

Jog from PLC

Note: This article is an extension of the original [PLC: Jog from PLC](#) section. For more info on PLC programming, see the original article.

The Jog from PLC feature in myCNC allows motion control commands to be executed directly from within PLC programs. This enables dynamic axis control (changing direction, speed, or enabling/disabling movement) without stopping machine motion. Such functionality is especially useful for multi-axis homing or probing operations.

Jogging motion is managed using the following global variables, which define acceleration, speed, and direction.

Variable Name	ID	Description
GVAR_G0PLC_SPEED	8630	Jog feedrate
GVAR_G0PLC_TIME	8631	Jog acceleration time in ms
GVAR_G0PLC_SPEED_UNITS	8632	Jog speed
GVAR_PLC_JOGSPEED_UNITS	8634	Jog speed + axis mask (axis << 24 speed)
GVAR_PLC_JOG	8635	Jog direction control bitmask

Example Implementations

Example 1: X-Axis Homing

```
#include pins.h
#include func_homing.h

main()
{
    gvarset(5539,1); //switch to fast g0moveA implementation
    gvarset(8631,10); //acceleration time 100ms = 0.1s
    speed=1000;
    gvarset(8634,((1<<24)|speed)); //Jog speed for X
    gvarset(8634,((2<<24)|speed)); //Jog speed for Y

    forward=1;
    backward=(1<<8);
    sensor=INPUT_HOME_X;

    probe();
    exit(99);
}
```

This PLC:

- Performs homing along the X-axis.
- Enables fast motion mode (GVAR 5539).
- Sets acceleration time (GVAR 8631) and jog speed (GVAR 8634).
- Defines direction bits for X-axis (forward / backward).
- Uses INPUT_HOME_X as the active home sensor.
- Calls the probe() routine to move toward and retract from the home switch (see the func_homing.h code below).

Example 2: Y-Axis Homing

```
#include pins.h
#include func_homing.h

main()
{
    gvarset(5539,1); //switch to fast g0moveA implementation
    gvarset(8631,10); //acceleration time 100ms = 0.1s
    speed=1000;
    gvarset(8634,((1<<24)|speed)); //Jog speed for X
    gvarset(8634,((2<<24)|speed)); //Jog speed for Y

    forward=(2<<8);
    backward=2;
    sensor=INPUT_HOME_Y;

    probe();
    exit(99);
}
```

- Performs homing along the Y-axis.
- Uses same motion parameters as the X-axis script above.
- Reads the Y home sensor (INPUT_HOME_Y).

Example 3: Simultaneous XY Homing

```
#include pins.h
#include func_homing.h

main()
{
    gvarset(5539,1); //switch to fast g0moveA implementation
    gvarset(8631,10); //acceleration time 100ms = 0.1s
    speed=1000;
    gvarset(8634,((1<<24)|speed)); //Jog speed for X
    gvarset(8634,((2<<24)|speed)); //Jog speed for Y
```

```

forward1=1;
forward2=2;
backward1=(1<<8);
backward2=(2<<8);

sensor1=INPUT_HOME_X;
sensor2=INPUT_HOME_Y;

probe_xy();
exit(99);
}

```

- Performs simultaneous homing of X and Y axes.
- Defines independent direction masks for each axis.
- Uses both X and Y home sensors.
- Calls probe_xy() to handle dual-axis probing until both sensors are reached.

Example 4: Probe and Jog Control Functions (func_homing)

func_homing.h

```

probe()
{
    gvarset(8635,forward); //Jog forward
    timer=60000;
    do
    {
        timer--;
        if ((timer&0xff)==0) {    gvarset(8635,forward);    };
        a=portget(sensor);
        if (a!=0) { timer=0;};
    }while(timer>0);

    gvarset(8634,((1<<24)|(speed/8))); //Jog speed for X
    gvarset(8634,((2<<24)|(speed/8))); //Jog speed for Y
    gvarset(8635,backward); //Jog backward

    timer=60000;
    do
    {
        timer--;
        if ((timer&0xff)==0) {    gvarset(8635,backward);    };
        a=portget(sensor);
        if (a==0) { timer=0;};
    }while(timer>0);

    gvarset(8635, 0); //Jog Stop
    do{ a=gvarget(6060); }while(a!=0x4d);
}

```

```
};

probe_xy()
{
    dir1=forward1;
    dir2=forward2;
    gvarset(8635,dir1|dir2); //Jog forward
    timer=60000;
    do
    {
        timer--;
        if ((timer&0xff)==0) {          gvarset(8635,dir1|dir2);    };
        if (dir1)
        {
            a=portget(sensor1);
            if (a!=0)
            {
                dir1=0;
                gvarset(8635,dir1|dir2); //Jog forward
            };
        };

        if (dir2)
        {
            a=portget(sensor2);
            if (a!=0)
            {
                dir2=0;
                gvarset(8635,dir1|dir2); //Jog forward
            };
        };
    }while(dir1|dir2);

    dir1=backward1;
    dir2=backward2;

    gvarset(8634,((1<<24)|(speed/8))); //Jog speed for X
    gvarset(8634,((2<<24)|(speed/8))); //Jog speed for Y
    gvarset(8635,dir1|dir2); //Jog backward

    timer=60000;
    do
    {
        timer--;
        if ((timer&0xff)==0) {          gvarset(8635,dir1|dir2);    };
        if (dir1)
        {
            a=portget(sensor1);
            if (a==0)
            {
```

```
    dir1=0;
    gvarset(8635,dir1|dir2); //Jog forward
};

if (dir2)
{
    a=portget(sensor2);
    if (a==0)
    {
        dir2=0;
        gvarset(8635,dir1|dir2); //Jog forward
    };
};
}while(dir1|dir2);

gvarset(8635, 0); //Jog Stop
do{ a=gvarget(6060); }while(a!=0x4d);
};
```

probe()

- Sends a forward jog command.
- Monitors the assigned input sensor until triggered.
- Reduces jog speed (1/8 of initial) and reverses.
- Moves backward until sensor deactivates.
- Stops jogging and waits for motion completion (GVAR 6060 == 0x4D).

probe_xy()

- Executes dual-axis jog for X and Y simultaneously.
- Each axis stops independently when its home sensor triggers.
- Once both sensors are active, machine jogs backward at reduced speed until both are clear.
- Ends motion and waits for confirmation of stop.

The Jog from PLC mode allows smooth and flexible axis control without halting motion or interrupting the main program execution.

From:
<http://cnc42.com/> - myCNC Online Documentation

Permanent link:
http://cnc42.com/plc/jog_from_plc

Last update: **2025/11/03 13:24**

