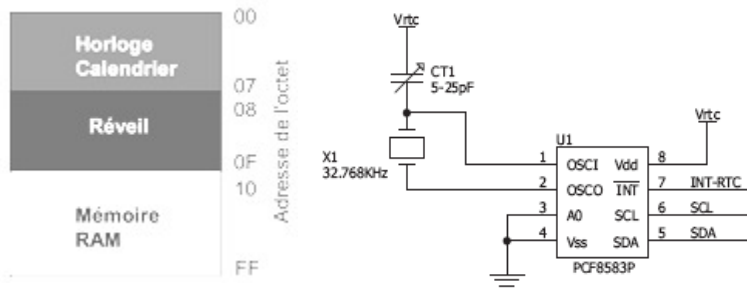


**TP5 M2 ESE BUS I2C en MicoC PRO avec affichage USTO 2020**

**HORLOGE TEMPS RÉEL PCF 8583P**

Le PCF8583 est une horloge avec calendrier combinée à une RAM de 240 octets. Les années sont gérées sur 4 ans, Il possède aussi une fonction alarme avec une sortie d'interruption. L'horloge de référence peut être un quartz d'horloge cadencé à 32,768 kHz



La communication avec le composant se fait par une liaison I<sup>2</sup>C, en lecture ou en écriture. Chaque information est contenue dans un octet. La mémoire interne est divisée en registres (octets) auxquels on peut accéder soit directement soit de manière séquentielle grâce à l'auto incrémentation du compteur d'adresse interne. Schématiquement on peut voir trois zones : the first 8 bytes of the RAM are used for the clock, calendar, and counter functions. Si la fonction alarme n'est pas utilisée, la zone mémoire correspondante peut être utilisée en tant que RAM d'usage général. Lors d'un accès direct, l'utilisateur peut s'adresser à un octet particulier du composant, il doit donc indiquer à chaque fois l'adresse de cet octet lors de la communication par I<sup>2</sup>C. L'accès séquentiel permet de profiter de l'auto-incrémentation du compteur d'adresse interne pour ne donner que l'adresse de départ de la suite d'octets avec lesquels on souhaite dialoguer. le registre d'adresse est incrémenté automatiquement après chaque lecture ou écriture. Les valeurs numériques concernant le temps ne sont pas données sous forme binaire mais sous forme BCD: on utilise la fonction transforme pour faire la conversion

```
char seconds, minutes, hours, day, month,
year; // variables Globales date/temps
// Software I2C connections C3 et C4
sbit Soft_I2C_Scl at RC3_bit;
sbit Soft_I2C_Sda at RC4_bit;
sbit Soft_I2C_Scl_Direction at TRISC3_bit;
sbit Soft_I2C_Sda_Direction at TRISC4_bit;
// fin Software I2C connections
```

```
// LCD module connections: port B
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// fin LCD module connections
```

```
//lecture temps et date du RTC (PCF8583P)
void Read_Time() {
  Soft_I2C_Start(); // signal start
  Soft_I2C_Write(0xA0); // Adresse ecriture PCF8583,
  Soft_I2C_Write(2); // debuter a adresse 2
  // pointeur sur le registre des secondes.
  Soft_I2C_Start(); // signal repeated start semblable a Start
  Soft_I2C_Write(0xA1); // Adresse pour lecture (R/W=1)
  seconds = Soft_I2C_Read(1); // lire octet seconds en BCD
  minutes = Soft_I2C_Read(1); // lire octet minutes
  hours = Soft_I2C_Read(1); // lire octet heures
  day = Soft_I2C_Read(1); // lire octet année/jour
  month = Soft_I2C_Read(0); // lire octet mois
  Soft_I2C_Stop(); // signal stop
}
//Formats: date et horloge du BCD au décimal
void Transform_Time() {
  seconds = ((seconds & 0xF0) >> 4)*10 + (seconds & 0x0F);
  minutes = ((minutes & 0xF0) >> 4)*10 + (minutes & 0x0F);
  hours = ((hours & 0xF0) >> 4)*10 + (hours & 0x0F);
  year = (day & 0xC0) >> 6;
  day = ((day & 0x30) >> 4)*10 + (day & 0x0F);
  month = ((month & 0x10) >> 4)*10 + (month & 0x0F);
}
```

```
//- Affichage LCD: decimal vers ASCII
```

```
void Display_Time() {
    Lcd_Chr(1, 6, (day / 10) + 48);
    Lcd_Chr(1, 7, (day % 10) + 48);
    Lcd_Chr(1, 9, (month / 10) + 48);
    Lcd_Chr(1, 10, (month % 10) + 48);
    Lcd_Chr(1, 15, year + 48);
    Lcd_Chr(2, 6, (hours / 10) + 48);
    Lcd_Chr(2, 7, (hours % 10) + 48);
    Lcd_Chr(2, 9, (minutes / 10) + 48);
    Lcd_Chr(2, 10, (minutes % 10) + 48);
    Lcd_Chr(2, 12, (seconds / 10) + 48);
    Lcd_Chr(2, 13, (seconds % 10) + 48);
}
```

```
//----- initialisation
```

```
void Init_Main() {
    TRISB = 0;
    PORTB = 0xFF;
    TRISA = 0xFF;
    ADCON1 = 0xFF;
    // Configure les E/S en digital
    Soft_I2C_Init(); // Initialise Soft I2C
    Lcd_Init();
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Cmd(LCD_CURSOR_OFF);
}
```

```
Lcd_Out(1, 1, "Date:");
Lcd_Chr(1, 8, ':');
Lcd_Chr(1, 11, ':');
Lcd_Chr(1, 16, ':');
Lcd_Out(2, 1, "Time:");
Lcd_Chr(2, 8, ':');
Lcd_Chr(2, 11, ':');
Lcd_Out(1, 12, "201");
}
```

```
//----- procedure Main
```

```
void main() {
    Delay_ms(500);
    Init_Main(); // effectue initialisation
    while (1) { // loop infinie
        Read_Time(); // lecture du temps du RTC
        Transform_Time();
        Display_Time();
    }
}
```

**Note :** Adressage 1 0 1 0 0 0 **A0** R/W

Adresse du PCF8583 : 0xA0 en écriture et 0xA1 en lecture, si l'entrée du circuit RTC **A0** est à 0.

Oscillator: External Clock, 8.0000 MHz

Ext. Modules: **RTC Board** on PORTC

- Pull-up I2C lignes de communication (RC3 et RC4)

```
//initialization du RTC a des valeurs arbitraires
```

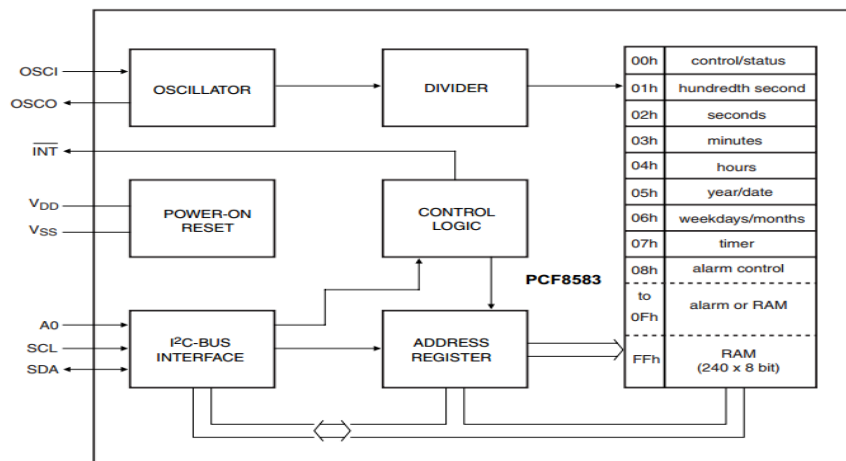
```
void main() {
    //Delay_ms(1000);
    Soft_I2C_Init(); // Initialize full master mode
    Soft_I2C_Start(); // Issue start signal
    Soft_I2C_Write(0xA0); // Address PCF8583,
    Soft_I2C_Write(0); // Start from address 0
    //(configuration memory location)
    Soft_I2C_Write(0x80); // Write 0x80 to
    //(configuration memory location (pause counter...))
    Soft_I2C_Write(0); // Write 0 to cents
    Soft_I2C_Write(0); // Write 0 to seconds
    Soft_I2C_Write(0x30); // Write 0x30 to minutes
    Soft_I2C_Write(0x12); // Write 0x12 to hours
    Soft_I2C_Write(0x18); // Write 0x18 to year/date
    Soft_I2C_Write(0x04); // Write 0x04 to /month
    Soft_I2C_Stop(); // stop signal
    Soft_I2C_Start(); // Issue start signal
    Soft_I2C_Write(0xA0); // Address PCF8583
    Soft_I2C_Write(0); // Start from address 0
    Soft_I2C_Write(0); // Write 0 to configuration
    //(enable counting)
    Soft_I2C_Stop(); // stop signal
}
```

- placer jumper J3 on position haute sur la carte  
- placer en position off PORTC LEDs (SW6 ). Placer led on (SW8)

- adresse pin A0 du PCF8583 doit être mise à 0V. (mikroElektronika RTC module ceci est fait par défaut)

Les Adresses et données sont transférées en série via les deux lignes bidirectionnelles du bus I2C.

Ce programme est développé en supposant I2C du PIC programmable.



### Compte rendu

1. Tracer les différents signaux SDA et SCL échangés entre le PIC et le module PCF8583
  2. Ecrivez un programme identique en prenant en considération le module I2C qui existe sous forme de matériel dans le PIC16F877 en utilisant le compilateur MICROCPRO
  - 3- Etudier la fiche technique du composant PCF8583P et résumer son fonctionnement
- Proposer une application en utilisant ce circuit