

# PWR Meter click

PID: MIKROE-3169

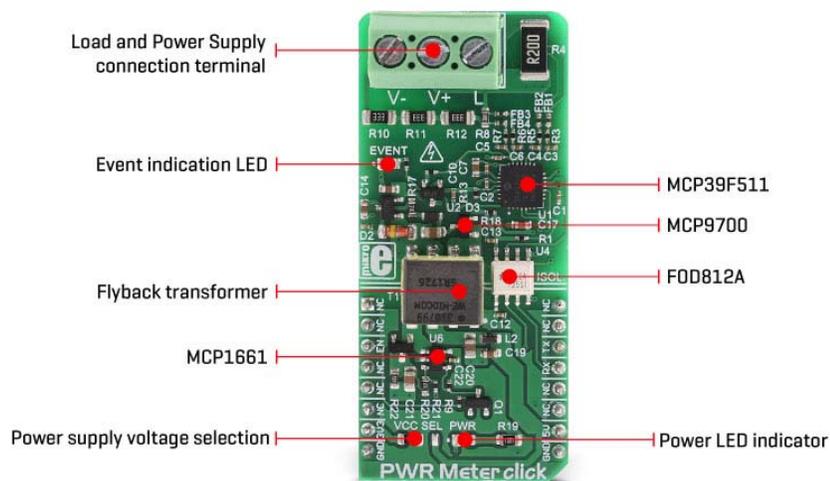
Weight: 31 g

**PWR Meter click** is a power measurement Click board™, capable of measuring voltage and current through the load, connected to either AC or DC power source. PWR Meter click uses the MCP39F511A, a very sophisticated monitoring IC from Microchip, with 16-bit processing core. It is used to calculate all the measurement parameters, returning values of multiple power parameters directly, over the UART interface, reducing the processing load on the host MCU. These parameters include active, reactive, and apparent power, current and voltage RMS, line frequency, and power factor.

The measurement circuitry is completely isolated from the mikroBUS™, thus preventing any power surges from the measured circuit, which might cause damage to the low voltage components. Featuring very accurate power monitoring IC, with the accuracy of 0.1% across the 4000:1 dynamic range, this Click board™ is a perfect solution for various advanced measurement and monitoring purposes. Having in mind its processing core, and simplified command protocol over the UART interface, it can be driven even by slower, 8-bit MCUs, providing all the complex measurements which would have a large performance impact, otherwise. It can be used for development of intelligent power distribution systems, real time power characteristics measurement for AC or DC power supply units, power monitoring and automation, and similar.

## How does it work?

PWR Meter click is based around the MCP39F511A, a power monitoring IC with on-chip 16-bit data processing and energy accumulation features, from Microchip. This IC is an advanced power monitoring IC, capable of calculating power characteristics, based on the measurements taken from the connected load and power supply. The IC is able to calculate active, reactive, and apparent power, current and voltage RMS, line frequency, and power factor. In addition, it features several useful functions, such as the programmable event reporting, with the onboard LED labeled as the EVENT.



PWR Meter click uses the UART interface for the communication with the host MCU. The communication is based on the SSI (Simple Sensor Interface) protocol and it is fairly simple to use. This protocol is widely used for a point-to-point communication between the host MCU and other sensor devices, such as the MCP39F511A. The default rate of the protocol is 9600 bps, but it can be configured by the user. More information about the communication protocol itself and the commands that can be used, can be found in the MCP39F511A datasheet. However, PWR Meter click comes with a library which is compatible with all the MikroElektronika compilers. It contains functions which make working with the PWR Meter click even simpler, saving a lot of development time.

The MCP39F511A incorporates two internal 24-bit Analog to Digital Converters (ADC), used to sample the voltage values on their differential inputs. One channel is used to measure voltage drop across the shunt resistor with the value of  $0.2\Omega$ , while the second channel samples the voltage across the voltage divider on the input terminal. Voltage drop across the shunt resistor allows calculating the current through the connected load, while the voltage divider allows voltage measurement across the connected load, scaling it down so that it can be measured. A third, 10-bit ADC is used to measure the ambient temperature, needed for compensation. It is connected to an output of the MCP9700A, a low power linear active integrated thermistor, from Microchip.

The load should be connected to the 3-pole input terminal, between the input labeled with the L letter and the input labeled as the V-. The power supply should be also connected to the Click board™. Its hot (positive) end should be connected to the V+ labeled input of the 3-pole terminal, while the negative end should be connected to the input labeled as V-. The power supply should not exceed 50V. To better understand connection scheme, please take a look at the picture below.

The Click board™ features a complete galvanic isolation of the measured circuit. The power for the high voltage section is provided by the MCP1661, an efficient integrated boost (step-up) DC/DC converter, from Microchip. The integrated boost converter is built using the flyback topology, allowing a complete galvanic isolation between the primary and secondary side, since it uses a transformer instead of a coil. The input voltage of the DC-DC converter is selected by an SMD jumper, labeled as the VCC SEL. The boosted voltage on the secondary of the transformer is further conditioned by the MCP17545, a LDO regulator from Microchip, and it is fixed to 3V. The output of the LDO is now galvanically isolated from the rest of the circuit, and it is used to supply the MCP39F511A monitoring IC and the additional thermal sensor with power.

Enabling or disabling the MCP1661 DC/DC converter controls the operation of the PWR Meter click itself. The CS pin of the mikroBUS™ is routed to the EN pin of the MCP1661 converter, allowing the user to turn off the power for the monitoring IC. Setting the CS pin to a HIGH logic level will disable the converter, allowing current to sink through the transistor, thus setting the EN pin to a LOW logic level. Otherwise, the EN pin is pulled up to a HIGH level with the resistor, and the converter is enabled by default (when the CS pin of the mikroBUS™ is left floating, or driven to a LOW logic level)

Galvanic isolation of the MCP39F511A data lines is done by using a bi-directional logic gate optocoupler, labeled as FOD812A from ON Semiconductors. UART RX and TX lines from the MCP39F511A IC run through the integrated optocouplers and are also completely isolated from the low voltage circuitry. The previously mentioned VCC SEL jumper also selects the voltage for the optocoupler, allowing both 3.3V and 5V tolerant MCUs to be interfaced with the Click board™.

## Specifications

---

<b>Applications</b>	PWR Meter click can be used for digital power monitoring and power metering in portable and automotive devices, embedded electronic applications, real time power characteristics measurement for AC or DC power supply units, smart power distribution systems and similar applications.
<b>On-board modules</b>	MCP39F511A, a power monitoring IC with on-chip 16-bit data processing and energy accumulation features; MCP1661, an efficient integrated boost (step-up) DC/DC converter, both from Microchip
<b>Key Features</b>	High resolution (24bit) dual sigma-delta A/D converter with differential inputs, additional 10bit ADC for temperature measurement, 16bit data processing offering real time readings for a wide range of derived power characteristics, low series resistance (0.2Ω), simple UART based command protocol, and more
<b>Interface</b>	UART
<b>Input Voltage</b>	3.3V or 5V
<b>Compatibility</b>	mikroBUS
<b>Click board size</b>	L (57.15 x 25.4 mm)

## Pinout diagram

---

This table shows how the pinout on **PWR Meter 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	NC	
Power Enable	<b>EN</b>	3	CS	RX	14	<b>TX</b>	UART TX
	NC	4	SCK	TX	13	<b>RX</b>	UART RX
	NC	5	MISO	SCL	12	NC	
	NC	6	MOSI	SDA	11	NC	
Power supply	<b>3.3V</b>	7	3.3V	5V	10	<b>5V</b>	Power supply
Ground	<b>GND</b>	8	GND	GND	9	<b>GND</b>	Ground

## Onboard settings and indicators

Label	Name	Default	Description
LD1	PWR	-	Power LED indicator
LD2	EVENT	-	Programmable event LED indicator
JP1	VCC SEL	Left	Power supply source selection: left position 3.3V, right position 5V
CN1	V-, V+, L	-	Load and Power Supply connection terminal

# Software support

---

We provide a library for the PWR Meter 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 performs the communication with the device via uart interface by sending commands and checking response from the device to the host. Commands perform writing to the registers, reading from the registers (data can be 8bit, 16bit, or 32bit), calibration, writing to the EEPROM and reading from the EEPROM. By using functions for reading user can read measurements that include voltage RMS, current RMS, power data (active, reactive, apparent) and many other things. For more details check documentation.

Key functions:

- `T_PWRMETER_RETVAL pwrmeter_readRegBytes( uint16_t regAddr, uint8_t nBytes, uint8_t *dataOut )` - Function reads data bytes from registers.
- `T_PWRMETER_RETVAL pwrmeter_writeRegWORD( uint16_t registerAddr, uint16_t dataIn )` - Function writes 16-bit data to the register.
- `T_PWRMETER_RETVAL pwrmeter_sendCommand( uint8_t commandByte )` - Function sends and performs a determined command.

## Example description

The application is composed of three sections :

- System Initialization - Initializes peripherals and pins.
- Application Initialization - Initializes UART interface, puts output of regulator in active state and configures gain channel and uart baud rate.
- Application Task - (code snippet) - Reads voltage, current and power measurements from data registers, converts this values to determined units and all results on uart terminal. Repeats operation every second.

```
void applicationTask()
{
    responseByte = pwrmeter_readRegWORD( _PWRMETER_VOLT_RMS_REG, &voltageRMS );
    checkResponse();
    responseByte = pwrmeter_readRegDWORD( _PWRMETER_CURR_RMS_REG, &trms );
    checkResponse();
    responseByte = pwrmeter_readRegDWORD( _PWRMETER_ACTIVE_PWR_REG, &activePower );
    checkResponse();
    responseByte = pwrmeter_readRegDWORD( _PWRMETER_REACTIVE_PWR_REG, &reactivePower );
    checkResponse();
}
```

```

responseByte = pwrmeter_readRegDWORD( _PWRMETER_APPARENT_PWR_REG, &apparentPower );
checkResponse();
responseByte = pwrmeter_readRegSigned( _PWRMETER_PWR_FACTOR_REG, _PWRMETER_16BIT_DATA, &power
Factor );
checkResponse();

measData[ 0 ] = (float)voltageRMS / 100;
measData[ 1 ] = (float)currentRMS / 1000;
measData[ 2 ] = (float)activePower / 100000;
measData[ 3 ] = (float)reactivePower / 100000;
measData[ 4 ] = (float)apparentPower / 100000;
measData[ 5 ] = (float)powerFactor / 32767;

responseByte = pwrmeter_getStatus( &statusByte );
checkResponse();

if ((statusByte & _PWRMETER_DCMODE_MASK) != 0)
{
mikrobus_logWrite( "DC mode", _LOG_LINE );
}
else
{
mikrobus_logWrite( "AC mode", _LOG_LINE );
}

FloatToStr( measData[ 0 ], text );
floatConv();
mikrobus_logWrite( "RMS voltage: ", _LOG_TEXT );
if (((statusByte & _PWRMETER_DCMODE_MASK) != 0) && ((statusByte & _PWRMETER_DCVOLT_SIGN_MASK)
== 0))
{
mikrobus_logWrite( "-", _LOG_TEXT );
}
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( "[V]", _LOG_LINE );

FloatToStr( measData[ 1 ], text );
floatConv();
mikrobus_logWrite( "RMS current: ", _LOG_TEXT );
if (((statusByte & _PWRMETER_DCMODE_MASK) != 0) && ((statusByte & _PWRMETER_DCCURR_SIGN_MASK)
== 0))
{
mikrobus_logWrite( "-", _LOG_TEXT );
}
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( "[mA]", _LOG_LINE );

FloatToStr( measData[ 2 ], text );
floatConv();
mikrobus_logWrite( "Active power: ", _LOG_TEXT );
if ((statusByte & _PWRMETER_PA_SIGN_MASK) == 0)
{
mikrobus_logWrite( "-", _LOG_TEXT );
}
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( "[W]", _LOG_LINE );

FloatToStr( measData[ 3 ], text );
floatConv();
mikrobus_logWrite( "Reactive power: ", _LOG_TEXT );

```

```

if ((statusByte & _PWRMETER_PR_SIGN_MASK) == 0)
{
mikrobus_logWrite( "-", _LOG_TEXT );
}
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( "[VAr]", _LOG_LINE );

FloatToStr( measData[ 4 ], text );
floatConv();
mikrobus_logWrite( "Apparent power: ", _LOG_TEXT );
mikrobus_logWrite( text, _LOG_TEXT );
mikrobus_logWrite( "[VA]", _LOG_LINE );

FloatToStr( measData[ 5 ], text );
floatConv();
mikrobus_logWrite( "Power factor: ", _LOG_TEXT );
mikrobus_logWrite( text, _LOG_LINE );
mikrobus_logWrite( "", _LOG_LINE );

Delay_ms( 1000 );
}

```

#### Additional Functions :

- void checkResponse() - Checks response from the device to the host and logs message if error occurred.
- void floatConv() - Makes the float values be rounded to two decimal places.

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

Other mikroE Libraries used in the example:

- Conversions
- 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.

