UnoLib documentation

ds1302rtc.pas

version 11/09/2025

The ds1302rtc.pas module contains implementation of TDS1302rtc class, providing the interface for the DS1302 Real-Time Clock (RTC) chip. This class simplifies the process of reading and setting the time and date on the DS1302 chip. It communicates with a microprocessor via a simple serial interface. The real-time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

Since FPC for AVR does not allow dynamic object creation, an object of type TDS1302rtc declared in the var section is ready to use and does not require the use of a constructor.

TDS1302rtc methods
procedure **Init**(SCLK_pin, DATA_pin, CE_pin: UInt8); Initializes the TDS1302rtc object and specifies the pins of the sensor used for serial communication. *Parameters* *SCLK_pin* – SCLK-pin of the DS1302 (Pin 7) *DATA_pin* - I/O-pin of the DS1302 (Pin 6) *CE_pin* - CE-pin of the DS1302 (Pin 5) Note: Originally constructor of the class.
procedure **GetTime**; Gets current data from the DS1302 storing values in fields: *seconds*, *minutes*, *hours*, *dayofweek*, *dayofmonth*, *month*, and *year*.
procedure **SetTime**(aseconds, aminutes, ahours, adayofweek, adayofmonth, amonth: UInt8; ayear: UInt16); Sets the time. *Parameters* *aseconds* - second to store in the DS1302 (0-59) *aminutes* - minute to store in the DS1302 (0-59) *ahours* - hour to store in the DS1302 (0-23) *adayofweek* - day of the week to store in the DS1302 (1-7; Monday is 1, Sunday is 7) *adayofmonth* - day of the month to store in the DS1302 (1-31) *amonth* - month to store in the DS1302 (1-12) *ayear* - *year* to store in the DS1302 (2000-2099)
procedure **Halt**(enable: UInt8);

Sets or clears the Clock Halt flag

Parameters
enable – enables (1) or disables (0) the flag.

```
procedure WriteProtect(enable: UInt8);
```

Sets or clears the Write Protect bit.

Parameters
enable – enables (1) or disables (0) the bit.

```
procedure setTCR(value: UInt8);
```

Sets the Trickle-Charge Register. Use the defined literals (TCR_) as *value* parameter to set the number of diodes and resistance used.

```
procedure WriteBuffer(r: TDS1302RAM);
```

Burst-writes the buffer to on-chip RAM.

Parameters
r – buffer containing 31 bytes (as TDS1302RAM type)

```
function ReadBuffer: TDS1302RAM;
```

Reads on-chip RAM returning data in TDS1302RAM type (31 bytes).

```
procedure Poke(addr, value: Uint8);
```

Writes a single byte to on-chip RAM.

Parameters
addr - address of byte to write (0-30)
value – value to write

```
function Peek(addr: Uint8): UInt8;
```

Reads one byte from on-chip RAM on specified address.

Parameters
addr - address of the byte (0-30)

TDS1302rtc fields

```
seconds: UInt8;
minutes: UInt8;
hours: UInt8;
dayofweek: UInt8;
dayofmonth: UInt8;
month: UInt8;
year: integer;
```

Fields containing date and time in following ranges:
seconds - (0-59)
minutes - (0-59)

hours - (0-23)
dayofweek - (1-7; Monday is 1, Sunday is 7)
dayofmonth - (1-31)
month - (1-12)
year - (2000-2099)