

Accel 8 Click

PID: MIKROE-3341 Weight: 23 g

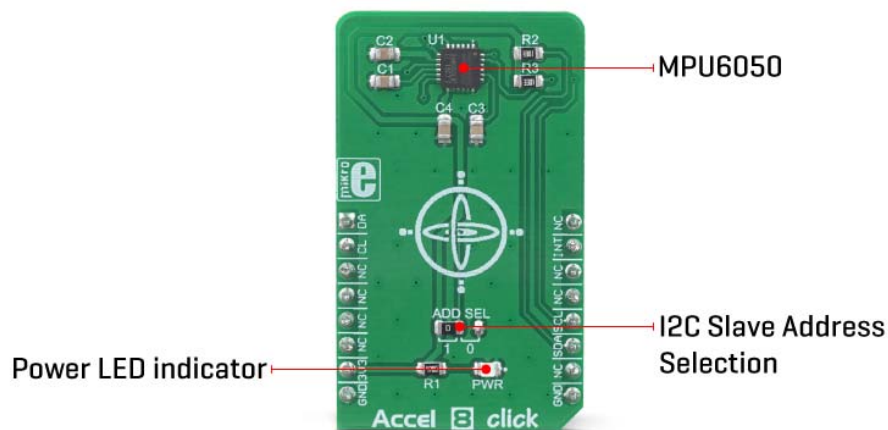
Accel 8 click is an advanced 6-axis motion tracking Click board™, which utilizes the MPU6050, a very popular motion sensor IC, equipped with a 3-axis gyroscope and 3-axis accelerometer. It also features Digital Motion Processor™, a very powerful processing engine, which reduces the firmware complexity and the processing load of the host MCU. The output of each axis is digitized by a separate 16-bit ADC and processed by the DMP engine, offering motion data over the I2C Interface. Packed with a set of very powerful options, this Click board™ represents an ideal solution for the development of MotionInterface™ based applications.

Accel 8 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.

MPU6050 is produced utilizing the innovative Nasiri Fabrication platform, specialized in MEMS production and its integration with accompanying logic sections, on a wafer level. This allows very high integration and very small dimensions, at an affordable cost. Besides the size, features such as low power consumption, high precision of motion detection, high shock tolerance, and programming capabilities, make it a MotionInterface™ enabled device, which can be used for development of different types of motion detection based applications, including gesture command, augmented reality, vehicle navigation, image stabilization in photography, and other similar applications.

How does it work?

Accel 8 click is based around the **MPU6050**, a motion tracking sensor IC, from **TDK Invensense**. It is an advanced, integrated microelectromechanical gyroscope and accelerometer sensor (MEMS), combined with the powerful data processing engine. There is a respective accelerometer or gyroscope MEMS on each axis. The output of each MEMS is processed and digitized by a separate sigma-delta 16-bit A/D converter (ADC). The outputs can be processed by a programmable low-pass filter, while their sample rate can be selected by the user.



Three-axis gyroscope MEMS can be programmed to measure the rotation about each axis, in four different ranges of rotational speed (degrees per angle, DPS): ± 250 , ± 500 , ± 1000 , and ± 2000 . Three-axis accelerometer MEMS can be programmed to measure the acceleration along each axis, in four different acceleration ranges: $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$. The user can select an optimal range for both properties, depending on the application requirements.

The embedded Digital Motion Processor™ can process complex 6-axis motion detection and gesture recognition algorithms without taking up processing cycles of the host microcontroller (MCU), making it perfectly suited for different kinds of low-power applications. The DMP engine offers a high output data rate (ODR), improving measurement accuracy. Due to DMP hardware-accelerated motion detection algorithms, MPU60x0 are very popular motion tracking ICs and there are many different designs using both MPU6050 and MPU6000 (e.g. [MPU IMU click](#)). Unlike the MPU6000, the MPU6050 uses the I2C communication interface.

MPU6050 incorporates a powerful programmable interrupt engine. The interrupt engine can generate a signal on the interrupt pin for several interrupt sources, including FIFO Buffer overflow, Data Ready, I2C Master Error, and I2C Slave Error. The interrupt is routed to the INT pin of the mikroBUS™.

A FIFO buffer helps to further reduce the processing load, offering temporary storage for the output data. The MPU6050 features a FIFO buffer with the capacity of 1024 bytes. The user can select which data will be stored in the FIFO buffer: gyro data, accel data, temperature readings, and auxiliary sensor readings. Once the FIFO buffer is full, it will start discarding the oldest data, allowing new data to be written. The FIFO buffer overflow condition can be used to trigger an interrupt, alerting the host MCU about its status.

Another powerful feature of the MPU6050 is its ability to be interfaced with an additional sensor, such as the 3-axis compass. By utilizing the MotionFusion™ firmware on the chip, along with the run-time calibration, it offers the complete 9-axis motion sensing solution. This sensor can be connected to the auxiliary I2C pins, routed to the mikroBUS™. Aux I2C clock pin is routed to the RST pin of the mikroBUS™ and it is labeled as CL, while the Aux I2C data pin is routed to the AN pin of the mikroBUS™, labeled as DA. Besides the compass sensor, other general-purpose sensors that use I2C interface can be connected, too. An interrupt can be generated if there is an error in the communication between the MPU6050 and the auxiliary sensor.

The I2C address of the MPU6050 can be selected by the ADD SEL jumper. This SMD jumper is used to select the least significant bit (LSB) of the 7-bit I2C address. The value of the LSB is decided by the position of this jumper.

Accel 8 click uses the I2C communication interface. It has pull-up resistors connected to the mikroBUS™ 3.3V rail. Proper conversion of logic voltage levels should be applied before the Click board™ is used with MCUs operated with 5V.

Specifications

Type	Motion
Applications	It is a perfect solution for development of different types of motion detection-based applications, including gesture command, augmented reality, vehicle navigation, image stabilization in photography, and other similar applications.
On-board modules	MPU6050, a six-axis (gyro + accelerometer) MEMS MotionTracking™ IC, from TDK Invensense.
Key Features	Digital Motion Processor™ engine with motion processing and gesture recognition algorithms, 6 independent MEMS, one on each axis, sampled by a separate 16-bit ADC, AUX I2C interface for additional sensor supported by the MotionFusion™ engine, output data filtering, selectable accel and gyro ranges...
Interface	I2C
Input Voltage	3.3V
Click board size	M (42.9 x 25.4 mm)

Pinout diagram

This table shows how the pinout on **Accel 8 Click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin					Pin	Notes
Aux I2C Data	DA	1	AN	PWM	16	NC	

Aux I2C Clock	CL	2	RST	INT	15	INT	Interrupt
	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	+3V3	7	3.3V	5V	10	NC	
Ground	GND	8	GND	GND	9	GND	Ground

Onboard settings and indicators

Label	Name	Default	Description
LD1	PWR	-	Power LED indicator
JP1	ADD SEL	Left	I2C slave address LSB selection: left position 1, right position 0

Software support

We provide a library for the **Accel 8 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 register and reads data form register. The library

includes function for read Accel X/Y/Z axis data, Gyro X/Y/Z axis data and device Temperature data. The user also has the function for configuration Accel and Gyro and function for read interrupt state.

Key functions:

- `void accel8_getAccelAxis(int16_t *x_axis, int16_t *y_axis, int16_t *z_axis)` - Functions for read Accel axis data.
- `void accel8_getGyroAxis(int16_t *x_axis, int16_t *y_axis, int16_t *z_axis)` - Functions for read Gyro axis data.
- `float accel8_getTemperature()` - Functions for read Temperature data in C.

Examples description

The application is composed of the three sections :

- System Initialization - Initializes I2C module and sets INT pin as INPUT.
- Application Initialization - Initialization driver init, reset chip and start configuration chip for measurement.
- Application Task - (code snippet) - Reads Accel X/Y/Z axis, Gyro X/Y/Z axis and device Temperature. All data logs on the USBUART every 2 sec.

```
void applicationTask()
{
    accel8_getAccelAxis(&X_accelAxis, &Y_accelAxis, &Z_accelAxis);
    accel8_getGyroAxis(&X_gyroAxis, &Y_gyroAxis, &Z_gyroAxis);
    Temperature = accel8_getTemperature();

    // LOGS DATA
    mikrobus_logWrite("_____ Accel 8 click _____", _LOG_LINE);
    mikrobus_logWrite("| Data | X axis | Y axis | Z axis | Range |", _LOG_LINE);
    mikrobus_logWrite("|_____||_____||_____||_____||_____||", _LOG_LINE);
    mikrobus_logWrite("| Accel |", _LOG_TEXT);
    IntToStr(X_accelAxis, demoText);
    mikrobus_logWrite(demoText, _LOG_TEXT);
    mikrobus_logWrite(" |", _LOG_TEXT);
    IntToStr(Y_accelAxis, demoText);
    mikrobus_logWrite(demoText, _LOG_TEXT);
    mikrobus_logWrite(" |", _LOG_TEXT);
    IntToStr(Z_accelAxis, demoText);
    mikrobus_logWrite(demoText, _LOG_TEXT);
    mikrobus_logWrite(" |", _LOG_TEXT);
```

```

IntToStr(accelRange, demoText);
mikrobus_logWrite(demoText, _LOG_TEXT);
mikrobus_logWrite(" g |", _LOG_LINE);
mikrobus_logWrite("|_____|_____|_____|_____|_____|", _LOG_LINE);
mikrobus_logWrite("| Gyro |", _LOG_TEXT);
IntToStr(X_gyroAxis, demoText);
mikrobus_logWrite(demoText, _LOG_TEXT);
mikrobus_logWrite(" |", _LOG_TEXT);
IntToStr(Y_gyroAxis, demoText);
mikrobus_logWrite(demoText, _LOG_TEXT);
mikrobus_logWrite(" |", _LOG_TEXT);
IntToStr(Z_gyroAxis, demoText);
mikrobus_logWrite(demoText, _LOG_TEXT);
mikrobus_logWrite(" |", _LOG_TEXT);
IntToStr(gyroRange, demoText);
mikrobus_logWrite(demoText, _LOG_TEXT);
mikrobus_logWrite("dps|", _LOG_LINE);
mikrobus_logWrite("|_____|_____|_____|_____|_____|", _LOG_LINE);
FloatToStr(Temperature, demoText);
demoText[ 5 ] = 0;
mikrobus_logWrite("| Temp |", _LOG_TEXT);
mikrobus_logWrite(demoText, _LOG_TEXT);
mikrobus_logWrite(" C |", _LOG_LINE);
mikrobus_logWrite("|_____|_____|", _LOG_LINE);

mikrobus_logWrite(" ", _LOG_LINE);
Delay_ms( 2000 );
}

```

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 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.

