

# Independent Pulse Generator

myCNC controllers Axis B can be switched to independent pulse generator output.

Independent Pulse Generator was added to firmware dated July 20, 2018.  
At the moment the firmware is available as the "Testing" version at the "Support" widget.

## High level access to the pulse generator

There are a number of global array registers to access to the independent pulse generator

| Register Name            | Address | Description                                                                                                     |
|--------------------------|---------|-----------------------------------------------------------------------------------------------------------------|
| GVAR_GENERATOR_FRQ_RAW   | 8130    | Pulse Generator RAW frequency value, [units]<br>1 unit = 0.000736 Hz                                            |
| GVAR_GENERATOR_ACCEL     | 8131    | Generator Acceleration, [units]<br>1unit = 0.736 1/c2                                                           |
| GVAR_GENERATOR_FRQ_RATIO | 8132    | Generator Frequency Ratio                                                                                       |
| GVAR_GENERATOR_FRQ       | 8133    | Generator Frequency.<br>Frequency*Ratio value is sent to the Generator and saved in the RAW frequency register. |

Originally the Pulse generator was supposed to use as Coolant control. Global register GVAR\_PLG\_COOLANT\_STATE (#7372) is used to detect the Current State of the Pulse generator.

If Generator Frequency register (#8133) is changed -

- RAW value is calculated and stored in the RAW register #8130
- If Coolant Register is NOT zero, the RAW value is sent to the myCNC controller to update current frequency. Changing the RAW register in myCNC controller takes effect immediately.
  - If the RAW value is zero, the generator is stopped (with given acceleration).
  - If the RAW value is not zero, the generator is started (or current frequency changed accordingly) with given acceleration.

If the RAW register (#8130) is written directly, the value will be sent to the Frequency Generator despite on Current Coolant state (#7372)

Global variable registers can be written in either Hardware or Software PLC.

Q: Why the Frequency Ratio need?

A: Internal frequency unit has no sense for a normal user. It is convenient to set up the ratio and has the Frequency value in a unit usable for a user. Depends on Frequency generator application the unit might be very different.

It can be [1Hz] if you need a simple frequency generator,  
or [ml/hour] for Coolant control  
or [rpm] for Spindle speed through pulse-dir servo controller.

## How to use the pulse generator

The first application we used the Pulse Generator was a Coolant control base on a stepper driver.

### Pulse Generator settings in User Widget

We added Pulse generator settings to a User Widget of "1280M5" profile. We are going to test it and distribute these settings for other profiles and other applications as well later.



The programming of Pulse the Coolant widget is shown below.

"user-m5.xml" configuration file contains a code for the User Widget. There are lines to add Coolant widget itself and include a xml file with Coolant widget content

```
<gitem where="user-widget" name="coolant-widget"
  bgColor="##b-widget" type="myitems"
  position="590;300" width="490" height="150" />

<include>user-coolant.xml</include>
```

The lines contain "coolant-widget" definition (a background color, position on the parent widget, a widget size and a widget type (myitems)).

"user-coolant.xml" code is shown below

## user-coolant.xml

```
<mycnc-configuration version="1.0">
<gitem where="coolant-widget"
  position="0;0" width="490" height="30" labelWidth="490" type="label"
  labelFgColor="white" labelBgColor="##f-display" labelFontSize="18"
  labelFontStyle="bold" >
  <message>Coolant</message>
  <message_ru>Охлаждение</message_ru>
  <message_vn>Chất làm mát</message_vn>
</gitem>

<gitem where="coolant-widget"
  position="10;40" width="220" height="60" displayWidth="90"
  labelWidth="130"
  type="bdisplay" action="item:cnc-gvariable-8133" name="display-cnc-
  gvariable-8133"
  bgColor="##b-display" fgColor="##f-display" fontSize="18"
  fontStyle="bold" format="%5.1f"
  K="1" labelFontSize="16" labelFontStyle="bold" labelFgColor="white" >
  <message>Rate, ml/hour</message>
  <message_ru>Расход, мл/час</message_ru>
</gitem>

<gitem where="coolant-widget"
  position="280;40" width="180" height="60" displayWidth="90"
  labelWidth="90"
  type="bdisplay" action="item:cnc-gvariable-8132" name="display-cnc-
  gvariable-8132"
  bgColor="##b-display" fgColor="##f-display" fontSize="14"
  fontStyle="bold"
  format="%d" K="1" labelFontSize="16" labelFontStyle="bold"
  labelFgColor="white" >
  <message>Ratio</message>
  <message_ru>Коэффициент</message_ru>
</gitem>

</screen>
```

The code contains 3 parts

- the widget label set up
- the frequency setup
- the Ratio set up

It's supposed operator no need to change frequency acceleration and this setting is hidden from an operator. The acceleration can be set up in the Software or Hardware PLC for example.

## Pulse Generator settings in the Software PLC

The rate, ratio and acceleration can be set up in the Software PLC as well.

“HANDLER\_INIT.plc” procedure is started just after the configuration is sent to the myCNC controller. A few lines to set the Frequency generator can be added there.

### HANDLER\_INIT.plc

```
main()  
{  
  gvarset(60000,1); //run Servo ON procedure  
  
  gvarset(8131, 8000); //set Frequency acceleration  
  gvarset(8132, 1359); //set Ratio  
  gvarset(8133, 0);    //Off the Generator.  
  
  exit(99);  
};
```

## (Coolant) Pulse Generator control through Hardware PLC

Function coolant\_motor\_start() is added to “mill-func.h” include file

### mill-func.h

```
coolant_motor_start()  
{  
  timer=10;do{timer--;}while(timer>0);  
  
  gvarset(8131,1000000); //acceleration  
  timer=10;do{timer--;}while(timer>0);  
  
  x=gvarget(8133); //get the speed (frequency)  
  k=gvarget(8132); //get the ratio  
  
  x=x*k; //calculate the RAW frequency  
  gvarset(8130,x); //send the raw frequency to the register  
  timer=30;do{timer--;}while(timer>0); //wait a time for the frequency  
  value to be delivered  
};
```

M08.plc procedure which starts the coolant motor would be

### M08.plc

```
#include pins.h
#include mill-func.h

main()
{
    gvarset(7372,1);
    portset(OUTPUT_FLOOD); //
    coolant_motor_start();
    exit(99);    //normal exit
};
```

A procedure M09.plc to stop a coolant motor is simpler. Just need to write “0” to the raw frequency register.

#### M09.plc

```
#include pins.h
main()
{
    gvarset(7373,0);
    gvarset(7372,0);

    portclr(OUTPUT_FLOOD);
    portclr(OUTPUT_MIST);

    gvarset(8130,0); //stop the pulse generator
    timer=30;do{timer--;}while(timer>0); //wait a time for the frequency
value to be delivered
    exit(99);    //normal exit
};
```

## Low level CNC registers to control independent pulse generator

This is for records only. Users don't have to use low-level access.

| Register Name       | Address | Description                                           |
|---------------------|---------|-------------------------------------------------------|
| EXT_GENERATOR_SPEED | 225     | Generator Frequency, [units]<br>1 unit = 0.000736 Hz  |
| EXT_GENERATOR_ACCEL | 226     | Generator Acceleration, [units]<br>1unit = 0.736 1/c2 |

Independent pulse output can be used for -

- Spindle control based on Servo motor and pulse-dir driver
- stepper motor based coolant system.

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

Permanent link:  
[http://docs.pv-automation.com/mycnc/independent\\_pulse\\_generator?rev=1535911608](http://docs.pv-automation.com/mycnc/independent_pulse_generator?rev=1535911608)

Last update: **2018/09/02 14:06**

