

## EEPROM 4 click

**PID: MIKROE-2536**

**Weight: 24 g**

EEPROM 4 click is 2,097,152 bits on a click board™, organized into 262,144 bytes. In other words, this click board™ is an EEPROM memory medium with the capacity of 256 KB. The used EEPROM module has very good endurance and it can withstand 1,000,000 write cycles, with the data retaining period of about 40 years. The EEPROM module on this click can work with power supply voltage ranging from 1.7V to 5.5V, it features the self-timed write cycles, doesn't require erase before writing, has a dedicated write protect pin for hardware protection of stored data, and has a dedicated hold pin used for holding the data transfer.

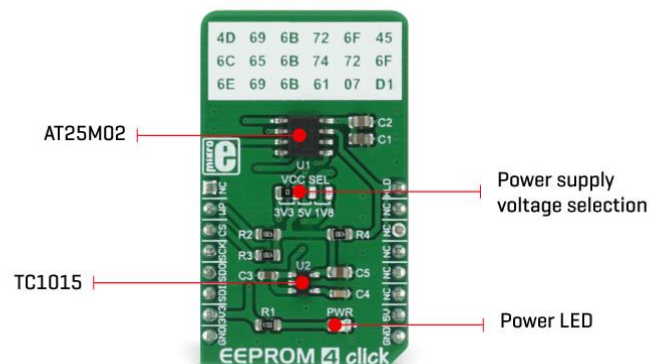
EEPROM 4 click is aimed towards industrial and commercial applications, which require low voltage and low power operational capabilities. It can be used for any kind of temporary or permanent data storage for various embedded electronic devices, simple data logging, storing various working parameters of a module or device, safeguarding the sensitive data in case of a power cycle, and other similar applications where EEPROM memory is needed.

## How the click works

The EEPROM module used on the EEPROM 4 click is the AT25M02, an SPI serial EEPROM from Microchip, with the memory cell density of 2 Mbits. The EEPROM density is usually expressed in bits, so exactly 2,097,152 bits are organized in units or words of 8 bits, which gives 262,144 bytes of data memory. Furthermore, the EEPROM is organized in so-called pages. One page holds 256 bytes and there are 1024 pages (1024 pages x 256 bytes = 262,144 bytes total). Having insight into how the memory cells are organized, is important for write and erase operations. The SPI pins are routed to the mikroBUS™ so the communication is easy and straightforward. The SPI can be clocked as high as 5 MHz, providing a fast throughput for the data transfer.

Some other EEPROM memory modules require erasing of the whole memory page before writing new data. This EEPROM doesn't require such operations. It features byte write and page write modes. Before attempting any write operations to the EEPROM, the write enable bit (WEL) of the Status Register needs to be set to 1. This bit is automatically set to 0 after some instructions. There are special instructions used to set and clear the WEN bit of the Status Register. These instructions are WREN (06h) and WRDI (04h). Usually, every write instruction will be prefixed with the WREN instruction.

- When using a Byte write instruction, one single byte will be written during the write cycle. After successful write cycle, the state of the WEL bit is set to 0 automatically and the device is ready to accept another byte to be written.
- When using a Page write instruction, 256 bytes will be written during one write cycle. It is possible to write less than 256 bytes, but if there is still incoming data when the end of the page is reached, the write pointer will roll over to the beginning of the same page, overwriting the content there. The special care should be taken to the page alignment. After successful write cycle, the state of the WEL bit is set to 0 automatically and the device is ready to accept another page of data to be written.



A dedicated #HOLD pin is routed to the PWM pin of the mikroBUS™. When the communication with the click board™ is initiated by setting the CS pin to a LOW logic state, it is possible to pause the serial data transfer without resetting the communication, if the #HOLD pin (PWM pin on the mikroBUS™) is set to a LOW logic state. To resume the communication, it is enough to set this pin to a HIGH logic state while the SCK is still running. Once the HOLD is initiated, the state of the SCK line is irrelevant and any serial data input will be ignored. This pin is pulled HIGH by the onboard resistor.

A dedicated #WP write protect pin is used to put the device into the hardware write protect mode. This pin is routed to the RST pin of the mikroBUS™. Hardware write protect works in conjunction with the Write Protect Enable (WPEN) bit of the Status Register. When this bit is set to 1 and the #WP pin is set to a LOW logic state, the device will ignore any attempt to write to the Status Register and the EEPROM memory regions, selected by the Block Write Protect bits of the Status Register (BP0 and BP1). WRSR instruction is used to write to the Status Register (01h). Again, before attempting to write to Status Register, WREN instruction should be executed first. Once the WPEN bit is set to 1 and the RST has been pulled to a LOW logic state, setting the WPEN bit to 0, won't disable write protection, as long as the #WP pin (RST) stays LOW. WPEN bit, as well as the BP0 and BP1 bits, are constructed as EEPROM cells, meaning that they are nonvolatile and will retain their states even after power off. The #WP pin is pulled HIGH by the onboard resistor.

The onboard SMD jumper labeled as VCCSEL is used to select the operating voltage between 3.3V and 5V, as usual. However, there's the third position for this jumper, which is used to set the operating voltage to 1.8V. This is achieved thanks to the TC1015, a small 100 mA LDO from Microchip, which is powered from the 5V rail.


As always, MikroElektronika provides libraries that simplify and speed up working with this device. The provided application example demonstrates the functionality of the provided libraries and can be used as the reference point for own development.

## Specifications

Type	EEPROM
Applications	For any kind of temporary or permanent data storage for various embedded electronic devices, simple data logging, storing various working parameters of a module or device, safeguarding the sensitive data in case of a power cycle, etc.
On-board modules	AT25M02, an SPI serial EEPROM from Microchip, with the memory cell density of 2 Mbits, TC1015, a small 100 mA LDO from Microchip
Interface	GPIO,SPI
Input Voltage	3.3V,5V
Click board size	M (42.9 x 25.4 mm)

## Pinout diagram

This table shows how the pinout on EEPROM 4 **click** corresponds to the pinout on the mikroBUSTM socket (the latter shown in the two middle columns).

Notes	Pin					Pin	Notes
		1	AN	PWM	16		
	NC	1	AN	PWM	16	HLD	Hold pin
Write protect pin	WP	2	RST	INT	15	NC	
Chip select pin	CS	3	CS	RX	14	NC	
Clock pin	SCK	4	SCK	TX	13	NC	
Slave data out pin	SDO	5	MISO	SCL	12	NC	
Slave data in pin	SDI	6	MOSI	SDA	11	NC	
Power Supply	+3V3	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 indication LED
JP1	VCC SEL	Left	Power supply voltage selection 3V3/5V/1V8, left position 3V3, middle position 5V, right position 1V8

## Software support

We provide a library for EEPROM 4 click on our LibStock page, as well as a demo application (example), developed using MikroElektronika compilers and mikroSDK. The provided click library is mikroSDK standard compliant. The demo application can run on all the main MikroElektronika development boards.

### Library Description

Initializes and defines SPI bus driver, driver functions which communicate with the memory array and configures click to the writable state.

## Key functions

`uint8_t eeprom4_sendCommand(uint8_t command_byte);` - The function sends command (instruction) to the click. In case that command byte is `_EEPROM4_LOW_POWER_WRITE_POLL_COMMAND` (0x08) function returns 0x00 if a part is not in a write cycle and returns 0xFF if the part is still busy completing the write cycle. In other cases, the function returns 0.

`void eeprom4_writeStatusReg(uint8_t data_value);` - The function writes data determined in the parameter of the function to the status register.

`uint8_t eeprom4_readStatusReg(void);` - The function reads a one-byte data value from the status register.

`void eeprom4_writeMemory(uint32_t memory_address, uint8_t *data_input, uint8_t nBytes);` - The function writes a number of bytes determined by `nBytes` parameter from buffer determined by `data_input` pointer to memory location determined by `memory_address` parameter.

`void eeprom4_readMemory(uint32_t memory_address, uint8_t *data_output, uint8_t nBytes);` - The function reads a number of bytes determined by `nBytes` parameter from memory location determined by `memory_address` parameter and stores bytes to buffer determined by `data_output` pointer.

`void eeprom4_enableWriteProtect(uint8_t state);` - The function sets RST pin on state value to enable or disable writing to status register and memory array. WP pin is used in conjunction with the block protection bits of the status register and with WPEN and WEL bits also.

`void eeprom4_enableHoldOperation(uint8_t state);` - The function enables or disables the Hold operation. To pause the serial communication with the master device without resetting the serial sequence, the HOLD pin must be brought low. To resume serial communication, HOLD pin must be brought high.

`uint8_t eeprom4_checkStatusReg(uint8_t check_bit);` - The function checks the value of the status register bit determined by `check_bit` parameter.

## Examples Description

The application is composed of three sections :

- System Initialization - Initializes peripherals and pins.
- Application Initialization - Initializes click driver and configures click that all memory block is unprotected. Also configures that HOLD operation is disabled, and the memory and status register are writable.
- Application Task - (code snippet) - Enables writing to the memory array, writes data from the buffer to the memory, checks if the part is in a write cycle, and if is not reads data from memory array and stores data to buffer. Shows data on USB UART.

```

void applicationTask()
{
    eeprom4_sendCommand(_EEPROM4_SET_WRITE_ENABLE_LATCH_COMMAND);

    eeprom4_writeMemory(_EEPROM4_FIRST_MEMORY_LOCATION, &data_write[0], 8);

    cnt = eeprom4_checkStatusReg(_EEPROM4_READY_BIT);

    check_state = eeprom4_sendCommand(_EEPROM4_LOW_POWER_WRITE_POLL_COMMAND);

    while(cnt | check_state)
    {
        cnt = eeprom4_checkStatusReg(_EEPROM4_READY_BIT);

        check_state =
eeprom4_sendCommand(_EEPROM4_LOW_POWER_WRITE_POLL_COMMAND);
    }

    eeprom4_readMemory(0x00000001, &data_read[0], 6);

    for(cnt = 0; cnt < 6; cnt++)
    {
        WordToStr(data_read[cnt], text);
        mikrobus_logWrite(text, _LOG_TEXT);
        if(cnt < 5)
        {
            mikrobus_logWrite(",", _LOG_TEXT);
        }else
        {
            mikrobus_logWrite("", _LOG_LINE);
        }
    }

    Delay_ms(2000);
}

```

The full application code, and ready to use projects can be found on our [LibStock](#) page.

Other MikroElektronika Libraries used in the example:

- Conversions
- SPI
- UART

## Additional notes and information

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.

## Downloads

mikroBUS™ standard specifications

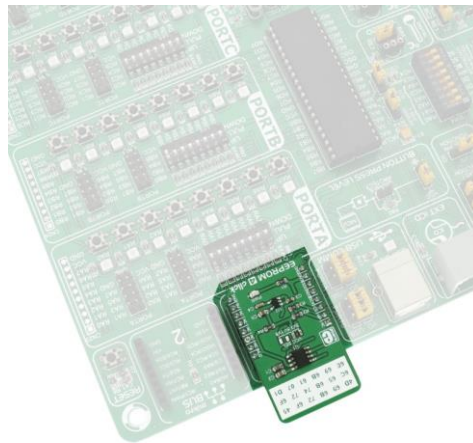
LibStock: mikroSDK

EEPROM 4 click schematic

LibStock: EEPROM 4 click library

AT25M02 datasheet

EEPROM 4 click - 2D and 3D files



<https://www.mikroe.com/EEPROM-4-click> 1-24-18