



# CAYLAR

INSTRUMENTATION SCIENTIFIQUE

## NMR20 Teslameter

High Precision Teslameter  
version 2022



# COMMUNICATION INTERFACE MANUAL

### Caylar SAS

14 avenue du Québec  
BP 612  
91140 Villebon-sur-Yvette  
Tél +33 1 69 29 91 62

January 2024  
Rev 1.1



# Summary

- Equipment setup ..... 4
  - a) Teslameter ethernet settings configuration ..... 4
- ETHERNET Communication Protocol ..... 6
  - a) Introduction..... 6
    - Sending commands :..... 6
    - Teslameter response to a command :..... 6
  - b) List of Commands .....7
- Programs examples..... 20
  - Python 3.7 example..... 20
  - LabVIEW Example .....21
- Notes.....22

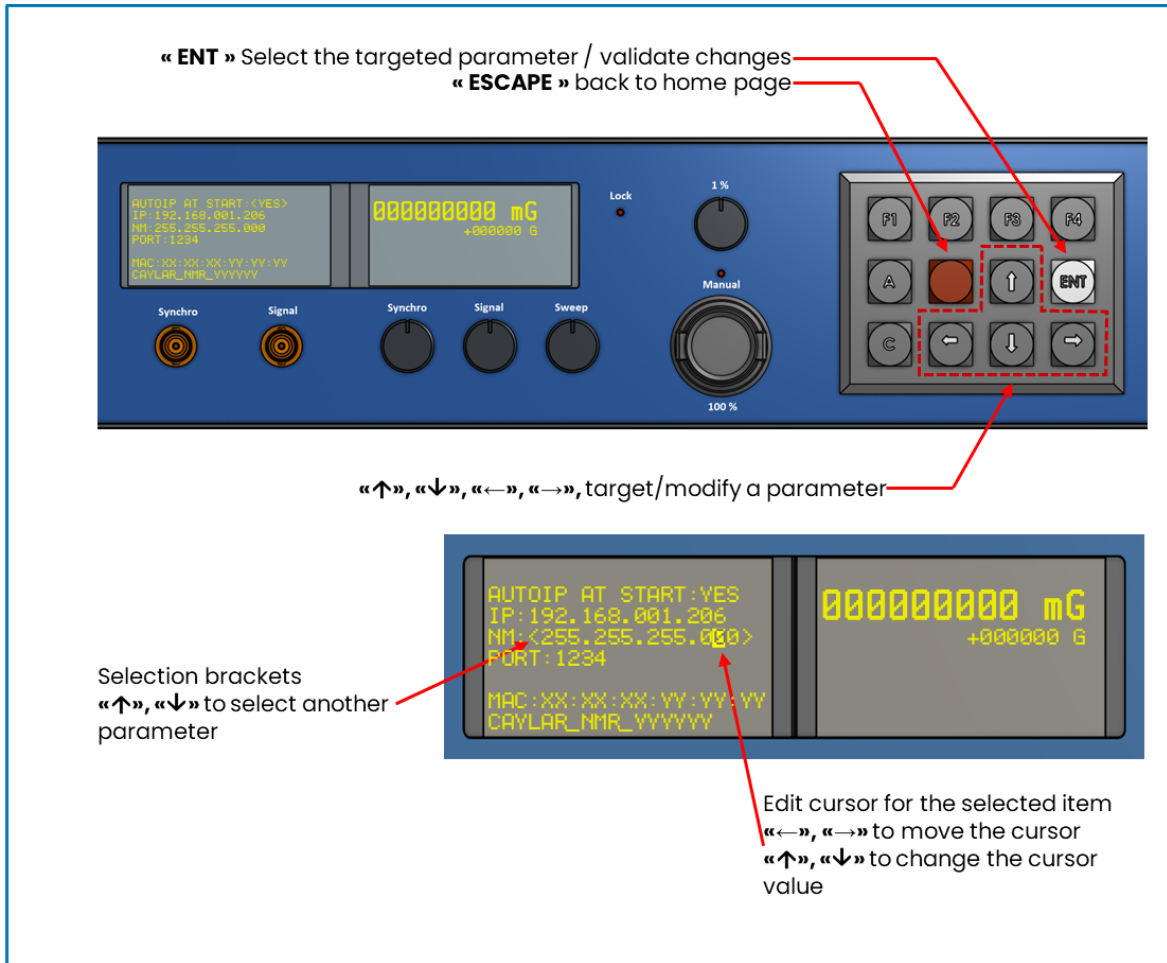
## Equipment setup

---

### a) Teslameter ethernet settings configuration

1. Turn on the Teslameter.
2. After the Teslameter initialization, push the « **F1** » keyboard key to enter into the « ETHERNET » menu. It allows you to configure the various parameters of the Ethernet interface. Each change made in this menu is then permanently saved in the power supply memory. You will therefore find the same parameters during the next power on.
3. **IMPORTANT NOTE:** At any time, you can press the red keyboard key (« **Escape** ») to exit the menu you are in.
4. For selecting the various parameters present in the "ETHERNET" menu, use the « **↓** » and « **↑** » keyboard keys. The selected parameter is surrounded by brackets (ex: <xxxx> where xxxx is the parameter).
5. To modify the selected parameter, push the « **ENT** » keyboard key. The editable part of the parameter is then highlighted to indicate the position of the edit cursor.
6. Then use « **←** » or « **→** » keyboard keys to move the edit cursor and « **↓** » or « **↑** » keys to change the selected value. Push « **C** » keyboard key if you want to cancel the modification. If you want to validate the modification then press the « **ENT** » keyboard key.
7. **NOTE:** Leaving the menu by pressing « **Escape** » key or by pressing a shortcut menu like « **F1, F2, F3, F4** » keys also automatically cancels the modification of the selected parameter.
8. Teslameter Ethernet settings are as follows:
  - « **AUTO IP AT START** » : The activation of an automatic IP search protocol on power on (close to DHCP – default: NO)
  - « **IP** » : IP address of the Teslameter (default: 192.168.001.123)
  - « **NM** » : Network Mask of the Teslameter (default: 255.255.255.000)
  - « **PORT** » : Port of the Teslameter (not editable - fixed to 1234)
  - « **MAC** » : Mac address of the Teslameter (not editable)
  - « **Host Name** » : Host name of the Teslameter if "AUTO IP AT START" is activated (not editable)

The figure below shows a view of the Teslameter front panel with keyboard and oled displays.



**View of the Teslameter front panel with Ethernet menu open**

## ETHERNET Communication Protocol

---

### a) Introduction

#### **Sending commands :**

Ethernet commands must be sent as a string (array of byte) in ASCII format **with an « \n », « \r\n » or « \r » character at the end of each order sent.** The character string must respect upper/lower characters and spaces.

Multiple commands can be sent in one time or a single command can be sent in multiple parts. Receiving one of the end-of-string characters « \n », « \r\n » or « \r », results in processing the received bytes stored in a 1024-byte buffer.

If a command requires additional arguments, these must be separated by a single space between each argument and with the command. Example: in the command "SET\_PROBE 2", the argument "2" is separated from the command SET\_PROBE by a single space.

#### **Teslameter response to a command :**

**The Teslameter gives a response for each order received.** Feedback responses are returned as strings in ASCII format with an end-of-line character (LF or « \n ») at the end of each response. The start of the response always corresponds to the copy of the command sent with an « \_OK » added if the command was executed without problem (except for "GET\_XXXXX" commands).

In case of errors during the commands processing the « \_OK » confirmation is replaced by « \_ERROR ».

In case of unrecognized commands, the power supply will return the following string: «WRONGCOMMAND \n».

In the case of a return of multiple arguments, these are separated by a single space between them.

#### **IMPORTANT NOTE :**

- It is important to check the response of Teslameter for each command sent to ensure that it has been processed correctly and to avoid Teslameter saturation in the case of commands sent too quickly at the same time.
- Some commands take time to run, so the Teslameter response may take time to arrive.

## b) List of Commands

*IDN?.....	9
GET_FIELD_NMR <FORMAT> .....	9
GET_FRQ_NMR.....	9
GET_FIELD_HALL <FORMAT>.....	9
GET_MUX.....	9
GET_PA_MUX.....	9
GET_PROBE.....	9
GET_FILTER .....	10
GET_FIELD_TYPE.....	10
GET_SIGNAL .....	10
GET_SWEEP.....	10
GET_LOCK .....	10
GET_MODE .....	10
GET_FIELD_SEARCH <FORMAT>.....	10
GET_MIN_PROBE <FORMAT> .....	11
GET_MAX_PROBE <FORMAT>.....	11
GET_FIELD_FORMAT .....	11
GET_NMR_SIGNAL .....	11
GET_REGUL_STATUS.....	12
GET_FIELD_SETPOINT <FORMAT>.....	12
GET_PARAMETER_RMN <PARAM> .....	12
GET_PARAMETER_HALL <PARAM> .....	12
GET_OUTPUT_VOLTAGE_MAX .....	13
GET_OUTPUT_VOLTAGE_MIN.....	13
GET_OUTPUT_VOLTAGE .....	13
SET_MODE <MODE> .....	14
SET_MUX <MUX> .....	14
SET_PA_MUX <PA_MUX>.....	14
SET_PROBE <PROBE> .....	14
SET_FILTER <FILTER>.....	14
SET_FIELD_TYPE <FIELD_TYPE> .....	15
SET_SIGNAL <SIGNAL> .....	15
SET_SWEEP <SWEEP> .....	15
SET_FIELD_SEARCH <FIELD> <Format> .....	15
SET_FIELD_FORMAT <Format> .....	16
SET_REGUL_ON .....	17
SET_REGUL_OFF .....	17
SET_REGUL_PAUSE_ON .....	17
SET_REGUL_PAUSE_OFF.....	17
SET_FIELD_SETPOINT <FIELD> <Format> .....	17
SET_PARAMETER_RMN <PARAM> <VALUE> .....	18
SET_PARAMETER_HALL <PARAM> <VALUE>.....	18

SET_OUTPUT_VOLTAGE_MAX <VOLTAGE> .....	18
SET_OUTPUT_VOLTAGE_MIN <VOLTAGE> .....	19
SET_OUTPUT_VOLTAGE <VOLTAGE> .....	19



<b>*IDN?</b>	
Response : <b>CAYLAR_2210_&lt;serial_number&gt;</b>	Ex : CAYLAR_2210_XXX
Returns the serial number of the Teslameter.	
<b>GET_FIELD_NMR &lt;FORMAT&gt;</b>	
Response : <b>&lt;NMR Field&gt; &lt;Format&gt;</b>	Ex : +0.234865968 T
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> <li>- if nothing is specified, Teslameter format</li> </ul>
Returns the last measurement of the field made by the Teslameter. The field is returned in the selected format, if there is none selected, the one used on the Teslameter is used.	
<b>GET_FRQ_NMR</b>	
Response : <b>&lt;FRQ&gt; Hz</b>	Ex : 10000001.213636 Hz
Returns the last measurement of the resonant frequency made by the Teslameter. the frequency is returned in Hz.	
<b>GET_FIELD_HALL &lt;FORMAT&gt;</b>	
Response : <b>&lt;HALL Field&gt; &lt;Format&gt;</b>	Ex : GET_FIELD_NMR 1
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> <li>- if nothing is specified, Teslameter format</li> </ul>
Returns the last measurement of the field made by the Hall sensor of the probe Teslameter. The field is returned in the selected format, if there is none selected, the one used on the Teslameter is used.	
<b>GET_MUX</b>	
Response : <b>&lt;MUX&gt; (0 to 4)</b>	Ex : 1
Returns the number of the selected MUX.	
<b>GET_PA_MUX</b>	
Response : <b>&lt;PA_MUX&gt; (1 to 8)</b>	Ex : 2
Returns the number of the selected PA_MUX.	
<b>GET_PROBE</b>	
Response : <b>&lt;PROBE&gt; (1 to 8)</b>	Ex : 3
Returns the number of the selected PROBE.	

GET_FILTER	
Response : <b>&lt;FILTER&gt; (1 to 4)</b>	Ex : 2
Returns the number of the selected FILTER.	

GET_FIELD_TYPE	
Response : <b>&lt; FIELD_TYPE &gt; (1 to 3)</b>	Ex : 1
< FIELD_TYPE >	<ul style="list-style-type: none"> <li>- 1 : MEDIUM FIELD</li> <li>- 2 : LOW FIELD</li> <li>- 3 : HIGH FIELD</li> </ul>
Returns the field type currently set on the Teslameter.	

GET_SIGNAL	
Response : <b>&lt;SIGNAL&gt; (0 to 100)</b>	Ex : 50
Returns the value of the signal currently set on the Teslameter.	

GET_SWEEP	
Response : <b>&lt;SWEEP&gt; (0 to 100)</b>	Ex : 50
Returns the value of the sweep currently set on the Teslameter.	

GET_LOCK	
Response : <b>&lt;LOCK&gt; (0 or 1)</b>	Ex : 0
Returns the status of the lock. 0 (if not lock) / 1 (if lock)	

GET_MODE	
Response : <b>&lt;MODE&gt;</b>	Ex : 3
<MODE>	<ul style="list-style-type: none"> <li>- 1 : MANUAL SEARCH</li> <li>- 2 : AUTO SEARCH</li> <li>- 3 : HALL TRACKING</li> </ul>
Returns the actual MODE set on the Teslameter.	

GET_FIELD_SEARCH <FORMAT>	Ex : GET_FIELD_SEARCH 2
Response : <b>&lt;Field search value&gt; &lt;Format&gt;</b>	Ex : +0.234865968 T
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> <li>- if nothing is specified, Teslameter format</li> </ul>
Returns the field value around which the Teslameter looks for a signal. The field is returned in the selected format, if there is none selected, the one used on the Teslameter is used.	

<b>GET_MIN_PROBE &lt;FORMAT&gt;</b>	<b>Ex: GET_MIN_PROBE 2</b>
Response : <b>&lt;min probe field &gt; &lt;Format&gt;</b>	Ex : +0.1600000 T
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> <li>- if nothing is specified, Teslameter format</li> </ul>
Returns the probe min field if the probe is "calibrated". The field is returned in the selected format, if there is none selected, the one used on the Teslameter is used.	

<b>GET_MAX_PROBE &lt;FORMAT&gt;</b>	<b>Ex: GET_MAX_PROBE 2</b>
Response : <b>&lt;max probe field &gt; &lt;Format&gt;</b>	Ex : +0.8000000 T
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> <li>- if nothing is specified, Teslameter format</li> </ul>
Returns the probe max field if the probe is "calibrated". The field is returned in the selected format, if there is none selected, the one used on the Teslameter is used.	

<b>GET_FIELD_FORMAT</b>	<b>Ex: GET_FIELD_FORMAT</b>
Response : <b>&lt;Format&gt;</b>	Ex : 2
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> </ul>
Returns the format used to display the field on the Teslameter	

<b>GET_NMR_SIGNAL</b>	<b>Ex: GET_NMR_SIGNAL</b>
Response : <b>&lt;500 bytes&gt;</b> <b>READ_OK</b>	Ex : <500 bytes> READ_OK
<byte>	- Unsigned byte straight binary (0 – 255 for +/- 15 V)
returns the NMR signal as an array of 500 bytes of 8 bits representing the signal in a range of +/- 15 V in straight binary format ( -15 V => 0, 0 V => 128, +15 V => 255) than return READ_OK. Maximum one acquisition every 20ms.	

<b>GET_REGUL_STATUS</b>	Ex : GET_REGUL_STATUS
Response : < STATUS >	Ex : REGULATION_OFF
< STATUS >	<ul style="list-style-type: none"> <li>- REGULATION_OFF</li> <li>- REGUL_HALL</li> <li>- REGUL_RMN</li> <li>- REGULATION_PAUSE</li> <li>- REGULATION_ERROR</li> </ul>
Returns the state of the regulation	

<b>GET_FIELD_SETPOINT &lt;FORMAT&gt;</b>	Ex : GET_FIELD_SETPOINT 2
Response : < SETPOINT > <Format>	Ex : +0.2500000 T
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> <li>- if nothing is specified, Teslameter format</li> </ul>
Returns the setpoint used for the field regulation.	

<b>GET_PARAMETER_RMN &lt;PARAM&gt;</b>	Ex : GET_PARAMETER_RMN P
Response : <VALUE>	Ex : 0.15
<PARAM>	PID parameter to be read for NMR control loop. <ul style="list-style-type: none"> <li>- P : Proportional</li> <li>- I : Integral</li> <li>- D : Derivate</li> </ul>
<VALUE>	Value of the parameter. Between +10 and -10.
Response in case of bad argument < PARAM > : <b>WRONG_PARAMETER</b>	
Response in case of incorrect number of argument : <b>BAD_ARG</b>	
Get the NMR loop PID parameter value.	

<b>GET_PARAMETER_HALL &lt;PARAM&gt;</b>	Ex : GET_PARAMETER_HALL P
Response : <VALUE>	Ex : 0.15
<PARAM>	PID parameter to be read for HALL control loop. <ul style="list-style-type: none"> <li>- P : Proportional</li> <li>- I : Integral</li> <li>- D : Derivate</li> </ul>
<VALUE>	Value of the parameter. Between +10 and -10.
Response in case of bad argument < PARAM > : <b>WRONG_PARAMETER</b>	
Response in case of incorrect number of argument : <b>BAD_ARG</b>	
Get the HALL loop PID parameter value.	

**GET\_OUTPUT\_VOLTAGE\_MAX**Response : **<VOLTAGE>**

Ex : +5.2

Returns the maximum voltage that can be applied to the control output

**GET\_OUTPUT\_VOLTAGE\_MIN**Response : **<VOLTAGE>**

Ex : -5.2

Returns the minimum voltage that can be applied to the control output

**GET\_OUTPUT\_VOLTAGE**Response : **<VOLTAGE> V**

Ex : +3.350000 V

Returns the actual voltage at the control output

<b>SET_MODE &lt;MODE&gt;</b>	Ex: SET_MODE 3
Response: <b>SET_MODE_OK &lt;MODE&gt;</b>	Ex: SET_MODE_OK 3
<MODE>	<ul style="list-style-type: none"> <li>- 1 : MANUAL SEARCH</li> <li>- 2 : AUTO SEARCH</li> <li>- 3 : HALL TRACKING</li> </ul>
Response in case of bad argument <MODE>: <b>BAD_ARG</b>	
Response in case of overrange in <MODE > argument: <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the MODE: <b>TESLAMETER BUSY</b>	
Allows the user to select the MODE to be used to search the signal.	

<b>SET_MUX &lt;MUX&gt;</b>	Ex: SET_MUX 1
Response: <b>SET_MUX_OK &lt;MUX&gt;</b>	Ex: SET_MUX_OK 1
<MUX>	The MUX number (0 to 4)
Response in case of bad argument <MUX> : <b>BAD_ARG</b>	
Response in case of overrange in <MUX> argument : <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the MUX : <b>TESLAMETER BUSY</b>	
Allows the user to select the MUX to be used to perform the measurement. The actual PA_MUX and PROBE doesn't change.	

<b>SET_PA_MUX &lt;PA_MUX&gt;</b>	Ex: SET_PA_MUX 2
Response: <b>SET_PA_MUX_OK &lt;PA_MUX &gt;</b>	Ex: SET_PA_MUX_OK 2
<PA_MUX>	The PA_MUX number (1 to 8)
Response in case of bad argument <PA_MUX> : <b>BAD_ARG</b>	
Response in case of overrange in <PA_MUX> argument : <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the PA_MUX : <b>TESLAMETER BUSY</b>	
Allows the user to select the PA_MUX to be used to perform the measurement. The actual MUX and PROBE doesn't change.	

<b>SET_PROBE &lt;PROBE&gt;</b>	Ex: SET_PROBE 1
Response: <b>SET_PROBE_OK &lt;PROBE&gt;</b>	Ex: SET_PROBE_OK 1
<PROBE>	The probe number (1 to 8)
Response in case of bad argument <PROBE> : <b>BAD_ARG</b>	
Response in case of overrange in <PROBE> argument : <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the probe : <b>TESLAMETER BUSY</b>	
Allows the user to select the probe to be used to perform the measurement. The actual MUX and PA_MUX doesn't change.	

<b>SET_FILTER &lt;FILTER&gt;</b>	Ex: SET_FILTER 2
Response: <b>SET_FILTER_OK &lt;FILTER&gt;</b>	Ex: SET_FILTER_OK 2
<FILTER>	The FILTER number (1 to 4)
Response in case of bad argument <FILTER> : <b>BAD_ARG</b>	
Response in case of overrange in <FILTER> argument : <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the FILTER : <b>TESLAMETER BUSY</b>	
Allows the user to select the FILTER to be used to perform the measurement.	

<b>SET_FIELD_TYPE &lt;FIELD_TYPE&gt;</b>	Ex: SET_FIELD_TYPE 1
Response: <b>SET_FIELD_TYPE_OK &lt;MODE&gt;</b>	Ex: SET_FIELD_TYPE_OK 1
<FIELD_TYPE>	<ul style="list-style-type: none"> <li>- 1 : MEDIUM FIELD</li> <li>- 2 : LOW FIELD</li> <li>- 3 : HIGH FIELD</li> </ul>
Response in case of bad argument <FIELD_TYPE> : <b>BAD_ARG</b>	
Response in case of overrange in <FIELD_TYPE> argument : <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the FIELD_TYPE : <b>TESLAMETER BUSY</b>	
Allows the user to select the FIELD_TYPE to be used to search the signal.	

<b>SET_SIGNAL &lt;SIGNAL&gt;</b>	Ex: SET_SIGNAL 35
Response: <b>SET_SIGNAL_OK &lt;SIGNAL&gt;</b>	Ex: SET_SIGNAL_OK 35
<SIGNAL>	The SIGNAL value (0 to 100)
Response in case of bad argument <SIGNAL>: <b>BAD_ARG</b>	
Response in case of overrange in < SIGNAL > argument: <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the SIGNAL: <b>TESLAMETER BUSY</b>	
Allows the user to change the value of the SIGNAL.	

<b>SET_SWEEP &lt;SWEEP&gt;</b>	Ex: SET_SWEEP 62
Response: <b>SET_SWEEP_OK &lt;SWEEP&gt;</b>	Ex: SET_SWEEP_OK 62
<SIGNAL>	The SIGNAL value (0 to 100)
Response in case of bad argument < SWEEP >: <b>BAD_ARG</b>	
Response in case of overrange in < SWEEP > argument: <b>OVERRANGE</b>	
Response in case the Teslameter cannot change the SWEEP: <b>TESLAMETER BUSY</b>	
Allows the user to change the value of the SWEEP.	

<b>SET_FIELD_SEARCH &lt;FIELD&gt; &lt;Format&gt;</b>	Ex : SET_FIELD_SEARCH 0.234865968 T
Response : <b>SET_FIELD_SEARCH_OK &lt;Field &gt;&lt;Format&gt;</b>	Ex : SET_FIELD_SEARCH_OK 0.234865968T
<FIELD>	Field value around which the Teslameter looks for a signal. Between min and max of probe range.
<FORMAT>	<ul style="list-style-type: none"> <li>- mG : mGauss</li> <li>- G : Gauss</li> <li>- T : Tesla</li> <li>- uT : uTesla</li> <li>- mT : mTesla</li> </ul>
Response in case of bad argument < Format >: <b>WRONG_FIELD_UNITE</b>	
Response in case of bad argument < FIELD > and < Format >: <b>BAD_ARG</b>	
Response in case of overrange in < FIELD > argument: <b>OVERRANGE</b>	
Set the field value around which the Teslameter looks for a signal. the field must be between the Min and Max of the probe range. The probe must be "calibrated" beforehand. Using this command switches the Teslameter to "Digital" mode.	

<b>SET_FIELD_FORMAT &lt;Format&gt;</b>	<b>Ex : SET_FIELD_FORMAT 2</b>
Response : <b>SET_FIELD_FORMAT_OK &lt;Format&gt;</b>	<b>Ex : SET_FIELD_FORMAT_OK 2</b>
<FORMAT>	<ul style="list-style-type: none"> <li>- 0 : mGauss</li> <li>- 1 : Gauss</li> <li>- 2 : Tesla</li> <li>- 3 : uTesla</li> <li>- 4 : mTesla</li> </ul>
Set the format used to display the field on the Teslameter.	



**Control available with the Field Control option.**
**SET\_REGUL\_ON**

 Response : **SET\_REGUL\_ON\_OK**

Set ON the field regulation.

The Teslameter starts to regulate the field according to the last setpoint.

**SET\_REGUL\_OFF**

 Response : **SET\_REGUL\_OFF\_OK**

Set OFF the field regulation.

The Teslameter stop the regulation and set the voltage output to 0V.

**SET\_REGUL\_PAUSE\_ON**

 Response : **SET\_REGUL\_PAUSE\_ON\_OK**

 Response if regulation is OFF : **REGULATION\_IS\_OFF**

Set the field regulation to PAUSE.

The Teslameter freeze the output voltage at the last value.

**SET\_REGUL\_PAUSE\_OFF**

 Response : **SET\_REGUL\_PAUSE\_OFF\_OK**

 Response if regulation is OFF : **REGULATION\_IS\_OFF**

Set field regulation to ON if it was paused.

The Teslameter resumes field regulation according to the last setpoint.

**SET\_FIELD\_SETPOINT <FIELD> <Format>**

Ex : SET\_FIELD\_SETPOINT 0.2348659 T

 Response : **SET\_FIELD\_SETPOINT\_OK <Field>  
>><Format>**

Ex : SET\_FIELD\_SETPOINT\_OK 0.2348659T

&lt;FIELD&gt;

Field value for regulation setpoint

&lt;FORMAT&gt;

- mG : mGauss
- G : Gauss
- T : Tesla
- uT : uTesla
- mT : mTesla

 Response in case of bad argument < Format > : **WRONG\_FIELD\_UNITE**

 Response in case of bad argument < FIELD > and < Format > : **BAD\_ARG**

Set the field setpoint used for field control.

<b>SET_PARAMETER_RMN &lt;PARAM&gt; &lt;VALUE&gt;</b>	Ex: SET_PARAMETER_RMN P 0.15
Response : <b>SET_PARAMETRE_RMN_OK &lt;PARAM&gt; &lt;VALUE&gt;</b>	Ex: SET_PARAMETRE_RMN_OK P 0.15
<PARAM>	PID parameter to be modified for NMR control loop. <ul style="list-style-type: none"> <li>- P : Proportional</li> <li>- I : Integral</li> <li>- D : Derivate</li> </ul>
<VALUE>	Value to apply to the parameter. Between +10 and -10.
Response in case of bad argument < PARAM >: <b>WRONG_PARAMETER</b>	
Response in case of bad argument < VALUE >: <b>OVERRANGE</b>	
Response in case of bad argument < PARAM > and < VALUE >: <b>BAD_ARG</b>	
Set the NMR loop PID parameter value.	

<b>SET_PARAMETER_HALL &lt;PARAM&gt; &lt;VALUE&gt;</b>	Ex: SET_PARAMETER_HALL P 0.15
Response : <b>SET_PARAMETRE_HALL_OK &lt;PARAM&gt; &lt;VALUE&gt;</b>	Ex: SET_PARAMETRE_HALL_OK P 0.15
<PARAM>	PID parameter to be modified for HALL control loop. <ul style="list-style-type: none"> <li>- P : Proportional</li> <li>- I : Integral</li> <li>- D : Derivate</li> </ul>
<VALUE>	Value to apply to the parameter. Between +10 and -10.
Response in case of bad argument < PARAM >: <b>WRONG_PARAMETER</b>	
Response in case of bad argument < VALUE >: <b>OVERRANGE</b>	
Response in case of bad argument < PARAM > and < VALUE >: <b>BAD_ARG</b>	
Set the HALL loop PID parameter value.	

<b>SET_OUTPUT_VOLTAGE_MAX &lt;VOLTAGE&gt;</b>	Ex: SET_OUTPUT_VOLTAGE_MAX 5
Response: <b>SET_OUTPUT_VOLTAGE_MAX_OK &lt;VOLTAGE&gt;</b>	Ex: SET_OUTPUT_VOLTAGE_MAX_OK 5
< VOLTAGE >	The voltage value (-10 to +10)
Response in case of bad argument < VOLTAGE >: <b>BAD_ARG</b>	
Response in case of overrange in < VOLTAGE > argument: <b>OVERRANGE</b>	
Response in case < VOLTAGE > argument is smaller than minimum output voltage: <b>OUT_MAX_SMALLER_THAN_OUT_MIN</b>	
Allows the user to define the maximum voltage that can be applied to the control output.	

<b>SET_OUTPUT_VOLTAGE_MIN &lt;VOLTAGE&gt;</b>	Ex: SET_OUTPUT_VOLTAGE_MIN -5
Response: <b>SET_OUTPUT_VOLTAGE_MIN_OK &lt;VOLTAGE&gt;</b>	Ex: SET_OUTPUT_VOLTAGE_MIN_OK -5
<VOLTAGE>	The voltage value (-10 to +10)
Response in case of bad argument < VOLTAGE >: <b>BAD_ARG</b>	
Response in case of overrange in < VOLTAGE > argument: <b>OVERRANGE</b>	
Response in case < VOLTAGE > argument is greater than maximum output voltage: <b>OUT_MIN_GREATER_THAN_OUT_MAX</b>	
Allows the user to define the minimum voltage that can be applied to the control output.	

<b>SET_OUTPUT_VOLTAGE &lt;VOLTAGE&gt;</b>	Ex: SET_OUTPUT_VOLTAGE -5
Response: <b>SET_OUTPUT_VOLTAGE_OK &lt;VOLTAGE&gt;</b>	Ex: SET_OUTPUT_VOLTAGE_OK -5
<VOLTAGE>	The voltage value (-10 to +10)
Response in case of bad argument < VOLTAGE >: <b>BAD_ARG</b>	
Response in case of overrange in < VOLTAGE > argument: <b>OVERRANGE</b>	
Allows the user to define the voltage applied to the control output.	

## Programs examples

---

### Python 3.7 example

This program connects to the teslameter via python socket Ethernet library. It requests the last field measurement made by the teslameter and print the result in the python console.

Screenshot of the program :

```
import socket
import traceback

Teslameter_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
Teslameter_sock.settimeout(5) # socket timeout
Teslameter_IP = "192.168.1.123" # Replace by the IP address of the Teslameter
Teslameter_PORT = 1234 # Port of the Teslameter (Fixe)

try:
    Teslameter_sock.connect((Teslameter_IP, Teslameter_PORT))
    Teslameter_sock.send(
        bytes("GET_FIELD_NMR\n", "ascii")) # send the command to read the field measure by the Teslameter
    NMR_FIELD, FIELD_FORMAT = Teslameter_sock.recv(50).decode("ascii").split(' ') # get field value and field format
    Teslameter_sock.send(bytes("GET_LOCK\n", "ascii")) # send the command to know if the Teslameter is lock
    NMR_Lock = int(Teslameter_sock.recv(50).decode("ascii").replace('\n', '').replace('\r', '')) # get lock status
    if NMR_Lock:
        print("Teslameter LOCK, Field = " + NMR_FIELD + ' ' + FIELD_FORMAT)
    else:
        print("Teslameter not LOCK")
except:
    print("socket error ...")
    traceback.print_exc()

finally:
    Teslameter_sock.close()
```

Screenshot of the program output (Console View) :

If the Teslameter is LOCK :

```
Teslameter LOCK, Field = +0.234865968 T
```

If the Teslameter is not LOCK :

```
Teslameter not LOCK
```

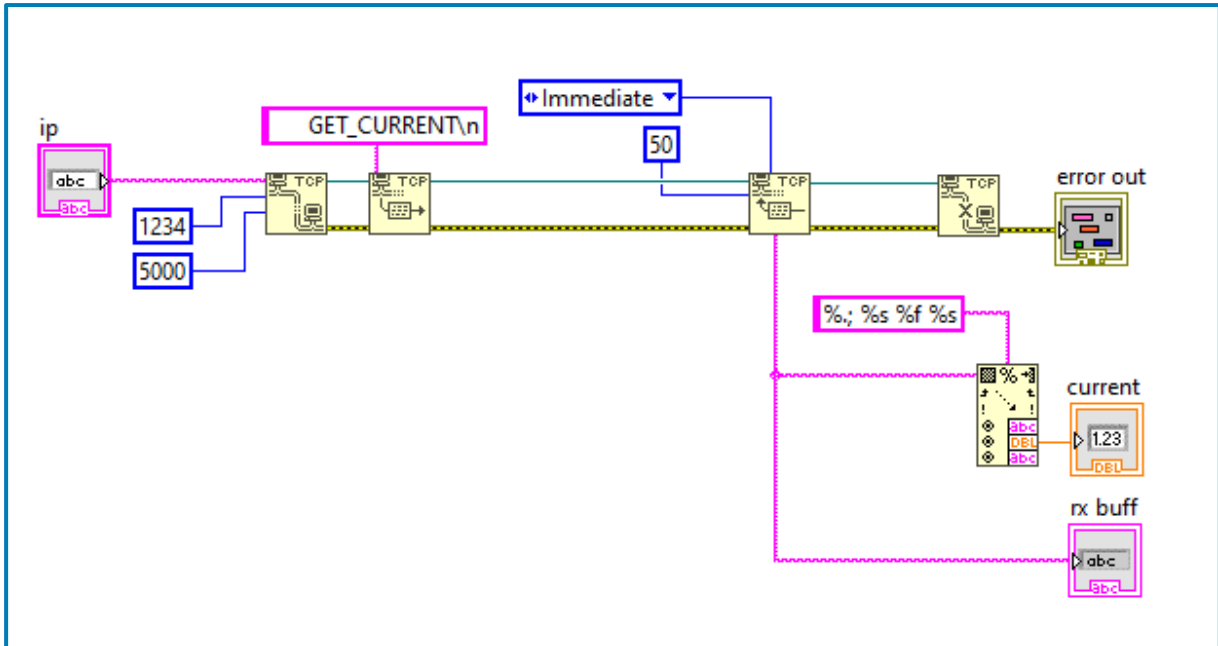
### LabVIEW Example

This program connects to the power supply via the TCP protocol. It requests the last current measurement made by the power supply and displays the result on the front panel of LabVIEW.

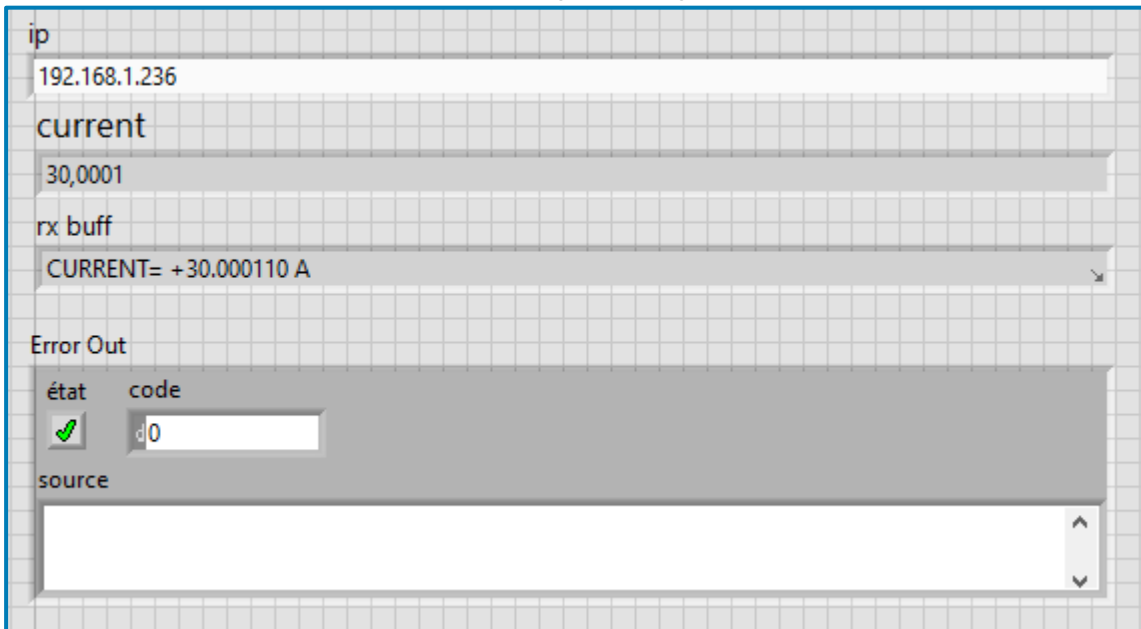
Use the LabVIEW TCP blocks in the block palette:

**Functions > Data Communication > Protocols > TCP.**

Screenshot of the program:



Screenshot of the front panel after running the program :



## Notes

---

A series of horizontal dotted lines for taking notes.



MANUFACTURER OF ELECTROMAGNETS,  
NMR TESLAMETERS,  
POWER SUPPLIES,  
MAGNETIC SYSTEMS

[www.caylar.net](http://www.caylar.net)