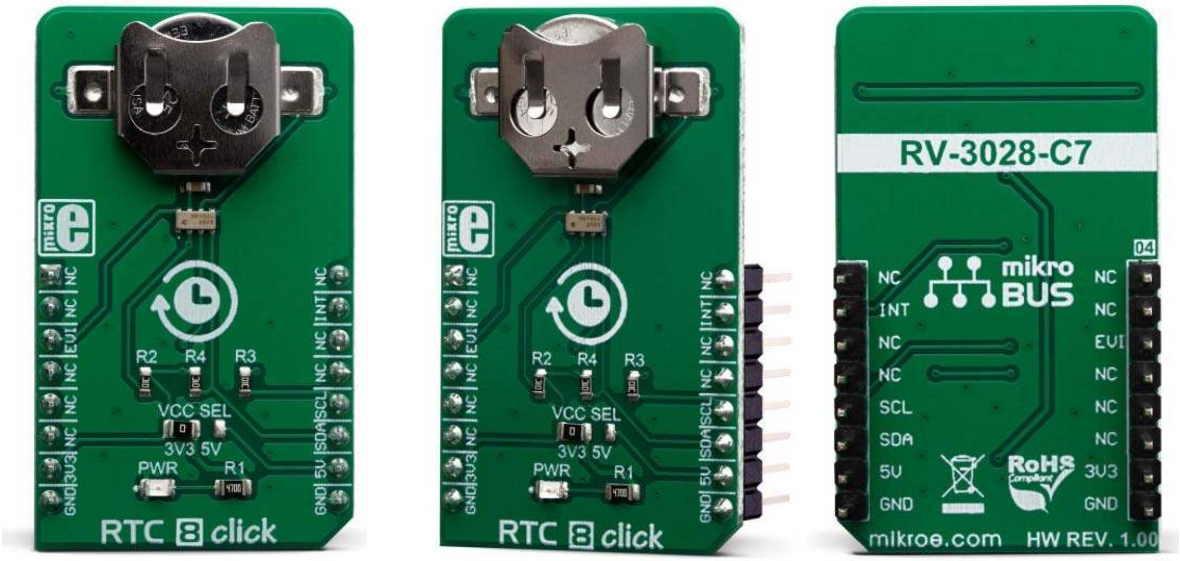


RTC 8 CLICK

PID: MIKROE-3456
Weight: 20 g

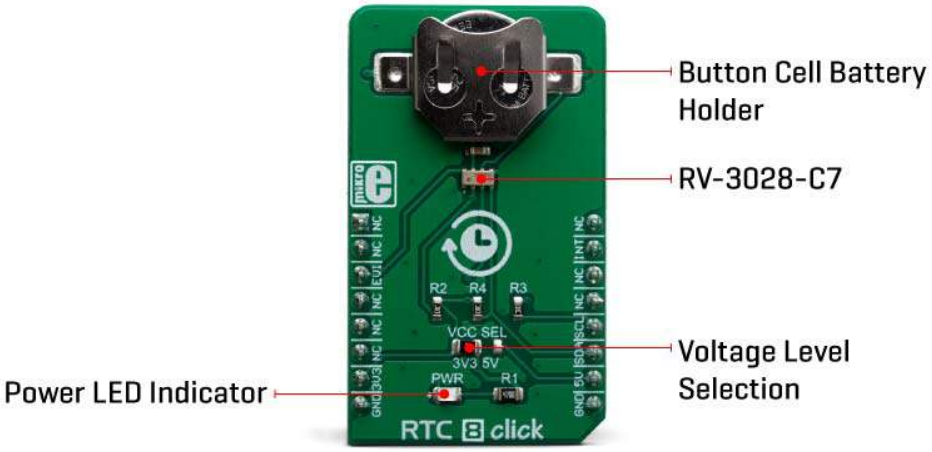
RTC 8 click is a real time clock module which has an extremely low power consumption, allowing it to be used with a single button cell battery, for an extended period of time. The RV-3028-C7 module built on the RTC 8 click is able to output the time in the standard format, as well as in the 32-bit UNIX format. Integrated, factory calibrated XTAL operating at 32.768 kHz ensures a very low time deviation. The compensation and offset data are stored within the internal non-volatile memory. There is also a possibility to password-protect the configuration parameters. An advanced interrupt feature allows many different uses such as alarm function, countdown timer function, external event detection function with time stamp and much more.

RTC 8 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 the ability to be powered over the button cell battery, external event capture pin, several time output formats including the 32-bit UNIX counter format, and above all – an extremely low power consumption, make RTC 8 click 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 8 click is based on the RV-3028-C7, an extreme low power real-time clock/calendar (RTC) module from Micro Crystal Switzerland. Thanks to its high integration level, this module provides high time accuracy, factory calibrated to 1 ppm, 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. In addition to a standard clock output function, it also offers a 32-bit UNIX Time counter. The operational parameters are stored within the internal non-volatile memory (EEPROM) allowing their persistence in the event of the complete power failure. The small dimension of the RV-3028-C7 module itself, allow it to be used in very space-constrained applications, including wearables, medical equipment, and similar.



In addition to the RV-3028-C7, RTC 8 click is equipped with the button cell battery holder compatible with the CR1225 battery. 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. Draining as low as 40nA of current, it can be operated with the standard button cell battery almost indefinitely. In addition, a trickle charge system will replenish the battery power while the RV-3028-C7 is powered over the main power terminals (VDD, VSS). The voltage of the main power supply can range between 1.2V up to 5.5V.

The RV-3028-C7 uses the I2C communication protocol for the communication with the host MCU. Besides the I2C bus lines, two additional pins are also available on the RV-3028-C7, allowing an interrupt to be reported to the host MCU, but also to capture an external event and marking it with an automatic timestamp. The user is able to set up standard clock and calendar functions (including seconds, minutes, hours, weekdays, date, months, years with leap year correction), as well as the interrupt functions for the periodic countdown timer, periodic time update, alarm, external event, automatic backup switchover and power on reset (POR) events. All these features are available when the module is operated over the backup power supply (battery)

A group of configuration registers is used to set up all the various working parameters of the device. To additionally prevent any unintentional changes of the internal registers, the RV-3028-C7 offers password protection of its configuration. If the password protection is set by an enable register in the non-volatile memory, each time the register configuration is attempted, the user will be required to enter the password first. Naturally, reading out the password registers will return 0 values; this register is write-only.

Besides other functions, EEPROM memory holds the offset correction values. Offset correction is required for fine-tuning the internal 32.768 kHz XTAL, as well as for compensating the aging phenomenon.

As mentioned before, the RV-3028-C7 is able to be operated across a wide voltage range, from 1.2V to 5.5V. This allows adding a small SMD jumper on the board, used to select between 3.3V and 5V for its main power supply.

SPECIFICATIONS

Type	RTC
Applications	RTC 8 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	RV-3028-C7, an extreme low power real-time clock/calendar (RTC) module from Micro Crystal Switzerland
Key Features	Extensive interrupt event reporting engine, ability to use a backup battery, trickle charging of the battery for extended operation, provides both standard time/date, as well as the UNIX 32-bit time counter, and more

Interface	I2C
Input Voltage	3.3V or 5V
Click board size	M (42.9 x 25.4 mm)

PINOUT DIAGRAM

This table shows how the pinout on **RTC 8 click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin	mikroBUS				Pin	Notes
	NC	1	AN	PWM	16	NC	
	NC	2	RST	INT	15	**INT**	Interrupt OUT
External Event IN	**EVI**	3	EVI	TX	14	NC	
	NC	4	SCK	RX	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

Designator	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 8 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 initializes and defines the I2C drivers and drivers that offer a choice of writing data in register and reading data from the register in BCD and DEC format. The library includes function for sets Time, Date, new Alarm and set/get UNIX time. The user can set and read Time and Date data, especially using the functions for reading and writing data. The library includes functions for setting the desired alarm, disabling alarm and checking whether an alarm has been activated.

Key functions:

- `uint8_t rtc8_readData(uint8_t reg)` - Read one byte data from register in DEC format.
- `uint32_t rtc8_getUNIX()` - Get current UNIX time.
- `uint8_t rtc8_setTime(uint8_t _hours, uint8_t _minutes, uint8_t _seconds)` - Set new start time - 24 hour format.
- `uint8_t rtc8_setDate(uint8_t _date, uint8_t _month, uint8_t _year)` - Set new start date.

Examples description
The application is composed of the three sections :

- System Initialization - Initializes I2C module and sets INT pin as INPUT and CS pin as OUTPUT.
- Application Initialization - Initialization driver init, settings RTC for work and sets start time/date and new alarm.
- Application Task - Read current Time, Date and UNIX time and checks if the alarm is active.

Note: Comment the lines for setting date and time if you would like the module to keep counting time after a reset or shut down.

```
void applicationTask()
{
    uint8_t seconds;
    uint8_t minutes;
    uint8_t hours;
    uint8_t _weekday;
    uint8_t date;
    uint8_t month;
    uint8_t year;
    uint8_t status;
    uint32_t UNIX_time;
    char demoText[50];

    // Time
    seconds = rtc8_readData(_RTC8_REG_SECONDS);
    minutes = rtc8_readData(_RTC8_REG_MINUTES);
    hours = rtc8_readData(_RTC8_REG_HOURS);
    // Date
    _weekday = rtc8_readData(_RTC8_REG_WEEKDAY);
    date = rtc8_readData(_RTC8_REG_DATE);
    month = rtc8_readData(_RTC8_REG_MONTH);
    year = rtc8_readData(_RTC8_REG_YEAR);
    status = rtc8_readByte(_RTC8_REG_STATUS);
    // Unix time
    UNIX_time = rtc8_getUNIX();



    mikrobus_logWrite( "Time:  ", _LOG_TEXT );
    // Houes
    IntToStr(hours, demoText);
    Ltrim(demoText);
    mikrobus_logWrite( demoText, _LOG_TEXT );
    // Minute
    mikrobus_logWrite(":", _LOG_TEXT);
    IntToStr(minutes, demoText);
    Ltrim(demoText);
    mikrobus_logWrite(demoText, _LOG_TEXT);
```

```
    // Seconds
    mikrobus_logWrite(":", _LOG_TEXT);
    IntToStr(seconds, demoText);
    Ltrim(demoText);
    mikrobus_logWrite(demoText, _LOG_LINE);

    mikrobus_logWrite( "Weekday: ", _LOG_TEXT );
    IntToStr(_weekday, demoText);
    Ltrim(demoText);
    mikrobus_logWrite( demoText, _LOG_LINE );

    mikrobus_logWrite( "Date:  ", _LOG_TEXT );
    // Date
    IntToStr(date, demoText);
    Ltrim(demoText);
    mikrobus_logWrite( demoText, _LOG_TEXT );
    // Month
    mikrobus_logWrite(".", _LOG_TEXT);
    IntToStr(month, demoText);
    Ltrim(demoText);
    mikrobus_logWrite(demoText, _LOG_TEXT);
    // Year
    mikrobus_logWrite(".", _LOG_TEXT);
    IntToStr(year, demoText);
    Ltrim(demoText);
    mikrobus_logWrite(demoText, _LOG_LINE);

    mikrobus_logWrite( "UNIX:  ", _LOG_TEXT );
    LongWordToStr(UNIX_time, demoText);
    Ltrim(demoText);
    mikrobus_logWrite( demoText, _LOG_LINE );

    if((status & 0x04)/4 != 0)
    {
        mikrobus_logWrite( " - Alarm Active!!!", _LOG_LINE );
        Delay_1sec();
        rtc8_writeByte(_RTC8_REG_STATUS, _RTC8_STATUS_REG_CLEAR); // Reset Alarm Flag
```

```
    }
    else
    {
        mikrobus_logWrite( " - No Alarm.", _LOG_LINE );
    }

    mikrobus_logWrite( "----------------------------------", _LOG_LINE );
    Delay_ms(1000);
}
```

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-8-click/6-5-19