Spindle Speed control through +\-10V DAC channel (ET10)

ET10 control board contains 6 channel +\-10V 12bits DAC outputs. The outputs normally controlled through PID and used for closed-loop analogue servo drivers control. However if PID is disabled, +\-10V DAC output can be used as general purpose DAC.

Output level of DAC output can be controlled through “DAC Offset” register.

ET10BB breakout board contains inverted operational amplifier, so zero (0) value in “DAC Offset” register corresponds to +10V output voltage level, maximum value (0x3ff) corresponds to -10V voltage level.

There are number of ways to access to DAC Offset register.

PLC controller (Hardware)

DAC offset register is mapped to address 0x32 of Extern CNC Variables array.

```
#define EXT_ET5_DAC_OFFSET 0x32
```

16 bits value should be written to this register. low 12 bits is DAC value to be written high 4 bits is DAC channel

Message PLCCMD_SET_CNC_EXTVAR should be sent to myCNC control core from PLC to get access Extern CNC Variables, then 2 ms delay should be initiated to push the message from PLC to myCNC Core.

```
#define PLCCMD_SET_CNC_EXTVAR 1020
```

PLC code example for ET10\DAC Value programming

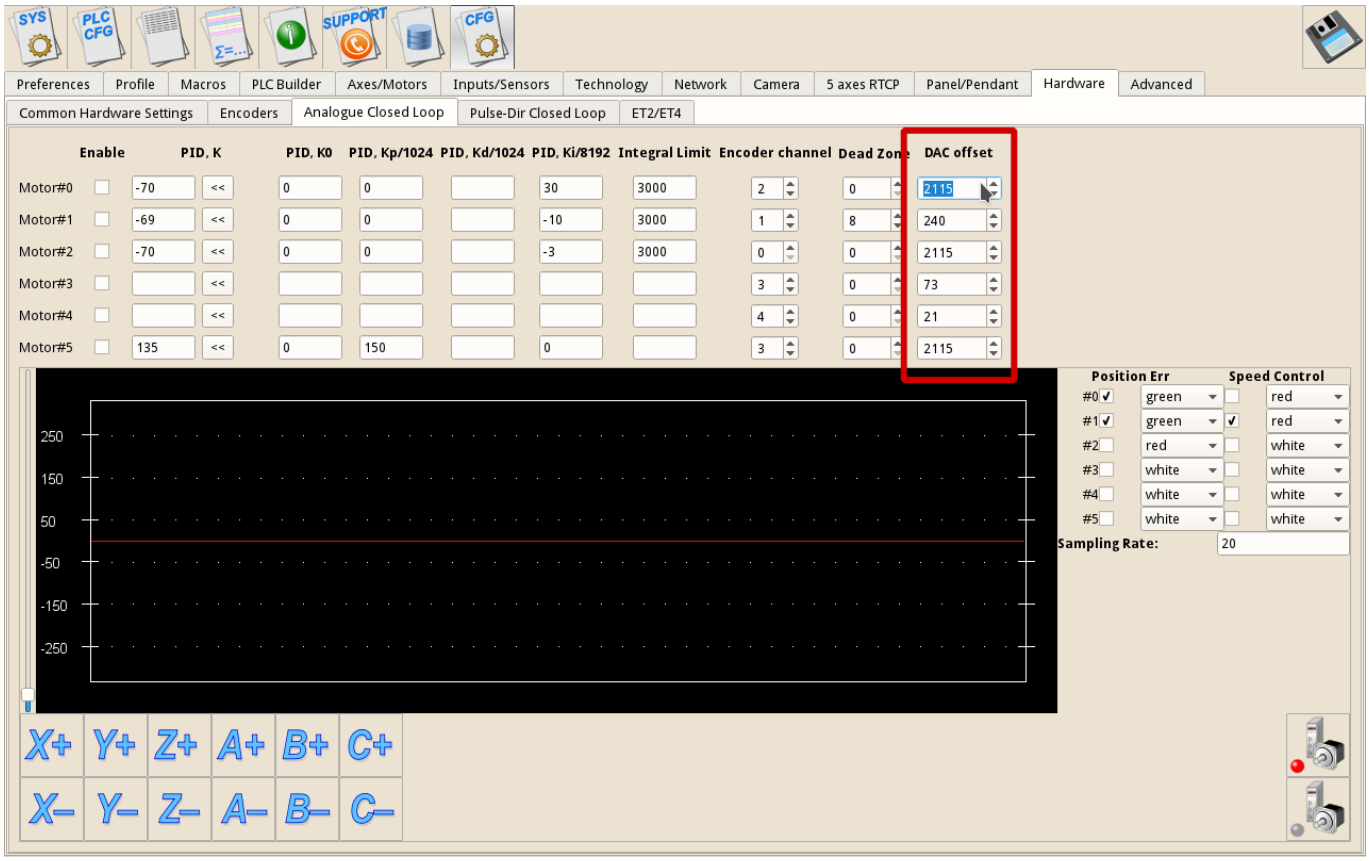
```
dac_value=555;                                  //DAC value to be sent

message=PLCCMD_SET_CNC_EXTVAR;                  //setup message variable to
PLC-to-CORE command
command=0x32;//EXT_ET5_DAC_OFFSET               //setup command register
(var00) for DAC register access
parameter=dac_value+(2<<12);                    //setup parameter register
(var01), DAC channel is #2
texit=timer+2;do{timer++;}while(timer<texit);   //2ms delay to push the
message
```

Another examples with +\-10V DAC programming can be found here - M03.plc, SPN.plc

Configuration dialogs

```
Settings->Cfg->Hardware->Analogue Closed Loop Dialog.
```

Each DAC outputs can be accessed by changing values in “DAC Offset” spinboxes.



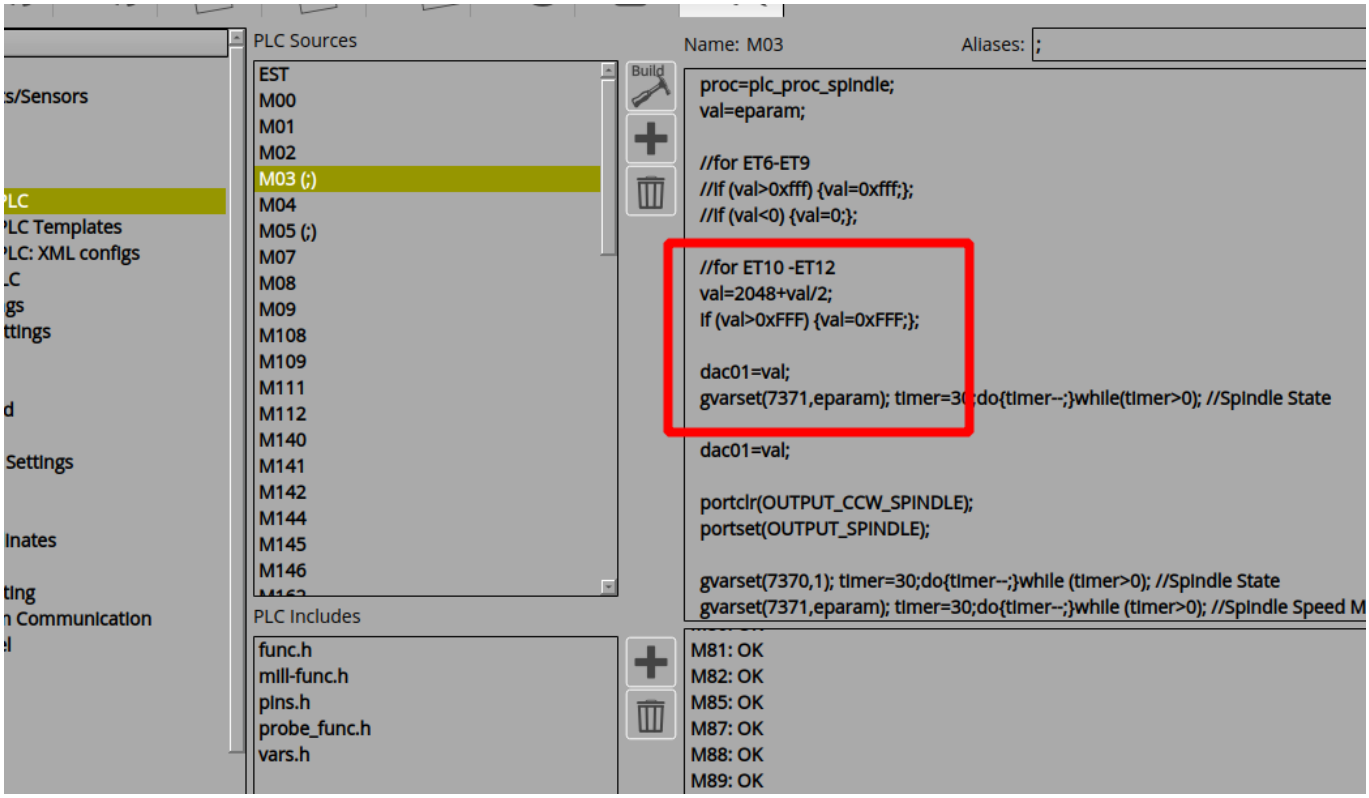
New functionality for DAC control

As of September 2022, the firmware for the ET10 boards has been updated to make DAC control easier. For instance, now users can run code such as this:

```
main()

{  timer=0;
  proc=plc_proc_spindle;
  val=eparam;  //for ET6-ET9

  //if (val>0xfff) {val=0xfff;};
  //if (val<0) {val=0;};  //for ET10 -ET12

  val=2048+val/2;

  if (val>0xFFF) {val=0xFFF;};  dac01=val;
  gvarset(7371,eparam);
  timer=30;do{timer--;}while(timer>0); //Spindle State

  dac01=val;

  portclr(OUTPUT_CCW_SPINDLE);
```

```
  portset(OUTPUT_SPINDLE);
  gvarset(7370,1);timer=30;do{timer--;}while (timer>0); //Spindle State
  gvarset(7371,eparam);timer=30;do{timer--;}while (timer>0); //Spindle Speed
Mirror register
  //gvarset(7372,0);//Mist State

  //timer=30;do{timer--;}while (timer>0); //
  //gvarset(7373,0);//Flood State
  //timer=30;do{timer--;}while (timer>0); //
```

Here, we are utilizing the usual `dac01` variable instead of the more complex solutions used in the past:



You can change the DAC used in the following config window, however it will still be listed as `dac01` in the PLC:

Please note that since it is necessary to have a voltage value between -10 and +10 V, the value of 2048 is assigned to 0V. Therefore, for the maximum value we will be adding 2048 (i.e., the original value divided by 2, as can be seen in the `val=2048+val/2` line).

From:
http://cnc42.com/ - **myCNC Online Documentation**

Permanent link:
http://cnc42.com/plc/spindle_speed_control_through_-10v_dac_channel_et10

Last update: **2022/09/13 13:51**