AN120

16-131983

SPI Communications for Current Control

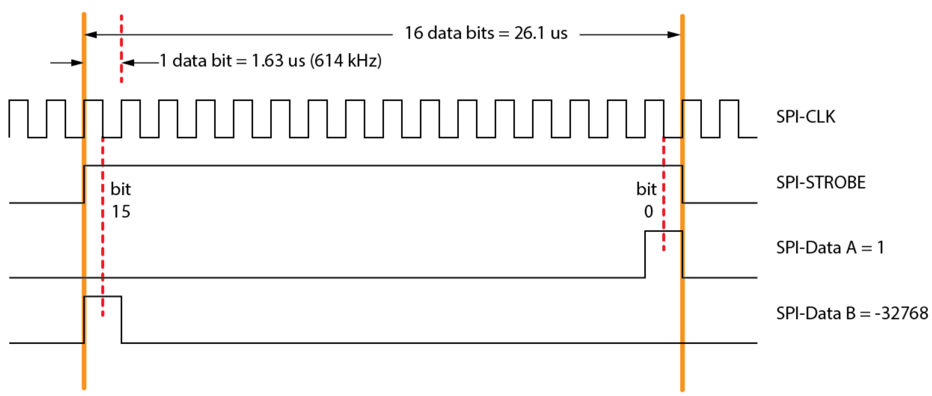
Introduction

SPI (Serial Peripheral Interface) enables direct transmission of numerical data for control of the current in the XEL/XPL servo drives. Two data channels are supported. A single data channel is used as a torque (current) command when the drive is commutating the motor. Two data channels are used for UV current control when the master is commutating the motor. U and V current data are used by the drive to produce a single current vector in the motor. Because the SPI data transmission is numerical, it is noise-free and does not have the bit-jitter that can occur with analog or PWM current control methods. In this implementation, the SPI data is sent at the drive's current loop frequency of 16 kHz.

Note: The examples and connections shown in this document were made with a TURBO PMAC2 CPU card and an ACC24E2 I/O card in a UMAC rack.

Overview of SPI

The SPI interface is comprised of four channels. The SPI-CLK clock runs continuously at ~614 kHz. The rising edge of the SPI-STROBE signal starts a data-transfer cycle in the drive and the falling edge terminates the data transfer. The SPI-Data A & B channels each have 16-bit pulse patterns that begin with bit 15 (the MSB of the data) and end with bit 0 (the LSB of the data). Each bit is sampled on the falling edge of the clock pulse which is in the center of the bit-time for greatest noise immunity.



UV Mode: Commutation in the PMAC

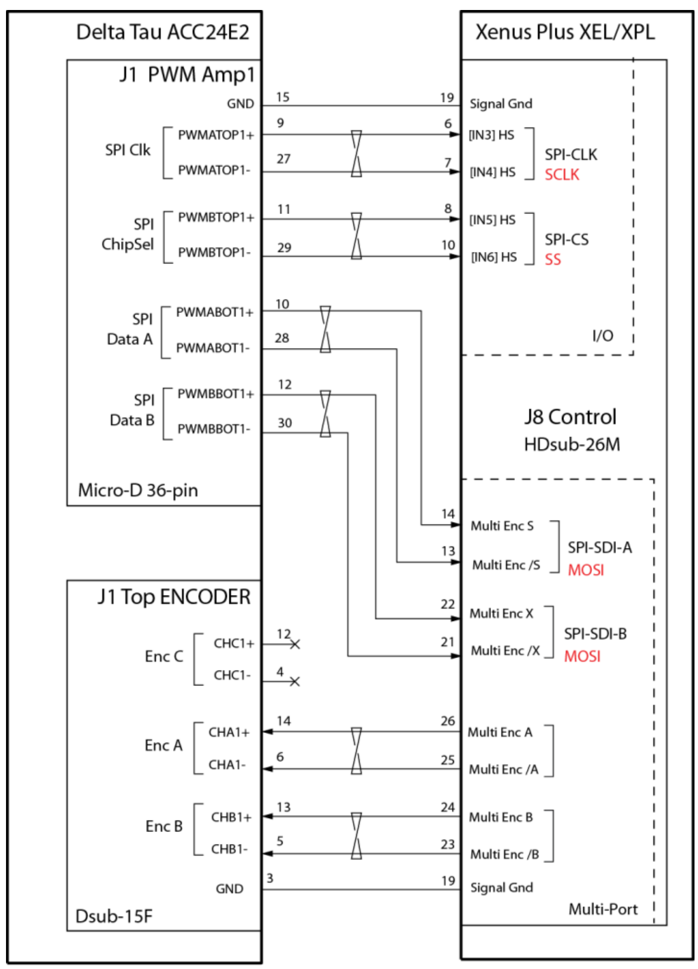
Two SPI data channels are used for the U and V current data. The drive converts these into a single current vector with magnitude and angle. The drive has no knowledge of the motor's position and Field-Oriented Control is not applied to the UV commands. Commutation is done in the PMAC.

Torque Mode: Commutation in the Servo Drive

In torque mode the PMAC uses a single data channel that commands the magnitude and polarity of the current vector. The servo drive uses position feedback from the motor for commutation and adjusts the angle of the current vector using Field-Oriented Control to optimize torque production in the motor.

Connections

The signals are high-speed differential format, and all connections should be made with twisted-pair cabling. If the setup is for commutation in the drive and torque control from the PMAC, then the SPI-Data B connection is not necessary.

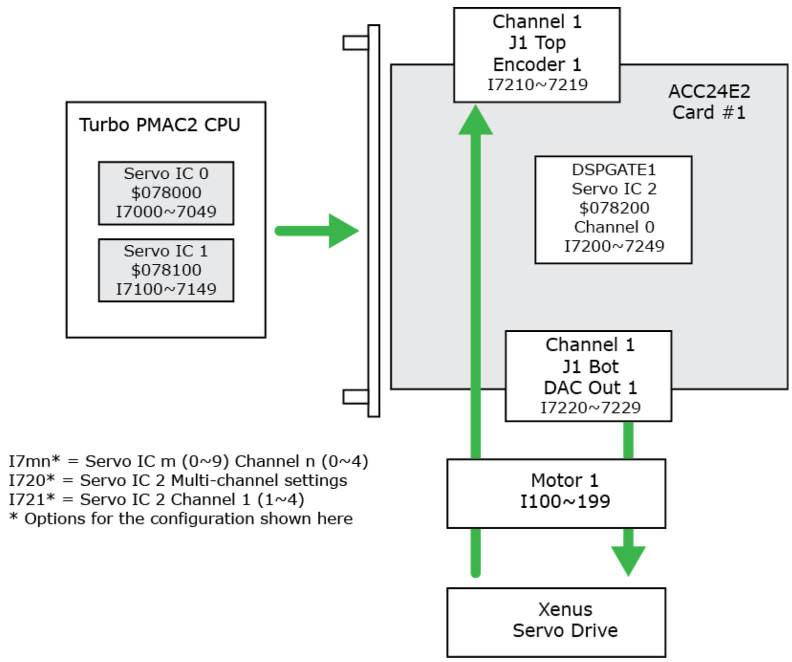


CONNECTIONS FOR UV CURRENT CONTROL VIA SPI
WITH COMMUTATION IN THE PMAC CONTROLLER

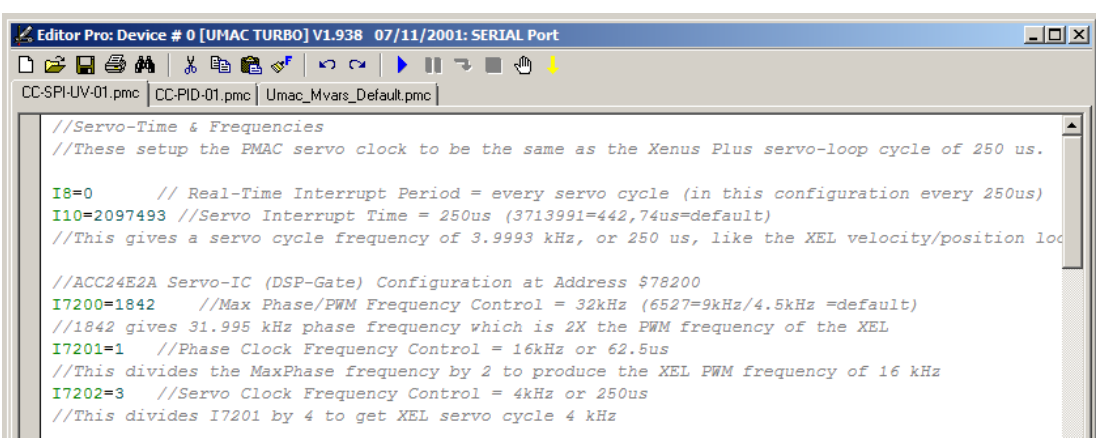
Items in RED are generic
SPI naming convention

PMAC Configuration

The following configuration is used for this application note:



PMAC I-Variables I7000~I7999 control the configuration of the Servo ICs. In this example, the XEL is Motor 1, controlled by Servo IC 2 on the ACC24E2 card (IC 0 & 1 are on the CPU card). This makes the I-Variables for the XEL I7200~I7219. I-Variables I700~I709 are "multi-channel" variables that affects the operation of all channels 1~4 of the Servo IC. I-Variables I7200~I7209 control the clock frequencies used by the Servo IC. I-Variables I710~I719 configure the channel 1 feedback and SPI outputs. The PMAC configuration file shown here is CC-SPI-UV-01.pmc. This file was made to contain the I-Variables that will set up the SPI communications for UV current control. Not included in this file are the parameters that set up a particular motor which will vary with the user's hardware. The motor 1 I-Variables are I100~I199.



PMAC I-Variables: SPI UV Mode

This is the content of the "CC-SPI-UV-01.pmc" file made for this example.

```
//Servo-Time & Frequencies
//These set up the PMAC servo clock to be the same as the Xenus Plus servo-loop cycle of 250 us.

I8=0                // Real-Time Interrupt Period = every servo cycle (in this configuration every 250us)
I10=2097493//Servo Interrupt Time = 250us (3713991=442,74us=default)
//This gives a servo cycle frequency of 3.9993 kHz, or 250 us, like the XEL velocity/position loop

//ACC24E2A Servo-IC (DSP-Gate) Configuration at Address $78200
I7200=1842          //Max Phase/PWM Frequency Control = 32kHz (6527=9kHz/4.5kHz =default)
//1842 gives 31.995 kHz phase frequency which is 2X the PWM frequency of the XEL
I7201=1             //Phase Clock Frequency Control = 16kHz or 62.5us
//This divides the MaxPhase frequency by 2 to produce the XEL PWM frequency of 16 kHz
I7202=3             //Servo Clock Frequency Control = 4kHz or 250us
//This divides I7201 by 4 to get XEL servo cycle 4 kHz

//Hardware Clock Control
//The DAC clock is set to 614 kHz to best read the SPI data bit stream with the XEL HS inputs
I7203=2450          //set DAC sample clock to 614 kHz and keep the other clocks at default

// DAC Strobe Settings
I7205=$FFFF00       // For Copley SPI interface, strobe signal (high for 16 clock cycles)

//Servo-IC2, Channel 1
I7210=7       // Encoder/Timer Decode Control = x4 quadrature decode CCW
I7212=1       //Encoder Capture Control = Capture on Index (CHCn) high
I7216=3       //Outputs A and B are DAC, Output C is PWM, for Copley SPI interface on Output A and B
I7219=0       //Hardware 1/T control = disabled

//Motor Setup I-Variables
I100=1        //Motor Activation Control
I101=1        //Motor Commutation Enable

I102=$78202//Command Output Address (ACC24E2 at address $78200)

I103=$3501          //Position Loop Feedback Address (ECT address of 1st entry)
I104=$3501          //Velocity Loop Feedback Address (ECT address of 1st entry)

//Flags and Position Capture
I124=$120001        //Motor 1 Flag Mode Control (no Amp-Fault, no HW-Limits)
I125=$78200//Motor 1 Flag Address
```

```
//Output Command Limit
I169=32767          //Maximum in units of 16-bit DAC bits

//Commutation setup
I170=2              //Number of Commutation Cycles (Ix71/Ix70=cts pro Commutation Cycle)
I171=4000           //Counts per N Commutation Cycles 1000 encoder lines/rev = 4000 cts/rev
I172=1365           //Commutation Phase Angle 120 deg (use I179 negative for manual phase
zero)
I129=0              //Output/First Phase Offset in units of 16-bit DAC bits
I179=0              //Output/Second Phase Offset in units of 16-bit DAC bits


//Commutation position address
I183=$78201// First Acc-24E, Encoder 1

//Phasing Parameters
I173=-3277          //Phase Finding Output Value = 10th of I169
I174=47             //Phase Finding Time in Servo Interrupt Cycles =
(3000ms/256*(8388608/I10)/(I8+1)
I180=6              //Phase finding mode = Stepper Method)
I181=$78200//Power-On Phase Position Address
I191=$0             //Power-On Phase Position Format

//ECT (Encoder Conversion Table)
I8000=$78200        //1/T Extension of Encoder 1, result in address $3501
I8001=$78208        //1/T Extension of Encoder 2, result in address $3502
I8002=$78210        //1/T Extension of Encoder 3, result in address $3503
I8003=$78218        //1/T Extension of Encoder 4, result in address $3504
I8004=0             //End of ECT
I8005=0             //End of ECT
```

Xenus Plus Configuration: SPI UV Mode

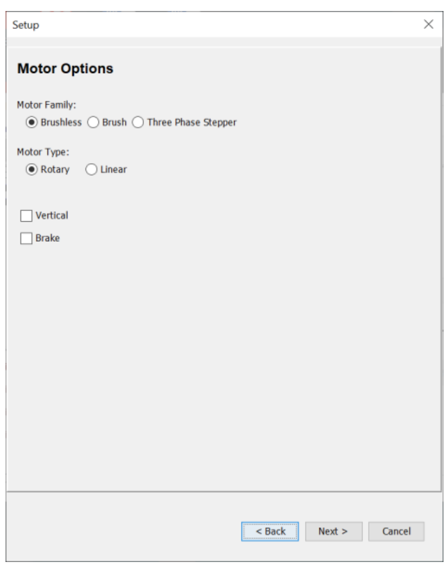
Note: Firmware version 2.43 or greater is required for SPI operation. When running in SPI mode, there should be no CVM (Copley Virtual Machines), camming, or other files and functions operating.

Basic Setup

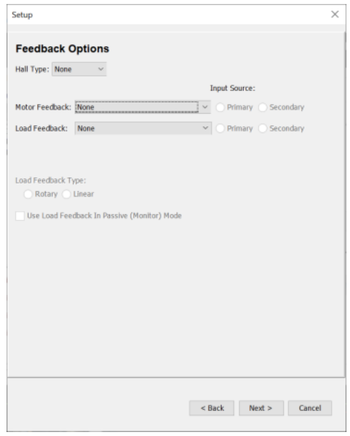
After clearing CVM programs, the drive can be configured in UV Current mode. Use the following steps to configure the drive in UV Current mode.

1. In CME, click the **Change Settings** button.

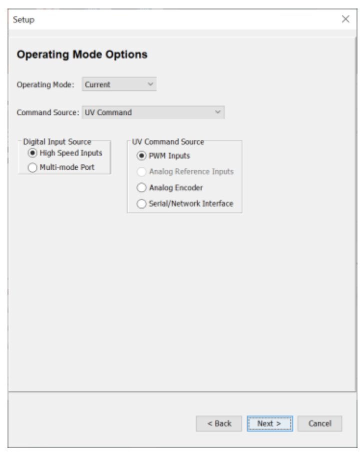
2. In the Motor Options screen, select **Brushless** for the Motor Family and **Rotary** for the Motor Type. Click the **Next** button to continue.



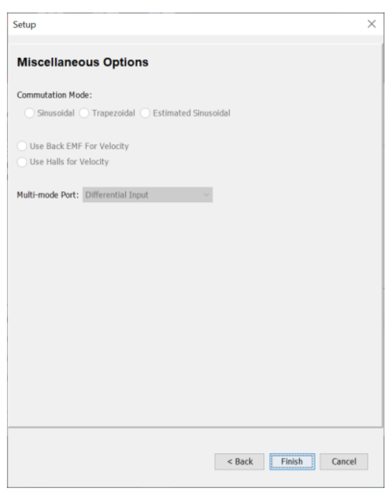
3. In the Feedback Options screen, select **None** for the Hall Type, Motor Feedback, and Load Feedback options. Click the **Next** button to continue.



4. In the Operating Mode Options screen, select **Current** in the Operating Mode field. Select **UV Command** in the Command Source field. Click the **Next** button to continue.



5. Click **Finish** to save the configuration to flash.

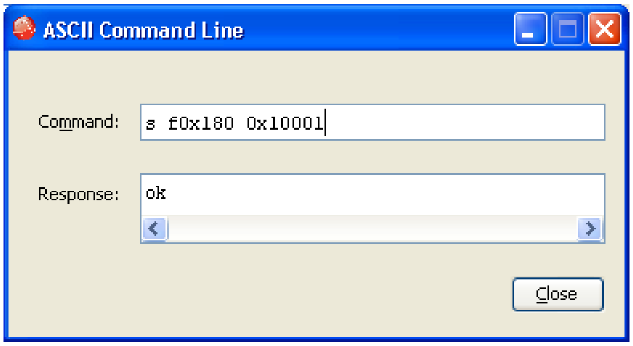


Configure Parameters for SPI and UV Current Mode

These are the parameters that set up SPI communications and configure the Xenus Plus for the same UV current control as the 7x25AC(F) analog servo amplifiers. In the data below, RF refers to the storage of the parameter: R = RAM, F = Flash

VAR	**Hex**	**Dec**	**RF**	**Remarks**
0x1800	0x10001	65537	RF	UV from 2 analog ref, Hall offset 0x4F added to UV angle
0xA9	0xF4	500	RF	UV input scaling, amps/100 when SPI data = 32767 (+100%) This example shows 5.00 A when SPI data is +100%
0x4E	0x01	1	F	Motor wiring: 1 = swap UV outputs
0x4F	0x1E	30	F	Hall offset = 30 degrees
0xAF	0x80	128	RF	Miscellaneous Amplifier Options, bit 7 set = SPI data replaces analog ref UV
Make these settings depending on the motor feedback: Quad A/B or absolute encoder connected to feedback port. Quad A/B Incremental Encoder:				
0x60	0x00	0	F	Encoder type is incremental, only A & B channels are used
0x5A	0x10	16	RF	Multi-mode port configuration, buffered encoder A/B signals on output channels A & B, and X channel is input for SPI-Data B
0x12A	0x08	8	F	Incremental encoder, X channel not used
Absolute EnDat 2.2 Absolute Encoder (or other absolute encoder using one or two digital channels:				
0x60	0x0B	11	F	Encoder type is incremental, only A & B channels are used
0x5A	0x3012	12306	RF	Multi-mode port configuration, emulated quad A/B outputs on channels A & B and X channel is input for SPI-Data B, b12~13 = 2 for 5 MHz data rate
0x12A	0x4nnnn	*	F	*Absolute encoder, b0~5 = number of bits of single-turn data, b8~12 = multi-turn bits b16 = 0 for no encoder sin/cos signals, b17 = 0 for encoder input from feedback port b18 = 1 for EnDat 2.2 encoders, b20~23 = 0 for number of bits to discard

In CME, open the Tools menu and select the ASCII Command Line option. The example below shows the ASCII Command Line when setting 0x180. The parameters are saved to RAM and Flash, depending on the parameter.



Enter the ASCII Command Line strings below and press the Enter key after entering each string to save that string. After pressing the Enter key an "ok" response should return, indicating that the command line was received by the drive successfully.

```
s r0x180 0x10001
s f0x180 0x10001
s r0xa9 500
s f0xa9 500
s f0x4e 1
s f0x4f 30
s r0x5a 16
s f0x5a 16
s r0xaf 128
s f0xaf 128
s f0x12a 8
```

From CME, Save to Flash, then from the Control Panel, press Reset. The drive will reboot with all of the settings saved above. In the Control Panel, click the Enable button. This will software-enable the drive and set it to UV mode. When this is done, it can be Hardware Enabled at which point the PMAC can control the current. From this point, the PMAC can auto-phase the motor, and proceed to make the PID adjustments to tune the position control settings.

SPI for Torque Control

In torque mode the SPI-DATA-B is not required. The same cabling as UV mode can be used and the drive will not use the SPI-DATA-B data.

PMAC I-Variables: Torque Mode

This is the content of the "CC-SPI-TRQ-01.pmc" file made for this example. The entries with strikeouts are not required when the servo drive is commutating the motor. They are shown here to clarify the difference in the PMAC setup when in torque mode compared to UV current mode. Entries in red are ones that will be changed for operation in torque mode.

```
//Servo-Time & Frequencies
//These set up the PMAC servo clock to be the same as the Xenus Plus servo-loop cycle of 250 us.

I8=0                    // Real-Time Interrupt Period = every servo cycle (in this configuration every 250us)
I10=2097493   //Servo Interrupt Time = 250us (3713991=442,74us=default)
//This gives a servo cycle frequency of 3.9993 kHz, or 250 us, like the XEL velocity/position loop

//ACC24E2A Servo-IC (DSP-Gate) Configuration at Address $78200
I7200=1842              //Max Phase/PWM Frequency Control = 32kHz (6527=9kHz/4.5kHz =default)
//1842 gives 31.995 kHz phase frequency which is 2X the PWM frequency of the XEL
I7201=1                 //Phase Clock Frequency Control = 16kHz or 62.5us
//This divides the MaxPhase frequency by 2 to produce the XEL PWM frequency of 16 kHz
I7202=3                 //Servo Clock Frequency Control = 4kHz or 250us
//This divides I7201 by 4 to get XEL servo cycle 4 kHz

//Hardware Clock Control
//The DAC clock is set to 614 kHz to best read the SPI data bit stream with the XEL HS inputs
I7203=2450              //set DAC sample clock to 614 kHz and keep the other clocks at default

// DAC Strobe Settings
I7205=$FFFF00           // For Copley SPI interface, strobe signal (high for 16 clock cycles)

//Servo-IC2, Channel 1
I7210=7         // Encoder/Timer Decode Control = x4 quadrature decode CCW
I7212=1         //Encoder Capture Control = Capture on Index (CHCn) high
I7216=3         //Outputs A and B are DAC, Output C is PWM, for Copley SPI interface on Output A and B
I7219=0         //Hardware 1/T control = disabled

//Motor Setup I-Variables
I100=1 //Motor Activation Control
I101=0 //Motor Commutation Disabled

I102=$78202   //Command Output Address (ACC24E2 at address $78200)

I103=$3501              //Position Loop Feedback Address (ECT address of 1st entry)
I104=$3501              //Velocity Loop Feedback Address (ECT address of 1st entry)

//Flags and Position Capture
I124=$120001 //Motor 1 Flag Mode Control (no Amp-Fault, no HW-Limits)
I125=$78200   //Motor 1 Flag Address
```

```
//Output Command Limit
I169=32767              //Maximum in units of 16-bit DAC bits


//ECT (Encoder Conversion Table)
I8000=$78200 //1/T Extension of Encoder 1, result in address $3501
I8001=$78208 //1/T Extension of Encoder 2, result in address $3502
I8002=$78210 //1/T Extension of Encoder 3, result in address $3503
I8003=$78218 //1/T Extension of Encoder 4, result in address $3504
I8004=0                 //End of ECT
I8005=0                 //End of ECT
```

Xenus Plus Configuration: SPI Torque Mode

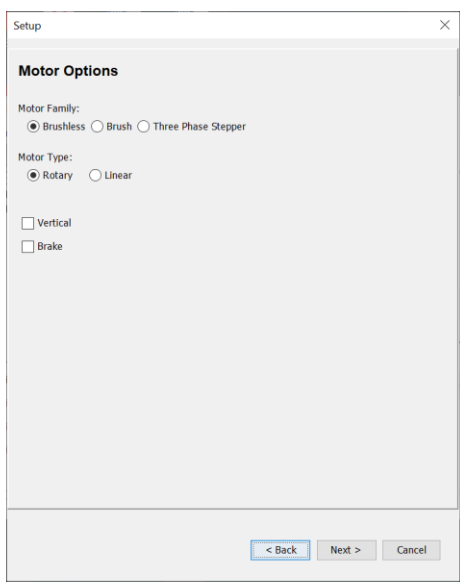
Note: Firmware version 2.43 or greater is required for SPI operation. When running in SPI mode, there should be no CVM, camming, or other files and functions operating.

Basic Setup

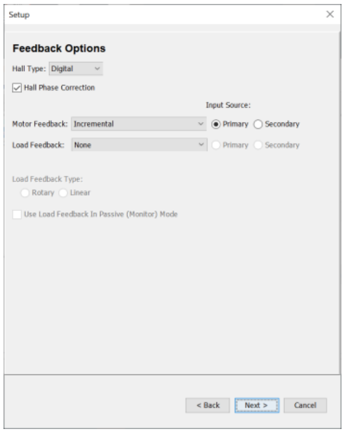
After clearing CVM programs, the drive can be configured in Analog Current mode. Use the following steps to configure the drive in Analog Current mode.

1. In CME, click the **Change Settings** button.

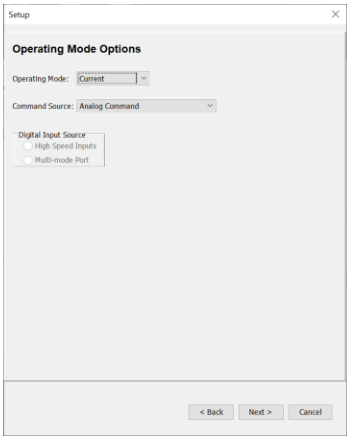
2. In the Motor Options screen, select **Brushless** for the Motor Family, and **Rotary** for the Motor Type. Click the **Next** button to continue.



3. In the Feedback Options screen, select **Digital** for the Hall Type, **Incremental** for the Motor Feedback, and **None** for the Load Feedback options. Click the **Next** button to continue.

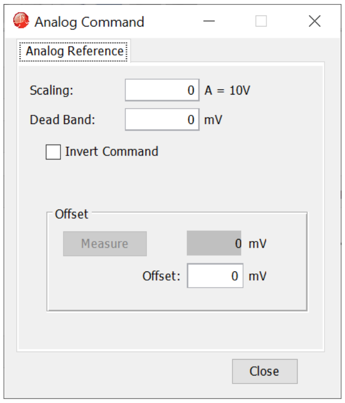


4. In the Operating Mode Options screen, select **Current** in the Operating Mode field. Select **Analog Command** in the Command Source field. Click the **Next** button to continue.



5. Click **Finish** to save the configuration to flash.

6. On the CME main page, click the Analog Command box. In the Scaling field, enter the current that the drive will output when the SPI command is at ±100%.



Motor Setup

For detailed steps on how to produce torque using CME, refer to the CME User Guide. When completed, the drive will be able to control current and produce torque in +/- directions.

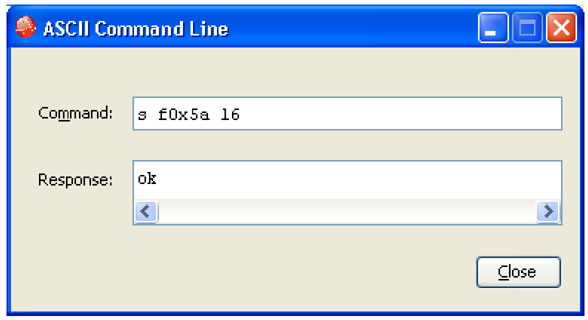
- Enter data into the Motor/Feedback box
- Auto-Phasing
- Current Loop tuning

Configure Parameters for SPI and Toque Mode

These are the parameters that set up SPI communications and configure the Xenus Plus for torque control and commutation of the motor.

VAR	Hex	Dec	RF	Remarks
0x5A	0x10	16	RF	Multi-mode port configuration, channels A-B buffered encoder outputs, and X channel is set as an input to be compatible with the PMAC output (although the data will not be used in torque mode
0xAF	0x80	128	RF	Miscellaneous Amplifier Options, bit 7 set = SPI data replaces analog input
0x12A	0x08	8	F	Motor wiring: 1 = swap UV outputs
0x4F	0x1E	30	F	Hall offset = 30 degrees
0xAF	0x80	128	RF	Incremental A & B encoder outputs to the PMAC (X channel is left as an input)

In CME, navigate to the Tools menu and select the ASCII Command Line option. The image below shows how the ASCII Command Line looks when setting 0x5A. The parameters are first saved to both RAM and Flash. Resetting the drive, after the settings are made, will restore them to working RAM memory.



Enter the ASCII Command Line strings below and press the Enter key after entering each string to save that string. After pressing the Enter key an "ok" response should return, indicating that the command line was received by the drive successfully.

```
s r0x5a 16
s f0x5a 16
s r0xaf 128
s f0xaf 128
s f0x12a 8
```

From CME, Save to Flash, then from the Control Panel, press Reset. The drive will reboot with all of the configured settings. In the Control Panel, click the Enable button. This will software-enable the drive and set it to Analog Current mode. However, the settings made above replace the analog input data with the SPI torque command. When this is done, it can be Hardware Enabled at which point the PMAC can control the current.

Revision History

Date	Version	Revision
8/31/2021	Rev 00	Initial release