DS3231 Real-Time Clock Commands

The DS3231.h include file provides both high- and low-level support for the DS3231 real-time clock chip, two alarms with a backup battery, and two IO ports. Insert the following directive in your code to make these new commands available:

```
#include < DS3231.h>
```

Here follows a list of the commands..

DS3231_EnableOscillator(flag)
enables the clock when flag is TRUE,
disables the clock when flag is FALSE

DS3231_OscillatorStopFlagStatus
A function that returns the status of the Oscillator.

DS3231_ClearOscillatorStopFlag
A method to clear the Oscillator stop flag. To be used after a power failure.

DS3231_ResetClock
resets clock completely to manufacturer’s original condition,
time to 00:00:00, day of the week to 01, date to 01/01/00,
also sets 24-hour mode and enables the clock.

DS3231_SetClock(hour, minute, second, DOW, date, month, year)
sets the entire clock: hours, minutes, seconds, day of week, date, month, year.
there is no error detection for out-of-range dates, (e.g., April 31)
also sets 24-hour mode and enables the clock.

DS3231_SetTime(hour, minute, second)
sets the time only: hours, minutes, seconds,.
also sets 24-hour mode and enables the clock.

DS3231_SetDate(dayoftheweek, date, month, year)
sets the date only: date, month, year,
there is no error detection for out-of-range dates, (e.g., April 31)

DS3231_ReadClock(hour, minute, second, flag, DOW, date, month, year)
reads the entire clock: hours, minutes, seconds, flag, day of week, date, month, year
flag = FALSE means a.m.,
flag = TRUE means p.m.

DS3231_ReadTime(hour, minute, second, flag)
reads the time only: hours, minutes, seconds, a.m. or p.m.,
flag = FALSE means a.m.,
flag = TRUE means p.m.

DS3231_ReadDate(date, month, year)
reads the date only: date, month, year

DS3231_SetHourMode(12|24)
sets the hour mode,
12 = 12-hour
24 = 24-hour
any other value defaults to 24-hour mode

DS3231_ReadHourMode(value)
returns the current hour mode,

DS3231_SetSQW(rate)
sets the square wave output pin mode:
0 = disable square wave output
1 = 1 Hz output
4 = 4096 Hz
8 = 8192 Hz
32 = 32768 Hz
any other value defaults to 1 Hz

DS3231_EnableSQW
sets the square wave output pin mode to on

DS3231_DisableSQW
sets the square wave output pin mode to off

DS3231_SetSQWInterruptControl(flag)
enables the SQW output when flag is TRUE,
disables the SQW output when flag is FALSE

DS3231_EnableSQWInterruptControl
Enables the SQW

DS3231_DisableSQWInterruptControl
Disables the SQW

DS3231_Set32kHz(flag)
enables the Set32kHz output when flag is TRUE,
disables the Set32kHz output when flag is FALSE

DS3231_SetControl (MFP_Value)
Sets Control address status to the value of the variable MFP_Value. The control - rtcc control register is at address 0xOE. Direct access to the control register permits reading and writing of the controls. Set bits usage as specified in the datasheet.

DS3231_ReadControl
This function returns the current value of the Control address. See DS3231_SetControl(for usage.

DS3231_SetControlStatus (MFP_Value)
Sets Control Status address status to the value of the variable MFP_Value. The control - rtcc

control register is at address 0xOE. Direct access to the control register permits reading and writing of the controls. Set bits usage as specified in the datasheet.

DS3231_ReadControlStatus
This function returns the current value of the Control Status address. See DS3231_SetControl(for usage.

DS3231_SetAlarm1 (Hour, Min, Sec, DOW, Date)
sets the alarm: hours, minutes, seconds, day of week, date. DOW or Date must BE 0. When DOW is non zero then the alarm if weekly, when Date is non zero then the alarm is monthly. There is no error detection for out-of-range dates, (e.g., April 31)
also sets 24-hour mode.

DS3231_SetAlarmMask1 (alarmAssertionMatch)
sets the alarm where Value can be any of the following.

DS3231_Alarm1Assertion_EverySecond	= 0x0F
DS3231_Alarm1Assertion_Seconds	= 0x0E
DS3231_Alarm1Assertion_MinutesSeconds	= 0x0C
DS3231_Alarm1Assertion_HoursMinutesSeconds	= 0x08
DS3231_Alarm1Assertion_DateHoursMinutesSeconds	= 0x00
DS3231_Alarm1Assertion_DayHoursMinutesSeconds	= 0x00

A match of these assertions will raise the alarm.

DS3231_ReadAlarm1 (Hour, Min, Sec, DOW, Date)
Returns the current settings for a specific alarm.

DS3231_ClearAlarm1
Clears a specific alarm after an alarm assertion.

DS3231_EnableAlarm1Interrupt
Enables the SQW output to be used to raise an external interrupt

DS3231_DisableAlarm1Interrupt
Disables the SQW output to be used to raise an external interrupt

DS3231_AlarmStatus1
This is a function. Returns a specific alarm status.
FALSE means the specific alarm has not met the assertion criteria
TRUE means the specific alarm has met the assertion criteria

DS3231_DisableAlarm1
Disables the alarm.

DS3231_SetAlarm2 (Hour, Min, DOW, Date)
sets the alarm: hours, minutes, day of week, date. DOW or Date must BE 0. When DOW is non zero then the alarm if weekly, when Date is non zero then the alarm is monthly.

There is no error detection for out-of-range dates, (e.g., April 31)
also sets 24-hour mode.

DS3231_SetAlarmMask1 (alarmAssertionMatch)
sets the alarm where Value can be any of the following.

DS3231_Alarm2Assertion_EveryMinute	= 0x07
DS3231_Alarm2Assertion_Minutes	= 0x06
DS3231_Alarm2Assertion_HoursMinutes	= 0x04
DS3231_Alarm2Assertion_DateHoursMinutes	= 0x00
DS3231_Alarm2Assertion_DayHoursMinutesSeconds	= 0x00

A match of these assertions will raise the alarm.

DS3231_ReadAlarm2 (Hour, Min, DOW, Date)
Returns the current settings for a specific alarm.

DS3231_ClearAlarm2
Clears a specific alarm after an alarm assertion.

DS3231_EnableAlarm2Interrupt
Enables the SQW output to be used to raise an external interrupt

DS3231_DisableAlarm2Interrupt
Disables the SQW output to be used to raise an external interrupt

DS3231_AlarmStatus2
This is a function. Returns a specific alarm status.
FALSE means the specific alarm has not met the assertion criteria
TRUE means the specific alarm has met the assertion criteria

DS3231_DisableAlarm2
Disables the alarm.

DS3231_ReadRegister (in DS_Value)
This is a function. Returns the value of the specific register as specified in DS_Value.

DS3231_WriteRegister (in DS_Value, in DS_Temp)
This method set the specific register as specified in DS_Value to the value specified in DS_Temp

DS3231_ReadRegister can be used to fetch the temperature values from the DS3231.
Unlike many other values in the DS3231 the most significant bit value returns a decimal value.
The value in the MSB can be between +127 to -127 degrees C. If the uppermost bit (bit.7) is set, the value is negative and the remaining bits hold the negative temperature. If this bit is not set the value is a positive one.

The least significant bit holds a fractional value in the two uppermost bits. All other bits in this value are zero.

Bit.7	Bit.6	Fractional Value
0	0	0.0
0	1	0.25
1	0	0.50
1	1	0.75

Example code for extracting the temperature:
This example uses an LCD display to show the temperature. For clarity the LCD initialisation is not shown here.

```
#Include <DS3231.h>

Dim TempMSB    As Byte
Let TempMSB = 0
Dim TempLSB     As Byte
Let TempLSB = 0
Dim Minus           As Bit
Let Minus        = 0

Let TempMSB = DS3231_ReadRegister(0x11)
Let TempLSB =  DS3231_ReadRegister(0x12)

   If TempMSB > 127 Then 'Minus value
      Let Minus = 1
      Let TempMSB = TempMSB - 128
   Else
      Let Minus = 0
   End If

   Select Case TempLSB
   Case 0
      Let TempLSB = 0
   Case 64
      Let TempLSB = 25
   Case 128
      Let TempLSB = 50
   Case 192
      Let TempLSB = 75
   Case Else
      Let TempLSB = 0
   End Select

   If Minus = 1 Then
      Print "-"
   End If
   Print TempMSB
   Print "."
   Print TempLSB
```

```
  If TempLSB = 0 Then
      Print "0"
  End If
```
Version 1.02