

# 1-Wire I2C click

PID: MIKROE-2750

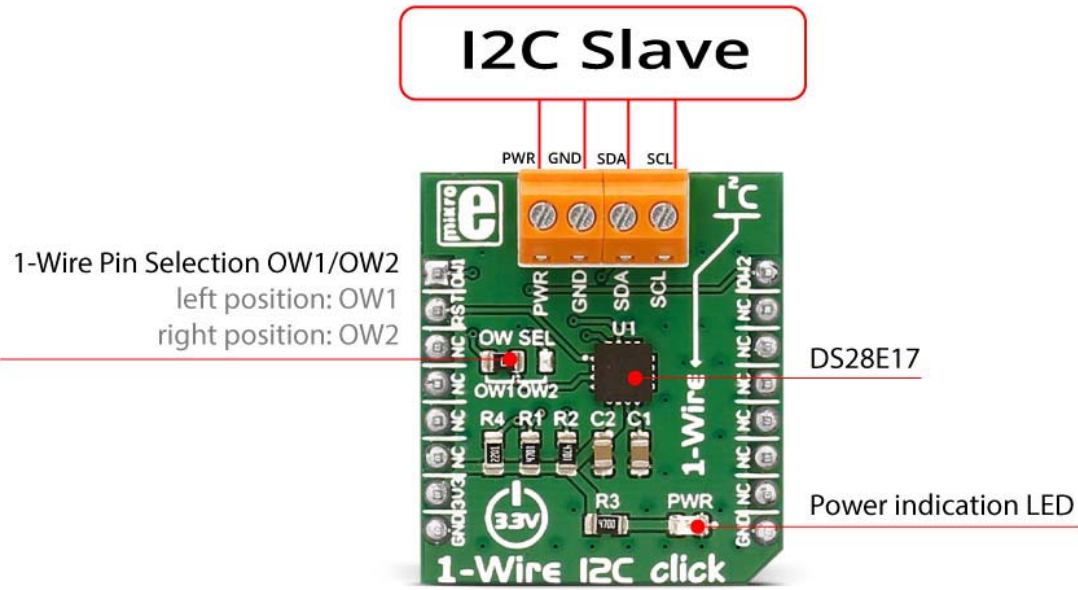


**1-Wire I2C click** carries DS28E17 1-Wire-to-I2C master bridge from Maxim Integrated. The click runs on a 3.3V power supply.

It communicates with the target microcontroller over 1-Wire® protocol, using the following pins on the mikroBUS™ line: AN, PWM, RST.

## How the click works

There are two on-board screw terminals used for connecting SCL, SDA, Vcc and GND of the I2C slave. After that, you are able to communicate with that slave through the onboard DS28E17 MCU.



### DS28E17 features

The DS28E17 is a 1-Wire slave to I2C master bridge device that interfaces directly to I2C slaves at standard (100kHz max) or fast (400kHz max). Data transfers serially through the **1-Wire® protocol**, which requires only a single data lead and a ground return. Every DS28E17 is guaranteed to have a unique 64-bit ROM registration number that serves as a node address in the 1-Wire network.

### Specifications

Type	1-wire
Applications	1-Wire I2C click can be used to extend the length of I2C lines by converting I2C to 1-wire
On-board modules	DS28E17 1-Wire-to-I2C master bridge from Maxim Integrated
Interface	1-wire,GPIO
Input Voltage	3.3V
Click board size	S (28.6 x 25.4 mm)

## Pinout diagram

This table shows how the pinout on **1-Wire I2C click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin					Pin	Notes
1-Wire 1st pin	<b>OW1</b>	1	AN	PWM	16	<b>OW2</b>	1-Wire 2nd pin
Reset pin	<b>RST</b>	2	RST	INT	15	NC	
	NC	3	CS	TX	14	NC	
	NC	4	SCK	RX	13	NC	
	NC	5	MISO	SCL	12	NC	
	NC	6	MOSI	SDA	11	NC	
Power supply	<b>+3.3V</b>	7	3.3V	5V	10	NC	
Ground	<b>GND</b>	8	GND	GND	9	<b>GND</b>	Ground

## Programming

Code examples for 1-Wire I2C click, written for MikroElektronika hardware and compilers are available on Libstock.

### *Code snippet*

The following code snippet shows 1-Wire I2C click communication with the Thermo 4 click. It uses `skipRom`, `writeI2C`, and `readI2C` commands to read the temperature data from the sensor, and displays it back via UART output.

```

01 void One_Wire_I2C_Task()
02 {
03     char uartText [20];
04     char IWireData [20];
05
06     IWireData [0] = 0x00;    //Thermo 4 temperature register address
07
08     OWI2C_skipRom ();
09     OWI2C_writeI2C ( OWI2C_WRITE_NORMAL_NOSTOP, 0x48, 1, &IWireData);
//0x48 is unshifted I2C address of Thermo 4 click
10     OWI2C_skipRom ();
11     OWI2C_readI2C ( 0x48, 2, &IWireData);
12
13     UART_Write (13);
14     UART_Write (10);
15     UART_Write_Text("Current temperature is: ");
16     ByteToStr(IWireData[0], uartText);
17     UART_Write_Text(uartText);
18
19     Delay_ms (2000);
20 }

```