

# Reliance Cool Muscle

This document will assist a new user in setting up and programming a Reliance Cool Muscle using CoolWorks Lite.

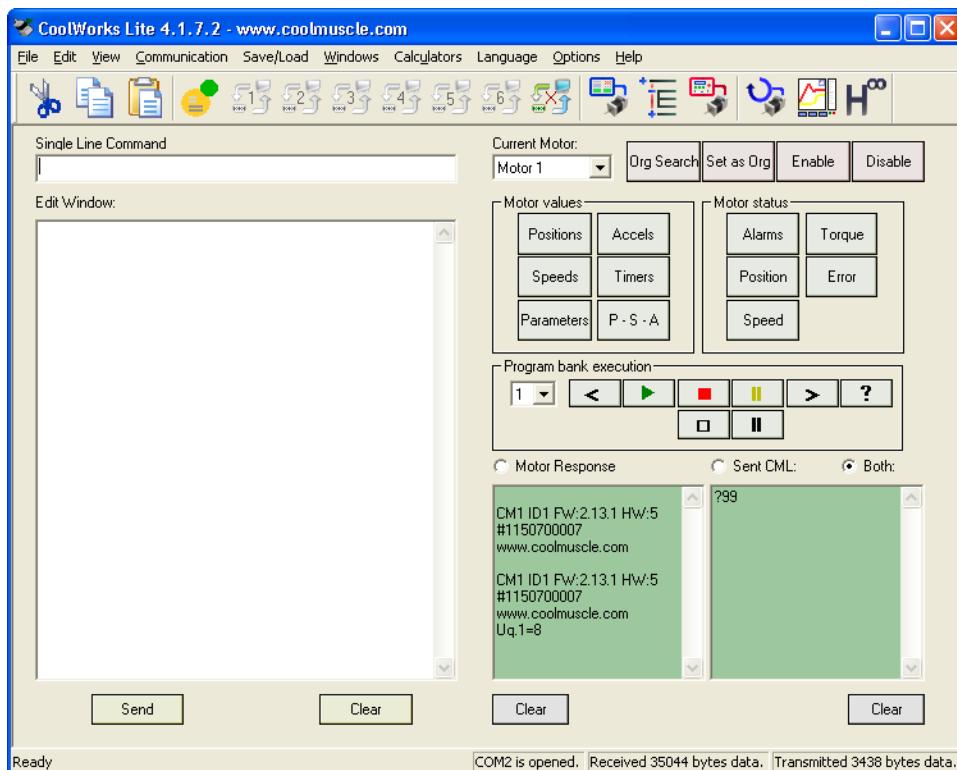
There are three different types of cool muscle motors. These are P type (motor is controlled with pulses on digital inputs), C type (motor is controlled via PC or its own internal PC. There is a possibility of upfront programming and in runtime we just call these programs.) and V type (we control motor via analog input).

## C type

### Programming tool Cool Works Lite

With Cool Works Lite we can modify K parameters, set speed, acceleration and position of the motor. Any program that we write it can be saved in a bank and in runtime we can call this bank. Using K parameters we setup motor behaving like e.g. way of response on the input signals, input/output functions, motor resolution (ratio pulses/revolution) etc.

We can setup up to 25 different positions, 15 speeds, 8 accelerations and 7 timers per motor. Command can be sent to the motor via entering them to Single Line Command (in continuation SLC) in this case they are executed immediately. If we write in SLC for example P1=1050 it will be transferred to the motors internal memory when we press enter key. In SLC we can also enter movement commands like |1 – this means that motor goes to its origin.



Picture 1 CoolWorks Lite main window

In Edit window (in continuation EW) we can modify parameters and write programs which are saved in banks. But the main difference between SLC were commands are sent immediatley when we press enter in EW its content is sent to the motor when we press button Send.

**IF WE WANT THAT BANKS WILL RUN THEN THE MOTOR HAS TO BE SET AS A C TYPE! This can be checked with parameter K16 which has to be 2 or 4 if we're using motor with RT3 firmware.**

**Example 1:** Let the motor to rotate with speed S=100, acceleration A=50 and position is increase every time for 250 pulses P+=250, this is written to EW.

```
S=100  
A=50  
P+=250  
^
```

With command ^ we execute motors movement in dynamic mode. This program is executed only once, if we want to execute it again we have to send it to the motor again. When we press Sent button the motor moves with defined speed acceleration to position P+=250. With command P+=250 motor moves for 250 impulses considering to its current position. If we press button Sent a few times the motor will move every time for 250 pulses forward from current position. If we used »-« instead of »+« the motor would then move to opposite direction than before.

**Example 2:** Example of a program where we use banks. First we define speeds, accelerations, positions and timers. In this bank there is executed the following procedure : when we call the bank motor goes with S1 and A1 to position P1, when he comes to this position it waits there that timer T1 ends. Then it goes to position P2 using the same speed and acceleration than before.

```
S1=200  
A1=45  
P1=1000  
P2=0  
T1=1000  
  
B1  
S1,A1,P1  
T1  
P2,T1  
X3  
$
```

We could also define S2 and A2, in this case the program would look like:

```
S1=200  
S2=250  
A1=45  
A2=10  
P1=1000  
P2=0  
T1=1000
```

```
B1
S1,A1,P1
T1
S2,A2,P2
T1
X3
$
```

Multiple repeating is enabled with X# (#- number from 0 to desired number of repeats, 0 means infinite loop.), while repeating the whole bank executes so many times as we set its number, for instance we set X3 so this means that the bank will repeat three times. With command \$ the program is saved to the motors eeprom.

**Example 3:** Inside of the bank we can during its activity call another bank. There are two different ways of calling another bank C (Call) and J (Jump). The difference between these commands is that if we use C when the called bank ends it returns to main bank and continues, but if we use J the program stops in a bank to which we jumped.

An example where we use command C, when bank B2 executes we return to bank B1 and it continues with next command after the jump.

```
S1=100,S2=250,S3=40
A1=60,A2=100,A3=44
P1=500,P2=250,P3=4000,P4=14000
T1=200,T2=1000,T3=10000
```

```
B1
S1,A1,P1
T1
C2
T1
C3
T2
X2
```

```
B2
S2,A2,P2
T1
```

```
B3
S3,A3,P3
T1
```

```
$
```

Here is an example using command J, when it jumps to bank B2 it executes B2 to its end and then stops. We added to bank B2 jump to bank B1 so when B2 ends it jumps to the first line of B1, this way we created an endless loop.

S1=100,S2=250  
A1=60,A2=100  
P1=1000,P2=2500  
T1=200,T2=1000

B1  
S1,A1,P1  
T2  
J2  
X2

B2  
S2,A2,P2  
T2  
J1

\$

In this case command X2 has no influence to program execution because it's never executed. Command J always goes to the first line in a bank. In case that we used C instead of J we would return to bank B1 to next command after jump.

## P type

In this mode the motor is controlled using digital inputs, K16 has to be set to 0. With K parameters we set up how are we going to control it. There are two possibilities, the first one is manual control (as long there is a signal present on input as long the motor moves with defined speed. The speed is defined using parameter K49) and the second one is controlling using impulses (with each impulse motor moves for defined number of impulses. Number of impulses is defined with parameter K50).

**Example 1:** We set K parameters K27=3400 (with input 4 we will move motor clockwise, using input 3 the motor will turn in an opposite direction) and K49=100 (the speed with which the motor will turn). When the signal will be present on input 4 the motor will start to move with speed that we defined using parameter K49. The same holds for input 3 just that the motor will turn in an opposite direction.

**Example 2:** We set K parameters K27=0000, K28=8900 (first we have to turn off manual control and switch to impulse control. When there is a pulse on input 4 motor turns in clockwise direction and when the pulse is present on input 3 the motor turns in an opposite direction) and K50=50 (with this parameter we set for how many pulses the motor will move).

## V type

In this mode the motor is being controled via analog input, K16 has to be set to 1. With parameter K38 we can chose between speed control and position control. Also the parameter K64 has to be set to 9 because then the motor works in V type.

**Example 1:** First we will look at speed controll. So we have to set K38=0, K64=9 and also maximum speed with K40. If the input voltage is 0V then motor turns with maximum speed in contrary clockwise direction. When voltage rises speed decreases and when it reaches 2,4V motor stops. It stops when the voltage is between 2,4V to 2,6V. If voltage is greater than 2,6V motor starts to turn in a clockwise direction and if voltage increases so does speed. When the input voltage is 4,8V the motor turns with maximum speed.

**Example 2:** We will use position control, so we have to set K38=1, K64=9 and final position with K41. When the input voltage is 4,8V the motor will move to final position that was set with K41. If voltage level starts to decrease towards 0V motor turns its starting position, at the moment when the voltage is 0V motor is in his starting position.

## **Reseting CoolMuscle**

When we reset the motor we have to be carefoul to chose the right Ksetting file. In his name has to be letter H infront! File structure that we are going to use has to be: H then your motor type and its dimension and at last firmware version, e.g.: H23L\_RT3. Ksetting Files with a letter C infront are only for markets in Asia!!!

## **Switching from P - type to a C – type**

In ‘Single Line Command’ you have to enter in this order:

- w=526
- K16=2 (2 - C type)
- ?K16 (testing if motor has stored new value).

After each entry you have to press enter! At last you should power down the motor and then back on. Then try to run a bank.

The procedure to switch back to P – type is the same except K16=2 you use K16=0.

Available values for K16 are :

- 0 sets motor to a P – type
- 1 sets motor to a V – type
- 2 sets motor to a C - type