

## Projet Enregistreur de température

### 1/ Principe :

Cet enregistreur est géré par un micro contrôleur PIC 18F2420 de Microchip.

La température est captée par un capteur DS18B20 Dallas ayant une plage de mesure -55 à 125° et une résolution de 0.0625 °C.

La base de temps d'acquisition est paramétrable à 5 s. 15 s. 30 s. 1 mn, 10 mn, 15 mn, 30 mn et 1 heure avec les 8 combinaisons des boutons B0, B1 et B3 et est sauvegardée dans l'eprom du PIC.

L'horloge RTC PCF 8583 permet d'horodater les enregistrements sous le format jj/mm/aaaa hh/mm/ss.

La mise à l'heure est assurée par les 4 boutons B0, B1, B2 et B3.

Un module lecteur enregistreur de carte permet d'archiver les acquisitions sur une carte SD de 2Go

Les données sont stockées dans un fichier texte "F1.Txt", ce qui permet l'exploitation avec une application spécifique sous Windows ou avec Excel.

Un enregistrement comprend : N°, date, heure, température.

La visualisation est assurée par un écran LCD 2x16 caractères qui affiche l'heure, la température, le nombre d'enregistrements et la base de temps

Un voyant à Led s'allume 3 s. avant chaque enregistrement et s'éteint 1 seconde après l'enregistrement.

Ceci permet d'arrêter l'appareil en dehors de la phase d'écriture sur la carte SD.

La carte est alimentée par un bloc d'alimentation 230V AC – 9V DC classique.

La consommation est de 0.6 wh soit 0.002 €/jour

### 2/ Mode d'emploi :

Insérer une carte SD dans la fente à l'arrière de l'appareil.

Brancher le bloc d'alimentation dans une prise de courant 230V.

Mise en service par l'interrupteur en haut à gauche.

L'interrupteur situé à coté de l'afficheur permet d'allumer ou d'éteindre le rétro éclairage.

Au message "Départ bouton bleu" appuyer le bouton bleu pour lancer l'enregistrement.

Pour arrêter et retirer la carte SD mettre l'interrupteur sur arrêt.

Ne pas arrêter l'appareil quand le voyant rouge est allumé.

Pour changer la base de temps, appuyer le bouton bleu et mettre en service l'appareil.

Appuyer sur les boutons marqués 1.

La base de temps s'affiche sur l'écran.

Pour sauvegarder appuyer le bouton bleu tout en maintenant les boutons de sélection.

B2	B1	B0	Temps
0	0	0	5 s.
0	0	1	15 s.
0	1	0	30 s.
0	1	1	1 mn.
1	0	0	10 mn.
1	0	1	15 mn.
1	1	0	30 mn.
1	1	1	1 h.

Pour régler l'horloge et le calendrier, maintenir B3 appuyé et mettre en marche l'appareil.

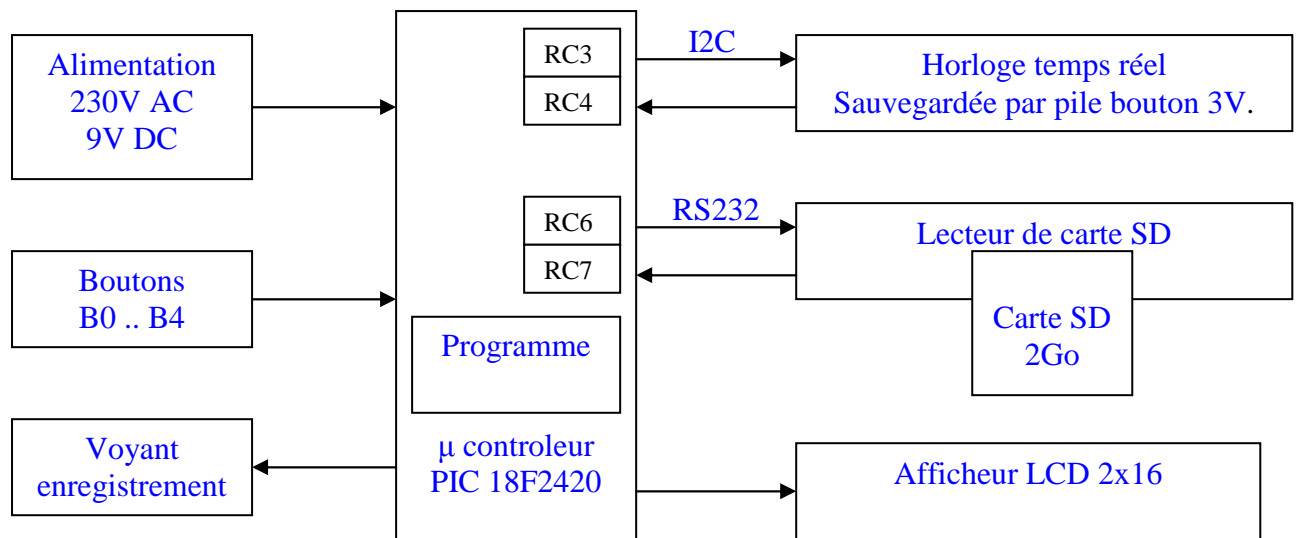
Appuyer B3 et B0 pour naviguer dans les valeurs jj, mm, aa, jj, mm.

B1 incrémente et B2 décrémente la valeur sélectionnée.

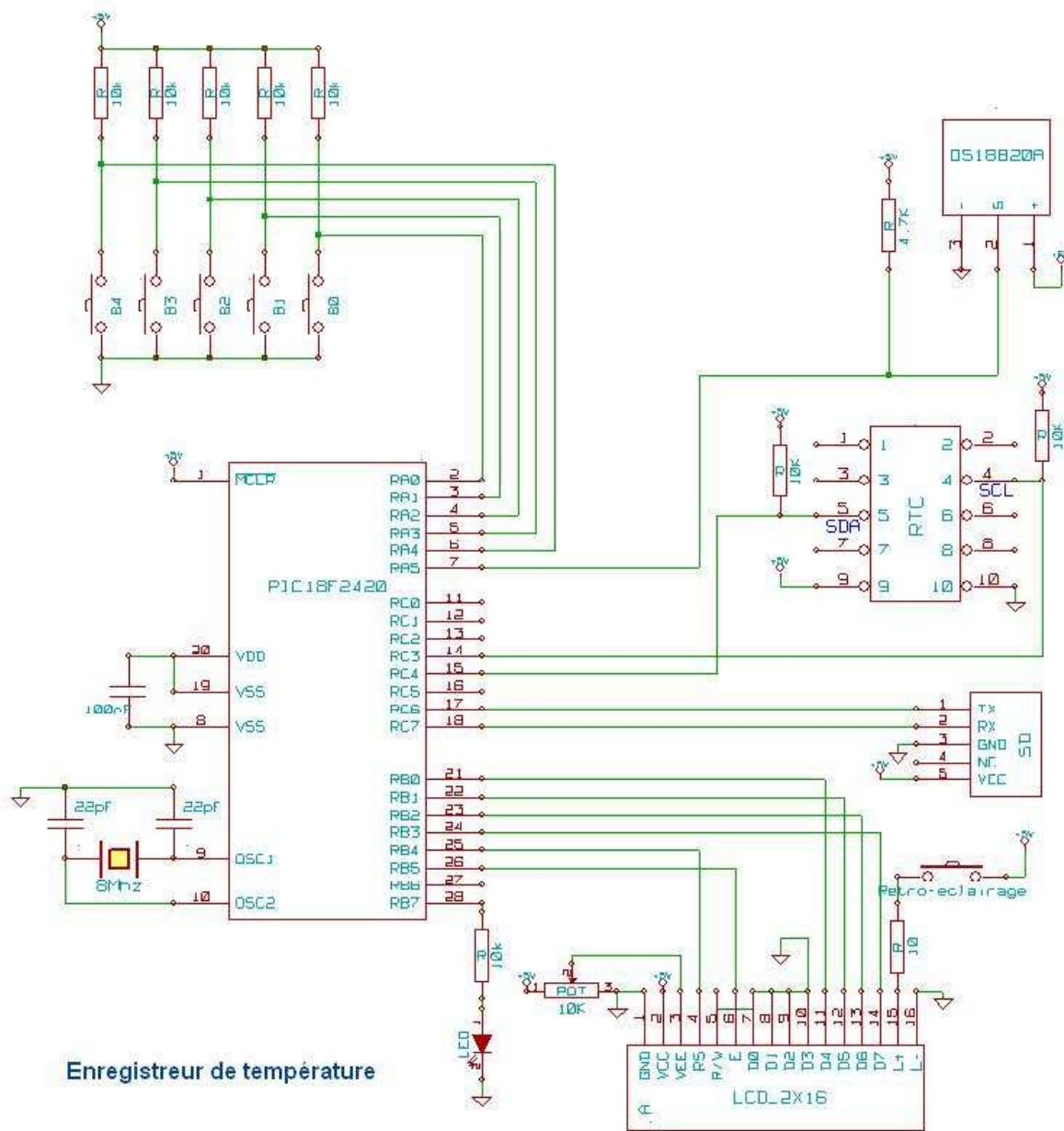
Pour valider appuyer B0 et B3, puis appuyer sur le bouton bleu.

Pour quitter sans valider, appuyer B3 et le bouton bleu.

### 3/ Schéma bloc :

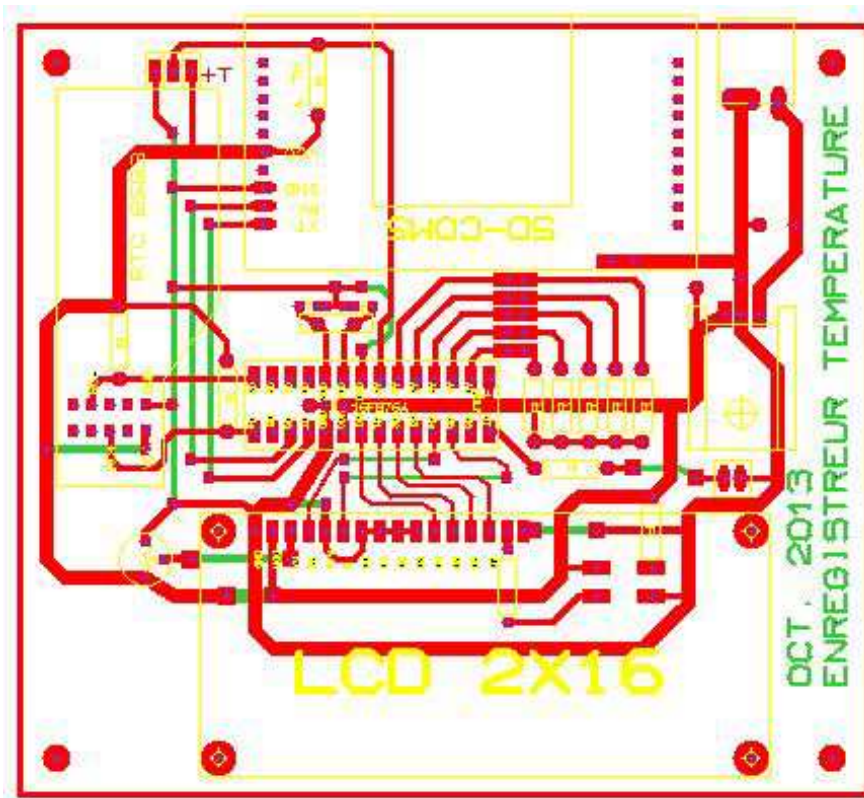


### 4/ Schéma électronique :



Enregistreur de température

**5/ Circuit imprimé :**



## 6/ Programme embarqué dans le micro contrôleur :

### **program RelTemp1**

'Enregistreur de T° avec stockage sur carte SD (PIC18F2420, RTC et PCF8583 le 29/11/2013

'Fréquence d'acquisitionsauvegardée dans l'eeprom du Pic.

'Sw1=RA0,RA1,RA2,RA3,RA4 = on pull up

'Sw3=Rc3 et Rc4 = on pull up

'Boutons sur RA0, RA1, RA2, RA3 et RA5

'Voyant sur RB7

const serial\_1 as byte[8]= (\$28,\$24,\$67,\$CE,\$04,\$00,\$00,\$86) 'Capteur de T° DS18S20

'Déclaration des variables

dim seconds, minutes, hours, \_day, \_month, year, Yh,Yb, \_Yh, \_Yb,Njs, Pm ,Pt, PtSet,P0 ,  
Mois as byte

dim Rse,Rmi,Rhe,Rjo,Rmo,Ran, Cpas, Pf,Ep as byte

dim RseTxt,RmiTxt,RheTxt,RjoTxt,RmoTxt,RanTxt,CpasTxt as string[3]

dim Ses,Mis,Hs,Js,Ms,Ahs,Als as string[3]

dim M1,M2,M3,M4,M5,M6 as string[1]

dim CntStr as string[10]

dim TimeStamp,Ds as string[45]

dim Dt as string[20]

dim Tt1 as String[23]

dim Dt12F, Tf1 as float

dim FinConv as bit

dim b, i, InA, Ct1, PtBt as byte

dim Csec,Bt as word

dim BtStr as string[6]

dim Cnt as LongInt

dim FrAcq as word

'Init I2C

dim Soft\_I2C\_Scl as sbit at RC3\_bit

Soft\_I2C\_Sda as sbit at RC4\_bit

```
Soft_I2C_Scl_Direction as sbit at TRISC3_bit
Soft_I2C_Sda_Direction as sbit at TRISC4_bit
```

```
'Init LCD
```

```
dim LCD_RS as sbit at RB4_bit
LCD_EN as sbit at RB5_bit
LCD_D4 as sbit at RB0_bit
LCD_D5 as sbit at RB1_bit
LCD_D6 as sbit at RB2_bit
LCD_D7 as sbit at RB3_bit
LCD_RS_Direction as sbit at TRISB4_bit
LCD_EN_Direction as sbit at TRISB5_bit
LCD_D4_Direction as sbit at TRISB0_bit
LCD_D5_Direction as sbit at TRISB1_bit
LCD_D6_Direction as sbit at TRISB2_bit
LCD_D7_Direction as sbit at TRISB3_bit
```

```
'Interruption toute les secondes avec comptage secondes
```

```
Sub procedure INTERRUPT()
```

```
if PIR1.TMR1IF = 1 then
  Ct1 = Ct1 + 1
  if Ct1 >= 4 then
    Csec = Csec + 1
    Ct1 = 0
    if Csec >= Bt then          'Bt = consigne temps
      Pt = 1                    'Ordre d'enregistrement
      Csec = 0
    end if
    TMR1H = 47
    TMR1L = 112
  end if
  PIR1.TMR1IF = 0
end if
end sub
```

```
'Lecture sonde de température
```

```
sub function ReadTemp12() as float          '12 bits DS18B20
```

```
  Dim temp12 as byte[2]
  Dim Td12 as longint
  Ow_Write(PORTA, 5, $BE)          'Lecture du scratchpad 2 octets
  temp12[0] = Ow_Read(PORTA, 5)    'Read octet bas Temp L
  temp12[1] = Ow_Read(PORTA, 5)    'Read octet haut Temp H
  Td12 = temp12[1] * 256 + temp12[0] 'Valeur température brute
  if (Td12 >>15) = 1 then          'Si bit poids fort = 1 alors négatif
    Td12=(Td12 xor $FFFF)          'Négatif
    Td12=Td12+1
    Dt12F=0-float(Td12)*0.0655     'Facteur pour 12 bits
  else
    Dt12F=float(Td12)*0.0655      'Positif
  end if
  Result = Dt12F
end sub
```

```
'Conversion température en string
```

```
sub function ConvFloatStr(Dim ValF as float) as string[23]
```

```
  Dim F1St as string[23]
  floattostr(ValF,F1St)            'Conversion temp. float en chaine
  if ValF > -10 then                'Mise en place des espaces avant
    if ValF < 0 then strappendpre(" ",F1St)
```

```

else
if ValF >= 0 then
if ValF < 10 then
strappendpre(" ",FISt)
strappendpre(" ",FISt)
else
if ValF > 10 then strappendpre(" ",FISt)
end if
end if
end if
end if
end if
Result = FISt
end sub
'Lecture Seconde, Minute, Heure, Jour, Mois sur RTC PCF8583
sub procedure Read_Time()
Soft_I2C_Start()           'Start
Soft_I2C_Write(0xA0)       'Adresse PCF8583, voir PCF8583 datasheet
Soft_I2C_Write(2)         'Lecture à l'adresse 2
Soft_I2C_Start()           'Start
Soft_I2C_Write(0xA1)       'Address PCF8583 pour R/W = 1
seconds = Soft_I2C_Read(1)  'lecture 8 bit secondes
minutes = Soft_I2C_Read(1) 'lecture 8 bit minutes
hours = (Soft_I2C_Read(1)) and $3F 'lecture 6 bit heures
_day = (Soft_I2C_Read(1)) and $3F 'lecture 6 bit jour
_month = (Soft_I2C_Read(0)) and $1F 'Lecture 5 bit mois
Soft_I2C_Stop()           'Stop
end sub
'Lecture Année sur RTC
sub procedure Read_Year()
Soft_I2C_Start()           'Start
Soft_I2C_Write(0xA0)       'Adresse carte RTC PCF8583
Soft_I2C_Write(16)        'Lecture à l' adresse 16
Soft_I2C_Start()           'Start
Soft_I2C_Write(0xA1)       'Adresses PCF8583 pour R/W = 1
Yh = Soft_I2C_Read(1)      'Lecture Année MC
Yb = Soft_I2C_Read(0)     'Lecture Année DU
Soft_I2C_Stop()           'Stop
end sub
'Formatage date et heure
sub procedure Transform_Time()
seconds = ((seconds and 0xF0) >> 4)*10 + (seconds and 0x0F) 'Adapte seconde
minutes = ((minutes and 0xF0) >> 4)*10 + (minutes and 0x0F) 'Adapte mois
hours = ((hours and 0xF0) >> 4)*10 + (hours and 0x0F) 'Adapte heure
year = (_day and 0xC0) >> 6 ' Extract. An.bisextille
Njs = (_month and 0x70)>>5
_day = ((_day and 0x30) >> 4)*10 + (_day and 0x0F) 'Adapte jour
_month = ((_month and 0x10) >> 4)*10 + (_month and 0x0F) 'Adapte mois
_Yh = ((Yh and 0xF0)>> 4)*10 + (Yh and 0x0F) 'Adapte année MC
_Yb = ((Yb and 0xF0)>> 4)*10 + (Yb and 0x0F) 'Adapte année DU
'Formatage du champs date heure en vue de l'archivage
Bytetostr(_Day,Js)
ltrim(Js)
if length(Js)<2 then strappendpre("0",Js) end if
strappendsuf(Js,47)

```

```

Bytetostr(_month,Ms)
ltrim(Ms)
if length(Ms)<2 then strappendpre("0",Ms) end if
strappendsuf(Ms,47)
Bytetostr(_Yh,Ahs)
ltrim(Ahs)
if length(Ahs)<2 then strappendpre("0",Ahs) end if
Bytetostr(_Yb,Als)
ltrim(Als)
if length(Als)<2 then strappendpre("0",Als) end if
strappendsuf(Als,32)
Bytetostr(hours,Hs)
ltrim(Hs)
if length(Hs)<2 then strappendpre("0",Hs) end if
strappendsuf(Hs,58)
Bytetostr(minutes,Mis)
ltrim(Mis)
if length(Mis)<2 then strappendpre("0",Mis) end if
strappendsuf(Mis,58)
Bytetostr(seconds,Ses)
ltrim(Ses)
if length(Ses)<2 then strappendpre("0",Ses) end if
TimeStamp = ""
strcat(TimeStamp,Js)
strcat(TimeStamp,Ms)
strcat(TimeStamp,Ahs)
strcat(TimeStamp,Als)
strcat(TimeStamp,Hs)
strcat(TimeStamp,Mis)
strcat(TimeStamp,Ses)
delay_ms(30)
end sub
'Affichage heure sur LCD
sub procedure Display_Time()
Lcd_Out(1,3," : ")
Lcd_Chr(1, 1, (hours / 10) + 48)           'Affiche D heure
Lcd_Chr(1, 2, (hours mod 10) + 48)        'Affiche U heure
Lcd_Chr(1,4, (minutes / 10) + 48)         'Affiche D minute
Lcd_Chr(1,5, (minutes mod 10) + 48)       'Affiche U minute
Lcd_Chr(1,7, (seconds / 10) + 48)         'Affiche D seconde
Lcd_Chr(1,8, (seconds mod 10) + 48)       'Affiche U seconde
if Pf = 1 then
delay_ms(50)
Lcd_Out(2,9," ")
Lcd_Out(2,3,"/ / ")
Lcd_Chr(2, 1, (_day / 10) + 48)           'Affiche D jour
Lcd_Chr(2, 2, (_day mod 10) + 48)        'Affiche U jour
Lcd_Chr(2,4, (_month / 10) + 48)         'Affiche D mois
Lcd_Chr(2,5, (_month mod 10) + 48)       'Affiche U Mois
Lcd_Chr(2,7, (_Yb / 10) + 48)           'Affiche D année
Lcd_Chr(2,8, (_Yb mod 10) + 48)         'Affiche U année
end if
end sub
'Init carte SD
Sub procedure InitCarte()

```

```

if P0 = 0 then
  Lcd_Cmd(_LCD_CLEAR)
  Lcd_Out(1,1,"Creation nouveau")
  Lcd_Out(2,1," Fichier ")
  Ds= "Del f1.txt "
  strAppendSuf(Ds,13)
  strAppendSuf(Ds,10)
  UART1_Write_Text(Ds)
  delay_ms(200)
  'Création fichier
  Dt="fcreate f1.txt"
  strAppendSuf(Dt,13)
  strAppendSuf(Dt,10)
  UART1_Write_Text(Dt)
  delay_ms(1000)
  Lcd_Cmd(_LCD_CLEAR)
  P0 = 1
end if

```

*'Nom du fichier f1.txt*

end sub

*'Ecriture sur carte SD*

**Sub procedure EcritureCarte()**

```

Ds ="fputs f1.txt /a "
Strcat(Ds, CntStr)
strAppendSuf(Ds,44)
Strcat(Ds, TimeStamp)
strAppendSuf(Ds,44)
Strcat(Ds,Tt1)
strAppendSuf(Ds,13)
strAppendSuf(Ds,10)
UART1_Write_Text(Ds)
delay_ms(150)
'Fermeture fichier
Ds="fclose"
strAppendSuf(Ds,13)
strAppendSuf(Ds,10)
UART1_Write_Text(Ds)
delay_ms(150)

```

end sub

*'Réglage RTC*

**Sub procedure SetRtc()**

```

Read_Time()
Read_Year()
Rjo = bcd2dec(_Day)
Rmo = bcd2dec(_Month )
Rhe = bcd2dec(hours)
Rmi = bcd2dec(minutes)
Rse = bcd2dec(seconds)
Ran = bcd2dec(Yb)
bytetostr(0,RseTxt)
bytetostr(Rmi,RmiTxt)
bytetostr(Rhe,RheTxt)
bytetostr(Rjo,RjoTxt)
bytetostr(Rmo,RmoTxt)
bytetostr(Ran,RanTxt)
Cpas = 1

```

*'Lecture ss, mm, hh, jj, mm*

*'Lecture aa DU*

*'Conversion BCD --> DEC*

*'Conversion en chaine*

*'Compteur de Pas = 1*



```

while PtSet = 1
if PtSet = 1 then
if ((PortA xor 255) and 9) = 9 then Inc(Cpas) end if 'Incrément Pas
If Cpas > 5 then Cpas = 1 end if 'Raz Pas
'Set jour


---


if Cpas=1 then
M1 = "*"
if (Rjo< 31) then
if ((PortA XOR 255) and $0A) = $0A then Inc(Rjo) end if 'Incrémente
end if
if (Rjo> 1) then
if ((PortA XOR 255) and $0C) = $0C then Dec(Rjo) end if 'Décrémente
end if
else M1 = "J"
end if
'Set mois


---


if Cpas=2 then
M2 = "*"
if (Rmo< 12) then
if ((PortA XOR 255) and $0A) = $0A then Inc(Rmo) end if
end if
if (Rmo> 1) then
if ((PortA XOR 255) and $0C) = $0C then Dec(Rmo) end if
end if
else M2 = "M"
end if
'Set An


---


if Cpas=3 then
M3 = "*"
if (Ran< 99) then
if ((PortA XOR 255) and $0A) = $0A then Inc(Ran) end if
end if
if (Ran> 1) then
if ((PortA XOR 255) and $0C) = $0C then Dec(Ran) end if
end if
else M3 ="A"
end if
'Set heure


---


if Cpas=4 then
M4 = "*"
if (Rhe< 23) then
if ((PortA XOR 255) and $0A) = $0A then Inc(Rhe) end if
end if
if (Rhe> 0) then
if ((PortA XOR 255) and $0C) = $0C then Dec(Rhe) end if
end if
else M4 = "H"
end if
'Set minute


---


if Cpas=5 then
M5 = "*"
if (Rmi< 59) then
if ((PortA XOR 255) and $0A) = $0A then Inc(Rmi) end if
end if
if (Rmi> 1) then

```

```

if ((PortA XOR 255) and $0C) = $0C then Dec(Rmi) end if
end if
else M5 ="M"
end if
Lcd_Cmd(_LCD_CLEAR)
Lcd_Out(1,1,M1)
  bytetostr(Rjo,RjoTxt)
  Lcd_out(1,2,RjoTxt)
Lcd_Out(1,6,M2)
  bytetostr(Rmo,RmoTxt)
  Lcd_out(1,7,RmoTxt)
Lcd_Out(1,11,M3)
  bytetostr(Ran,RanTxt)
  Lcd_out(1,12,RanTxt)
Lcd_Out(2,1,M4)
  bytetostr(Rhe,RheTxt)
  Lcd_out(2,2,RheTxt)
Lcd_Out(2,6,M5)
  bytetostr(Rmi,RmiTxt)
  Lcd_out(2,7,RmiTxt)
  Lcd_out(2,11,"S 00 ")
if ((PortA XOR 255) and $19) = $19 then
Soft_I2C_Init()
Soft_I2C_Start()
Soft_I2C_Write(0xA0)
Soft_I2C_Write(0)
Soft_I2C_Write(0x80)
Soft_I2C_Write(0)
Soft_I2C_Write(Dec2Bcd(0))
Soft_I2C_Write(Dec2Bcd(Rmi))
Soft_I2C_Write(Dec2Bcd(Rhe))
Soft_I2C_Write(Dec2Bcd(Rjo))
Soft_I2C_Write(Dec2Bcd(Rmo))
Soft_I2C_Stop()
  delay_ms(100)
Soft_I2C_Start
Soft_I2C_Write($A0)
Soft_I2C_Write($10)
Soft_I2C_Write(Dec2Bcd(20))
Soft_I2C_Write(Dec2Bcd(Ran))
Soft_I2C_Stop
  delay_ms(50)
Soft_I2C_Start()
Soft_I2C_Write(0xA0)
Soft_I2C_Write(0)
Soft_I2C_Write(0)
Soft_I2C_Stop()
  delay_ms(50)
Lcd_Cmd(_LCD_CLEAR)
Lcd_Out(1,1,"Set RTC termine")
  delay_ms(1500)
Lcd_Cmd(_LCD_CLEAR)
PtSet = 0
end if
end if

```

*'Seconde = 0*  
*'Validation par RA0 \* RA3 \* RA4*  
*'Init I2C*  
*'Start*  
*'Adresse carte RTC PCF8583*  
*'Start from address 0*  
*'Ecriture \$80 à l'adresse 0*  
*'Ecriture 0 à l'adr 1 contenant 1/100ème de secondes*  
*'Ecriture Ss à l'adresse 2 contenant les secondes*  
*'Ecriture Sn à l'adresse 3 contenant les minutes*  
*'Ecriture Sh à l'adresse 4 contenant les heures*  
*'Ecriture Sj à l'adresse 5 contenant les Jour*  
*'Ecriture Sm à l'adresse 6 contenant les mois*  
*'Stop*  
*'Start*  
*'adresse carte RTC PCF8530*  
*'start à l'adresse 16*  
*'Ecriture M/C (2000)*  
*'Ecriture D/U*  
*'Stop*  
*'Start*  
*'Adresse carte RTC PCF8530*  
*'Start à l'adresse 0*  
*'Ecritue 0 enable comptage*  
*'Stop*  
*'Clear Lcd display*

```

if ((PortA XOR 255) and $18) = $18 then PtSet = 0 end if
  delay_ms(600)
wend
  Lcd_Cmd(_LCD_CLEAR)           'Clear Lcd display'
  Lcd_Out(1,1," Quitte RTC set")
  Delay_ms(2000)
end sub
'Init fréquence d'acquisition'
sub procedure SetBaseTemps()
  select case FrAcq
  case 0
    Bt = 5                       'Consigne en secondes'
    BtStr = " 5s"                'Consigne en chaine'
  case 1
    Bt = 15
    BtStr = " 15s"
  case 2
    Bt = 30
    BtStr = " 30s"
  case 3
    Bt = 60
    BtStr = " 1mn"
  case 4
    Bt = 600
    BtStr = " 10mn"
  case 5
    Bt = 900
    BtStr = " 15mn"
  case 6
    Bt = 1800
    BtStr = " 30mn"
  case 7
    Bt = 3600
    BtStr = " 1h"
  end select
end sub
'Init principale'
sub procedure Init_Main()
  PORTA = $0
  PORTB = $0
  PORTC = $0
  TRISA = $1F
  TRISB = $0
  TRISC = $80
  ADCON0 = $0
  ADCON1 = $0F
  T1CON = $F1
  PIE1.TMR1IE = 1
  INTCON.PEIE = 1
  INTCON.GIE = 1
  UART1_Init(9600)             'Init Uart1'
  Delay_ms(100)
  Soft_I2C_Init()              'Initialise Soft I2C'
  Lcd_Init()                   'Init LCD'
  Lcd_Cmd(_LCD_CLEAR)         'Clear LCD'

```

```

Lcd_Cmd(_LCD_CURSOR_OFF)      'Efface curseur LCD
end sub
'Programme principal
main:
Init_Main()
'Init carte SD
Dt ="init"
strAppendSuf(Dt,13)
strAppendSuf(Dt,10)
UART1_Write_Text(Dt)
delay_ms(200)
Cnt = 0
Pt = 1
P0 = 0
Pf = 0
PtSet = 0
Ptbt = 0
FrAcq = 0
FrAcq = EEPROM_Read($0000)      'Lecture à l'adresse $0 de l'Eeprom
delay_ms(100)
SetBaseTemps()                  'Mise à jour Base de temps
Lcd_Cmd(_LCD_CLEAR)
Lcd_out(1,1, " ENREGISTREUR")
Lcd_out(2,1," DE TEMPERATURE")
Delay_ms(2000)
if ((PortA XOR 255) and $10) = $10 then PtBt = 1 end if
while PtBt = 1
InA = (PortA xor $0F) and $0F      'Lecture 4 bits faible sur portA
FrAcq = word((PortA xor $0F) and $0F )
SetBaseTemps()                  'Mise à jour Base de temps
Lcd_Cmd(_LCD_CLEAR)
Lcd_Out(1,1," Base de temps ")
Lcd_Out(2,4,BtStr)
delay_ms(1000)
if ((PortA XOR 255) and $10) = $10 then
EEPROM_Write($0000, FrAcq)      'Ecriture valeur base de temps eeprom
delay_ms(100)
PtBt = 0
Lcd_Out(1,1," Quitte ")
Lcd_Out(2,1," Base de temps ")
delay_ms(2000)
end if
wend
if ((PortA xor 255) and $08) = $08 then      'Reglage RTC
PtSet = 1
SetRtc()
end if
Lcd_Cmd(_LCD_CLEAR)
while ((PortA xor 255) and $10) <> $10      'Attente bouton RA4
Lcd_out(1,1," Depart ")
Lcd_out(2,1," Bouton bleu ")
delay_ms(100)
wend
Pm = 0
while TRUE                              'Boucle sans fin

```

```

if Pm = 0 then
if (PortA xor $10) and $10 = $10 then
InitCarte()
Pm = 1
Ct1 = 0
Csec = 0
Cnt= 0
Pt = 1
end if
Lcd_Out(2,11,BtStr)
end if
if Pm = 1 then
Lcd_Chr(1,3,":")
Lcd_Chr(1,6,":")
Read_Time()
Read_Year()
Transform_Time()
Display_Time()
delay_ms(700)
'Conversion pour tous les capteurs de température
Ow_Reset(porta, 5)
Ow_Write(porta, 5, $CC)
Ow_Write(porta, 5, $44)
while Finconv <> 1
FinConv = Ow_Read(PortA, 5)
wend
'Capteur 1
Ow_Reset(porta, 5)
Ow_Write(porta, 5, $55)
for i=0 to 7
b = serial_1[i]
Ow_Write(porta, 5, b)
Delay_ms(10)
next i
Tf1 = ReadTemp12
Tt1 = ConvFloatStr(Tf1)
Lcd_out(1,10,Tt1)
Lcd_Chr(1,15,$DF)
Lcd_Out(1,16,"C")
if Pt = 1 then
Pt=0
Cnt=Cnt+1
LongWordtostr(Cnt, CntStr)
ltrim(CntStr)
if Cnt <= 540000 then EcritureCarte() end if
delay_ms(100)
end if
end if
if (PortA xor $10) and $10 = $10 then PF = 1 else Pf = 0 end if
if Pf = 0 then
Lcd_Out(2,1,"")
Lcd_Out(2,1,CntStr)
end if
if (Csec>(Bt-3)) and (Csec<(Bt+1)) then
setbit(PORTB,7)

```

'Préparation carte SD  
'Delete file si Ra4=1

'Affiche base de temps

'Lecture RTC(PCF8583)  
'Lecture année sur RTC  
'Formate date et temps  
'Affiche date et temps sur RTC

'Reset OW  
'Envoi CMD à tous les capteurs  
'Comde conversion à tous les capteurs  
'Attente fin conversion

'Reset OW  
'Commande sélection capteur  
'Sélection du capteur  
'envoi des 8 octets d'adresse au capteur  
'Délai entre envois  
'Température en float  
'Température en string  
'Affiche température  
'Affiche du signe degré  
'Affiche C pour centigrade  
'Ordre d'enregistrement interruption

'Compteur d'enregistrements  
'Conversion compteur en chaine  
'Supprime les espaces à gauche  
'Enregistrement

'Affich. date  
'Condition affichage date  
'Affiche nombre d' enregistrement

'Condition voyant sur RB7  
'Voyant enregistrement imminent = 1

```

else
clearbit(PORTB,7)
end if
delay_ms(50)
wend
end.

```

'Voyant enregistrement imminent = 0

## 7 Traitement des données :

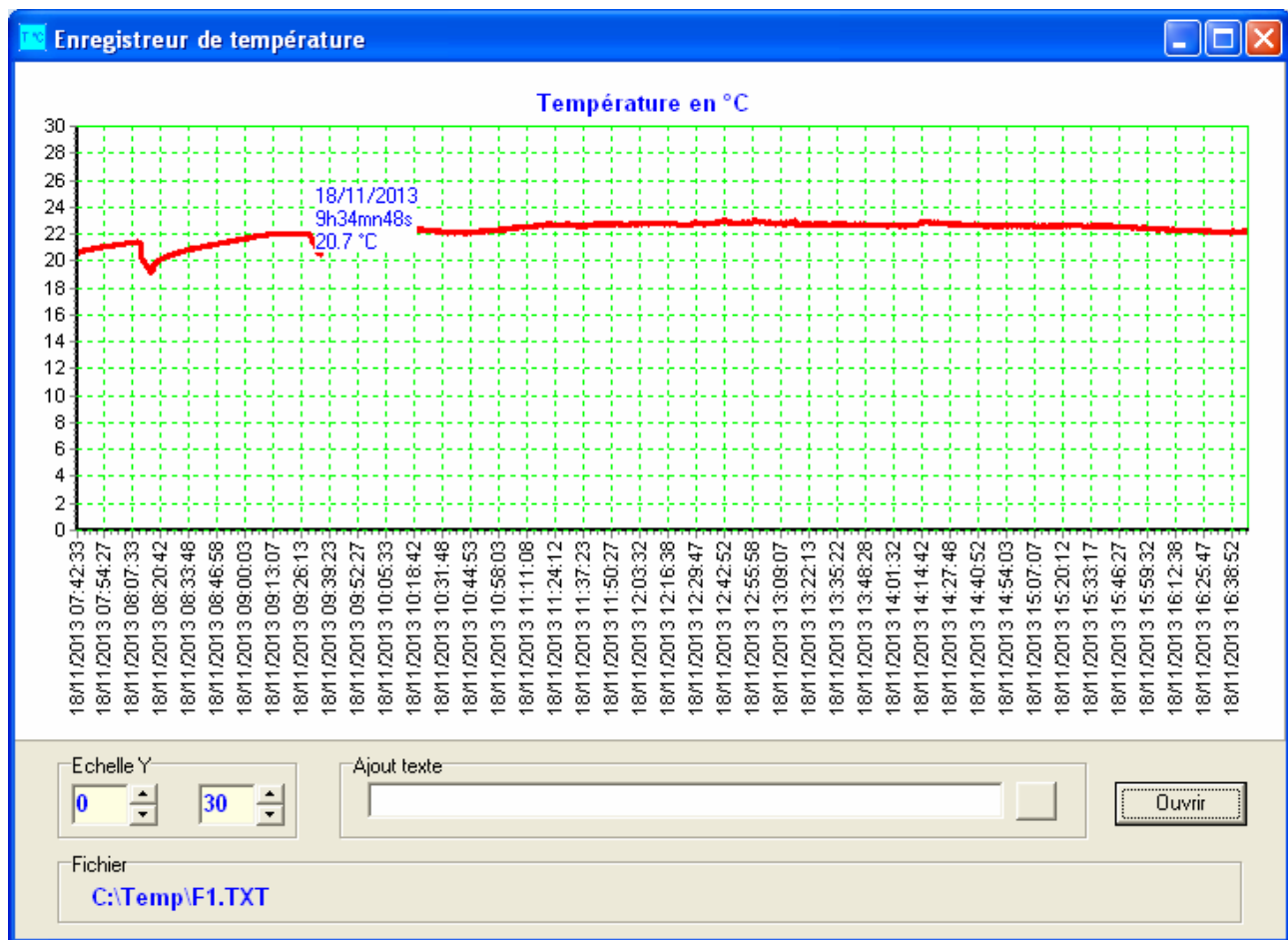
Une application développée sous Delphi permet visualiser l'évolution de la température sur un graphique.

Pour agrandir une partie du graphe, sélectionner la zone avec le bouton gauche et en tirant la souris vers la gauche.

Annuler l'agrandissement avec le bouton gauche et en tirant la souris vers la gauche.

Déplacer le graphe avec le bouton droit et en déplaçant la souris.

Afficher les données en pointant la courbe avec le pointeur de la souris.



## 8/ Liste des composants :

Qu	Designation	Réf.	Fourn.	PU	Prix TTC
1	Pot 10k	T7YA10K	Gotronic	0.75	0.75
1	PIC 18F2420		Gotronic	8.00	8.00
1	Bornier 3	49931	Gotronic	1.20	1.20
1	Res 4.7k	04044	Gotronic	0.06	0.06
8	Res 10k	04148	Gotronic	0.06	0.48
1	Res 1k	04036	Gotronic	0.06	0.06
1	Led 5 mm	03025	Gotronic	0.08	0.08
1	Res 10	04012	Gotronic	0.06	0.06
1	Afficheur LCD 2x16	03318	Gotronic	10.95	10.95
1	Radiateur	10000	Gotronic	0.40	0.40
1	Reg 5V	02450	Gotronic	0.50	0.50
1	Quartz 8Mhz	05258	Gotronic	1.00	1.00
2	Condo 22 pF céram.	15816	Gotronic	0.10	0.20
1	Condo 100nF polyest.	04817	Gotronic	0.15	0.15
1	Condo 100 nF céram	04764	Gotronic	0.25	0.25
1	Embbase alim.CI215	48148	Gotronic	0.65	0.65
1	Fiche alim.FA215	48133	Gotronic	0.30	0.30
1	Connect 2x5 M	08011	Gotronic	0.15	0.15
2	Connecteur 10 points F	48161	Gotronic	0.80	1.60
1	Support CI 28 b.	08158	Gotronic	0.28	0.28
4	Bouton poussoir noir	07100	Gotronic	0.75	3.00
1	Bouton poussoir bleu	07102	Gotronic	0.90	0.90
2	Interrupteur inverseur	07005	Gotronic	1.50	3.00
1	Sonde DS18B20		Ebay	1.45	1.45
1	Coffret G313	11163	Gotronic	10.80	10.80
1	Presse étoupe	11325	Gotronic	2.40	2.40
1	Carte SD	SD-Com	Lextronic	32.00	32.00
1	RTC PCF8583	PIC3RTC	Lextronic	21.99	21.99
					<b>102.66</b>