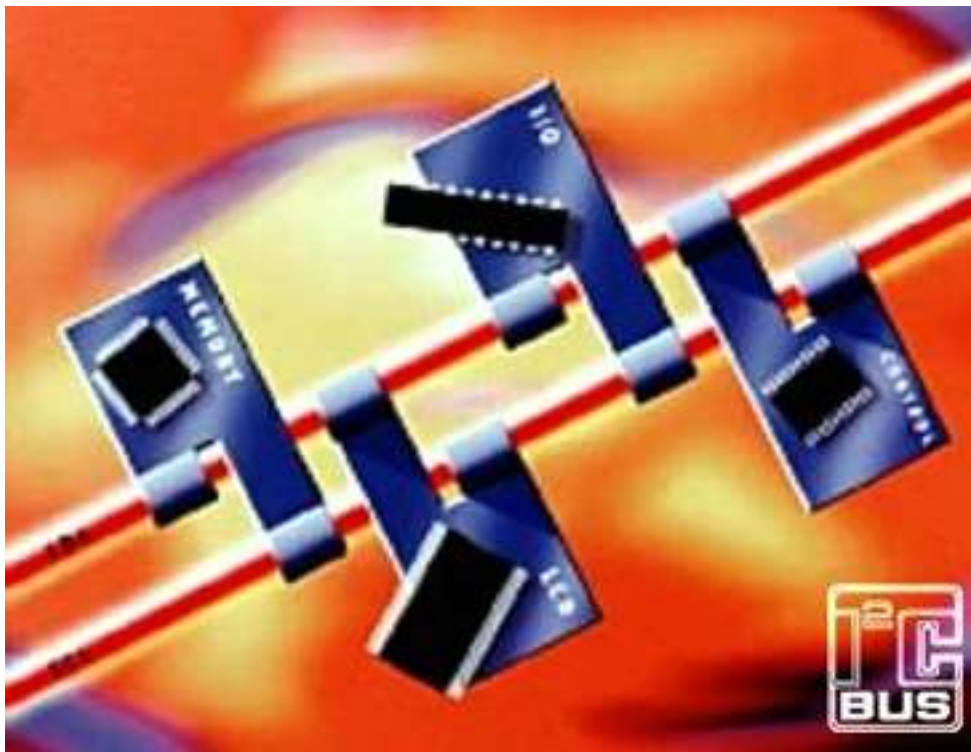


Bus I2C

XC8



Version 2.2
22 mars 2018

Exemple de programme en C

Pour μ C de type PIC16F

Frédéric GIAMARCHI
IUT de Nîmes – Centre Spatial Universitaire
Université de Montpellier

Sommaire

Le bus I2C.....	3
Généralités	3
Composants I2C.....	3
Remarques : Un PIC en esclave	3
Commandes I2C.....	4
Dialogue avec un composant I2C	6
PCF8574(A) : extension pour 8 entrées-sorties parallèles	6
PCF8591 : convertisseur AN et NA.....	7
Utilisation du CNA	7
Utilisation du CAN	7
PCF8573: horloge calendrier en temps réel	8
Enregistrement de la date et de l'heure :	8
Lecture de la date et de l'heure :	8
PCF8583: horloge calendrier avec RAM 2ko	9
Configuration du registre d'état :	9
Lecture de l'heure, des minutes et des secondes :	9
DS1621: thermomètre numérique et thermostat	10
Initialisation du composant :	10
Lecture de la température :	10
X2404 : mémoire EEPROM 4k	11
Programme d'écriture	11
Programme de lecture	11
MCP23008 : extension pour 8 bits E/S	12
Bibliographie.....	13
Annexes	14
PCF 8573	15
PCF 8574	16
PCF 8583	17
PCF 8591	18
DS 1621	19
X24C04.....	20
MCP23008	21

Le bus I2C

L'utilisation d'un bus I2C, aussi appelé liaison I2C, pour dialoguer entre composants, réduit le nombre de lignes et permet l'échange de données entre plusieurs composants de type maître et esclave. Ce que ne permet pas la liaison série.

Le bus I2C a été créé par Philips. Il est nécessaire de placer deux résistances de l'ordre de $3,3k\Omega$ sur chaque ligne en rappel au +Vcc.

Généralités

L'objet de ce document est de donner des exemples de programmation en C pour la série des pic16Fx. Le langage est le C de XC8.

Seul le mode mono-maître sera vu ici. Cela concerne donc le dialogue entre un PIC en maître et plusieurs esclaves dont certains peuvent être des PICs.

La vitesse est de 100kHz en mode standard (slow) et 400kHz en mode rapide (fast).

Composants I2C

Les composants I2C disponibles possèdent leur propre adresse. Il est possible de mettre plusieurs composants identiques sur le même bus, voir en fonction des composants.

Remarques : Un PIC en esclave

Lorsqu'un PIC est utilisé en esclave, il est plus simple d'utiliser les interruptions pour le dialogue I2C.

Commandes I2C

void I2C_Init(void)

```

{
    // paramétrage des registres MSSP en mode I2C
    TRISBbits.TRISB1 = 1;           // RB1 en entrée →SDA
    TRISBbits.TRISB4 = 1;           // RB4 en entrée →SCL

    SSPCON1 = 0x28;                 // Valide le mode I2C sur les lignes RB1 et RB4
    SSPCON2 = 0;                     // Reset du registre de contrôle MSSP
    SSPADD = 79;                     // 100kHz
    SSPSTAT = 0xC0;                 // Slew rate disable
}

```

void I2C_Idle(void)

```

{
    // Attendre que le bus I2C soit libre pour un nouveau dialogue synchrone
    while ((SSPCON2 & 0xF) || (SSPSTATbits.R_nW))
}

```

void I2C_Start(void)

```

{
    I2C_Idle();                     // Attendre que le bus I2C soit libre
    SSPCON2bits.SEN = 1;             // Start I2C transmission
    while (SSPCON2bits.SEN);         // Attend la fin de l'émission du start
}

```

void I2C_Stop(void)

```

{
    SSPCON2bits.PEN = 1;             // Stop I2C transmission
    while (SSPCON2bits.PEN);
}

```

void I2C_Write(uint8_t data)

```

{
    I2C_Idle();
    SSPBUF = data;                   // Envoie d'une donnée sur le bus I2C.
    while (SSPSTATbits.BF);         // Attend que le cycle d'écriture soit fini
}

```

uint8_t I2C_Read(void)

```
{
    I2C_Idle();
    SSPCON2bits.RCEN = 1;           // Valide le maitre en mode réception
    while (!SSPSTATbits.BF);       // Attend que le buffer soit plein
    return (SSPBUF);               // Renvoie le résultat
}
```

void I2C_Ack(void)

```
{
    SSPCON2bits.ACKDT = 0;         // Le maitre valide la réception de données
    SSPCON2bits.ACKEN = 1;        // Valide l'action
}
```

void I2C_No_Ack(void)

```
{
    SSPCON2bits.ACKDT = 1;         // Le maitre valide la dernière donnée à lire
    SSPCON2bits.ACKEN = 1;        // Valide l'action
}
```

Dialogue avec un composant I2C

PCF8574(A) : extension pour 8 entrées-sorties parallèles

Généralités

Le circuit PCF8574 est un composant CMOS qui permet le pilotage de 8 entrées ou sorties par le bus I2C.

Adressage



Adresse du composant PCF8574 : 0x40 en écriture et 0x41 en lecture, avec les trois lignes d'adresse A2, A1, et A0 à 0.

Pour le PCF8574A : 0x70 en écriture et 0x71 en lecture, avec A2, A1, et A0 à 0.

Exemple de programme pour PIC 16F

```

I2C_Start();           // Démarre un dialogue I2C
I2C_Write(0x41);      // Adresse du PCF 8574 en lecture
data = I2C_Read() & 0x01; // Lecture le contenu du composant et sélection P0
I2C_Stop();           // Fin du dialogue I2C

I2C_Start();
I2C_Write(0x40);      // Adresse du PCF 8574 en écriture
if (data == 0x01)     // Si la ligne P0 est au niveau 1
    I2C_Write(0x81);  // Mettre P7 à 1 et mettre P0 à 1
else
    I2C_Write(0x01);  // Mettre P7 = 0
I2C_Stop();

```

Remarques : particularités des lignes du composant PCF8574(A)

Le port d'Entrée/ Sortie est un port quasi-bidirectionnel ne possède qu'un registre d'écriture et de lecture. Lorsque l'on veut utiliser une broche en entrée, on commence par y écrire un « 1 » logique. Ensuite on lit le port complet et on réalise un masque pour isoler la ligne.

En sortie, il est conseillé d'utiliser le principe des sorties en collecteur ouvert.

En entrée, la ligne possède une résistance de rappel au niveau haut.

PCF8591 : convertisseur AN et NA

Généralités

Le circuit PCF8574 est un composant CMOS convertisseur AN et NA possédant 4 entrées analogiques et une sortie analogique configurables par le bus I2C.

Adressage

1	0	0	1	A2	A1	A0	R/W
---	---	---	---	----	----	----	-----

Adresse du composant PCF8591 : 0x90 en écriture et 0x91 en lecture, avec les trois lignes d'adresse A2, A1, et A0 à 0.

Registre de contrôle

0	X	X	X	0	X	X	X
---	---	---	---	---	---	---	---

Le bit 6 valide le convertisseur numérique analogique.

Les bits 4 et 5 sélectionnent la configuration des entrées.

Le bit 2 valide l'auto incrémentation du numéro des entrées analogiques afin que la lecture des entrées se fasse en continu, sans avoir besoin de choisir une entrée.

Les bits 1 et 0 sélectionnent le numéro de l'entrée analogique.

Utilisation du CNA

```

I2C_Start();
I2C_Write(0x90);           // Adresse du PCF 8591 en écriture
I2C_Write(0x40);          // Valide le Convertisseur Numérique Analogique
I2C_Write(data);          // charge la valeur dans le CNA
I2C_Stop();

```

Utilisation du CAN

```

I2C_Start();
I2C_Write(0x91);           // Adresse du PCF 8591 en lecture
data_1 = I2C_Read();       // Lecture du 1er canal analogique
data_2 = I2C_Read();       // Lecture du 2ème canal analogique
data_3 = I2C_Read();       // Lecture du 3ème canal analogique
data_4 = I2C_Read();       // Lecture du 4ème canal analogique
I2C_Stop();

```

PCF8573: horloge calendrier en temps réel

Généralités

Le composant PCF8573 remplit les fonctions d'horloge en temps réel et calendrier avec un dialogue I2C. Il possède des sorties secondes, minutes et alarme.

Adressage

1	1	0	1	0	A1	A0	R/W
---	---	---	---	---	----	----	-----

Adresse de la mémoire : 0xD0 en écriture et 0xD1 en lecture, avec les deux lignes d'adresse A1, et A0 à 0.

Enregistrement de la date et de l'heure :

```

I2C_Start();
I2C_Write(0xD0);           // Adresse du PCF 8573 en écriture
I2C_Write(0x00);           // Pointeur de mode sur le registre des heures
I2C_Write(2);              // Enregistrement de l'heure, ici 2h
I2C_Write(31);             // Enregistrement des minutes, ici 31mn
I2C_Write(22);             // Enregistrement du jour, ici le 22ème
I2C_Write(0x03);          // Enregistrement du mois, ici mars
I2C_Stop();

```

Lecture de la date et de l'heure :

```

I2C_Start();
I2C_Write(0xD0);           // Adresse du PCF 8573 en écriture
I2C_Write(0x00);           // Pointeur de mode sur le registre des heures
I2C_Stop();

I2C_Start();
I2C_Write(0xD1);           // Adresse du PCF 8573 en lecture
compt_heure = I2C_Read();   // Lecture de l'heure
compt_minut = I2C_Read();   // Lecture des minutes
compt_jour = I2C_Read();    // Lecture du jour
compt_mois = I2C_Read();    // Lecture du mois
I2C_Stop();

```


PCF8583: horloge calendrier avec RAM 2ko

Généralités

Le composant PCF8583 remplit les fonctions d'horloge temps réel, de calendrier et de compteur d'évènement avec un dialogue I2C. Il possède aussi une fonction alarme et 2ko de RAM.

Adressage

1	0	1	0	0	0	A0	R/W
---	---	---	---	---	---	----	-----

Adresse de la mémoire : 0xA0 en écriture et 0xA1 en lecture, si A0 est à 0.

Configuration du registre d'état :

```
I2C_Start();
I2C_Write(0xA0);           // Adresse du PCF 8583 en écriture
I2C_Write(0x00);           // Adresse du registre d'état
I2C_Write(0x00);           // Configuration du registre d'état
I2C_Stop();
```

Lecture de l'heure, des minutes et des secondes :

```
I2C_Start();
I2C_Write(0xA0);           // Adresse du PCF 8583 en écriture
I2C_Write(0x02);           // Pointeur sur le registre des secondes
I2C_Stop();

I2C_Start();
I2C_Write(0xA1);           // Adresse du PCF 8583 en lecture
compt_sec = I2C_Read();     // Lecture des secondes (format BCD)
compt_min = I2C_Read();     // Lecture des minutes (format BCD)
compt_heu = I2C_Read();     // Lecture de l'heure (format BCD)
I2C_Stop();
```

DS1621: thermomètre numérique et thermostat

Généralités

Le composant DS1621 remplit les fonctions de thermomètre et thermostat avec un dialogue I2C. Il possède une sortie pour thermostat.

Adressage

1	0	0	1	A2	A1	A0	R/W
---	---	---	---	----	----	----	-----

Adresse de la mémoire : 0x90 en écriture et 0x91 en lecture, avec les trois lignes d'adresse A2, A1 et A0 à 0.

Initialisation du composant :

```

I2C_Start();
I2C_Write(0x90);           // Adresse du DS1621 en écriture
I2C_Write(0xAC);          //
I2C_Write(0x09);          // Conversion en continu
I2C_Stop();

I2C_Start();
I2C_Write(0x90);
I2C_Write(0xEE);          // Lance les mesures
I2C_Stop();

```

Lecture de la température :

```

I2C_Start();
I2C_Write(0x90);           // Adresse du DS1621 en écriture
I2C_Write(0xAA);          // Demande de lecture de la température
I2C_Stop();

I2C_Start();
I2C_Write(0x91);           // Adresse du DS1621 en lecture
temp_hi = I2C_Read();      // Lecture de l'octet de poids fort
temp_lo = I2C_Read();      // Lecture de l'octet de poids faible
I2C_Stop();
delayMs(1000);             // temps de conversion à respecter

```

Remarques

Il est nécessaire de respecter le temps de conversion de 1 seconde.

X2404 : mémoire EEPROM 4k

Généralités (4k = 512 x 8)

Le circuit X2404 est une mémoire EEPROM de 512 octets pour le bus I2C.

Adressage

1	0	1	0	A1	A0	PA	R/W
---	---	---	---	----	----	----	-----

Adresse de la mémoire : 0xA0 en écriture et 0xA1 en lecture, avec les deux lignes d'adresse A1, et A0 à 0. Les 2 blocs (ou pages) sont adressables par le bit PA.

Programme d'écriture

```

I2C_Start();
I2C_Write(0xA0);           // Adresse du X2404 en écriture page 0
I2C_Write(0x00);          // Début des adresses mémoire à écrire
for(i=0 ;i<255 ;i++)
    I2C_Write(0xFF);      // Donnée à écrire pour effacer la mémoire
I2C_Stop();
delayMs(11);              // Tempo en fonction de la mémoire, voir data sheet

```

Programme de lecture

```

I2C_Start();
I2C_Write(0xA0);
I2C_Write(0x00);          // Début des adresses mémoire à lire
I2C_Stop();

I2C_Start();
I2C_Write(0xA1);          // Adresse du X2404 en lecture page 0
for(i=0 ;i<10 ;i++)
    buffer[i] = I2C_Read(); // Les données lues sont enregistrées dans une table.
I2C_Stop();

```

MCP23008 : extension pour 8 bits E/S

Généralités

Le composant MCP23008 et sa version 16bits (MCP23017) est une extension (expander) pour 8 lignes en entrées ou sorties avec une interface série I2C. Il accepte les vitesses classiques 100kHz et 400kHz, mais aussi 1,7MHz. On peut placer 8 composants identiques sur le même bus en choisissant ses adresses. Il peut remplacer le PCF8574 et ils ont les mêmes adresses.

Adressage

0	1	0	0	A2	A1	A0	R/W
---	---	---	---	----	----	----	-----

Adresse du composant en écriture : 0x40 et en lecture 0x41, avec les trois lignes d'adresse A2, A1, et A0 à 0.

Exemple de programme pour PIC 16F

```
I2C_Start();           // Démarre un dialogue I2C
I2C_Write(0x40);       // Adresse du MCP23008 en écriture
I2C_Write(0x0A);       // Registre des sorties
I2C_Write(0x01);       // Bit 0 au niveau 1
I2C_Stop();           // Fin du dialogue I2C
```

```
I2C_Start();
I2C_Write(0x40);
I2C_Write(0x09);       // Registre des entrées
I2C_Stop();
```

```
I2C_Start();           // Relance le dialogue I2C
I2C_Write(0x41);       // Adresse du MCP23008 en lecture
data = I2C_Read();     // Lecture du registre des entrées
I2C_Stop();
```

Remarques :

Ce composant possède plusieurs registres qui permettent beaucoup de possibilités. Il est conseillé de consulter le data sheet pour paramétrer au mieux de votre application (résistances de pull-up, interruption, comparaison, ...).

Bibliographie

Le bus I2C : communication inter-IC	Elektor n° xxx (janvier 1991)
Interface I2C encartable pour PC	Elektor n° 163 (janvier 1992)
Convertisseur A/N-N/A et E/S pour I2C	Elektor n° 164 (février 1992)
Module à afficheurs 7 segments à LED	Elektor n° 165 (mars 1992)
I2C power switch	Elektor n° 187 (janvier 1994)
Affichage alphanumérique I2C	Elektor n° 188 (février 1994)
Prolongateur de bus I2C	Elektor n° 192 (juin 1994)
Interface parallèle I2C	Elektor n° 210 (décembre 1995)

Annexes



PCF 8573 : horloge/calendrier en temps réel avec alarme

PCF 8574 : extension pour 8 entrées-sorties parallèles.

PCF 8583 : horloge/calendrier avec RAM 2ko

PCF 8591 : convertisseur AN et NA

X24C04 : mémoire EEPROM 4ko

DS 1621 : thermomètre numérique et thermostat

Philips
Semiconductors



PHILIPS

PCF 8573

Clock/calendar with serial I/O

PCF8573

5 BLOCK DIAGRAM

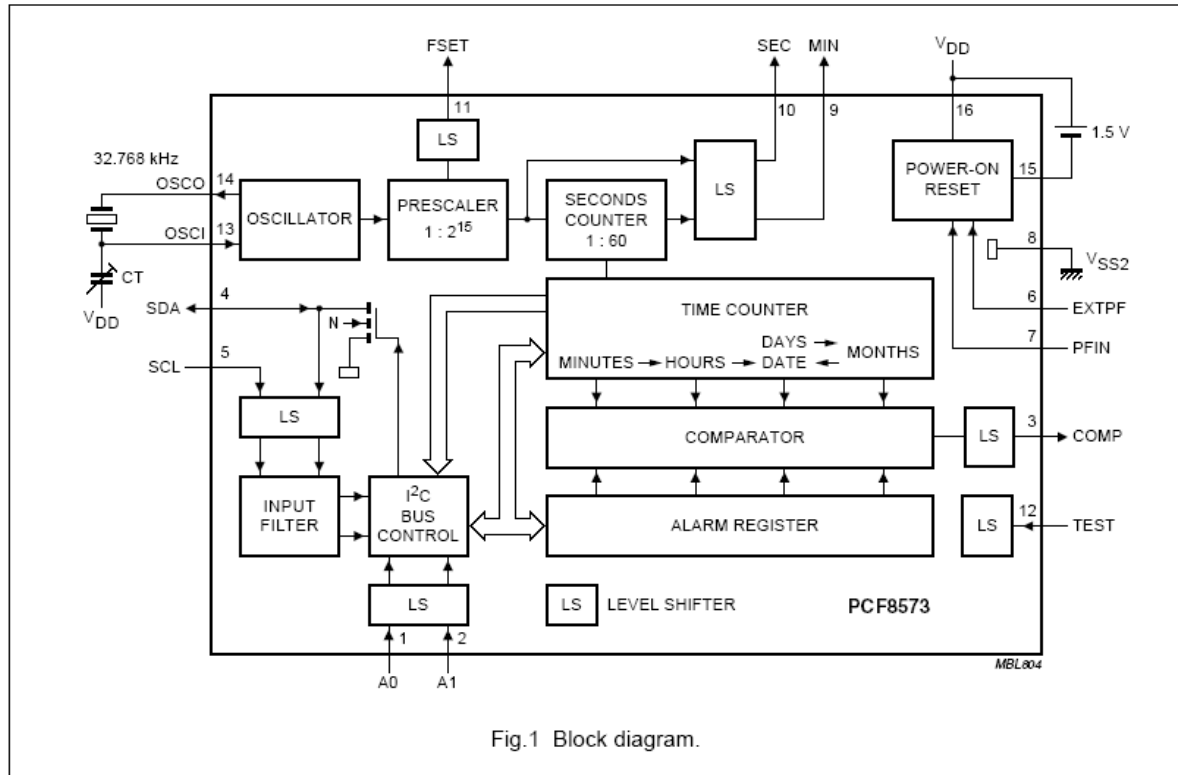


Fig.1 Block diagram.

6 PINNING

SYMBOL	PIN	DESCRIPTION
A0	1	address input
A1	2	address input
COMP	3	comparator output
SDA	4	serial data line; I ² C-bus
SCL	5	serial clock line; I ² C-bus
EXTPF	6	enable power fail flag input
PFIN	7	power fail flag input
V _{SS2}	8	negative supply 2 (I ² C interface)
MIN	9	one pulse per minute output
SEC	10	one pulse per second output
FSET	11	oscillator tuning output
TEST	12	test input; connect to V _{SS2} if not in use
OSCI	13	oscillator input
OSCO	14	oscillator input/output
V _{SS1}	15	negative supply 1 (clock)
V _{DD}	16	common positive supply

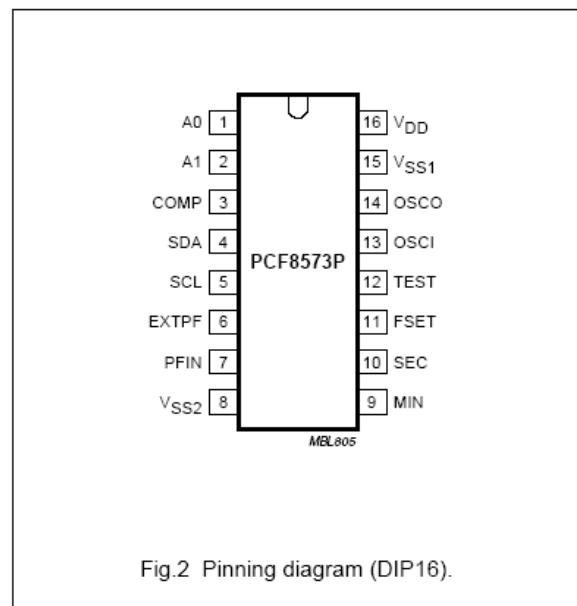


Fig.2 Pinning diagram (DIP16).

PCF 8574

Remote 8-bit I/O expander for I²C-bus

PCF8574

4 BLOCK DIAGRAM

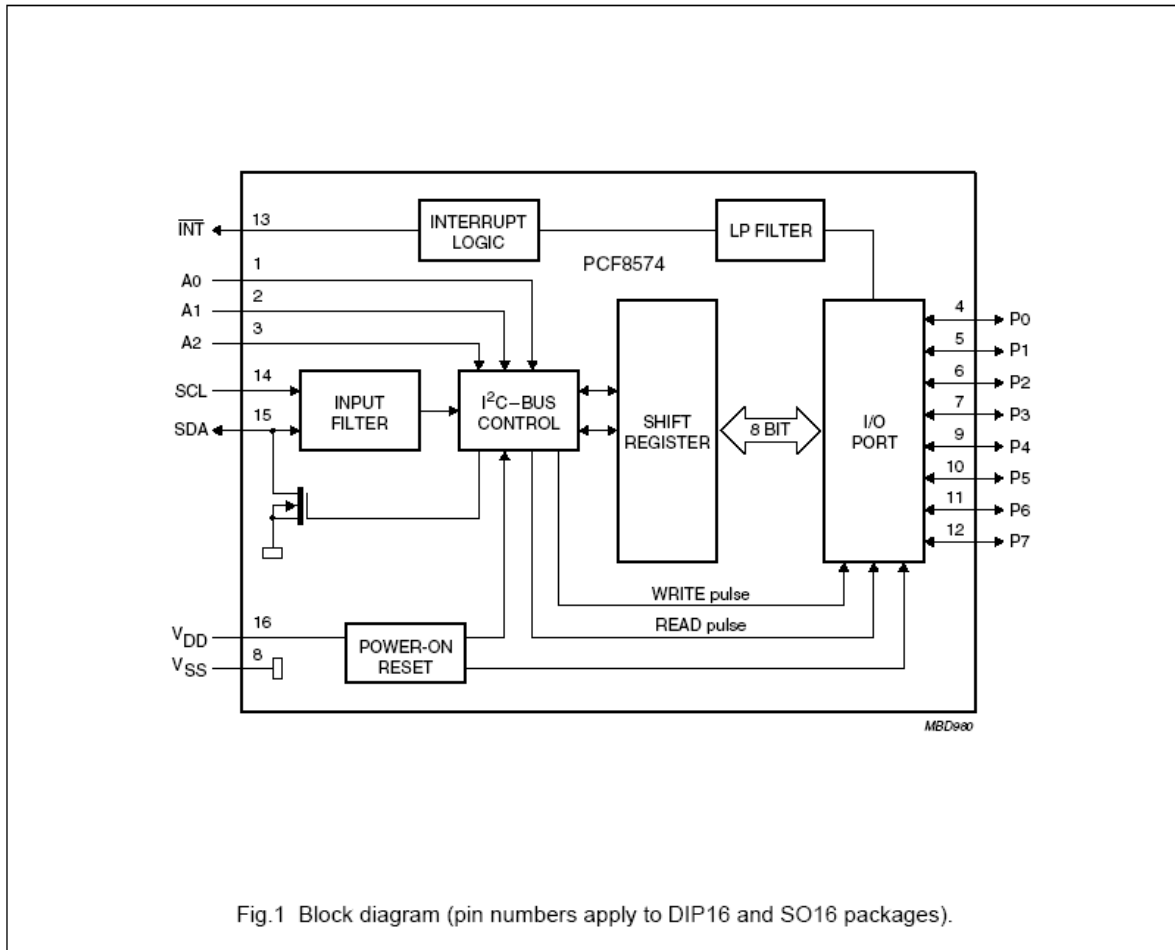


Fig.1 Block diagram (pin numbers apply to DIP16 and SO16 packages).

PCF 8583

Philips Semiconductors

Product specification

Clock/calendar with 240 × 8-bit RAM

PCF8583

5 BLOCK DIAGRAM

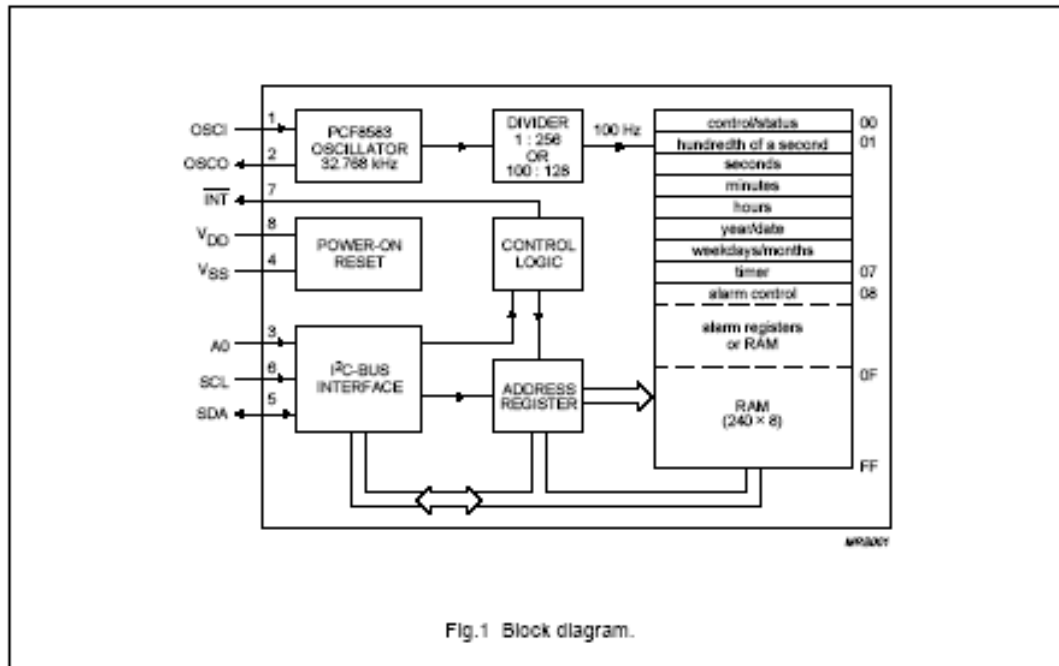


Fig.1 Block diagram.

6 PINNING

SYMBOL	PIN	DESCRIPTION
OSCI	1	oscillator input, 50 Hz or event-pulse input
OSCO	2	oscillator output
AD	3	address input
V _{SS}	4	negative supply
SDA	5	serial data line
SCL	6	serial clock line
INT	7	open drain interrupt output (active LOW)
V _{DD}	8	positive supply

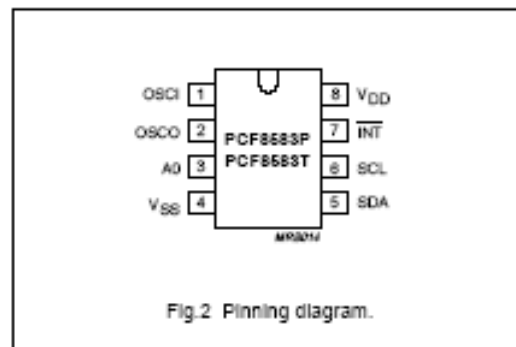


Fig.2 Pinning diagram.

PCF 8591

Philips Semiconductors

Product specification

8-bit A/D and D/A converter

PCF8591

5 BLOCK DIAGRAM

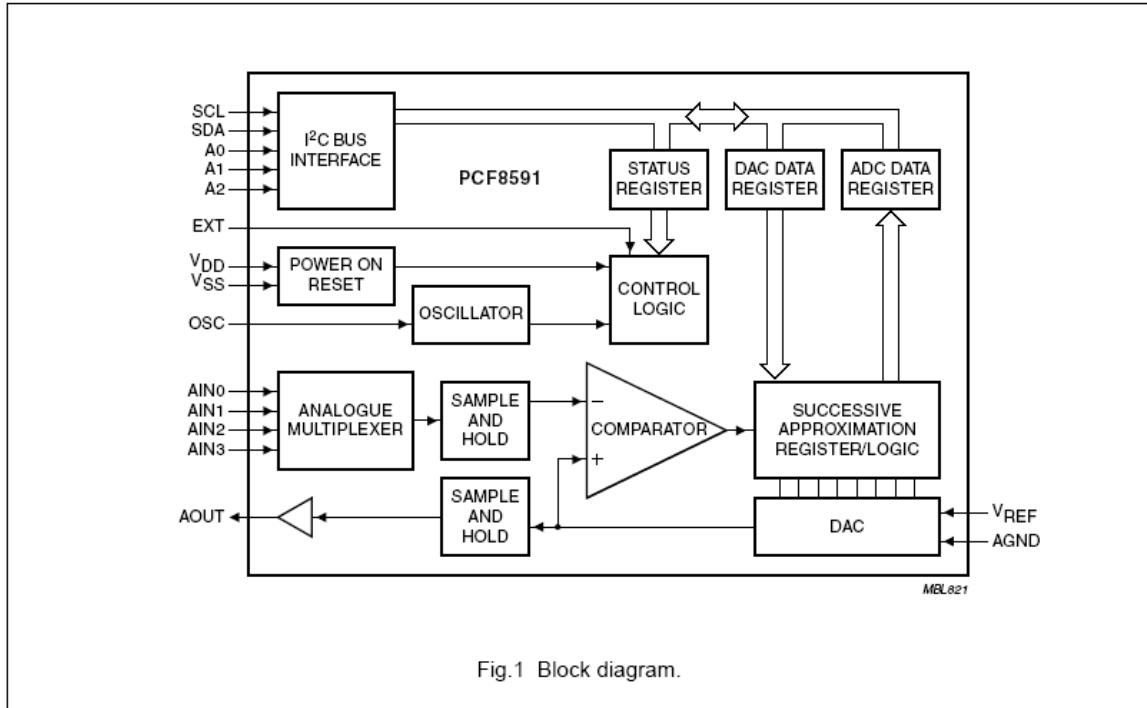


Fig.1 Block diagram.

6 PINNING

SYMBOL	PIN	DESCRIPTION
AIN0	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware address
A1	6	
A2	7	
V _{SS}	8	negative supply voltage
SDA	9	I ² C-bus data input/output
SCL	10	I ² C-bus clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground
V _{REF}	14	voltage reference input
AOUT	15	analog output (D/A converter)
V _{DD}	16	positive supply voltage

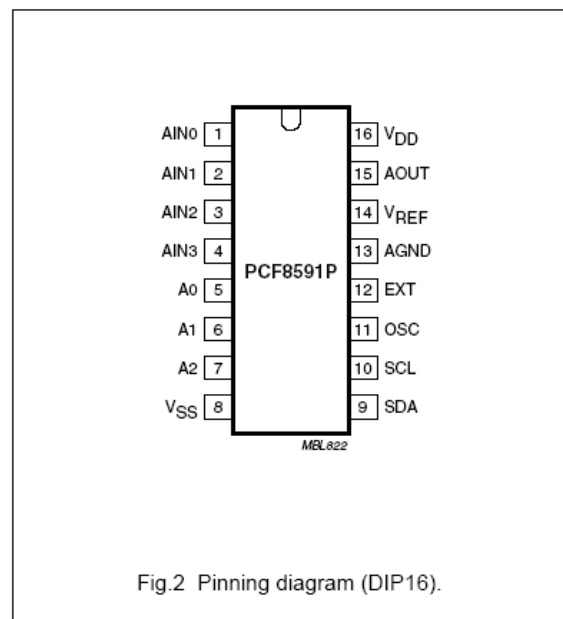


Fig.2 Pinning diagram (DIP16).

DS 1621

DALLAS
SEMICONDUCTOR
DS1621
 Digital Thermometer and Thermostat

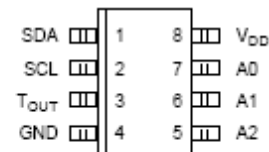
FEATURES

- Temperature measurements require no external components
- Measures temperatures from -55°C to $+125^{\circ}\text{C}$ in 0.5°C increments. Fahrenheit equivalent is -67°F to 257°F in 0.9°F increments
- Temperature is read as a 9-bit value (two byte transfer)
- Wide power supply range (2.7V to 5.5V)
- Converts temperature to digital word in 1 second
- Thermostatic settings are user definable and nonvolatile
- Data is read from/written via a 2-wire serial interface (open drain I/O lines)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermal sensitive system.
- 8-pin DIP or SOIC package (150 MIL and 208 MIL)

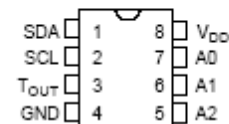
DESCRIPTION

The DS1621 digital thermometer and thermostat provides 9-bit temperature readings which indicate the temperature of the device. The thermal alarm output, T_{OUT} , is active when the temperature of the device exceeds a user-defined temperature TH. The output remains active until the temperature drops below user defined temperature TL, allowing for any hysteresis necessary.

PIN ASSIGNMENT



DS1621S 8-PIN SOIC (150 MIL)
 DS1621V 8-PIN SOIC (208 MIL)
 See Mech. Drawings Section



DS1621
 8-PIN DIP (300 MIL)
 See Mech. Drawings Section

PIN DESCRIPTION

SDA	– 2-Wire Serial Data Input/Output
SCL	– 2-Wire Serial Clock
GND	– Ground
T_{OUT}	– Thermostat Output Signal
A0	– Chip Address Input
A1	– Chip Address Input
A2	– Chip Address Input
V_{DD}	– Power Supply Voltage

User defined temperature settings are stored in non-volatile memory, so parts may be programmed prior to insertion in a system. Temperature settings, and temperature readings are all communicated to/from the DS1621 over a simple 2-wire serial interface.

X24C04

4K

X24C04

512 x 8 Bit

Serial E²PROM

FEATURES

- 2.7V to 5.5V Power Supply
- Low Power CMOS
 - Active Read Current Less Than 1 mA
 - Active Write Current Less Than 3 mA
 - Standby Current Less Than 50 μ A
- Internally Organized 512 x 8
- 2 Wire Serial Interface
 - Bidirectional Data Transfer Protocol
- Sixteen Byte Page Write Mode
 - Minimizes Total Write Time Per Byte
- Self Timed Write Cycle
 - Typical Write Cycle Time of 5 ms
- High Reliability
 - Endurance: 100,000 Cycles
 - Data Retention: 100 Years
- 8 Pin Mini-DIP, 8 Pin SOIC and 14 Pin SOIC Packages

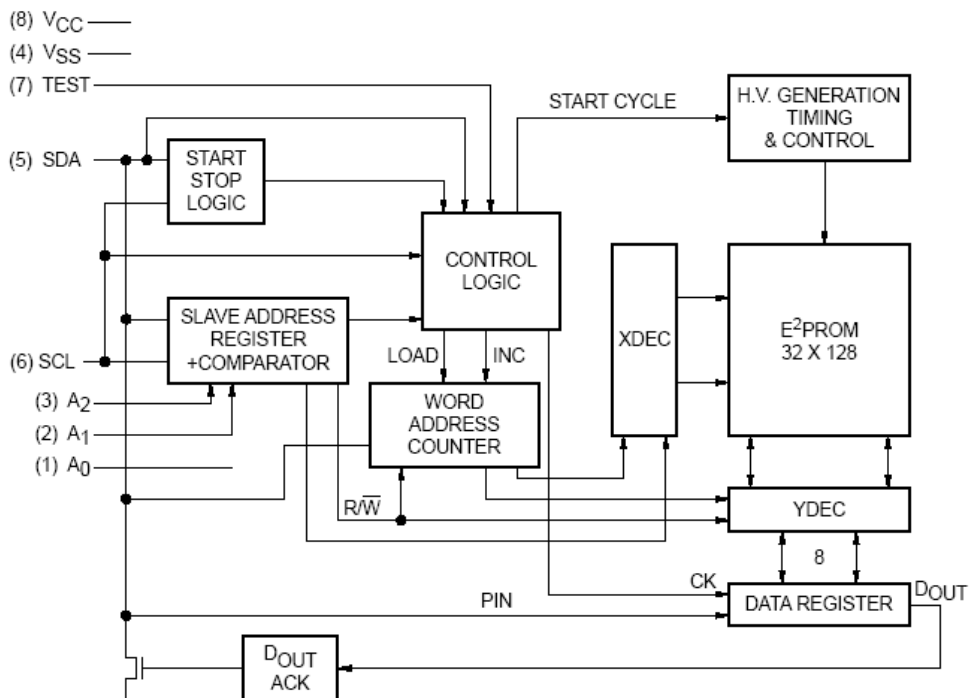
DESCRIPTION

The X24C04 is a CMOS 4096 bit serial E²PROM, internally organized 512 x 8. The X24C04 features a serial interface and software protocol allowing operation on a simple two wire bus.

The X24C04 is fabricated with Xicor's advanced CMOS Textured Poly Floating Gate Technology.

The X24C04 utilizes Xicor's proprietary DirectWrite™ cell providing a minimum endurance of 100,000 cycles and a minimum data retention of 100 years.

FUNCTIONAL DIAGRAM



3839 FHD F01

DirectWrite™ is a trademark of Xicor, Inc.

MCP23008



MICROCHIP MCP23008/MCP23S08

8-Bit I/O Expander with Serial Interface

Features

- 8-bit remote bidirectional I/O port
 - I/O pins default to input
- High-speed I²C™ interface (MCP23008)
 - 100 kHz
 - 400 kHz
 - 1.7 MHz
- High-speed SPI interface (MCP23S08)
 - 10 MHz
- Hardware address pins
 - Three for the MCP23008 to allow up to eight devices on the bus
 - Two for the MCP23S08 to allow up to four devices using the same chip-select
- Configurable interrupt output pin
 - Configurable as active-high, active-low or open-drain
- Configurable interrupt source
 - Interrupt-on-change from configured defaults or pin change
- Polarity Inversion register to configure the polarity of the input port data
- External reset input
- Low standby current: 1 μA (max.)
- Operating voltage:
 - 1.8V to 5.5V @ -40°C to +85°C
I²C @ 100 kHz
SPI @ 5 MHz
 - 2.7V to 5.5V @ -40°C to +85°C
I²C @ 400 kHz
SPI @ 10 MHz
 - 4.5V to 5.5V @ -40°C to +125°C
I²C @ 1.7 kHz
SPI @ 10 MHz

Packages

- 18-pin PDIP (300 mil)
- 18-pin SOIC (300 mil)
- 20-pin SSOP
- 20-pin QFN

Block Diagram

