![MIKROE logo]

RTC 10 CLICK

PID: MIKROE-3770
Weight: 18 g

RTC 10 Click is a real-time clock module which has an extremely low power consumption, allowing it to be used with a single button cell battery, for an extended period of time. It features the DS3231M, a low-cost, extremely accurate, I2C real-time clock (RTC) from Maxim Integrated. The device incorporates a battery input and maintains accurate timekeeping when main power to the device is interrupted. The integration of the microelectromechanical systems (MEMS) resonator enhances the long-term accuracy of the device and reduces the piecepart count in a manufacturing line.

RTC 10 click is supported by a mikroSDK compliant library, which includes functions that simplify software development. This Click board™ comes as a fully tested product, ready to be used on a system equipped with the mikroBUS™ socket.

Features such as being highly accurate and providing a complete clock calendar funcionality with timekeeping accuracy of ±5ppm (±0.432 Second/Day) from -45°C to +85°C, make the RTC 10 click is a perfect solution for the development of the IoT,

wearable and portable applications, logging devices, industrial and health-related time metering applications, and all the other applications that require an accurate RTC for their operation.

HOW DOES IT WORK?

RTC 10 click is based on the DS3231M, a low-cost, extremely accurate, I2C real-time clock (RTC) from Maxim Integrated. Thanks to its high integration level, it provides high time accuracy, with a very low count of external components required. It has a full RTC function, offering programmable counters, alarms, and an interrupt engine with selectable event reporting sources. The small dimensions of the DS3231M itself, allow it to be used in very space-constrained applications, including wearables, medical equipment, and similar.
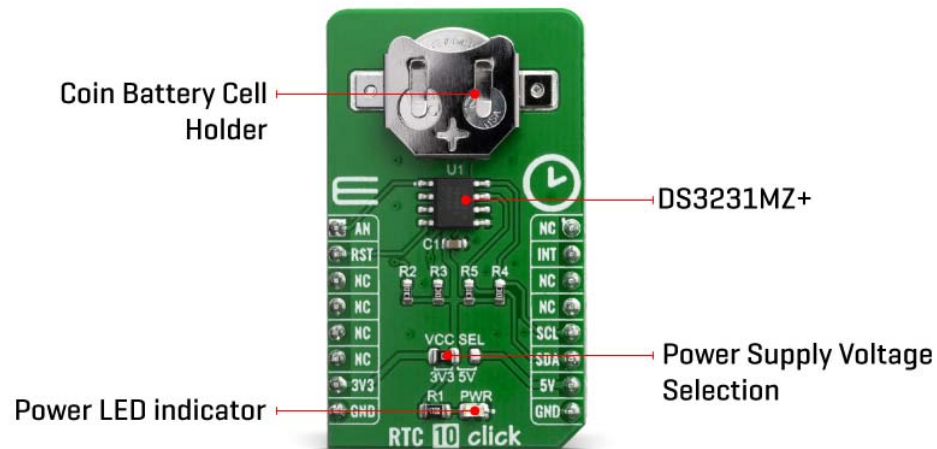
In addition to the DS3231M, RTC 10 click is equipped with the button cell battery holder compatible with the 3000TR batteryholder, suitable for 12mm Coin Cell batteries. By utilizing an automatic backup switch, the IC is able to use an external battery power source when there is no power supply on its main power terminals, thus allowing for uninterrupted operation.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of day alarms and a 1Hz output are provided. Address and data are transferred serially through an I2C bidirectional bus. A precision temperature-compensated voltage reference and comparator circuit monitors the status of VCC to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a microprocessor reset and is routed to the RST pin on the mikroBUS™ socket.

The temperature sensor, oscillator, and digital adjustment controller logic form the highly accurate time base. The controller reads the output of the on-board temperature sensor and adjusts the final 1Hz output to maintain the required accuracy. The device is trimmed at the factory to maintain a tight accuracy over the operating temperature range. When the device is powered by VCC, the adjustment occurs once a second. When the device is powered by VBAT, the adjustment occurs once every 10s to conserve power. Adjusting the 1Hz time base less often does not affect the device's long-term timekeeping accuracy. The device also contains an Aging Offset register that allows a constant offset (positive or negative) to be added to the factory-trimmed adjustment value.

The I2C interface is accessible whenever either VCC or VBAT is at a valid level. If a microcontroller connected to the device resets because of a loss of VCC or other event, it is possible that the microcontroller and device's I2C communications could become unsynchronized. e.g., the microcontroller resets while reading data from the device. When the microcontroller resets, the device's I2C interface can be placed into a known state by toggling SCL until SDA is observed to be at a high level. At that point the microcontroller should pull SDA low while SCL is high, generating a START condition.

SPECIFICATIONS

Type	RTC
Applications	RTC 10 click is a perfect solution for the development of the IoT, wearable and portable devices, logging devices, industrial and health-related time metering applications, and all the other applications that require an accurate time-base for various purposes.
On-board modules	DS3231M, a low-cost, extremely accurate, I2C realtime clock (RTC) from Maxim Integrated
Key Features	Highly accurate RTC with Integrated MEMS resonator
Interface	I2C
Click board size	M (42.9 x 25.4 mm)
Input Voltage	3.3V or 5V

PINOUT DIAGRAM

This table shows how the pinout on RTC 10 click corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin	mikroBUS				Pin	Notes
Analog ECG signal OUT	AN	1	AN	PWM	16	NC	
Reset Out	RST	2	RST	INT	15	INT	Interrupt OUT
	NC	3	CS	RX	14	NC	
	NC	4	SCK	TX	13	NC	
	NC	5	MISO	SCL	12	SCL	I2C Clock
	NC	6	MOSI	SDA	11	SDA	I2C Data
Power Supply	3.3V	7	3.3V	5V	10	5V	Power Supply
Ground	GND	8	GND	GND	9	GND	Ground

ONBOARD SETTINGS AND INDICATORS

Label	Name	Default	Description
PWR	PWR	-	Power LED Indicator
VCC SEL	VCC SEL	Left	Power supply voltage selection: left position 3.3V, right position 5V

SOFTWARE SUPPORT

We provide a library for the RTC 10 click on our LibStock page, as well as a demo application (example), developed using MikroElektronika compilers. The demo can run on all the main MikroElektronika development boards.

Library Description
The library covers all the necessary functions to control RTC 10 click board. A library performs a standard I2C interface communication.

Key functions:

- `void rtc10_setTime( uint8_t timeHours, uint8_t timeMinutes, uint8_t timeSeconds )` - Set time hours, minutes and seconds function.
- `void rtc10_getTime( uint8_t *timeHours, uint8_t *timeMinutes, uint8_t *timeSeconds )` - Get time hours, minutes and seconds function.
- `void rtc10_setDate( uint8_t dayOfTheWeek, uint8_t dateDay, uint8_t dateMonth, uint16_t dateYear )` - Set date hours, minutes and seconds function.

Examples description
The application is composed of three sections :

- System Initialization - Initializes GPIO, I2C and LOG structures, sets INT pin as input and RST pin as output and start to write log.

- Application Initialization - Initialization driver enable's - I2C, hardware reset, set start time and date, enable counting also, write log.

- Application Task - (code snippet) This is an example which demonstrates the use of RTC 10 Click board. RTC 10 Click communicates with register via I2C interface, set time and date, enable counting and display time and date values, also, display temperature value for every 1 sec. Results are being sent to the Usart Terminal where you can track their changes. All data logs write on Usart Terminal changes for every 1 sec.

```
void applicationTask()
{
    rtc10_getTime( &timeHours, &timeMinutes, &timeSeconds );

    Delay_10ms();


    rtc10_getDate( &dayOfTheWeek, &dateDay, &dateMonth, &dateYear );

    Delay_10ms();


    if ( secFlag !=  timeSeconds )
    {
        mikrobus_logWrite( " Time:  ", _LOG_TEXT );
```

```c
        displayLogUart( timeHours );
        mikrobus_logWrite( ":", _LOG_TEXT );

        displayLogUart( timeMinutes );
        mikrobus_logWrite( ":", _LOG_TEXT );

        displayLogUart( timeSeconds );
        mikrobus_logWrite( "", _LOG_LINE );

        mikrobus_logWrite( " Date: ", _LOG_TEXT );

        displayLogUart( dateDay );
        mikrobus_logWrite( ".", _LOG_TEXT );

        displayLogUart( dateMonth );
        mikrobus_logWrite( ".", _LOG_TEXT );

        mikrobus_logWrite( "20", _LOG_TEXT );

        displayLogUart( dateYear );
        mikrobus_logWrite( ".", _LOG_LINE );

        displayDayOfTheWeek( dayOfTheWeek );

        if ( timeSeconds == 0 )
        {
            temperature = rtc10_getTemperature();

            mikrobus_logWrite( " Temp.: ", _LOG_TEXT );
            FloatToStr( temperature, logText );
            ltrim( logText );
            rtrim( logText );
            mikrobus_logWrite( logText, _LOG_TEXT );
            mikrobus_logWrite( degCel, _LOG_LINE );
        }
```

```
        mikrobus_logWrite( "-------------------", _LOG_LINE );


        secFlag =  timeSeconds;

    }
}
```

Additional Functions :

- `void displayLogDayOfTheWeek( uint8_t dayOfTheWeek )` - Write day of the week log on USART terminal.
- `void displayLogUart( uint8_t value )` - Write the value of time or date as a two-digit number.
  The full application code, and ready to use projects can be found on our LibStock page.
  Other mikroE Libraries used in the example:

- I2C
- UART
- Conversions

Additional notes and informations
Depending on the development board you are using, you may need USB UART click, USB UART 2 click or RS232 click to connect to your PC, for development systems with no UART to USB interface available on the board. The terminal available in all MikroElektronika compilers, or any other terminal application of your choice, can be used to read the message.

MIKROSDK

This Click board™ is supported with mikroSDK - MikroElektronika Software Development Kit. To ensure proper operation of mikroSDK compliant Click board™ demo applications, mikroSDK should be downloaded from the LibStock and installed for the compiler you are using.

For more information about mikroSDK, visit the official page.

https://www.mikroe.com/rtc-10-click/10-15-19