ALARM & RESET HANDLING WITH TRAJEXIA

OYMC-MAC - Mechatronics Applications Center

Created : CR **Date :** May 5, 2006

This document summarizes how Trajexia with **fw 1.6487** behaves in front of reading alarms from a Mechatrolink servo and how handles the reset in all possible situations.

At the same time explains the requirements to make this sequence more coherent and user friendly for the application engineer.

ALARM READING.

HOW IS WORKING NOW:

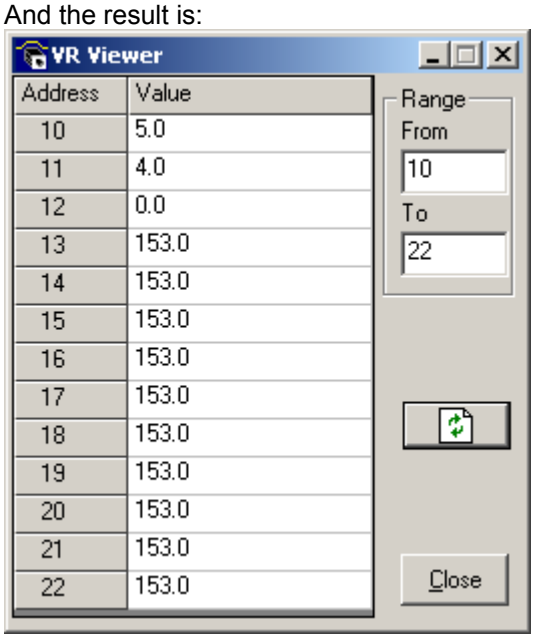
Now we have the next command to read alarms from a Mechatrolink servo. This command has been developed for development purposes:

MECHATROLINK(<Unit>,53,<Axis_n>,<setting>,<VR>)

Where:

<Unit>	is the TJ1-ML16 unit number (order position in the rack). 0 in our example.
53	is the Mechatrolink instruction "Read Alarm". Corresponds with the ALM_RD mechatrolink command.
<Axis_n>	0 in our example
<setting>	0 reads the current alarms =1 reads the Alarm History. We will use 0 in this document.
<VR>	Is a pointer to the first VR where the information from the servo is stored.

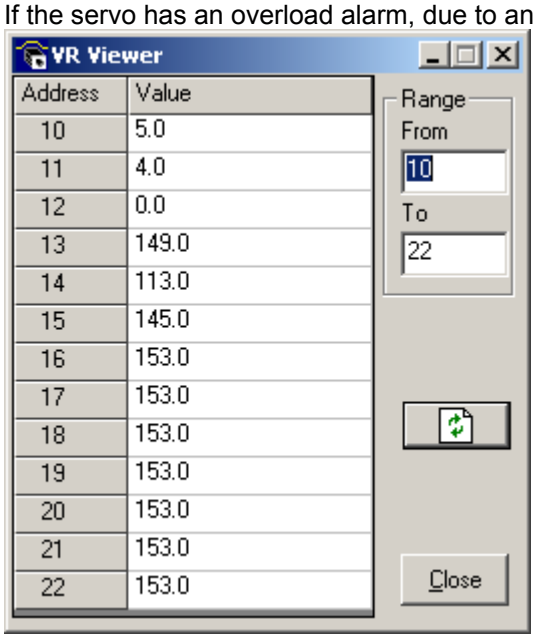
EXAMPLE:
In a healty servo we execute:
>>MECHATROLINK(0,53,0,0,10)
And the result is:



Where:

VR(10)=5	Is the code for ALM_RD command
VR(11)=4	???
VR(12)=0	???
VR(13)=153	Latest alarm code. 153=99Hex means "No alarm or warning"
…	
VR(22)=153	10^{TH} Latest alarm

If the servo has an overload alarm, due to an excessive cycle or load:



In this case the sequence of alarms is next:

VR(13)=149 149=A.95 Mechatrolink command could not be executed.
VR(14)=113 113=A.71 Overload alarm
VR(22)=145 145=A.91 Overload warning

The event sequence has been next.

- Previous to the alarm we have the warning A.91
- Then the alarm occurs A.71
- But in the same cycle, we have a position command from mechatrolink that could not be executed because the servo is faulty so we have A.95

The good part of this command is that you have the complete event sequence. The bad part is that it is difficult to analyze and not friendly at all

REQUEST FOR ALARM READ

In most situations you are interested to read just the current alarm, not the complete sequence. The request is to implement the read only parameter or command

SERVO_ALARM, SERVO_ALARM_READ or similar

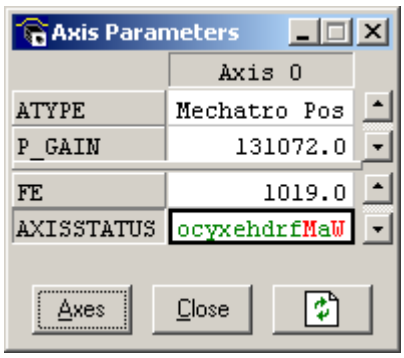
That contains the current alarm from the servodrive, ignoring the warnings. In the example above, the value of SERVO_ALARM will be 99Hex if the servo is healthy and 71Hex when it has the overload alarm. Note: the warnings have values from 90Hex to 98Hex. Check the JUSP-NS115 manual for details.

The idea is to keep MECHATROLINK(x,53,…) not documented and just open it to advanced users case by case, and Omron engineers.

RESETTING SERVODRIVE ALARMS.

HOW IS WORKING NOW:

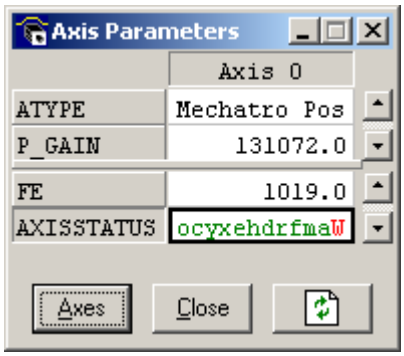
A Servodrive alarm is shown in AXISSTATUS setting bit 3=1 (M in the software tool). At the same time, this creates an Axis error.



The sequence to reset this is:

>>DRIVE_CLEAR

The alarm in the servodrive is resetted and the corresponding bit (BIT 3) in AXISSTATUS is resetted too



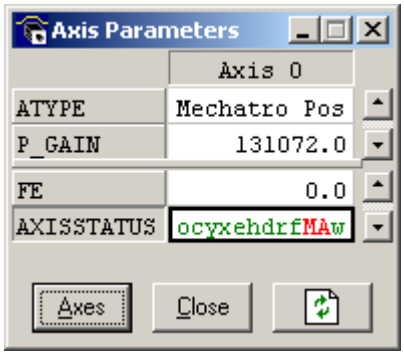
The Axis error remains so you have to execute DATUM(0) afterwords.

RESETTING MECHATROLINK COMMUNICATION ALARMS.

HOW IS WORKING NOW:

A mechatrolink communication alarm is shown in AXISSTATUS setting bit 2=1 (A in the software tool). At the same time, this creates an Axis error and, depending on the kind of failure, a servodrive error too (A.E6 in case the mechatrolink cable is disconnected).

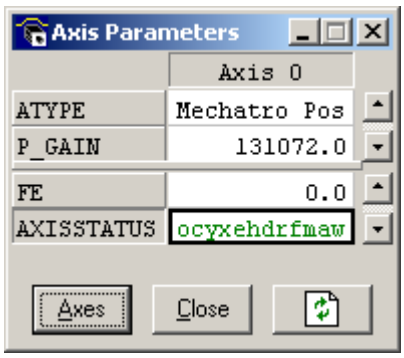
The message “Err”, “U.00” is shown in the TrajeXia display



The way to reset a Mechatrolink communication alarm is to execute:

MECHATROLINK(0,0)

The Mechatrolink alarm is resetted and the corresponding bit (BIT 2) in AXISSTATUS is resetted too. If there is a servodrive error, it is resetted too.



The axis error dissapear automatically so it is not necessary to execute DATUM(0). Nevertheless the display in Trajexia continues showing “Err”, “U.00” although you can work normally with it.

REQUEST FOR MECHATROLINK ALARM RESET

The display in TrajeXia must return to the normal status (OFF, run, …) when the error disappear.