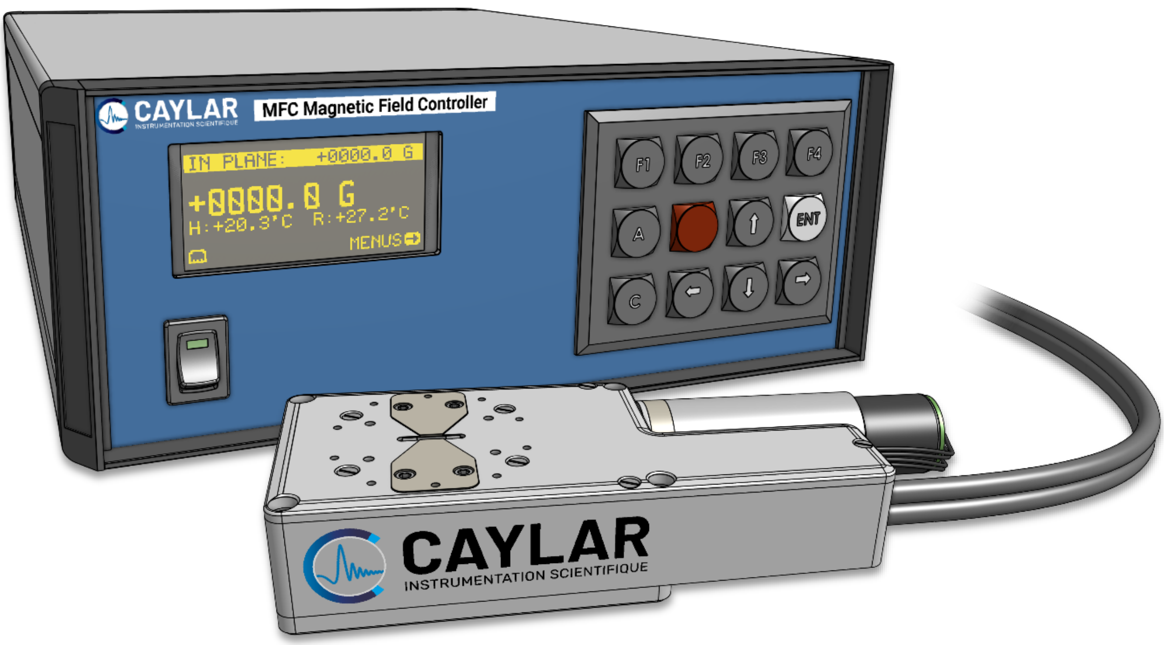
Magnetic Field System for Atomic Force Microscopy



USER MANUAL

Magnetic Field Controller and Module V24

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

MFC5002-016 to 021

December 2024
Rev 24.01

Summary

Introduction

The aim of this system is to generate a variable bipolar magnetic field in two plane configuration modes: IN and OUT of plane. For the in-plane mode, the magnetic field axis orientation is mainly included in the user's sample surface. For the out-plane mode, the magnetic field axis orientation is mainly normal to the user's sample surface.

The system can be controlled fully manually from the front panel (keyboard + oled screen) or fully remotely via ethernet, USB or RS232 communication protocols.

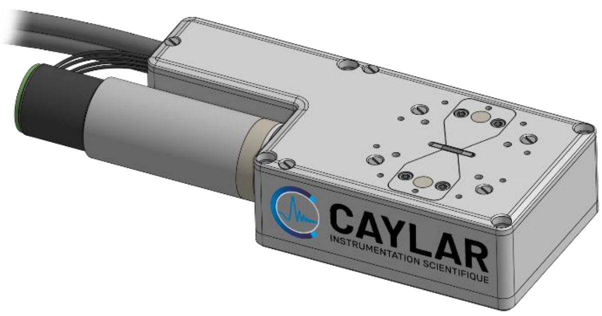
It has a field closed loop control using an internal hall sensor for field measurement but it can also be used in open loop. All the system is fully open so you can achieve more complex regulation by controlling the motor by yourself thank to the internal or an external sensor for example.

The system consists of two parts:

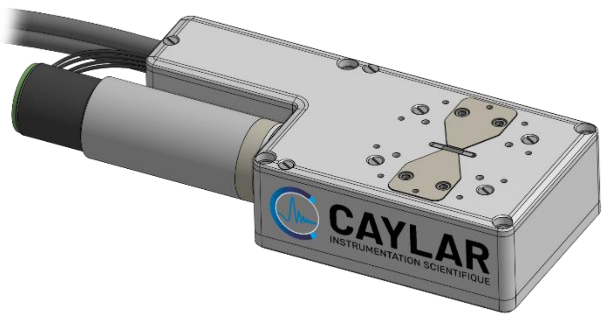
- A **M**agnetic **F**ield **C**ontroller (**MFC**) which contain all the electronic for supplies, communication and field module control.

- A **M**agnetic **F**ield **M**odule (**MFM**) which contain a stepper motor and a permanent magnet in order to generate the magnetic field. This module has two configurations manually settable by the user: IN and OUT of plane (two set of poles). The module includes a ~ 3 meters connection cable to the controller.

Out-plane version:



In-plane version:



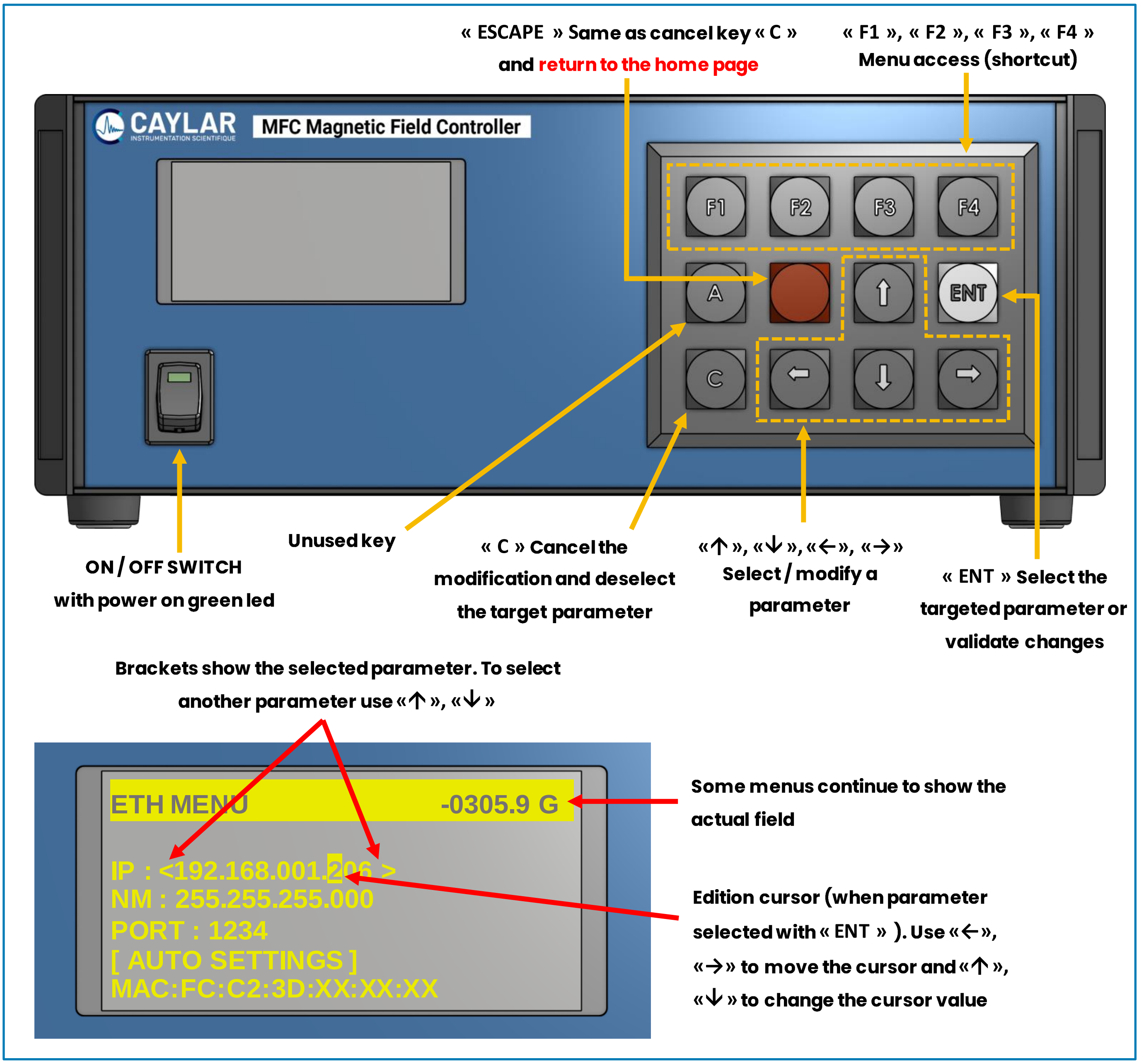
Magnetic Field Controller (MFC) Presentation

The MFC is a 2U rack cabinet which contains all the electronic controls.

You can control and monitoring all the system manually from its front panel keyboard and oled screen. Each change made in MFC menus are then permanently saved on the controller SD card. You will therefore find the same parameters after power recycle.

From its rear panel you have access to 3 communication connectors (ethernet, RS232 and serial via USB) for remote control.

a) Front panel – Keyboard and screen

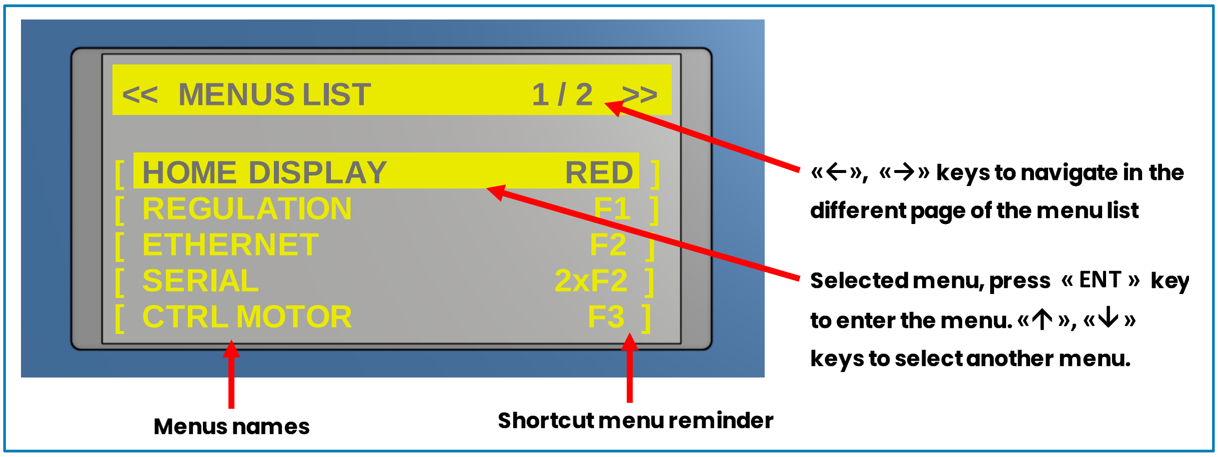


MFC front panel view / example of menu displayed on the oled screen

b) Menus list

The various menus are explained in their associated sections of this documentation. A list of the MFC menus is available from the home page (**« ESCAPE »** key) by pressing the **«→»** key. Then use **«←»** and **«→»** keys to navigate in the different page of the menus list or return to the home page. You can selec a menu with **«↑»**, **«↓»** keys and enter the menu with **« ENT »** key.

The menu list also reminds you the shortcuts associated with each menu. The shortcuts are given at the right of the menu names.



Oled screen view of MENUS LIST page 1 / 2 on MFC front panel

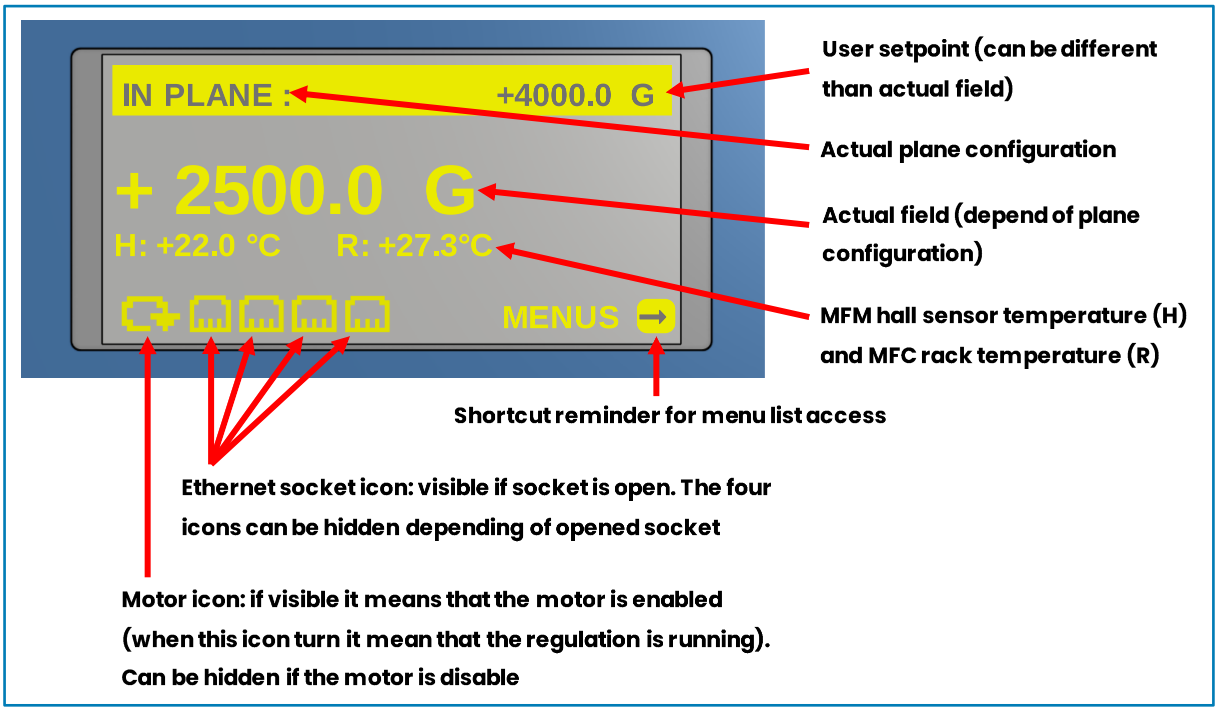
List of MFC menus and pages:

Menu name	Function	Shortcut
Home page	Displays the overall system status (field, setpoint, temperature, ethernet socket connected….)	**ESCAPE**
REGULATION	Control the regulation manually (field setpoint, plane configuration…)	**F1**
OUTP PARAMS	Change the regulation parameters for the out-plane configuration	**F1 x2**
INP PARAMS	Change the regulation parameters for the in-plane configuration	**F1 x3**
ETHERNET	Change the ethernet parameters	**F2**
SERIAL	Change the serial parameters	**F2 x2**
CTRL MOTOR	Control the motor manually	**F3**
DIVERS CONFIG	Change other parameters (field unit, sound…)	**F4**
MFC INFOS	Display MFC / MFM information's	**F4 x2**

CAYLAR
INSTRUMENTATION SCIENTIFIQUE

c) Home page

The home page displays the overall system status. It's the startup page. You can access it from other menus with the **« ESCAPE »** key (red).

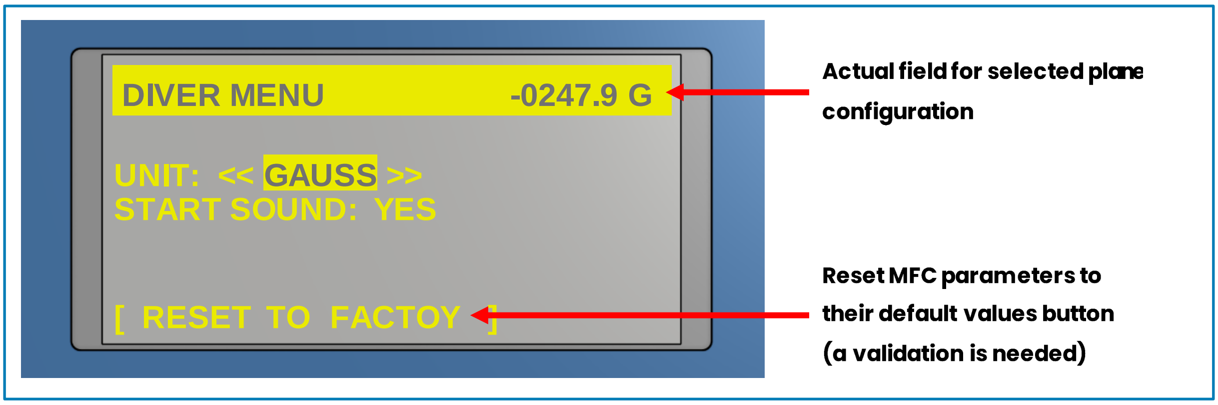


Oled screen view of the home page on MFC front panel

d) Diver menu

To access the "DIVER MENU", press the **« F4 »** key. On this menu you can change the MFC display unit (Gauss / mTesla / Tesla), enable or disable the starting sound of the system and reset the various parameter of the MFC to their default value.

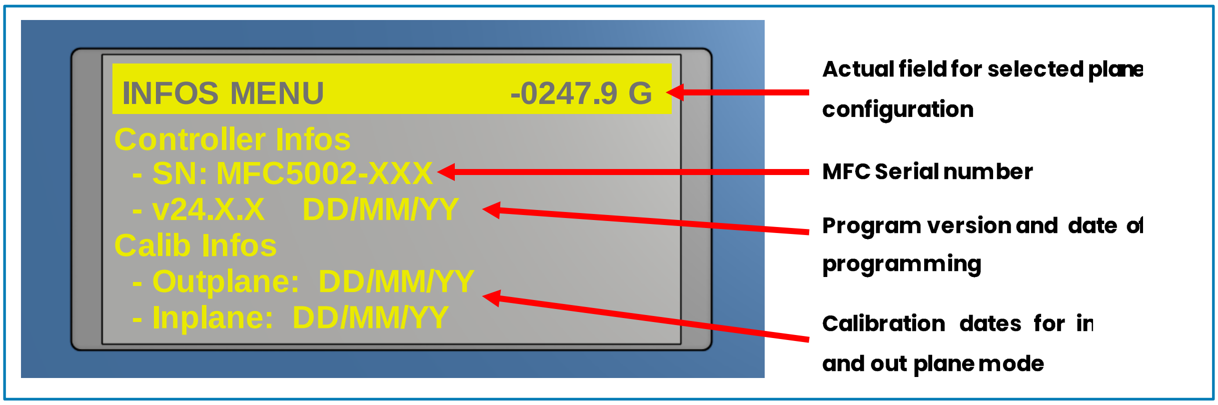
NOTE: the user magnetic field unit is only for screen display of the MFC. The remote commands and their returned values are in Gauss.



Oled screen view of DIVER MENU on MFC front panel

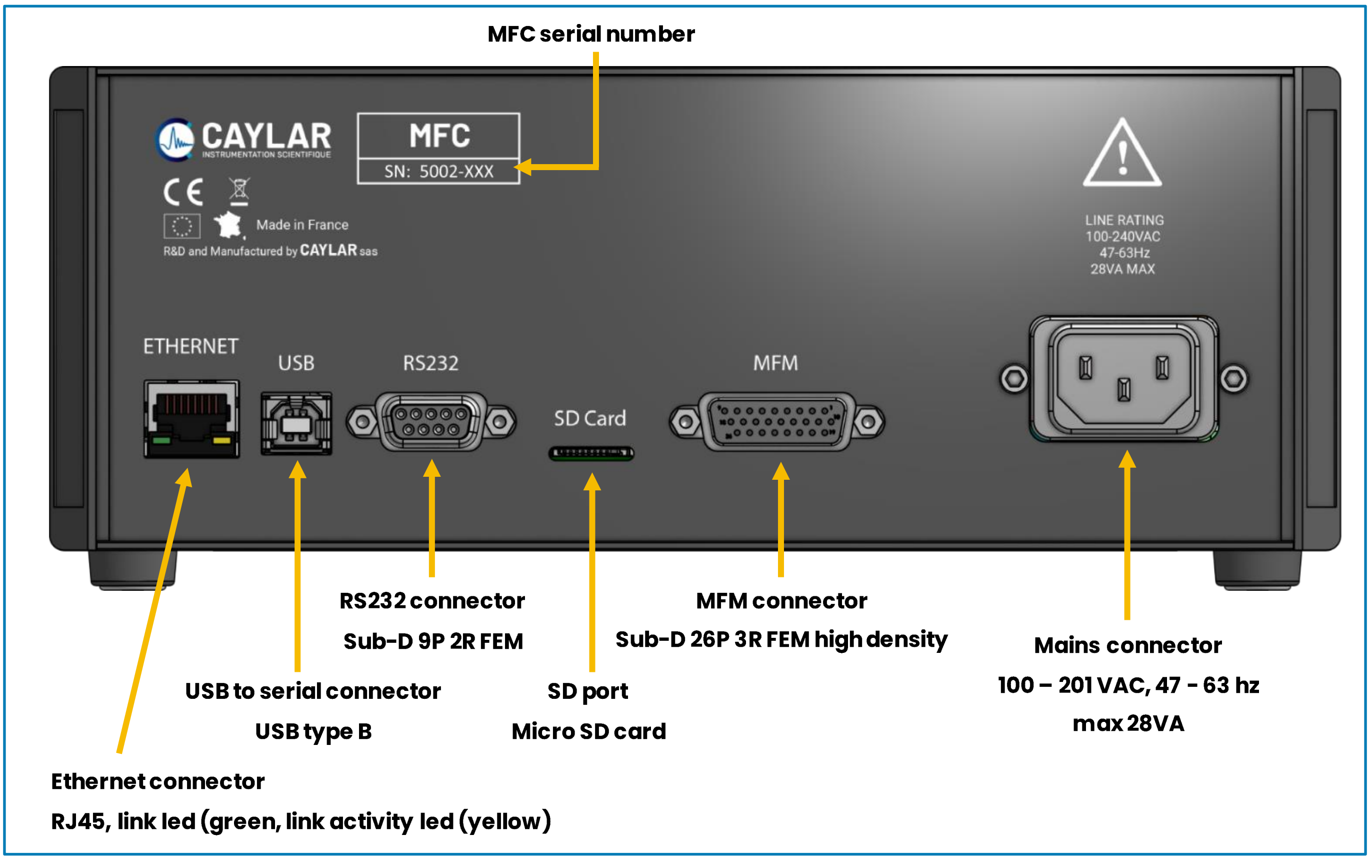
e) Infos menu

To access the "INFOS MENU", press the **«F4»** key twice. This menu shows you the serial number of the MFM / MFC and the program date and revision.



Oled screen view of INFOS MENU on MFC front panel

f) Rear panel – Communication connectors

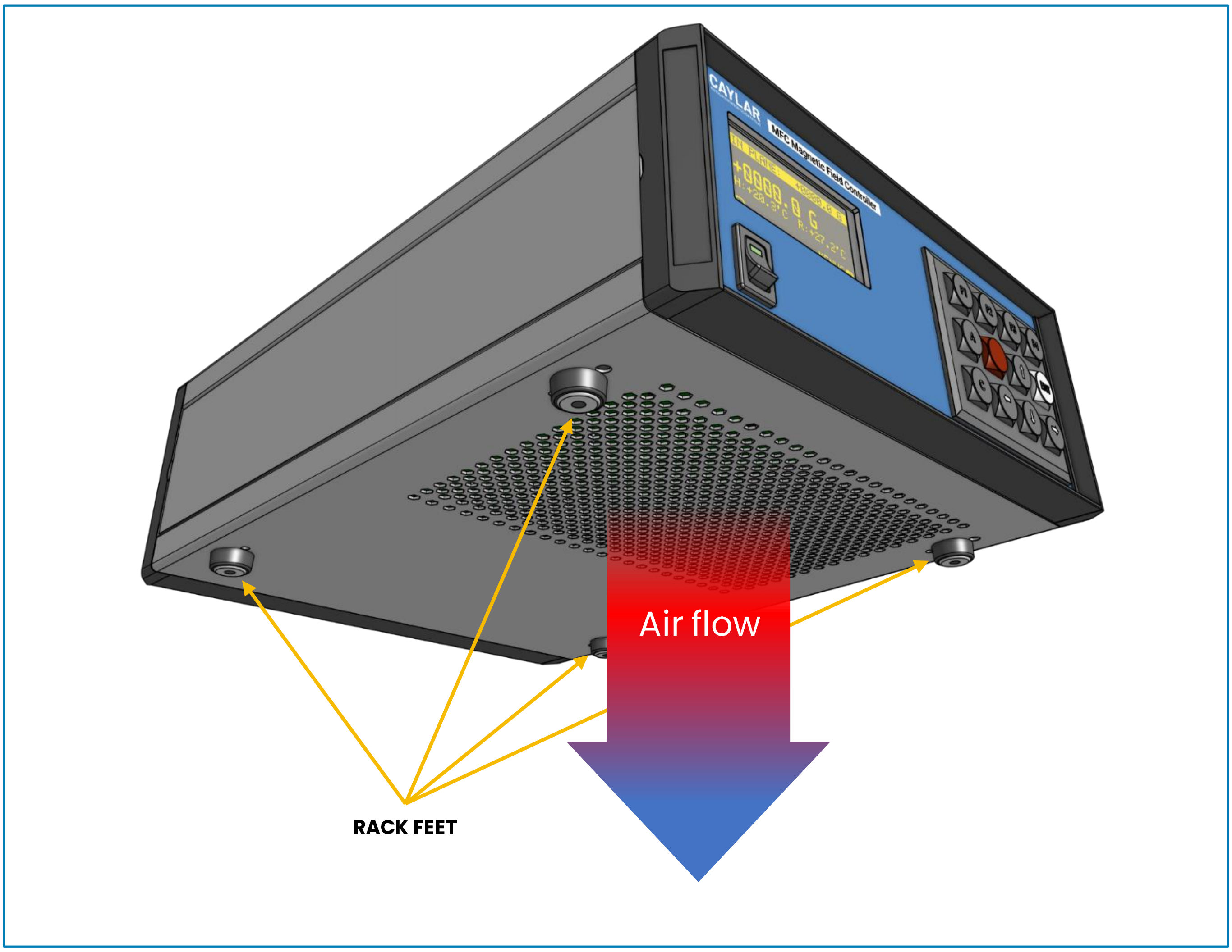


MFC rear panel view

NOTE: the SD card is used to save the user configuration and calibration datas. It is also used for program update. Don't remove or modify without instructions from Caylar.

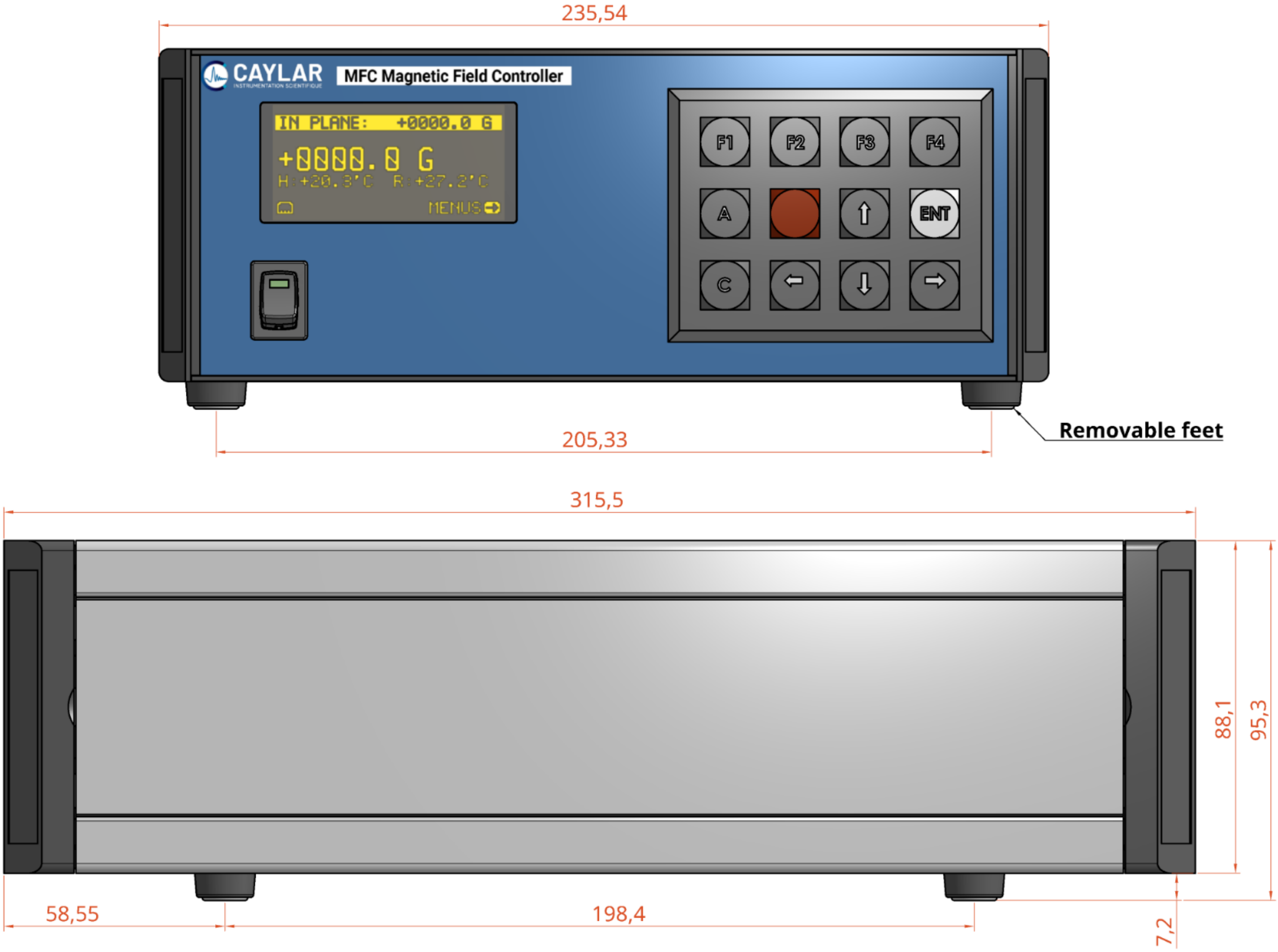
g) Rack Air flow

The MFC include a fan which extracts the internal hot air through the bottom grid of the rack. The rack can be placed on a flat surface. Its feet allow it to have a sufficient space with the holding surface and the grid so the air can move freely. If the rack feet are removed, be careful to leave the fan outlet free enough to allow sufficient air flow.



MFC bottom view – Air flow

h) Rack dimensions



Magnetic Field Module (MFM) Presentation

The MFM is the field generator placed on the microscope just under the user sample.

It uses a bipolar permanent magnet connected to a stepper motor to generate the variable magnetic field. The magnet is placed at the middle of a ferromagnetic core which have a U-shape. At the extremities of the U-shape are placed two matched poles. There is two different shape of poles depending of the field orientation configuration: IN or OUT of plane. The rest of the module structure is composed of non-magnetic parts.

To change the magnetic field, the permanent magnet is rotated by the stepper motor to orient more or less its magnetic field lines with the ferromagnetic core.

The produced magnetic field is read back by an internal hall sensor. It permits to achieve a closed loop control of the magnetic field.

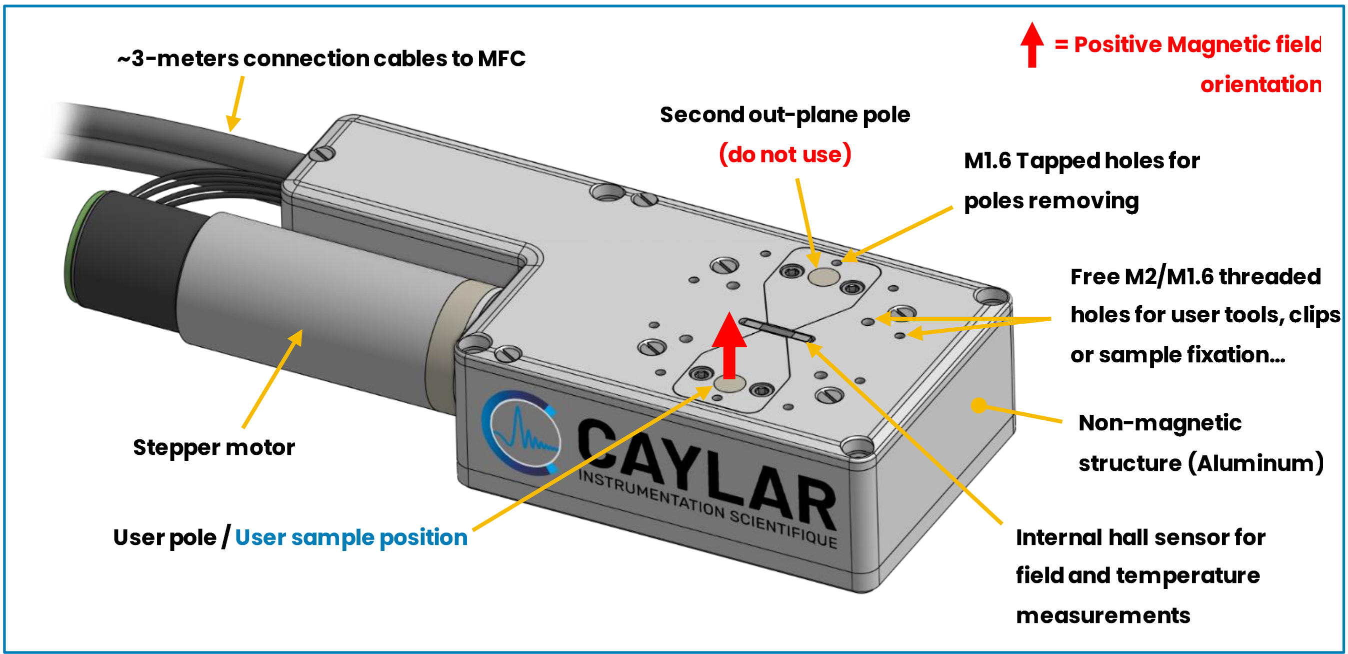
The module also uses the internal hall sensor to monitoring the MFM temperature. **NOTE:** No temperature calibration is performed by Caylar. The temperature is only given as an indication to the user.

Free M2 and M1.6 holes are available on the module surface around the poles for user tools, clips or samples fixation (see module dimension section).

a) Out of plane configuration (OUTP)

For the out of plane configuration (OUTP), the generated magnetic field is oriented perpendicular (normal) to the user sample surface.

The user sample is placed on the user pole surface (used for calibration) and the other pole shouldn't be used.

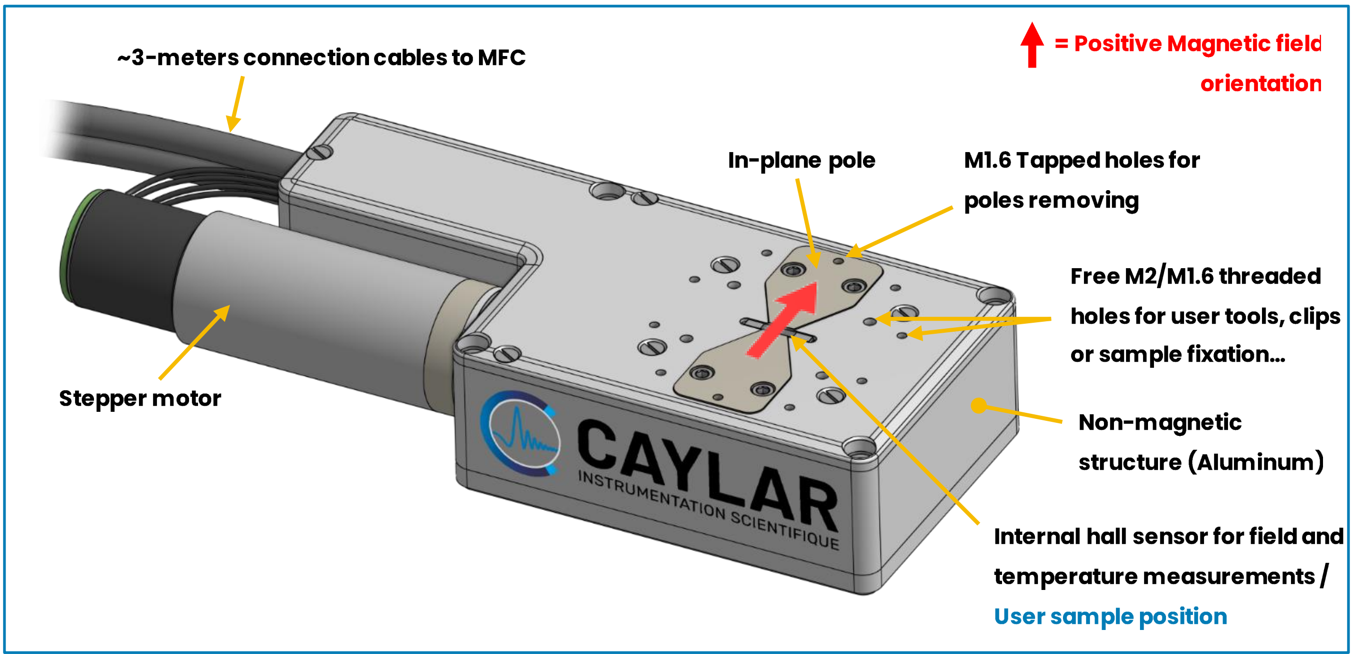


MFM out of plane configuration – Parts presentation

b) In-plane configuration (INP)

For the in-plane configuration (INP), the generated magnetic field is oriented within the user sample surface.

The user sample is placed at the center of the two in plane pole, just above the in-plane hall sensor.



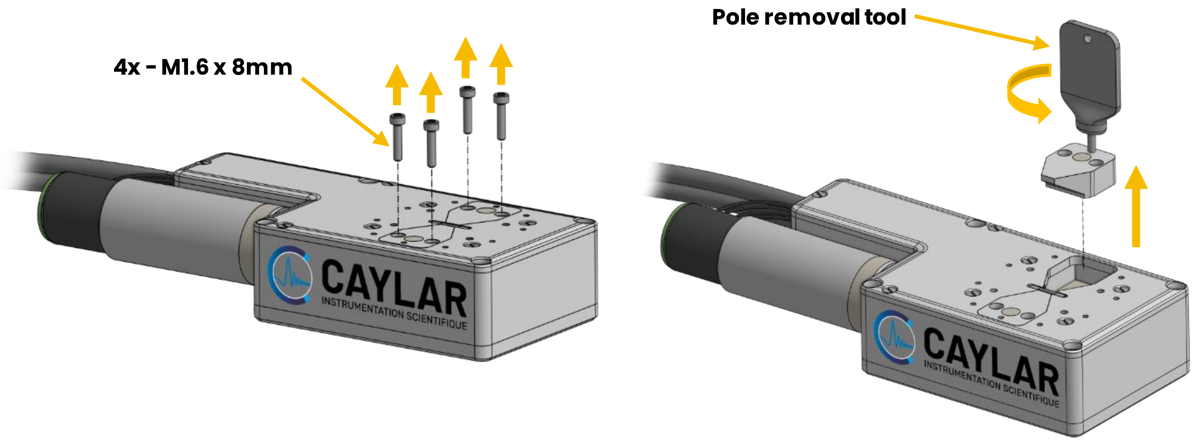
MFM in-plane configuration – Parts presentation

c) Change module plane configuration.

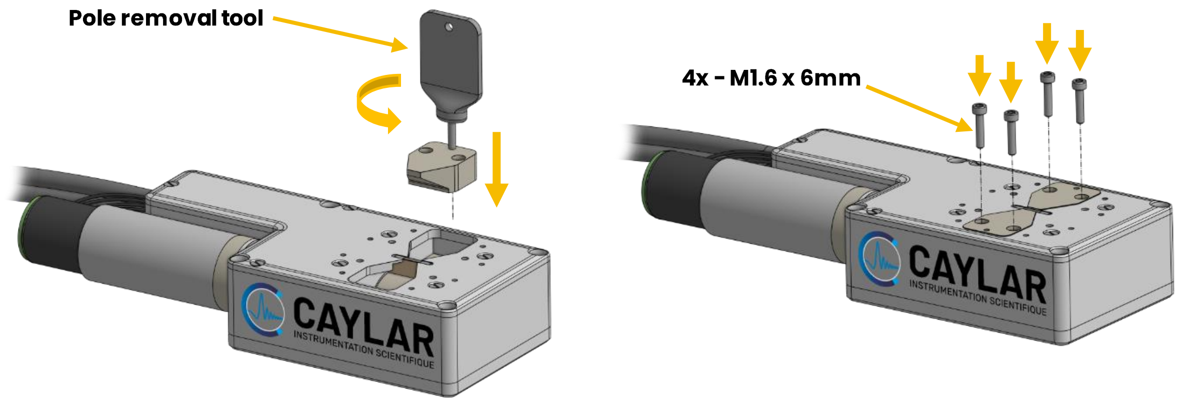
NOTE: Handle the module in a clean environment without metal parts / dust that could be attracted by the magnet of the module. Pay attention to the hall sensor at the center of the MFM when replacing poles.

Step 1: Turn on the MFC with the MFM connected to it and set the magnetic field to zero in order to minimize the attraction force on the poles we want to remove for change. Then turn off the MFC.

Step 2: Remove the four M1.6 poles screws and use the poles removal tool (in the poles box) to help you to remove the poles from their location. The tool needs to be screwed to the pole.

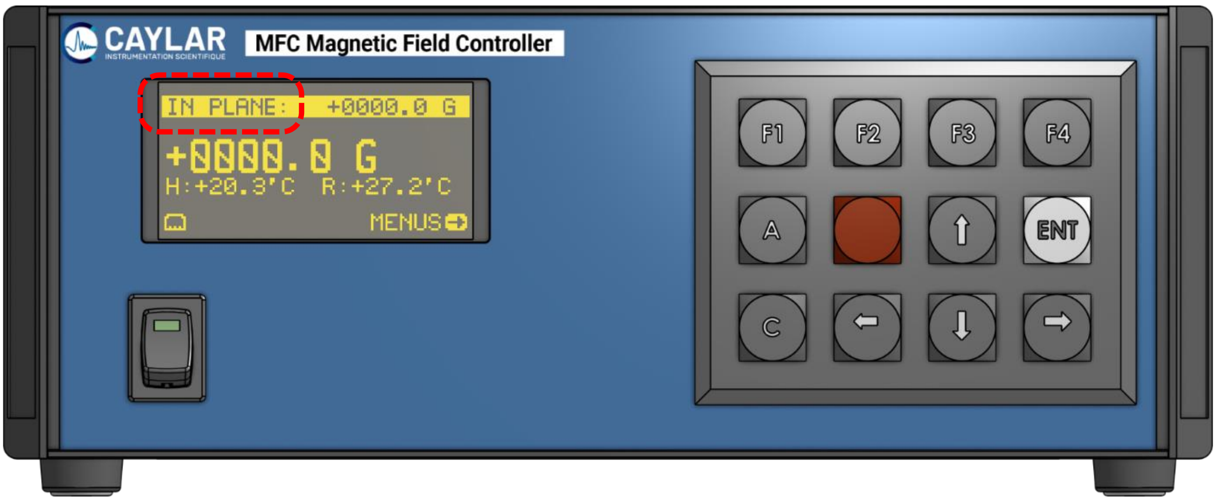


Step 3: Use the poles removal tool to help you to pick and place the new set of poles from the poles box to the MFM. Insert the two new poles on the module with the four M1.6 screws. Tighten the screws but don't overtighten them **(need to be 14.5 N.cm – max 16 N.cm)**. Be very careful when inserting the poles. Do not touch the center pcb sensor.



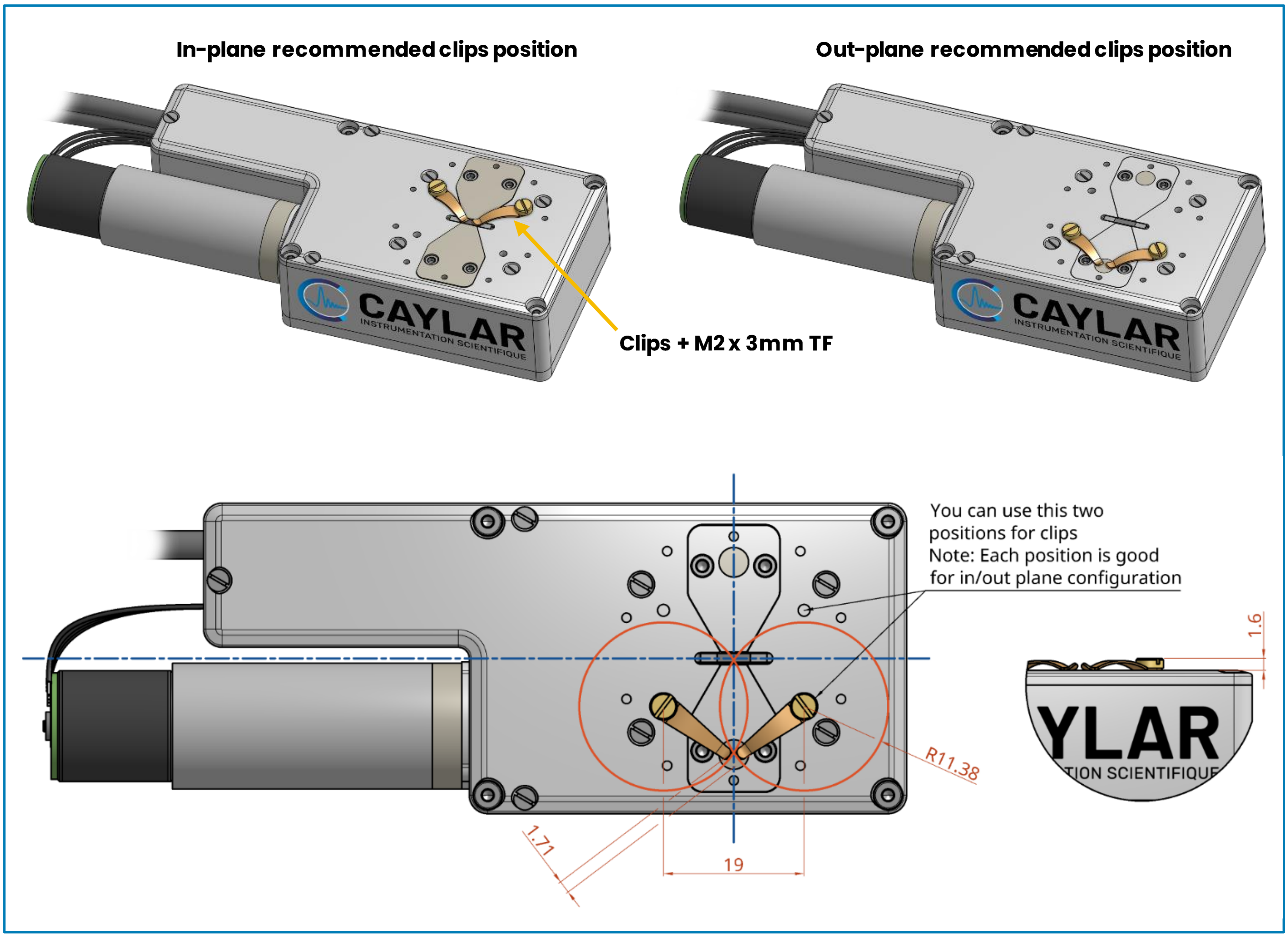
Step 4: Power on the MFC and change the plane configuration manually (REGUL MENU – Regul plane listbox) or remotely (see remote control section).

NOTE: The actual configuration is displayed on the home page of the MFC screen (see image below) or you can check the configuration by remote commands (see remote control section).



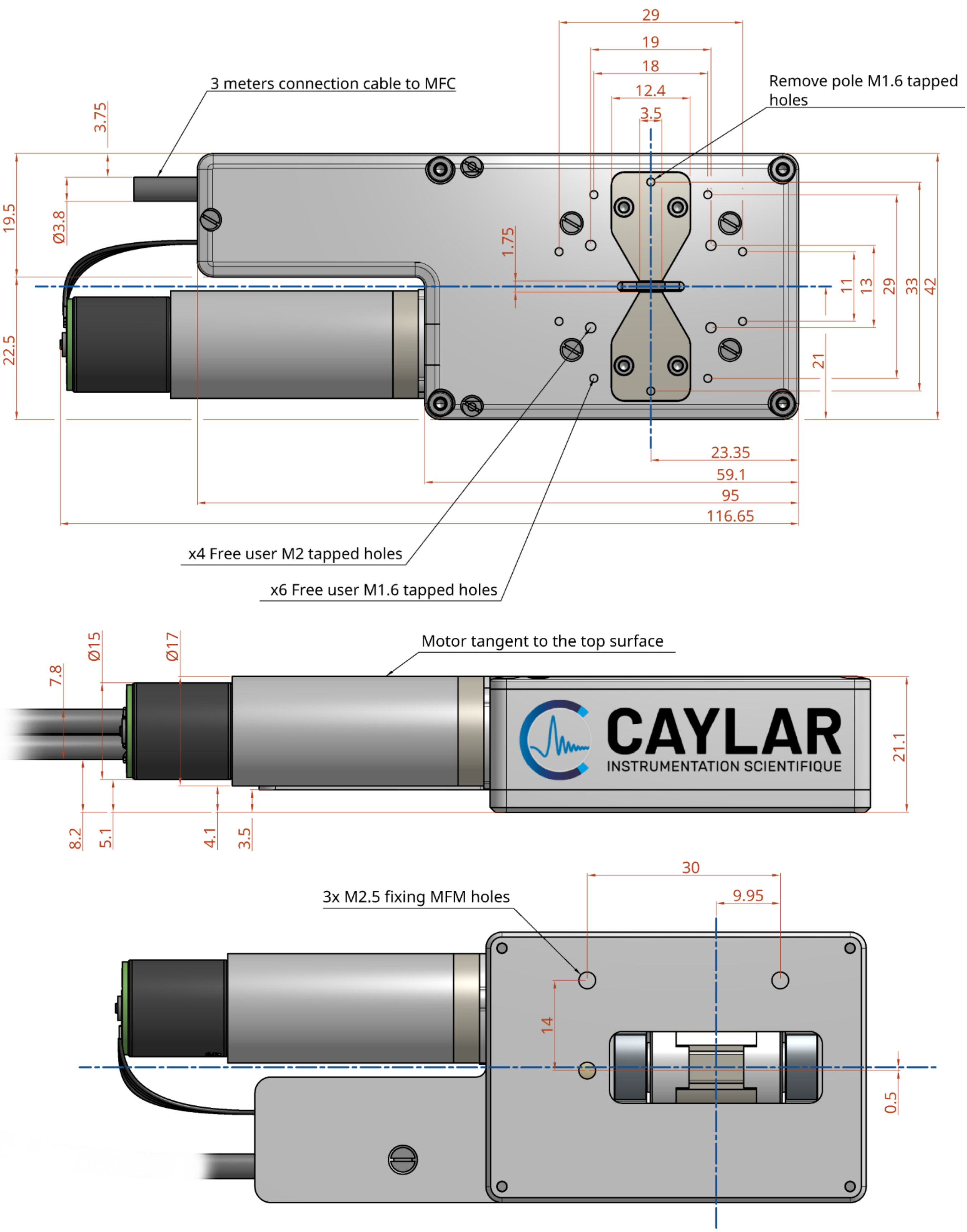
d) Clips for user sample fixation

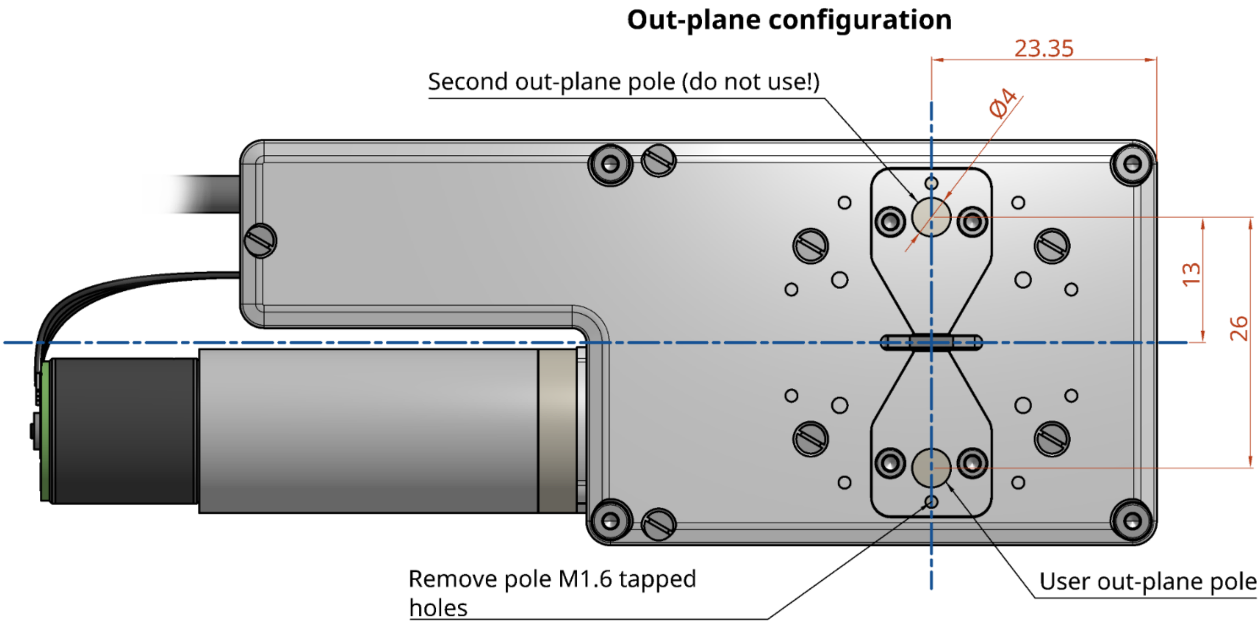
Two clips are provided with the MFM in the pole box to easily fixing your samples. They can be mounted on the M2 tapped holes on the MFM surface around the poles (see image bellow):



MFM Clips presentation

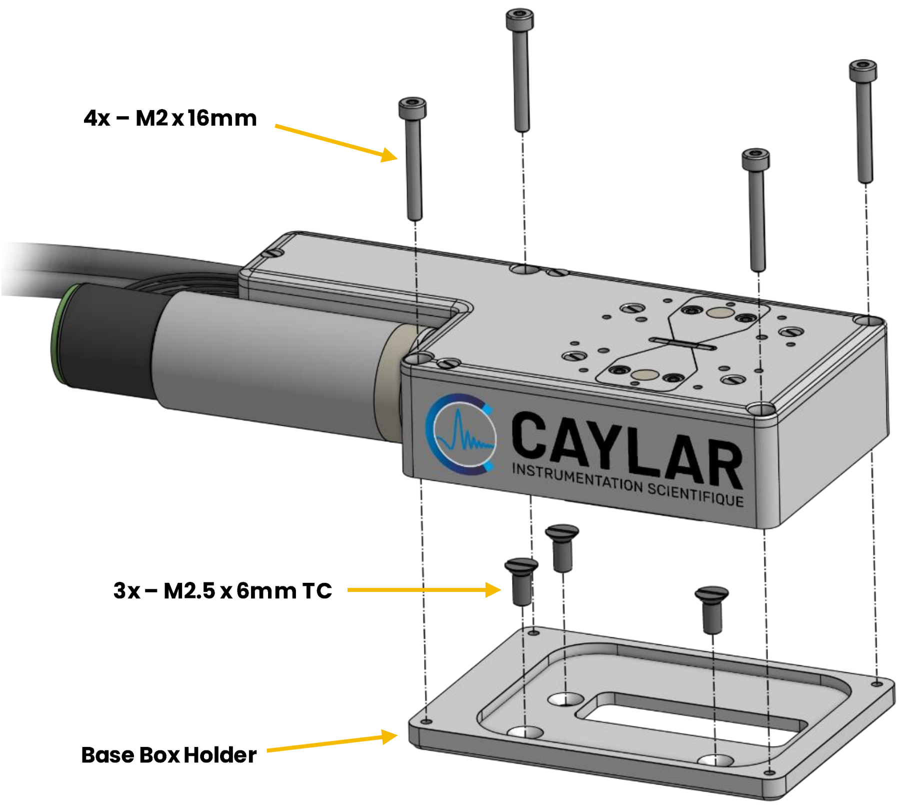
e) Module dimensions drawings





f) Fixing the module

To fix the MFM to an AFM microscope, remove the four M2 x 16mm screws in order to remove the base box holder. Then fix the base box holder with three M2.5 x 6mm TC screws and put back the MFM on it with the four M2 x 16 mm screws.



g) Calibration process

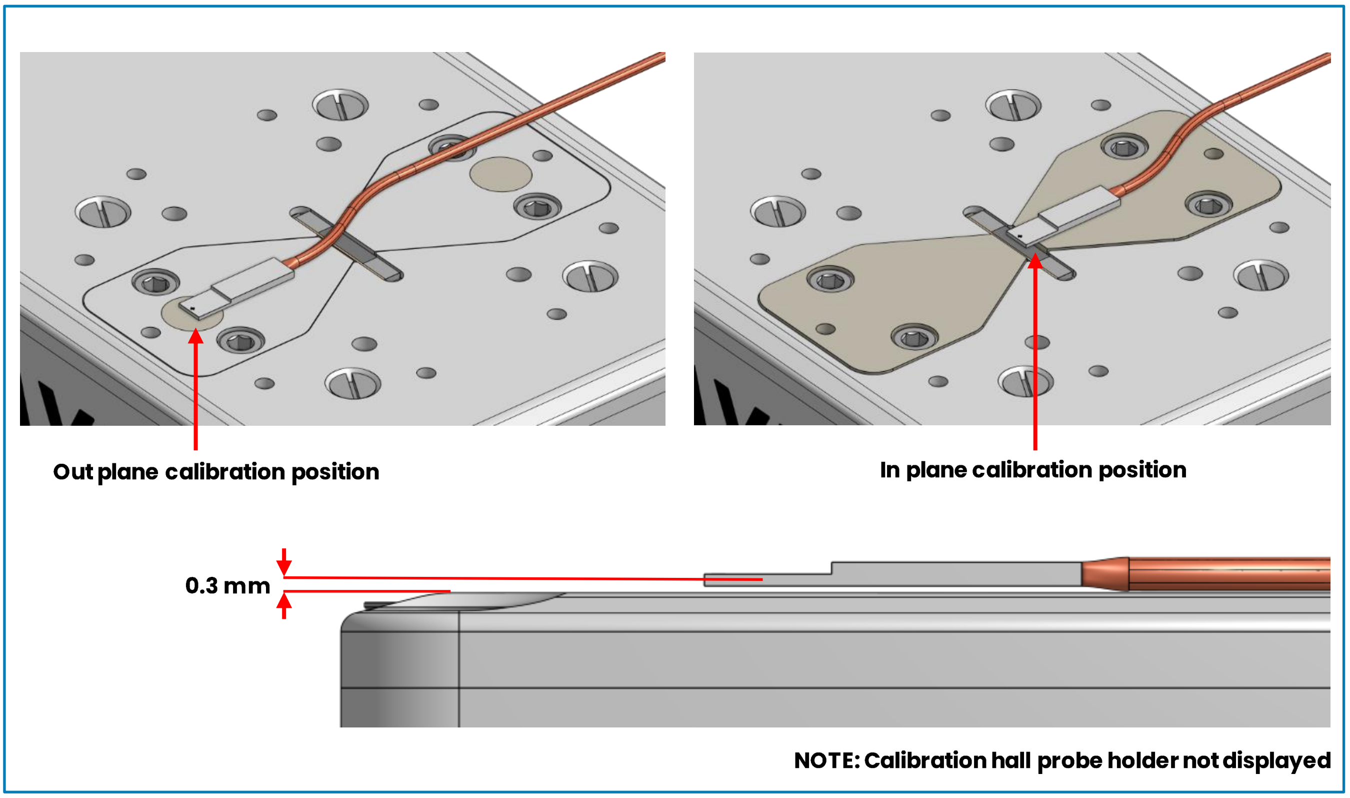
The MFM and MFC are calibrated together. So MFM can be used only with his associated MFC.

The system is calibrated for out-plane and in-plane configuration with an external hall probe reference placed at the two user sample positions: center of the two poles for the in-plane configuration and at the center of the round user poles for the out-plane configuration. The probe sensor is located at **0.3 mm** from the poles surface for in/out of plane configuration (most common user sample thickness). The probe is precisely positioned thanks to a special holder attached to the MFM chassis during calibration. Images of the calibration positions are given on the figure bellow.

The calibration reference instrument is a 3-Axis digital teslameter 3MH3A-0.1%-1T from SENIS with 3-Axis Hall Probe type F3A-03KS02C.

NOTE: Because the external calibration hall probe (user sample position) isn't at the same place as the internal MFM hall probe there are some hysteresis errors that affect the calibration. To minimize these errors we made two different calibrations with a rising and a falling slope of the full field range. At the end we made an average of these two slopes to minimizing hysteresis errors.

After the calibration we keep the calibration hall probe in place and we test the system during more than 4 hours with 150 random setpoints for each module and plane configuration. Calibration datas and test datas are available for each module.



Calibration hall probe position for IN and OUT of plane configuration

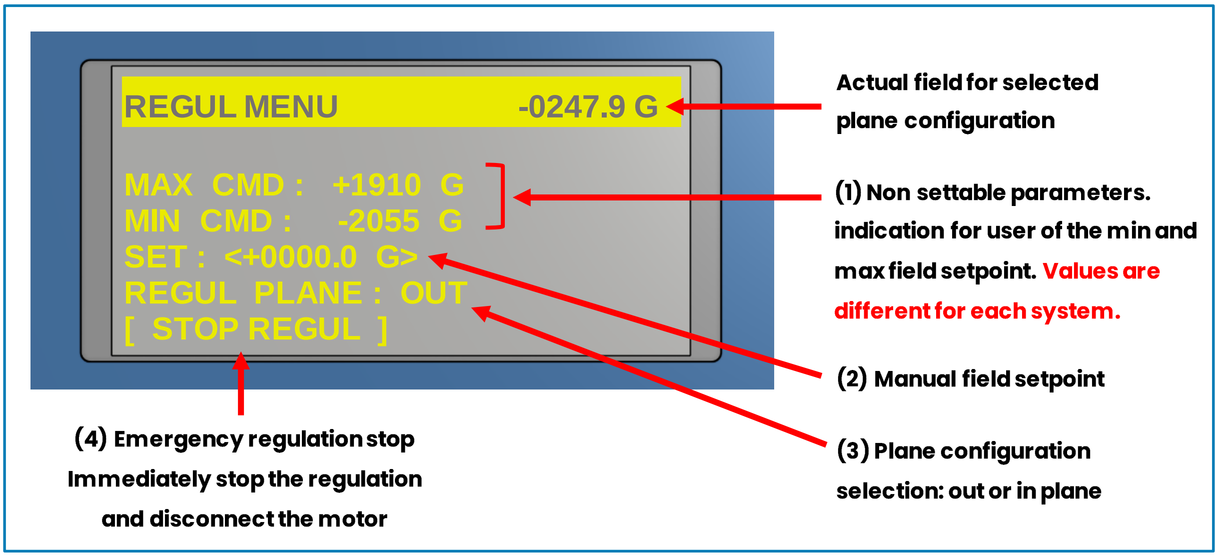
Field regulation operation – Closed loop

a) Main regulation menu – set the field / plane mode

The magnetic field regulation allows the user to set a desired field with an adjustable control of the regulation speed, precision and overshoot. The regulation can be controlled manually or by remote control (see remote control section).

WARNING: Be careful when you often change the field setpoint within short time intervals. When the motor is used it create heat from its driving current. This heat dissipation can affect the permanent magnet inside the MFM (and the user sample) and creates long-term temperature drift of the magnetic field. Temperature measurement of the hall sensor is available to monitor this temperature variations. This allows the user to know if the engine is used too often. Our tests show that a setpoint change every 3 minutes does not affect the module significantly but it depends of various factors as external temperature, external environment, etc.

For manual control press **« F1 »** key to enter the "REGUL MENU":



Oled screen view of REGUL MENU on MFC front panel

For remote control here is a list of remote commands (see remote control section) related to "REGUL MENU":

(1) GET_REG_INP_MAX_SETPOINT
(1) GET_REG_INP_MIN_SETPOINT
(1) GET_REG_OUTP_MAX_SETPOINT
(1) GET_REG_OUTP_MIN_SETPOINT
(2) SET_FIELD
(2) GET_FIELD
(3) SET_REG_PLANE_MODE
(3) GET_REG_PLANE_MODE
(4) SET_REG_STOP (Note: to restart the regulation use the SET_FIELD command with the same setpoint)
(4) GET_REG_STATE

b) Operating principle – Regulation start / stop conditions:

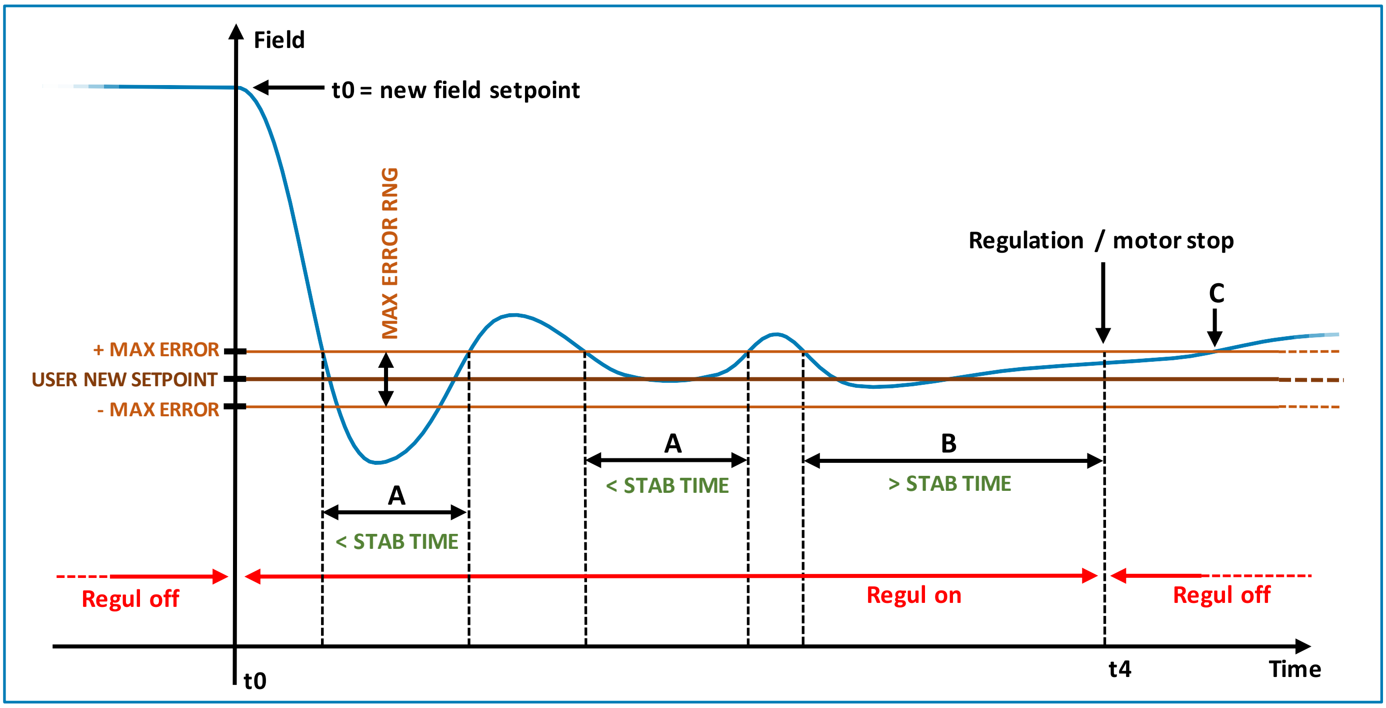
The operating principle of the MFC field regulation is a little bit different than a conventional PID systems. It doesn't drive continuously the module to keep the field constant.

When the user requests a new setpoint, the MFM motor and the MFC regulation loop are activated in order to reach the desired field. When the feedback field enters and stays in a certain interval range around the user field setpoint (**MAX ERROR**) during a certain period of time (**STAB TIME**), the controller fully disconnects the module motor and stop the regulation loop in order to keep the permanent magnet orientation. It means that the controller assume it has reached and stabilized the user setpoint for enough time.

The purpose of fully disconnect the motor and stop the regulation is to reduce the electrical noise / interferences and heat from the motor that can interfere with user measurements during an atomic force macroscope scan for example and also causing magnetic field drift.

The maximum error and stabilization time are settable by the user (see remote control section or manual parameter edition present on this section). These parameters are different for the in-plane and out-plane configuration.

The figure bellow shows an example of field regulation run :



Example of field regulation error evolution and stop conditions

t0 – The user set a new field setpoint. The module motor and regulation start running and the field start to reach the new value.

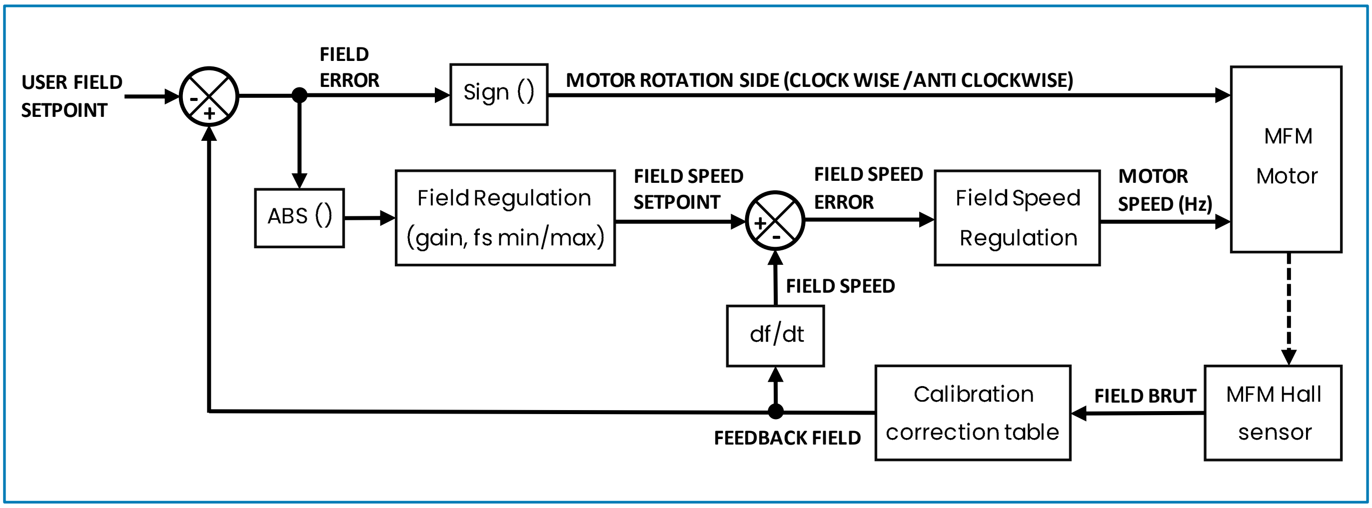
Interval A – The regulation didn't manage to maintain the field in the configured interval range around the user setpoint during the configured stabilization time. Each time the field exit the interval range, the counter of the stabilization time is reset to zero and restart at the next interval range re-entry.

Interval B – The regulation manages to maintain the field in the interval range during more than the configured stabilization time. The controller assumes that it has reached and stabilized the user setpoint. The field regulation is then stopped and the module motor is fully disconnected from the controller at **t4**.

WARNING: C – The field can continue to drift a little bit for multiple reasons like module mechanical deformations due to magnetic field force applied on mobile part and external/module temperatures variations. So, it can exit the interval range after the regulation stop. Since the regulation is stopped, it won't try to restart itself in order to avoid disturbing the user during his measurements. However, the user can monitor the field and restart the regulation if desired by applying the same setpoint.

c) Operating principle – Field regulation:

Magnetic field regulation doesn't directly control the stepper motor of the MFM. It controls a field speed regulation which controls the stepper motor speed. The diagram bellow resume how the field regulation work.



Field regulation block diagram

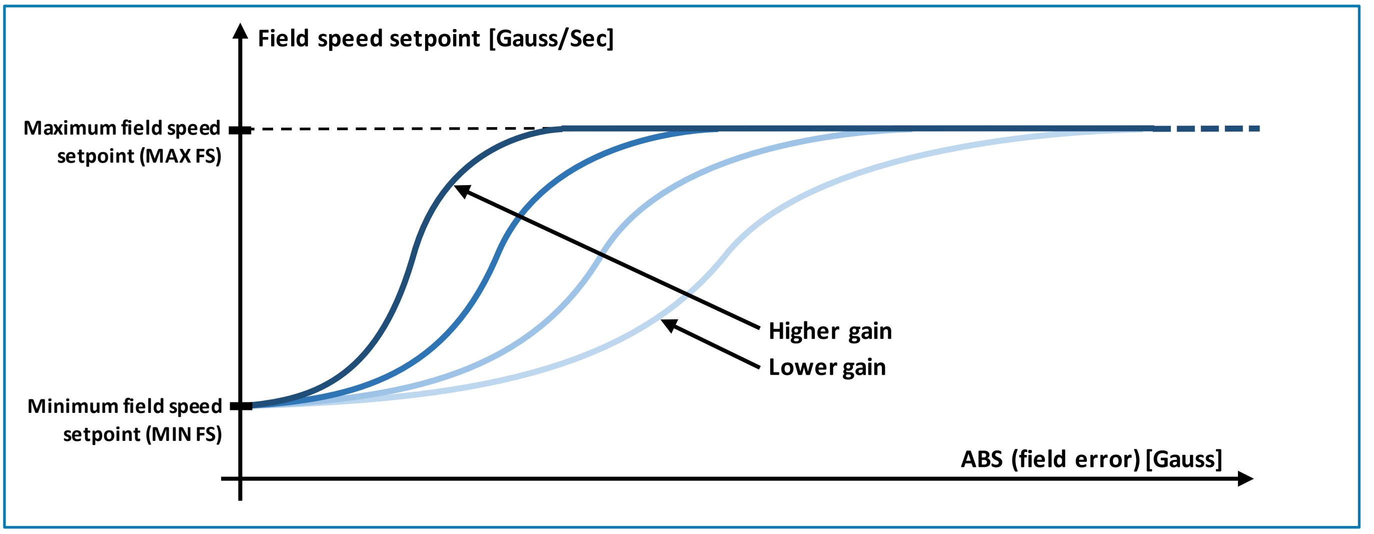
The parameters of the field speed regulation are factory set but the user can adjust the field regulation characteristics thank to three parameters in addition to the stop condition parameters explained in the previous chapter. There are the regulation min/max field speed (FS) and the regulation gain. These parameters are different for in-plane and out-plane configuration.

When the field error is minimal, the speed required by the field regulation to the field speed regulation is then the parameter "min fs". On the contrary, when the error is maximum the requested field speed is then equal to the parameter "max fs". The "gain" parameter is used to define the maximum error from which the requested field speed is "max fs". The gain is similar to the proportional coefficient for a PID regulation but with a little bit of dumping around the regulation error extremum. More we increase the gain parameter, faster the regulation will react to a field error by requiring a higher field speed to the field speed regulation. To minimize the overshoot, it is better to lower the value of this parameter.

Note: Be careful with the "min fs" parameter, the module must have enough speed control at low error to be able to change the field efficiently and quickly if you want a relatively fast response.

It is recommended to keep the basic factory settings which gives the best compromise between speed and no overshoot. You can always reset this settings from the "reset to factory" button in the " diver menu".

The figure below explains the effect of these 3 parameters for regulation characteristics:

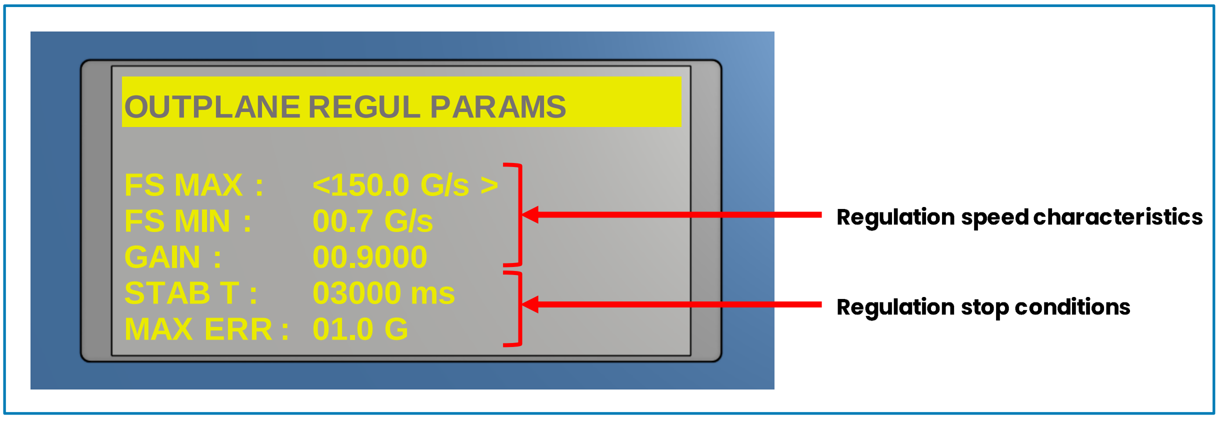


Field speed setpoint vs field error during regulation vs various parameter

d) Regulation Parameters Editing

There are two identical menus to manually editing the in-plane or out-plane regulation parameters.

Press **« F1 »** key twice for the "OUTPLANE REGUL PARAMS" menu or three times for the "INPLANE REGUL PARAMS" menu:



Oled screen view of OUTPLANE REGUL PARAMS menu on MFC front panel

Caution: you must set the « MAX ERR » parameter correctly to take into account the measurement noise that is not the same over the entire field range. If the amplitude of the measurement noise is greater than the given interval range then the regulation will never be able to stop because the field will always exit the setpoint interval range.

List of regulation parameters and related remote commands:					
Name	**Related remote command (1)**	**min**	**Default**	**max**	**Unit**
Minimum field speed	GET_REG_OUTP_ MIN_FS GET_REG_INP_ MIN_FS GET_REG_MIN_FS SET_REG_MIN_FS	0	1.0 (INP) 0.7 (OUTP)	10	G/Sec
Maximum field speed	GET_REG_OUTP_ MAX_FS GET_REG_INP_ MAX_FS GET_REG_MAX_FS SET_REG_MAX_FS	0	380 (INP) 150 (OUTP)	350	G/Sec
Gain	GET_REG_OUTP_GAIN GET_REG_INP_GAIN GET_REG_GAIN SET_REG_GAIN	0.0001	0.9 (INP) 0.7 (OUTP)	5	No Unit
Stabilization time	GET_REG_OUTP_ STAB_TIME GET_REG_INP_ STAB_TIME GET_REG_STAB_TIME SET_REG_STAB_TIME	0	3000 (INP) 3000 (OUTP)	99999	ms
Max error range	GET_REG_OUTP_ MAX_ERR GET_REG_INP_ MAX_ERR GET_REG_MAX_ERR SET_REG_MAX_ERR	0.5	1.2(INP) 1.0 (OUTP)	99.9	G

(1) See remote control section

Motor operation – Open loop

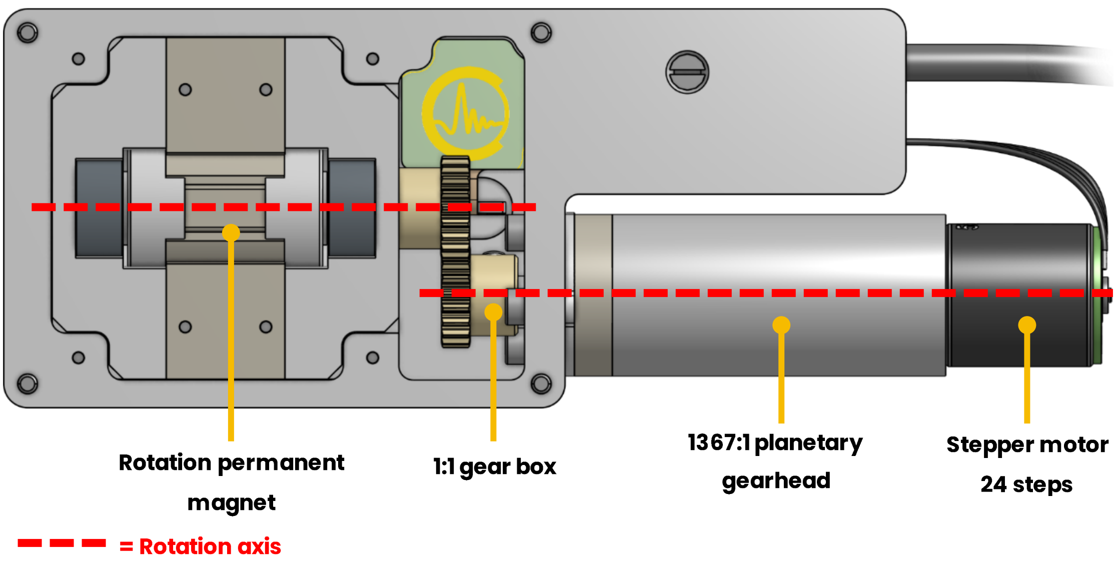
a) Introduction

The system is designed to be scalable and fully open to the user. The user can therefore control the MFM motor by himself to create his own regulation for example.

WARNING: When the motor is used, it generates a heating effect due to its driving current. This heat dissipation creates long-term drift of the magnetic field and can affect the user sample to. Temperature sensor is present on the module to monitor the temperature variations. This allows the user to know if the engine is used too often. Our tests show that using the motor 1 min at full speed every 4min does not affect the module significantly.

The motor is controlled by three parameters:

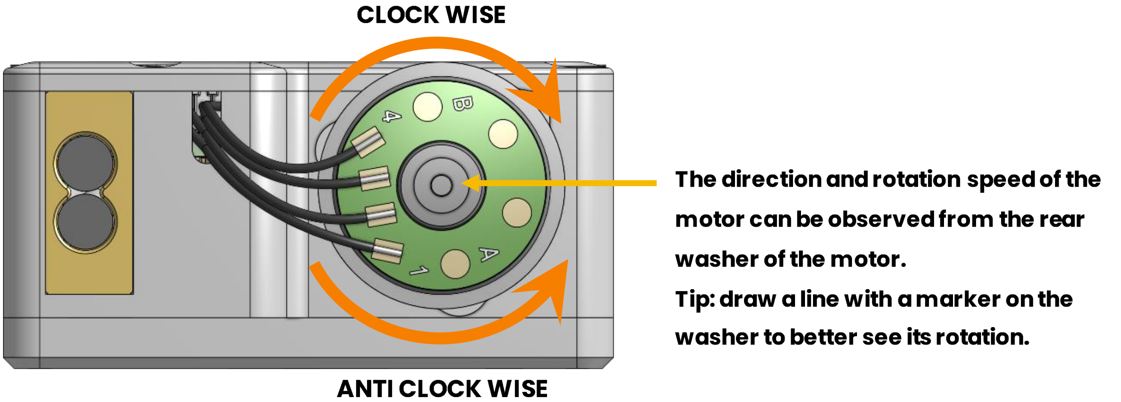
- **Speed:** The speed corresponds to the stepper motor steps frequency in Hz. The motor has 24 steps per turn. It is connected to a 1367:1 planetary gearhead reducer. This reducer is connected to the permanent magnet thanks to 1:1 gear box. So a full turn of the permanent magnet inside the module correspond to 24 * 1367 * 1 = 32808 steps.



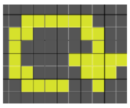
The equation bellow gives the conversion between stepper motor steps speed in Hz and permanent magnet speed in turn/minutes:

$$magnet\ rotation\ speed\left[\frac{turn}{min}\right] = \frac{32\ 808}{motor\ freq\,[Hz] \times 60}$$

- **Rotation side:** define the rotation side of the stepper motor view from the back of the motor.
  WARNING: the rotation side does not give the direction of field variation! During a full turn of the permanent magnet in the same direction, the field increases and decreases (or vice versa) between the minimum and maximum field! If you use the controller field regulation, we use the part of the field curve where the clockwise side decreases the magnetic field and vice versa.



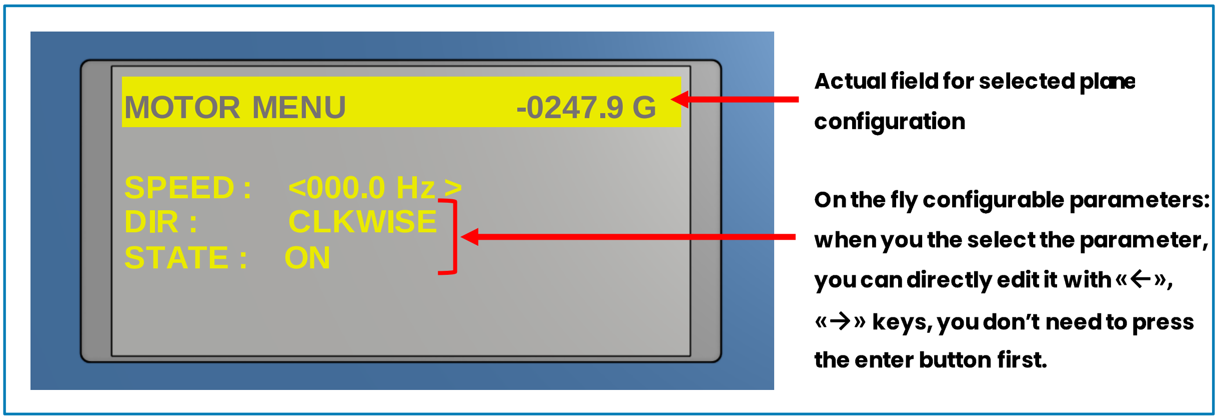
- **Motor state:** The state defines if the module motor is disable (fully disconnected from the MFC) or enable. When enabled, the motor start to turn if the configured speed is superior to 0Hz.
  Note: you can enable the motor and set a speed of zero to stop the motor but the current will continue to flow thought the motor. This creates heat that can affect the user sample and the permanent magnet of the MFM by create long-term magnetic field drift. It is hardly recommended to disable the motor when you don't want to change the module magnetic field.



  Note 2: When the motor is enabled, a dedicated icon appears on the Home page (see home page section). It is removed when the motor is disable. The icon is static for manual motor control and dynamic (icon turning) when the field regulation drives the motor (closed loop).

b) Motor parameters editing

For manual motor control press the **« F3 »** key to enter the "MOTOR MENU":



Oled screen view of MOTOR MENU on MFC front panel

List of motor parameters and related remote commands:

Name	Related remote command (1)	Values			
		min	**Default**	**max**	**Unit**
Speed	SET_MOTOR_FREQ GET_MOTOR_FREQ	000.0	000.0	350.0	Hz
Direction (Rotation side)	SET_MOTOR_DIR GET_MOTOR_DIR	Clock wise / Anti clock wise			
State	SET_MOTOR_STOP SET_MOTOR_START GET_MOTOR_STATE	ON / OFF			

(1) See remote control section

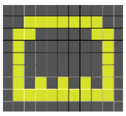
Remote Control (Ethernet / Serial)

a) Ethernet introduction / menu

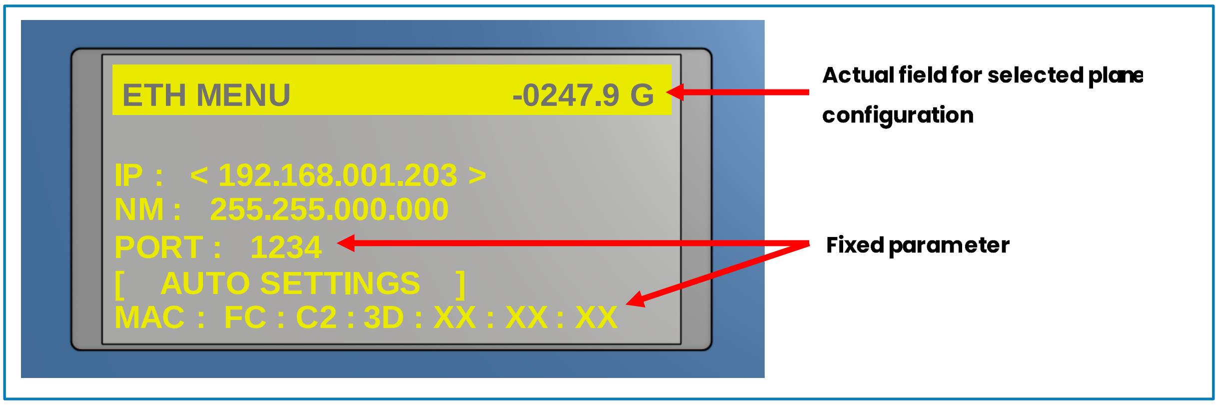
The system can be fully controlled via a sockets TCP/IP communication protocol from the rear RJ45 ethernet connector (see rear panel section of this documentation). The communication port is 1234.

To configure the IP and mask go to the "ETH MENU" by pressing **«F2»** key. You can do it manually or select the "AUTO SETTINGS" button to lunch a DHCP request and find IP and MASK automatically if the DHCP is available on your network.

The system manages the **simultaneous connection of 4 sockets maximum**.



Each time an ethernet socket is open, a dedicated icon appears on the Home page (see home page section). It is removed when the connection with the socket ends.



Oled screen view of DIVER MENU on MFC front panel

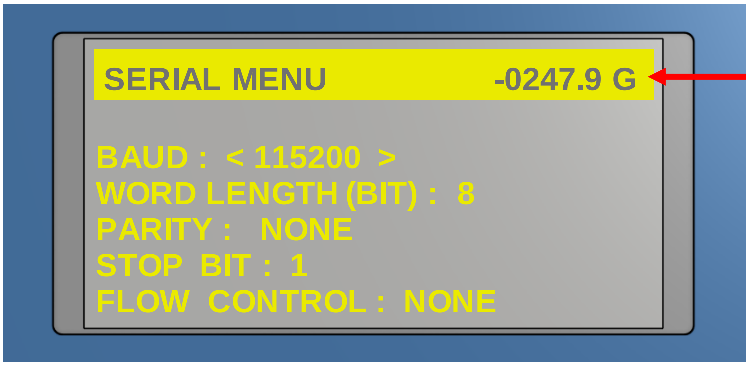
b) Serial configuration / menu

The system can be fully controlled via serial communication protocol from the rear RS232 or USB connectors (see rear panel section of this documentation).

The USB connector is connected to an USB to serial UART interface (FT234XD). The MFC appear as a normal port COM on your PC and you just have to open a serial communication with it.

IMPORTANT NOTE: The two RS232 and USB connectors work together and their drivers are internally connected to the same serial uart communication with the MFC microprocessor. It means that when the MFC sends a response to a command from USB or RS232 connector, the response is sent at the same time on the two communication channels (both connectors). It also means that these two communication channels cannot be used at the same time as it would cause collisions between received commands.

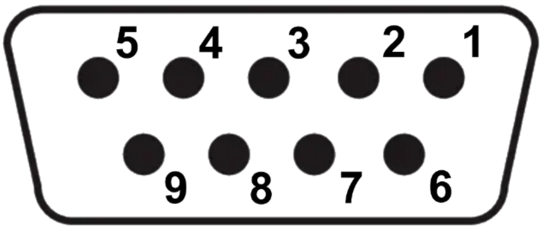
To configure the serial communication parameters, go to the "SERIAL MENU" by pressing **«F2»** key twice.



Actual field for selected plane configuration

Oled screen view of DIVER MENU on MFC front panel

RS232 Sub-D connector pinning:



1 – N.C	6 – N.C
2 - RX	7 - RTS
3 - TX	8 - CTS
4 – N.C	9 – N.C
5 – 0V (floating from rack ground)	

c) Sending / receive command (Ethernet, RS232, USB)

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 string isn't case sensitive so you can use upper or lower case. This documentation presents the commands with upper case.

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 first command find on the 1024-byte receive buffer.

If a command requires additional arguments, these must be separated by a single space between each argument. Example: in the command "SET_FIELD +100.5\n", the argument +100.5 is separated from the command SET_FIELD by a single space.

Response to a command:

The MFC respond for each command 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.

For « SET » commands that have the format « SET_XXX » (where « XXX » is the command), the start of the response always corresponds to the copy of the command « SET_XXX » sent with an « _OK » added if the command was executed without problem. Example: for the command « SET_REG_STOP » the response is « SET_REG_STOP_OK ». In case of errors during the command processing, the « _OK » confirmation is replaced by « _ERROR » followed by an argument which explain the error source. For example, if « SET_FIELD 999999» command is sent, the controller will returns « SET_FIELD_ERROR OVERRANGE » because the module can't reach this magnetic field setpoint.

For « GET » commands type that have the format « GET_XXX » (where « XXX » is the name of the value to get), the response always starts with a copy of the value name « XXX » sent with an « = » added. Example: for the command « GET_FIELD » the response is « FIELD= -309.58 G »

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

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

Some « GET » commands return the value with the unit as additional argument after the value like the example of « GET_FIELD ».

<u>IMPORTANT NOTE :</u>

- It is important to check the response of the MFC for each command sent to ensure that it has been processed correctly and to avoid MFC microcontroller saturation in the case of commands sent too quickly at the same time.
- Some commands take time to run, so the MFC response may take few seconds to arrive.
- All commands and responses are in Gauss independently of the unit configured for the MFC display.

d) List of Commands (Ethernet, RS232, USB)

*IDN?	
Response : **MFC<serial_number>**	Ex: MFC5002-015
Return the unique serial number of the MFC. <serial_number> is a string corresponding to the serial number with the format MFM5002-XXX where XXX is the unique number of this product. This serial number is unique and specific to CAYLAR products.	

HELP	
Response : **LIST OF COMMANDS AND HELPS**	
Return a list of more precise help commands and/or available commands.	

GET_FIELD	
Response : **FIELD= <field> G**	Ex: FIELD= +100.17 G
Return the last measurement of the field made by the MFM hall sensor corrected by the calibration table (in-plane or out-plane). It is measured 5 times per seconds (5 Hz) and returned in Gauss unit. <field> format is « %+.2 lf ». • If the configuration of the MFC plane (in-plane or out-plane) doesn't match the physical configuration of the MFM, the field reading is incorrect. • Related command: « SET_FIELD ».	

GET_FIELD_BRUT	
Response : **FIELD_BRUT= <volts> V**	Ex: FIELD_BRUT= -0.203137 V
Return the last voltage measured on the MFM hall sensor without any correction or calibration (brut value). The voltage is proportional to the magnetic field seen by the hall sensor but the ratio is not linear. It is measured 5 times per seconds (5 Hz) and returned in Volts unit. <volts> format is « %+.6 lf ». • If the configuration of the MFC plane (in-plane or out-plane) doesn't match the physical configuration of the MFM, the field brut reading is incorrect.	

GET_FIELD_SPEED	
Response : **FIELD_SPEED= <f_speed> G/Sec**	Ex: FIELD_SPEED= +0.00 G/Sec
Return the last field speed calculated by the MFC from the field measurement. It is calculated 5 times per seconds (5 Hz) and returned in Gauss per second unit. <f_speed> format is « %+.2 lf ».	

GET_FIELD_SPEED_F	
Response : **FIELD_SPEED_F= <f_speed> G/Sec**	Ex: FIELD_SPEED_F= +0.00 G/Sec
Same as GET_FIELD_SPEED commands but with a moving average calculation of 5 measures. <f_speed> format is « %+.2 lf ».	

GET_REG_PLANE_MODE

Response : **REG_PLANE_MODE= <plane>**	Ex: REG_PLANE_MODE= 0

Return the actual plane configuration given by the user and used by the MFC for field measurement and regulation.

<plane> is a Boolean corresponding to the plane configuration: 0 = in-plane mode / 1 = out-plane mode.

- Related command: « SET_REG_PLANE_MODE ».

GET_REG_ERROR

Response : **REG_ERROR= <error> G**	Ex: REG_ERROR= -0.43 G

Return the last calculated error of the field regulation. The error formula is: $\text{field}_{\text{measure}} - \text{field}_{\text{user setpoint}}$. This value is recalculated 5 times per seconds (5 Hz) and returned in Gauss unit.

<error> format is « %+.2 lf ».

GET_REG_SETPOINT

Response : **REG_SETPOINT= <setpoint> G**	Ex: REG_SETPOINT= +1000.00 G

Return the user field setpoint. The field setpointis returned in Gauss unit.

<setpoint> format is « %+.2 lf ».

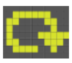
- Related command: « SET_FIELD » (change the field setpoint).

GET_REG_STATE

Response : **REG_STATE= <state>**	Ex: REG_STATE= 0

Return the field regulation state.

<state> is a Boolean corresponding to the field regulation state: 0 = regulation is stopped / 1 = regulation is active (motor is running and field is changing to reach the user-defined setpoint).

-  This icon appears on OLED screen in front of the MFC rack to indicate that the regulation is active (warning: it also means that the user can manually drive the motor).
- Related commands: « SET_FIELD » (start the regulation), « SET_REG_STOP ».

GET_REG_STAB_TIME (GET_REG_OUTP_STAB_TIME / GET_REG_INP_STAB_TIME)

Response : **REG_<plane>_STAB_TIME= <stab_time> ms**	Ex: REG_OUTP_STAB_TIME= 3000 ms

Return the stabilization time parameter of the field regulation loop for the actual plane mode (see field regulation operation section of this manual for more information about this parameter). Use GET_REG_OUTP_STAB_TIME or GET_REG_INP_STAB_TIME to get the parameter corresponding to the desired plane mode.

<stab_time> is an Interger given in milliseconds.

<plane> is a string corresponding to the returned parameter plane mode: «INP» = in-plane / «OUTP» = out-plane.

- Related command: « SET_REG_STAB_TIME ».

GET_REG_MAX_ERR (GET_REG_OUTP_MAX_ERR / GET_REG_INP_MAX_ERR)

Response : **REG_<plane>_MAX_ERR= <max_error> G**	Ex: REG_INP_MAX_ERR= +1.2 G

Return the maximum error parameter of the field regulation loop for the actual plane mode (see field regulation operation section of this manual for more information about this parameter). Use GET_REG_OUTP_MAX_ERR or GET_REG_INP_MAX_ERR to get the parameter corresponding to the desired plane mode.

< max_error > format is « %+.1 lf » given in Gauss unit.

<plane> is a string corresponding to the returned parameter plane mode: «INP» = in-plane / «OUTP» = out-plane.

- Related command: « SET_REG_MAX_ERR ».

GET_REG_MAX_FS (GET_REG_OUTP_MAX_FS / GET_REG_INP_MAX_FS)

Response : **REG_<plane>_MAX_FS= <max_fs> G/Sec**	Ex: REG_INP_MAX_FS= +380.0 G/Sec

Return the maximum field speed parameter of the field regulation loop for the actual plane mode (see field regulation operation section of this manual for more information about this parameter). Use GET_REG_OUTP_MAX_FS or GET_REG_INP_MAX_FS to get the parameter corresponding to the desired plane mode.

< max_fs > format is « %+.1 lf » given in Gauss per second unit.

<plane> is a string corresponding to the returned parameter plane mode: «INP» = in-plane / «OUTP» = out-plane.

- Related command: « SET_REG_MAX_FS ».

GET_REG_MIN_FS (GET_REG_OUTP_MIN_FS / GET_REG_INP_MIN_FS)

Response : **REG_<plane>_MIN_FS= <min_fs> G/Sec**	Ex: REG_INP_MIN_FS= +1.0 G/Sec

Return the minimum field speed parameter of the field regulation loop for the actual plane mode (see field regulation operation section of this manual for more information about this parameter). Use GET_REG_OUTP_MIN_FS or GET_REG_INP_MIN_FS to get the parameter corresponding to the desired plane mode.

< min_fs > format is « %+.1 lf » given in Gauss per second unit.

<plane> is a string corresponding to the returned parameter plane mode: «INP» = in-plane / «OUTP» = out-plane.

- Related command: « SET_REG_MIN_FS ».

GET_REG_GAIN (GET_REG_OUTP_GAIN / GET_REG_INP_GAIN)

Response : **REG_<plane>_GAIN= <gain>**	Ex: REG_INP_GAIN= 0.700000

Return the gain parameter of the field regulation loop for the actual plane mode (see field regulation operation section of this manual for more information about this parameter). Use GET_REG_OUTP_GAIN or GET_REG_INP_GAIN to get the parameter corresponding to the desired plane mode.

< gain > format is « %.6 lf » given without unit.

<plane> is a string corresponding to the returned parameter plane mode: «INP» = in-plane / «OUTP» = out-plane.

- Related command: « SET_REG_GAIN ».

CAYLAR
INSTRUMENTATION SCIENTIFIQUE

GET_REG_MAX_SETPOINT (GET_REG_INP_MAX_SETPOINT/ GET_REG_OUTP_MAX_SETPOINT)

Response : **REG_<plane>_MAX_SETPOINT= <max_sp> G**	Ex: REG_INP_MAX_SETPOINT= 6030 G

Return the maximum setpoint of the field regulation loop for the actual plane mode. Use GET_REG_OUTP_MAX_SETPOINT or GET_REG_INP_MAX_SETPOINT to get the parameter corresponding to the desired plane mode.

< max_sp > is an integer given in gauss unit.

<plane> is a string corresponding to the returned parameter plane mode: «INP» = in-plane / «OUTP» = out-plane.

GET_REG_MIN_SETPOINT (GET_REG_INP_MIN_SETPOINT/ GET_REG_OUTP_MIN_SETPOINT)

Response : **REG_<plane>_MIN_SETPOINT= <min_sp> G**	Ex: REG_INP_MIN_SETPOINT= -6020 G

Return the minimum setpoint of the field regulation loop for the actual plane mode. Use GET_REG_OUTP_MIN_SETPOINT or GET_REG_INP_MIN_SETPOINT to get the parameter corresponding to the desired plane mode.

< min_sp > is an integer given in gauss unit.

<plane> is a string corresponding to the returned parameter plane mode: «INP» = in-plane / «OUTP» = out-plane.

GET_MOTOR_FREQ

Response : **MOTOR_FREQ= <freq> Hz**	Ex: MOTOR_FREQ= +0.0 Hz

Return the actual motor control frequency which corresponds to the motor/magnet rotation speed. The frequency is returned in Hz. The frequency range is [0.0 to 350.0] Hz.

<freq> format is « %+.1 lf ».

- **WARNING:** The frequency may be zero (no motor/magnet rotation) but the motor may still be activated.
- Related command: « SET_MOTOR_FREQ ».

GET_MOTOR_DIR

Response : **MOTOR_DIR= <dir>**	Ex: MOTOR_DIR= 0

Return the actual motor/magnet rotation direction.

<dir> is a Boolean corresponding to the motor/magnet rotation direction seen from the rear motor axle : 0 = clockwise / 1 = anticlockwise.

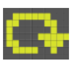
- Related command: « SET_MOTOR_DIR ».

GET_MOTOR_STATE

Response : **MOTOR_STATE= <state>**	Ex: MOTOR_STATE= 0

Return whether or not the motor is currently enabled (powered on).

<state> is a Boolean corresponding to the motor power state: 0 = motor is disabled (completely disconnected from the MFC rack) / 1 = motor is enabled (powered on).

-  This icon appears on OLED screen in front of the MFC rack to indicate that the motor is enable.
- **WARNING:** A powered motor cause electromagnetic interferences and module heating that can cause long-term field drift. For good field stability performance, do not use it too repeatedly and for too long periods (see « GET_HALL_TEMP » command to monitor the MFM temperature).
- **WARNING 2:** A powered motor doesn't mean that it is rotating if its control frequency is 0 Hz.
- Related command: « SET_MOTOR_STATE ».

GET_HALL_TEMP

Response : **HALL_TEMP= <temp> Deg**	Ex: HALL_TEMP= +21.56 Deg

Return the last measured temperature of the MFM hall sensor. It is measured 5 times per second (5Hz) and returned in degree Celsius unit.

<temp> format is « %+.2 lf ».

- **WARNING:** We measure the temperature thank to the hall sensor itself so the measure can be a little bit affected by the field value (less than ±0.1°C on the full range for both plane configuration).
- **WARNING 2:** No calibration is performed on this measure. It is given as an indication of external variations but it is not an absolute measure.

GET_RACK_TEMP

Response : **RACK_TEMP= <temp> Deg**	Ex: RACK_TEMP= +29.66 Deg

Return the last measured temperature of the MFC electronic near to the analog to digital converter (ADC) thank to a NTC thermistor. It is measured 5 times per second (5Hz) and returned in degree Celsius unit.

<temp> format is « %+.2 lf ».

- **WARNING:** No calibration is performed on this measure. It is given as an indication of electronic variations but it is not an absolute measure.

GET_STATUS

Response : **STATUS= <status_byte>**	Ex: STATUS= 63 (out-plane, motor on, field regulation active, motor anti-clockwise, init ended and init ok)

Return an integer containing the various status of the MFC as an array of 8 bits where each bit represents a Boolean.
- bit 0 (LSB) - plane mode: 0 = in-plane, 1 = out-plane.
- bit 1 – field regulation state: 0 = inactive, 1 = active.
- bit 2 – motor state: 0 = inactive, 1 = active.
- bit 3 – motor rotation side: 0 = clockwise, 1 = anti-clockwise.
- bit 4 – init ended: 0 = no, 1 = yes.
- bit 5 – init ok: 0 = init ended with problems, 1 = init ended without problems.
- bit 6,7 – Not used.

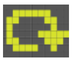
< status_byte > is an Integer which represent an unsigned byte of 8 bits.

- The purpose of this command is to monitor the various status of the MFM/MFC quickly without use to many commands at the same time like « GET_MOTOR_DIR », « GET_MOTOR_STATE », « GET_REG_STATE » ...

SET_FIELD <setpoint>	Ex: SET_FIELD 1200.25 / SET_FIELD -120
Response : **SET_FIELD_OK <field> G**	Ex: SET_FIELD_OK +1200.25 G / SET_FIELD_OK -120.00 G
Response in case of unrecognized argument <setpoint> : **SET_FIELD_ERROR BAD_ARG**	
Response in case of overrange in <setpoint> argument : **SET_FIELD_ERROR OVERRANGE**	

Set the field setpoint for the actual plane mode (in-plane or out-plane) and start the regulation to reach the new user setpoint.

<setpoint> format can be double, float or integer / signed or not signed given in gauss.
<field> format is « %+.2 lf ». <field> is the recopy of <setpoint> for the MFC feedback response.

- Related commands : « GET_REG_SP », « GET_REG_MIN_SETPOINT », « GET_REG_MAX_SETPOINT ».
-  While the regulation is active this icon appears on the MFC screen main page.

SET_UNIT <unit>	Ex: SET_UNIT GAUSS
Response : **SET_UNIT_OK <unit>**	Ex: SET_UNIT_OK GAUSS
Response in case of unrecognized argument <side> : **SET_UNIT _ERROR BAD_ARG** or **SET_UNIT_ERROR UNKNOWN_UNIT**	

Set the MFC display field unit.

<unit> is a string corresponding to MFC front panel display field unit, can be : GAUSS, TESLA or mTESLA (respect case).

- Only changes the values units on display screen : Does not change the ethernet commands unit and ethernet response.

SET_MOTOR_DIR <side>	Ex: SET_MOTOR_DIR 1
Response : **SET_MOTOR_DIR_OK <side>**	Ex: SET_MOTOR_DIR_OK 1
Response in case of unrecognized argument <side> : **SET_MOTOR_DIR_ERROR BAD_ARG**	
Response if field regulation is running : **SET_MOTOR_DIR_ERROR REGUL_RUNNING**	

Set the motor rotation side.

<side> is a Boolean corresponding to the motor/magnet rotation side seen from the rear motor axis, can be : 0 = clockwise / 1 = anticlockwise.

- Related command : « GET_MOTOR_DIR ».

SET_MOTOR_FREQ <freq>	Ex: SET_MOTOR_FREQ 250.251
Response : **SET_MOTOR_FREQ_OK <freq_feedback> Hz**	Ex: SET_MOTOR_FREQ_OK +250.3 Hz
Response in case of unrecognized argument <freq> : **SET_MOTOR_FREQ_ERROR BAD_ARG**	
Response if field regulation is running : **SET_MOTOR_FREQ_ERROR REGUL_RUNNING**	
Response in case of overrange in <freq> argument : **SET_MOTOR_FREQ _ERROR OVERRANGE**	

Set the motor control frequency which corresponds to the motor/magnet rotation speed.

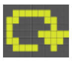
<freq> format can be double, float or integer / signed or not signed given in Hertz. Positive number only. The configurable frequency range is [0 - 350.0] Hz.
<freq_feedback> format is « %+.1 lf ». <freq_feedback> is the copy of <freq> for the MFC feedback response.

- Related command : « GET_MOTOR_FREQ ».

SET_MOTOR_STATE <state>	Ex: SET_MOTOR_STATE 1
Response : **SET_MOTOR_STATE_OK <state_feedback>**	Ex: SET_MOTOR_STATE_OK 1
Response in case of unrecognized argument <freq> : **SET_MOTOR_STATE_ERROR BAD_ARG**	
Response if field regulation is running : **SET_MOTOR_STATE_ERROR REGUL_RUNNING**	

Turn on or turn off (fully disconnect) the MFM motor.

<state> format can be Boolean or Integer: 0 = motor off / >=1 = motor on.
<state_feedback> is the copy of <state> for the MFC feedback response.

- Related command: « GET_MOTOR_STATE ».
-  This icon appears on the MFC screen main page when the motor is active.
- **WARNING:** A powered motor cause electromagnetic interferences and module heating that can cause long-term field drift. For good field stability performance, do not use it too repeatedly and for too long periods (see « GET_HALL_TEMP » command to monitor the MFM temperature).
- **WARNING 2:** A powered motor doesn't mean that it is rotating if its control frequency is 0 Hz.

SET_REG_PLANE_MODE <plane>	Ex: SET_REG_PLANE_MODE 1
Response : **SET_REG_PLANE_MODE_OK <plane_feedback>**	Ex: SET_REG_PLANE_MODE_OK 1
Response in case of unrecognized argument <plane> : **SET_REG_PLANE_MODE_ERROR BAD_ARG**	
Response if field regulation is running : **SET_REG_PLANE_MODE_ERROR REGUL_RUNNING**	
Response in case of bad <plane> argument : **SET_ REG_PLANE_MODE_ERROR BAD_PLANE_MODE**	

Configuration the plane mode (in-plane or out-plane) corresponding to the physical plane mode of the MFM.

<plane> is a Boolean: 0 = in-plane / 1 = outplane.
<plane_feedback> is the copy of <plane> for the MFC feedback response.

- Related command: « GET_REG_PLANE_MODE ».

SET_REGUL_STOP
Response : **SET_REGUL_STOP_OK**

Turns off the field regulation if active. Turn off the motor and set the motor control frequency to zero.

- Related command: « GET_REG_STATE ».

SET_REG_MAX_FS <plane> <max_fs>	Ex: SET_REG_MAX_FS 1 150
Response : **SET_REG_MAX_FS_OK <fb_plane> <fb_max_fs> G/Sec**	Ex: SET_REG_MAX_FS_OK 1 +150.0 G/Sec
Response in case of unrecognized argument <plane> or <max_fs> : **SET_REG_MAX_FS_ERROR BAD_ARG**	
Response in case of bad <plane> argument : **SET_REG_MAX_FS_ERROR BAD_PLANE_MODE**	
Response in case of <max_fs> argument overrange: **SET_REG_MAX_FS_ERROR FREQ_OVERRNG**	
Set the maximum field speed parameter of the field regulation loop for a given plane mode (see field regulation operation section of this manual for more information about this parameter). <plane> is a Boolean: 0 = in-plane / 1 = outplane. <max_fs> format can be double, float or integer / signed or not signed given in Gauss per second. Positive number only. The configurable range is [0 - 350.0] G/Sec. <fb_plane> is the copy of <plane> for the MFC feedback response. <fb_max_fs> is the copy of <max_fs> for the MFC feedback response with format « %+.1 lf ». • Related command: « GET_REG_MAX_FS ».	

SET_REG_MIN_FS <plane> <min_fs>	Ex: SET_REG_MIN_FS 1 0.7
Response : **SET_REG_MIN_FS_OK <fb_plane> <fb_min_fs> G/Sec**	Ex: SET_REG_MIN_FS_OK 1 +0.7 G/Sec
Response in case of unrecognized argument <plane> or <min_fs> : **SET_REG_MIN_FS_ERROR BAD_ARG**	
Response in case of bad <plane> argument : **SET_REG_MIN_FS_ERROR BAD_PLANE_MODE**	
Response in case of <min_fs> argument overrange: **SET_REG_MIN_FS_ERROR FREQ_OVERRNG**	
Set the minimum field speed parameter of the field regulation loop for a given plane mode (see field regulation operation section of this manual for more information about this parameter). <plane> is a Boolean: 0 = in-plane / 1 = outplane. <min_fs> format can be double, float or integer / signed or not signed given in Gauss per second. Positive number only. The configurable range is [0 - 10.0] G/Sec. <fb_plane> is the copy of <plane> for the MFC feedback response. <fb_min_fs> is the copy of <min_fs> for the MFC feedback response with format « %+.1 lf ». • Related command: « GET_REG_MIN_FS ».	

SET_REG_GAIN <plane> <gain>	Ex: SET_REG_GAIN 1 0.9
Response : **SET_REG_GAIN_OK <fb_plane> <fb_gain>**	Ex: SET_REG_GAIN_OK 1 +0.90000
Response in case of unrecognized argument <plane> or <gain> : **SET_REG_GAIN_ERROR BAD_ARG**	
Response in case of bad <plane> argument : **SET_REG_GAIN_ERROR BAD_PLANE_MODE**	
Response in case of <gain> argument overrange: **SET_REG_GAIN_ERROR GAIN_OVERRNG**	
Set the gain parameter of the field regulation loop for a given plane mode (see field regulation operation section of this manual for more information about this parameter). <plane> is a Boolean: 0 = in-plane / 1 = outplane. <gain> format can be double, float or integer / signed or not signed. Positive number only. The configurable range is [0.0001 - 5.0] (no unit). <fb_plane> is the copy of <plane> for the MFC feedback response. <fb_gain> is the copy of <gain> for the MFC feedback response with format « %+.5 lf ». • Related command: « GET_REG_GAIN ».	

SET_REG_STAB_TIME <plane> <stab_t>	Ex: SET_REG_STAB_TIME 1 3000
Response : **SET_REG_STAB_TIME_OK <fb_plane> <fb_stab_t> ms**	Ex: SET_REG_STAB_TIME_OK 1 3000 ms
Response in case of unrecognized argument <plane> or <stab_t> : **SET_REG_STAB_TIME_ERROR BAD_ARG**	
Response in case of bad <plane> argument : **SET_REG_STAB_TIME_ERROR BAD_PLANE_MODE**	
Response in case of <stab_t> argument overrange: **SET_REG_STAB_TIME_ERROR STAB_T_OVERRNG**	
Set the stabilization time parameter of the field regulation loop for a given plane mode (see field regulation operation section of this manual for more information about this parameter). <plane> is a Boolean: 0 = in-plane / 1 = outplane. <stab_t> is an unsigned Integer. Positive number only. The configurable range is [0 - 99999] milliseconds. <fb_plane> is the copy of <plane> for the MFC feedback response. <fb_stab_t> is the copy of <stab_t> for the MFC feedback response with format « %d ». • Related command: « GET_REG_STAB_TIME ».	

SET_REG_MAX_ERR <plane> <max_err>	Ex: SET_REG_MAX_ERR 1 1.0
Response : **SET_REG_MAX_ERR_OK <fb_plane> <fb_max_err> G**	Ex: SET_REG_MAX_ERR_OK 1 +1.0 G
Response in case of unrecognized argument <plane> or <max_err> : **SET_REG_MAX_ERR_ERROR BAD_ARG**	
Response in case of bad <plane> argument : **SET_REG_MAX_ERR_ERROR BAD_PLANE_MODE**	
Response in case of <max_err> argument overrange: **SET_REG_MAX_ERR_ERROR MAX_ERR_OVERRNG**	
Set the maximum error parameter of the field regulation loop for a given plane mode (see field regulation operation section of this manual for more information about this parameter). <plane> is a Boolean: 0 = in-plane / 1 = outplane. <max_err> format can be double, float or integer / signed or not signed given in Gauss. Positive number only. The configurable range is [0.5 - 99.9] G. <fb_plane> is the copy of <plane> for the MFC feedback response. <fb_max_err> is the copy of <max_err> for the MFC feedback response with format « %+.1 lf ». • Related command: « GET_REG_MAX_ERR ».	

e) Python ethernet and Serial example

This program example connects to the MFC via an ethernet socket. It requests the last magnetic field measurement and print it into the python console:

```
import socket
import traceback

mfc_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:

    mfc_socket.connect(("192.168.1.163", 1234)) # change this line with your MFC IP address
    mfc_socket.send(bytes("GET_FIELD\n", "ascii"))
    rx_buff = mfc_socket.recv(50).decode("ascii").split(' ')
    mfc_field = float(rx_buff[1])
    print(f"Actual field = {mfc_field} Gauss")

except:

    traceback.print_exc()

finally:

    mfc_socket.close()
```

Screenshot of the program output (Console View):

```
Actual field = 494.31 Gauss

Process finished with exit code 0
```

This program example connects to the MFC via a port com serial connection (can be thanks to an USB or RS232 connection). It requests the last magnetic field measurement and print it into the python console:

```
from serial import *
import traceback

try:

    mfc_serial = Serial(
      port="COM6", # change these parameters with your MFC port com and serial configuration
      baudrate=115200,
      bytesize=EIGHTBITS,
      parity=PARITY_NONE,
      stopbits=STOPBITS_ONE,
      rtscts=False,
    )
    mfc_serial.write(bytes("GET_FIELD\n", "ascii"))
    rx_buff = mfc_serial.readline().decode("ascii").split(' ')
    mfc_field = float(rx_buff[1])
    print(f"Actual field = {mfc_field} Gauss")

except:

    traceback.print_exc()

finally:

    mfc_serial.close()
```

Screenshot of the program output (Console View):

```
Actual field = 494.42 Gauss

Process finished with exit code 0
```

Notes

MANUFACTURER OF ELECTROMAGNETS,
NMR TESLAMETERS,
POWER SUPPLIES

www.caylar.net