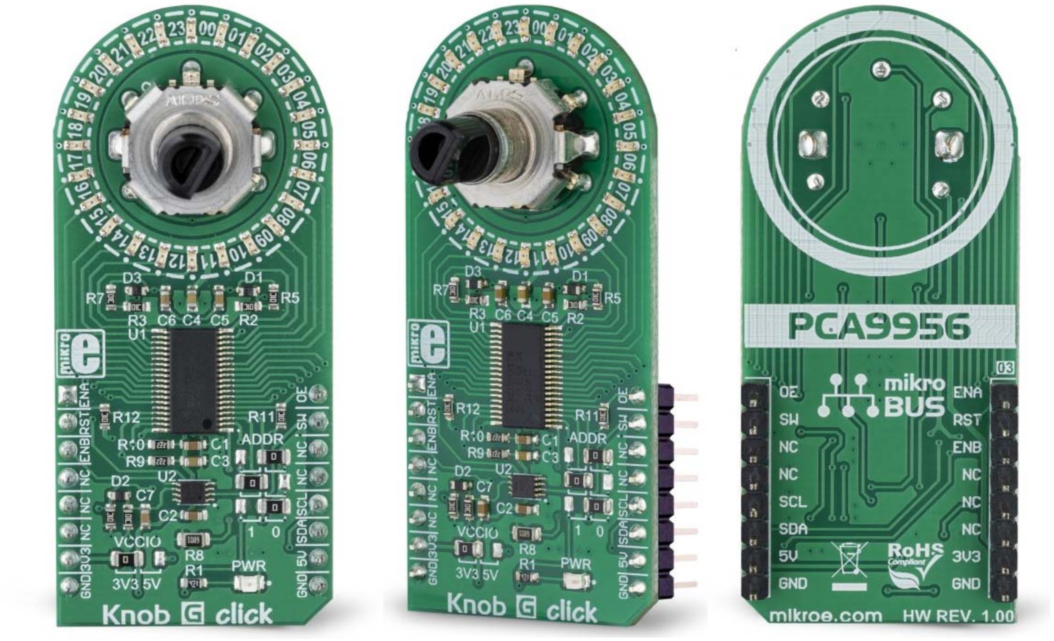
MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS



Knob G Click

PID: MIKROE-3299
Weight: 27 g

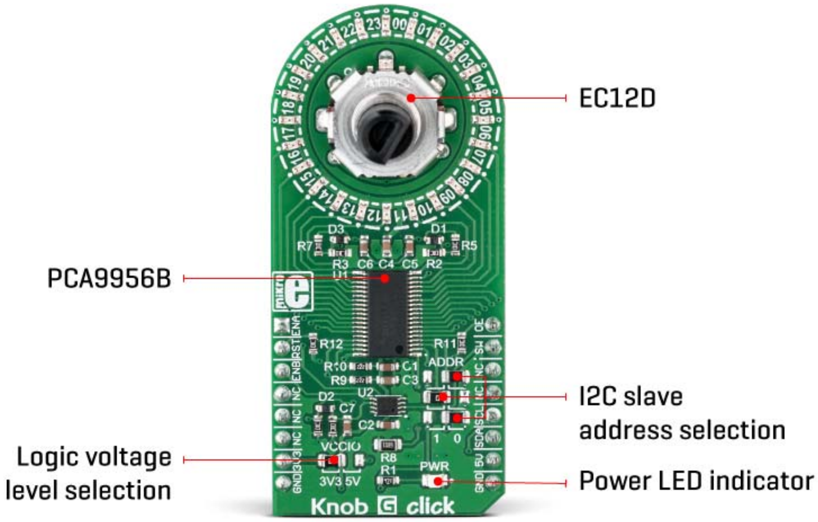
Knob G click features a combination of high-quality quadrature rotary encoder, and a LED ring composed of 24 individual green LEDs. Outputs of the rotary encoder are processed by a debouncing circuitry, which ensures reliable switching action, making the encoder directly usable in interrupt service routines (ISR) without the need for a bulky debouncing code. The 24-segment LED ring is driven by an advanced and flexible LED driver IC, which can dim each LED individually, allowing various lighting effects to be programmed. The LED ring is independent of the encoder, and it can be used as a very flexible position or value indicator.

It comes in the package which also includes the mikroSDK™ software and a library with all the functions. The Click board™ comes as a fully tested and approved prototype, making it a reliable device ready to use on the development board.

Featuring highly advanced LED driver IC which offers an unparalleled flexibility, 24 individual LEDs which form a ring around the encoder, a high-quality rotary encoder with push-button by ALPS, and the debouncing circuitry based on a triple Schmitt trigger IC, Knob G click is an ideal solution for building various HMI applications where precise input is required. Also, it can be used to add various interesting visual effects to any application.

How does it work?

Knob G click consists of two distinctive sections: the first section is the rotary quadrature encoder with its outputs routed to the GPIO pins of the mikroBUS. The encoder is debounced by a dedicated circuitry composed of passive elements and a triple inverting Schmitt trigger IC. The second section is the LED driver IC, with accompanying LEDs positioned in a form of a ring around the encoder, making them perfectly suited for encoder position indicators.



This Click Board uses the EC12D, a 15-pulse incremental rotary encoder with a push-button, from ALPS. This encoder has very good mechanical specifications: debouncing time for its internal switches goes down to 2ms, and it can withstand a huge number of switching cycles, up to 30,000. The supporting debouncing circuitry allows contacts to fully settle before the output is triggered.

When encoder contacts are closed, the capacitor will start to discharge through the resistor, via the contacts, and to the GND. The Schmitt trigger connected to the capacitor will output a LOW logic level when its low voltage threshold level is reached; and vice versa - when encoder contacts are open, the capacitor will start to charge through the resistor, and the Schmitt trigger will output a HIGH logic level, when its high voltage threshold level is reached. This allows reading the states of two encoder contacts and one push-button contact directly from the code, with no bulky software debouncing applied. This makes Knob G click usable directly within the the interrupt-on-change ISR, allowing an absolute accuracy and no skipped pulses (which might occur when a regular software polling technique is used). Output pins of the encoder contacts are labeled as ENA and ENB for the quadrature encoder contacts, and SW for the push-button contact. These pins are routed to the AN, CS, and INT pin of the Mirko BUS, respectively.

The LED ring is composed of 24 individual green LEDs which are driven by the PCA9956B, an 8-bit, 24-channel, constant current LED driver, from NXP. This driver IC has many LED driving features, including constant current sinking capability, which greatly simplifies the design: maximum current through LEDs is determined by a single resistor. The PCA9956B has registers for controlling each channel individually, along with a single register which controls all channels at once. It supports PWM-mode dimming, as well as the current-mode dimming (by scaling down the maximum LED current). The PCA9956B can use a lower frequency signal from a secondary integrated PWM oscillator to modulate the PWM dimming signal. While the PWM frequency of the driver is fixed at 31.25kHz reducing the visible LED flickering completely, the modulating low-frequency signal can range from 0 to 122Hz, allowing interesting blinking effects to be produced without using the computing power of the microcontroller (MCU). The driver produces smooth dimming of LEDs, since the resolution of the PWM duty cycle is 8 bits. More details and features can be found found in the PCA9956B datasheet, in the download section, below.

The OE pin of the PCA9956B is routed to the PWM pin of the mikroBUS. When a LOW logic level is applied to the OE pin, LED outputs will be enabled. This pin is pulled to a HIGH logic level by a resistor. The OE pin can also be driven by an external PWM signal, offering an alternative way of dimming all LEDs at once.

LED driver IC can be reset by pulling the RST pin to a LOW logic level. This pin is pulled to a HIGH level by the resistor. The reset pulse can be very short (2.5 µs) but the device will not be ready for another 1.5 ms after the pulse. Besides the hardware reset, the PCA9956B also supports a software reset, which is required if the device is going to be operated in the fast I2C mode (including clock speed above 100kHz). The datasheet of the PCA9956B offers a detailed explanation on performing the software reset and using the fast I2C mode (FM+)

The PCA9956B IC uses the 3.3V rail of the mikroBUS as the LED power supply, so there is no significant voltage drop causing thermal dissipation. However, turning all LEDs ON at once with the maximum current set in the current registers, might cause the Knob G click to dissipate some heat. This is expected, as the thermal dissipation of each channel is adding up to the sum dissipation of the entire IC. The PCA9956B IC features thermal shutdown protection, along with the set of error reporting features. The PCA9956B can report both open-circuit event and short-circuit event for each LED. These errors will be written in the ERR registers, one for each LED channel. The datasheet explains how to interpret the values of these registers.

Slave I2C address of the PCA9956B device can be selected by using three SMD jumpers, grouped under the ADDR label. The PCA9956B allows its slave I2C address to be selected from a wide range of 125 different values. Each of the address pins (A0 to A3) can be left floating, pulled up, pulled down and shorted to VCC or GND. However, some I2C addresses are reserved, so they should be used with care. The datasheet of the PCA9956B offers tables with resistor values for each state of the address pins and reserved I2C addresses. Knob G click uses three 0 Ω SMD jumpers to set the states of these address pins.

The logic voltage level can be selected by the SMD jumper, labeled as VCCIO. This jumper determines levels for the logic signals from Schmitt triggers, as well as for the I2C interface of Knob G click. This allows it to be interfaced with a wide range of different MCUs, both compatible with 3.3V and 5V logic voltage levels.

Specifications

Type	Rotary encoder
Applications	Knob G click is an ideal solution for building various HMI applications, where a precise input is required. Also, it can also be used to add various interesting visual effects to any application.

On-board modules	PCA9956B, an 8-bit, 24-channel, constant current LED driver, from NXP; 74HC3G14DC, a triple inverting Schmitt trigger IC, from Nexperia
Key Features	A reliable symmetrical debouncing circuit, a very flexible and powerful PWM LED driver which features several different dimming options, constant current through LEDs can be selected by a single resistor, smooth dimming in 256 steps for each LED channel, 24-segment LED ring that can be used for the visual feedback, etc.
Interface	I2C,GPIO
Input Voltage	3.3V,5V
Compatibility	mikroBUS
Click board size	L (57.15 x 25.4 mm)

Pinout diagram

This table shows how the pinout on the **Knob G Click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin	mikro™ BUS				Pin	Notes
Encoder OUT A	**ENA**	1	AN	PWM	16	**OE**	Control PWM Signal
PCA9956B Reset	**RST**	2	RST	INT	15	**SW**	Encoder Button OUT

Encoder OUT B	**ENB**	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	**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 LED indicator
JP0, JP2	ADDR	Right	I2C slave address selection: left position GND (0), right position VCC (1)
JP1	ADDR	Left	I2C slave address selection: left position GND (0), right position VCC (1)
JP3	VCCIO	Left	Logic voltage level selection: left position 3.3V, right position 5V

Note: please consult the datasheet of the PCA9956B if you need to change the default I2C slave address

Software support

We provide a library for the Knob G 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 bus driver and drivers that offer a choice for writing data in the register and reads data from the register. The library includes the function for reading the Encoder position and sets a new start position. The user also has the functions for full control LEDs.

Key functions:

- `void knobg_setLedState(uint8_t led, uint8_t state)` - Functions for setting the led state
- `void knobg_getEncoderPosition(int32_t *position, uint8_t *dir)` - Functions for getting Encoder position
- `void knobg_encoderNewStartPosition(int32_t position)` - Functions for set new start encoder position

Examples description
The application is composed of the three sections :

- System Initialization - Initializes I2C module, sets INT pin, AN pin and CS pin as INPUT and sets RST pin and PWM pin as OUTPUT
- Application Initialization - Initialization driver init, reset the device, enable led output, set output gain on maximum and sets new start position of the encoder.
- Application Task - (code snippet) - The Task application has 3 test modes: * The first example is setting BRIGHTNESS on all LEDs. * Other examples put the LED in the position read from the encoder. * The third example sets the LED to be read while the encoder registers the clockwise movement and turn off those LEDs that the encoder reads when moving in a counterclockwise direction. * The example is changed by pressing the SW button

```
void applicationTask()
{
    knobg_getEncoderPosition(&encoder_newPosition, &direction);

    if(knobg_getSWButtonState() == 0)
    {
        SW_State++;
        if(SW_State >= 3) SW_State = 0;

        knobg_setBrightness(_KNOBG_BRIGHTNESS_ALL_LED, 0x00);
        Delay_ms( 300 );
    }

    /* Logs position */
    if(encoder_newPosition != encoder_oldPosition)
    {
        WordToStr(encoder_newPosition, demoText);
        mikrobus_logWrite(" EnCoder position : ", _LOG_TEXT);
        mikrobus_logWrite(demoText, _LOG_LINE);
    }
    encoder_oldPosition = encoder_newPosition;

    switch(SW_State)
    {
        /* Brightness */
        case 0:
        {
            cnt++;
            if(cnt > 127) cnt = 0;
            knobg_setBrightness(_KNOBG_BRIGHTNESS_ALL_LED, cnt);
            Delay_ms( 15 );
            break;
        }
        /* Encoder with one led*/
        case 1:
        {
            if(encoder_newPosition > 24) knobg_encoderNewStartPosition( 1 );
            if(encoder_newPosition < 1) knobg_encoderNewStartPosition( 24 );

            if(direction == 1)
            {
                knobg_setLedState(encoder_newPosition, _KNOBG_LED_ON);
                knobg_setLedState(encoder_newPosition - 1, _KNOBG_LED_OFF);
            }
            else
            {
                knobg_setLedState(encoder_newPosition, _KNOBG_LED_ON);
                knobg_setLedState(encoder_newPosition + 1, _KNOBG_LED_OFF);
            }
            Delay_1ms();
            break;
        }
        /* Encoder with all led */
        case 2:
        {
            if(encoder_newPosition > 24) knobg_encoderNewStartPosition( 1 );
            if(encoder_newPosition < 1) knobg_encoderNewStartPosition( 24 );

            if(direction == 1)
            {
                knobg_setLedState(encoder_newPosition, _KNOBG_LED_ON);
            }
            else
            {
                knobg_setLedState(encoder_newPosition + 1, _KNOBG_LED_OFF);
            }
            Delay_1ms();
            break;
        }
    }

}
```

The full application code, and ready to use projects can be found on our LibStock page.

Other mikroE Libraries used in the example:

- `I2C`

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.