

## PROXIMITY 7 CLICK PID: MIKROE-3330 Weight: 26 g

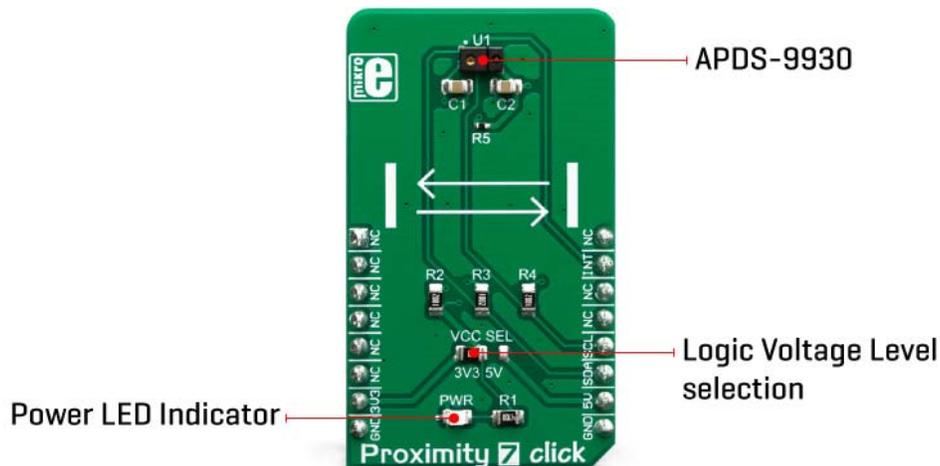
**Proximity 7 click** is an advanced proximity and ambient light sensing Click board™. It features the ADPS9930, a digital sensor IC equipped with two photodiodes (PD) and an IR LED, driven by a proprietary LED driver circuit. It allows an accurate proximity detection for a maximum distance of 100mm. The proprietary LED current driving technique eliminates the need for calibration, simplifying the design. A huge dynamic range allows ambient light sensing closely matched to the response of the human eye, in a variety of light conditions. An extensive interrupt engine allows for a firmware optimization.

Proximity 7 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.

With its ability to accommodate to different light conditions especially behind a dark glass, the ADPS9930 represents an ideal solution for dimming TFT and LCD displays on various devices. Positioned behind a semi-transparent bezel, it can still detect proximity and ambient light amount, regulating the brightness of the screen, accordingly, saving a lot of power that way. The ADPS9930 IC has a very low power consumption itself, that can be further optimized. This makes it a perfect choice for a range of applications that rely on the proximity and ambient light sensing (ALS), including PC and laptop monitors, POS displays, embedded displays, proximity-activated short-range security, etc.

## HOW DOES IT WORK?

The main component of the Proximity 7 click is the ADPS9930, a digital ambient and proximity sensor, by Broadcom (Avago Technologies). It is an accurate and reliable proximity and ambient light sensor, aimed towards the power saving in applications that use TFT or LCD panels. By offering a huge dynamic range, the ADPS9930 sensor allows to be placed behind a dark glass or a semi-transparent screen bezel, but also to be exposed to a bright sunlight. A proprietary design of the integrated constant-current LED driver enables plug and play proximity detection up to 100mm, eliminating the need for a calibration procedure. By integrating micro-optics elements within the casing, ADPS9930 greatly simplifies the application design.



Proximity of an object is detected using an IR LED, which emits pulses of light towards the object. The amount of the reflected IR light is measured by an integrated IR photodiode on channel 1. During LED ON time, the amount of the reflected IR light is measured and integrated. The background IR light is also measured and integrated, during LED OFF time. It is then subtracted from the final result, allowing for an accurate measurement with the reduced amount of the background IR noise. After it has been

scaled to a 16-bit value, the final result is available on the output registers, in the LOW/HIGH byte format.

Commonly, photosensitive elements are most sensitive to IR light. A human eye does not detect IR light. Therefore, the PD element has to filter out IR light so that only the visible part of the light is allowed through. The channel0 is equipped with such PD, making it usable for the ALS sensing. During the ALS measurement, both channels are measured. The datasheet of the ADPS9930 offers a conversion formula that can be used to obtain results in physical units (lx). These formulas also take the IR measurement from the channel 1 into the consideration, completely reducing its influence on the final result. By adjusting the integration time (also known as oversampling), the flickering effect of a fluorescent light can be completely eliminated.

The extensive interrupt engine allows an optimized firmware to be written. Four registers are used to specify the low and the high threshold for the ALS and proximity measurements. Whenever these thresholds are exceeded, an interrupt status bit will be set in the respective register. The user has the ability to assign an external pin to an interrupt, so the MCU can be alerted whenever an interrupt event occurs. The interrupt is generated whenever the threshold value is exceeded for a programmed number of times (interrupt persistence). This is useful to prevent false and erratic interrupt reporting.

The power consumption is mainly affected by the integration time. It is a mean value of a programmable number of consecutive measurements, which is performed in order to reduce the noise, and improve the sensitivity, resolution, etc. However, it has an adverse impact on the overall power consumption, as the time frame during which the device is active, is extended: more measurements, longer activity. The internal state machine puts the ADPS9930 in a standby mode between readings, thus reducing the overall power consumption.

Proximity 7 click uses an I2C interface to communicate with the host MCU. It is equipped with a SMD jumper labeled as VCC SEL. This jumper is used to select the power supply for the pull-up resistors on the I2C bus, allowing both 3.3V and 5V MCUs to be interfaced with this Click board™.

## SPECIFICATIONS

|                     |  |
|---------------------|--|
| <b>Type</b>         | Proximity  |
| <b>Applications</b> | It is a perfect choice for a range of applications that rely on the proximity and ambient light sensing (ALS), including PC and laptop monitors, POS displays, embedded displays, proximity-activated short-range security, etc. |

|                         |  |
|-------------------------|--|
| <b>On-board modules</b> | ADPS9930, a digital ambient and proximity sensor, by Broadcom (Avago Technologies).  |
| <b>Key Features</b>     | High dynamic range allows using the sensor in both very bright and very dim light conditions. Extensive interrupt engine allows simplified firmware to be written. Low power consumption that can be further tuned according to needs. Proprietary design eliminates the need for calibration. |
| <b>Interface</b>        | I2C  |
| <b>Input Voltage</b>    | 3.3V,5V  |

## PINOUT DIAGRAM

This table shows how the pinout on **Proximity 7 click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

| Notes        | Pin  |  |      |     |    | Pin | Notes            |
|--------------|------|--|------|-----|----|-----|------------------|
|              | NC   | 1  | AN   | PWM | 16 | NC  |                  |
|              | NC   | 2  | RST  | INT | 15 | INT | Interrupt output |
|              | NC   | 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 | 3.3V | 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  |
| VCC SEL | VCC SEL | Left    | Logic voltage level selection: left position 3V3, right position 5V |

## SOFTWARE SUPPORT

We provide a library for the **Proximity 7 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

Library contains functions for setting and getting register content Library contains functions for setting proximity and als integraton times as well as wait time Library contains functions for setting proximity and als channel 0 low and high thresholds Library contains functions for setting proximity and asl interrupt persistances Library contains functions for setting proximity pulse count and proximity offset Library contains function for setting constants for Lux calculation Library contains functions for getting Lux level and Int pin status Library contains functions for getting Als data from channels 0 and 1 Library contains function for getting proximity data.

Key functions:

- `void proximity7_setRegister( uint8_t *writeBuffer_, uint8_t nRegisters_ )` - sets register(s) content.
- `float proximity7_getLuxLevel( void )` - calculates LUX level based on Ch0 and Ch1 data and constants set by `setConstants()`; - function.
- `void proximity7_setConstants( float glassAttenuation, float constantB, float constantC, float constantD, float deviceFactor )` - sets constants for LUX calculation.

### Examples description

The application is composed of the three sections :

- System Initialization - Initializes I2C and LOG and sets INT pin as INPUT.
- Application Initialization - Initializes I2C driver and writes basic settings to device registers.
- Application Task - Logs lux level and proximity data.

Note:

- When setting LED drive strength please note that if "proximity drive level - PDL" bit in "configuration register" is set to 1, LED drive current values are reduced by 9.
- When setting wait time note that if "wait long - WLONG" bit is set to 1, time is 12x longer. Therefore if WLONG == 1 set time between 33ms and 8386.56ms.
- When setting ALS gain note that if "ALS gain level - AGL" bit is set to 1, ALS gains are scaled by 0.16, otherwise, they are scaled by 1.

```

void applicationTask( )
{
    proximity7_getRegister( &readBuffer[0], _PROXIMITY7_STATUS, _PROXIMITY7_REPEATED_BYTE, 1 )
;

    alsValid = readBuffer[0] & _PROXIMITY7_ALS_VALID_MASK;
    proximityValid = readBuffer[0] & _PROXIMITY7_PROXIMITY_VALID_MASK;

    if (alsValid != 0 && proximityValid != 0)
    {
        mikrobus_logWrite( " ", _LOG_LINE );

        luxLevel = proximity7_getLuxLevel( );
        FloatToStr( luxLevel, text );
        mikrobus_logWrite( "> > > Lux level   : ", _LOG_TEXT );
        mikrobus_logWrite( text, _LOG_TEXT );
        mikrobus_logWrite( " lx", _LOG_LINE );

        proximity = proximity7_getProximityData( );
        FloatToStr( proximity, text );
        mikrobus_logWrite( "> > > Proximity   : ", _LOG_TEXT );
        mikrobus_logWrite( text, _LOG_TEXT );

        writeBuffer[0] = _PROXIMITY7_SPECIAL_FUNCTION | _PROXIMITY7_PROXIMITY_AND_ALS_INT_CLEAR;
        proximity7_setRegister( &writeBuffer[0], 1 );
    }

    Delay_ms(300);
}

```

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

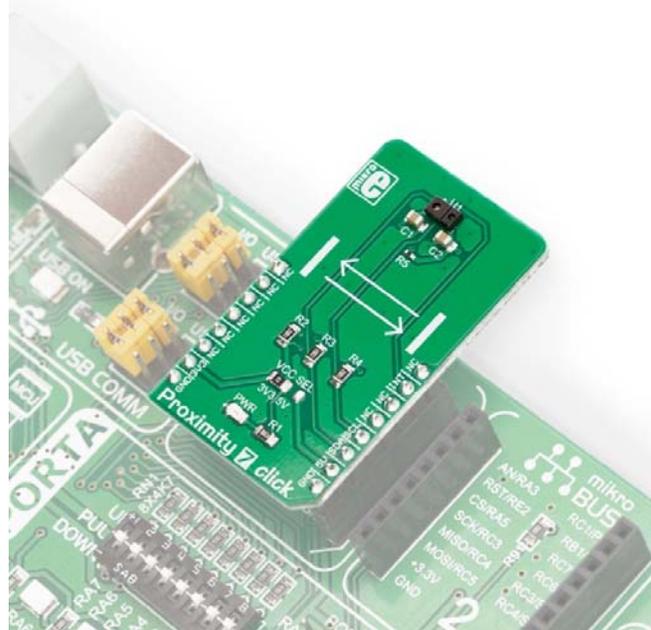
### **Additional notes and informations**

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/proximity-7-click//4-9-19>