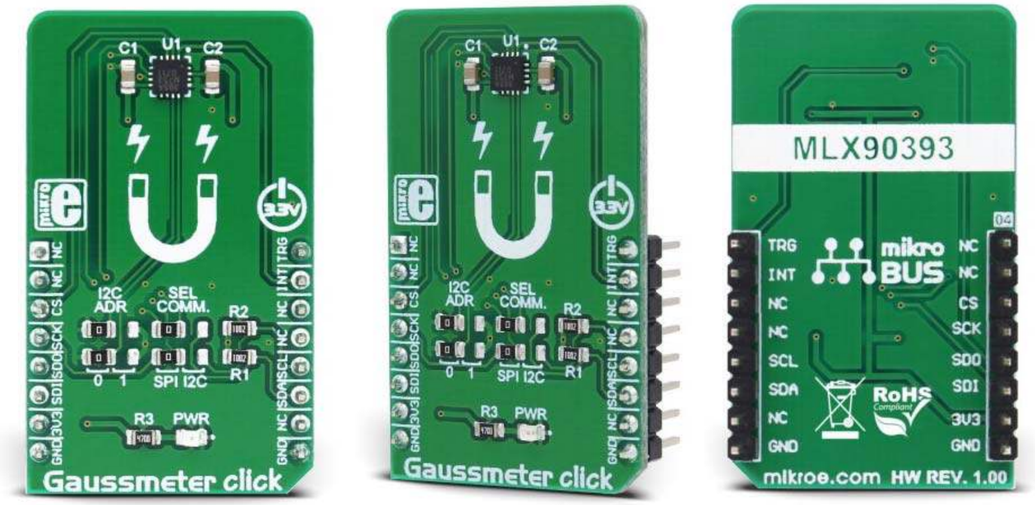
MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS



Gaussmeter click

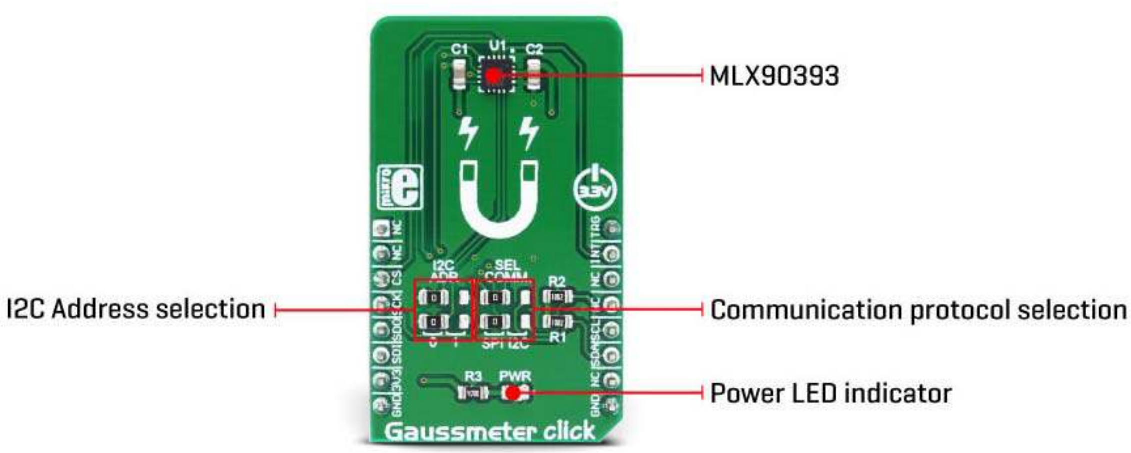
PID: MIKROE-3099
Weight: 23 g

Gaussmeter click is a device that is used for measuring the magnetic field in X, Y and Z axes. This Click board™ features the MLX90393, a micropower magnetometer based on the proprietary Triaxis® technology, from Melexis. It can detect the magnetic field down to 0.8 µT (8 mG) thanks to this technology. The integrated temperature sensor provides data for the thermal compensation, while the on-chip non-volatile memory can store the various working and compensation parameters, so they can be reused after the Power ON reset cycle (POR). The Click board™ supports both I2C and SPI communication protocols, allowing it to be interfaced with the wide range of different MCUs.

A range of on-chip features including low power consumption, programmable interrupt engine, 1KB of volatile + 1 KB of non-volatile memory, integrated thermal sensor, onboard digital filtering, high resolution sampling, and more, make this sensor a perfect solution for various low power applications used to detect a magnetic and electromagnetic field (EMF) presence along all three axes. It can be used to detect a rotary or linear movement of the magnetic elements or to measures EMF from mains electrical devices and other (very low frequency) EMF sources, such as from mains electricity, power lines, high voltage lines, transformers, and so on.

How does it work

Gaussmeter click employs the MLX90393, a micropower magnetometer based on the proprietary Triaxis® technology, from Melexis. This IC is based on the Hall effect principle, which allows it to detect very small fluctuations in the magnetic field. The Hall sensor plates, featuring the patented IMC technology, are located in the center of the die, which is located in the center of the package.



The measurement current, generated as a result of the Hall effect, is passed through the transimpedance amplifier (TIA) and sampled by the 19-bit A/D converter (ADC). The output is the truncated to 16-bits, by applying the bit-shifting operation, programmed by the user (RES_XYZ bits). This allows the range to be dynamically set, according to measurement conditions, leaving unused MSBs or LSBs out. Additionally, it is possible to set the TIA gain level in the range from 0 to 7 to best match the field strength. The value of the RES bits and the gain level both affect the sensitivity of the sensor. The MLX90393 datasheet contains a table with the RES and GAIN, and corresponding µT/LSB values.

When required, it is possible to set the oversampling rate of the ADC decimation filter. This will provide less noise and more consistent readings. However, oversampling affects the data acquisition time, as the sampling process has to be repeated a number of times, depending on the oversampling rate. If the fast response is required, the oversampling and digital filtering functions should be turned off.

The measurement is affected by the temperature. Therefore, the MLX90393 is also equipped with the temperature sensor, used to provide the required measurements. The thermal sensitivity drift compensation can be enabled by the appropriate bit in the configuration register. Two sensitivity drift compensation factors can be used, one for temperatures greater than the reference and other for the temperatures lesser than the reference value.

Wakeup on Change (WOC) mode is used to alert the MCU via the interrupt pin when certain conditions are met. It can be configured to trigger an interrupt on the INT pin if the difference between the reference measurement value and the current measurement value exceed a threshold defined by the user. The reference measurement value can be set in three different ways:

- The first measurement in WOC mode is set as the measurement reference value, which is then used for comparison.
- The reference value is always the previous measurement. Every new measurement sets the previous one as the reference value.
- Both the thresholds and the reference value are programmed by the user.

The interrupt is reported on the INT pin of the IC, routed to the INT pin of the mikroBUS™, and/or on the INT/TRIG pin, routed to the mikroBUS™ PWM pin (labeled as TRG), if it is configured that way. The INT/TRIG pin can also be set as the trigger input, by configuring the corresponding bits (TRIG_INT_SEL and EXT_TRIG). These pins are active HIGH.

Besides the WOC mode, there is also Burst and Single Measurement modes. Both of these modes can use the INT pin to signalize that there is a conversion data ready to be read. Once the MCU reads the data, the INT/Data Ready event will be cleared. The Burst mode provides data in programmed intervals, while Single Measurement mode will provide one reading when commanded, signal it via the INT pin and revert to IDLE mode, consuming less power.

The MLX90393 sensor contains 1KB of volatile (RAM) memory. This memory is used to store config parameters and register values, but also there are some free locations for storing user information, e.g. compensation values and similar. Besides 1KB of volatile memory, there is also 1KB of non-volatile memory. During the POR, the complete content of the NV memory is automatically copied to the RAM restoring the saved working parameters that way. There are also commands available for the user, in order to store to or restore RAM data from the non-volatile (NV) memory. However, it is not recommended to write to the NV memory area too often, as this type of memory has an inherently limited life cycle.

This Click board™ allows both SPI and I2C communication protocols. To select a proper protocol, the SMD jumpers labeled as SEL COM should be moved to the appropriate position (I2C or SPI). Please note that both jumpers need to be at the same setting (both as SPI or both as I2C).

The I2C address of the Click board™ is selectable by the onboard SMD jumpers, labeled as I2C ADDR. These two jumpers directly set the values of the LSB address of the IC. The 7-bit address of the device is 00011XXZ, where XX are the values set by these jumpers, while Z is the R/W bit.

Specifications

Type	Hall effect
Applications	It can be used to detect a rotary or linear movement of the magnetic elements, or to measures EMF from mains electrical devices and other EMF sources, such as from mains electricity, power lines, high voltage lines, transformers, and so on
On-board modules	MLX90393, a micropower magnetometer based on the proprietary Triaxis® technology, from Melexis
Key Features	A proprietary Triaxis® technology allows sensing of wide range of magnetic field values, high accuracy and low noise, thermal sensing element included, allows both I2C and SPI communication protocols, NV memory for storing config data after POR
Interface	I2C,SPI
Input Voltage	3.3V
Click board size	M (42.9 x 25.4 mm)

Pinout diagram

This table shows how the pinout on **Gaussmeter 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	**TRG**	Interrupt/Trigger
	NC	2	RST	INT	15	**INT**	Interrupt

SPI Chip Select	**CS**	3	CS	RX	14	NC	
SPI Clock	**SCK**	4	SCK	TX	13	NC	
SPI Data OUT	**SDO**	5	MISO	SCL	12	**SCL**	I2C Clock
SPI Data IN	**SDI**	6	MOSI	SDA	11	**SDA**	I2C Data
Power supply	**3.3V**	7	3.3V	5V	10	NC	
Ground	**GND**	8	GND	GND	9	**GND**	Ground

Onboard jumpers and settings

Label	Name	Default	Description
LD1	PWR	-	Power LED indicator
JP_A0, JP_A1	I2C ADR	Left	I2C address bit selection: left position 0, right position 1
JP_1, JP_2	SEL COM	Left	Communication protocol selection: left position SPI, right position I2C

Software Support

We provide a library for the Gaussmeter 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 initializes and defines I2C and SPI driver and has the ability to write to registers and read from registers by both driver communication. Library also performs magnetic field measurement via x, y and/or z-axis and converts results to µT. The library can measure 16-bit temperature value. For more details check the documentation.

Key functions

void gaussmeter_enableMeasure(uint8_t zAxis, uint8_t yAxis, uint8_t xAxis, uint8_t temp) - Function determines which axis be included in the measurement.

uint8_t gaussmeter_sendCommand(uint8_t command_) - Function performs desired command.

uint8_t gaussmeter_readReg(uint8_t register_address, uint16_t *dataOut) - Function reads 16-bit data from register.

uint8_t gaussmeter_writeReg(uint8_t register_address, uint16_t transfer_data) - Function writes 16-bit data in register.

uint8_t gaussmeter_getData(float *dataOut) - Function gets selected measured data.

Examples Description

The application is composed of three sections :

System Initialization - Initializes peripherals and pins.
Application Initialization - Initializes I2C or SPI driver and performs click configuration.
The device is configured to work in Wake Up On Change mode.
Application Task - (code snippet) - Gets x, y and z-axis measured data (without temperature) in µT.
Logs the desired results on USBUART and repeats measurements every 400ms.

```
void applicationTask()
{
errorBit = gaussmeter_getData( &bufferData[ 0 ] );

if (!errorBit)
{
axisCheck = 1;
indx = 0;

for (counter = 0; counter < 4; counter++)
{
switch (commandByteLow & axisCheck)
{
case 1 :
{
mikrobus_logWrite( "16-bit temperature value is: ", _LOG_TEXT );
FloatToStr( bufferData[ indx ], text );
floatConv();
mikrobus_logWrite( text, _LOG_LINE );
indx++;
break;
}
case 2 :
{
mikrobus_logWrite( "X axis is: ", _LOG_TEXT );
FloatToStr( bufferData[ indx ], text );
floatConv();
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( " microT", _LOG_LINE );
indx++;
break;
}
case 4 :
{
mikrobus_logWrite( "Y axis is: ", _LOG_TEXT );
FloatToStr( bufferData[ indx ], text );
floatConv();
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( " microT", _LOG_LINE );
indx++;
break;
}
case 8 :
{
mikrobus_logWrite( "Z axis is: ", _LOG_TEXT );
FloatToStr( bufferData[ indx ], text );
floatConv();
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( " microT", _LOG_LINE );
indx++;
break;
}
default :
{
break;
}
}

axisCheck <<= 1;
}
mikrobus_logWrite( "", _LOG_LINE );
Delay_ms( 400 );
}
}
```

Additional Functions :

- void floatConv() - Makes the float values be rounded to two decimal places

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

Other mikroE Libraries used in the example:

- Conversions

- I2C
- 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.