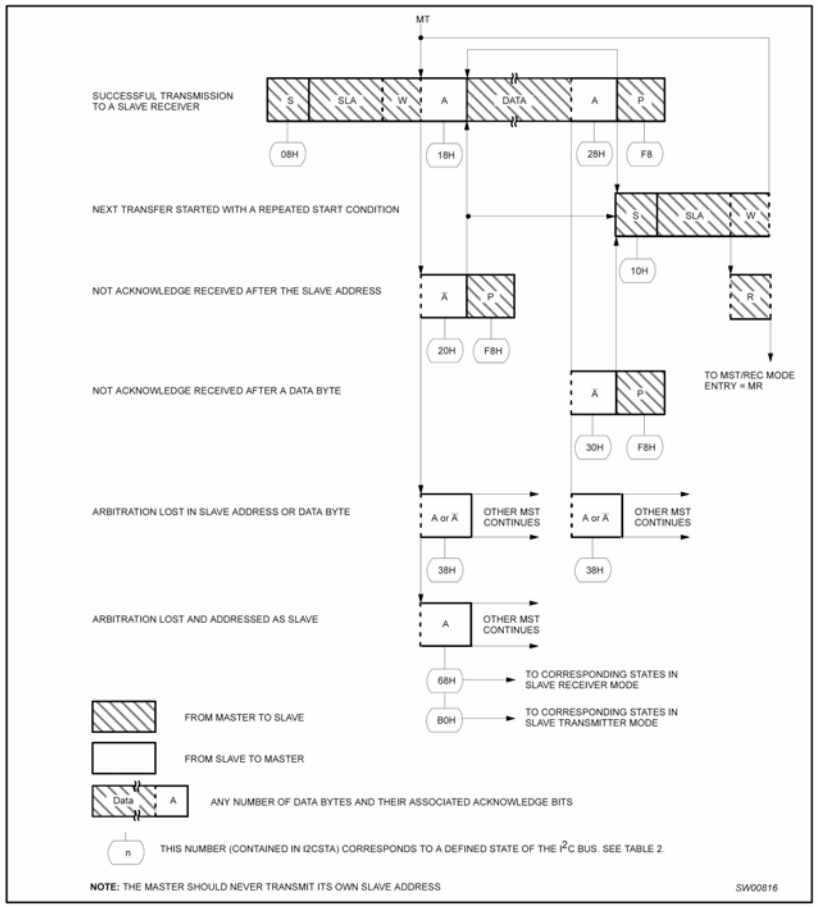


Figure 2. Format and states in the master transmitter mode

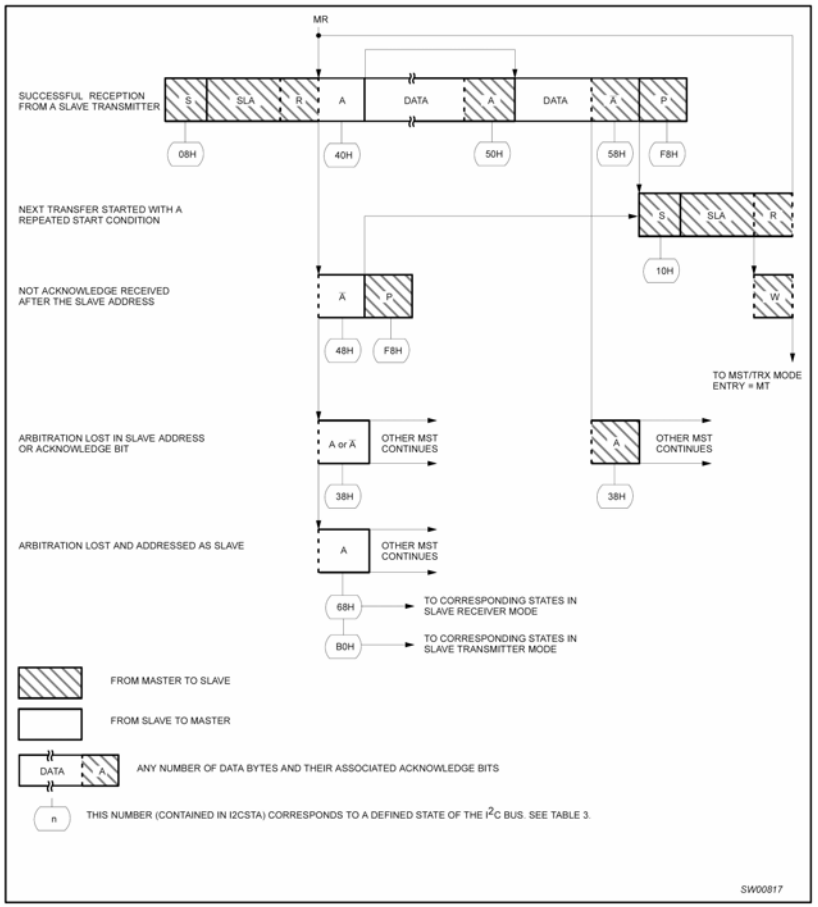


Figure 3. Format and states in the master receiver mode

Table 2. Master Transmitter Mode

STATUS CODE (I2CSTA)	STATUS OF THE I²C BUS AND SIO HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO HARDWARE
		TO/FROM I2CDAT	TO I2CCON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	X	0	X	SLA+W will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+W or	X	X	0	X	As above
		Load SLA+R	X	X	0	X	SLA+R will be transmitted; SIO will be switched to MST/REC mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
28H	Data byte in I2CDAT has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
30H	Data byte in I2CDAT has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+W or Data bytes	No I2CDAT action or	0	0	0	X	I²C-bus will be released; not addressed slave will be entered
		No I2CDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free (STOP or SCL and SDA high)

Table 3. Master Receiver Mode

STATUS CODE (I2CSTA)	STATUS OF THE I²C BUS AND SIO HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO HARDWARE
		TO/FROM I2CDAT	TO I2CCON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+R	X	X	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or	X	X	0	X	As above
		Load SLA+W	X	X	0	X	SLA+W will be transmitted; SIO will be switched to MST/TRX mode
38H	Arbitration lost in NOT ACK bit	No I2CDAT action or	0	0	0	X	I²C-bus will be released; SIO will enter a slave mode
		No I2CDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No I2CDAT action or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		no I2CDAT action	0	0	0	1	Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	No I2CDAT action or	1	0	0	X	Repeated START condition will be transmitted
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
50H	Data byte has been received; ACK has been returned	Read data byte or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		read data byte	0	0	0	1	Data byte will be received; ACK bit will be returned
58H	Data byte has been received; NOT ACK has been returned	Read data byte or	1	0	0	X	Repeated START condition will be transmitted
		read data byte or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+R	No I2CDAT action or	0	0	0	X	I²C-bus will be released; not addressed slave will be entered
		No I2CDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free

PCF8563 : Horloge temps réel

(Extrait du datasheet)

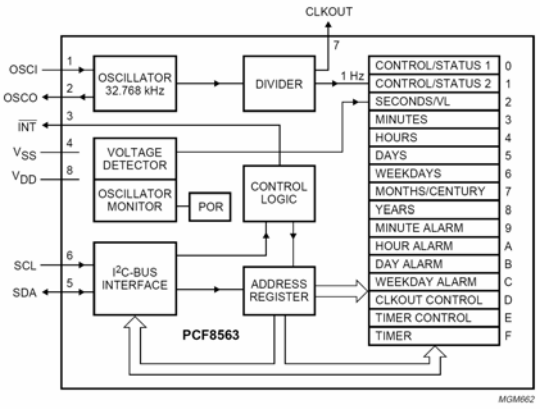
Functional description

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.

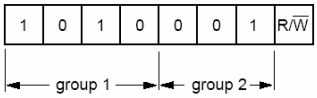
All 16 registers are designed as addressable 8-bit parallel registers although not all bits are implemented. The first two registers (memory address 00H and 01H) are used as control and/or status registers. The memory addresses 02H through 08H are used as counters for the clock function (seconds up to 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

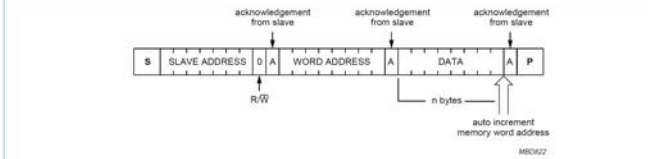


Fig 13. Master transmits to slave receiver (write mode).

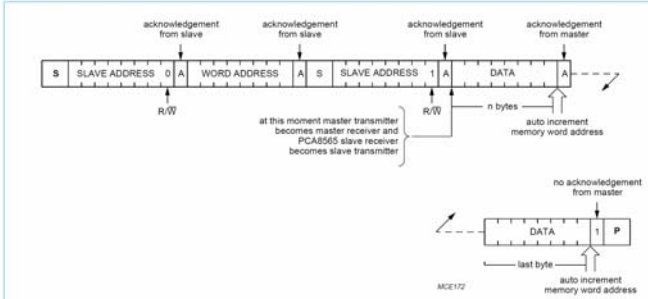


Fig 14. Master reads after setting word address (write word address; read data).

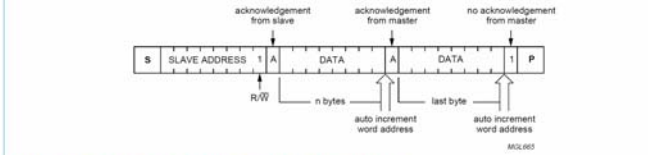


Fig 15. Master reads slave immediately after first byte (read mode).

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