

Application Midi-Ingénierie : **AppMiCom_v2025**

Cette application est construite autour de la bibliothèque de communication midi-ingénierie , disponible pour nos clients. (**MiCom_v202x**)

à partir des arguments précisés sur sa ligne d'appel, l'applicatif va mettre en œuvre les appels MiCom_v202x nécessaires pour assurer le dialogue avec le module MIDI-INGENIERIE concerné.

Le retour d'information est mis à disposition de l'appelant par le Presse Papier (Clipboard windows)

Exemple :

Commande **00qv** sur un **MAC23** , à **38400** baud , le port COM est fixé sur **COM17**

La ligne d'appel à mettre en œuvre est alors : (en supposant l'app localisée en c:\temp)

C:\temp\AppMiCom_v2025.exe MAC23 COM7 38400 00qv*

si le mode debug est activé, un écran console reste alors affiché :

```
C:\WINDOWS\system32\cmd.exe

c:\temp>AppMiCom_v2025 MAC23 COM17 38400 00qv debug
AppMiCom_v2025.exe v1.0 07/2025 Midi-Ingenierie (HMS) by MN
Driver := MAC23
Port := COM17
Baudrate := 38400
Cmde := 00qv
Open [0000]
Send [0003] -->00ev A13810B0 MIDI INGENIERIE_MAC23 _9138-02772_09/04/14_ BT1 1.3
[Enter] pour quitter :

c:\temp>
```

Le retour d'information est disponible dans le clipboard :

00ev A14210B0 MIDI INGENIERIE_MAC34 _9142-22339_28/03/19_28/03/19 BT1 1.3

Limitations :

C'est l'interprétation du contenu du clipboard qui informe d'une erreur de dialogue ou pas :

>	La commande a été généré avec succès (00gs , 00di , 00wx 6000 0
! xxxxx [hhhh]	Signalement d'une erreur Open ou Send et détail du cr MiCom_v202x (voir annexe)
reponse	Si la commande entraine une réponse du module MIDI, celle-ci est copiée dans le ClipBoard

*pour la commande 00qv la forme "00qv" n'est pas nécessaire. Elle est obligatoire pour les commandes combinées : " 00gf 2000 " , " 00gi 20 " , ...

Ex :

c:\temp>AppMiCom_v2025 MAC23 COM17 38400 "00gf 2000"

c:\temp>AppMiCom_v2025 MAC23 COM17 38400 00ge

c:\temp>AppMiCom_v2025 MAC23 COM17 38400 "00gi 20"

Détail du CR MiCom

Le compte rendu est commun à toutes les fonctions., Il est de type Integer sur 16bits utiles organisés comme suit :

à partir de v1.26, (en annexe le mapping pour les versions antérieures)

b15	Fail / Error
b14	com lost
b13	recept timeout
b12	emis failed
b11	recept nack or failed
b10	control
b09	out of range
b08	open send
b07	close
b06	mac converter error
b05	baudrate
b04	driver
b03	port
b02	timeout
b01	mac converter
b00	recept from driver

Information :

recept from driver : la commande générée est une requête, le retour d'information est correctement reçu

mac converter : la commande générée est destinée à un module Midi de type MAC23 ou MAC34

Défaut:

mac converter error : la conversion de la commande en mode EXPERT MAC23/MAC34 a échoué

baudrate : valeur négative ou non numérique

driver : la référence fournie est inconnue

port : la référence est nulle ou non conforme

timeout : valeur out of range

Warning :

com lost : la communication est non opérationnelle, le nb de timeOut dépasse un seuil fixé. Positionné également en cas **emis failed**.

recept timeout : la commande n'a pas été acquittée dans le délai imparti. (Câble débranché, défaut d'adresse, erreur de baudrate,)

emis failed : émission de la commande en erreur (le port COM ?)

recept nack or failed : la commande est refusée par le driver (NACK) ou la trame retournée par le driver est incorrecte

control : précise que l'erreur trame est de type CHECKSUM

out of range : précise que l'erreur est de type NB CARACT

open send : générée par la fonction Open (port déjà utilisé ou absent)
ou par la fonction Send (absence du Open)

close : générée lors de la fonction Close

Fail / Error:

Ce flag est automatiquement positionné dès qu'un flag autre que Recept from driver / MacConverter est positionné.

Il n'est jamais positionné SEUL, la valeur 0x8000 n'existe pas : **un Status > 0x8000 signale une erreur**

Exploitation du CR

b15	b14	b13	b12	b11	b10	b09	b08	b07	b06	b05	b04	b03	b02	b01	b00
Fail/Error	com lost	recept timeout	emis failed	recept nack or failed	control	out of range	open send	close	mac converter error	baudrate	driver	port	timeout	mac converter	recept from driver

Les Cr suivants sont inférieurs à 0x8000 : la communication est correcte

0x0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Nominal hors requête (driver autre que MAC34/MAC23)
0x0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Nominal pour requête (driver autre que MAC34/MAC23)
0x0002	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	MAC34 / MAC23 : nominal hors requête
0x0003	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	MAC34 / MAC23 : nominal pour requête

Les Cr supérieurs à 0x8000 signalent un défaut de **communication** ou une **configuration incorrecte** :

0x8004	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	valeur de timeout hors limite
0x8008	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	portname non valide ⁴
0x8010	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	driver inconnu
0x8020	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	baudrate incorrect
0x8100	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	erreur : Aléa ¹ sur fonctions Open ou Send
0x8800	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	défaut commande ² , erreur de syntaxe
0xA000	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	absence de réception (TimeOut)
0xE000	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	perte de la communication ³ (trop de recept timeout consécutifs)

1 : La fonction **Open** positionne cette erreur dans le cas où le **port** précisé en appel est déjà en cours d'utilisation par une autre application, si ce port n'existe pas (COM55 pour COM5) ou si ce port valide est un port pour l'usage Bluetooth
La fonction **Send** positionne cette erreur essentiellement dans le cas où il n'y a pas eu d'appel valide à la fonction **Open** AVANT l'appel à la fonction **Send**

2 : Cette erreur signale une réponse non nominale de la part du driver. (Voir Annexe)
En l'absence des erreurs **control** ou **out of range**, le détail de l'erreur est exploitable par la fonction Get_Info(« DETAIL_CR »)

3 : Cette erreur signale une « perte » de la communication :
0xE000 : combinée avec **recept timeout** → le nombre de timeout consécutif dépasse un seuil fixé (15)
0xD000 : combinée avec **emis failed** → les émissions sont impossibles (?? aléa grave sur l'uart associé au port)

4 : Port invalide : L'information **port** peut être fixée de 2 façons :

a) Indépendamment de la fonction Open, la fonction Open se faisant alors par une signature réduite :

```
Midi_Ingenierie.MiComm_2022 myCom = new Midi_Ingenierie.MiComm_2022();
myCom.PortName = "COM10"; // sélection du Port ,indépendamment de la fonction OPEN
//....
myCom.Open("MIBL") ; // fonction OPEN , signature réduite
```

b) Directement par la fonction Open (signature complète):

```
Midi_Ingenierie.MiComm_2022 myCom = new Midi_Ingenierie.MiComm_2022();
myCom.Open("MIBL","COM10","115200") ; // fonction OPEN signature complète
```

Dans tous les cas, l'erreur **Port** est positionnée si l'information PortName est nulle ou non conforme.
L'erreur **Open Send**, est positionnée lors de la fonction **open**, la présence du COM est vérifiée et validée :
→ un **COMxx** de type VCP doit être installé AVANT l'appel à la fonction **Open** (driver , **COMxx**,..)
Open Send sur la fonction **send** signale l'absence du port (désinstallé entre le **open** et le **send**)