Myostat Motion Control Inc .
Cool Muscle 1
RT3 Application Note

General Notes for Cool Muscle Language

1. General Notes

1. Using Comments

Comments are allowed in CML code and are indicated with a single forward slash '/'. There are a few rules regarding comments:

Comments should be on their own line
Comments are not allowed inside program banks or logic banks.
Comments should not exceed 40 characters from '/' to the carriage return.

Examples:

Valid Comments	Invalid Comments

/This comment is not more
/than 40 characters long
/P1 sets the bottom position
P1.1=1000
S1.1=10
A1.1=5
	/This comment is more than 40 characters long
	P1.1=1000 /P1 sets the bottom position
	S1.1=10
	A1.1=5

/Bank1 will move to the bottom position **B1.1** **A1.1,S1.1,P1.1** **END.1**	B1.1 /set acc, speed and position A1.1,S1.1,P1.1 /move to the bottom position END.1

2. Calculating Speed (pulses/sec ←→RPM)

Speed values (S0, S1, etc.) are set in the motor in pulses/second with a speed unit defined in K37. Speed units are 100p/s, 10p/s or 1p/s depending on the K37 value.

The general calculation to from pulses/second to RPM is:

$$\text{RPMrevolutionminute}=\text{SPEED}\times\text{SpeedUnitpulsessecond}\times\text{60secondminute}\div\text{RESOLUTIONpulsesrevolution}$$

$$\text{SPEED}=\text{RPMrevolutionminute}\div\text{60secondminute}\times\text{RESOLUTIONpulsesrevolution}\div\text{SpeedUnitpulsessecond}\ ($$

Example 1 (RPM→ SPEED):

K37=3 which defines a speed unit of 100 pulses/second and a resolution 1000pulses/revolution. If we want to run at 120 rpm this gives is an S value:

$$\text{SPEED}=\text{RPMrevolutionminute}\div\text{60secondminute}\times\text{RESOLUTIONpulsesrevolution}\div\text{SpeedUnitpulsessecond}$$

$$\text{SPEED}=120\div\text{60secondminute}\times\text{1000pulsesrevolution}\div\text{100pulsessecond}$$

SPEED=120÷6

SPEED=20

So for the case of K37=3 you can easily always divide by six to get RPM. E.g. S = 1500rpm/6 = 250.

Example 2 (SPEED→RPM):

K37=30 which defines a speed unit of 10 pulses/second and a resolution of 50000 pulses/revolution. If we have set an S0=33500 what is the RPM equivalent?

RPMrevolutionminute=SPEED×SpeedUnitpulsessecond×60secondminute÷RESOLUTIONpulsesrevolution

RPMrevolutionminute=33500×10pulsessecond×60secondminute÷50000pulsesrevolution

RPMrevolutionminute=402

3. Motor ID

Motors on a network auto ID themselves according to their position on the network. A single motor will always be assigned ID 1.

When a motor is powered up you will see the ID in the power up information.

E.g: “ID1 :CM1v3.12CH.5 #01236”

In this case the motor has ID1, firmware version 3.12 and hardware version 5. It has an internal serial number 01236.

The ID is assigned during programming by appending a “.ID” at the end of the command or register name.

Example CML Code:

```
/sets motor 1 position 1 to 1000.
P1.1=1000
/sets motor 2 speed 3 to 56
S3.1=56

/write a basic bank in motor 1
B1.1
A1.1,S1.1, P1.1
END.1

/write a basic bank in motor 2 calling motor 3.
/Note that the ID is set back on the END.2 and not assumed to be 2.
B1.2
```

```
A1.3,S1.3,P1.3
END.2

/set output 2 on motor 1 on
O2.1
/set output 2 on motor 5 off
F2.5
```

It is good practice to always append the ID on the end of all commands however it is not necessary. Once a motor ID has been called it will be the default until it is changed.

4. Switching off RS485 mode – ‘{#’

K62 sets the RS485 node ID. If K62=0 RS485 is off. If K62 is set to any other number that number is the node ID.

The motor is in RS485 slave mode if it is streaming ‘{‘ with a number following it on power up. The motor is requesting that it have communication access as it has something to report. To switch back to standard serial we need to:

1) give the motor communication access,
2) let it report its message,
3) gain communication access again
4) set K62=0

The process to do this is the following

1) Send the motor bank its request. I.e if the motor is streaming "{1" send back "{1". If it is send "{56" send back "{56".
2) At this point the motor will send some information (power up ID position, etc) and then "{0" to close communication.
3) To gain bank communication access we send the RS485 ID again. I.e. "{1" or "{56" as in the above examples.
4) Send "K62=0" to switch off RS485.

If there is a logic bank or program bank running that is reporting information constantly timing is critical. As soon as the {0 is received from the motor we must send bank {ID. If we don't and the motor has more to report it will start streaming the communication request again.