

Global Variables Description

Related articles

[Global Variables](#)

A global variable is a variable which is used by the myCNC application to store some information about a particular process, flag or setting. A full list of global variables is available [here](#).

The values stored in these global variables are defined in the myCNC profile that is being used, within the `cnc-variables.xml` file. This file contains a list of CNC and global variables that the program utilizes, as well as their values and minimum/maximum possible values.

The myCNC program auto-updates the `cnc-variables.xml` file when the program is closed, to save the current state of the global variables.

NOTE: Only the files that are already in the `cnc-variables` list are updated. When writing a variable (via the `gvarset` command, for example) from the Software PLC, the variable is added to the global variables list. If the variable is simply being read, or used to display some value changes on the screen, it will not be added to `cnc-variables.xml` file and the value will be lost on program restart - this is highly relevant for custom gvariables (#3500-3900).

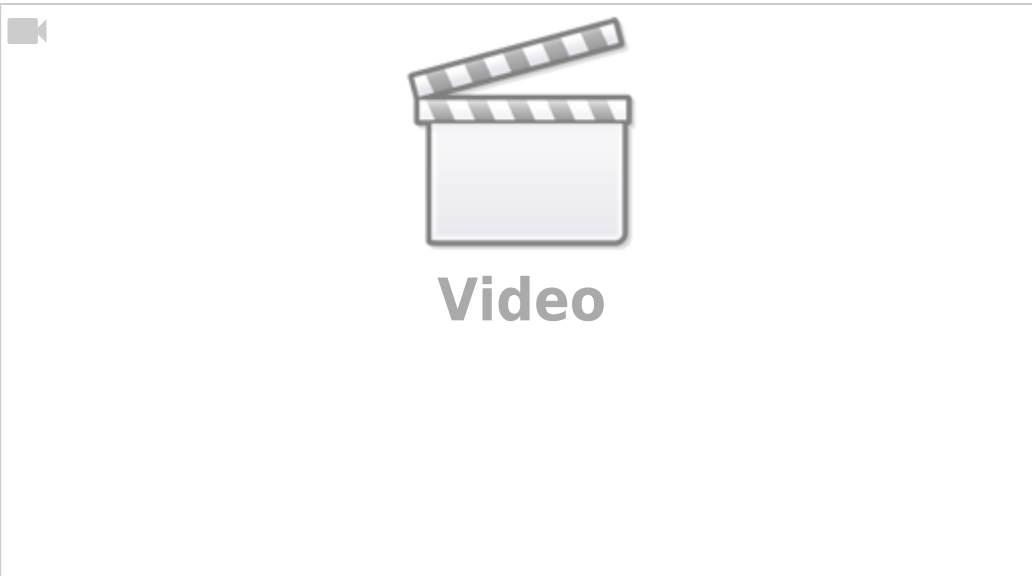
The min/max values are useful to set up once during the machine setup to provide realistic bounds for the range of a certain global variable.

The syntax then is the following:

```
<value name="motion-acceleration-z" min="1" max="20000"
step="50">1000</value>
```

where:

- **name** is the variable name
- **min** and **max** represent the bounds of the range
- **step** represents the step value for when the variable value is increased or decreased (useful for on-screen buttons)
- in this example, **1000** represents the current value of the aforementioned variable



List of global variables

5521

Writing "1" to register #5521 turn off Hardware Limit Sensors control temporarily. For example, it can be used if Home sensor is used as limit sensor as well. In this case Hardware Limits control should be disabled temporarily during Homing procedure to not abort Homing when Limit sensor triggered. At the end of Homing procedure Hardware Limits control should be enabled again, however it will be made automatically after procedure running finished.

Writing Constant value to Register can be made with G10 L80 code. Example -

```
G10 L80 P5521 Q1 (Write 1 to #5521)
```

5525

Writing "1" to register #5525 turn off Software Limits control temporarily. For example, it can be used for Homing procedure when Machine position is not yet defined or for Auto Tool change procedure if Tool Changer position is situated outside of working area. Software Limits control can be enabled by resetting the register (writing "0"). The register will be reset automatically when g-code running finished.

Writing Constant value to Register can be made with G10 L80 code. Example -

```
G10 L80 P5525 Q0 (Write 0 to #5525)
```

Writing a value larger than 2 will initialize a timer after the macro containing Global Variable #5525, delaying the subsequent turning on of Software Limits control. This is useful in situations where a prolonged procedure requires a longer time within which the Software Limits are turned off than the macro containing P5525 Q1 itself.

Example of code:

```
G10 L80 P5525 Q15 (15 sec timer)
```

60010

Value written to this register is used as Modbus ID of device to communicate with

60011

Writing to this register change Modbus protocol.

```
"0" - Modbus/RTU,  
"1" - Modbus/ASCII
```

60012

Writing to this register will change RS485/Modbus speed. Available speeds are

```
1200, 2400, 4800,  
9600, 19200, 38400,  
57600, 115200
```

60013

Change UART connection parameters: number of bits (8 or 7), parity (none, odd, even), number of stop bits (1, 2). Data comes in 3 low nibbles. The lowest is stop bits, then parity, then number of bits. Example:

```
0x801=8,N,1  
0x712=7,0,2  
0x822=8,E,2
```

60019

Writing to register will latch the value in shadow register. Value from shadow register will be used on **write** operation

60020

Writing to register will latch **Address to read** in shadow register

60030

Writing to register will send value from shadow register to Modbus device to address given in written value

60031

Read from this register will send read inquiry to Modbus device (PLC controller will be in till Register value received from Modbus device). Writing to this register will send ready inquiry to Modbus device. The value written is used as Register address to read

64200

Writing an ID number to this register (specified in the XML files as id="NUMBER") will show a widget. Similarly, the global variable 64201 allows the user to hide a widget by using a specific ID number.

An example of such a widget can be controlled via a Software PLC:

```
main()
{
    v5528=gvarget(5528);

    do
    {
        v=gvarget(5528);
        if (v!=v5528)
        {
            if (v!=0)
            {
                gvarset(64200,200);
            }else
            {
                gvarset(64201,200);
            };
            v5528=v;
        };
    }while(1);

    exit(99);
};
```

In the example above, the system will continuously monitor (through a usage of a do-while loop) the state of global variable #5528 (Simulation Mode), and will open/hide a widget on the screen whenever the state of that global variable changes. The widget is then described by the following lines in an .xml file within the profile folder:

```
<gitem where="xp" name="x-trial-speed" id="200" position="750;570"
width="190" height="100" type="myitems" bgColor="##b-main" hidden="0" />
```

Here, the widget (of the type myitems) is given an ID in addition to a name, with the line id="200", therefore corresponding to the gvarset(64200,200) line from the earlier software PLC.

Video illustrating the usage of global variable 64200:



Video

Global Variables Table

From:

<http://docs.pv-automation.com/> - **myCNC Online Documentation**

Permanent link:

http://docs.pv-automation.com/mycnc/global_variables_description

Last update: **2022/08/04 10:58**

