

# Firmware Master (Betreuer)

```
'MMR Projekt Master JCF
```

```
'Mess-Intervall in Sekunden (Zeit zwischen 2 Aufzeichnungen)
```

```
Dim Minterval As Word
```

```
Dim Ee_minterval As Eram Word
```

```
$regfile = "8535def.dat"
```

```
$crystal = 4000000
```

```
$hwstack = 100
```

```
$swstack = 100
```

```
$framesize = 100
```

```
' I2C-Pins
```

```
Config Sda = Portc.1
```

```
Config Scl = Portc.0
```

```
'variables for real time clock
```

```
Config Clock = User
```

```
Config Date = Dmy , Separator = .
```

```
Dim Weekday As Byte
```

```
Config Lcd = 20 * 2
```

```
Config Lcdpin = Pin , Db7 = Portc.7 , Db6 = Portc.6 , Db5 = Portc.5 , Db4 = Portc.4 , E = Portc.3 , Rs = Portc.2
```

```
Config Watchdog = 2048
```

```
Stop Watchdog
```

```
Config Portb.0 = Output
```

```
'LED
```

```
Config Portd.2 = Input
```

```
'Taster
```

```
Portd.2 = 1
```

```
'Pullup Taster
```

```
Config Portd.6 = Output
```

```
'Tristate= R/W\
```

```
Portd.6 = 1
```

```
'erst mal tristate
```

```
'Timer1 für ls - Interrupt
```

```
Ocr1ah = High(15624)
```

```
Ocr1al = Low(15624)
```

```
Config Timer1 = Timer , Prescale = 256 , Clear Timer = 1 , Compare A = Disconnect
```

```
On Ocla Timer1_isr
```

```
Enable Ocla
```

```
Enable Interrupts
```

```
Dim Secticks As Long
```

```
Dim New_flag As Bit
```

```
'Software-UART to send and read to and from the station
```

```
Open "comd.4:9600,8,n,1" For Output As #1
```

```
Open "comd.3:9600,8,n,1" For Input As #2
```

```
'Hardware-UART zur Kommunikation mit dem PC
```

```
$baud = 9600
```

```
'Input byte
```

```
Dim Ok As Byte
```

```
'Speicherplatz für einkommende Daten
```

```
Dim Secs As Long
```

```
Dim Temperature_outside As Single
```

```
Dim Temperature_inside As Single
```

```
Dim Temperature_inside_near As Single
```

```
Dim Onflag As Byte
```

```
Dim Co2valid As Byte
```

```
Dim Co2value As Word
```

```
Dim Heatingpercent As Byte
```

```
Dim Commandflag As Bit
```

```
'temporary variables
```

```
Dim Btmp As Byte
```

```
Dim I As Byte
```

```
Dim S(10) As String * 10
```

```
Dim Stmp As String * 4
```

```

'test
Dim J As Byte
Dim S1 As String * 1

'Total time in seconds (master time)
Dim Masterseconds As Long

'-----
'   MAIN
'-----
'Messintervall (zur Sicherheit Default-wert 600s)
Minterval = Ee_minterval
If Minterval = 0 Then Minterval = 10
If Minterval = &HFFFF Then Minterval = 10

'Uhrzeit bereit haben für 1. Messung!
Gosub Readclock

Cls
Lcd "Master station ready"

Print "Master station ready"
Print "Interval = " ; Minterval
Print " ,Date, Time, Master-Seconds, Slaveticks, Temp. outside,";
Print " Temp. inside far, Temp inside near, Heating 1/0, CO2[ppm],Heating%"
Commandflag = 1

Waitms 50
Start Watchdog
'-----
Do

  'if button pushed: measure
  Debounce Pind.2 , 0 , Readvalues , Sub

  'when time is ready to measure: do it
  If Commandflag = 1 Then
    Gosub Getdata
    Gosub Printvalues
    Gosub Displaylcd
  End If

  'connection OK? If not watchdog resets uC
  Reset Watchdog

  'input from terminal?
  Ok = Inkey()
  Gosub Dispatchinput

  Waitms 10

Loop

'-----
Dispatchinput:
'dispatch input from terminal

Stop Watchdog

Select Case Ok

  'On "s" from terminal set time + date
  Case "s"
    Gosub Setclock

  'on "t" from terminal set time interval
  Case "t"
    Print "Current interval: " ; Minterval
    Input "time interval in secs:" , Minterval
    Ee_minterval = Minterval

  'On "m" : measure!
  Case "m"
    Commandflag = 1

  'on "?" help
  Case "?"

```

```

Print "s = set clock"
Print "t = set time interval"
Print "m = force measure"

End Select

Start Watchdog
Return

'-----
Getdata:
'get data from measuring station

'LED
Set Portb.0.0

'make cable driver ready to transmit
Portd.6 = 0

Waitms 5

'Command to station
Print #1 , "***";

Waitms 10

'switch cable driver to read data
Portd.6 = 1

'Read 8 pieces of data terminated by CR
For I = 1 To 8
    Input #2 , S(i)
    Reset Watchdog
Next I

'correction if LF (10) left in the buffer
' (this occurs on all but first item, as input cuts behind the CR)
For I = 1 To 8
    If Left(S(i) , 1) = Chr(10) Then S(i) = Mid(S(i) , 2)
    Reset Watchdog
Next I

'''Print : Print : Print

'Assign data to variables for processing
Secs = Val(S(1))
Temperature_outside = Val(S(2))
Temperature_inside = Val(S(3))
Temperature_inside_near = Val(S(4))
Onflag = Val(S(5))
Co2valid = Val(S(6))
Co2value = Val(S(7))
Heatingpercent = Val(S(8))

Commandflag = 0

'LED
Reset Portb.0

Return

'-----

Printvalues:

'TAB so values begin in 2. column
Print Chr(9);

'clock
Gosub Printclock

'''Reset Watchdog

'Echo values to PC

Print Secs;
Print Chr(9);
Print Temperature_outside;
Print Chr(9);
Print Temperature_inside;

```

```

Print Chr(9);
Print Temperature_inside_near;
Print Chr(9);
Print Onflag ;
Print Chr(9);
Print Co2valid ;
Print Chr(9);
Print Co2value;
Print Chr(9);
Print Heatingpercent
'''Reset Watchdog
Return
-----
Displaylcd:

Locate 2 , 1 : Lcd Spc(20)
'Locate 1 , 1 : Lcd Secs
Locate 2 , 1 : Stmp = Fusing(temperature_outside , "##.##") : Lcd Stmp
Locate 2 , 10 : Lcd Co2value
'''Reset Watchdog
Return
-----
Readvalues:

Commandflag = 1

Return
-----
Timer1_isr:
'Jede Sekunde

'Gosub Readclock (früher)
Gosub Lcdclock

Incr Secticks
If Secticks >= Minterval Then
Commandflag = 1
Secticks = 0
End If

Incr Masterseconds

'''Toggle Portb.0
Return
-----
Lcdclock:
'display time + date
Locate 1 , 1 : Lcd Spc(20) : Locate 1 , 1
'''Lcd _hour : Lcd ":" : Lcd _min : Lcd ":" : Lcd _sec
Lcd Time$
Locate 1 , 10
'''Lcd _day : Lcd "." : Lcd _month : Lcd ".20" : Lcd _year
Lcd Date$
'''Reset Watchdog
Return
-----
Printclock:

'Lsyssec = Syssec(seconds)
Print Lsyssec;
Print Chr(9);

'Print _day;
Print "." ;
Print _month;
Print "." ;
Print _year;
Print " ";
Print _hour;
Print ":";
Print _min;
Print ":";
Print _sec;
Print Date$;
Print Chr(9);
Print Time$;
Print Chr(9);
Print Syssec();
'''Print Masterseconds;
Print Chr(9);
Return
-----

```

```

Getdatetime:
'must be called so, because BASCOM looks for it when TIME$ and DATE$ is used
Readclock:
'this is the previous name
'Read real time clock DS1307

    'set register pointer to 0
I2cstart
I2cwrite &HD0
I2cwrite 0
I2cstop

'read registers
I2cstart
I2cwrite &HD1
I2cwrite _sec , Ack
I2cwrite _min , Ack
I2cwrite _hour , Ack
I2cwrite Weekday , Ack
I2cwrite _day , Ack
I2cwrite _month , Ack
I2cwrite _year , Nack
I2cstop

'convert BCD to decimal
_sec = Makedec(_sec)
_min = Makedec(_min)
_hour = Makedec(_hour)
_day = Makedec(_day)
_month = Makedec(_month)
_year = Makedec(_year)

Return
-----
Setclock:
'set new time and date and eventually turn on oscillator + SQW out
'to init clock: oscillator must be turned on (bit7=0 in register 0 )
'to turn on SQW/OUT: bit 4 = 1 in register 7 (Control register)
'both are done automatically (if Seconds <60)

'input new time
Input "Day (1-31):" , _day
Input "Month (1-12):" , _month
Input "Year (00-99):" , _year
Input "Hour (0-23):" , _hour
Input "Minutes (0-59):" , _min
Input "Seconds (0-59):" , _sec

    _sec = Makebcd(_sec)
    _min = Makebcd(_min)
    _hour = Makebcd(_hour)
    _day = Makebcd(_day)
    _month = Makebcd(_month)
    _year = Makebcd(_year)

'write to DS1307
I2cstart
I2cwrite &HD0
I2cwrite 0
I2cwrite _sec
I2cwrite _min
I2cwrite _hour
I2cwrite Weekday
I2cwrite _day
I2cwrite _month
I2cwrite _year
I2cwrite &H10
I2cstop
'Register 0 for secs
'Seconds
'Minutes
'Hours
'Weekday
'Day
'Month
'Year
'SQW enabled
Return
-----

```