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

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: <xxxxx> where xxxxx is the paramater).

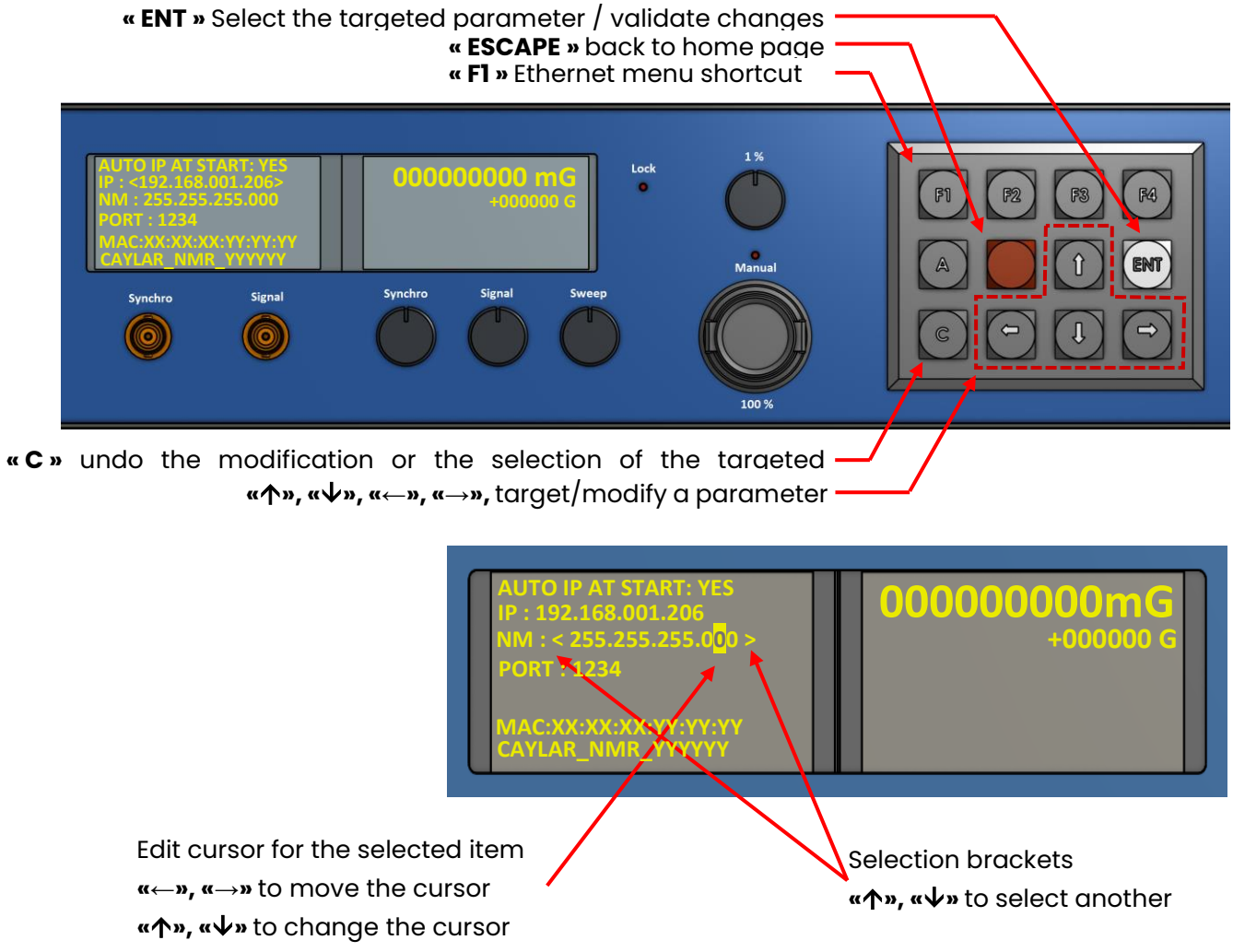
5. To modify the selected parameter, push the « **ENT** » keyboard key. The editable part of the parameter is then highlighted to indicate the position of the edit cursor.

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.
Multiples 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 (expect 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?**	
Response : **CAYLAR_2210_<serial_number>**	Ex : CAYLAR_2210_XXX
Returns the serial number of the Teslameter.	

GET_FIELD_NMR <FORMAT>	Ex : GET_FIELD_NMR 2
Response : **<NMR Field> <Format>**	Ex : +0.234865968 T
<FORMAT>	- 0 : mGauss - 1 : Gauss - 2 : Tesla - 3 : uTesla - 4 : mTesla - if nothing is specified, Teslameter format
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.	

GET_FRQ_NMR	
Response : **<FRQ> Hz**	Ex : 10000001.213636 Hz
Returns the last measurement of the resonant frequency made by the Teslameter. the frequency is returned in Hz.	

GET_FIELD_HALL <FORMAT>	Ex : GET_FIELD_NMR 1
Response : **<HALL Field> <Format>**	Ex : +2348.000 G
<FORMAT>	- 0 : mGauss - 1 : Gauss - 2 : Tesla - 3 : uTesla - 4 : mTesla - if nothing is specified, Teslameter format
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.	

GET_MUX	
Response : **<MUX> (0 to 4)**	Ex : 1
Returns the number of the selected MUX.	

GET_PA_MUX	
Response : **<PA_MUX> (1 to 8)**	Ex : 2
Returns the number of the selected PA_MUX.	

GET_PROBE	
Response : **<PROBE> (1 to 8)**	Ex : 3
Returns the number of the selected PROBE.	

GET_FILTER	
Response : **<FILTER> (1 to 4)**	Ex : 2

Returns the number of the selected FILTER.

GET_FIELD_TYPE	
Response : **< FIELD_TYPE > (1 to 3)**	Ex : 1
< FIELD_TYPE >	- 1 : MEDIUM FIELD - 2 : LOW FIELD - 3 : HIGH FIELD
Returns the field type currently set on the Teslameter.	

GET_SIGNAL	
Response : **<SIGNAL> (0 to 100)**	Ex : 50
Returns the value of the signal currently set on the Teslameter.	

GET_SWEEP	
Response : **<SWEEP> (0 to 100)**	Ex : 50
Returns the value of the sweep currently set on the Teslameter.	

GET_LOCK	
Response : **<LOCK> (0 or 1)**	Ex : 0
Returns the status of the lock. 0 (if not lock) / 1 (if lock)	

GET_MODE	
Response : **<MODE>**	Ex : 3
<MODE>	- 1 : MANUAL SEARCH - 2 : AUTO SEARCH - 3 : HALL TRACKING
Returns the actual MODE set on the Teslameter.	

GET_FIELD_SEARCH <FORMAT>	Ex : GET_FIELD_SEARCH 2
Response : **<Field search value> <Format>**	Ex : +0.234865968 T
<FORMAT>	- 0 : mGauss - 1 : Gauss - 2 : Tesla - 3 : uTesla - 4 : mTesla - if nothing is specified, Teslameter format
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.	

GET_MIN_PROBE <FORMAT>	Ex : GET_MIN_PROBE 2
Response : **<min probe field > <Format>**	Ex : +0.1600000 T
<FORMAT>	- 0 : mGauss - 1 : Gauss - 2 : Tesla - 3 : uTesla

	- 4 : mTesla - if nothing is specified, Teslameter format
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.	

GET_MAX_PROBE <FORMAT>	Ex : GET_MAX_PROBE 2
Response : **<max probe field > <Format>**	Ex : +0.8000000 T
<FORMAT>	- 0 : mGauss - 1 : Gauss - 2 : Tesla - 3 : uTesla - 4 : mTesla - if nothing is specified, Teslameter format
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.	

GET_FIELD_FORMAT	Ex : GET_FIELD_FORMAT
Response : **<Format>**	Ex : 2
<FORMAT>	- 0 : mGauss - 1 : Gauss - 2 : Tesla - 3 : uTesla - 4 : mTesla
Returns the format used to display the field on the Teslameter	

GET_NMR_SIGNAL	Ex : GET_NMR_SIGNAL
Response : **<500 bytes>** **READ_OK**	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.	

SET_MODE <MODE>	Ex: SET_MODE 3
Response: **SET_MODE_OK <MODE>**	Ex: SET_MODE_OK 3
<MODE>	- 1 : MANUAL SEARCH - 2 : AUTO SEARCH - 3 : HALL TRACKING
Response in case of bad argument <MODE>: **BAD_ARG**	
Response in case of overrange in < MODE > argument: **OVERRANGE**	
Response in case the gaussmeter cannot change the MODE: **GAUSSMETER BUSY**	
Allows the user to select the MODE to be used to search the signal.	

SET_MUX <MUX>	Ex: SET_MUX 1
Response: **SET_MUX_OK <MUX>**	Ex: SET_MUX_OK 1
<MUX>	The MUX number (0 to 4)
Response in case of bad argument <MUX> : **BAD_ARG**	
Response in case of overrange in <MUX> argument : **OVERRANGE**	
Response in case the gaussmeter cannot change the MUX : **GAUSSMETER BUSY**	
Allows the user to select the MUX to be used to perform the measurement. The actual PA_MUX and PROBE doesn't change.	

SET_PA_MUX <PA_MUX>	Ex: SET_PA_MUX 2
Response: **SET_ PA_MUX_OK < PA_MUX >**	Ex: SET_ PA_MUX_OK 2
<PA_MUX>	The PA_MUX number (1 to 8)
Response in case of bad argument <PA_MUX> : **BAD_ARG**	
Response in case of overrange in <PA_MUX> argument : **OVERRANGE**	
Response in case the gaussmeter cannot change the PA_MUX : **GAUSSMETER BUSY**	
Allows the user to select the PA_MUX to be used to perform the measurement. The actual MUX and PROBE doesn't change.	

SET_PROBE <PROBE>	Ex: SET_PROBE 1
Response: **SET_PROBE_OK <PROBE>**	Ex: SET_PROBE_OK 1
<PROBE>	The probe number (1 to 8)
Response in case of bad argument <PROBE> : **BAD_ARG**	
Response in case of overrange in <PROBE> argument : **OVERRANGE**	
Response in case the gaussmeter cannot change the probe : **GAUSSMETER BUSY**	
Allows the user to select the probe to be used to perform the measurement. The actual MUX and PA_MUX doesn't change.	

SET_FILTER <FILTER>	Ex: SET_FILTER 2
Response: **SET_FILTER_OK <FILTER>**	Ex: SET_FILTER_OK 2
<FILTER>	The FILTER number (1 to 4)
Response in case of bad argument <FILTER> : **BAD_ARG**	
Response in case of overrange in <FILTER> argument : **OVERRANGE**	
Response in case the gaussmeter cannot change the FILTER : **GAUSSMETER BUSY**	
Allows the user to select the FILTER to be used to perform the measurement.	

SET_FIELD_TYPE <FIELD_TYPE>	Ex: SET_FIELD_TYPE 1
Response: **SET_FIELD_TYPE_OK <MODE>**	Ex: SET_FIELD_TYPE_OK 1
<FIELD_TYPE>	- 1 : MEDIUM FIELD - 2 : LOW FIELD - 3 : HIGH FIELD
Response in case of bad argument <FIELD_TYPE> : **BAD_ARG**	
Response in case of overrange in <FIELD_TYPE> argument : **OVERRANGE**	
Response in case the gaussmeter cannot change the FIELD_TYPE : **GAUSSMETER BUSY**	
Allows the user to select the FIELD_TYPE to be used to search the signal.	

SET_SIGNAL <SIGNAL>	Ex: SET_SIGNAL 35
Response: **SET_SIGNAL_OK <SIGNAL>**	Ex: SET_SIGNAL_OK 35
<SIGNAL>	The SIGNAL value (0 to 100)
Response in case of bad argument <SIGNAL>: **BAD_ARG**	
Response in case of overrange in < SIGNAL > argument: **OVERRANGE**	
Response in case the gaussmeter cannot change the SIGNAL: **GAUSSMETER BUSY**	
Allows the user to change the value of the SIGNAL.	

SET_SWEEP <SWEEP>	Ex: SET_SWEEP 62
Response: **SET_ SWEEP_OK <SWEEP>**	Ex: SET_ SWEEP_OK 62
<SIGNAL>	The SIGNAL value (0 to 100)
Response in case of bad argument < SWEEP >: **BAD_ARG**	
Response in case of overrange in < SWEEP > argument: **OVERRANGE**	
Response in case the gaussmeter cannot change the SWEEP: **GAUSSMETER BUSY**	
Allows the user to change the value of the SWEEP.	

SET_FIELD_SEARCH <FIELD> <Format>	Ex : SET_FIELD_SEARCH 0.234865968 T
Response : **<Field search value> <Format>**	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>	- 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**	
Response in case of overrange in < FIELD > argument: **OVERRANGE**	
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 gaussmeter to "Digital" mode.	

SET_FIELD_FORMAT <Format>	Ex : SET_FIELD_FORMAT 2
Response : **<Format>**	Ex : SET_FIELD_FORMAT_OK 2
<FORMAT>	- 0 : mGauss - 1 : Gauss - 2 : Tesla - 3 : uTesla - 4 : mTesla
Set the format used to display the field on the Teslameter.	

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 :

```python
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
```

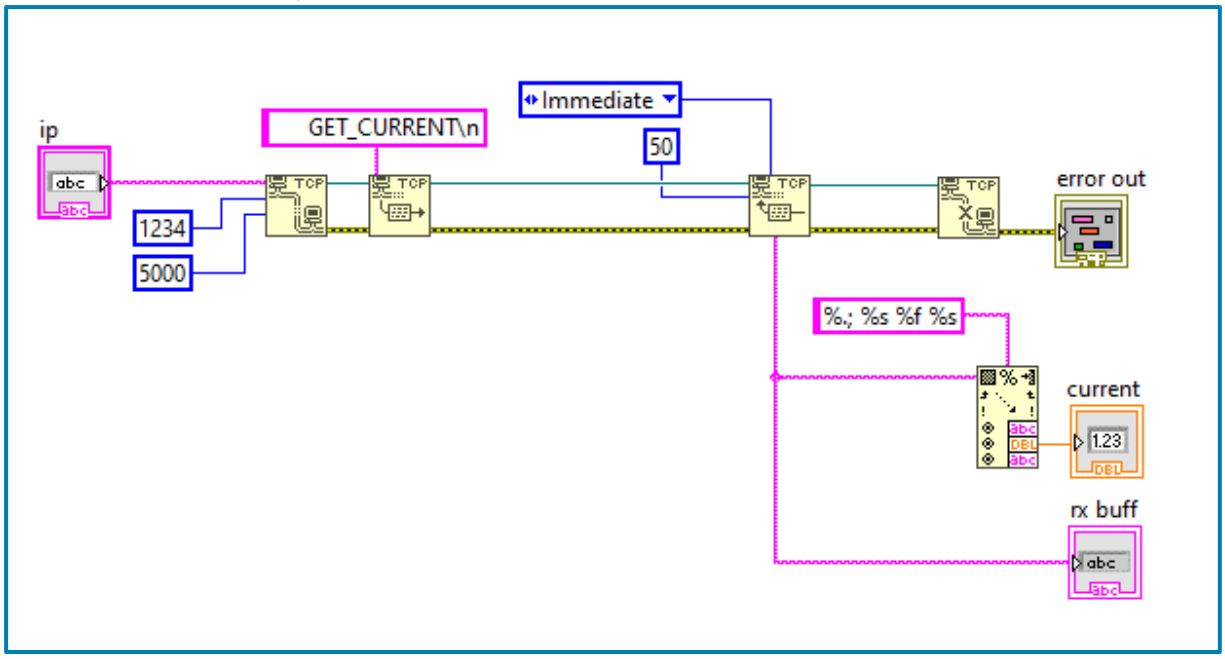
LabVIEW Example

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.

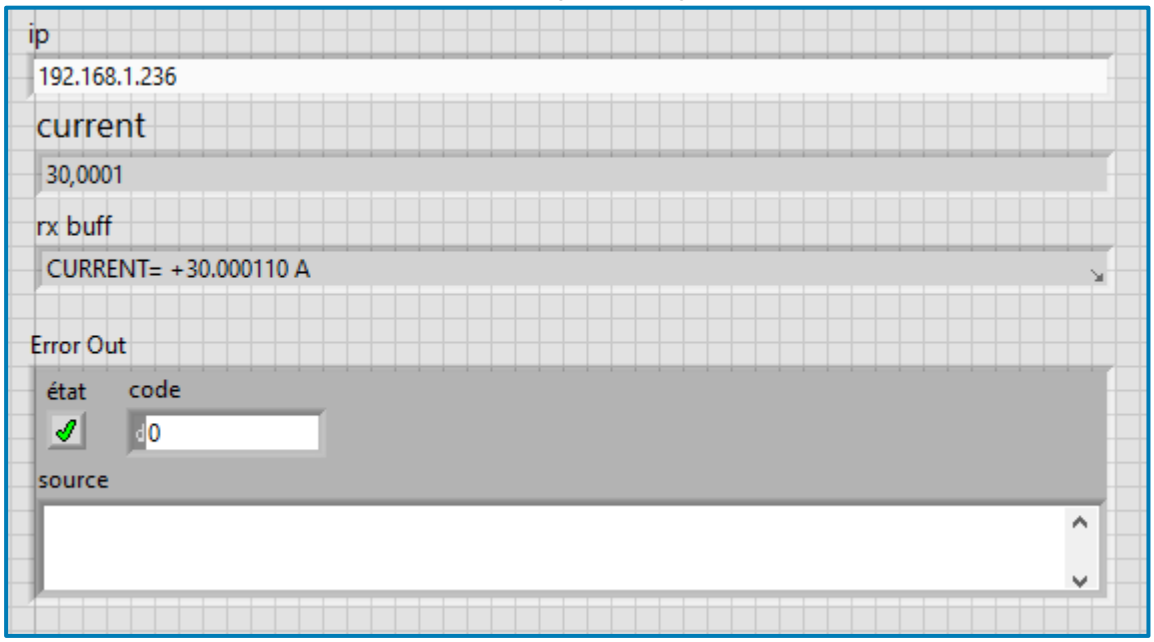
Use the LabVIEW TCP blocks in the block palette:

Functions > Data Communication > Protocols > TCP.

Screenshot of the program:



Screenshot of the front panel after running the program :



Notes

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

www.caylar.net