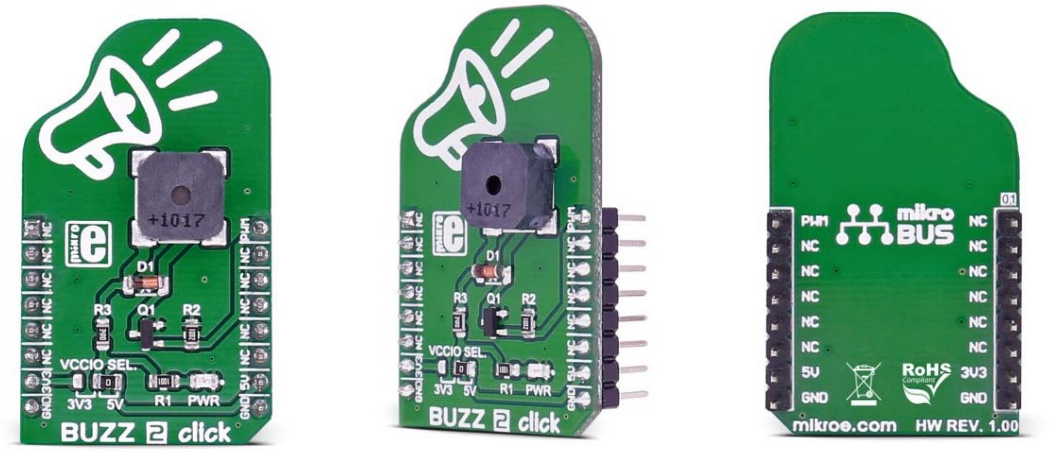
MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

BUZZ 2 click

PID: MIKROE-2720



BUZZ 2 click carries the CMT-8540S-SMT magnetic buzzer transducer. The buzzer’s resonant frequency is 4kHz. The click is designed to run on either 3.3V or 5V power supply.

How the click works

The CMT-8540S-SMT magnetic buzzer is controlled by the PWM pin on the mikroBUS™ line.

You can create different sound patterns using the Sound library supported in our compilers, or utilize the microcontroller internal PWM module to create the signal for the buzzer. Signal frequency determines the sound pitch, and the duty cycle determines the amplitude (sound volume).

Power supply selection

Onboard VCCIO SEL zero-ohm resistor (SMD jumper) is used to determine whether 5V or 3.3V power supply is used. This resistor is placed in 5V position by default.

Specifications

Type	Magnetic
Applications	The click is ideal for adding audio signalization feature to your prototype device
On-board modules	CMT-8540S-SMT magnetic buzzer transducer
Key Features	4kHz resonant frequency
Interface	PWM
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 **BUZZ 2 click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	**Pin**	mikro™ BUS				**Pin**	**Notes**
	NC	1	AN	PWM	16	**PWM**	PWM input
	NC	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	**+3.3V**	7	3.3V	5V	10	**+5V**	Power supply
Ground	**GND**	8	GND	GND	9	**GND**	Ground

Programming

Code examples for BUZZ 2 click, written for MikroElektronika hardware and compilers are available on Libstock.

Code snippet

The following code snippet defines the notes and initializes the Sound library.

```
// notes
#define cS  261
#define dS  294
#define e   329
#define fS  349
#define g   391
#define gS  415
#define aHa 440
#define aS  455
#define b   466
#define cH  523
#define cSH 554
#define dH  587
#define dSH 622
#define eH  659
#define fH  698
#define fSH 740
#define gH  784
#define gSH 830
#define aH  880
void systemInit()
{
    AD1PCFG = 0xFFFF;
    TRISD = 0;
}
void Buzz_2_Init()
{
    Sound_Init(&LATD, 0);
}
void Buzz_2_Task()
{
    Sound_Play(aHa, 500);
    Sound_Play(aHa, 500);
    Sound_Play(aHa, 500);
    Sound_Play(fS, 350);
    Sound_Play(cH, 150);
    Sound_Play(aHa, 500);
    Sound_Play(fS, 350);
    Sound_Play(cH, 150);
    Sound_Play(aHa, 650);
    delay_ms(500);
    Sound_Play(eH, 500);
    Sound_Play(eH, 500);
    Sound_Play(eH, 500);
    Sound_Play(fH, 350);
    Sound_Play(cH, 150);
    Sound_Play(gS, 500);
    Sound_Play(fS, 350);
    Sound_Play(cH, 150);
    Sound_Play(aHa, 650);
}
void main()
{
    systemInit();
    Buzz_2_Init();
    while( 1 )
    {
        Buzz_2_Task();
    }
}
```