

## FEATURES

- Small form factor: 13.22 x 22.27 x 3.8 mm
- 1% error for voltage, current, power, energy measurement.
- Current sensing range up to 40A (RMS).
- Four high-accuracy channels for monitoring up to four loads
- Four 16-bit ADC channels for residual current detection
- $V_{ref}$  output pin to offset external inputs.
- Zero-crossing detection output.
- Operating voltage range 2.7 - 3.6V.
- Current consumption: 25 mA.
- 14 IO pins (with 2 open-drain pins).
- Two serial port interfaces.
- Easy to use interfaces: AT commands and Modbus RTU (optional).
- Control function supports both latching and non-latching relays.
- Frequency measurement.

## APPLICATIONS

- Automotive charging station
- Vehicle charger
- Electric meter
- Smarthome
- Industrial energy monitoring



## OVERVIEW

Tony Meter is a compact solution to control and monitor AC loads. Each module has 4 channels. Roughly speaking, a Tony Meter is equivalent to 4 independent single-phase meters. Each channel can monitor the basic parameters of its load: voltage, current, power, and delivered energy, as shown in Figure 2. Besides, the module provides some functions to control AC loads: zero-crossing, overload protection, no-load protection, residual current protection.

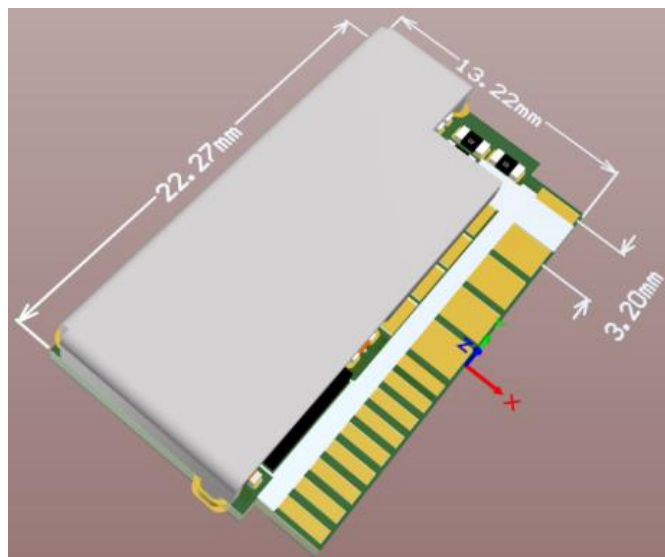
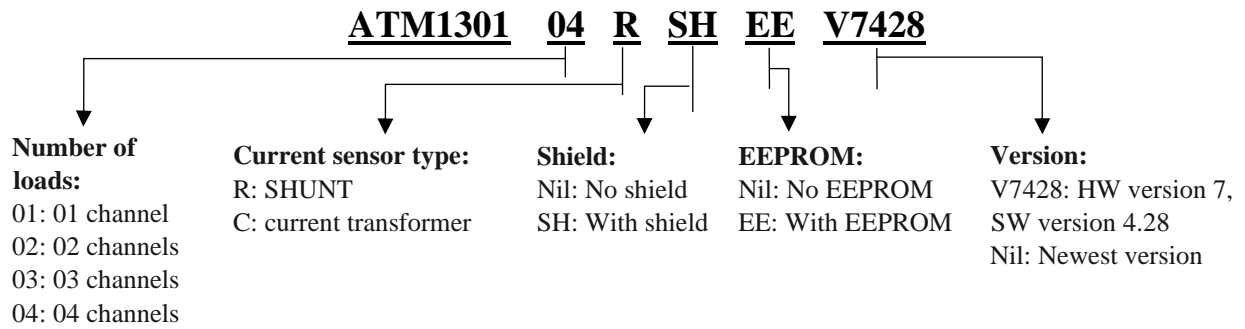


Figure 1: Module overview

## ORDERING INFORMATION:



## BLOCK DIAGRAMS

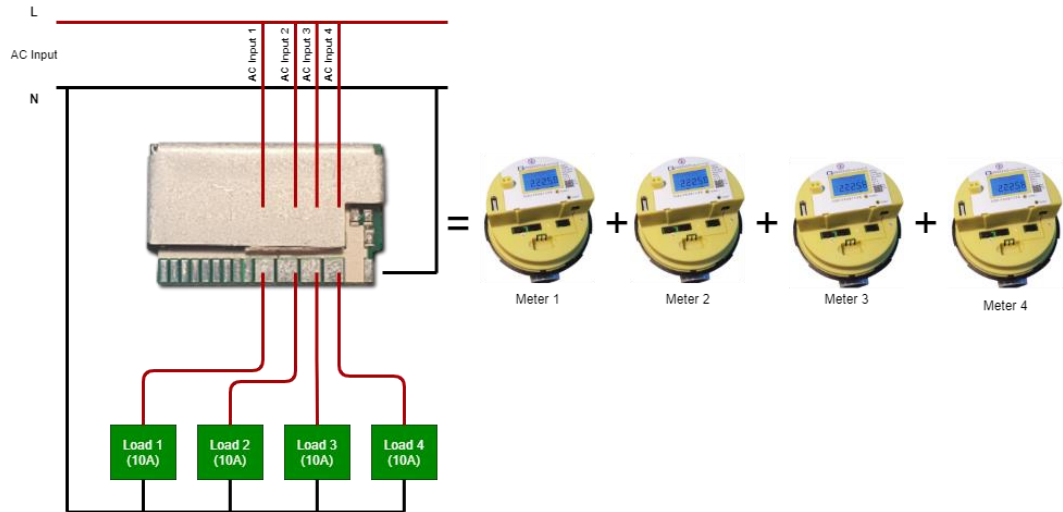


Figure 2: Tony meter measurement function.

The configuration of the module is flexible, with plenty of options. The flexibility helps the module adapt to any design quickly.

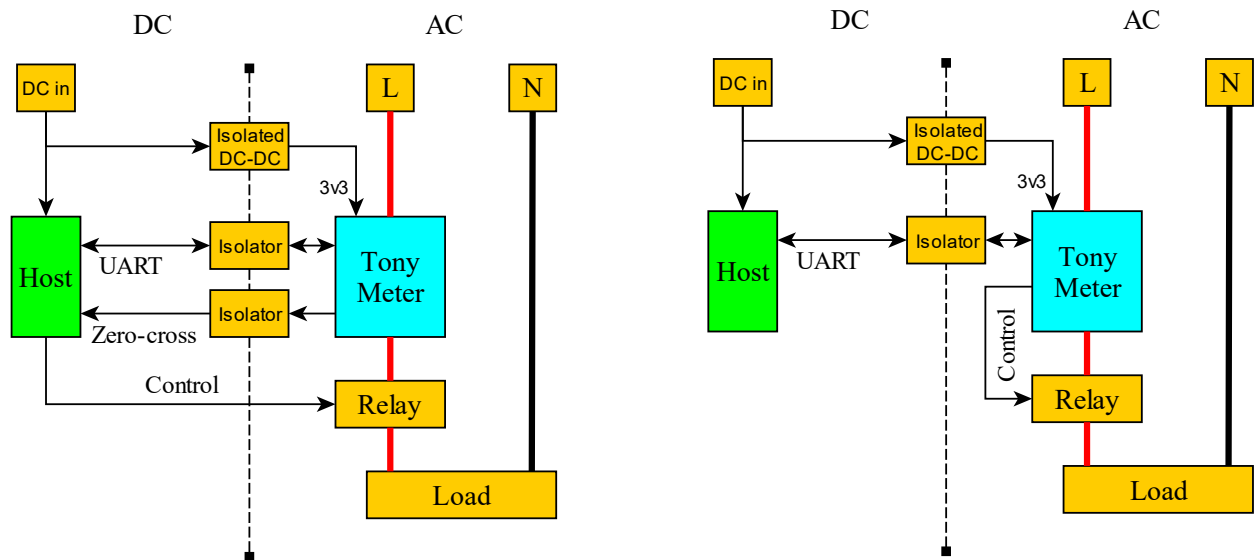


Figure 3: Module connection

The diagram on the right of Figure 3 shows an example of the module connection. A host controller communicates with the module by a *UART bus*. Through this bus, the host can read all parameters of the load by using AT commands or Modbus RTU. On the other hand, the diagram on the left of Figure 3 shows another case of module usage. In which, the host control the relay directly by using the *Zero-cross* signal from the module.

## DIMENSIONS

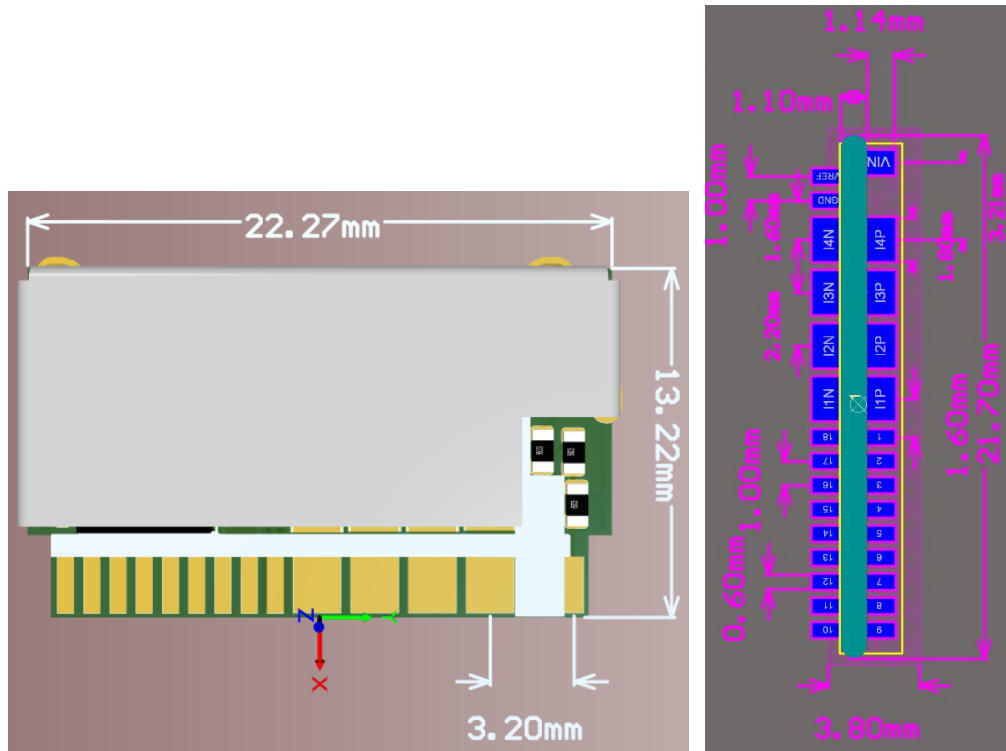


Figure 4: Module dimensions

## PIN DESCRIPTION

1	VREF		
2	GND	VIN	29
3	I4N	I4P	28
4	I3N	I3P	27
5	I2N	I2P	26
6	I1N	I1P	25
7	GND	GND	24
8	VCC	VCC	23
9	IO13	IO0	22
10	IO12	IO1	21
11	IO11	IO2	20
12	IO10	IO3	19
13	IO9	IO4	18
14	IO8	IO5	17
15	IO7	IO6	16

Figure 5: Module pinout

Pin	Name	Description
1	V <sub>REF</sub>	Referent voltage (0.6 V)
2	GND	GND
3	I4N	Channel 4 negative pin
4	I3N	Channel 3 negative pin
5	I2N	Channel 2 negative pin
6	I1N	Channel 1 negative pin
7	GND	GND
8	V <sub>CC</sub>	Isolated 3.3VDC (board's power supply)
9	IO13	GPIO (Open drain)

10	IO12	GPIO
11	IO11	GPIO/ADC channel 0
12	IO10	GPIO/ADC channel 1
13	IO9	GPIO/ADC channel 2
14	IO8	GPIO/ADC channel 3/UART2 TX <sup>[1]</sup>
15	IO7	GPIO/UART2 RX <sup>[1]</sup>
16	IO6	GPIO (Open drain)
17	IO5	UART1 RX
18	IO4	UART1 TX
19	IO3	GPIO
20	IO2	GPIO
21	IO1	GPIO
22	IO0	GPIO
23	V <sub>CC</sub>	Isolated 3.3VDC (board's power supply)
24	GND	AC Line.
25	I1P	Channel 1 positive pin.
26	I2P	Channel 2 positive pin.
27	I3P	Channel 3 positive pin.
28	I4P	Channel 4 positive pin.
29	V <sub>IN</sub>	AC Neutral.

[1] UART2 is only available in customize firmware, contact us for more information

Table 1: Pin function table

## TECHNICAL SPECIFICATIONS

### Maximum Absolute Rating

Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V <sub>CC</sub>		3.6	V
AC Input Voltage	V <sub>IN</sub>	AC Voltage	380	V
Digital I/O Voltage	V <sub>DIO</sub>		3.6	V
Maximum Continuous Current per Channel	I <sub>C</sub> MAX		10	A
Maximum Continuous Current	I <sub>MAX</sub>		40	A
Operating Ambient Temperature	TA		-40 to 85	°C
Junction Temperature	TJ (max)		100	°C
Storage Temperature	Tstg		-40 to 125	°C

### Isolation Characteristics

Characteristic	Symbol	Notes	Rating	Unit
Dielectric Strength Test Voltage	V <sub>ISO</sub>	Agency type-tested for 60 seconds per UL 60950-1 (edition 2) and UL 62368-1 (edition 1); Production tested at 3000 VRMS for 1 second, in accordance with UL 60950-1 (edition 2) and UL 62368-1 (edition 1)	4800	VRMS

Working Voltage for Basic Isolation	V <sub>WVBI</sub>	Maximum approved working voltage for basic (single) isolation according to UL 60950-1 (edition 2) and UL 62368-1 (edition 1)	1480	VPK or VDC
			1047	VRMS
Working Voltage for Reinforced Isolation	V <sub>WVRI</sub>	Maximum approved working voltage for reinforced isolation according to UL 60950-1 (edition 2) and UL 62368-1 (edition 1)	730	VPK or VDC
			517	VRMS
Clearance	D <sub>cl</sub>	Minimum distance through air from IP leads to signal leads	1.95	mm
Creepage	D <sub>cr</sub>	Minimum distance along package body from IP leads to signal leads	1.95	mm
Comparative Tracking Index	CTI	Material Group II	400 to 599	V

## ESD Ratings

Characteristic	Symbol	Notes	Value	Unit
Human Body Model	VHBM	Per JEDEC JS-001	±2	kV
Charged Device Model	VCDM	Per JEDEC JS-002	±1	kV

## Thermal Characteristics

Characteristic	Symbol	Test Conditions	Value	Units
Package Thermal Resistance (Junction to Ambient)	R <sub>θJA</sub>		23	°C/W
Package Thermal Resistance (Junction to Lead)	R <sub>θJL</sub>		5	°C/W

## Common Electrical Characteristics

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Unit
<b>DC SPECIFICATION</b>						
Supply Voltage	V <sub>CC</sub>	3.3 V variant	2.7	3.3	3.6	V
Supply Current	I <sub>CC</sub>	V <sub>CC(min)</sub> ≤ V <sub>CC</sub> ≤ V <sub>CC(max)</sub>	-	20	25	mA
<b>AC VOLTAGE MEASUREMENT SPECIFICATION</b>						
Voltage	V <sub>IN</sub>		90	-	380	V
Display Resolution			-	10	-	mV
Accuracy			-1	-	1	%
Number Of Channels			-	1	-	
<b>AC CURRENT MEASUREMENT SPECIFICATION</b>						
Current	I <sub>xP</sub> / I <sub>xN</sub>		0.05	-	10	A
Display Resolution			-	1	-	mA
Accuracy			-1	-	1	%
Number Of Channels			-	4	-	
<b>POWER MEASUREMENT SPECIFICATION</b>						
Power	P		4.5	-	3800	W
Display Resolution			-	10	-	mW

Accuracy			-1	-	1	%
Number Of Channels				4		channel
<b>ENERGY MEASUREMENT SPECIFICATION</b>						
Energy	E		0.0	-	10000.0	kWh
Display Resolution			-	1	-	Wh
Accuracy			-1	-	1	%
Number Of Channels			-	4	-	channel
<b>ADC SPECIFICATION</b>						
Resolution			-	16	-	bits
Reference Voltage	$V_{REF}$		-	0.6	-	V
Range			0	-	1.2	V
<b>ZERO – CROSSING SPECIFICATION</b>						
Zero – Crossing Error	$T_d$		-3	-	3	ms
Output frequency	$f_{zc}$	The grid frequency / 2	-	25	-	Hz
<b>INTERFACE</b>						
Serial port			-	2	-	port <sup>1</sup>
Baudrate			-	19200	-	bps
<b>TEMPERATURE</b>						
Maximum Temperature		At continuous 40A (with shield)	-	-	60	°C

<sup>1</sup> The secondary port is not provided in the standard firmware, contact us for a customized firmware.

## ZERO CROSSING OUTPUT

A voltage zero crossing is reported as a square wave that changes state on each zero crossing of raising voltage direction (the falling direction does not affect the state). It means the output wave's frequency is half of AC input voltage.

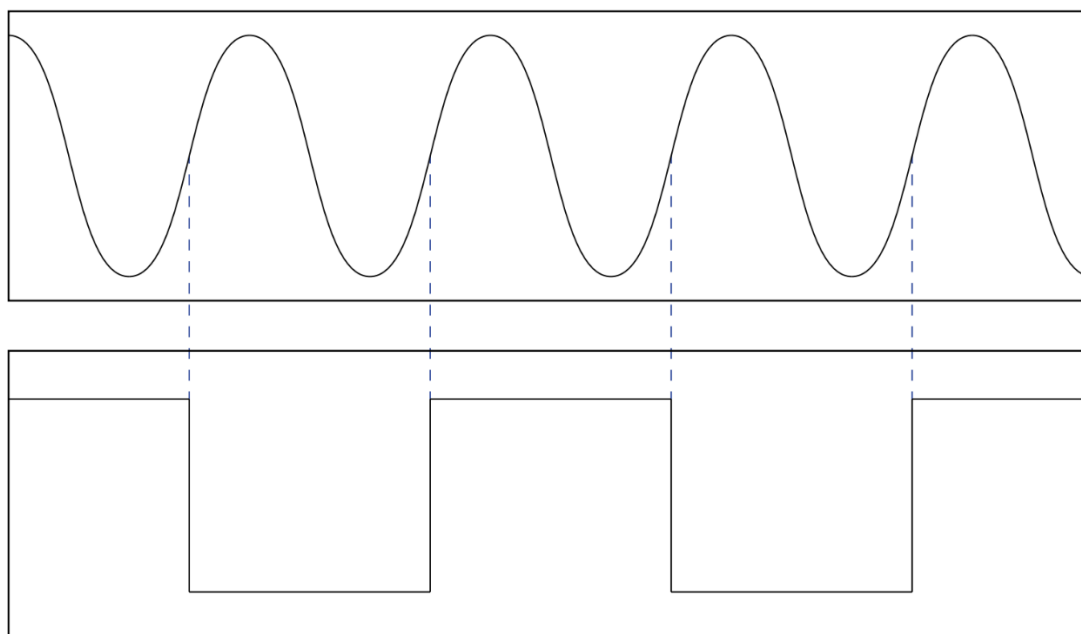


Figure 6: Zero crossing output

## AT COMMAND

### AT Command Syntax

Each AT command starts with "AT+" and ends with <CR><LF> (0x0D and 0x0A). There are three types of commands:

- Write command **AT+<x>=<...>**: This command sets user-definable parameters.
- Read command **AT+<x>?<...>**: This command gets parameters or statuses. It also provides an optional parameter to indicate which channel to get.
- Execution command **AT+<x>**: This command executes a function of the module.

Depend on the requested command, the meter may or may not return a response. If there is a response, it starts with +<x>: where <x> is the requested command. If the command does not need any specific reply, the meter returns **OK** instead.

If there is something wrong when the module is executing the command, it sends back an error message. The format of an error message is **ERROR:<reason>**.

The module also provides URC (Unsolicited Result Code) to notify users whenever an event occurs on the loads, and it needs users' attention.

### Module Operation

For providing more flexible in the PCB design process, the module uses logical channels (in this document, we use "channel" for logical channels) to manage loads. Current sense channels and residual sense channels can be freely assigned to one of the logical channels, as long as they do not conflict with the others. Each channel has an independent set of configurations. The settings of the channels are:

- Current sense channel
- Residual sense channel
- Control pin(s)
- Thresholds.

There are 4 logical channels in each module, and they are enumerated from 0 to 3. Figure 7 shows the operation of each channel.

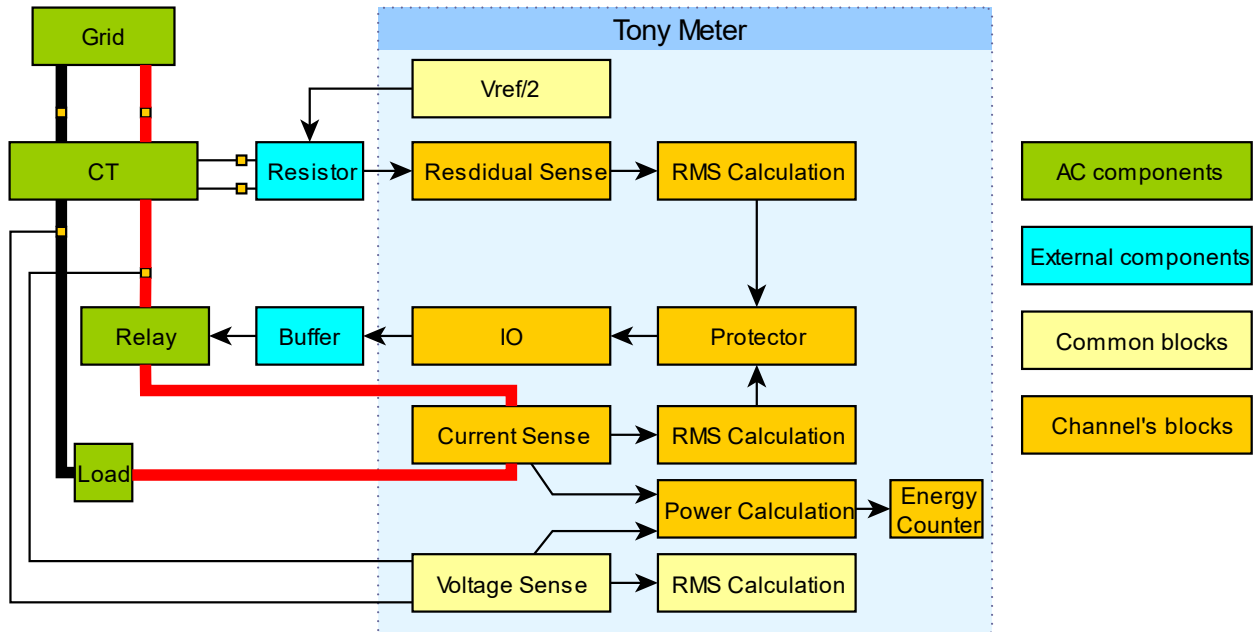


Figure 7: Tony Meter operation



---

## Energy Management

### RESETWH - Resets Energy Counter

Reset the energy counter of a logical channel. The “channel” parameter is a number from 0 to 3.

*Syntax:*

```
AT+RESETWH=<channel>\r\n
```

### READ - Reads AC parameters

This command reads AC parameters of a logical channel. The “channel” parameter is a number from 0 to 3.

*Syntax:*

```
AT+READ?<channel>\r\n
```

*Response:*

```
+READ:<channel>,<voltage>,<current>,<power>,<energy>\r\n
```

The scale factors for the parameters are 100, 1000, 100, and 1, respectively. And, the units of them are V, A, W, and Wh, respectively.

*Example:*

```
+READ:0,22000,5000,110000,10
```

The response means the module is running with 220.00 V, the current of the channel 0 is 5A, and the power is 1100W. The energy counter for this channel is 10Wh.

### TOTAL – Total Power

This command reads total parameters from all channels.

*Syntax:*

```
AT+TOTAL?\r\n
```

*Response:*

```
+TOTAL:<voltage>,<current>,<power>,<energy>\r\n
```

The format of the values in the response is the same as the +READ command.

## Load Management

### RELAY – Controlling Relays

Users cannot call this command on a disabled channel.

*Syntax:*

```
AT+RELAY=<channel>,<level>\r\n
```

Each module has an embedded EEPROM to store the last relay status. Users can get these statuses by using the query form of this command.

*Syntax:*

```
AT+RELAY?\r\n
```

*Response:*

```
+RELAY:<channel 0>,<channel 1>,<channel 2>,<channel 3>\r\n
```

*Example:*

```
+RELAY:0,1,1,0\r\n
```

The response means the channel 0 and 3 is off, the channel 1 and 2 is on.

### STORED – Getting the Stored Relay Status.

When a user controls a load, the status of the load is stored in a built-in EEPROM. The user can read these values to restore the last status if necessary.

*Syntax:*

```
AT+STORED?\r\n
```

*Response:*

```
+STORED:<channel 0>,<channel 1>,<channel 2>,<channel 3>\r\n
```

*Example:*

+STORED:0,1,1,0\r\n

### TIMEOUTCTL – Controlling Relays with Timeout

This command sets a channel to a state and toggles it after a delay. If users control the relay during the interval, the scheduled command is canceled. The maximum time for this command is 518400 seconds (one week).

Syntax:

AT+TIMEOUTCTL=<channel>,<level>,<timeout in seconds>\r\n

Syntax:

AT+TIMEOUTCTL?<channel>\r\n

Response:

+TIMEOUTCTL:<channel>,<level>,<remain time in seconds>\r\n

Example:

AT+TIMEOUTCTL=0,1,10\r\n

AT+TIMEOUTCTL=0,0,10\r\n

The first command turns on the relay of channel 0 and turns it off after 10 seconds. The second command turns off the relay of channel 1 and turns it on after 10 seconds.

### ENABLE – Enabling/Disabling a Channel

Enabling or disabling a channel. In this command, “1” means enabling, and “0” means disabling. When users call this command, the module enables this channel if all settings are valid, and the settings do not conflict with other enabled channels. The module returns an error code if there is any problem.

Syntax:

AT+ENABLE=<channel>,<0/1>\r\n

Users can check which channels are enabled by using the query form.

Syntax:

AT+ENABLE?\r\n

Response:

+ENABLE:<channel 0>,<channel 1>,<channel 2>,<channel 3>\r\n

Example:

+ENABLE:0,1,1,0

The response means the channel 1 and 2 are enabled; the channel 0 and 3 are disabled.

### ADC – Setting ADC Channel

This command assigns a high-accuracy ADC to a channel. Users cannot call this command if the channel is enabled. Users can reverse the current of the ADC channel by setting the **reverse** parameter to 1, as shown in Figure 8.

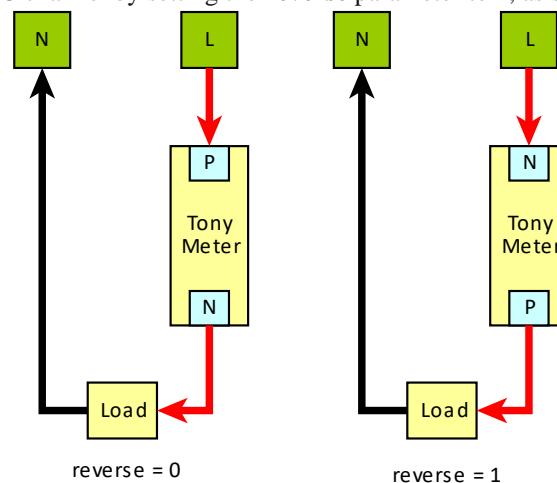


Figure 8: The reverse parameter.

Syntax:

**AT+ADC=<channel>,<adc channel>,<reverse>\r\n**

Users can check the ADC channel of a control channel by using the query form.

Syntax:

**AT+ADC?<channel>\r\n**

Response:

**+ADC:<channel>,<adc channel>,<reverse>\r\n**

Example:

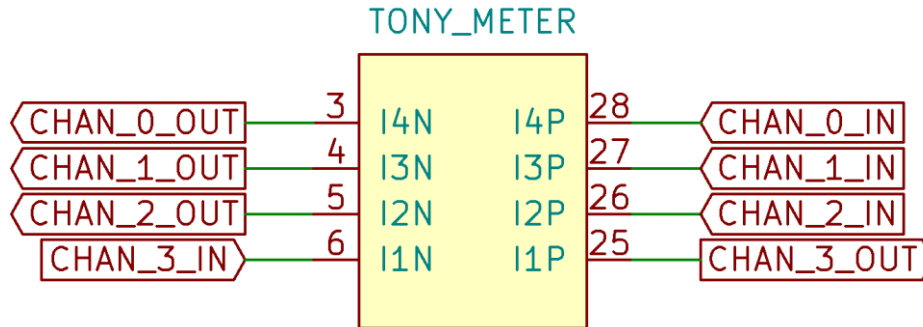


Figure 9: A sample of current sensing connections.

For the connection in Figure 9, users should use the following commands to configure the module. Note that we need to enable the “reverse” parameter of the channel 3 because the CHAN\_3\_IN signal is connected to I1N.

**AT+ADC=0,3,0\r\n**  
**AT+ADC=1,2,0\r\n**  
**AT+ADC=2,1,0\r\n**  
**AT+ADC=3,0,1\r\n**

### RELAYPINS – Changing Relay Pin(s)

This command assigns a pin (or pins for 2-coil latching relays). Depending on the coil type, this command may require 2 or 3 parameters. Users cannot call this command if the channel is enabled.

Number	Type
0	Non-latching coil
1	1-coil latching
2	2-coil latching

Syntax:

**AT+RELAYPINS=<channel>,<coil type>,<pin 1>[,<pin 2>]\r\n**

Example:

**AT+RELAYPINS=1,0,2\r\n**  
**AT+RELAYPINS=0,2,6,7\r\n**

In the example, the channel 1 controls a non-latching relay (type 0) with pin 2, and channel 0 controls a 2-coil latching relay with pin 6 and pin 7. Users can check which pin is using by using the query form.

Syntax:

**AT+RELAYPINS?<channel>\r\n**

Response:

**+RELAYPINS:<channel>,<coil type>,<pin 1>[,<pin 2>]\r\n**

Example:

**AT+RELAY?1\r\n**  
**+RELAYPINS:1,0,2\r\n**

## POLARITY – Control Signal Polarity

Sometimes, users need to invert the polarity of the control signal to compatible with their designs. The users can reverse the logic level of control by this command. Users cannot call this command if the channel is enabled. By default, the activated state is logic 1, and the inactive state is logic 0. Figure 10 shows examples circuit for the two types of the parameter. Note that IO6 and IO13 are open-drain pins; they do not work with the left circuit in Figure 10.

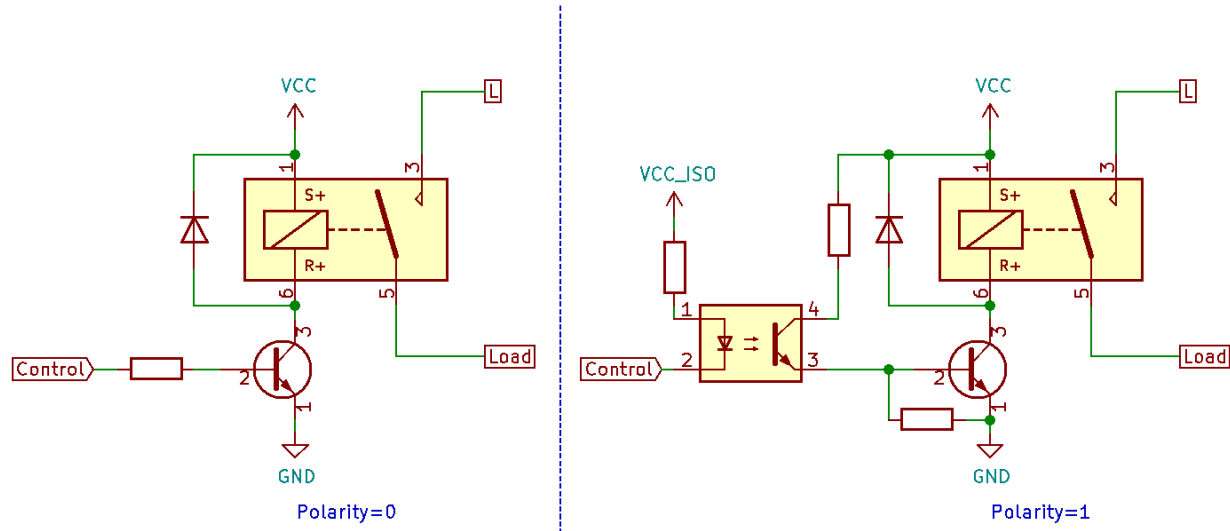


Figure 10: Sample applications for the polarity parameter.

Syntax:

**AT+POLARITY=<channel>,<0/1>\r\n**

Users can also query this parameter by using its query form.

Syntax:

**AT+POLARITY?<channel>\r\n**

Response:

**+POLARITY:<channel>,<0/1>\r\n**

## RELAYCNT – Counting the Number of Relay Switching

Users can check how many times a relay change its state.

Syntax:

**AT+RELAYCNT?<channel>\r\n**

Response:

**+RELAYCNT:<channel>,<count>\r\n**

## ONDELAY and OFFDELAY - Relay Time Compensation

When a relay changes its state, it takes a small amount of time (several milliseconds) to switch to the new state. The time varies between relay parts. Therefore, users must tune these parameters to get an accurate result. The number is represented in sixth-millisecond blocks. For example, if the set time is 2 milliseconds, the parameter is 12.

Syntax:

**AT+ONDELAY=<channel>,<set time>\r\n**

**AT+OFFDELAY=<channel>,<reset time>\r\n**

Users can check these parameters by using their query form.

Syntax:

**AT+ONDELAY?<channel>\r\n**

**AT+OFFDELAY?<channel>\r\n**

Response:

**+ONDELAY:<channel>,<set time>\r\n**

**+OFFDELAY:<channel>,<reset time>\r\n**

---

## OVERLOAD – Setting the Overload Protection Threshold

Set the overload protection limit for the module. The module switches off the relay if the current of a channel exceeds this value for the “**delay time**”. Users can disable this feature by setting the current threshold to the rating current.

*Syntax:*

**AT+OVERLOAD=<channel>,<current in mA>,<delay time in ms>\r\n**

Users can check these parameters by using its query form.

*Syntax:*

**AT+OVERLOAD?<channel>\r\n**

*Response:*

**+OVERLOAD=<channel>,<current in mA>,<delay time in ms>\r\n**

## NOLOAD – Setting No-load Protection Threshold

Set the no-load protection threshold for the module. The module switches off a relay if the current of the channel is under a threshold after the “**delay time**”. Users can disable this feature by setting the current threshold to 0 mA.

*Syntax:*

**AT+NOLOAD=<channel>,<current in mA>,<delay time in ms>\r\n**

Users can check these parameters by using its query form.

*Syntax:*

**AT+NOLOAD?<channel>\r\n**

*Response:*

**+NOLOAD=<channel>,<current in mA>,<delay time in ms>\r\n**

## LOSTLOAD – Setting No-load Protection Threshold

Set the lost-load protection threshold for the module. The module switch off a channel if the current is smaller than the threshold after the “**delay time**”. Users cannot set the delay time under 100 ms.

*Syntax:*

**AT+LOSTLOAD=<channel>,<threshold in mA>,<delay time in ms>\r\n**

Users can check these parameters by using its query form.

*Syntax:*

**AT+LOSTLOAD?<channel>\r\n**

*Response:*

**+LOSTLOAD=<channel>,<threshold in mA>,<delay time in ms>\r\n**

## UNDERVOLT – Setting Under-Voltage Threshold

Set the under-voltage protection threshold for the module. The module switches off all relays if the voltage is smaller than a threshold after the “**delay time**”. And it will stop alert if the voltage larger than the recover threshold. Users can disable this feature by setting the threshold to 0V. The default threshold is 0V.

*Syntax:*

**AT+UNDERVOLT=<threshold>,<recover threshold>,<delay time in ms>\r\n**

*Syntax:*

**AT+UNDERVOLT?\r\n**

*Response:*

**+UNDERVOLT:<threshold>,<recover threshold>,<delay time in ms>\r\n**

*Example:*

**AT+UNDERVOLT=20000,21000,1000\r\n**

After executing the command, the module sends a +UNDERVOLTALERT message if the voltage is smaller than 200V for 1s. The alert will stop if the voltage is larger than 210V.

## OVERVOLT – Setting Over-Voltage Threshold

Set the over-voltage protection threshold for the module. The module switches off all relays if the voltage is bigger than a threshold after the “delay time”. Users can disable this feature by setting the threshold to a large value. The default threshold is 400V.

*Syntax:*

**AT+OVERVOLT=<threshold>,<recover threshold>,<delay time in ms>\r\n**

*Syntax:*

**AT+OVERVOLT?\r\n**

*Response:*

**+OVERVOLT:<threshold>,<recover threshold>,<delay time in ms>\r\n**

*Example:*

**AT+OVERVOLT=30000,25000,1000\r\n**

After executing the command, the module sends a +OVERVOLTALERT message if the voltage is larger than 300V for 1s. The alert will stop if the voltage is smaller than 250V.

## IO – Controlling GPIO pins

Users can control the output of unassigned IO pins. If the pin is busy, the module returns the **USED-PIN** code.

*Syntax:*

**AT+IO=<pin>,<logic>\r\n**

## IOCLEAR – Clearing the Setting of a GPIO

Clear the setting of a specific GPIO pin. After executing this command, the pin becomes high-impedance input.

*Syntax:*

**AT+IOCLEAR=<pin>\r\n**

## RESDETECT – Enable Residual Current Detection

This command enables the ADC feature of the pins from IO8 to IO11. If the pin is busy, the module returns the **USED-PIN** code. The “**adc pin**” parameter is a number from 0 to 3. The “**thresh**” is the RMS of residual current.

*Syntax:*

**AT+RESDETECT=<channel>,<adc pin>,<gain>,<thresh>\r\n**

Users also can check which channel is enabled by using the query form.

*Syntax:*

**AT+RESDETECT?<channel>\r\n**

*Response:*

**+RESDETECT:<channel>,<gain>,<thresh>\r\n**

*Example:*

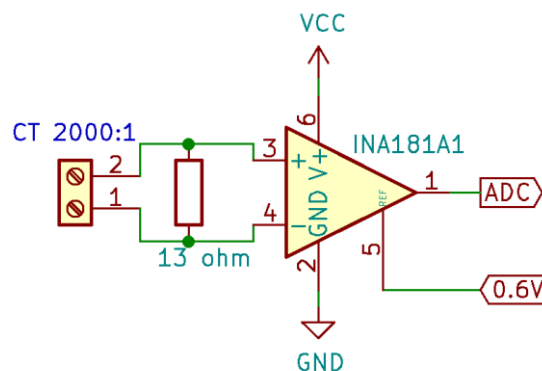


Figure 11: An example of a residual current detection circuit.

Suppose we want to turn off our load at channel 0 if the residual current is larger than 30 mA and have the circuit like Figure 11. The ratio of the current transformer (CT) is 2000:1, the resistor is 13 ohms, the gain factor of the op-amp is 20, and the ADC pin is 1. The command should be:

```
AT+RESDETECT=0,1,130,30\r\n
```

Where the gain is:  $\frac{1}{2000} \times 13 \times 20 \times 1000 = 130$ .

## RESVAL – Reading Residual Current

This command reads the residual current of a channel. The “**channel**” parameter is a number from 0 to 3. The value returned by this command depends on the **gain** parameter of the **+RESDETECT** command. By default, the **gain** parameter is 1.

*Syntax:*

```
AT+RESVAL?<channel>\r\n
```

*Response:*

```
+RESVAL:<channel><current in mA>\r\n
```

*Example:*

```
+RESVAL:0,10
```

The response means that the residual current of channel 0 is 10 mA.

## FREQ – Getting the Frequency

This command gets the current frequency of the voltage signal. The scale factor of the frequency is 100.

*Syntax:*

```
AT+FREQ?\r\n
```

*Response:*

```
+FREQ:<frequency>\r\n
```

*Example:*

```
+FREQ:5000\r\n
```

The response means the frequency of the voltage signal is 50 Hz.

## Module Management

### ID – Getting the Identify Number

This command gets the unique 128-bit ID number of the module. The result of this command is in the hexadecimal format.

*Syntax:*

```
AT+ID?\r\n
```

*Response:*

```
+ID:<id number in hexadecimal>\r\n
```

*Example:*

```
+ID:F151000054EA00260025200331534E42\r\n
```

### UGAIN – Voltage Calibration Factors

This command gets the calibration factors of the voltage measurement.

*Syntax:*

```
AT+UGAIN=<factor 1>,<factor 2>,<factor 3>\r\n
```

Users can check these factors by using the query form.

*Syntax:*

```
AT+UGAIN?\r\n
```

*Response:*

```
+UGAIN:<factor 1>,<factor 2>,<factor3>\r\n
```

*Examples:*

```
+UGAIN:2072134208,1752,-4992\r\n
```

---

## IGAIN – Current Calibration Factors

This command gets the calibration factors of the current measurement.

*Syntax:*

```
AT+IGAIN=<channel>,<factor 1>,<factor 2>,<factor 3>\r\n
```

Users can check these factors by using the query form.

*Syntax:*

```
AT+IGAIN?<channel>\r\n
```

*Response:*

```
+IGAIN:<channel>,<factor 1>,<factor 2>,<factor 3>\r\n
```

*Example:*

```
+IGAIN:0,1289960380,1024,143409\r\n
```

## ZEROCROSS – Enable zero-cross output

If users want to control their loads, the module can send out zero-cross pulses to help them control the relays precisely. If the pin is busy, the module returns the **USED-PIN** code. The pin generates a square-wave signal. The frequency of the signal is half of the mains frequency.

*Syntax:*

```
AT+ZEROCROSS=<pin>\r\n
```

## RESSEN – Residual Sensitivity

This command sets the sensitivity level of the module. The more sensitivity, the quicker the module reacts to the residual current, but the probability of false alarms also larger. There are three levels of sensitivity. They are enumerated from 0 to 2. The default sensitivity is level 1.

*Syntax:*

```
AT+RESSEN=<level>\r\n
```

Users can check the sensitivity level by using the query form.

*Syntax:*

```
AT+RESSEN?\r\n
```

## AUTOCALIB – Start Auto-calibration Process

This command starts the auto-calibration process of the module. Once this mode is activated, users can only stop it by resetting the module or the wait for the calibration success. **Users should not use this command.**

*Syntax:*

```
AT+AUTOCALIB\r\n
```

## REBOOT – Reboot the Module

When a module receives this command, it resets immediately and does not give any feedback.

*Syntax:*

```
AT+REBOOT\r\n
```

## URC (Unsolicited Result Code)

### +SYSSTART – System Start

*Syntax:*

```
+SYSSTART\r\n
```

The module sends this code every time it finishes the boot-up process. If users receive this message, the users can start to set the module's parameter. If the users receive this URC during the module operation, probably, there are problems in the system.

### +RESIDUALALERT – Residual Current Detected

*Syntax:*



### **+RESIDUAL:<channel>\r\n**

The module sends this code if there is a leakage current on a channel.

### **+UNDERVOLTALERT and +OVERVOLTALERT – Voltage Alerts**

*Syntax:*

**+UNDERVOLTALERT\r\n**

**+OVERVOLTALERT\r\n**

The module sends this code if the voltage is larger or smaller than a predefined range. The range can be set by **OVERVOLTALERT** and **UNDERVOLTALERT** command.

### **+NOLOADALERT and +OVERLOADALERT – Current Alerts**

*Syntax:*

**+NOLOADALERT:<channel>\r\n**

**+OVERLOADALERT:<channel>\r\n**

The module sends this code if the current of a channel is larger or smaller than a predefined range. The range can be set by **NOLOADALERT** and **OVERLOADALERT** commands.

### **+LOSTLOADALERT – Lost Load Alerts**

*Syntax:*

**+LOSTLOADALERT:<channel>\r\n**

The module sends this code if the load is disconnected from the module.

### **+ STICKINGALERT**

*Syntax:*

**+STICKINGALERT:<channel>\r\n**

The module sends this code if a relay is sticking.

### **+TIMEOUTNOTIFY – Notify Timeout**

*Syntax:*

**+TIMEOUTNOTIFY:<channel>,<state>\r\n**

The module sends this code if a deadline of **+TIMEOUTCTL** is met.

## **Error Codes**

Error string	Problems	Resolve
NOT-FOUND	The command is wrong.	Users should check the input commands.
TOO-LONG	The line is too long to process (more than 128 bytes).	Make sure all commands end with “\r\n”.
INVALID-PARAM	The parameters for the requested command are invalid.	Users should check the parameters.
INVALID-CHARACTER	The command contains an invalid character.	If the module sends this code rapidly without any input, users should check the serial port connection. If the module sends this code after executing a command, retry the command. If the error code appears frequently, users should check the serial port connection.
RELAY-CONTROL-FAILED	The module cannot control the relay because the module cannot detect zero-crosses.	Replace another module.

USED-PIN	The module cannot assign a function to a defined pin.	Check the settings.
DENIED	The module does not allow to execute the command.	If users receive this error after executing a <b>RELAY</b> , a <b>RELAYPINS</b> , or an <b>ADC</b> command, make sure the channel is <b>disabled</b> . If users receive this error after an <b>IOCLEAR</b> command, make sure the parameter is not one of the pins the serial port (IO4 and IO5).

Table 2: Error codes

## SAMPLE CONFIG

Figure 12 shows an example application of Tony Meter. The module in the sample application controls 4 relays through 4 IO pins: 0, 1, 2, 3.

<pre> AT+ENABLE=0,0 AT+ENABLE=1,0 AT+ENABLE=2,0 AT+ENABLE=3,0 AT+ADC=0,3,0 AT+ADC=1,2,0 AT+ADC=2,1,0 AT+ADC=3,0,0 AT+RELAYPINS=0,0,3 AT+RELAYPINS=1,0,0 AT+RELAYPINS=2,0,1 AT+RELAYPINS=3,0,2 AT+ONDELAY=0,90 AT+ONDELAY=1,90 AT+ONDELAY=2,90 AT+ONDELAY=3,90 AT+OFFDELAY=0,90 AT+OFFDELAY=1,90 AT+OFFDELAY=2,90 AT+OFFDELAY=3,90 AT+RESDETECT=0,0,130,30 AT+ENABLE=0,1 AT+ENABLE=1,1 AT+ENABLE=2,1 AT+ENABLE=3,1 </pre>	<p>Disable all channels to prevent unexpected conflicts.</p> <p>Channel 0 uses I3 without reverse. Channel 1 uses I2 without reverse. Channel 2 uses I1 without reverse. Channel 3 uses I0 without reverse.</p> <p>Channel 0 uses IO3. Channel 1 uses IO0. Channel 2 uses IO1. Channel 3 uses IO2.</p> <p>The relay set time is 15 ms The relay set time is 15 ms. The relay set time is 15 ms. The relay set time is 15 ms.</p> <p>The relay reset time is 15 ms. The relay reset time is 15 ms. The relay reset time is 15 ms. The relay reset time is 15 ms.</p> <p>The residual detection threshold is 30 mA, and the gain is 0.13.</p> <p>Enable all channels, apply the configurations.</p>
--	---

Table 3: Sample Configuration

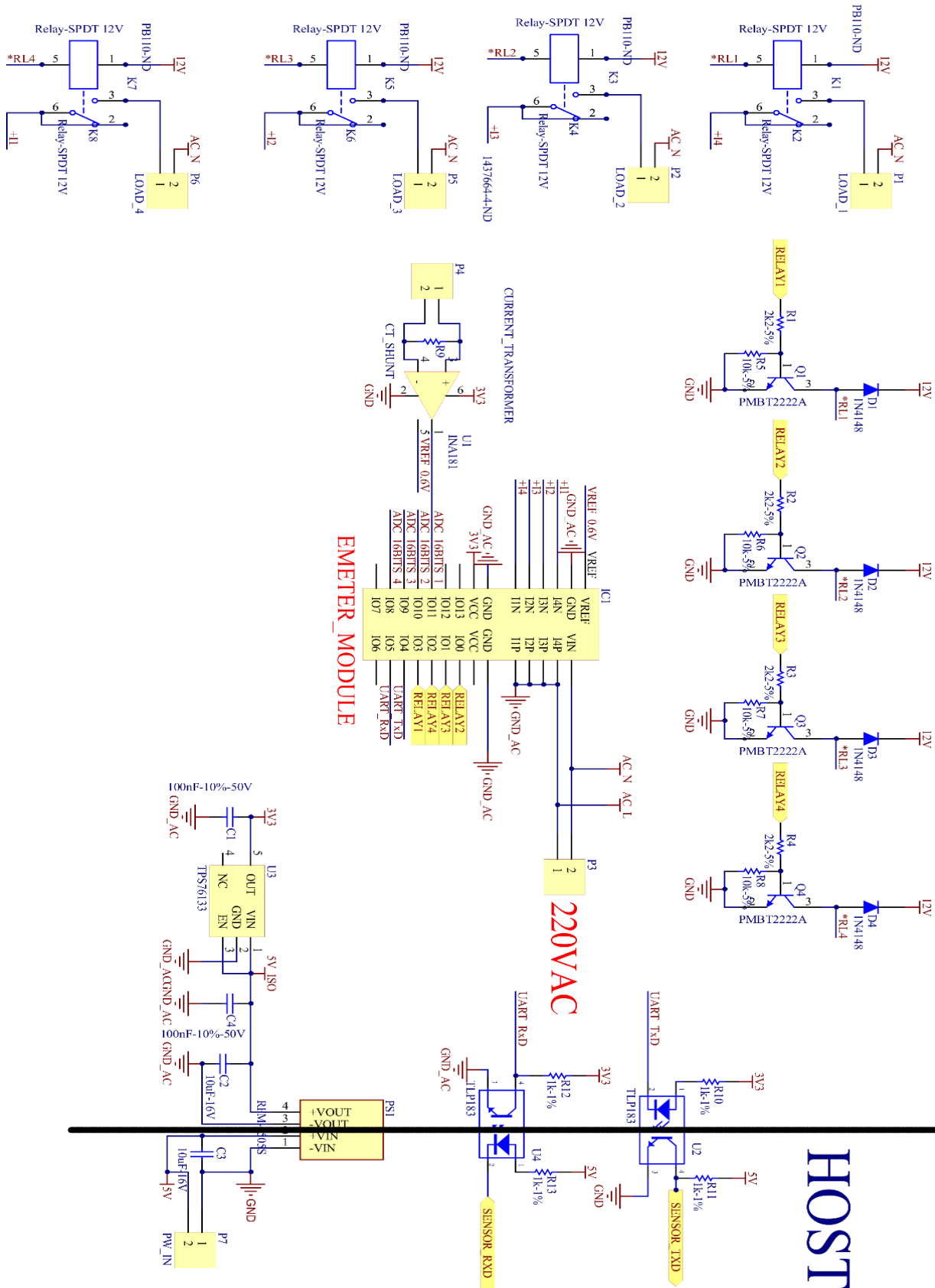


Figure 12: Sample application

## RMS & POWER OUTPUT ERROR VS APPLIED INPUT

When using the ATM1301 to measure for RMS calculations and power monitoring, it is important to consider the error specifications of the device. The following figures display the measurement error for the RMS current and active power.

Testing machine: KP-S3000-12 Three-phase energy meter test bench.

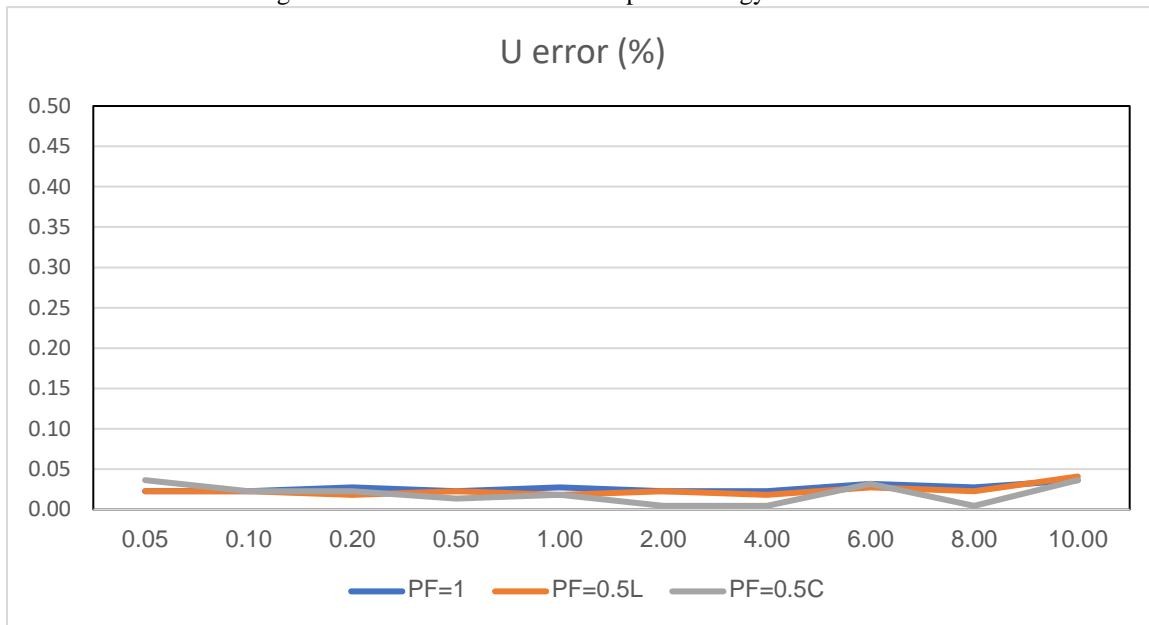


Figure 13: The measurement error for the voltage

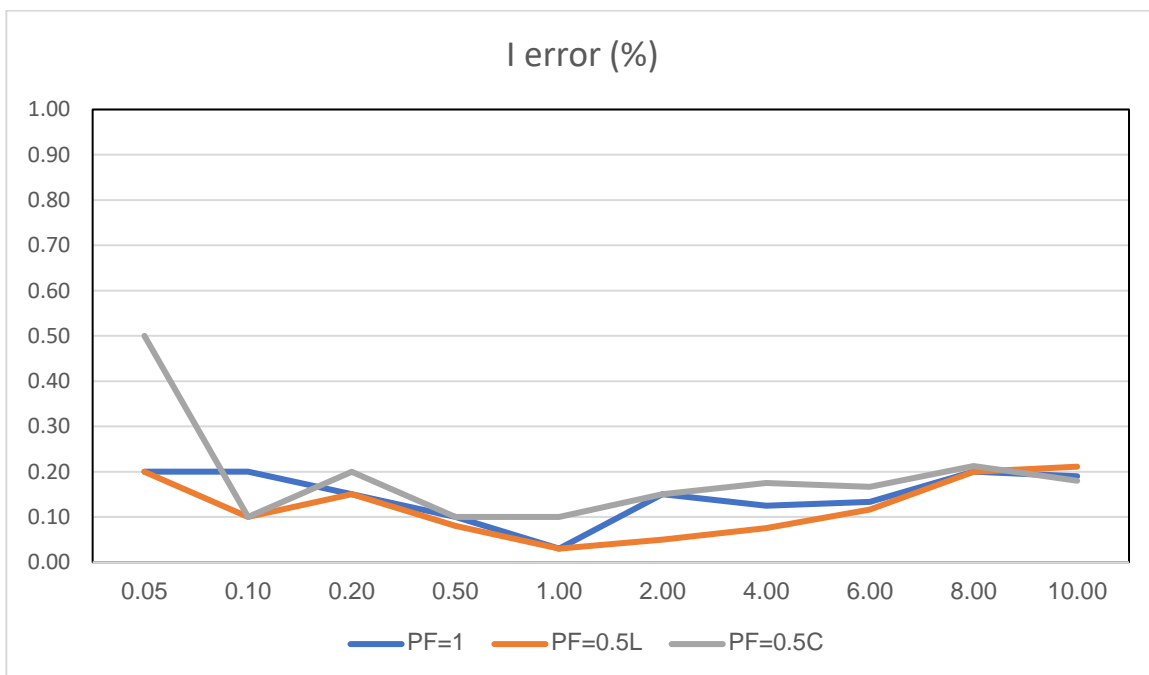


Figure 14: The measurement error for the RMS current

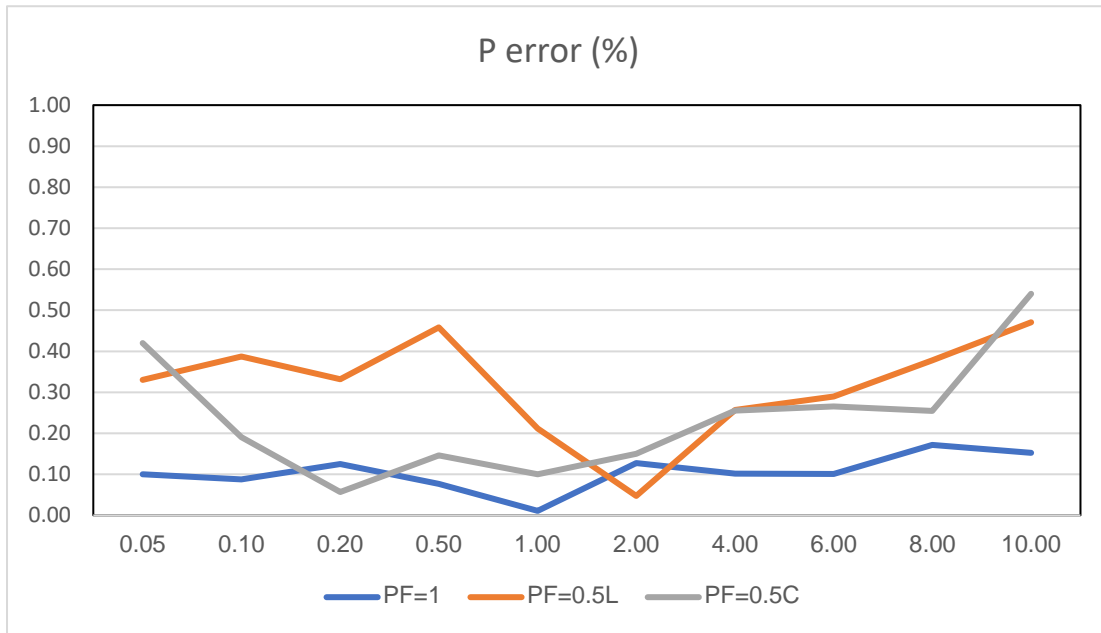


Figure 15: The measurement error for the active power

## THERMAL PERFORMANCE

Self-heating due to the flow of current should be considered during the design of any current sensing system. The module, printed circuit board (PCB), and contacts to the PCB will generate heat as current moves through the system. The thermal capacity of the ATM1301 should be verified by the end user in the application’s specific conditions. The maximum junction temperature should not be exceeded.

### Thermal Test

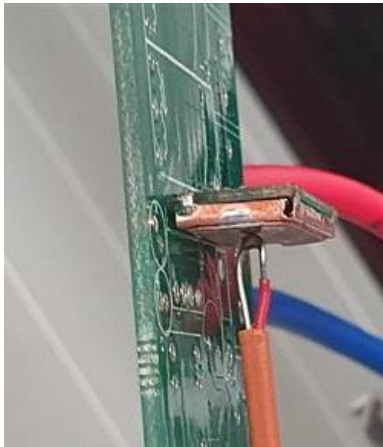


Figure 16: Thermal test position



Figure 17: Thermal test board (bottom)

Current (A)	Max Temperature (°C)
30	50
40	60

## OPTIONS

Number of channels	The maximum current of each channel (A)
1	10
1	20
1	30
1	40
2	10
2	20
3	10
4	10
1	CT
2	CT
3	CT
4	CT

---

## REVISION HISTORY

Version	Date	Description
0.1	13/01/2016	First version