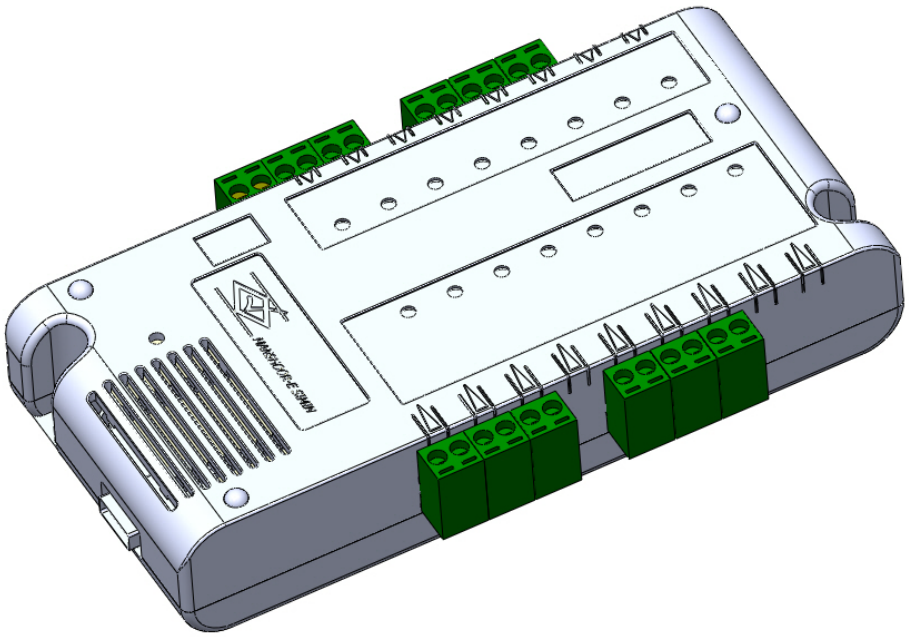
NANO Stepper, AC, DC Motor
4 Channels



Stepper,AC,DC motor Interface from a PC's USB port
Installation and Users Manual
&
Software Guide

Available exclusively from
Manshoore Simin Ltd Co.
www.msbbs.com

Contents

1. Introduction

The electric motors are being produced in variety of models & types. But 3 main types & models are:

- Alternating Current (AC) motors.
- Direct Current (DC) motors
- Stepping motor;

These common types of motors describes in below, nutshell.

- AC motors are further subdivided into single phase and three-phase motors. Single phase AC electrical supply is what is typically supplied in a home. Three-phase electrical power is commonly only available in a factory setting. The most common single phase AC motor is known as a universal motor. This is because this motor can also run with DC current.

This type of motor is very inefficient but can be very inexpensively made. It is also used almost exclusively for small factional horse power AC motors. The other advantage this AC motor has is that the rotational speed of the motor can be easy changed. This type of AC motor is commonly found in mixers, hand drills, and any other application requiring variable speed and low cost and small size.

For larger single phase AC motors, an electrical component known as a capacitor is used to create a second phase from the single phase AC current. This type of AC motor is known as an induction motor and there are two basic types; a capacitor start motor and a capacitor run motor. The capacitor is used to create a second phase from the single phase power source and it is the interaction between these two-phases that causes the motor to turn.

This introduction of a second-phase eliminates the need for the brushes used in a universal AC motor. This type of motor is a fixed speed motor. It is commonly used as the drive for refrigerator compressors, shop air compressors, and as a general utility type AC motor.

AC motors are usually sized in horsepower. The most common sizes are what are called fractional horsepower motors, i.e. ½ horse power or ¼ horsepower. Larger motors are typically only found in factories, where they can range in size to thousands of horsepower.

AC motors also come with various speed ratings. Speed is usually specified as rotations per minute (RPM) at no load condition. As the motor is loaded down, the speed will slow down. When the AC motor is running at its rated power draw, the speed of the shaft measured in RPM is the full load speed. If the electric motor is loaded too heavily, the motor shaft will stop. This is

known as the stall speed and should be avoided. All of these speeds are typically listed on the specification sheet for an AC motor.

Finally, before you order an AC motor, you should determine the mounting type you require, the startup torque, the type of enclosure required, and the type of shaft output required. There are many choices in each of these categories. Hopefully, you just need to replace an existing AC motor that has failed and the salesperson can help you find a direct replacement. Otherwise, specifying the correct AC motor can be a daunting task.

- DC motors typically consist of a rotating armature coil inside of a stationary magnetic field which is generated by either a permanent magnet or a stationary electromagnet connected in series or parallel with the armature coil (the series connection often being referred to as a Universal Motor). The fact that these motors can be driven by DC voltages and currents makes them very attractive for low cost applications. To convert the armature current from DC into AC (which is required for rotation), a mechanical solution consisting of brushes and a commutator is employed. However, the arcing produced by the armature coils on the brush-commutator surface generates heat, wear, and EMI, and represents the most significant drawback of this motor type.

- A stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical running. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. The motors rotation has several direct relationships to these applied input pulses. The sequence of the applied pulses is directly related to the direction of motor shafts rotation. The speed of the
motor shafts rotation is directly related to the frequency of the input pulses and the length of rotation is directly related to the number of input pulses applied.

Stepper Motor Advantages and Disadvantages
Advantages
1. The rotation angle of the motor is proportional to the input pulse.
2. The motor has full torque at standstill (if the windings are energized)
3. Precise positioning and repeatability of movement since good stepper motors have an accuracy of 3 – 5% of a step and this error is non cumulative from one step to the next.
4. Excellent response to starting/ stopping/reversing.
5. Very reliable since there are no contact brushes in the motor. Therefore the life of the motor is simply dependant on the life of the bearing.
6. The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.
7. It is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft.
8. A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses.

Disadvantages
1. Resonances can occur if not properly controlled.
2. Not easy to operate at extremely high speeds.

2. Getting Started

By connecting Motor Controller Interface to USB PC port, the HID device driver will install automatically. In other word, Windows will automatically recognize it and configure it's driver accordingly. Windows 2000, XP, Vista and Windows 7, support the device.
After install, insert the device CD into CD drive and go to CD root, then copy the root folder in to your hard disk and execute the "File Register" exe file and click on "Register File" button and

select " MSUSBI.ocx" then ok if the file registers successfully. Then execute "HID_Device_Sample.exe".

3. Connecting NANO Motor Interface To The Motor Driver

The NANO Motor Interface has four channels. Each channel has six pins which are used to makes connections to the external motor drivers.

Speed rate for this product is 10 ms in start motor.

This do not need any Power Supplier, rather is supplied from PC USB Port. Designed into the interface that USB voltage first convert to 12V then regulate to 5V for reason, the voltage level do not dependent on USB voltage.

Figure1 illustrates the interface pin out.

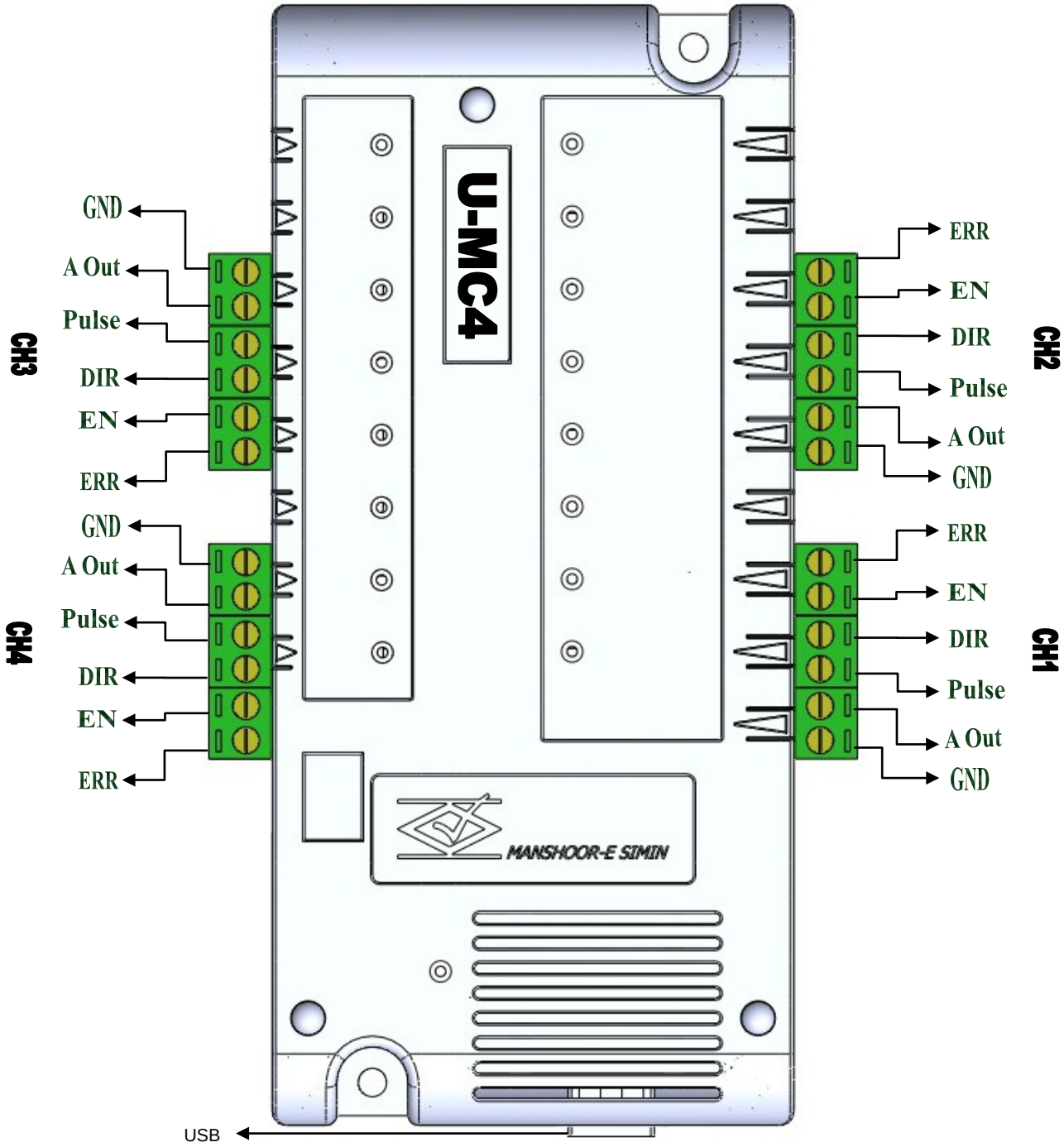


Figure 1

Pin	Description
GND	Grand
A Out	Analog output (10 bit digital for 0-5 Volte Regulated) for DC or AC motor or Full/Half for Stepper motor.
Pulse	Clock for motor (8 H to 20 KH)
DIR	Direction
EN	Motor enable or disable
ERR	Input pin for indicating error of motor

4. Software Sample

The Nano Motor Interface is supplied with this ready sample which makes it very easy for the beginner to get quickly up and run with motor control.

Installation, as described above, is painless and easy requiring only a Windows 2000, XP, Vista or Windows 7 computer with fairly modest specifications. To run the software double click on the " HID_Device_Sample.exe". The "HID_Device_Sample" environment screen will then appear providing the workspace for doing operation. After running the software, last interface selects automatically, that indicated in "Select Interface" combo box. User can select interface from combo box or insert the device index and then click on "Select Device" button.
By click on "Device Name" button, after insert index of the device, the interface name show in the "Name" edit box.
To show number of interfaces that connect to USB port, click on "Device Count".
When the interface would be connect or disconnect on USB port, click on "Update Device" button to refresh list of interfaces. (figure2)

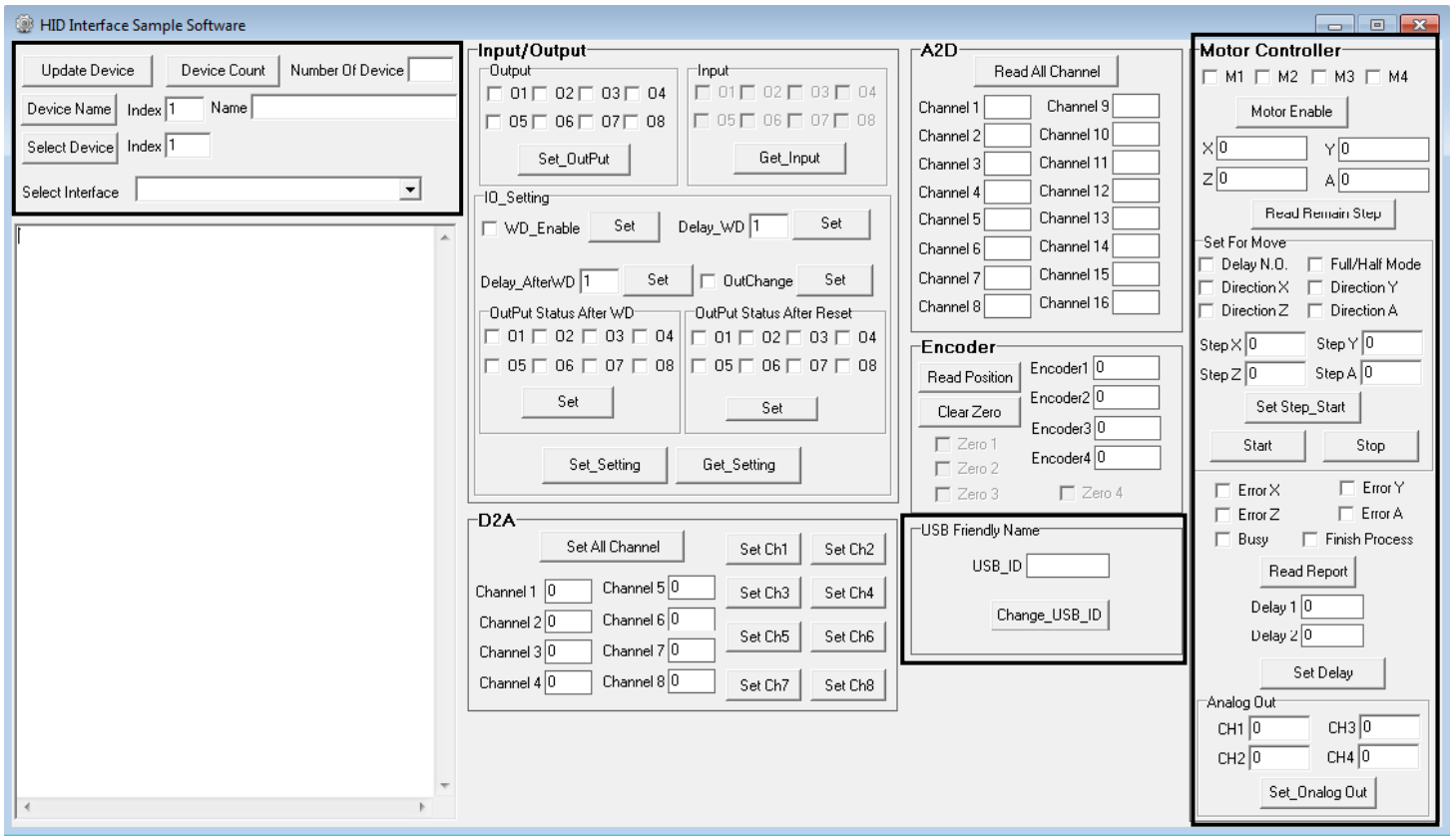


Figure 2

5. Motor Controller in Software

5.1. Motor Enable/Disable

To enable or Disable motors, have to checked or unchecked, "M1","M2" ,"M3" and "M4" check box then click on "Motor Enable" Button.

5.2. Read Remain Step

To check how many step are remained after running the motor click on "Read Remain Step"button.

(Remain step) – (number of step that set for move) = Pass Step.

5.3. Set for Move

5.3.1 Delay N.O.

Setting value on "Delay1" and "Delay2" edit box one may chose the needed rapid rate in "Delay N.O." check box.

5.3.2 Full/Half Mode

To control, the type of step (half or full).

5.3.3 Direction

To change, the direction of motor running.

5.3.4 Start/Stop

Setting value on "StepX","StepY","StepZ","StepA" in edit box, by a click on "Set Step_Start" button, all 4 motors move synchronically.

To stop motors click on "Stop" button and to continue remain running steps click on "Start" button.

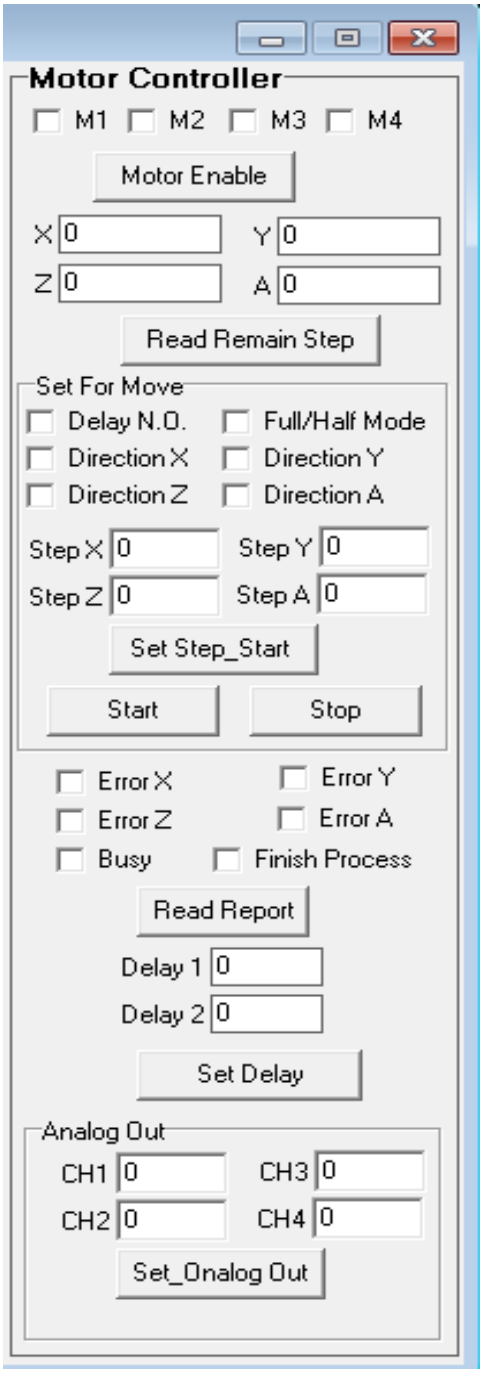


Figure 3

6. Report

"Read Report" button indicates the occurred problems in "ErrorX", "ErrorY", "ErrorZ" and "ErrorA check boxes.

"busy" shows that motor is running and "Finish Process" shows if process is done.

7. Delay

2 delays set for motor speed, with (Rapid and normal speed), to set this 2 delays by setting "Delay1" and "Delay2" edit box a click on "Set delay" button.

8. Analog Output

To set the values of any of 4 channels.

0-1023 digital value for 0-5 volte analog output.

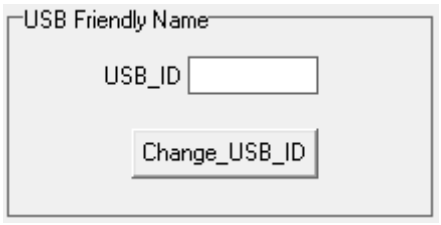


Figure 4

9. Change USB ID(Figure 4)

By connecting more than, one similar interface type into USB port, all of these interfaces have only one friendly name in the list, for distinct of these similar names, user can change ID of the interface, default ID for these interfaces is 0000. User can change these four characters. For example if USB friendly name in the device list is "M.S.D/A 0000", user can change "0000" to any character like "Test", after this change, the interface must be disconnect and then connect to the USB port, by connecting , USB friendly name in the device list change to "M.S.D/A Test". With this change user can distinct between two similar interfaces.

10. Writing your own software for NANO Motor Controller Interface

Provided with this interface an ocx(Active X) file, called "MSUSBI.ocx". This file encapsulates the functions used by sample software in communicating with motor controller interface across the USB interface into any simple functions easily understood and used in custom software. Although the ocx is written in "Delphi" it can be used (called) by programs written in a number of popular languages, the popular of which are Visual BASIC, C#, Visual C. Below described are the techniques to use the ocx in Delphi and Visual Basic. If you program in another language please refer to your compiler manual on the details of importing and calling a ocx functions, which will be very similar to the techniques described below.

10.1. Using NANO Motor Controller Interface by Delphi.

Using NANO Motor Controller Interface with your own programs written in Delphi is very simple. At the head of your program, before using any of the NANO Motor Controller Interface function, you must Import "MSUSBI.ocx" in your programming language and insert the ocx in your program form....

//If Connect or disconnect a interface to USB port call this function//

MSUSBDevice1.DeviceUpdate();

//

//Check how many device is connect to USB port//

integer := MSUSBDevice1.DeviceCount();

//

// Get friendly name of USB interface//

string := MSUSBDevice1.ListOfDevice(Device index:integer);

//

//Select the interface via Device number before doing any work//

```
if not MSUSBDevice1.SelectDevice(Device NO:integer) then
  ShowMessage('Device Could not Select');
////////////////////////////////////////////////////////////
```

//Enable or disable step motor in 4 channel //

```
if not MSUSBDevice1.Motor_Enable(Motor1, Motor2, Motor3 , Motor4:Boolean) then
        Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

//Read remain step after motor(s) start running //

```
if MSUSBDevice1.Motor_Read_Pass_Step(X, Y, Z, A:integer) then
 begin
  Remain_X := X;
  Remain_Y := Y;
  Remain_Z := Z;
  Remain_A := A;
 end
 else
  Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

//Set data for start stepper motor and start running//

```
if not MSUSBDevice1.Motor_Set_Start(Delay NO, Direction X, Direction Y, Direction Z
                    ,DirectionA:Booean, Step NO X, Step NO Y, Step NO Z, Step NO A then
   Memo1.Lines.Add('Failed To Send2.');
////////////////////////////////////////////////////////////
```

//continue stepper motor after stop//

```
if not MSUSBDevice1.Motor_Start then
  Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

//Stop stepper motor //

```
if not MSUSBDevice1.Motor_Stop then
  Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

//Read report //

```
if      MSUSBDevice1.Motor_ReadReport(Error_X,      Error_Y,      Error_Z,      Error_A,
End_Of_Process, Busy:boolean) then
begin
  Error_From Driver_X := Error_X;
  Error_From Driver_Y := Error_Y;
  Error_From Driver_Z := Error_Z;
  Error_From Driver_A := Error_A;
  Finish_Motor_Running_Process := End_Of_Process;
  Motor_Is_Running := Busy;
 end
 else
```

```
  Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

//Set 2 delay mode for motor running speed//

```
if not  MSUSBDevice1.Motor_Set2Delay(Speed_Mode1, Speed_Mode2:Integer) then
 Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

//Set 4 channel 10 bit digital for 0-5 volte output analog//

```
if not MSUSBDevice1.Motor_Analog_Out(CH1, CH2, CH3, CH4:Integer) then
   Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

//Full Or half stepping mode for stepper motor //

```
if not MSUSBDevice1. Motor_HalfStep(X,Y,Z,A: Boolean) then
   Memo1.Lines.Add('Failed To Send.');
```

// Change_USB_ID //

```
if not MSUSBDevice1.Change_USB_ID(USB_ID:string[4])then
 Memo1.Lines.Add('Failed To Send.');
////////////////////////////////////////////////////////////
```

11. Electronic Schema