

```

*****
'MUSEE DES MINES RUMELANGE
'PROJET T3 EC+EE 2010/2011
'BETREUER: JEAN-CLAUDE FELTES
*****
'MESS-STATION (CO2, Temperatur IN/IN_Near/OUT + Regelung, Tristate R/W, LCD)
*****

'uC-INIT-----

$regfile = "m8def.dat"
$crystal = 8000000
$baud = 9600
$hwstack = 100
$swstack = 100
$framesize = 100

'DISPLAY-----

'Config Lcd = 20 * 2
'Config Lcdpin = Pin , Db4 = Portb.2 , Db5 = Portb.3 , Db6 = Portb.4 , Db7 = Portb.5 , Rs =
Portb.0 , E = Portb.1
'Cursor Noblink
'Cursor Off

'CO2-----

Config Scl = Portc.4           'Pin0 am PortD als Taktleitung deklarieren
Config Sda = Portc.5         'Pin1 am PortD als Dateneingang deklarieren
                               '!!! Niemals die Pins am selben

LCD-Pins benutzen sonst wird die Werte verfälscht !!!
Config Pinc.3 = Input        'Definiert Pin4 am PortD als Eingang
Set Portc.3                  'Pull-Up Widerstand einschalten

Dim Operation_status As Byte 'Variable für den Operation Status Byte
Dim Highbyte As Integer     'Variable für den Highbyte
Dim Lowbyte As Byte         'Variable für den Lowbyte
Dim Check_sum As Byte       'Variable für den Check Sum Byte
Dim Check_sumcompare As Byte 'Variable für den Vergleichwert Check SumCompare
Dim Co2 As Word             'Variable für den Co2 Wert
Dim I As Integer            'Variable für den Zähler Korrektur I
Dim Multiplier As Integer   'Variable für den Multiplier

'TEMPERATURLESUNG-----

Config lwire = Portd.5
Config lwire = Portd.6
Config lwire = Portd.7

Dim Sci(9) As Byte
Dim Ti As Integer
Dim Temp As Single
Dim Thpinr As Byte
Dim Temperatur_out As Single
Dim Temperatur_inmc As Single
Dim Temperatur_inheiz As Single
Dim Temperatur_out_rund As String * 5
Dim Temperatur_inmc_rund As String * 5
Dim Temperatur_inheiz_rund As String * 5           'Pull-Up Widerstand einschalten

'TEMPERATURREGLUNG-----

Config Pinc.0 = Output
Dim Onflag As Bit

'VARIABLEN DECLARIEREN-----

Dim Sendeflag As Bit
Dim Commandflag As Bit
Dim Ok As Byte
Dim Slavesseconds As Long
Slavesseconds = 1

'TRISTATE-----

Config Portd.2 = Output
Portd.2 = 1           '1 = read

'Timer 1s-----

Config Pind.4 = Output           'LED 1s
Config Timer1 = Timer , Prescale = 1024 , Clear Timer = 1 , Compare A = Disconnect
Ocr1ah = High(7812.5)

```

```

Ocr1al = Low(7812.5)
Tccr1a = 0
On Compare1a Tim1_1s
Enable Compare1a
Enable Interrupts

'SERIALINTERRUPT-----

On Urxc Serialinterrupt
Enable Urxc

'HAUPTPROGRAMM-----

Cls
Commandflag = 1

Do

  If Commandflag = 1 Then
    Gosub Co2_messen
    Gosub Temperaturlesung
    Gosub Lcd_anzeige
    Commandflag = 0
  End If

  If Sendeflag = 1 Then
    Waitms 10
    'write
    Portd.2 = 0
    Waitms 10

    Gosub Terminal

    Sendeflag = 0
    Waitms 10
    'read
    Portd.2 = 1
  End If

Loop

End

'UNTERPROGRAMME-----
Tim1_1s:

  Toggle Portd.4
  Commandflag = 1

  Incr Slavesecnds

  If Temperatur_inheiz < 20 Then
    Set Portc.0
    Onflag = 1
  Else
    Reset Portc.0
    Onflag = 0
  End If

Return

'-----
Serialinterrupt:

  Ok = Udr
  Select Case Ok
    Case "*"
      Sendeflag = 1
  End Select

Return

'-----
Co2_messen:

  Gosub Choose_k30k33_sr           'Wähle Modus für K30 oder K33
  Gosub Check_ops_chs_sr          'Ermittle Operation Byte und Check Sum Byte
  Gosub Correction_sr            'Korrekturverfahren mit Operation Byte und Check Sum
  Byte
  Gosub Co2_wert_sr              'Berechne Co2 Wert
  Waitms 50                      'Ausgabe an der LCD Display

```

Return

 Temperaturlesung:

```
Gosub Temperatur_out_sr
Gosub Temperatur_inmc_sr
Gosub Temperatur_inheiz_sr
Gosub Runden_sr
```

Return

 'Unterprogramme CO2-----

Choose_k30k33_sr:

```
If Pinc.3 = 0 Then          'Falls Pin4 am PortD am Masse liegt (Jumper)
  Multiplicator = 10
Else
  Multiplicator = 1
End If
```

Return

 Check_ops_chs_sr:

```
I2cstart          'Start Bedingung
I2cwbyte &B11010000 'schicke Adressbit
I2cwbyte &B00100010 '1. Nibbel :schicke Befehl 2: lese aus Ram 2.
Nibbel : Inhaltspeicherplatz betragt 2 Byte
I2cwbyte &B00000000
I2cwbyte &B00001000 'Lese im Adress im Ram :0x08
I2cwbyte &B00101010 'Checksumme
I2cstop          'Stop Bedingung

Waitms 50        'Warte 50ms
I2cstart          'Start Bedingung
I2cwbyte &B11010001 'spreche Slave an und schicke Adressbit mit Lese
Befehl
I2crbyte Operation_status , Ack 'lese aus variable Operation_status
I2crbyte Highbyte , Ack 'lese Highbyte
I2crbyte Lowbyte , Ack 'lese Lowbyte Der CO2-Wert besteht aus den Highbyte &
Lowbyte
I2crbyte Check_sum , Nack 'lese Check_sum
I2cstop          'Stop Bedingung

Check_sumcompare = Operation_status + Highbyte 'Check_sumcompare besteht aus den
Addition von 1. - 3. Byte
Check_sumcompare = Check_sumcompare + Lowbyte 'Check_sumcompare wird berechnen für
eine späteren Vergleich mit Check_sum
```

Return

 Correction_sr:

```
If Operation_status = 1 Or Check_sum <> Check_sumcompare Then 'Bedingung: Falls
Operation Status Byte = 1 ODER Check Sum Byte ist
  For I = 1 To 10 'nicht gleich wie der berechnete Check Sumcompare
dann Schleife bis 10
  Gosub Check_ops_chs_sr
  If Operation_status = 0 And Check_sum = Check_sumcompare Then Exit For 'Ausnahme für
die Schleife: Operation Status Byte = 0 UND
  Next I 'Check Sum Byte = Check Sumcompare
Else
  I = 0 'Zähler wird zurück gesetzt
End If
```

Return

 Co2_wert_sr:

```
Highbyte = Highbyte * 256 'Highbyte berechnen
Co2 = Highbyte + Lowbyte 'Co2-Werte berechnen
Co2 = Co2 * Multiplicator
```

Return

 'Unterprogramme Temperaturlesung-----

```

-----
Temperatur_out_sr:

    Thpinnr = 6
    Gosub Temperaturmessung_sr
    Temperatur_out = Temp

Return

-----
Temperatur_inmc_sr:

    Thpinnr = 7
    Gosub Temperaturmessung_sr
    Temperatur_inmc = Temp

Return

-----
Temperatur_inheiz_sr:

    Thpinnr = 5
    Gosub Temperaturmessung_sr
    Temperatur_inheiz = Temp

Return

-----
Temperaturmessung_sr:

    'Mißt Temperatur
    'Variablen benutzt: "Thpinnr" Bit des Portes des Sensors, muss vorher gesetzt werden
                        "Scl" Ausgelesener Wert
                        "Ti" Integer der Temperatur
    'Ausgabe in Variable "Temp" Temperatur in dezimaler Zahl

    'Messen
    lwwrite Pind , Thpinnr          'Reset lWire
    lwwrite &HCC , 1 , Pind , Thpinnr 'skip rom alle Temperatursensoren
    lwwrite &H44 , 1 , Pind , Thpinnr 'Convert T alle Temperatursensoren
    Waitus 200                     'Warte 200 us

    'Auslesen
    lwwrite Pind , Thpinnr          'reset lWire
    lwwrite &HCC , 1 , Pind , Thpinnr 'skip rom Temperatursensor innen
    lwwrite &HBE , 1 , Pind , Thpinnr 'read scratchpad Temperatursensor innen

    'Auswertung
    Sci(1) = lwwrite(9 , Pind , Thpinnr)
    Ti = Makeint(Sci(1) , Sci(2))    'interger aus sci(1) und sci(2)
    Temp = Ti                       'aus dem interger wird ein single gemacht
    Temp = Temp / 16                'der wert wird durch 16 geteilt um den richtigen
    dezimal wert anzuzeigen

Return

-----
Runden_sr:

    Temperatur_out_rund = Fusing(temperatur_out , "#.##")
    Temperatur_inmc_rund = Fusing(temperatur_inmc , "#.##")
    Temperatur_inheiz_rund = Fusing(temperatur_inheiz , "#.##")

Return

-----
Lcd_anzeige:

Config Lcd = 20 * 2
Config Lcdpin = Pin , Db4 = Portb.2 , Db5 = Portb.3 , Db6 = Portb.4 , Db7 = Portb.5 , Rs =
Portb.0 , E = Portb.1
Cursor Noblink
Cursor Off
Cls

Locate 1 , 1 : Lcd "
Locate 1 , 1 : Lcd "O:" ; Temperatur_out_rund ; "C"
Locate 1 , 11 : Lcd "I:" ; Temperatur_inmc_rund ; "C"
Locate 2 , 1 : Lcd "
Locate 2 , 1 : Lcd "H:" ; Temperatur_inheiz_rund ; "C"
Locate 2 , 11 : Lcd Co2 ; "ppm"

```

Return

```
-----  
-----
```

Terminal:

```
Print Slavesecods  
Print Temperatur_out_rund  
Print Temperatur_inmc_rund  
Print Temperatur_inheiz_rund  
Print Onflag  
Print "none"  
Print Co2
```

Return