

Software PLC examples

NOTE: The myCNC team recommends utilizing the examples provided in this manual (as well as other manuals in this documentation) as a starting point for your machine setup. When possible (and applicable), it is recommended to keep changes to a minimum. In general, using these examples as the basis for your PLCs/macro commands allows for an easier setup process.

General examples

- [How to add mandatory Homing after Emergency Button and-or Servo ready alarm](#)
- [Button to toggle select output pin with indication](#)
- [Oil Change counter](#)
- [Controller Peripherals Test - BV17](#)
- [Charge Pump](#)

Endless loop for G-code program

An endless loop can be made in Software PLC procedure. PLC procedure contains an endless loop. In the loop it checks the current status of myCNC controller, if the controller is in Idle mode and ready to run a new program, it reloads G-code file (if needed), reset current program pointer and starts running g-code.

PLC code is -

[ENDLESS.plc](#)

```
main()
{
    do
    {
        do{ a=gvarget(6065); }while(a!=0);    //wait Motion Controller ready
to accept new program
        gvarset(100001,1);                    //Reset G-code pointer to start
        timer=5; do{ timer--;}while(timer>0); //0.5 sec Pause
        gvarset(100002,1);                    //Start G-code
    }while(1);                                //endless loop

    exit(99);
};
```

If **Stop** button pressed, g-code will be stopped, the procedure will see Motion controller is ready for next loop entry and restart g-code again. This behaviour can be inappropriate for many applications.

A simple solution is to add global variable **single/auto** and check this variable in the loop.

[NOT_ENDLESS.plc](#)

```

main()
{
    do
    {
        do{ a=gvarget(6065); }while(a!=0);    //wait Motion Controller ready
to accept new program
        gvarset(100001,1);                    //Reset G-code pointer to start
        timer=5; do{ timer--;}while(timer>0); //0.5 sec Pause
        gvarset(100002,1);                    //Start G-code
    }while(gvarget(400)==0);    //endless loop, if #400==0, otherwise exit
from procedure

    exit(99);
};
//If variable #400 will be set to non-zero value, NOT_ENDLESS.plc
process will be finished

```

In case no need to exit from ENDLESS.plc and variable #400 should be monitored constantly and run g-code when #400 is set bu user -

ENDLESS2.plc

```

main()
{
    do
    {
        if (gvarget(400)!=0)                    //if #400!=0, run g-code,
otherwise continue to test #400
        {
            do{ a=gvarget(6065); }while(a!=0);    //wait Motion Controller ready
to accept new program
            gvarset(100001,1);                    //Reset G-code pointer to start
            timer=5; do{ timer--;}while(timer>0); //0.5 sec Pause
            gvarset(100002,1);                    //Start G-code
        };
    }while(1);    //endless loop

    exit(99);
};
//If variable #400 will be set to non-zero value, ENDLESS.plc process
will be finished

```

Endless loop with switch to left & right sides of working area

LEFT-RIGHT.plc

```
#define MAX_X 800
#define MAX_Y 800

main()
{
do
{

    auto=gvarget(501); //Automatic mode

    if (auto)
    {

        do { rect_not_ready=gvarget(7369); }while(rect_not_ready==0);

        left_right=1;
        xmax=gvarget(7350);
        if (xmax>MAX_X){ left_right=0;};
        ymax=gvarget(7351);
        if (ymax>MAX_Y){ left_right=0;};

        gvarset(502,left_right); //502 - shows both sides or single side
        LR=gvarget(500);

        if (LR==0) //left
        {
//            gvarset(9100,1); //show message #1
//            //do{ a=portget(11); }while(a==0); //press pedal, port 11
//            //do{ a=gvarget(6065); }while(a!=0); //press pedal, port 11
//            gvarset(9100,0); //clear the message

            portclr(10); //Vacuum Left OFF
            portset(9); //

            portset(11); //
            portset(22); //

//            gvarset(9101,1); //show message #2
//            //do{ a=portget(13); }while(a==0); //press pedal, port 13
//            gvarset(9101,0); //убрать

            portset(10); //Vacuum Left
            portclr(9); //
            timer=5; do{ timer--; }while(timer>0); //wait 0.5 seconds while
the vacuum is working

            portclr(11); //выключить упоры
            portclr(22); //

////
```

```
//ждать пока УП завершится
    do{ a=gvarget(6065); }while(a!=0);    //wait
////

do {
    do{ a=gvarget(6065); }while(a!=0);    //wait for the system to be
ready
    gvarset(100010,54); //Дать код G54
    timer=5; do{ timer--; }while(timer>0); //wait 0.5s for the
system to switch
        n=gvarget(5220); //gets the current coordinate system number
        } while(n!=1);    //while the current system is not G54

    do{ a=gvarget(6065); }while(a!=0);    //wait for the system to be
ready
    gvarset(100001,1); //go to the beginning of the program
    timer=5; do{ timer--; }while(timer>0); //wait 0.5s
    do{ a=gvarget(6065); }while(a!=0);    //wait for the system to be
ready
    gvarset(100002,1); //start the program
}else    //right
{
//    gvarset(9110,1); //show message #1
//    do{ a=portget(12); }while(a==0); //press pedal, port 11
//    gvarset(9110,0); //clear the message

    portclr(15); //Vacuum Left OFF
    portset(16); //

    portset(14); //
    portset(23); //

//    gvarset(9111,1); //show message #2
//    do{ a=portget(14); }while(a==0); //press pedal, port 13
//    gvarset(9111,0); //clear

    portset(15); //Vacuum Left
    portclr(16); //
    timer=5; do{ timer--; }while(timer>0); //wait 0.5 seconds while
the vacuum is working

    portclr(14); //
    portclr(23); //

////
    do{ a=gvarget(6065); }while(a!=0);    //wait
//wait until the control program is finished
////

do{
```

```

do{ a=gvarget(6065); }while(a!=0);    //wait until the program is
ready
    gvarset(100010,55);    //assigns G54 coordinate system
    timer=5; do{ timer--; }while(timer>0); //wait 0.5 seconds while
the coordinate system switch is happening
    n=gvarget(5220); //obtain the current coordinate system
number
} while(n!=2); //while the system is not G55

do{ a=gvarget(6065); }while(a!=0);    //wait while the processor
is busy
gvarset(100001,1); //return to the beginning of the program
timer=5; do{ timer--; }while(timer>0); //wait 0.5 seconds
do{ a=gvarget(6065); }while(a!=0);    //wait for the system to be
ready
gvarset(100002,1); //start the program

};

timer=20; do{ timer--; }while(timer>0);    //3 second pause
do{ a=gvarget(6065); }while(a!=0);    //wait until the program is
complete

    LR=LR^1;    //switch to the other side
    gvarset(500,LR);

}; //auto

}while(1);

exit(99);
};

```

Jog Step (0.001, 0.01, 0.1 1.0) Indication with external LED display

There are global variables which represent current jog step size

Variable Name	Variable Number	Description
GVAR_JOG_STEP_SIZE	5522	represents current jog step size (double)
GVAR_JOG_STEP_0_0001	7381	the value is "1" if Current Jog Step Size is "0.0001", otherwise is "0" (integer)
GVAR_JOG_STEP_0_001	7382	the value is "1" if Current Jog Step Size is "0.001", otherwise is "0" (integer)
GVAR_JOG_STEP_0_01	7383	the value is "1" if Current Jog Step Size is "0.01", otherwise is "0" (integer)
GVAR_JOG_STEP_0_1	7384	the value is "1" if Current Jog Step Size is "0.1", otherwise is "0" (integer)

Variable Name	Variable Number	Description
GVAR_JOG_STEP_1_0	7385	the value is "1" if Current Jog Step Size is "1.0", otherwise is "0" (integer)
GVAR_JOG_STEP_10	7386	the value is "1" if Current Jog Step Size is "10", otherwise is "0" (integer)

Software PLC procedure can be made to handle LED display according to "Current Jog Step Size".

1. The procedure contains an endless loop.
2. Variables a7382...a7385 keep previous state or "Current Jog Step Size"
3. There are testing "Current Jog Step Size" global variables values, switching binary output if variables were changed and store new states in the a7382...a7385 local variables

JOG_STEP_LEDS

```
main()
{

    a7382=gvarget(7382);
    a7383=gvarget(7383);
    a7384=gvarget(7384);
    a7385=gvarget(7385);

    do{

        if (gvarget(7382)!=a7382)
        {
            a7382=gvarget(7382);
            if (a7382==0){portclr(14);}else {portset(14)};};
        };
        if (gvarget(7383)!=a7383)
        {
            a7383=gvarget(7383);
            if (a7383==0){portclr(13);}else {portset(13)};};
        };
        if (gvarget(7384)!=a7384)
        {
            a7384=gvarget(7384);
            if (a7384==0){portclr(12);}else {portset(12)};};
        };
        if (gvarget(7385)!=a7385)
        {
            a7385=gvarget(7385);
            if (a7385==0){portclr(11);}else {portset(11)};};
        };

    }while(1);

    exit(99);
```

```
};
```

Switching an output ON for a certain period of time

```
#define OUTPUT_TIMER 15
main()
{
    count=0;
    do{
        a=gvarget(7373);

        if (a!=0)
        {
            count++;
            if (count<30) { portset(OUTPUT_TIMER); }
            else {
                portclr(OUTPUT_TIMER);
                if (count>6000) { count=0; };
            };
        }else
        {
            portclr(OUTPUT_TIMER);
            count=0;
        };
    }while(1);
    exit(99);
};
```

From:

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

Permanent link:

http://docs.pv-automation.com/plc/software_plc_examples

Last update: **2022/03/29 15:55**

