

**MFCSTM Series Software Development Kit**

*Version 3.1.6*

*Version 3.1.5.150615*

Introduction 3

System Requirements 3

MFCS Functions Prototyping 3

Types equivalencies table 5

Installation Instructions for LabVIEW Wrapper 6

Installation Instructions for MATLAB Toolbox 7

Plug-in for µ-Manager : Overview and installation Instructions 8

# Introduction

The MFCSTM Series Software Development Kit allows you to fully integrate the MFCSTM device in your application. We provide a thread safe Dynamic Link Library that you can call in you development environment. The calling convention is C which means that the caller is responsible for cleaning up the stack.

# System Requirements

This installation requires one of the following Microsoft operating systems:

* Windows XP Service Pack 3
* Windows Vista (32 and 64 bits)
* Windows 7 (32 and 64 bits)
* Windows 8 (32 and 64 bits)
* Windows 10 (32 and 64 bits)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| function | return | parameter | | comment |
| mfcs\_initialisation | UL handle | US serial number | Initialize USB connection and launches a continuous check for a MFCS with the specified serial number.   * handle = 0 if no USB connection possible. * handle is not null even if no MFCS is now detected. | |
| mfcsez\_initialisation | UL handle | US serial umber | Initialize USB connection and launches a continuous check for a MFCS-EZ with the specified serial number.   * handle = 0 if no USB connection possible. * handle is not null even if no MFCS-EZ is now detected. | |
| mfcs\_close | B OK | UL handle | Close USB connection. MUST be called  before leaving the application. | |
|
| mfcs\_set\_purge\_on | C error | S string | Open purge on channel 1. | |
|
| UL handle |
| mfcs\_set\_purge\_off | C error | UL handle | Close purge on channel 1. | |
|
| S string |
|  |
|
|

# MFCS Functions Prototyping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| function | return | parameter | | comment |
| mfcs\_get\_purge | C error | UL handle | Get the purge state. | |
| PB purge state |
| mfcs\_get\_status | C error | UL handle | Get MFCS status :  0 if the MFCS is reset  1 if normal  2 if overpressure  3 if MFCS needs to be rearmed. | |
| PC Status |
| mfcs\_read\_chan | C error | UL handle | Read the pressure value (mBar) of the specified channel with the timing (time unit 25ms). | |
|
| C channel |
| PF pressure |
|
|
| PUS chrono |
| mfcs\_data\_chan | C error | UL handle | Read the sensor data :  - sensor unit :  0 = no sensor,  1 = "H2O (2.4908 mBar)  2 = psi (68.946 mBar)  - full scale in pressure unit  - zero value sensor (U12)  - direct pressure measure (U12)  - chrono (time unit 25ms) | |
| C channel |
| PC sensor unit |
| PUS sensor max |
| PUS zero |
| PUS measure |
| PUS chrono |
| mfcs\_get\_serial | C error | US handle | Get the serial number of the MFCS. | |
| PUS Serial |
| mfcs\_set\_auto | C error | US handle | Regulate pressure (mBar) on the specified channel. (if 0, the same for all channel). | |
| C channel |
| F pressure |
| mfcs\_set\_alpha | C error | UL handle | Set alpha value (U8). This value is linked to the PID performance. The default value is 5 and a preheating of the electro-valves is a mandatory (45% alimentation for 10 min). | |
| C channel |
| C alpha |
| mfcs\_set\_manual | C error | UL handle | Set electro-valve voltage (%) on the specified channel (if 0, the same for all channel). The manual control of the electro-valve is not recommended and the output pressure is no longer regulated. | |
| C channel |
| FEV |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| function | return | parameter | | comment |
| mfcs\_set\_zero | C error | UL handle | Save Zero sensor value on the firmware. To get this value, use mfcs\_data\_chan (PUS mesure). | |
| C channel |
| US Zéro |
| mfcs\_detect | C error | US[256]  Serial number table | Detect the connected MFCS and set the first values of the table to the serial numbers. The last values of the table are set to 0x0000 The order of the serial numbers in table is the order detected by Windows. | |
| mfcsez\_detect | C error | US[256]  Serial number table | Detect the connected MFCS-EZ and set the first values of the table to the serial numbers. The last values of the table are set to 0x0000 The order of the serial numbers in table is the order detected by Windows. | |

Definition of [C error]:

- 0 = OK - 1 = USB closed - 2 = Wrong channel

## Types equivalencies table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbol | Bits | C++ | LabVIEW | VB |
| UL | 32 | Unsigned long | U32 | ByVal Long |
| US | 16 | Unsigned short | U16 | ByVal Integer |
| PUS | 32 | Pointer to unsigned short | Pointer to U16 | ByRef Integer |
| S | 32 | Char[] | String | ByVal String |
| C | 8 | Unsigned char | U8 | ByVal Byte |
| PC | 32 | Unsigned char\* | Pointer to U8 | ByRef Byte |
| B | 8 | Char for Boolean  result 1 = true, 0 = false | U8 | ByVal Byte |
| PB | 32 | Pointer to Boolean | Pointer to U8 | ByRef Byte |
| F | 32 | Float | Single | ByVal Single |
| PF | 32 | Pointer to float | Pointer to Single | ByRef Single |
| US[256] | 32 | Unsigned short table[256] | Array of U16 | ByRef Integer |

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Bits | Java | MATLAB |
| UL | 32 | NativeLong | Ulong |
| US | 16 | Int | Uint16 |
| PUS | 32 | ShortByReference | Uint16Ptr |
| S | 32 | Char[] | String |
| C | 8 | Char | Uint8Ptr |
| PC | 32 | IntByReference | Uint |
| B | 8 | Boolean | Uint8 |
| PB | 32 | ByteByReference | Uint8Ptr |
| F | 32 | Float | Single |
| PF | 32 | FloatByReference | SinglePtr |
| US[256] | 32 | Int[256] | Uint16[256] |

NB :

1. In Visual Basic and Java, all values are signed.
2. In Java, IntByReference, ShortByReference, FloatByReference and ByteByReference belong to the Java Native Access API. See below for further details.
3. After calling mfcsez\_initialisation or mfcs\_initialisation, a delay (0.5s) must be set before calling other functions.
4. A mfcs\_close must be called before leaving the application, to avoid Windows memory allocation error.
5. In MATLAB, calling a library function with a pointer input argument is by using a variable from the corresponding variable type as both an input and an output argument

# Installation Instructions for MFCSTM API for LabVIEW

You need to have the JKI VI Package Manager 2014 or higher to be able to install fluigent\_mfcs\_driver-3.3.0.2.vip. Open the fluigent\_mfcs\_driver-3.3.0.2.vip within VIPM and follow the instruction. The VI Package is compatible with LabVIEW 8.6.1 version and higher for both 32 and 64 bits versions.

After the package installation you will have the Fluigent palette available from your VI block diagram.

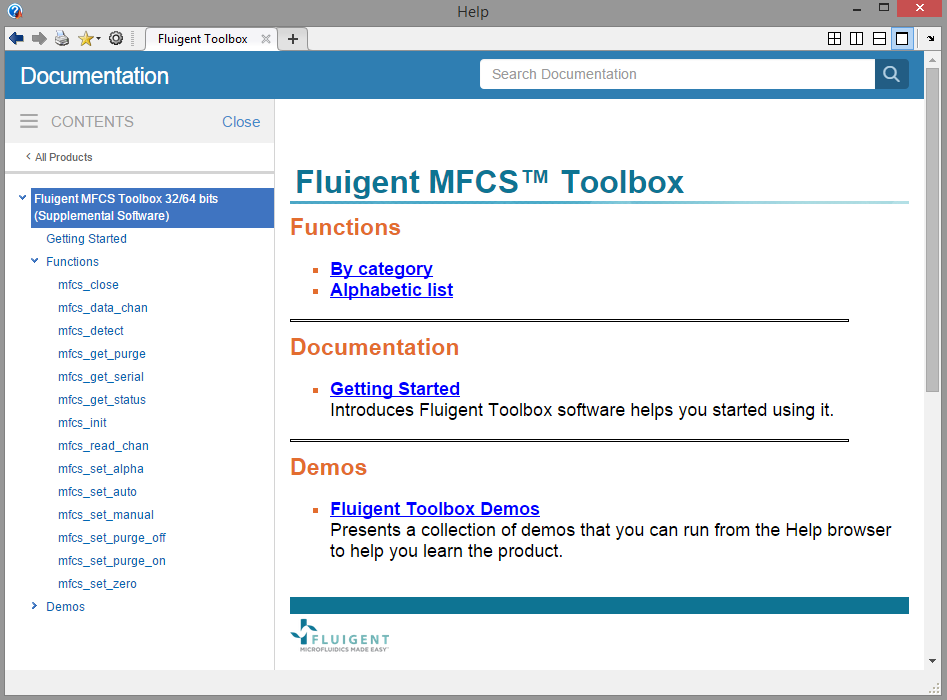
You will find the MFCSTM API for LabVIEW documentation in user.lib\Fluigent\MFCS\Doc\MFCS series API Help.chm. You also have the context help available directly from the VI block diagram of your VI. You can find examples using the NI Example Finder with Fluigent and MFCS as keywords.

# Installation Instructions for MATLAB Toolbox

Start MATLAB with Administrator privileges. Go the root of the Toolbox location. Run the toolbox installer “Fluigent MFCS 32bit toolbox.mltbx” or “Fluigent MFCS 64bit toolbox.mltbx” depending on used MATLAB version. The Fluigent Toolbox is compatible with MATLAB R2015a and higher. Please contact us if you are using previous MATLAB versions.

IMPORTANT: Fluigent toolbox needs a C compiler for MATLAB in order to use library functions. You can use default Lcc compiler provided with MATLAB or install new one using “mex –setup” command.

After the Toolbox installation is complete, all functions are available as any other MATLAB ones. The Toolbox documentation will be visible from the help function, the Function browser (Maj+F1) or the Help Documentation (F1). You can find examples and demo in the documentation.



# Plug-in for µ-Manager: Overview and installation Instructions

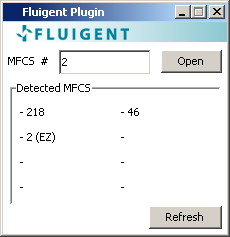
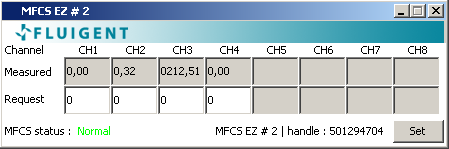
Micro-Manager software is a Java application, conceived as a plug-in for the ImageJ API.

1. Drop the “…\Fluigent Plugin for uManager\FluigentMMPlugin” folder in the mmplugins directory of the Micro-Manager-1.4 directory "...\Micro-Manager-1.4\mmplugins”
2. Drop the “…\Fluigent Plugin for uManager\ jna.jar ” in the extension folder of the JRE used by Micro-Manager “...\Micro-Manager-1.4\jre\lib\ext”
3. Run Micro-Manager. The Fluigent plug-in will appear in the plug-in menu.
4. When starting, the MFCSTM selection menu will appear.
5. Your connected MFCS TM devices will be displayed. If you connected a new device, wait a few seconds and click the Refresh button.
6. Choose your MFCS TM, enter its serial number in the corresponding text field, then click on “Open”
7. The new opened window enable to read a set the pressure (in mBars) on each level: write the desired pressure on each channel of the Request line, and then click on the “Set” Button. Information about the MFCS state are shown on the bottom of the window.

We provide the source code of the plug-in in “…\Examples\uManager” folder.

IMPORTANT:

* You can safely close each window opened by the Fluigent plug-in separately (even the selection menu) without affecting the other windows, but you must close all MFCS TM pressure controlling window before leaving Micro-Manager.
* You can safely reopen the plug-in several times from one instance of Micro-Manager, but you should not open two instances of the plug-in in two different instances of Micro-manager, or when using the MAESFLO software.





FLUIGENT  
Biopark

1 mail du professeur Mathé

94 800 Villejuif

FRANCE

Phone: +331 77 01 82 68

Fax: +331 77 01 82 70

[www.fluigent.com](http://www.fluigent.com)

Technical support:

[support@fluigent.com](mailto:support@fluigent.com)

General information :

[contact@fluigent.com](mailto:contact@fluigent.com)