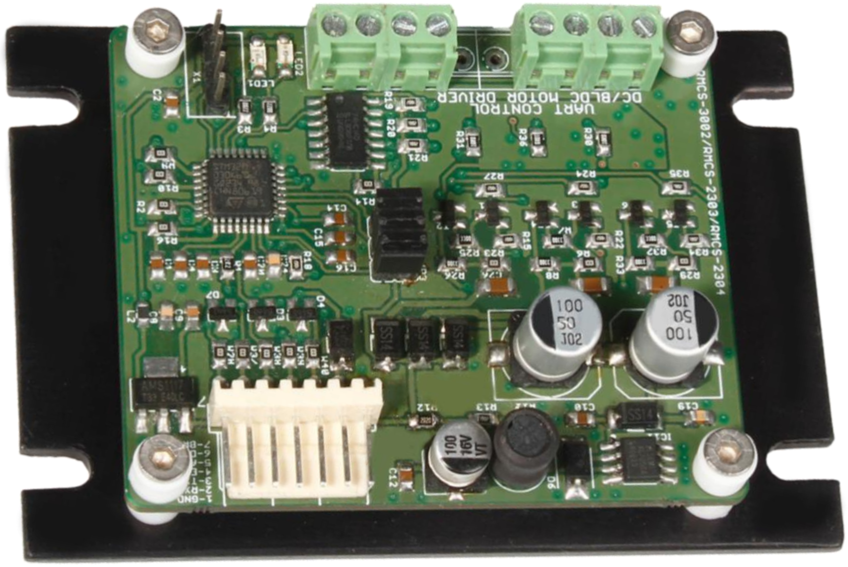
Rhino Motion Controls

DC Servo Drive

for Rhino Encoder DC Servo Motor

(Model No : RMCS – 2303)



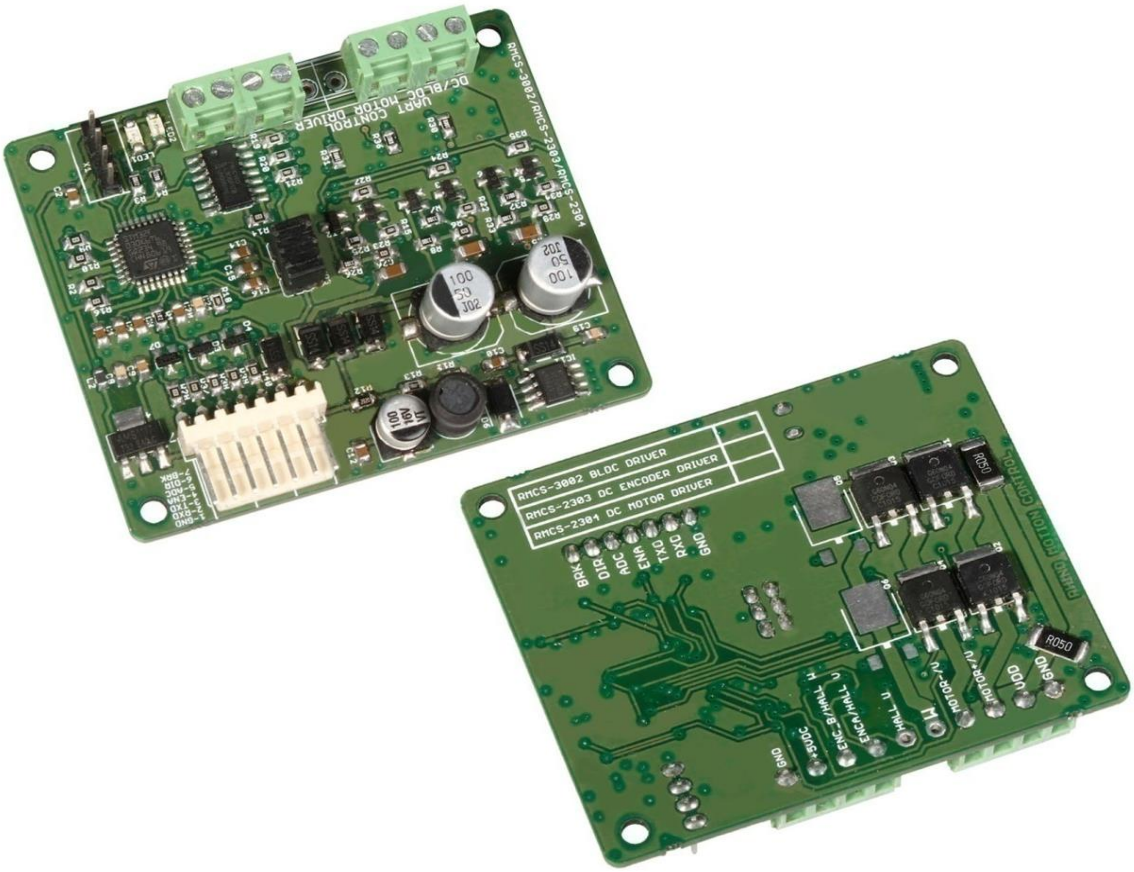
Operating Manual v1.0

Contents

Introduction – Salient Features

Rhino Motion Controls RMCS-2303 with UART ASCII is a high performance dc servo drive (10–30 V DC) designed for optimized operation of Rhino DC Servo motors with encoder feedback. This is an amazing cost effective solution to provide closed loop servo control for various applications. The salient features of this drive:

- This drive provides a closed loop speed control for the dc servo system.
- The motor programmed speeds are maintained irrespective of the voltages supplied.
- Also by using this drive, the rated torque of the motor is available at all speeds and the torque does not decrease with change in speeds.
- It is possible to run the motor in three different modes, analog speed control mode, digital speed control mode and position control mode.
- It has short-circuit protection for the motor outputs, over-voltage and under-voltage protection and will survive accidental motor disconnects while powered-up.
- This drive is configured using MODBUS ASCII protocol via UART.
- There is a function in the drive for setting the Modbus Slave Address from 1 to 7 using physical jumpers (Hardware Setting) or using MODBUS Tool Device (Software Setting).
- There are three user modes in the drive :
    - Mode0 - Analog Control Mode
    - Mode1 - Digital Speed Control Mode
    - Mode2 - Position Control Mode

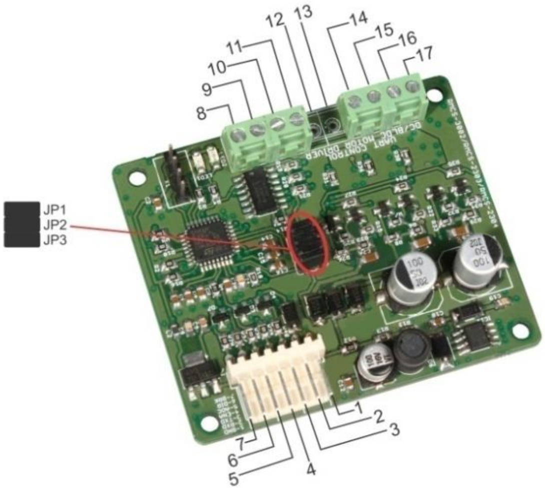


Technical specifications and Pin description

Supply Voltage and Current

Specification	Min	Max	Units	Comments
Supply Voltage	10	30	Volts DC	Between +Ve and GND
Phase Current	0.5	3	Amps	Peak 5 Amps per phase

Pin description of the drive is as per below image:



Pin No.	Description
1	GND
2	RXD
3	TXD
4	ENA
5	ADC
6	DIR
7	BRK

(Pins 1-7 are used for drive Configuration and UART control)

Pin No.	Description
8	GND
9	+5VDC
10	ENC_B/Hall W
11	ENC_A/Hall U
12	Hall U
13	W
14	Motor- / V
15	Motor+ / U
16	VDD
17	GND

(Pins 8 - 17 are connected to motor and power supply as described)

Pin No.	Description
JP1	Jumper 1
JP2	Jumper 2
JP3	Jumper 3

(Jp1 to JP3 are used for Hardware setting of slave ID)

Description of the three modes of Operation and Slave ID settings

Before we describe the configuration settings and connection diagram of the drive with the motor, we would like to describe the three modes in which the motor can run and slave ID addressing which are unique features of the drive.

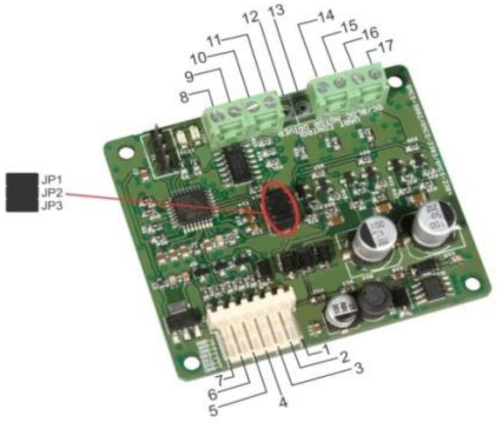
Different Modes of Operation: The drive can be configured in the following three modes by the user -

1. **Mode 0 Analog speed control mode :**
   a. In this mode the speed of the Rhino DC Servo motor can be controlled by an externally connected Potentiometer.
   b. So in this mode the user can increase or decrease the speed manually based on requirement.
   c. The drive will provide full torque at all speeds within the range.
   d. However the potentiometer has to be connected via a voltage divider to provide a maximum of 3.3 volts so as to not damage the drive.
   e. Also the Enable, Brake and direction pins have to be connected as per configuration requirements.

2. **Mode 1 Digital Speed control mode with direction control** :
   a. In this mode the speed and direction of the Rhino DC servo motor is settable / controllable via a Computer / Arduino Controller board / any other Modbus ASCII compatible device.
   b. As in the analog mode here there is no compromise in the torque output of the motor irrespective of the operational speed and voltage supply and control at higher speeds.
   c. This mode is used when multiple motors are to be used to run at exactly the same RPM and same torque even though the voltage supply might be different.
   d. Also in this mode the direction of the motor can be controlled digitally via modbus ASCII commands to run the dc servo motor in both directions
   e. In applications like conveyor belts where speed control is critical and industrial robotic applications like solar cleaning robots where straight line motion is critical for correction operation of the equipment, the digital speed control mode can provide optimal results.

3. **Mode 2 Position Control Mode** :
   a. In this mode position of Rhino DC Servo Motor is controllable. This means the user can program the number of rotations.
   b. Also the direction of rotation is to be configured using the Modbus commands
   c. The user needs to know the no of lines in the encoder used in the servo motor to use this mode correctly. The no of lines of encoder also needs to be set in the drive as one of the parameters.
   d. The position control mode is suitable in applications where the distance travelled by a robot is to be fixed or the no of rotations is to be fixed.

Slave ID Addressing: The second unique feature in this drive is that a single controller can be used to control seven drives at the same time using the physical jumpers. Using the Modbus Poll software, the slave ID can be set from 1 to 247. However, in this driver, we can set slave ID from 1 to 7 using physical jumpers also. As shown in the image below there are three jumpers marked by a red circle shown in the image below.



The three jumpers JP1, JP2 and JP3 can be set in the configuration as per the below table to provide a physical slave address to the drive. In the below table a value of ‘0’ corresponds to a state where the jumper is not connected and a value of ‘1’ corresponds to a state where the jumper is connected. If none of the jumpers are connected the drive has been programmed to use the Slave ID 7 in default mode.

Slave ID	JP1	JP2	JP3	Image of connection on the Drive
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

How to use the drive:

The drive needs to be configured to run in any one of the three modes described in the above section to make a closed loop control system. The drive can be configured using a PC with Modbus Poll software / Modbus controller / Arduino Board. We shall be discussing the configuration through Modbus Poll software in this document.

To describe the configuration of the drive for a closed loop system we shall be using the following:

1. PC with Modbus Poll software. (The modbus poll software can be downloaded from the link -

   https://www.modbustools.com/download.html)

2. RMCS-2303 UART ASCII Encoder DC Motor Driver

   https://robokits.co.in/motor-drives-drivers/encoder-dc-servo/rhino-industrial-encoder-dc-motor-driver-50w-with-uart-ascii-compatible-10-to-30-v-10a

3. Encoder DC Servo Motor (any of the below motors would work with the drive)

   https://robokits.co.in/rhino-planetary-encoder-dc-servo

   https://robokits.co.in/rhino-ig32-precision-dc-servo

   https://robokits.co.in/motors/encoder-dc-servo/high-torque-dc-encoder-motor

   https://robokits.co.in/motors/encoder-dc-servo/high-torque-high-precision-motor

4. Industrial Power Supply (below is a recommended supply. It can vary as per your requirements)

   https://robokits.co.in/power-supply/industrial-power-supply/24v-10a-industrial-power-supply

5. RKI-1154 CP2102 USB UART Module

   https://robokits.co.in/control-boards/interface-boards/cp2102-usb-uart-module

6. External 10K Potentiometer (for analog Speed Control mode only).

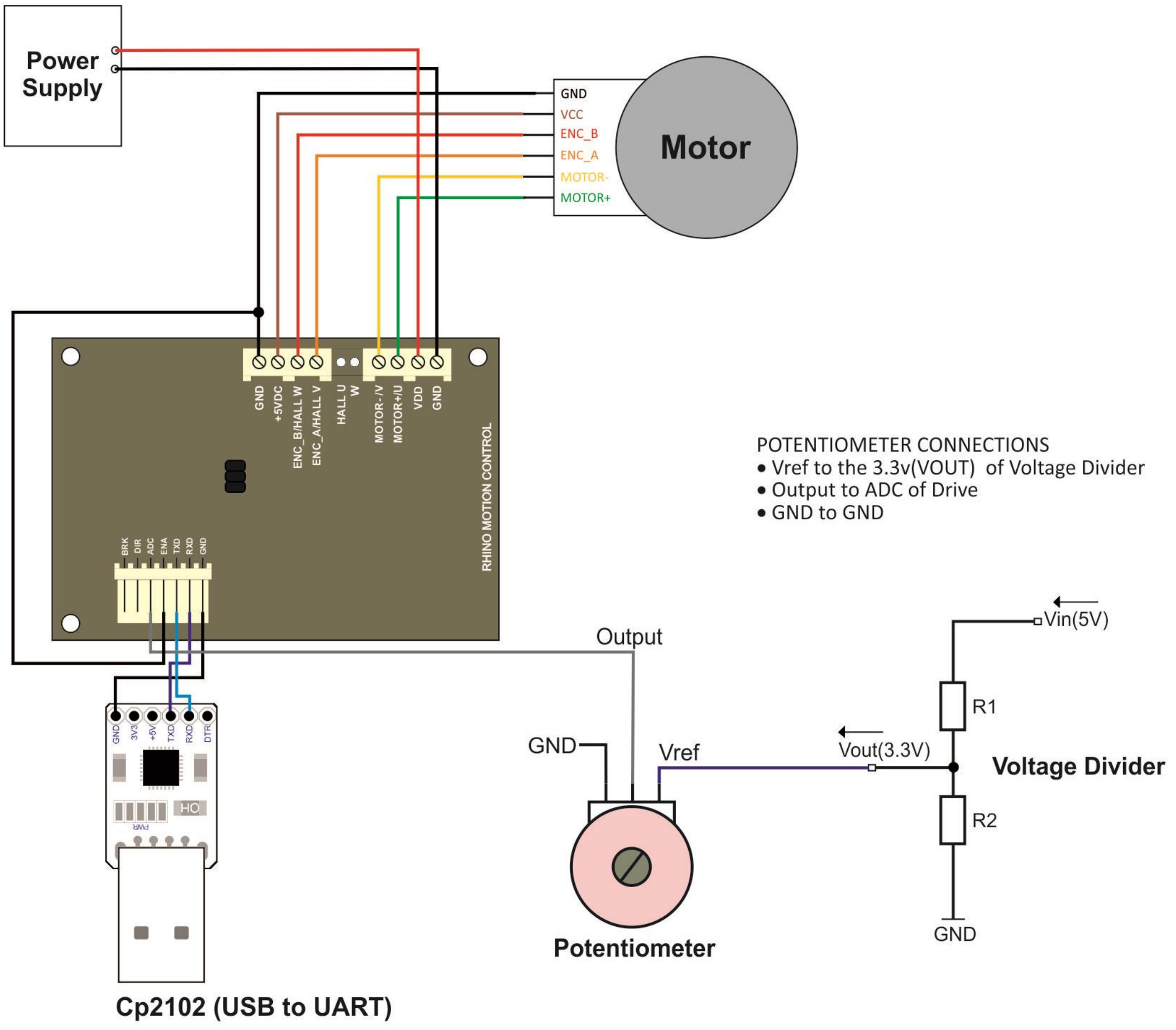
7. Voltage divider from 5V to 3.3V or Voltage converter from 5V to 3.3V (for analog speed control mode only).

Steps to configure the Drive using Modbus Poll Software :

1. Hardware Connection
2. Connect Modbus UART ASCII drive with Modbus Poll Software
3. Set Slave ID in Read/Write Function according to the physical jumper or software settings
4. Go to Function-Write Single register
5. Set the Lines per rotation as per the encoder and motor specifications
6. Set one of the three user Mode as per requirement
7. Save and Reset the settings in drive

STEP 1: Hardware Connection

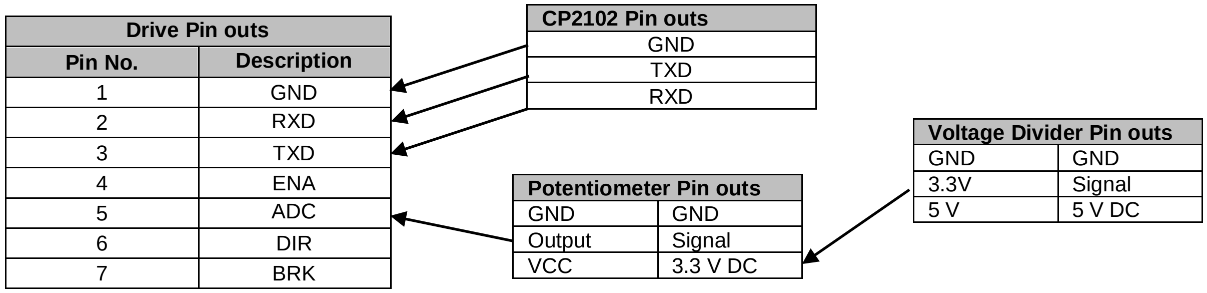
The hardware connections need to be setup between the motor, drive and PC as per the below block diagram and the pin out tables provided below. The potentiometer connections need to be provided only to run the driver and motor in Mode0 analog speed control mode. For analog mode connect ENA (Pin 4) to GND (Pin 8) to enable motor.



Drive Pin outs	
Pin No.	**Description**
8	GND
9	+5VDC
10	ENC_B/Hall W
11	ENC_A/Hall U
12	Hall U
13	W
14	Motor- / V
15	Motor+ / U
16	VDD
17	GND

Motor Pin outs	
Motor	**Wire Color**
GND	Black
VCC(5 V DC)	Brown
ENC_B(Encoder B)	Red
ENC_A(Encoder A)	Orange
M-(Motor-)	Yellow
M+(Motor+)	Green

Power Supply Pin outs
VCC – 10 to 30V
GND



Drive Pin outs	
Pin No.	**Description**
1	GND
2	RXD
3	TXD
4	ENA
5	ADC
6	DIR
7	BRK

CP2102 Pin outs
GND
TXD
RXD

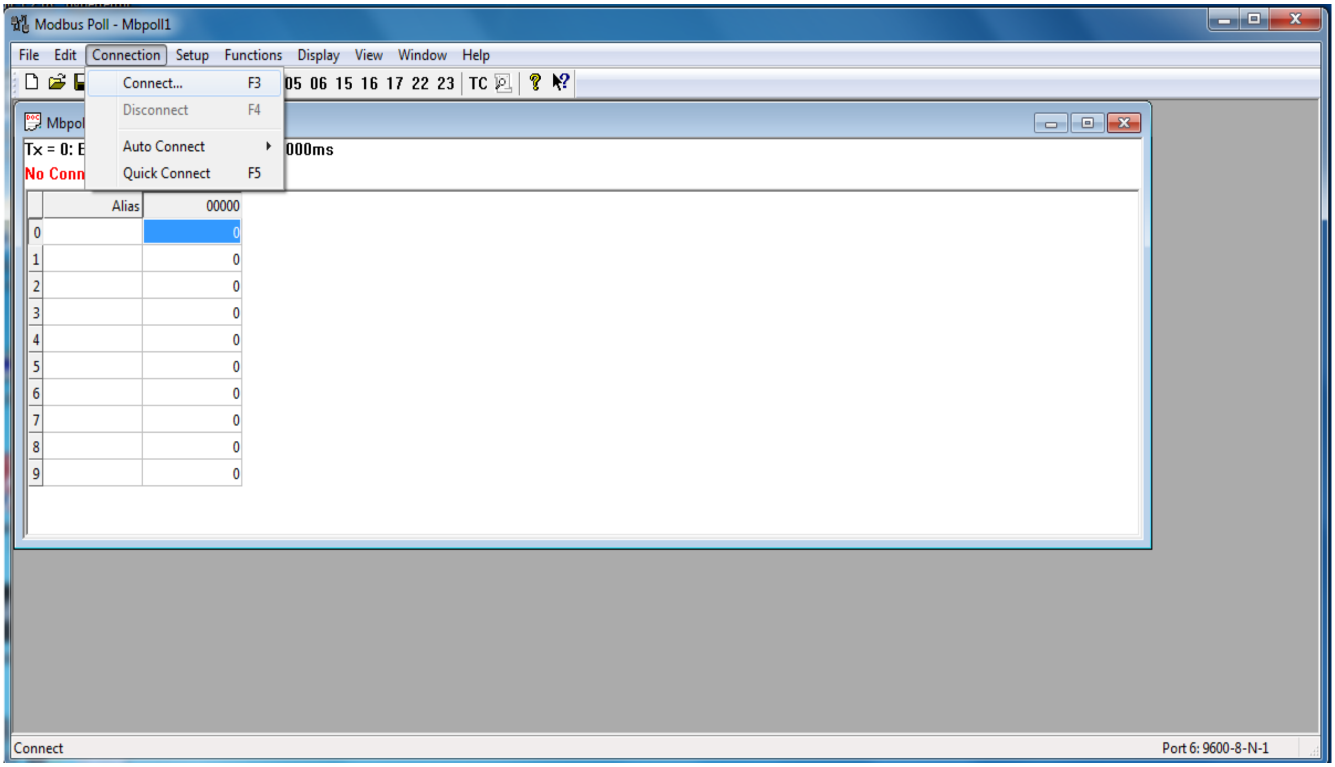
Potentiometer Pin outs	
GND	GND
Output	Signal
VCC	3.3 V DC

Voltage Divider Pin outs	
GND	GND
3.3V	Signal
5 V	5 V DC

STEP 2: Connect Modbus UART ASCII Drive with Modbus Poll Software

a) Once the Modbus Poll software is installed and the CP2102 is connected, please run the Modbus Poll software.

b) Proceed to the Connection Tab on the screen and click the same as per below image.



Tx = 0: E ... 000ms

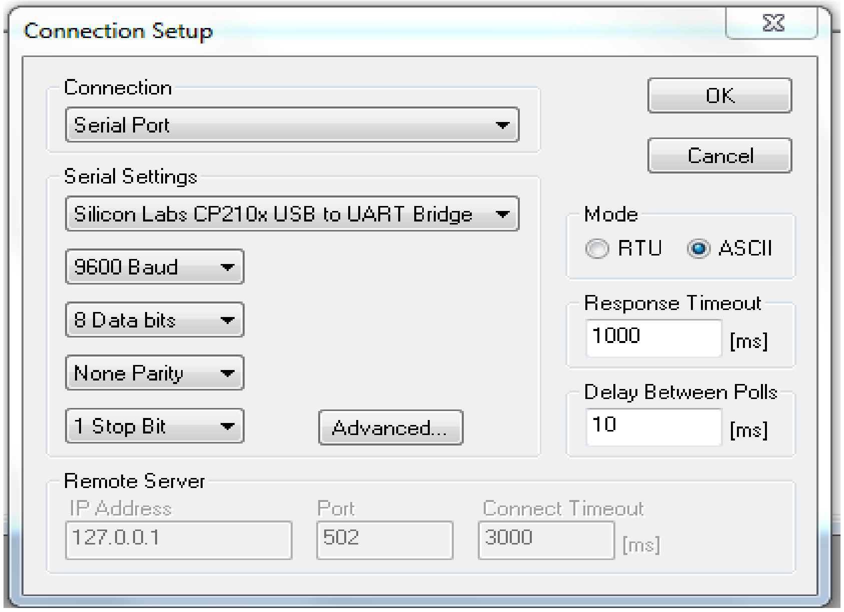
No Conn

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

Connect

Port 6: 9600-8-N-1

c) Then click on the connect tab which will open the connection settings as per the next image.



Connection: Serial Port

Serial Settings:
Silicon Labs CP210x USB to UART Bridge
9600 Baud
8 Data bits
None Parity
1 Stop Bit

Mode: ○ RTU ● ASCII

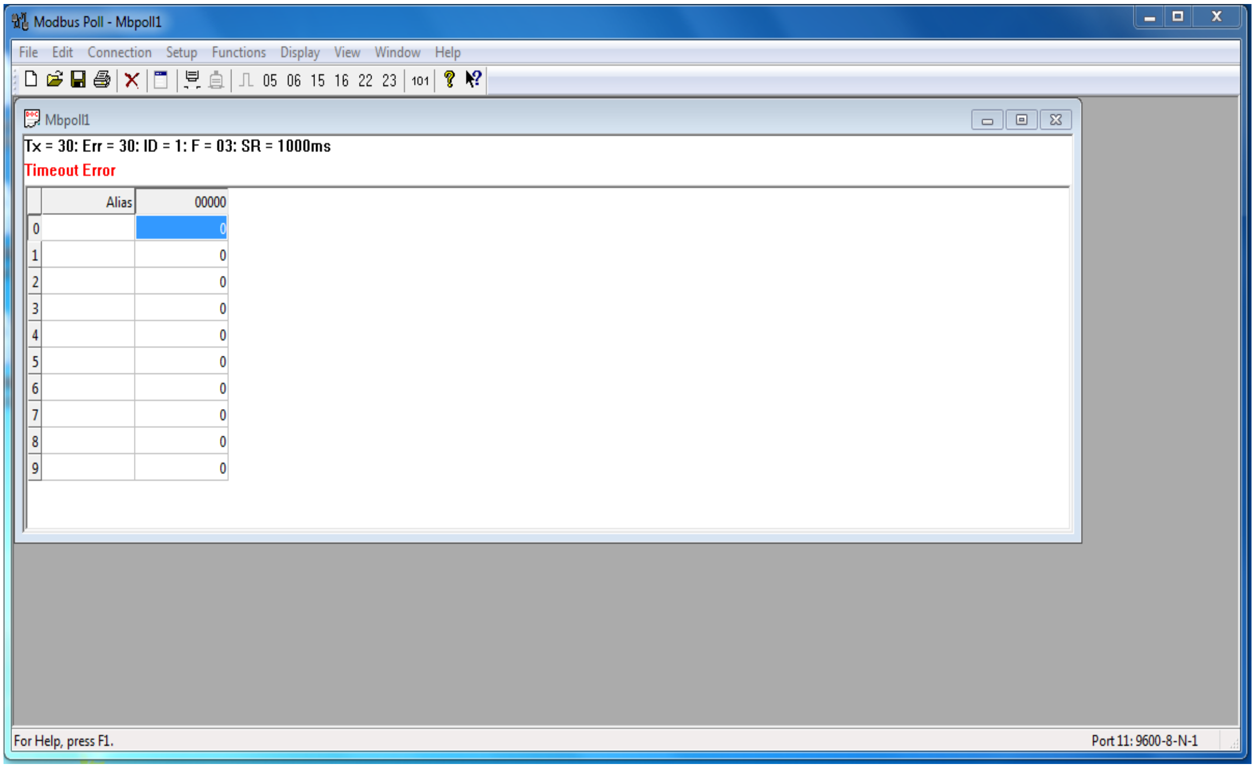
Response Timeout: 1000 [ms]

Delay Between Polls: 10 [ms]

Remote Server:
IP Address: 127.0.0.1
Port: 502
Connect Timeout: 3000 [ms]

d) Please make sure the connection setup settings are as per above for all parameters including the parity and stop bit.

e) Once the OK button is pressed in the connection setup, the screen should show connected and it should be as per below image.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1

Tx = 30: Err = 30: ID = 1: F = 03: SR = 1000ms

Timeout Error

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

For Help, press F1.

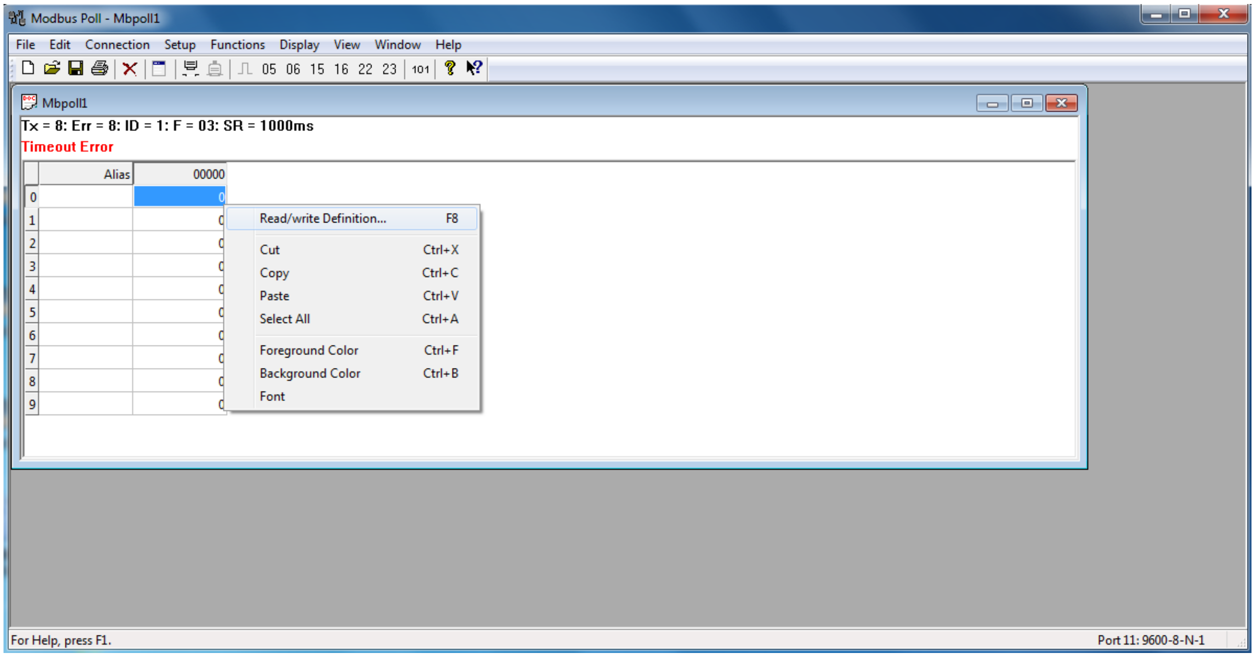
Port 11: 9600-8-N-1

f) This concludes **STEP 2** of the configuration procedure.

Step 3: Set slave ID in Modbus Poll Software to send and receive Data

Read/Write Definition:

a) Right Click on the Table shown on the home screen of Modbus Poll software. It should pop up a menu as per below image. Change screenshot

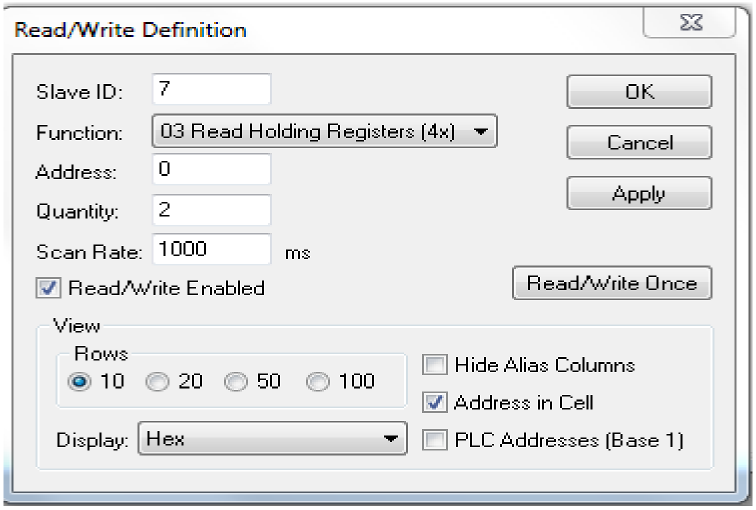


Tx = 8: Err = 8: ID = 1: F = 03: SR = 1000ms
Timeout Error

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

For Help, press F1.
Port 11: 9600-8-N-1

b) Now select the first option Read / write Definition. It should open up a window as per below image.



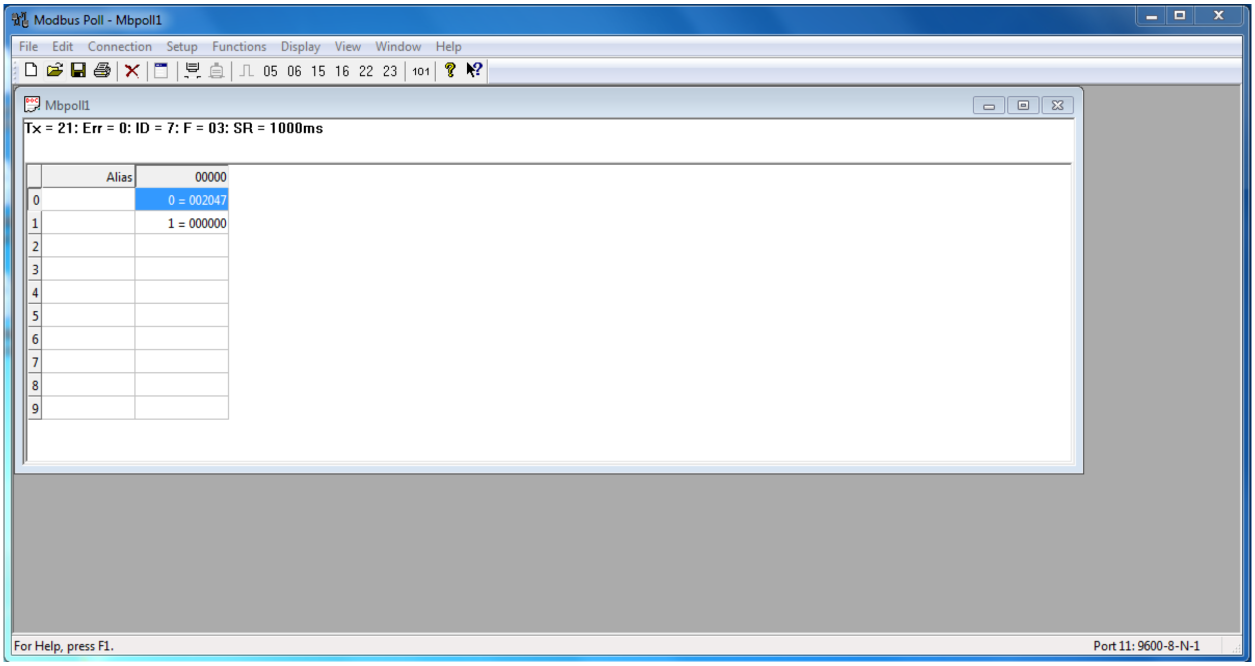
Slave ID: 7
Function: 03 Read Holding Registers (4x)
Address: 0
Quantity: 2
Scan Rate: 1000 ms
☑ Read/Write Enabled
Rows: ● 10 ○ 20 ○ 50 ○ 100
Display: Hex
□ Hide Alias Columns
☑ Address in Cell
□ PLC Addresses (Base 1)

c) Make all the settings as per above image except the slave ID. The slave ID needs to be set as per your requirement or physical jumper settings. For a slave ID of 7, please ensure below settings are updated.

1. Slave ID: As per jumper setting as shown on page 4.
2. Function: Select Function as 03 Read Holding registers (4x).
3. Select quantity as 2.
4. Select Display as “Hex”.

d) Once all settings are done, click on apply then select ok.

e) You should receive data in the main table as per below image.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1

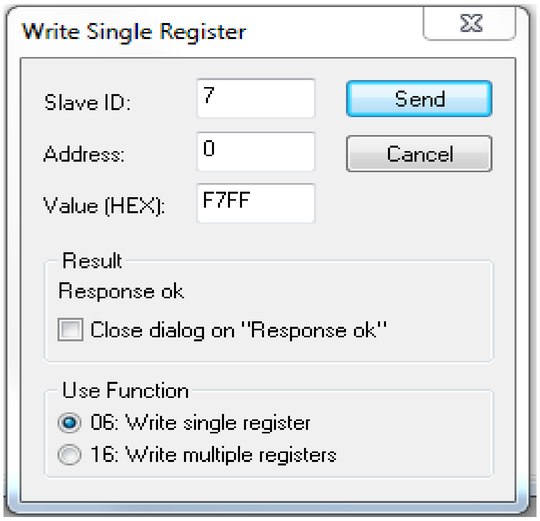
Tx = 21: Err = 0: ID = 7: F = 03: SR = 1000ms

	Alias	00000
0		0 = 002047
1		1 = 000000
2		
3		
4		
5		
6		
7		
8		
9		

For Help, press F1.

Port 11: 9600-8-N-1

f) The procedure to set Slave ID through software setting is explained below with images. User can set 1 to 247 Slave ID through software setting. For example If user wants to set slave ID 247 then Send Hex value F7FF (247) to address 0 as per below image.



Slave ID: 7

Address: 0

Value (HEX): F7FF

Result

Response ok

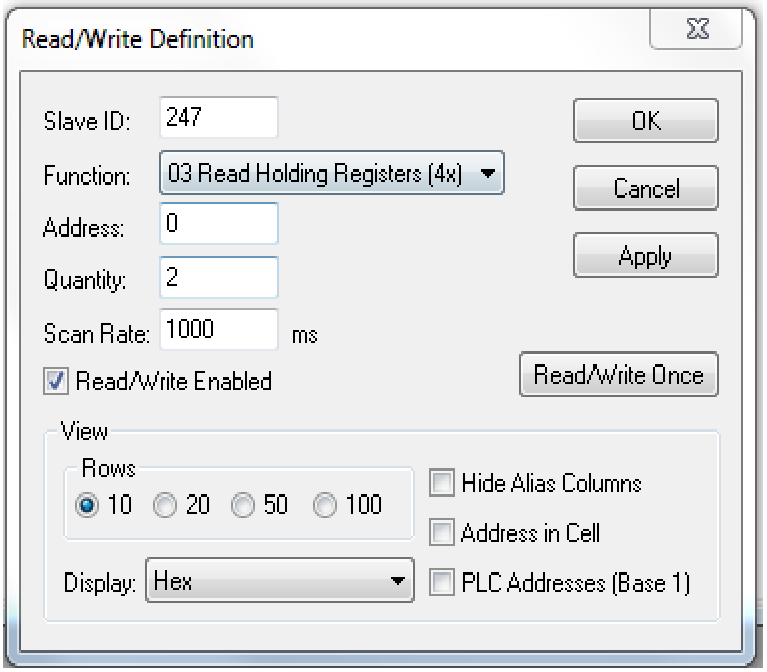
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

g) Then with power reset also remove all the three jumpers and enter Slave ID 247 as per below image.



Read/Write Definition

Slave ID: 247

Function: 03 Read Holding Registers (4x)

Address: 0

Quantity: 2

Scan Rate: 1000 ms

☑ Read/Write Enabled

View

Rows

● 10 ○ 20 ○ 50 ○ 100

□ Hide Alias Columns

□ Address in Cell

Display: Hex

□ PLC Addresses (Base 1)

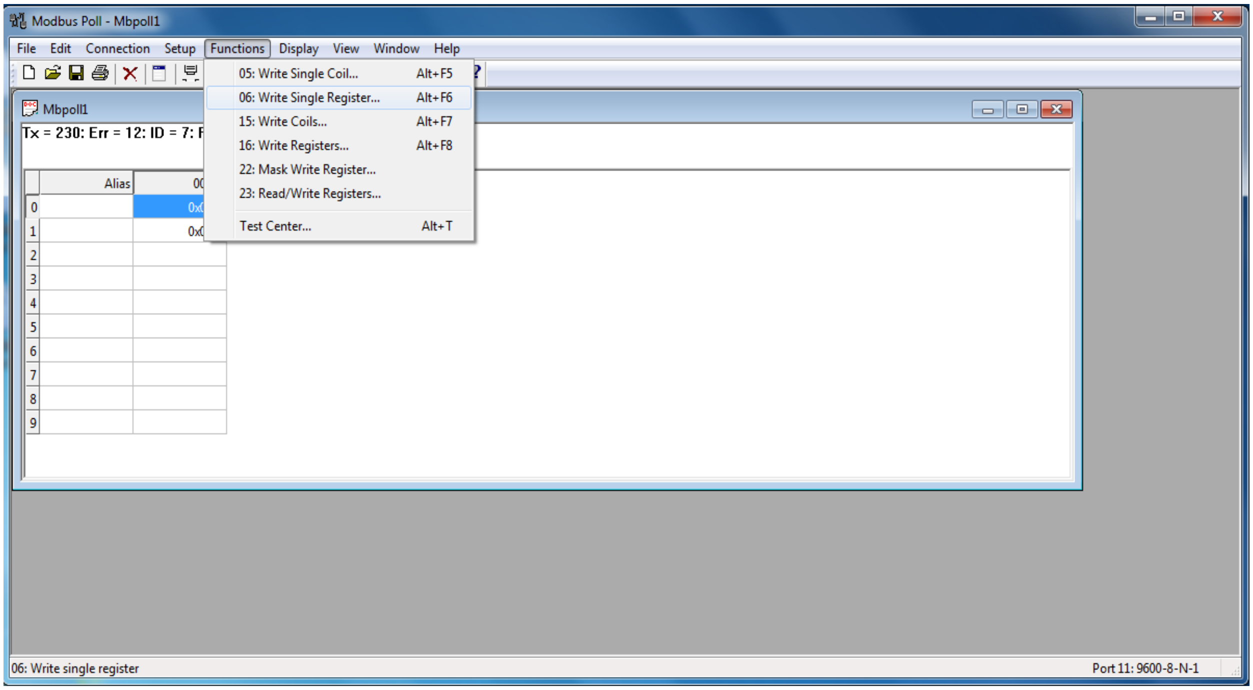
h) Then your drive is connected with Modbus Poll Software and Slave ID is 247.

i) This concludes **STEP3** of the configuration procedure.

Step 4: Go to Function

This step is to understand the procedure to write the data to the drive to run it as per the requirement.

a) Go to Functions tab on the home screen and click on “06: Write Single Register” as per the below image.

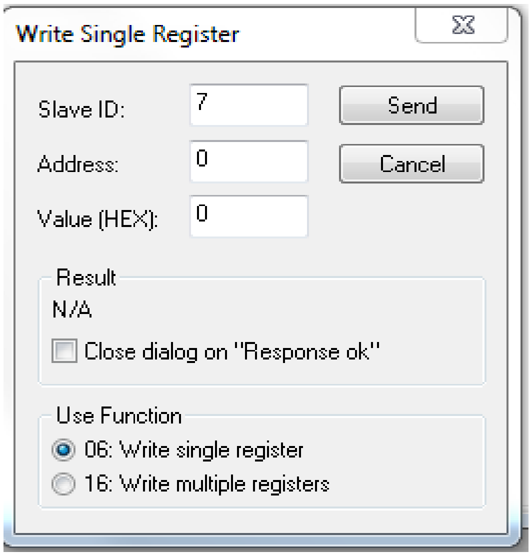

Modbus Poll - Mbpoll1
File
Edit
Connection
Setup
Functions
Display
View
Window
Help
05: Write Single Coil... Alt+F5
06: Write Single Register... Alt+F6
15: Write Coils... Alt+F7
16: Write Registers... Alt+F8
22: Mask Write Register...
23: Read/Write Registers...
Test Center... Alt+T
Mbpoll1
Alias
0
1
2
3
4
5
6
7
8
9

Tx = 230: Err = 12: ID = 7: F

06: Write single register

Port 11: 9600-8-N-1

b) It should open up a screen as per below image.



Slave ID: 7

Address: 0

Value (HEX): 0

Result

N/A

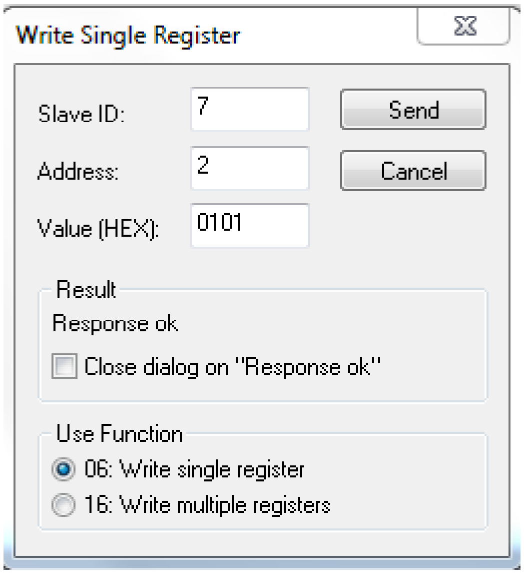
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

c) This pop Window enables us to write data in HEX value to the drive in different Address Registers. Eg. When we write HEX value 0101 in the Address register 2 with a slave ID 7, this command will enable the Digital speed control mode of the drive.
d) Once the data is input in the window, click send.
e) Response OK message will be displayed as per below.



Write Single Register

Slave ID: 7

Address: 2

Value (HEX): 0101

Result

Response ok

□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

f) Once the response OK is received, it means that the data setting is complete.
g) The complete list of registers for drive settings and their functionality and examples are provided in Appendix 1.
h) This concludes **STEP4** of the configuration procedure.

STEP 5: Set the Lines per rotation as per the encoder and motor specifications

Lines per rotation concept is very important for Position Control mode. We have two types of quad encoders for Rhino DC Servo Motors which are :

- High Precision encoder
- Quad Encoder

Both these encoders are quad encoders. Also the base motor specs are provided on in datasheet of each motor.

High Precision Encoder

Lines per rotation = 334

For one rotation of encoder, the no of steps = 334 x 4 = 1336 (As it is quad encoder)

So for 1336 steps, the base motor would move one rotation.

However as the motors are geared motors, the ratio of the gear box determines the rotation of the output shaft of the motor for each rotation of the base motor.

Eg For a motor of 200 RPM with base motor of 18000 rpm, the gear ratio is 90. Hence for a full rotation of the output shaft in this motor, the no of steps to be programmed would be 1336 x 90 = 120,240 steps.

Quad Encoder

Lines per rotation = 41

For one rotation of encoder = 41 x 4 = 164(As it is quad encoder)

So for 164 steps, the base motor would move one rotation.

For one rotation at output shaft of motor for Base Motor 18000 RPM and 200 RPM at Output shaft of Motor

One rotation of encoder x gear ratio

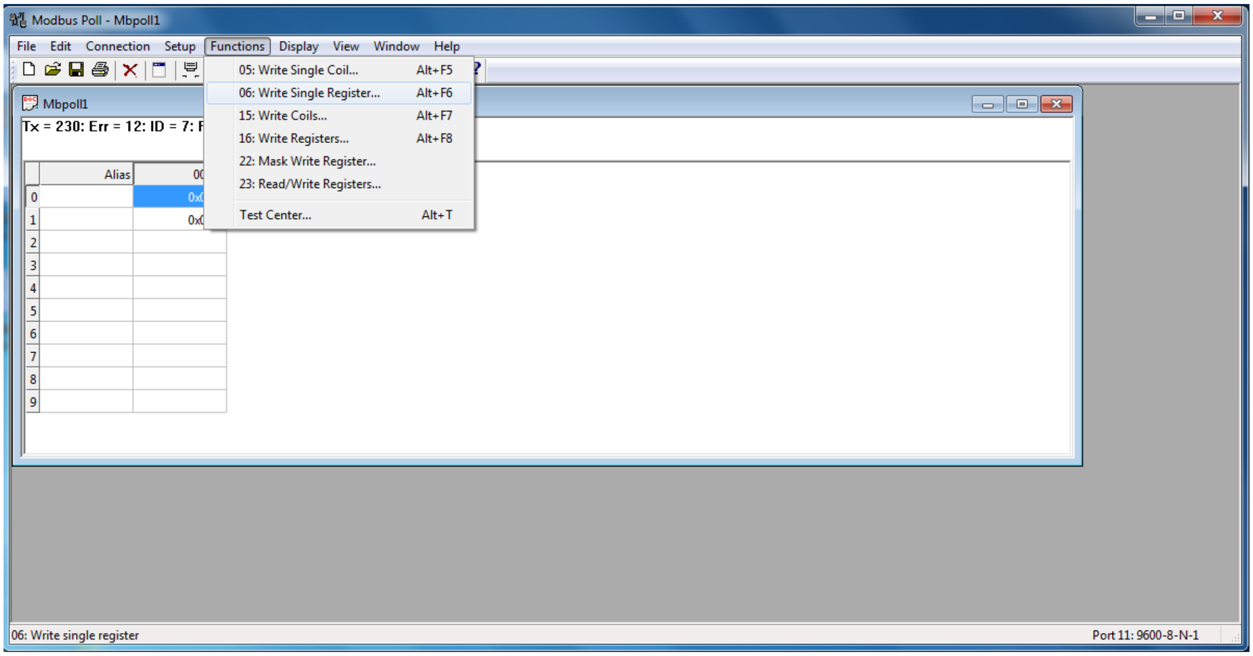
Gear ratio = Base Motor RPM / RPM at output shaft of motor

18000 / 200 = 90

So for one rotation of motor at output shaft of motor = 164 x 90 = 14760

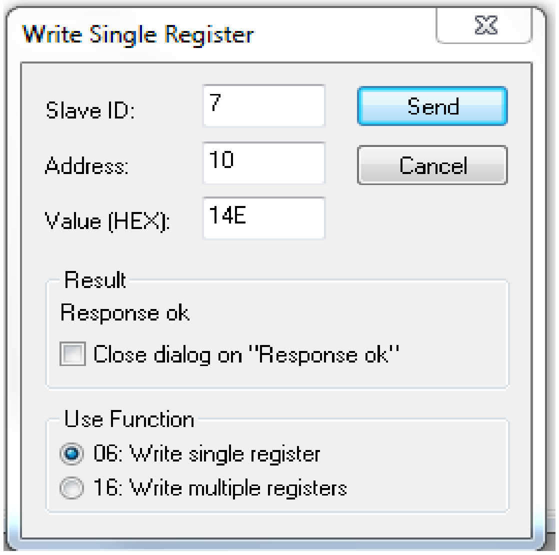
The procedure to set lines per rotation in this drive is explained below:

a) Go to Functions tab on the home screen and click on “06: Write Single Register” as per the below image.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05: Write Single Coil... Alt+F5
06: Write Single Register... Alt+F6
15: Write Coils... Alt+F7
16: Write Registers... Alt+F8
22: Mask Write Register...
23: Read/Write Registers...
Test Center... Alt+T
Mbpoll1
Alias
06: Write single register
Port 11: 9600-8-N-1

Tx = 230: Err = 12: ID = 7: F

b) Send lines per rotation Hex value to address 10. For High Precision encoder motor lines per rotation is 334 so send Hex value 14E (334) to address 10 as per below image.



Write Single Register

Slave ID: 7

Address: 10

Value (HEX): 14E

Result

Response ok

□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

c) This concludes **STEP 5** of the configuration procedure.

STEP 6: Program the drive to run in any one of the three modes:

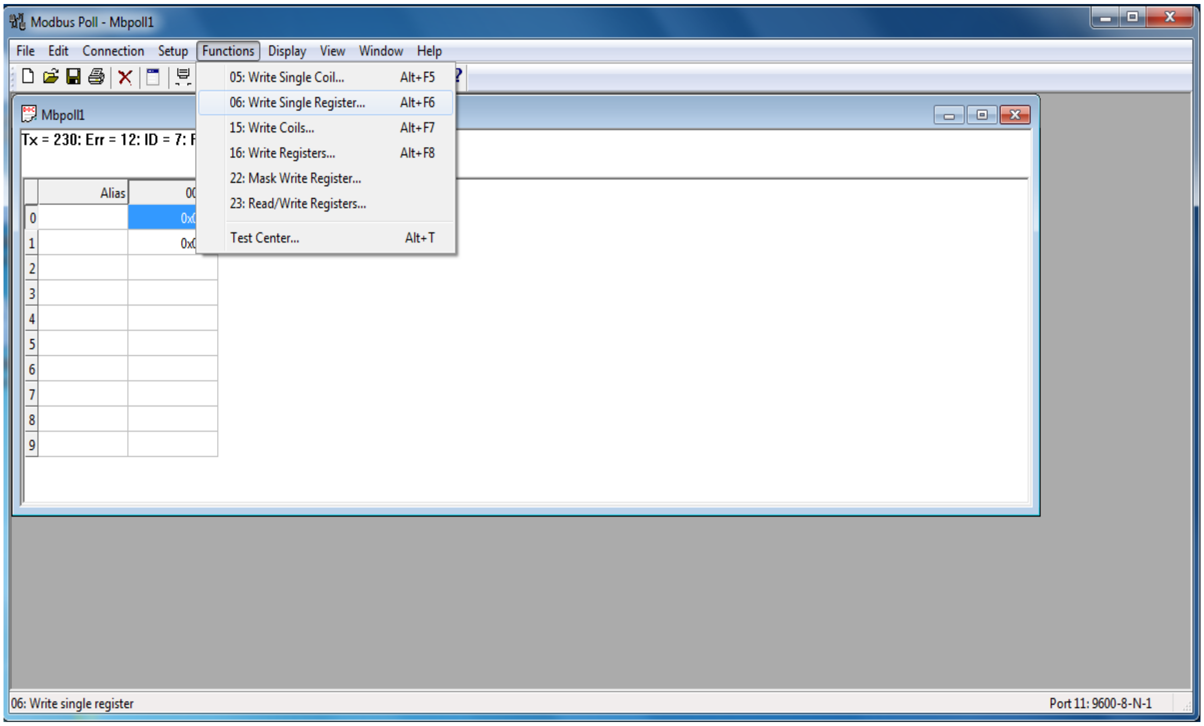
- **Mode0 - Analog Control Mode**
- **Mode1 - Digital Speed Control Mode**
- **Mode2 - Position Control Mode**

Mode 0 –Analog Control Mode

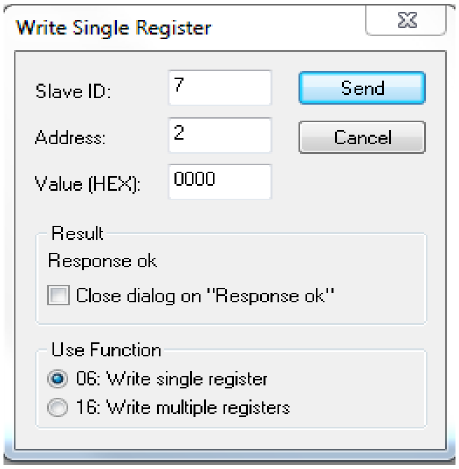
a) For analog control mode we need to have the proper hardware connections as per STEP 1 and set the values in address 2 as per below table which is described from point no. b.

Slave ID	Address	Value(hex)	Description
7	2	0000	Enables mode 0 (analog mode)
7	2	0001	Enable the motor

b) To prgram the address click the functions tab and click write singe register as per below image.


Modbus Poll - Mbpoll1
File
Edit
Connection
Setup
Functions
Display
View
Window
Help
05: Write Single Coil... Alt+F5
06: Write Single Register... Alt+F6
15: Write Coils... Alt+F7
16: Write Registers... Alt+F8
22: Mask Write Register...
23: Read/Write Registers...
Test Center... Alt+T
Mbpoll1
Tx = 230: Err = 12: ID = 7: F
Alias
0
1
2
3
4
5
6
7
8
9
06: Write single register
Port 11: 9600-8-N-1

c) This should open up a window as per below image. Write the HEX value 0000 to address 2 to enable mode 0 and click send.


Write Single Register

Slave ID: 7

Address: 2

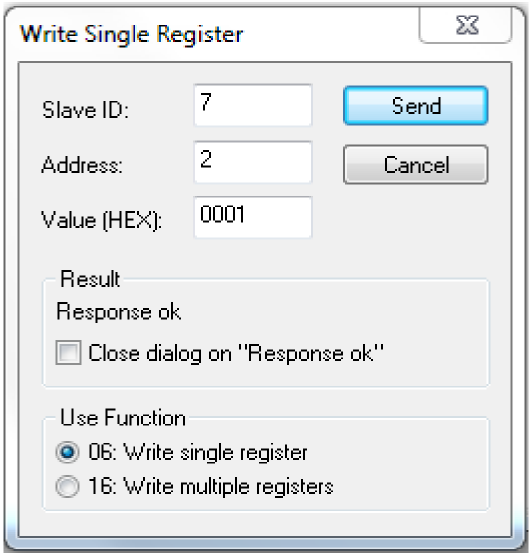
Value (HEX): 0000

Result
Response ok

□ Close dialog on "Response ok"

Use Function
● 06: Write single register
○ 16: Write multiple registers

d) After receiving response OK in the window, we need to enable motor.
e) Send Hex value 0001 to address 2 as per below image to enable the motor.



Slave ID: 7
Address: 2
Value (HEX): 0001

Response ok

□ Close dialog on "Response ok"

● 06: Write single register
○ 16: Write multiple registers

f) For analog mode connect external potentiometer to change speed of motor.
g) Connect enable (Pin 4) of drive to GND (Pin 8) to enable motor in analog mode.
h) User can change direction of motor in analog mode to connect the Direction (Pin 6) to the GND (Pin 8).
i) User can lock motor by connecting brake (Pin 7) to GND (Pin 8).
j) This concludes the settings for Mode0.

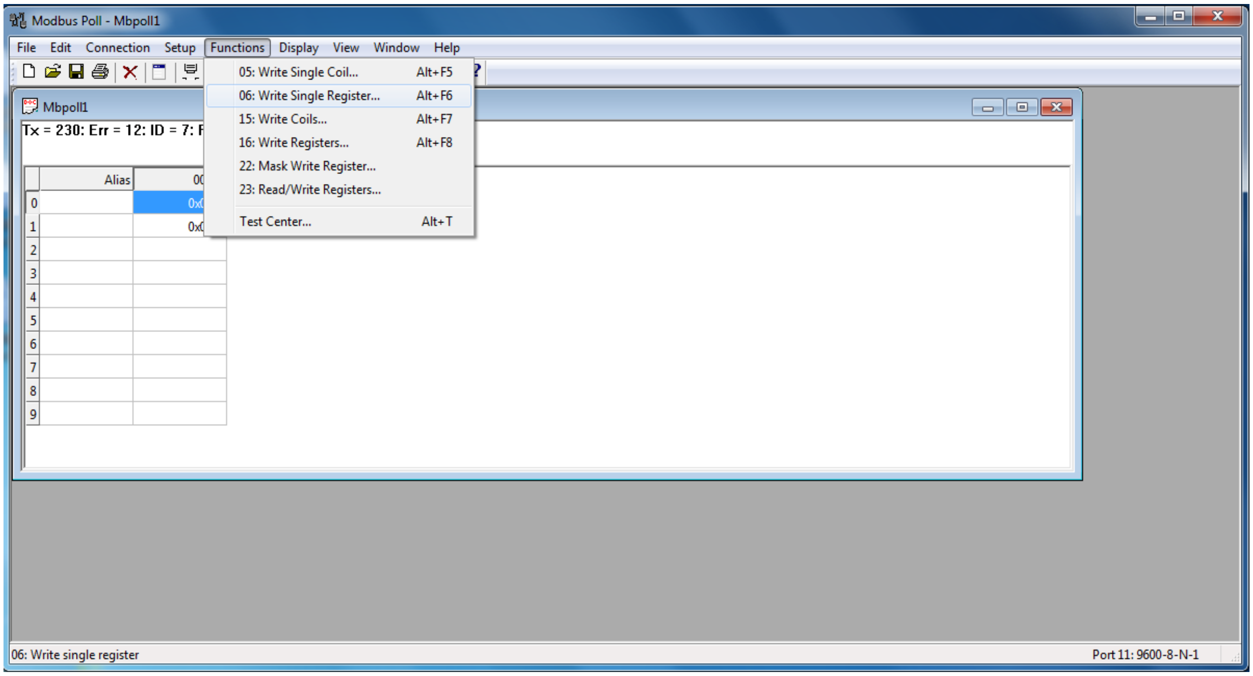
The Gain and acceleration of the motor can be set as per described in Apendix 1. If you intend to use the motor in Mode0 as described above, please skip to Step 7 to save settings.

Mode 1 – Digital Speed Control Mode

a) For digital speed control mode the register values need to be set as per the below table. (The procedure for same along with screenshots is provided after the table) refer below table settings and procedure is also explained below from point b.

Slave ID	Address	Value(hex)	Description
7	2	0100	Enable Mode 1(digital mode)
7	2	0101	Enable the motor
7	2	0109	Changes the direction of motor. Direction changes in low speed then user can increase speed of motor.
7	14	For example: Min- 0 Max-0x4650(hex),18000(decimal)	Changes the speed of the motor as per Base Motor RPM.
7	2	0000	Motor stops

b) To prgram the address click the functions tab and click write singe register as per below image.

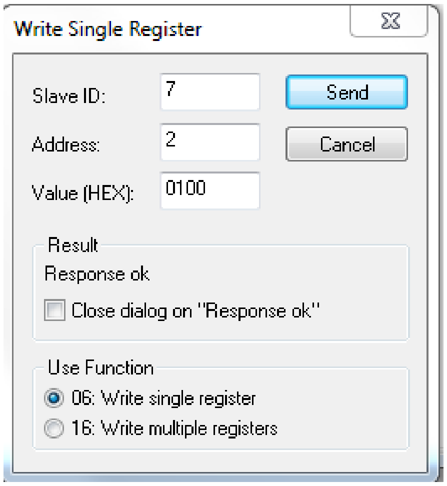

Modbus Poll - Mbpoll1
File
Edit
Connection
Setup
Functions
Display
View
Window
Help
05: Write Single Coil... Alt+F5
06: Write Single Register... Alt+F6
15: Write Coils... Alt+F7
16: Write Registers... Alt+F8
22: Mask Write Register...
23: Read/Write Registers...
Test Center... Alt+T
Mbpoll1
Alias
0
1
2
3
4
5
6
7
8
9

Tx = 230: Err = 12: ID = 7: F

06: Write single register

Port 11: 9600-8-N-1

c) This should open up a window as per below image. Write the HEX value 0100 to address 2 to enable mode 1 and click send.



Write Single Register

Slave ID: 7

Address: 2

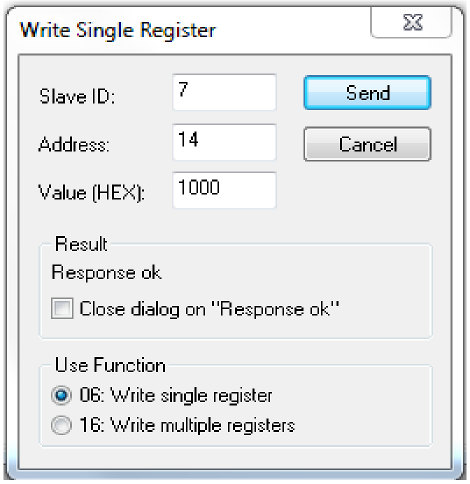
Value (HEX): 0100

Result
Response ok

□ Close dialog on "Response ok"

Use Function
● 06: Write single register
○ 16: Write multiple registers

d) Now we need to set speed in digital speed control mode. The Speed is as per rpm of base motor. Eg. For a base motor with 18,000 rpm and the gear ratio of 180, the motor at maximum rpm would be 100 rpm at output shaft. So to set maximum RPM for the motor we would have to program 4650 (HEX value of 18000) in the value and send. To set 50 RPM output for the motor, we would have to set 50% rpm for the base motor. So we would set 2328 (HEX value of 18000) in the register 14. In the below image we have set the HEX value of 1000 which gives 4096 rpm to base motor.



Write Single Register

Slave ID: 7

Address: 14

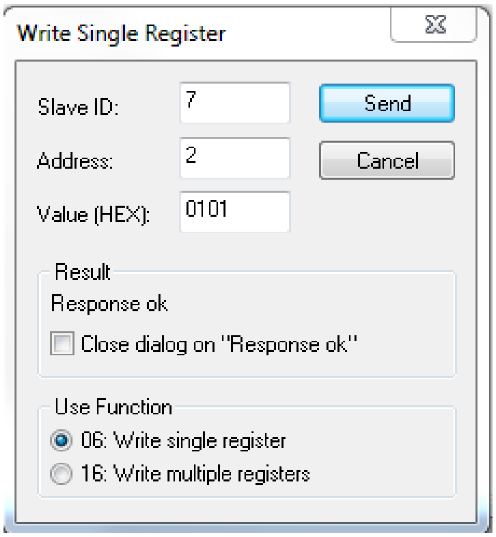
Value (HEX): 1000

Result
Response ok

□ Close dialog on "Response ok"

Use Function
● 06: Write single register
○ 16: Write multiple registers

e) After speed setting user needs to enable motor by sending Hex value 0101 to address 2 as per below image.



Write Single Register

Slave ID: 7

Address: 2

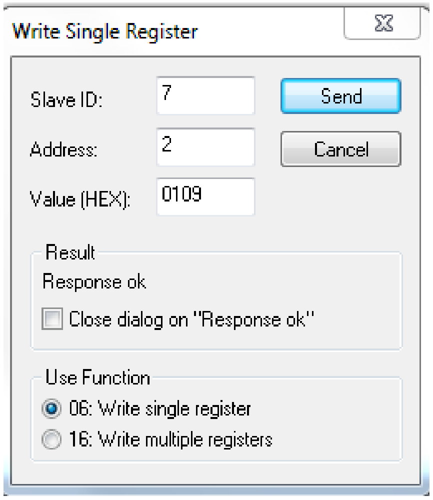
Value (HEX): 0101

Result
Response ok

□ Close dialog on "Response ok"

Use Function
● 06: Write single register
○ 16: Write multiple registers

f) To change direction of motor send Hex value 0109 to address 2 as per below image.



Write Single Register

Slave ID: 7

Address: 2

Value (HEX): 0109

Result

Response ok

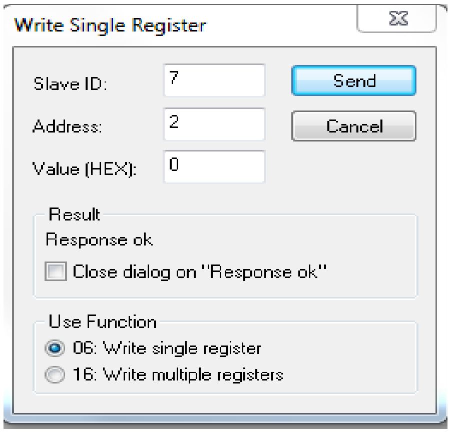
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

g) To stop motor send Hex value of 0 to address 2 as per below image.



Write Single Register

Slave ID: 7

Address: 2

Value (HEX): 0

Result

Response ok

□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

h) This concludes the settings for Mode1.

The Gain and acceleration of the motor can be set as per described in Apendix 1. If you intend to use the motor in Mode1 as described above, please skip to Step 7 to save settings.

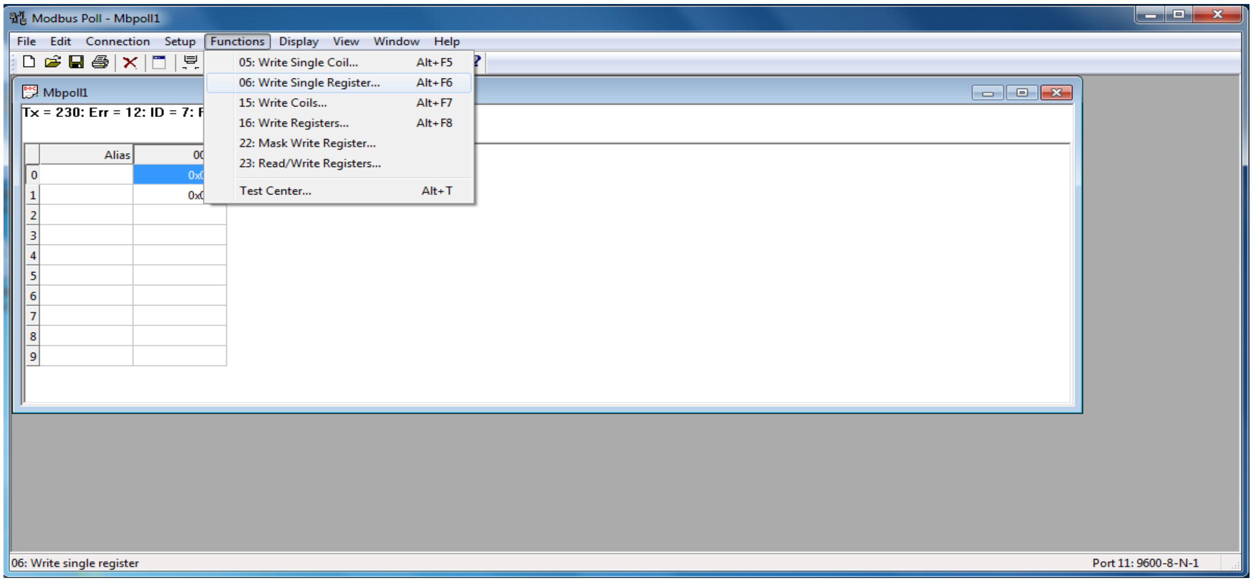
Mode 2 – Position Control Mode

a) For position control mode the register values need to be set as per the below table. (The procedure for same along with screenshots is provided after the table) refer below table settings and procedure is also explained below

Slave ID	Address	Value(Hex)	Description
7	2	0200	Enters into Mode 2
7	16 (LSB)	For 334 lines per rotation 538 (334x4=1336)	Units for command are in steps, so depends on the lines per rotation of the encoder. This Hex Value 538 is for one rotation of encoder.
	18 (MSB)	0	Units for command are in steps, so depends on the lines per rotation of the encoder
7	2	0201	Enable the Motor. First set rotation of encoder then enable motor

7	16(LSB)
	18(MSB)
		0000	Initial position
			(Zero Position)

b) To prgram the address click the functions tab and click write singe register as per below image.

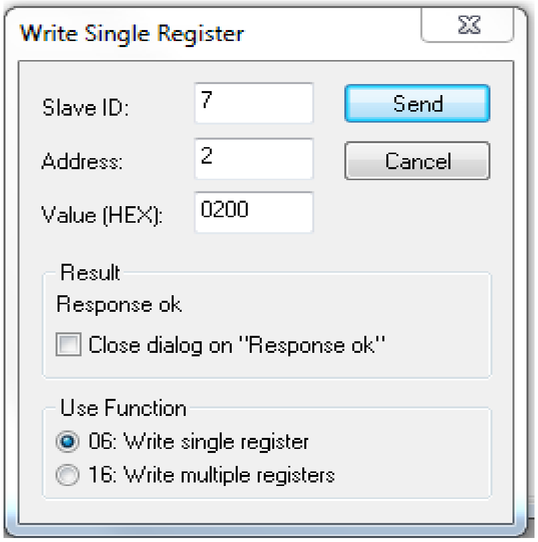

Modbus Poll - Mbpoll1
File
Edit
Connection
Setup
Functions
Display
View
Window
Help
05: Write Single Coil... Alt+F5
06: Write Single Register... Alt+F6
15: Write Coils... Alt+F7
16: Write Registers... Alt+F8
22: Mask Write Register...
23: Read/Write Registers...
Test Center... Alt+T
Mbpoll1
Alias
00
0x0
0x0
0
1
2
3
4
5
6
7
8
9

Tx = 230: Err = 12: ID = 7: F

06: Write single register

Port 11: 9600-8-N-1

c) For Position control mode (mode 2) send Hex value 0200 to address 2 to enable this mode as per below image.



Write Single Register

Slave ID: 7

Address: 2

Value (HEX): 0200

Result

Response ok

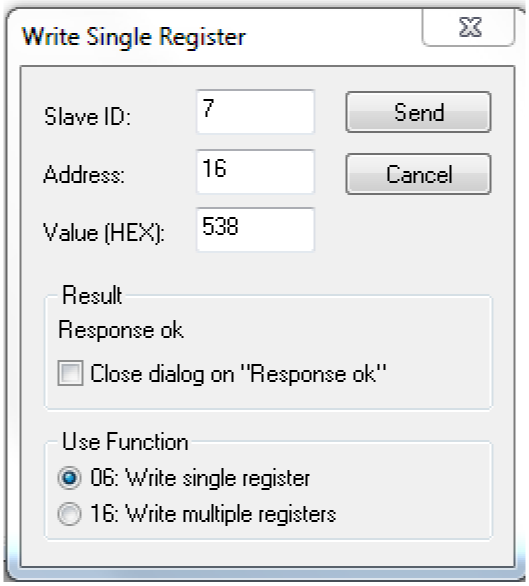
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

d) User can set position of motor as per requirement. For 334 lines per rotation user need to send Hex value 538 (Decimal value is 1336) to address 16 for 1 rotation of encoder as per below image. To understand lines per rotation concept refer page 16.



Write Single Register

Slave ID: 7

Address: 16

Value (HEX): 538

Result

Response ok

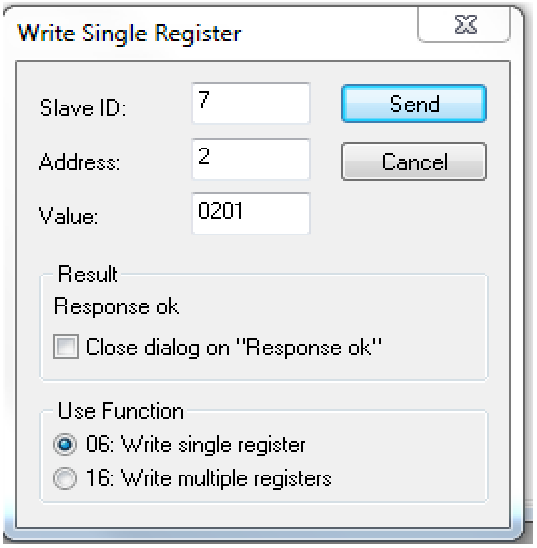
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

e) To enable motor in position control mode send Hex value 0201 to address 2 as per below image.


Write Single Register

Slave ID: 7

Address: 2

Value: 0201

Result

Response ok

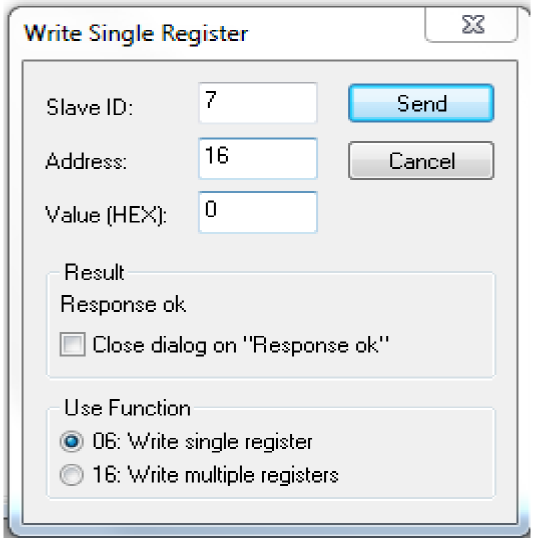
☐ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

f) To set encoder at initial position send Hex value 0000 to address 16 as per below image.


Write Single Register

Slave ID: 7

Address: 16

Value (HEX): 0

Result

Response ok

☐ Close dialog on "Response ok"

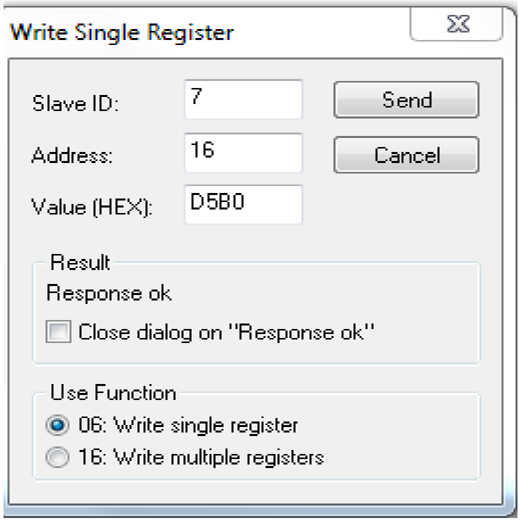
Use Function

● 06: Write single register

○ 16: Write multiple registers

g) The procedure to set 1 rotation at output shaft of motor is explained below with images. Refer Page 14 for calculations. For one rotation at output shaft (334 lines per rotation) Hex value is 1D5B0 (120240 decimal value). Register 16 and Register 18 is 16 Bit so you can send maximum value up to 65535 in one register. This value is greater than 65535 so use need to send value in both register (LSB and MSB) and then enable motor.

h) Send Hex value D5B0 to address 16 as per below image.



Write Single Register

Slave ID: 7

Address: 16

Value (HEX): D5B0

Result

Response ok

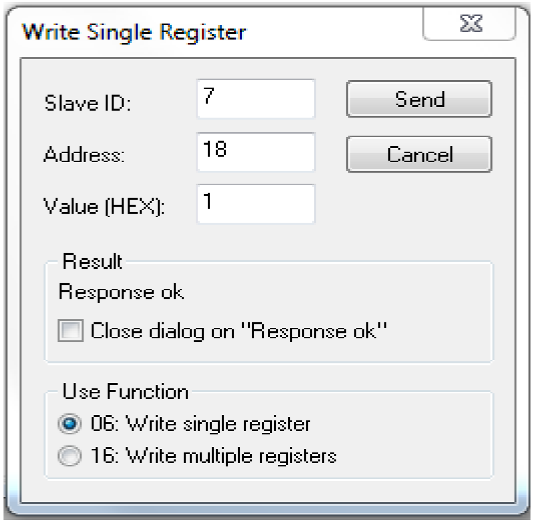
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

i) Send Hex value 1 to address 18 as per below image.



Write Single Register

Slave ID: 7

Address: 18

Value (HEX): 1

Result

Response ok

□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

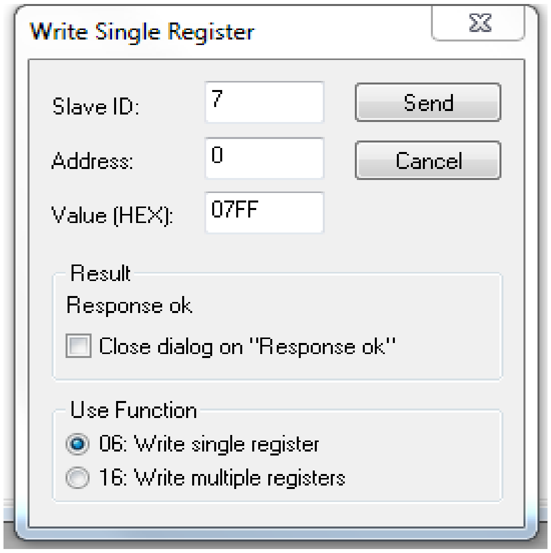
j) Then enable the motor by sending Hex value 0201 to address 2. Motor takes one rotation at output shaft. By this method user can set motor at any position.

i) This concludes the settings for Mode2.

The Gain and acceleration of the motor can be set as per described in Apendix 1. If you intend to use the motor in Mode2 as described above, proceed to Step 7 to save settings.

STEP 7: Save and Reset settings in Drive

a) To save settings in drive send Hex value 07FF to address 0. If user saves setting in drive then user just need to power up drive then motor will run according to saved setting. Users don't need to connect the drive through software again to run the motor. In hex value 07FF 07 is slave ID and FF value is ignored.



Write Single Register

Slave ID: 7

Address: 0

Value (HEX): 07FF

Result

Response ok

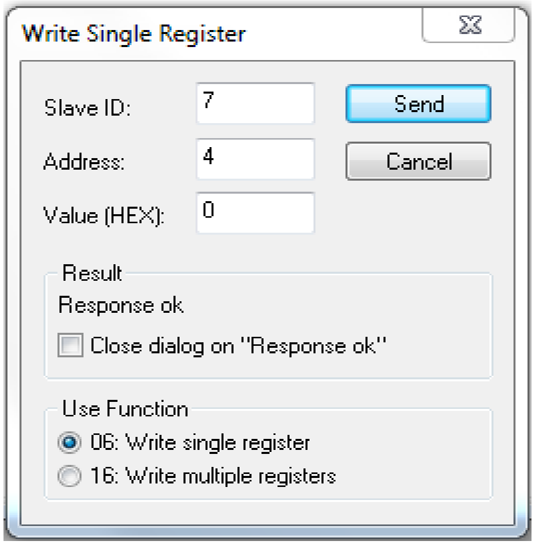
☐ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

b) To Reset setting in drive send Hex value 0 to address 4 as per below image then save to eeprom by sending Hex value 07FF to address 0. Then on power reset the drive will have original parameters.



Write Single Register

Slave ID: 7

Address: 4

Value (HEX): 0

Result

Response ok

☐ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

c) This concludes **STEP 7** of the configuration procedure.

Appendix 1

In this section we describe all the MODBUS register and additional settings like gain and acceleration to optimize the motor driving parameters depending on the load and application of the motor. However, we would recommend to change the advanced settings after gaining some experience in using the drive. Also please test the changed settings in actual use case before finalizing same to make sure drive performs as per requirement. For going back to default mode please refer STEP 7.

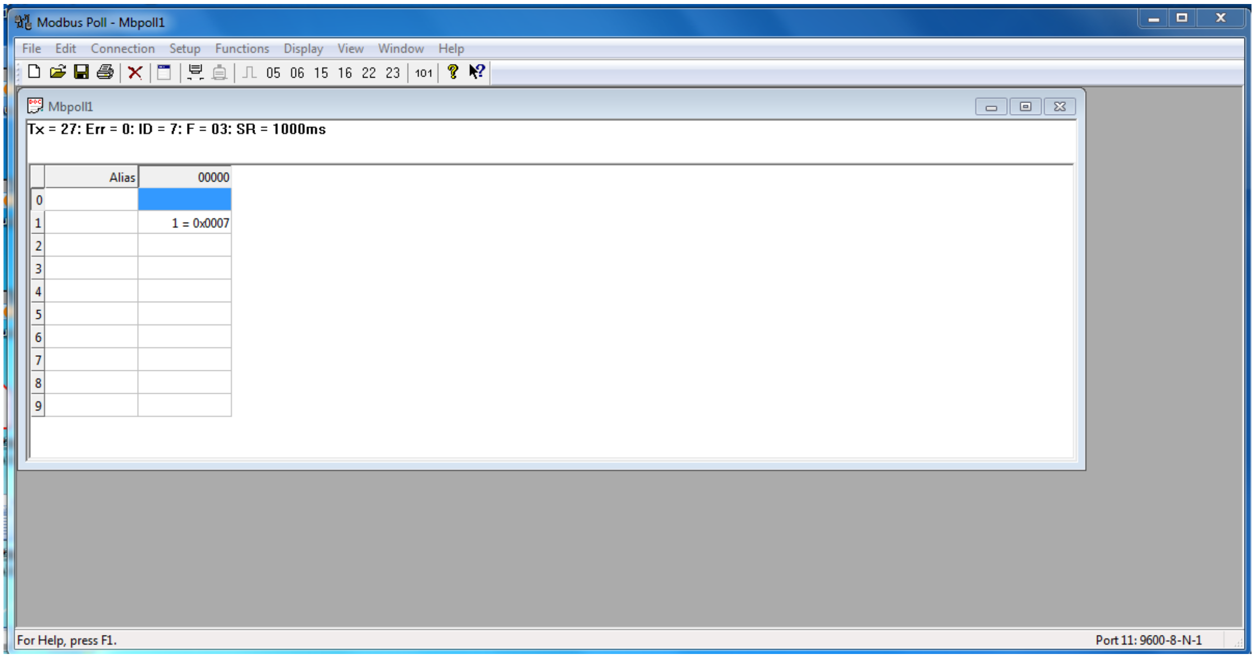
Modbus Registers:

Control Register	Modbus Input Register	No. of Bit	Maximum Value	Default Values (Hex)	Description
DEVICE_MODBUS_ADDRESS	1	-		0007	Device Address
INP_CONTROL_BYTE	2	8 Bit	255	0000	Input Control Byte
INP_MODE_BYTE	3	8 Bit	255	2000	Input Mode Byte
Motor and Encoder Dependent	**Modbus Input Register**	**No. of Bits**	**Maximum Value**	**Default Values (Hex)**	**Description**
VP_GAIN_BYTE	4	8 Bit	255	FF20	Velocity proportional gain
VI_GAIN_BYTE	6	8 Bit	255	FF10	Velocity integral gain
VF_GAIN_BYTE	8	8 Bit	255	FF20	Velocity feed forward gain
LINES_PER_ROT	10	16 Bit	65535	14E	Lines per Rotation
Motion Profiles	**Modbus Input Register**	**No. of Bits**	**Maximum Value**	**Default Values (Hex)**	**Description**
TRP_ACL_WORD	12	16 Bit	65535	4E20	Acceleration
TRP_SPD_WORD	14	16 Bit	65535	0800	speed of motor
CMD_LSB_WORD	16	32 Bit	65535	0000	Units for command are in steps that depends on the lines per rotation of the encoder
CMD_MSB_WORD	18		65535	0000	Units for command are in steps that depends on the lines per rotation of the encoder
User Feedback	**Modbus Input Register**	**No. of Bits**	**Maximum Value**	**Default Values (Hex)**	**Description**
POS_LSB_WORD	20	32 Bit	65535	0000	Position feedback
POS_MSB_WORD	22		65535	0000	Position feedback
ACT_SPD_WORD	24	16 Bit	65535	0048	Speed feedback

Brief Description of all configuration registers.

A brief decription of the procedure to use the registers to configure the drive is provided below for registers listed in the above table

Register 1: This register shows device address. Its default value is 0007 where 00 shows mode is disabled & 07 is slave ID.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1

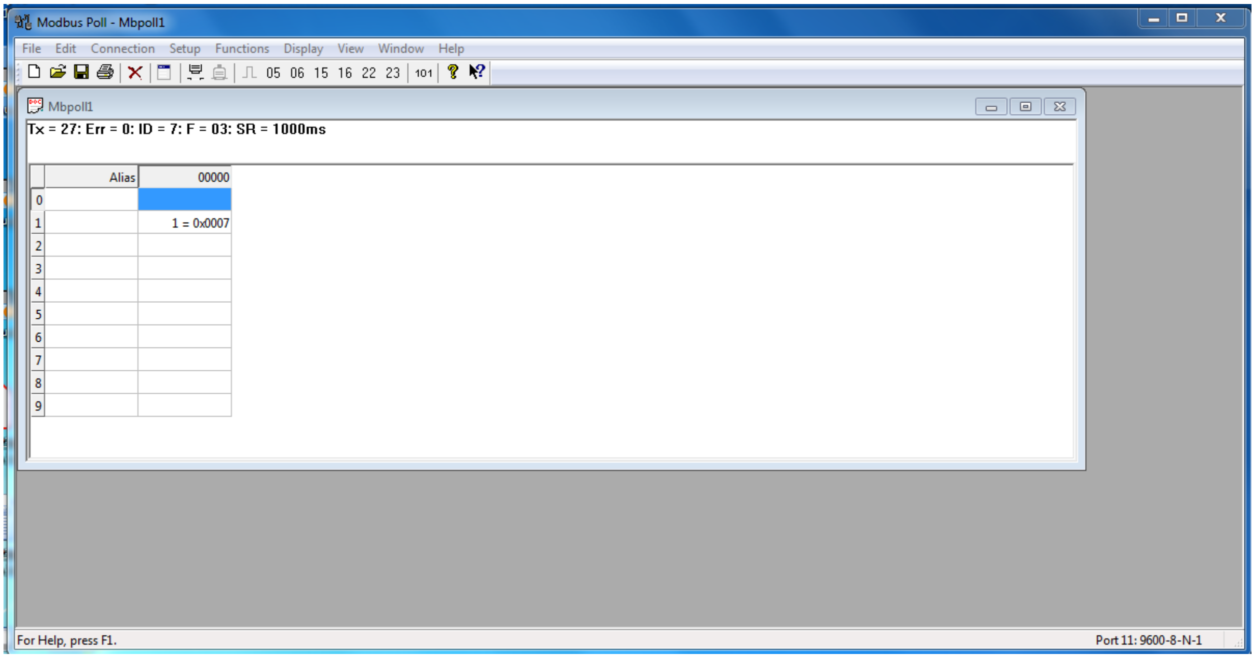
Tx = 27: Err = 0: ID = 7: F = 03: SR = 1000ms

	Alias	00000
0		
1		1 = 0x0007
2		
3		
4		
5		
6		
7		
8		
9		

For Help, press F1.

Port 11: 9600-8-N-1

Register 1 is related to register 2. As per below image when user sends Hex value 0101 to register 2 to enable mode 1 then this value becomes 0107 where 01 shows mode is enable and 07 is slave ID.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1

Tx = 27: Err = 0: ID = 7: F = 03: SR = 1000ms

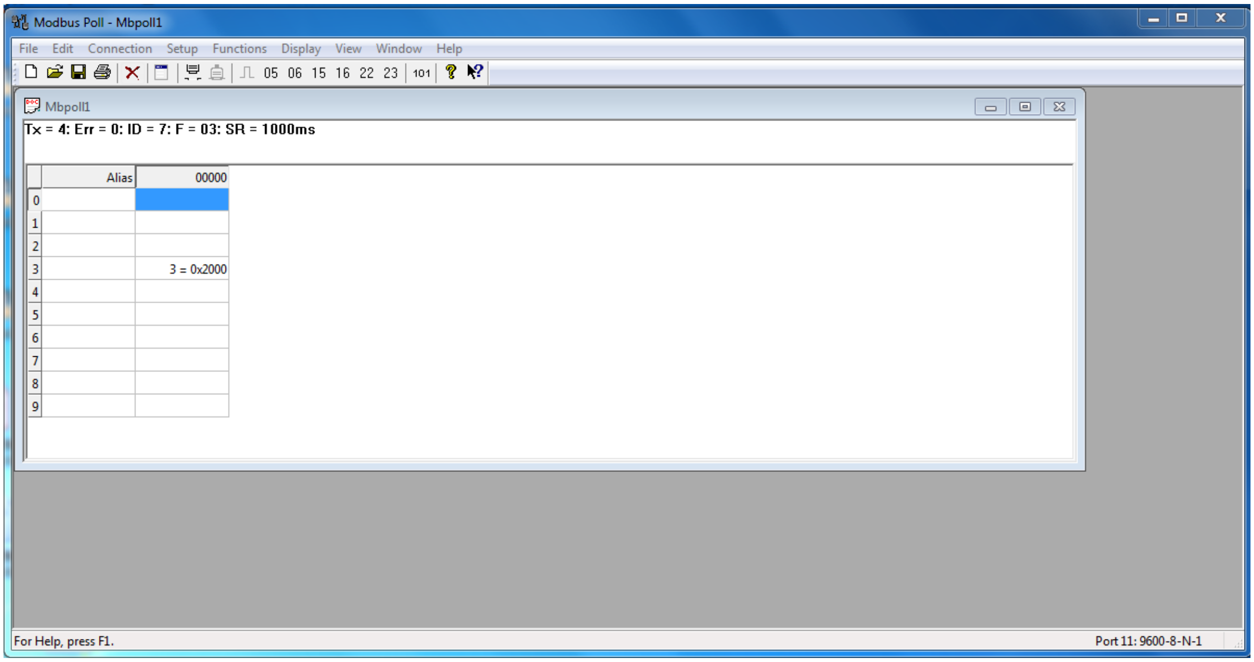
	Alias	00000
0		
1		1 = 0x0007
2		
3		
4		
5		
6		
7		
8		
9		

For Help, press F1.

Port 11: 9600-8-N-1

Register 2: This Register is Input Control Byte. It is already explained in STEP 6

Register 3: This register is Input Mode Byte. It is related to register 2. Refer below image for understanding its default Hex value is 2000 where 2 is Input control byte register and 00 shows it is in mode 0.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1

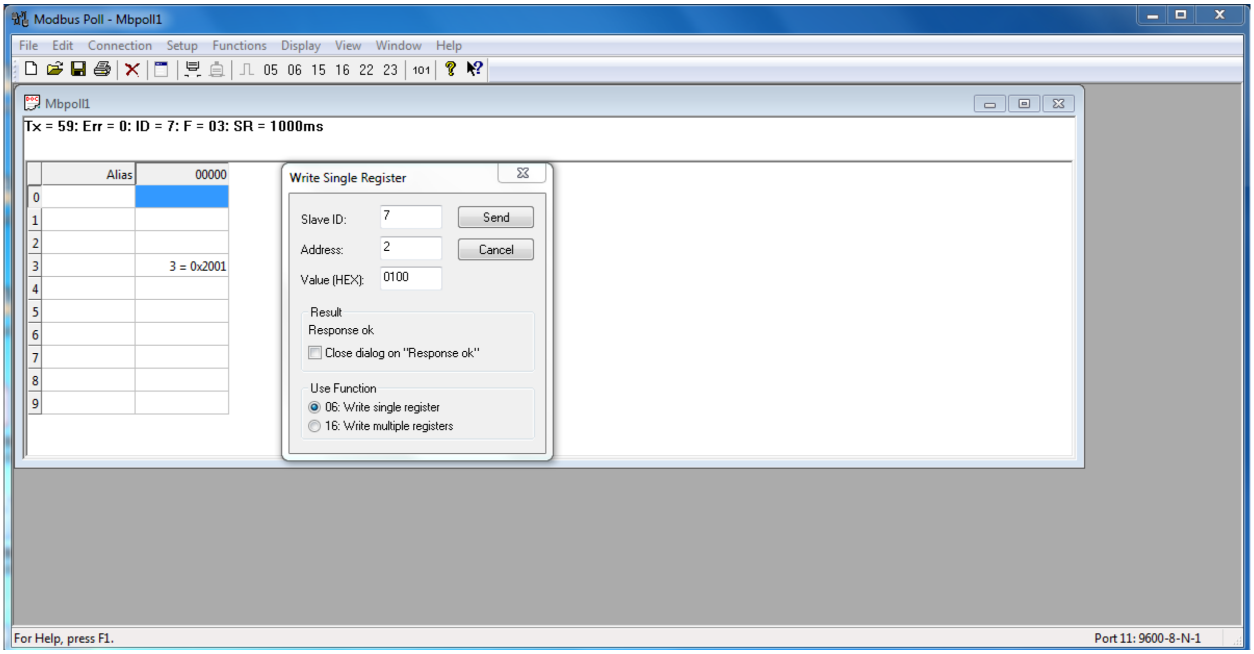
Tx = 4: Err = 0: ID = 7: F = 03: SR = 1000ms

	Alias	00000
0		
1		
2		
3		3 = 0x2000
4		
5		
6		
7		
8		
9		

For Help, press F1.

Port 11: 9600-8-N-1

As per below image when user sends Hex value 0100 in register 2 to enable mode 1 then register 3 value becomes 2001 where 2 is register 2 and 01 is mode 1.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1
Send
Cancel

Tx = 59: Err = 0: ID = 7: F = 03: SR = 1000ms

	Alias	00000
0		
1		
2		
3		3 = 0x2001
4		
5		
6		
7		
8		
9		

Write Single Register

Slave ID: 7

Address: 2

Value (HEX): 0100

Result

Response ok

□ Close dialog on "Response ok"

Use Function

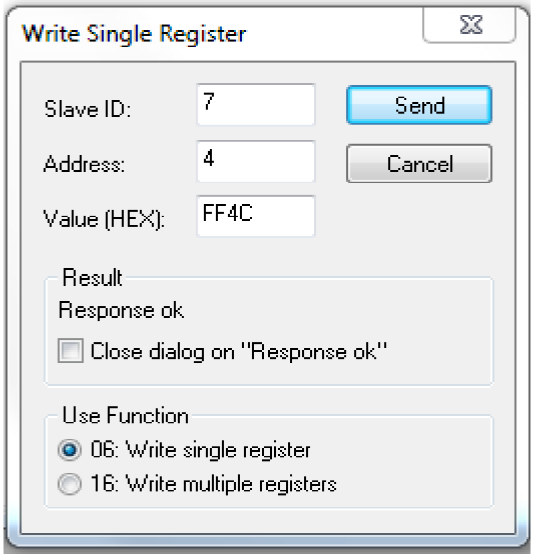
● 06: Write single register

○ 16: Write multiple registers

For Help, press F1.

Port 11: 9600-8-N-1

Register 4: This register is for Velocity Proportional gain (P gain parameter). Its default value is FF20 (-224). User can change this value as per requirement. Refer below method to change Velocity proportional gain of motor. Eg. Hex value FF4C (-180) send to register 4 to set velocity proportional gain of motor to 180 as in below image.



Write Single Register

Slave ID: 7

Address: 4

Value (HEX): FF4C

Result

Response ok

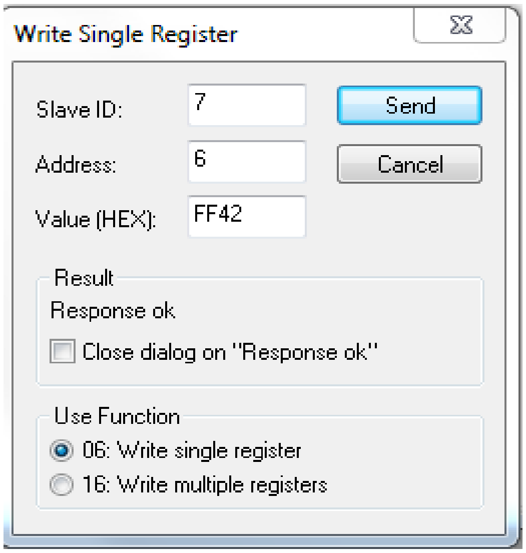
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

Register 6: This register is for Velocity Integral gain (I gain parameter). Its default value is FF10 (-240). User can change this value as per requirement. Refer below method to change Velocity Integral gain of motor. Eg.Hex value FF42 (-190) send to register 6 to set velocity Integral gain of motor to 190 as in below image.



Write Single Register

Slave ID: 7

Address: 6

Value (HEX): FF42

Result

Response ok

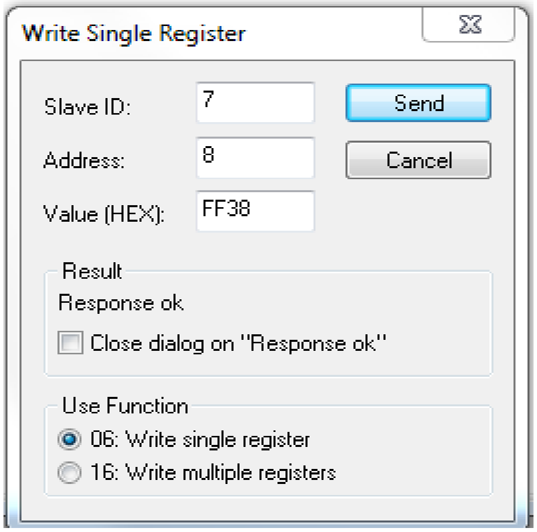
□ Close dialog on "Response ok"

Use Function

● 06: Write single register

○ 16: Write multiple registers

Register 8: This register is for Velocity Feed Forward gain. Its default value is FF20 (-240). User can change this value as per requirement. Refer below method to change Velocity Feed Forward gain of motor. eg Hex value FF38 (-200) send to register 8 to set velocity feed forward gain of motor to 200 as in below image .



Write Single Register

Slave ID: 7

Address: 8

Value (HEX): FF38

Result

Response ok

□ Close dialog on "Response ok"

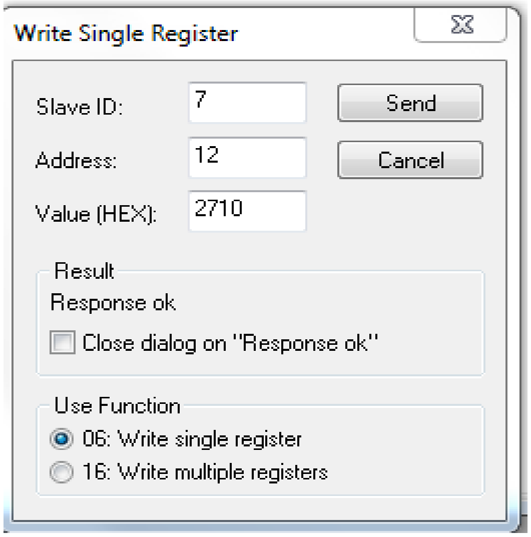
Use Function

● 06: Write single register

○ 16: Write multiple registers

Register 10: This register is for setting of lines per rotation. It is already explained in STEP 5 of document

Register 12: This register is for setting of acceleration. Its default Hex value is 4E20 (20000).User can change this value as per requirement. Eg. Hex value 2710 (10000) is send to register 12 to set acceleration of motor to 10000.



Write Single Register

Slave ID: 7

Address: 12

Value (HEX): 2710

Result

Response ok

□ Close dialog on "Response ok"

Use Function

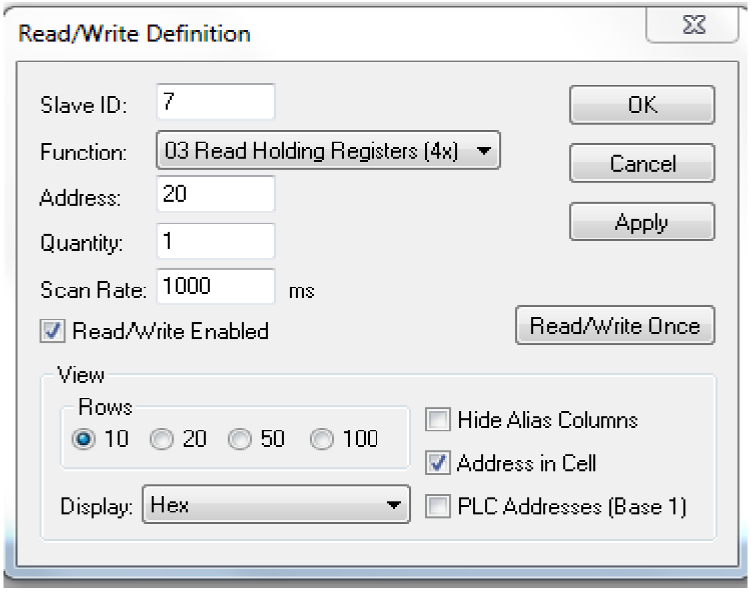
● 06: Write single register

○ 16: Write multiple registers

Register 14: This register is used to set speed of motor. It is already explained in Mode1 – Digital Speed Control mode settings on Page 20

Register 16 and Register 18: These register are used for Position control mode. It is explained on Page 22 in Mode2 setting.

Register 20 and Register 22: These registers are used for position feedback of encoder. Default value is 0000. User can read feedback of encoder by right click on the address table and select Read/Write definition a window will open and do same setting as per below image. This is an example to read Register 20.



Slave ID: 7

Function: 03 Read Holding Registers (4x)

Address: 20

Quantity: 1

Scan Rate: 1000 ms

☑ Read/Write Enabled

Rows: ● 10 ○ 20 ○ 50 ○ 100

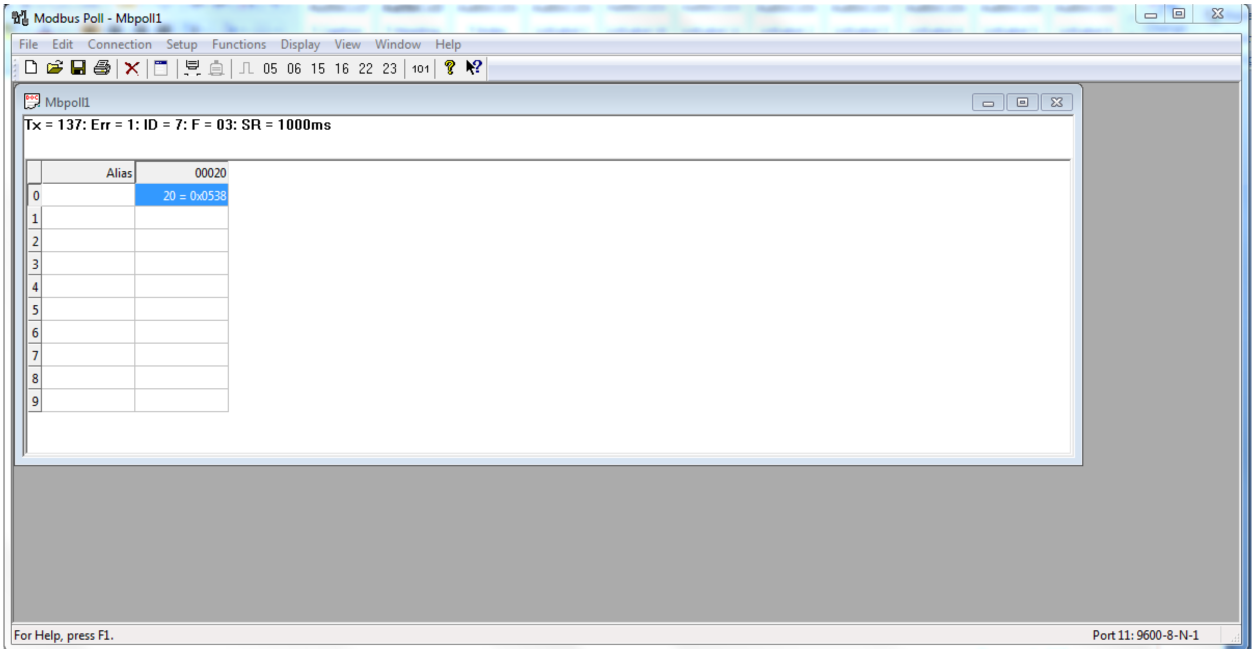
□ Hide Alias Columns

☑ Address in Cell

Display: Hex

□ PLC Addresses (Base 1)

For example if user set one rotation of encoder (1336 = 4 x 334 lines per rotation) then position feedback of encoder is Hex value 538 (334 x 4 = 1336). From register 20 user can read LSB position of motor and from register 22 user can read MSB position of encoder.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1

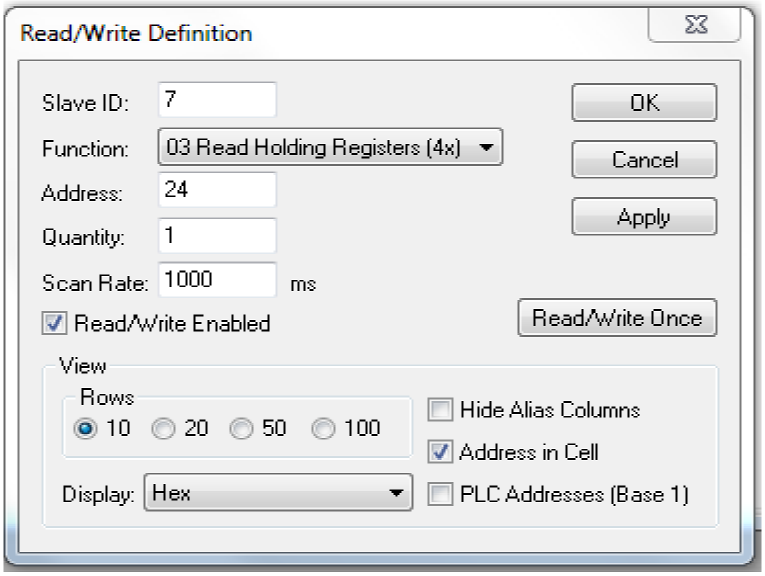
Tx = 137: Err = 1: ID = 7: F = 03: SR = 1000ms

	Alias	00020
0		20 = 0x0538
1		
2		
3		
4		
5		
6		
7		
8		
9		

For Help, press F1.

Port 11: 9600-8-N-1

Register 24: This register is used to take speed feedback of motor. Its default value is 0800 When motor is in operation, user can read feedback of motor. Right click on address table and select Read/Write definition a window will open refer same setting as per below image.



Read/Write Definition

Slave ID: 7

Function: 03 Read Holding Registers (4x)

Address: 24

Quantity: 1

Scan Rate: 1000 ms

☑ Read/Write Enabled

View

Rows: ● 10 ○ 20 ○ 50 ○ 100

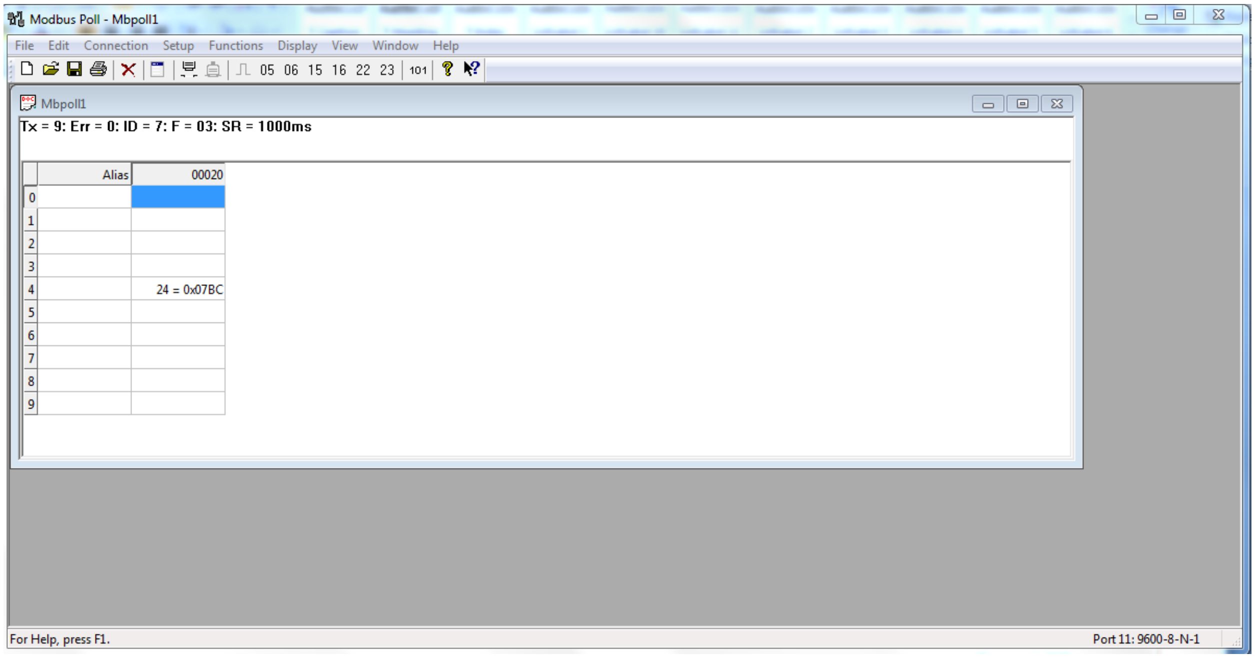
Display: Hex

□ Hide Alias Columns

☑ Address in Cell

□ PLC Addresses (Base 1)

Then user can read feedback of motor speed as per below image.



Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 22 23 101
Mbpoll1

Tx = 9: Err = 0: ID = 7: F = 03: SR = 1000ms

	Alias	00020
0		
1		
2		
3		
4		24 = 0x07BC
5		
6		
7		
8		
9		

For Help, press F1.

Port 11: 9600-8-N-1

Troubleshooting:

- Timeout Error:
  - When slave address is not set. Set slave ID.
- Byte Missing Error:
  - Check Connections and reset power.
  - Check Jumpers connection as per slave ID.
  - If no jumper is connected then default slave ID is 7.
- Error in position control Mode :
  - Check Lines per rotation value.
  - If Motor is not moving check speed of Motor.
  - Change direction of motor in low speed only.