

HIWIN MIKROSYSTEM CORP.

LINEAR SERVO PLANAR MOTOR (LMSP) USER'S MANUAL

Version 1.0.2
Feb. 6, 2002

Feb. 6, 2002

TABLE OF CONTENTS

I. GENERAL INFORMATION.....	1
II. SYSTEM DESCRIPTION.....	2
Product Features.....	3
System	3
Controller Box	3
Motor	3
System Specifications	6
Reference Numbers	7
LMSP Dimensions	8
Controller Box Dimensions and Connections	9
DSP C32 Card.....	10
Interface of DSP C32 Card	11
UC48 Card.....	12
Interface of UC48 Card	13
IDP4 Card.....	14
Interface of IDP4 Card	15
MB5 Board	16
PU2 Card.....	17
PAT Board	18
Power Board.....	18
Motor	19
HS4 Board.....	19
III. SYSTEM SETUP.....	20
Before You Start	20
Hardware Installation	20
Motion Controller	20
Motor	20
Controller Box	21
Cables	21
Connection of Controller Box (LMDX) and User's Controller	22
Pin Assignment of LMDX IDP4	22
IDP4 and Controller	23
IDP4 and PLC	24
Connection of Motor and Driver	25
HS4 and UC48	25
Driver to Motor Cable	25
IV. SYSTEM STARTUP	26

Communication Settings	26
Command Usage	26
Simple Examples	27
V. SYSTEM OPERATION	29
Stand-alone Mode	29
Calibration.....	38
VI. PROGRAM DOWNLOAD FROM PC TO DSP	39
VII. HELP.....	46
APPENDIX A PULSE COMMAND FORMAT.....	47

I. GENERAL INFORMATION

Linear Servo Planar Motor, abbreviated as LMSP hereinafter, is a planar motor with built-in position sensors. With *Hiwin Mikrosystem Corp.*'s advanced technology, great accuracy has been achieved before it hits the market. And *Hiwin*'s engineers keeps improving its unparalleled performance. Unlike the traditional planar stepping motors, *Hiwin*'s LMSP is a servo motor which has thrust force larger than the stepping ones at high speed. Besides, it has a lot of attractive features such as yaw control, automatic alignment, automatic homing, stall detectability, error compensation ability, and so on.

In the following sections, we'll give you the idea of how to implement with our LMSP. And the manual will be updated whenever its performance is made even better.

This manual consists of the following sections: Section I has general information of LMSP. Section II gives the description of the LMSP system, features and specifications. Section III reminds users of things to be cautious and describes the setup procedures. Section IV shows how to make communication settings and gives some examples. Section V explains how to use available commands to operate the motor. Section VI gives the users the idea about how to upgrade the software version and calibration procedures. Section VII tells *HIWIN* information for technical support. Appendix A shows the STEP/DIR pulse format to be used.

Should you have any suggestions, please contact us using the information in Section VII.

II. SYSTEM DESCRIPTION

A typical LMSP system consists of the following components: an LMSP (a forcer and a stator), a controller box (PID controller and drivers inside), a host controller (usually a computer installed with a motion control card), and required accessories.

These accessories include

- driver power cable <1>
- motor cable <2>
- sensor cable <3>
- driver-end controller cable <4>
- terminal block <5>
- host-end controller cable <6>
- RS-232 cable <7>
- air filter and regulator (user supplied)
- air pipe (user supplied)
- auxiliary 24V power supply (user supplied) for motion control card LMKPCI4P <8>

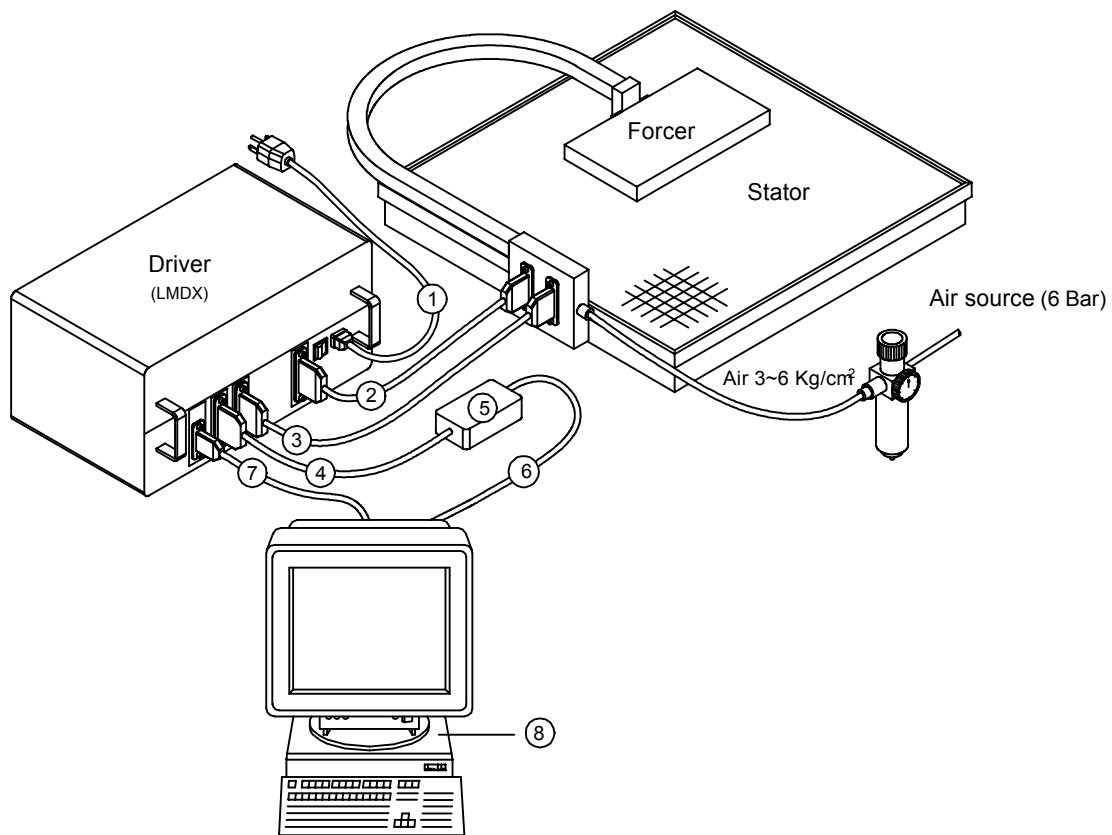


Figure 2.1 System Configuration

Product Features

System

- Closed-loop control (position sensors inside)
- Non-stall
- Auto-alignment
- Sensorless auto-homing

Controller Box

- 2 operation modes: Pulse command mode and stand-alone mode
- Simple motion program without any motion controller
- Linear and yaw error compensation table
- 3-axis control (yaw control included)

Motor

- Hybrid type planar stepping motor
- Precision position sensors
- Air bearing

For the LMSP system, the communication between each component and the power to LMDX controller box are shown in figure 2.2. The system consists of 3 parts: an LMSP motor, an LMSP controller box, and a host controller.

An LMSP motor has one planar stator with tooth structure, and one planar forcer with position sensors that discern LMSP from the commonly seen planar stepping motors. (Multiple forcers may be used for overlapped motion path; however, more drivers shall be needed accordingly.) There are 4 electromagnetic modules in an LMSP forcer where 2 modules for one direction (X1 and X2), and 2 for the other direction (Y1 and Y2). Each module is driven by 2-phase currents, i.e., phase A and phase B, to produce a reluctance force to move the forcer to an equilibrium point. 3 precision position sensors are used to inform the driver the updated position, 1 in updating direction Y, and 2 in updating direction X1 and X2. The 2 sensors for X1 and X2 are arranged symmetrically to the center, so the yaw error can be detected in addition to the X direction position. The encoder in each axis sends differential signals in sin and cos. The signals are amplified to peak to peak magnitude of 1 volt in HS4 board inside the forcer, then transmitted to UC48 board in the LMSP controller box.

An LMSP controller box has 1 power board, 1 power supply of +5V, 4 PU2 current amplifiers, 1 UC48, 1 motherboard MB5, 1 PAT board, 1 IDP4 board, and 1 DSP C32 board. The

power board provides ± 36 V to the current amplifiers. The power supply provides +5V and ± 12 V for MB5, PAT, UC48, IDP4, and DSP C32 board. As mentioned in the previous paragraph, PU2's provide power to drive the motor. PU2-1 amplifies current to drive axis X1, PU2-2 to drive axis X2, PU2-3 to drive axis Y1, and PU2-4 to drive axis Y2. The servo board UC48 takes in analog signals from HS4 outputs, and sends commands to PU2. IDP4 may be used to receive commands from host controller or PLC (programmable logic circuit) while DSP C32 receives commands from PC through RS-232.

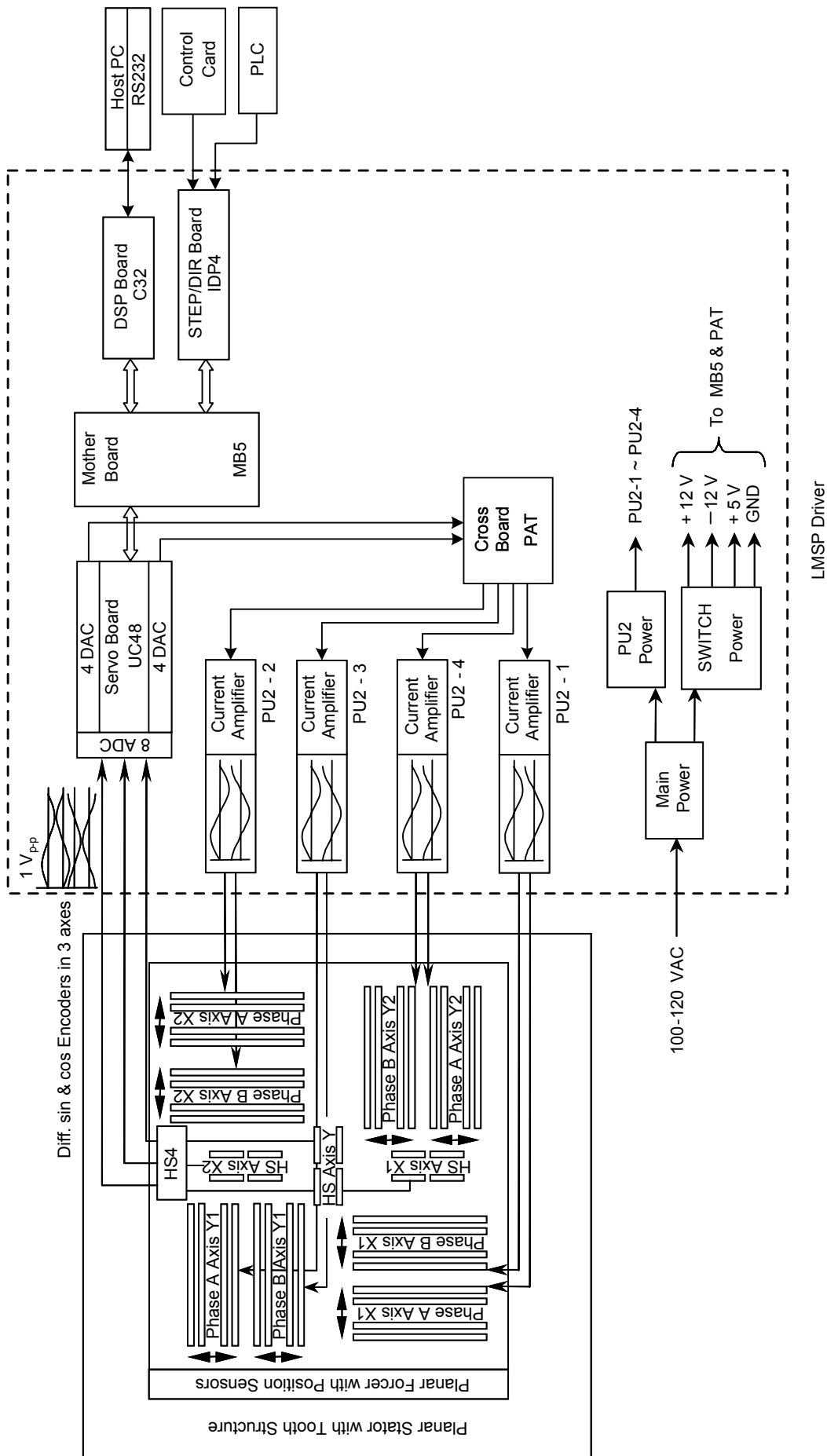


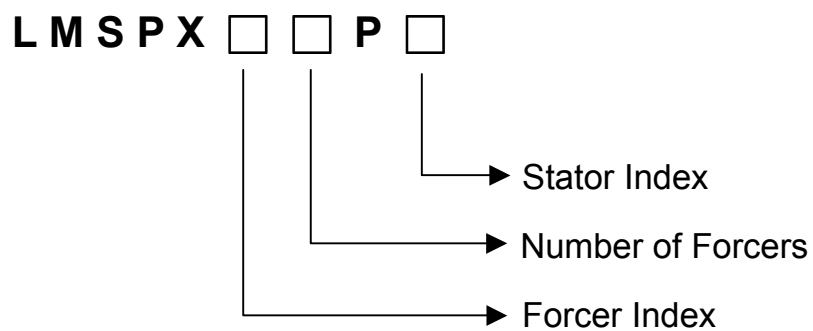
Figure 2.2 System Diagram

System Specifications

Item			Specifications	
			LMSPX1	LMSPX2
Design	Motor		Forcer: 3-axis (X-Y-θ) hybrid planar motor Stator: integral grid platen	
	Support		Air bearing	
	Position sensing		Direct feedback through position sensors in 3 axes (X1/X2/Y)	
	Control	Position	Direct feedback servo control	
		Commutation control	Amplitude/phase control using 2-phase sine wave linear amplifier	
Motor	Max. force		75N	140N
	Max. speed		0.9m/s	0.8m/s
	Forcer weight		1.8kg	3.7kg
	Max. load		14kg	28kg
	Air flow rate		6.4 liter/min	21 liter/min
	Air supply pressure for bearing		3±0.3bar	
	XY stage accuracy		±15μm; after compensation mapping: ±5μm	
	Position sensor resolution	X, Y	1μm	
		θ	0.8 arc-second	
	Positioning resolution	X, Y	1μm	
	Repeatability	X, Y	3μm	
Controller Box	Interface		Parameter setting: RS-232 Status signal: RDY Pulse format: STEP/DIR	
	Power supply		100 to 120V AC ±10%, single-phase 200 to 240V AC ±10%, single-phase	

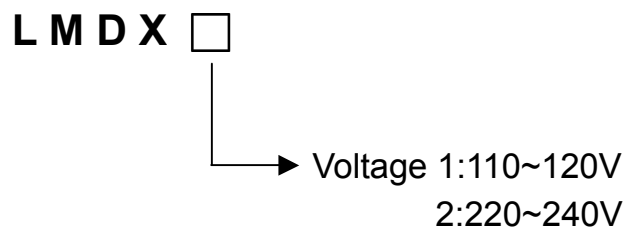
Reference Numbers

Motor



See LMSP Dimensions for detailed information of stator index and forcer index.

Controller Box



Cables : LMACP20B (110V LMDX Power Cable, NEMA 5-15P Plug)

LMACP20F (220V LMDX Power Cable, CEE 7 Standard)

LMACD20D (Motor Cable)

LMACE20C (Encoder Cable)

LMACK20H (Driver-end Controller Cable)

LMSP Dimensions

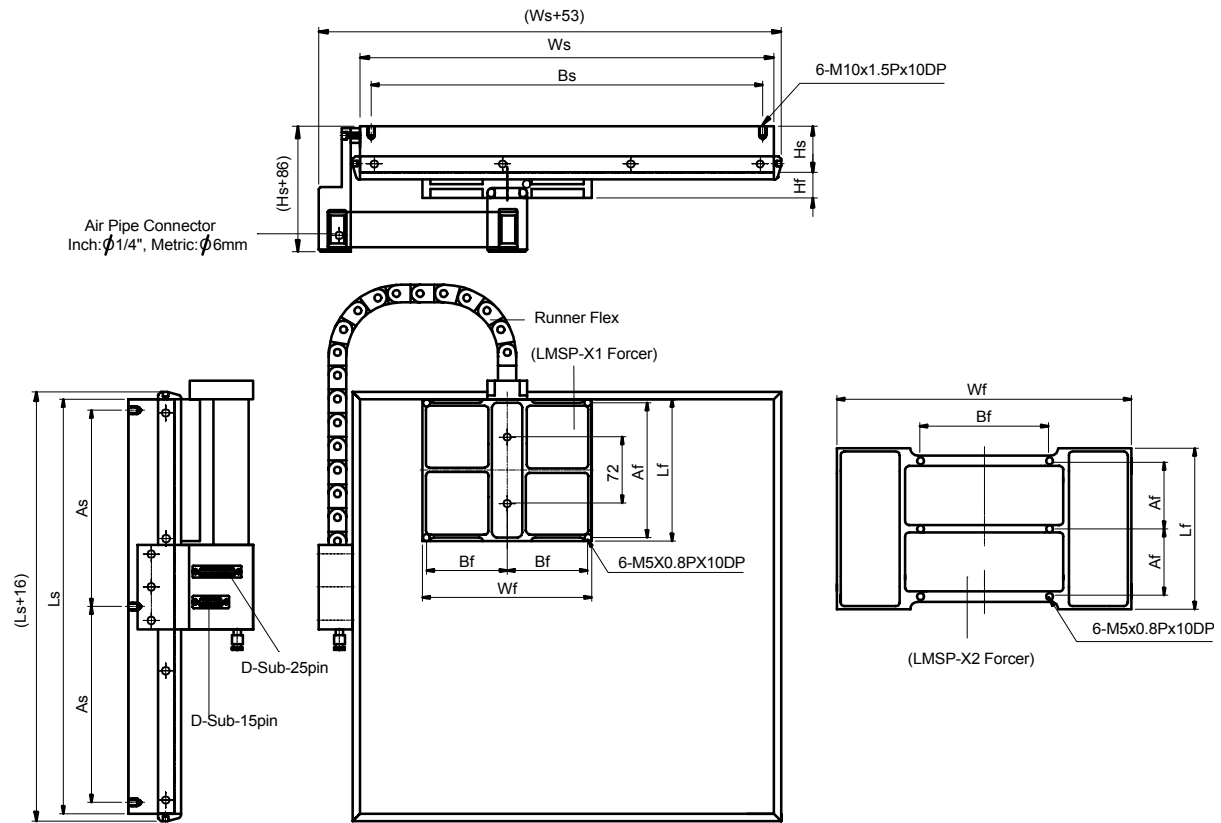


Figure 2.3 LMSP Dimensions

Forcer

Forcer Index		X1	X2
Forcer Length	L_f (mm)	154	175
Forcer Width	W_f (mm)	184	320
Forcer Height	H_f (mm)	28	30
Fix. Distance	A_f (mm) \times B_f (mm)	146×87.5	72×140

Stator

Stator Index		P1	P2	P3	P4	P5	P6
Stator Size	L_s (mm) \times W_s (mm)	350×330	450×450	650×450	600×600	1000×600	850×850
Max. Stroke (mm) (One Forcer)	LMSPX1	196×146	296×266	496×266	446×416	846×416	696×666
	LMSPX2	175×10	275×130	475×130	425×280	825×280	675×530
Stator Height	H_s (mm)	50	50	70	70	100	80
Stator Mass	M_s (kg)	27	36	52	66	120	180
Fix. Distance	A_s (mm) \times B_s (mm)	165×310	213×426	288×426	288×576	$(318-324-318) \times (280-280)$	400×400

Controller Box (LMDX) Dimensions and Connections

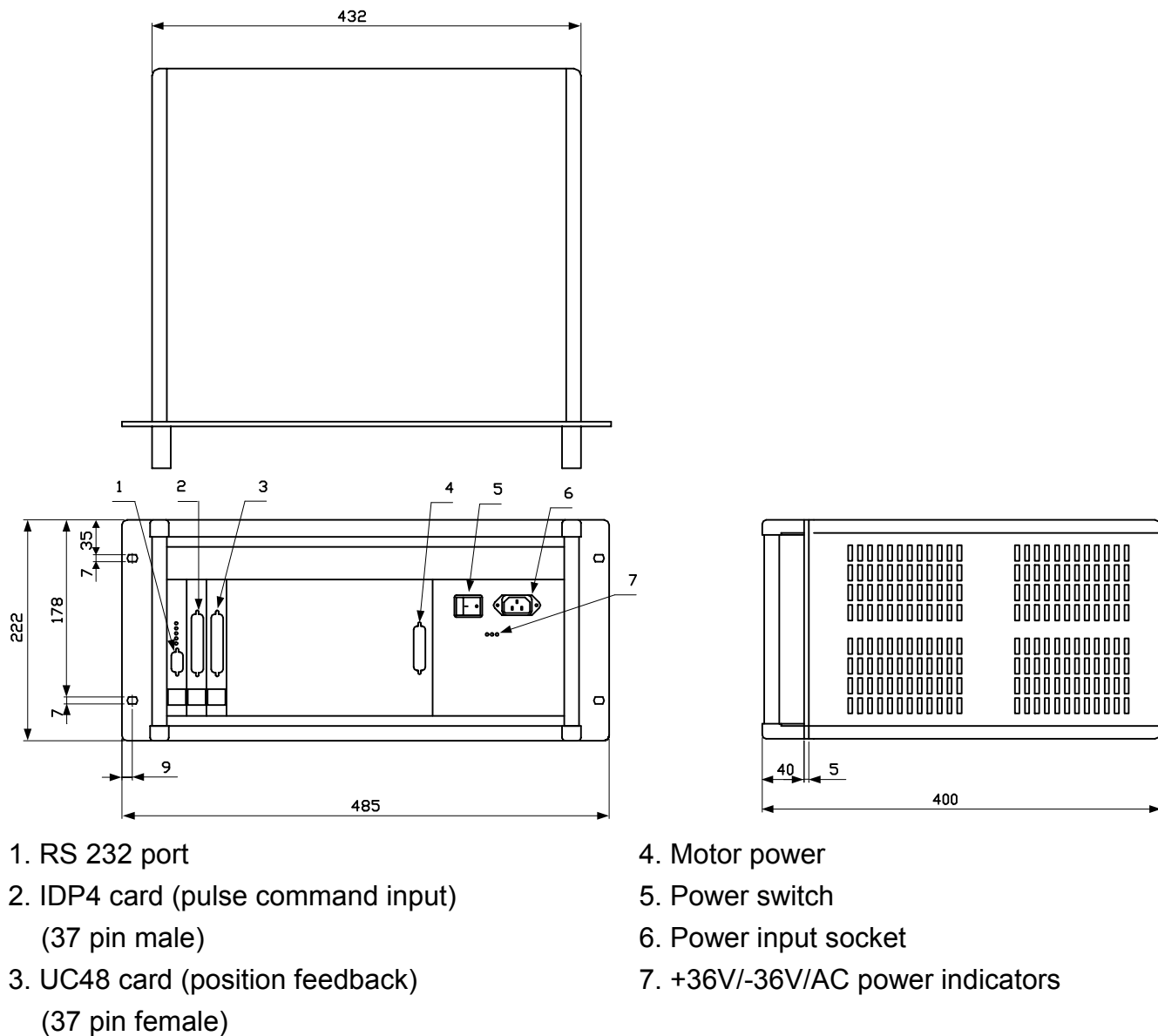


Figure 2.4 LMDX Controller Box Dimensions and Connections

DSP C32 Card

- DSP TMS320C32, 50MHz, 50Mflops, Floating-point commands, 2 timers, 2 DMA, program boot from Flash ROM or Link
- 512K x 16 bit Boot Flash ROM, 256K x 32 bit SRAM, Program easy update from RS-232
- Serial interface RS-232 up to 38400 baud rate, serial synchronous interface RS422 TI 10 M baud rate
- Watchdog Flash ROM-Software for linear and circle interpolation, 3 Axes, DIN/ISO or HPGL

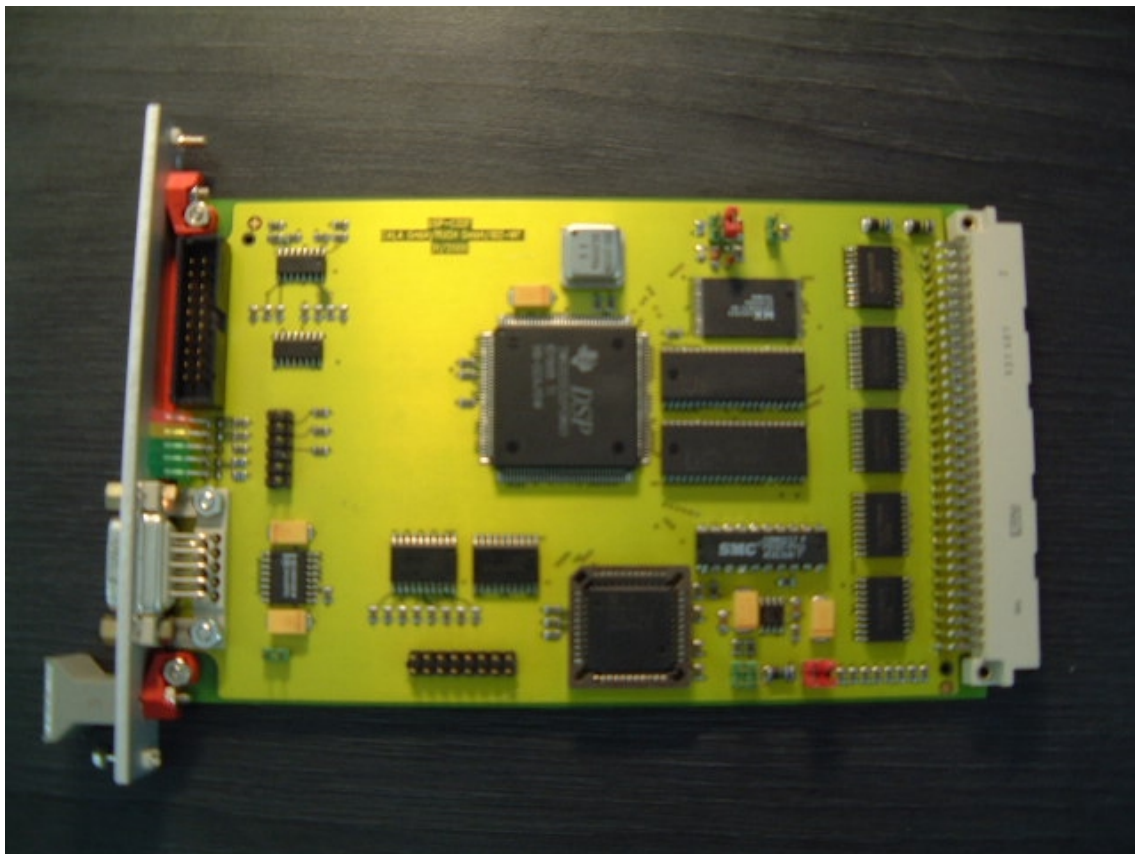


Figure 2.5 DSP C32 Card

Interface of DSP C32 Card

Schematics of inputs and outputs

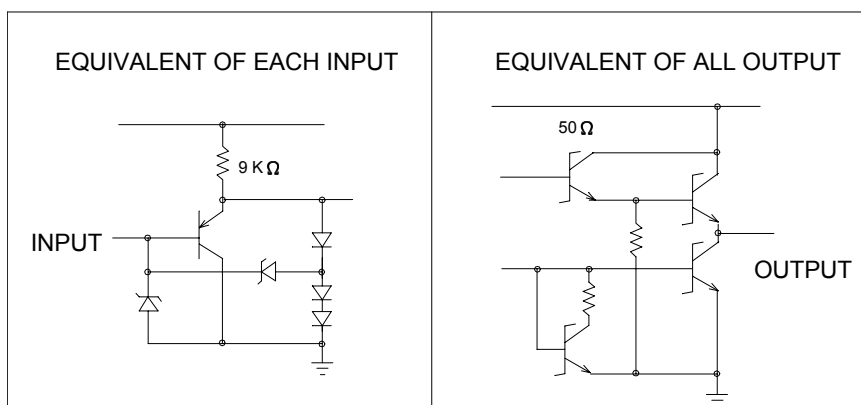


Figure 2.6 DSP C32 Input Interface SN74LS245

Characteristics of DSP C32 Input Interface SN74LS245

Electrical characteristics over recommended operating free-air temperature range (unless otherwise specified)

PARAMETER	MIN	TYPICAL	MAX	UNIT
V_{IH} High-level Input Voltage	2			V
V_{IL} Low-level Input Voltage			0.8	V

All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25\text{ }^{\circ}\text{C}$.

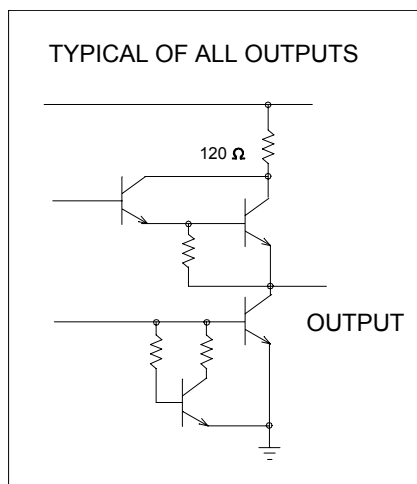


Figure 2.7 DSP C32 Output Interface SN74LS273

Characteristics of DSP C32 Output Interface SN74LS273

Electrical characteristics over recommended operating free-air temperature range (unless otherwise specified)

PARAMETER	MIN	TYPICAL	MAX	UNIT
V_{OH} High-level Output Voltage	2.4	3.4		V
V_{OL} Low-level Output Voltage			0.4	V

All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25\text{ }^{\circ}\text{C}$.

UC48 Card

- Control step and brushless PM motors
- 4 channels for analog inputs sin-cos-ref 11 μ A for 4 encoders
- Software interpolations for encoder resolutions up to 4096 times
- Nanometer resolutions with standard 20 μ m encoders
- 8 inputs analog 12 Bit 10 V bipolar for demand currents or velocities of axes
- 4 opto-coupled inputs 24 V for reference switches
- Complete control up to 4 servo-axis (Portal X1,X2,Y,Z)

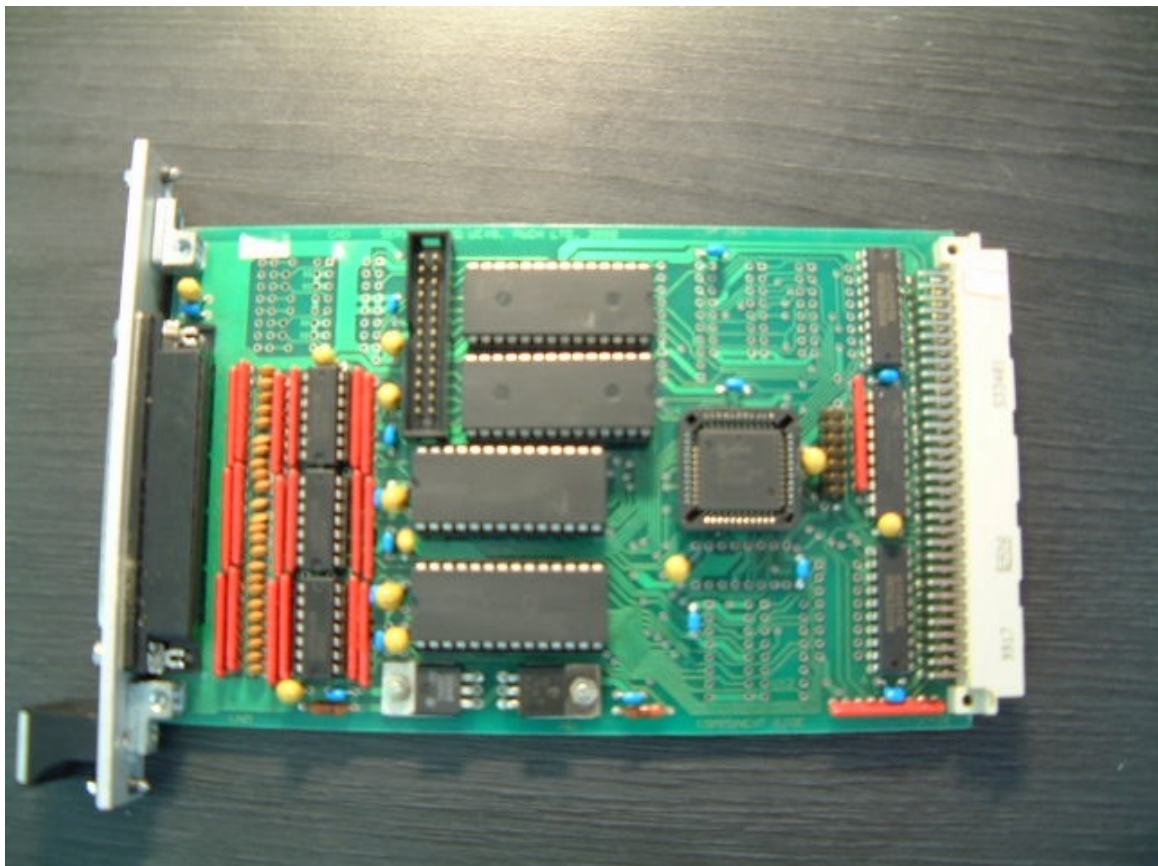


Figure 2.8 UC48

Interface of UC48 Card

The interface of UC48 card is PC847 whose information is shown in figure 2.9 and listed in the bottom table.

PC847

(Unit: mm)

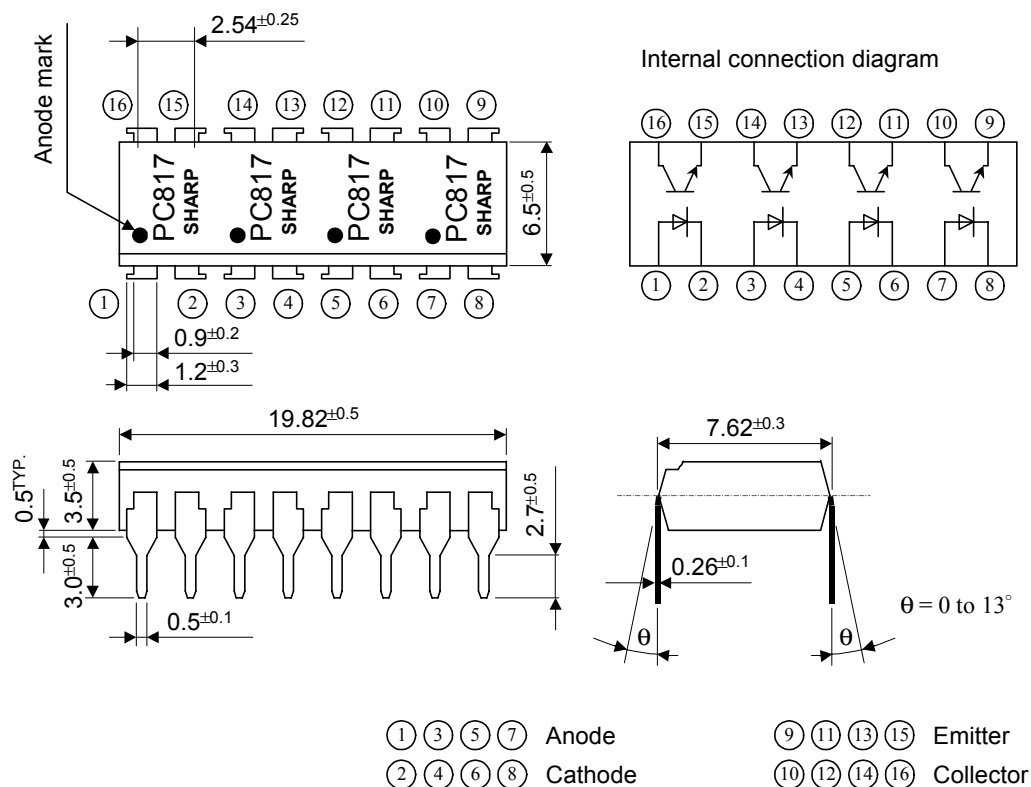


Figure 2.9 UC48 Input Interface PC847

Absolute Maximum Ratings of PC847

(Ta = 25°C)

	Paramter	Symbol	Rating	Unit
Input	Forward current	IF	50	mA
	Peak forward current	IFM	1	A
	Reverse voltage	VR	6	V
	Power dissipation	P	70	mW
Output	Collector-emitter voltage	VCEO	35	V
	Emitter-collector voltage	VECO	6	V
	Collector current	IC	50	mA
	Collector power dissipation	PC	150	mW
Total power dissipation		Ptot	200	mW
Isolation voltage		Viso	5000	Vrms
Operating temperature		Topr	-30 to +100	°C
Storage temperature		Tstg	-55 to +125	°C
Soldering temperature		Tsol	260	°C

*1 Pulse width ≤ 100μs, Duty ratio : 0.001

*2 40 to 60%RH, AC for 1 minute

*3 For 10 seconds

IDP4 Card

- Step / Direction Interface for 4 axis
- Euro-board 160 x 100 mm, Eurobus connection with 64 pin connector
- D-Sub-37 connector for four Step, Dir, Reset and Ready signals
- Complete control up to 4 stepper or servo-axis (Portal X1,X2,Y,Z)
- 4 channels RS422 Interface
- Base address settings per jumper, multiple boards at the same bus possible

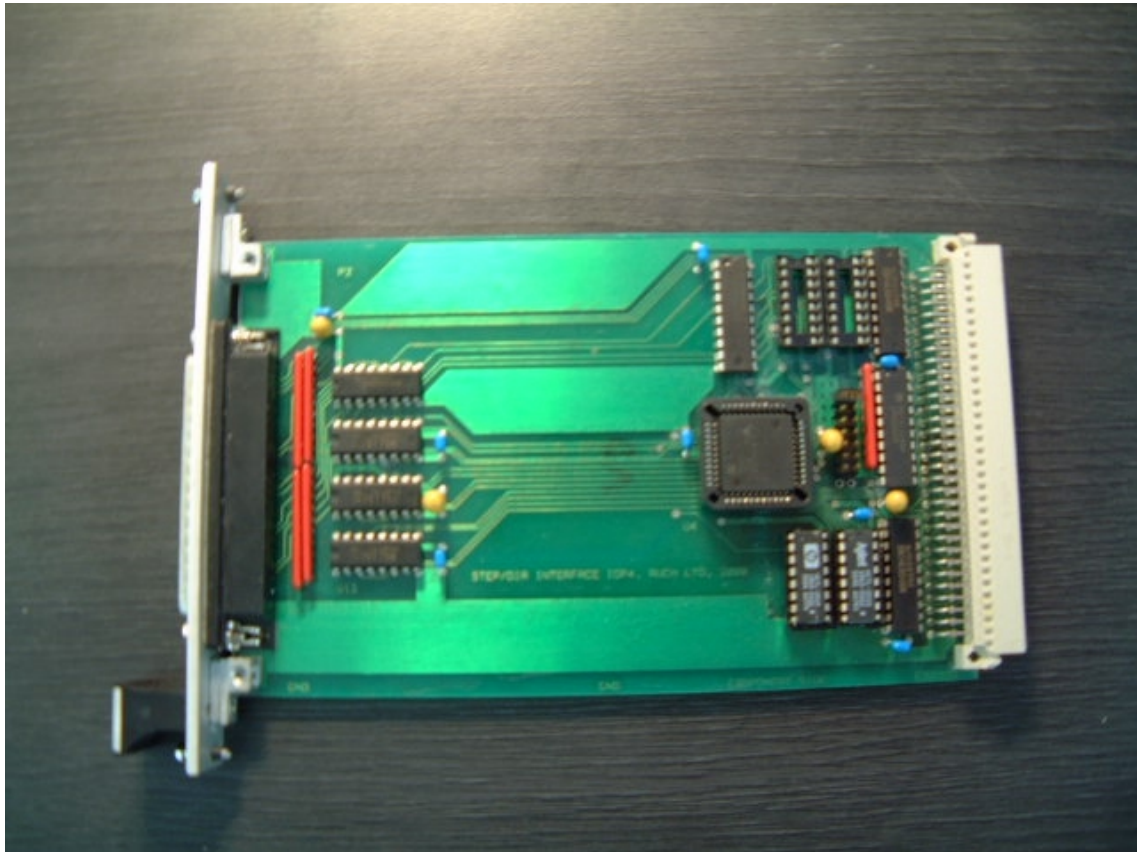


Figure 2.10 IDP4 Card

Interface of IDP4 Card

- Meet or Exceed the Requirements of ANSI Standard EIA/TIA-422-B, RS-423-B, and RS-485
- Meet ITU Recommendations V.10, V.11, X.26, and X.27
- Designed for Multipoint Bus Transmission on Long Bus Lines in Noisy Environments
- 3-State Outputs
- Common-Mode Input Voltage Range –12V to 12V
- Input Sensitivity ... $\pm 200\text{mV}$
- Input Hysteresis ... 50mV Typ
- High Input Impedance ... 12 k Ω Min
- Operate from Single 5V Supply
- Low-Power Requirements
- Plug-in Replacement for MC3486

LOGIC DIAGRAM (POSITIVE LOGIC)

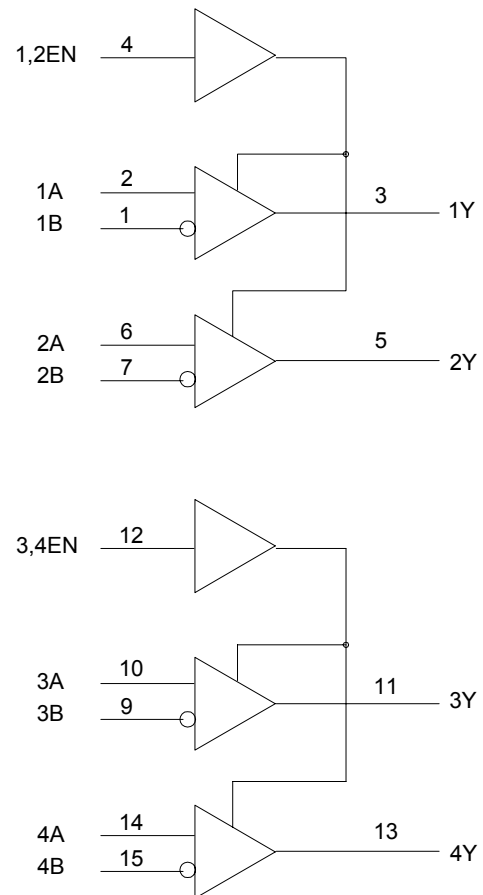


Figure 2.11 IDP4 Interface SN75175

MB5 Board

- Mother Board for 5 Euro-boards 160X100mm, Euribus connection with 64-pin connector
- Standard PC 12 pin power connection for +5V,+12V,-12V and GND power supply
- Resistance networks for address data and control bus to prevent reflections
- Ceramic capacitor for power voltages

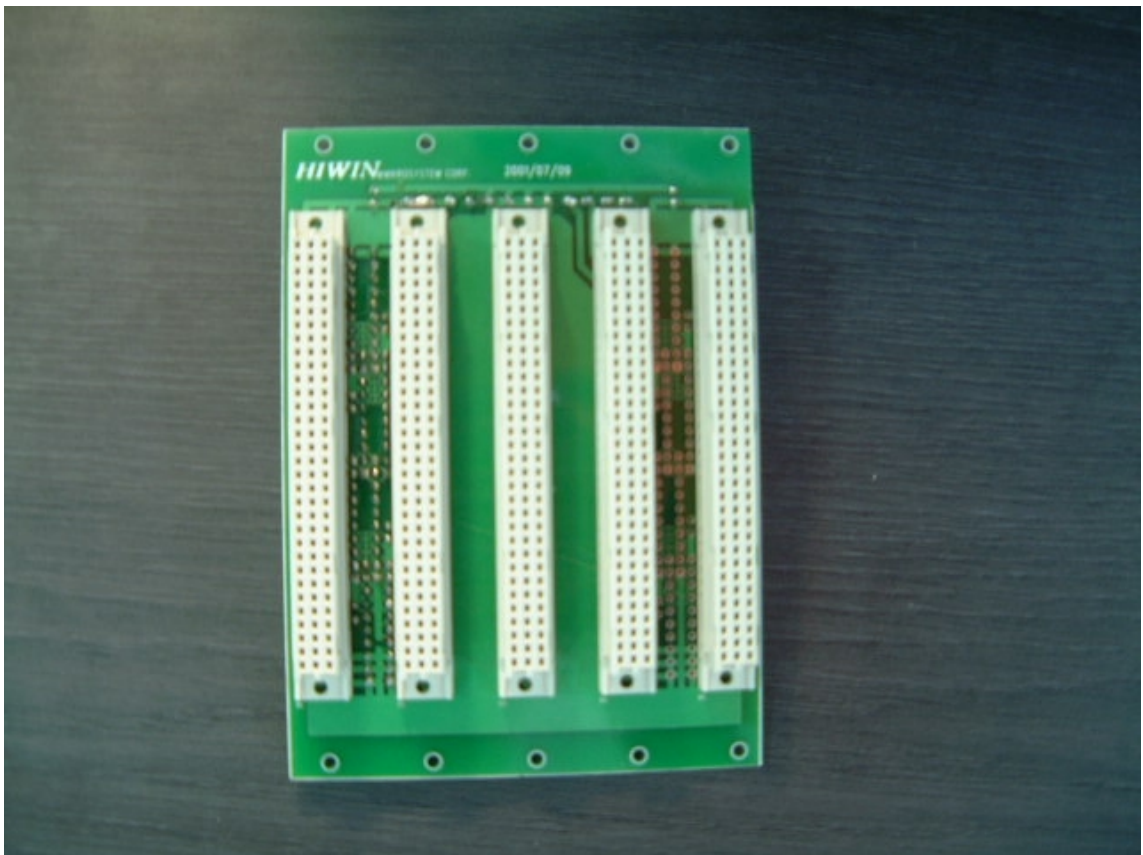


Figure 2.12 MB5 Board

PU2 Card

- 2 current-feedback half-bridge amplifiers 48 pin
- Power supply voltage 35-50 V, output current 3 A
- Two bipolar analog inputs 10V for demand current
- Complete solution for two-phase micro-step motor

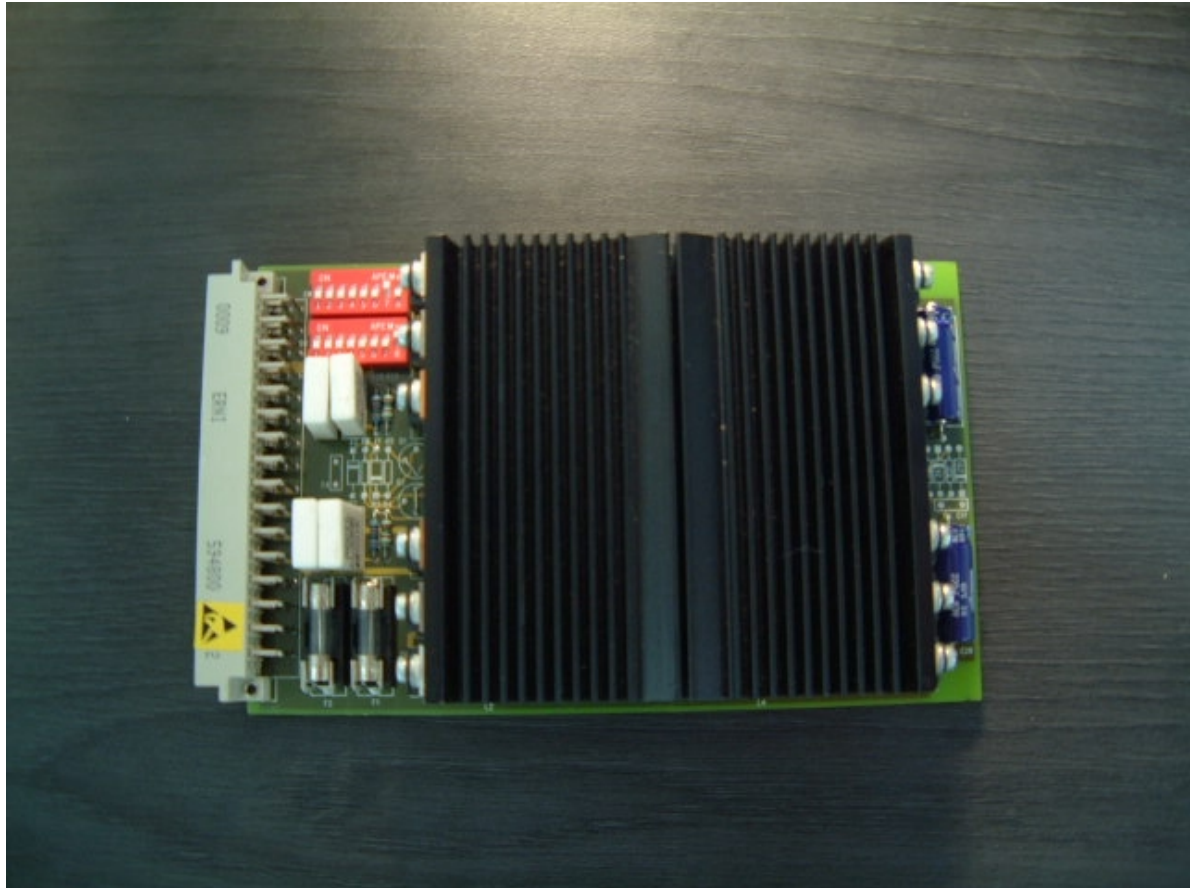


Figure 2.13 PU2 Card

PAT Board

- Euro-boards 160X100mm, Power Bus connection with four 48 pin connector
- 1 connector for power supply
- 2 connectors for 4 two phase stepping motor for 4 power amplifier PU2
- 19 inch standard rack compatible

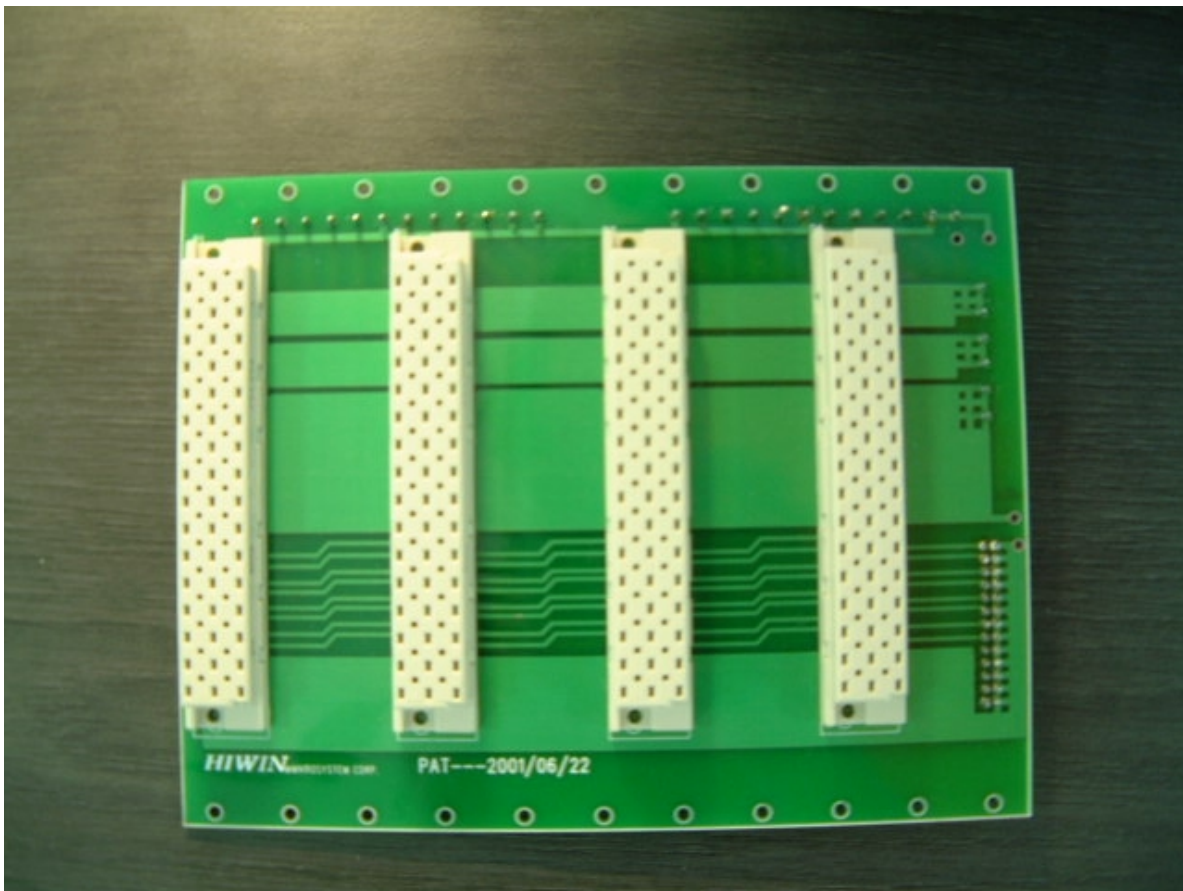


Figure 2.14 PAT Board

Power Board

- AC Power Input with EMI filter
- AC110V and AC220V can be switching change
- Supply +36V and -36V and GND for power amplifier PU2
- Fuse Holder with 10A Fuse

Motor

HS4 Board

- 3 position sensors to detect position feedback
- 2 OPA AD824AR output A & B Phase differential signals
- Power Supply +5V and output $\pm 1V_{p-p}$

III. SYSTEM SETUP

Before You Start

- Please check if there is any damage from transportation.
- Please coat the stator surface with antirust oil or wax for keeps. Remember to wipe off the antirust oil before use next time.
- NEVER disassemble the motor or the driver. *HIWIN* is not responsible for any damage caused therewith.
- Always shut off driver power before making any wiring change.
- Study the manual thoroughly before you start.
- If you have any questions, please contact *HIWIN* or its authorized agents right away.

Hardware Installation

Motion Controller

- Refer to LMKPCI4P manual
- Terminal block, 100-pin cable, and +24V power supply are required for using *HIWIN* LMKPCI4P motion control card.

Motor

- Install the stator.
- Please use high-pressure air and soft textile cloth to thoroughly clean the planar motor's surface.
- Please plug in the air pipe (as shown in Fig. 1) to supply filtered, clean, dry air (pressure around 3.0~6.0 bar dependent on load, and particle size below 3 micron).
- Gently move the forcer to ensure air is normally supplied.
- Align the forcer to the grids of stator to some extent.

Controller Box

- Fix the driver in place, and make correct shielding and ground for power supply, and so on.
- Always shut off the driver before making any change in wiring.

Cables

- Ground the cable to avoid EMC problems.
- Avoid cables cross each other.

Connection of Controller Box (LMDX) and User's Controller

Pin Assignment of LMDX IDP4 (Male)

(Applicable cable LMACK20H: 37 pin D-sub connector)

Description	Signal	Wire Color	Pin	Pin	Wire Color	Signal	Description
			20	1	Blue/ 1 Black dot	+5V	Signal voltage supply
			21	2			
			22	3			
			23	4			
Reset Y -	RYM	Blue/ 2 Red dots	24	5	Blue/ 2 Black dots	RYP	Reset Y +
Reset X -	RXM	Pink/ 1 Red dot	25	6	Pink/ 1 Black dot	RXP	Reset X +
			26	7			
			27	8			
			28	9			
			29	10			
			30	11			
			31	12			
Ready Y -	GYM	Pink/ 2 Red dots	32	13	Pink/ 2 Black dots	GYP	Ready Y +
DIR Y -	DYM	Green/ 1 Red dot	33	14	Green/ 1 Black dot	DYP	DIR Y +
STEP Y -	SYM	Green/ 2 Red dots	34	15	Gray/ 2 Black dots	SYP	STEP Y +
Ready X -	GXM	Yellow/ 1 Red dot	35	16	Yellow/ 1 Black dot	GXP	Ready X +
DIR X -	DXM	Yellow/ 2 Red dots	36	17	Yellow/ 2 Black dots	DXP	DIR X +
STEP X -	SXM	Gray/ 1 Red dot	37	18	Gray/ 1 Black dot	SXP	STEP X +
				19	Blue/ 1 Red dot	GND	Signal ground

Sxx, Dxx, Rxx: Inputs

Gxx: Output

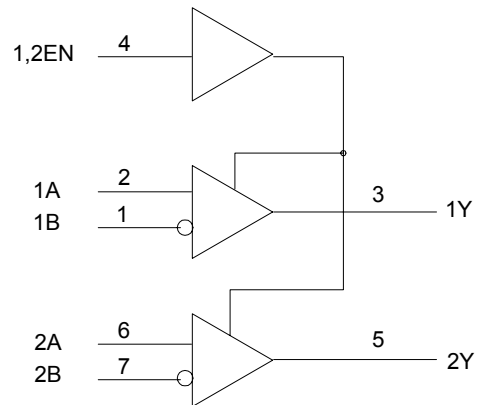
IDP4 and Controller

EIA-422-b Output IC is AM26LS31 or SN75174

LOGIC DIAGRAM (POSITIVE LOGIC)

SXP, SYP, SZP, SWP
SXM, SYM, SZM, SWM

DXP, DYP, DZP, DWP
DXM, DYM, DZM, DWM



GXP, GYP, GZP, GWP
GXM, GYM, GZM, GWM

RXP, RYP, RZP, RWP
RXM, RYM, RZM, RW,

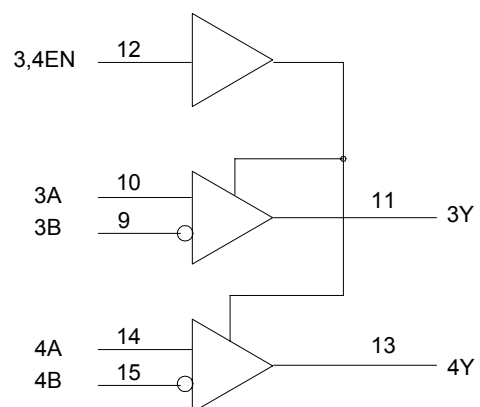


Figure 3.1 IDP4 Interface Connection

IDP4 and PLC

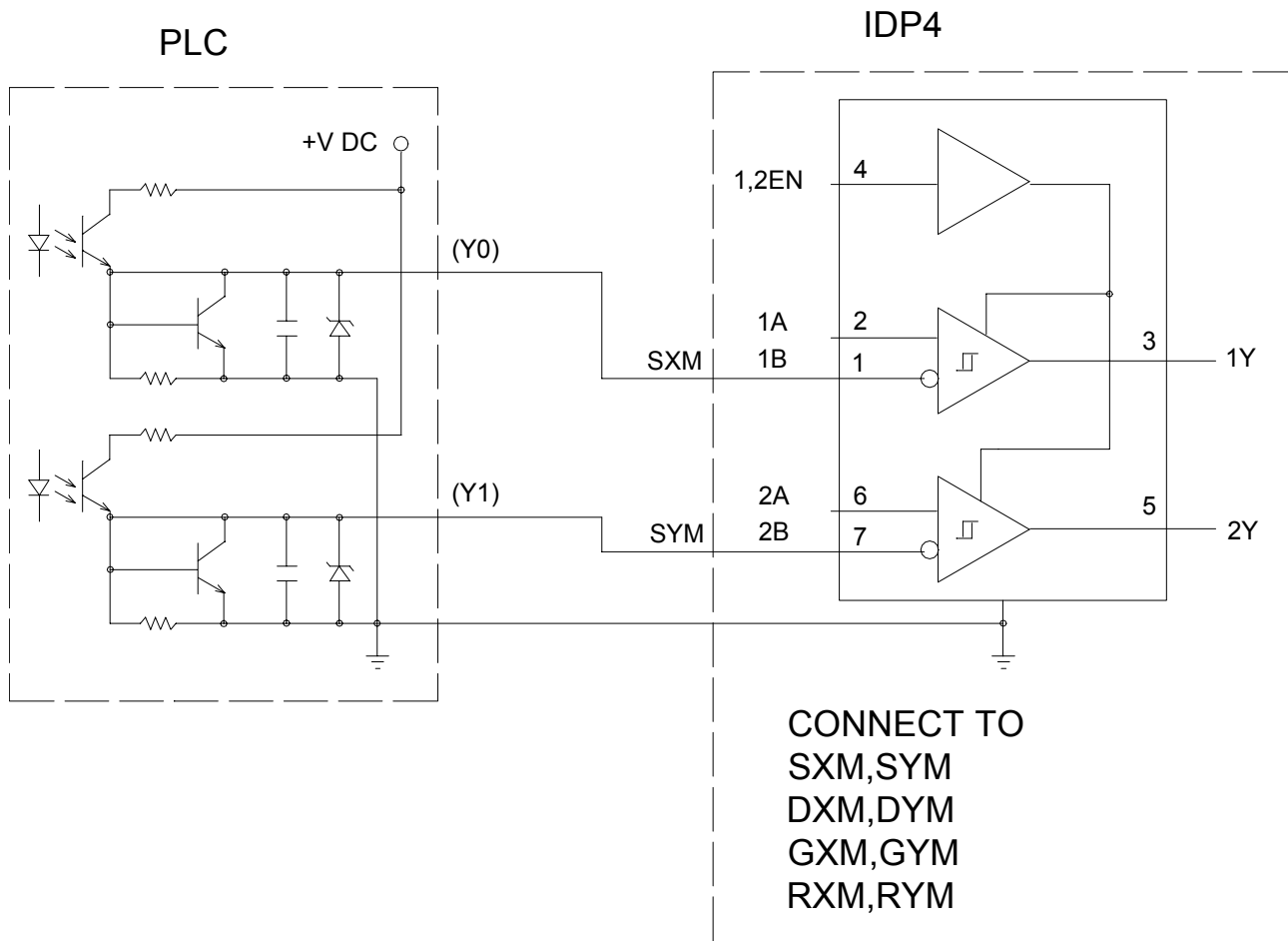


Figure 3.2 IDP4 and PLC Connection

Connection of Motor and Driver HS4 and UC48

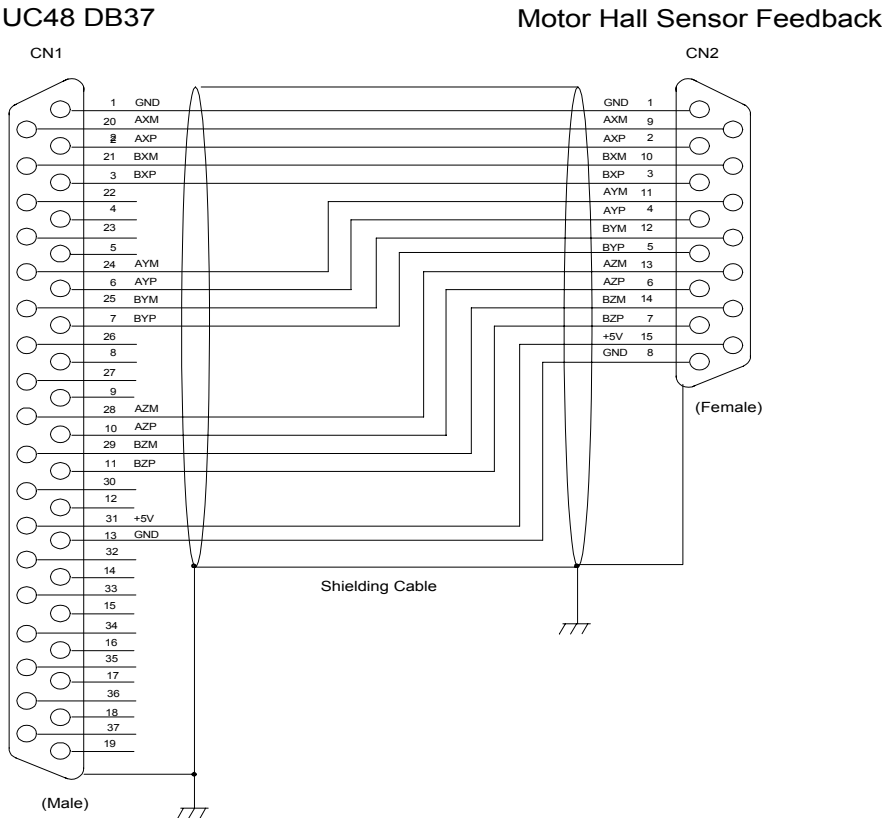


Figure 3.3 HS4 to UC48 Cable

Driver to Motor Cable

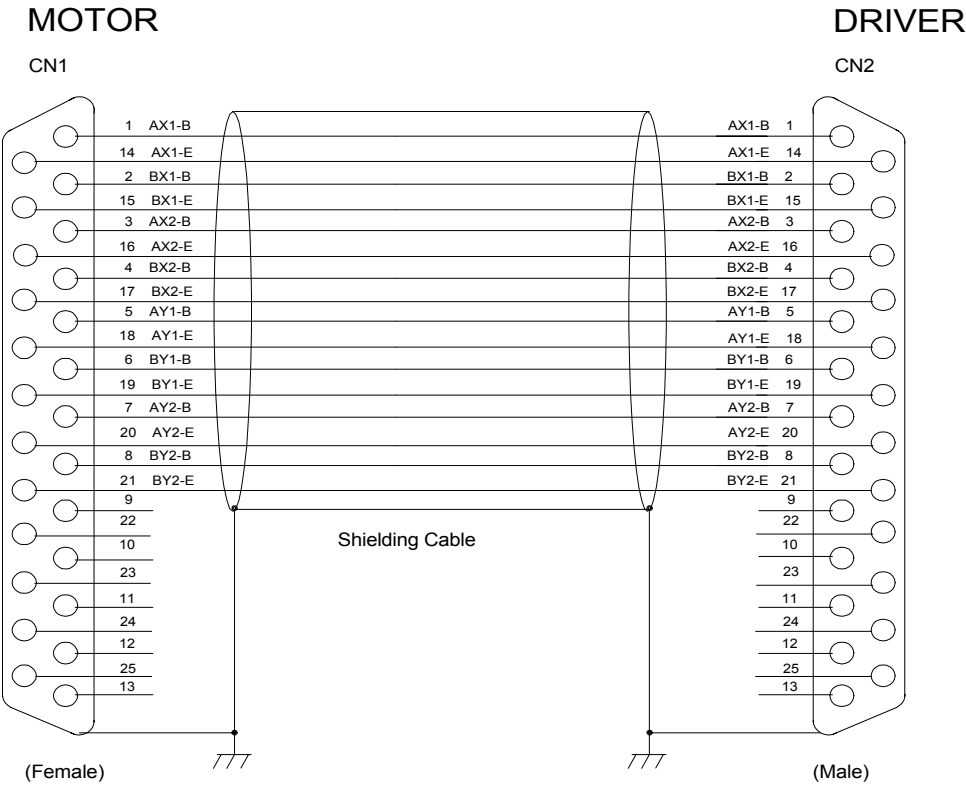


Figure 3.4 Driver to Motor Cable

IV. SYSTEM STARTUP

Wiring as shown in Fig. 2.2 after following the installation in Section III. This section takes you through the necessary procedures to make a simple test in stand-alone mode operation.

Communication Settings

A personal computer communicates with an LMDS driver, in stand-alone mode, through RS-232 to its serial port COM1 or COM2. **Be sure that the driver power is off while connecting RS-232.** Plug in the driver power cable after RS-232 has been connected. Accessory “HyperTerminal” in Windows 95/98/NT/2000 can be used for communication. In addition, a communication program named “et.exe” is included in the shipped-along *HIWIN* CD-ROM, in case “HyperTerminal” is not readily available. However, “et.exe” works only in MS-DOS mode. Run the communication program, either “HyperTerminal” in Windows or “et.exe” in MS-DOS mode. Please make the settings of the corresponding communication port in HyperTerminal as follows:

Baud rate	9600
Data bits	8
Parity check	odd
Stop bits	2

In MS-DOS mode, enter “et 1” if COM1 is used, or “et 2” if COM2 is used instead. The program itself will configure the corresponding port to the aforementioned settings.

The communication program is ready to receive commands whenever the prompt “>” is displayed on the screen. Please see Section VI for explanation of commands in details. Also, brief command explanation can be seen by using help command “?”.

Command Usage

Every command line is issued after “,” or “enter” has been keyed in. Not all the commands have parameters. Those commands with parameters might have more than one parameter. A command line is parsed into the issued command and its parameters by at least one “space”. If one parameter is to be changed, those parameters ahead of it are still needed, while any unchanged parameters after it can be omitted. Therefore, for a command with at least two parameters, it’s unnecessary to key in all the parameters unless the last parameter should be changed. It’s enough to key in the command line up to the parameter you want to change. Note that the function of a “space” is to distinguish a parameter from anything before it. If there is any space in the end of a command line, the driver will mistake that the user is going to set one more parameter’s value. In other words, **ending a command line with a “space” will set any**

parameter immediately after it to zero. Therefore, it's advised that never leave any space in the end of a command line.

Simple Examples

- a) Verify the aforementioned wiring is correct and the communication is normal.
- b) Turn on the driver power. Please wait for a while to see if the forcer aligns to the grids of the stator correctly. If alignment is made successfully, the motor continues to execute automatic homing. After homing, you may go to step (d) to start basic tests. If an error message shows that the motor fails to make alignments, due to large yaw error, proceed with step (c).
- c) Whenever the motor is unable to make alignments, shut off the driver power, then manually adjust the forcer to approximately align with the grids of the stator. Start over with step (b) again.
- d) Test of command "PR". (Movement in relative coordinate)
Issue "PR" to move the forcer +10 mm in X-axis and +20 mm in Y-axis, relative to the current position. (Note the unit of the input arguments is μm .) First, check if there is enough room in the moving direction. Then enter the following command:

PR 10000 20000

- e) Test of command "PA". (Movement in absolute coordinate)
Practice to move the forcer to 0.1 mm in X-axis and 0.1 mm in Y-axis in absolute coordinate by "PA". Enter the following command:

PA 100 100

- f) Test of command "FA". (Setting of velocity and acceleration)
Set the velocity and acceleration of the forcer to 10mm/sec and 1.5m/sec^2 , respectively. Give the following command:

FA 10 1.5

- g) Test of command "GS". (Homing)
To do automatic homing, give the following command:

GS

- h) Practice of simple programming
Example:

```
@
FA 500 2
PA 150000,200000
DL 200      (delay in ms)
FA 200 1
PA 0,0
```

@

In the above example, command "DL 200" make the forcer pause for 200 ms, or 0.2 sec., after previous motion command has been executed. Also, use command "@" to do programming, the program will be written to EEPROM automatically.

After programming, you can always use the command "!" to run the program.

i) Test of pulse mode operation

First, make required settings in stand-alone mode, e.g.,

SD 1 1 50 1 1

where the first parameter 1 switches operation to pulse mode, and the fourth parameter and the fifth parameter 1 set the distance (in μm) for each pulse to travel in axis X and Y, respectively. After setting the required parameters by commands, remember to write them to the EEPROM using the command "WE". Please refer to Appendix I and II to verify the controller's output pulse format is right and the wiring is correct. Be cautious that wiring should be done only when the driver power is off. Then you can use the controller to send STEP/DIR pulse commands to the driver.

V. SYSTEM OPERATION

Stand-alone Mode

Below each command is explained. After setting the required parameters by commands, remember to write them to the EEPROM using the command “WE”.

1. Motion related commands “PA”, “PR”, and “DL”

- a) Translational movement made in an absolute (fixed) coordinate “PA”

Usage:

“PA *xxx yyy*,”

xxx – X-axis new position, in μm . (+/- permitted)

yyy – Y-axis new position, in μm . (+/- permitted)

- b) Translational movement made in a relative (moving) coordinate “PR”

Usage:

“PR *xxx yyy*,”

xxx – X-axis new position relative to the current one, in μm (+/- permitted)

yyy – Y-axis new position relative to the current one, in μm . (+/- permitted)

- c) Pause for a specified period of time “DL”

Usage:

“DL *ttt*,”

ttt – duration in ms

2. Velocity and acceleration command “FA”, velocity scale command “KV”, and stop command “BF”

- a) Velocity and acceleration command “FA”

Usage:

“FA *p1 p2*,”

p1 – velocity (mm/sec)

p2 – acceleration (m/sec^2)

Both parameters should be positive. The motor moves at the velocity and acceleration specified by this command. New values go into effect after they are updated by this command.

- b) Velocity scale command “KV”

Usage:

“KV *p1*,”

p1 – velocity scale; $0. < p1 \leq 4.0$

Velocity of all movements will be multiplied with the latest updated scale value.

- c) Emergency stop and motion buffer memory size command “BF”

Usage:

“BF 0,”

The motor stops immediately, and unexecuted motion commands are cleared.

“BF;”

displays a parameter $p1$, where $p1$ is a decimal number (0 to 31) showing the internal motion buffer memory being occupied. If $p1$ is 0, it means all motion commands have been executed, and the motor stops. If $p1$ is 31, it means all the motion buffer memory have been taken up, so it can not take any new motion command.

3. Coordinate display “DD” and “DR”

a) Display of command position “DD”

Usage:

“DD;”

After executing the command, “ $p1 \ p2$ ” will be displayed.

$p1$ – X-axis command position, in μm

$p2$ – Y-axis command position, in μm

b) Display of actual position “DR”

Usage:

“DR;”

After executing the command, “ $p1 \ p2 \ p3$ ” will be displayed.

$p1$ – X1-axis actual position, in μm

$p2$ – X2-axis actual position, in μm

$p3$ – Y-axis actual position, in μm

4. Error related command “DE”

a) Display of errors in each axis “DE”

Usage:

“DE;”

After executing this command, “ $p1 \ p2 \ p3$ ” are displayed.

$p1$ – rotational axis closed-loop error, in μm .

$p2$ – X-axis closed-loop error, in μm

$p3$ – Y-axis closed-loop error, in μm

“DE 0 0 0;” clears all the errors. Other commands such as “GS;”, “FX1;”, “FX2;” and “RES;” also clear all the errors.

5. Analog to digital conversion (ADC) signals calibration commands “CL”, “CLS”, “LN”, “LNS”, “NP”, and “GS”

a) Calibration of ADC signals “CL”

Usage:

“CL;”

This command calibrates the amplitude and offset of signals from the position sensor.

The signals are collected for calibration while the motor moves 2 mechanical pitches back and forth along axis X and Y.

b) Setting of calibration parameters “CLS”

Usage:

“CLS *p1 p2 p3 p4 p5 p6 p7 p8*;

p1 – calibration velocity, in mm/s

p2 – parameter to decide if ADC calibration of the amplitude and offset is continued during motion. 0 for no; and 1 for yes.

p3 – filter factor to determine the weightings of ADC signals.

p4 – time duration for calibration, in ms.

p5 – minimum distance to travel for every calibration, in μm .

p6 – maximum value of amplitude scale. During each calibration, the measured ADC signals with amplitude greater than $2048/p6$ will be recorded.

p7 – time delay before calibration, in μs

p8 – delay time before current rise, in μs

These settings will be used in executing “CL”. If the travelling distance in some axis is less than the minimum distance (set by parameter *p5*) during *p4* ms calibration, the calibration in that axis is invalid.

c) Alignment with grids “LN”

Usage:

“LN;”

The motor makes alignments with grids of the stator.

d) Update calibrated parameters “LNS”

Usage:

“LNS;”

This command updates the calibrated variables in workspace after using “CL” and “NP”.

e) Define Null angle sub-periods “NP”

Usage:

“NP;”

This command calibrates the null angle sub-periods from the position sensor. The null angle sub-periods are automatically calibrated after the motor moves 1.5 mechanical pitches back and forth along axis X and Y.

f) Homing “GS”

Usage:

“GS;”

The motor moves back home.

6. Power on settings command “RP”

Usage:

“RP *p1 p2*;

p1 – decimal number. If *p1* is set as –1, default system function is used. For *p1* from 1 to 63, it can be decomposed into the sum of the following numbers:

1 – closed-loop mode after power-on

2 – open loop mode after power-on

4 – ADC signals auto-calibration

8 – automatic homing

16 – automatic alignments with grids

32 – automatic execution of motion program written to EEPROM by user.

To enable the function you want, simply assign the sum of the number corresponding to the function to *p1*. For example, to enable the motor to execute automatic homing, automatic alignments with grids, and motion programs, *p1* should be 56 (8 + 16 + 32).

p2 – incremental/decremental ratio of velocity while stored program is executed.

7. Pulse mode related command “SD”

Usage:

“SD *p1 p2 p3 p4 p5 p6*;

p1 – 1: switch to STEP/DIR mode; 0: switch to stand-alone mode.

p4 – the distance traveled in X-axis for each pulse it receives (μm /pulse)

p5 – the distance traveled in Y-axis for each pulse it receives (μm /pulse)

8. Motion programming related commands “@”, “DP”, “I”, and “PS”.

a) Loading user’s motion program to EEPROM “@”

Usage:

“@

(first line in programs)

...

(last line in programs)

@”

The capacity of memory for programs is 11900 bytes.

b) Display of motion programs “DP”

Usage:

“DP;”

Press the “Enter” key to stop display, then press any key to continue displaying next page.

c) Execution of motion programs “!”

Usage:

“!,”

Note: The following characters works during motion:

“+/-“ - increases/decreases velocity

“P” – pauses/continues the execution of motion programs

“Enter” – cancels/stops the execution of motion programs

d) Step by step execution of motion programs

Usage:

“PS *p1*,”

p1 – 1: Step by step execution of motion programs is enabled. (Press any key other than “Enter” to execute the motion program step by step.)

0: Step by step execution of motion programs is disabled. (Default value after power-on.)

9. Display of brief command description “?”

Usage:

“?,”

Press the “Enter” key to stop display, then press any key to continue.

10. PID control related commands “KX”, “KY”, “KF”

Usage:

“Kn *p1 p2 p3 p4 p5*,”

where *n* stands for the axis it acts.

n – “X”: X-axis

“Y”: Y-axis

“F”: rotational axis

p1 – proportional gain

p2 – integration gain

p3 – differential gain

p4 – acceleration factor

p5 – filter factor in the range of 0 to 1.

11. Read from/write to EEPROM commands “RE” and “WE”

a) “RE”: reads all the stored parameters from EEPROM

Usage:

“RE,”

This function is automatically executed after power on if nothing wrong with EEPROM.

b) "WE": writes all the parameters to EEPROM

Usage:

"WE;"

12. Setting of open loop or closed-loop motion "FX"

Usage:

"FX *p1*;"

p1 – 1: closed-loop control. After it is switched to this mode, the motor will automatically make alignment to grids.

2: open loop

13. Display of driver's version "VER"

Usage:

"VER;"

14. Reset the driver "RES"

Usage:

"RES;"

Reset the driver, similar to power-on.

15. Error compensation related commands "TKX", "TKY", "TKR", "TKU", "TKV", "TKA", "TKB", "TKC", "NKX", "NKY", "NKR", "NKU", "NKV", "NKA", "NKB", and "NKC"

a) Loading linear error compensation table "TKX", "TKY", "TKA", and "TKB"

Usage:

"TK*n i M dat0 dat1 dat2 ... datN*;"

where *n* specifies the direction and axis the forcer moves

n – X, X-axis forward

Y, Y-axis forward

A, X-axis backward

B, Y-axis backward

i – error compensation begins with the *i*-th input data, *i* is an integer no less than 0. If *i* = 0, input data begins with *dat0*

M – total number of input data

dat0 ... datN – compensation data table, $N \leq 599$

In the compensation data table, all the data are in μm except the first value *dat0* is in mm, they are explained as follows:

dat0 – the first one is not a compensation value, but used to indicate the distance interval where measurement is made to build the compensation table. For example, if measurement is made every 10 mm, *dat0* is 10.

dat1 – error of the initial point, i.e., interval*0

dat2 – error at interval*1

dat3 – error at interval*2

...

datN – error at interval*(*N* – 1), $N \leq 599$

When input the compensation table, the real measured errors are used, e.g., If the measured position is 29997 μm while it has been commanded to move 3000 μm , then 3 μm is input to build the table. The compensation table works in the way that interpolation is applied for any value in each interval; while movement in the negative coordinate, or negative position in the absolute coordinate, will not be compensated. Two-way compensation can be made to have better accuracy. Note that all the compensated values are integer only and in the range of –127 to 127. These four compensation tables “TKX”, “TKY”, “TKA”, and “TKB” works alone even the others do not work at all.

b) Loading “YAW” error compensation table “TKR” and “TKC”

Usage:

“TK*n i M dat0 dat1 dat2 ... datN*,”

n – R: “YAW” in forward movement

C: “YAW” in backward movement

i - change begins with the *i*-th input data, *i* is an integer no less than 0, if *i* = 0, input data begins with *dat0*

M – total number of input data

dat0 ... datN – compensation data table, $N \leq 599$

In the compensation data table, all the data are in arc sec except *dat0*. They are explained as follows:

dat0 – the error compensation data table works only when *dat0* is 1.

The interval for making the compensation table should be the same as in linear compensation table because they are based on the same forward and backward movement, respectively.

dat1 – error of the initial point, i.e., interval*0, usually 0.

dat2 – error at interval*1

dat3 – error at interval*2

...

datN – error at interval*(*N* – 1), $N \leq 599$

When input the compensation table, the real measured errors are used, e.g., If the measured yaw error is –3 arc sec, then 3 is input to the table. The compensation table works in the way that interpolation is applied for any value in each interval while movement in the negative coordinate, or negative position in the absolute coordinate,

will not be compensated. Two-way compensation can be made to have better accuracy. Note that all the compensated values are integer only and in the range of -127 to 127 .

“TKX”, “TKY”, “TKA”, and “TKB” works even “TKR” and “TKC” do not work. However, “TKR” and “TKC” do not work if “TKX” does not work because “TKX”’s *dat0* is used by “TKR” and “TKC”.

c) Loading the local error compensation table “TKU” and “TKV”

Usage:

“TKn i M *dat0 dat1 dat2 ... datN*,”

n – “U”: forward movement in X-axis

“V”: forward movement in Y-axis

i – change begins with the i-th input data, i is an integer no less than 0, if i = 0, input data begins with *dat0*

M – total number of input data

dat0 ... datN – compensation data table, $N \leq 599$

Different from “TKX”, “TKY”, “TKA”, and “TKB”, the local error compensation table is applicable only to one single pitch. In the compensation data table, all the data are in micron except *dat0*, they are explained as follows:

dat0 – the first one is not a compensation value, but an index, which multiplied by 5 micron is the distance interval between two consecutive measurement points, for the compensation table. There is only one interval value for a compensation table. For example, if measurement is made every 40 micron, *dat0* is 8.

dat1 – error of the initial point, i.e., interval*0

dat2 – error at interval*1

dat3 – error at interval*2

...

datN – error at interval*(N – 1), $N \leq 599$

The way of making a compensation table for “TKU” and “TKV” is similar to that for “TKX”, “TKY”, “TKA” and “TKB”. The only difference lies in that any measurement outside a mechanical pitch does not contribute to accuracy improvement. The compensation table works in the way that interpolation is applied for any value in each interval while movement in the negative coordinate, or negative position in the absolute coordinate, will not be compensated. Two-way compensation can be made to have better accuracy. Note that all the compensated values are integer only and in the range of -127 to 127 . If any *dat0* of the two compensation tables “TKU” and “TKV” is less than 1, neither one of them will work.

d) Display of error compensation table "NKX", "NKY", "NKR", "NKU", "NKV", "NKA", "NKB", "NKC".

Usage:

"NK n i M ;"

n – X: forward movement in X-axis

Y: forward movement in Y-axis

R: forward movement in YAW-axis

U: forward movement in X-axis

V: forward movement in Y-axis

A: backward movement in X-axis

B: backward movement in Y-axis

C: backward movement in YAW-axis

i – change begins with the i -th input data, i is an integer no less than 0, if $i = 0$, input data begins with *dat0*

M – total number of input data

For example, key in "NKX 0 10;" shows the error compensation values "*dat0 dat1 dat2 ... dat9*" for forward movement in X-axis.

16. In-Position command "INP"

Usage:

"INP $p1$;"

$p1$ – This determines the tolerance window for the In-Position signal. Unit: micron.

For example, if $p1=5$, the In-Position signal would be ON when the LMSP motor becomes within +/-5 micron from target position. The In-Position signal is arranged to be at the pin OT6-C of DXIO card. B5 LED on the DXIO card also shows In-Position signal.

Calibration

It's necessary to do calibration when the system is unable to make automatic alignments after power on, a new forcer is used, or a new driver is used. Follow the procedure below to calibrate the system. (Generally, HIWIN has completed calibration for paired driver and motor for customers.)

- a) Set the motor to open loop,
"FX 2"
- b) Calibrate the Position Sensor
"CL"
- c) Define Null periods (position sensor period) after the forcer come to a full stop.
"NP"
- d) Save parameters
"LNS"
- e) Write parameters to EEPROM
"WE"
- f) Set the system to closed-loop again.
"FX 1"

VI. PROGRAM DOWNLOAD FROM PC TO DSP

In case there is an updated version of the DSP software, which drives the LMSP motor, it is possible to re-burn the flash memory and update the DSP codes inside LMDX controller box. It is also possible to download motor parameter files (*.mot) to LMDX. Or it is preferred to write motion programs off-line on PC and download it later to LMDX. For your convenience, *HIWIN* provides such a tool, named “DspHost”, to download programs through RS-232 connection. The program could also serve as an general purpose terminal program.

Included in the CD-ROM, there is a directory called “DspHost”. Run the “setup.exe” program in this directory to install.

- 1) Run “DspHost.exe” and a window as shown in Fig. 6.1 will be brought up.

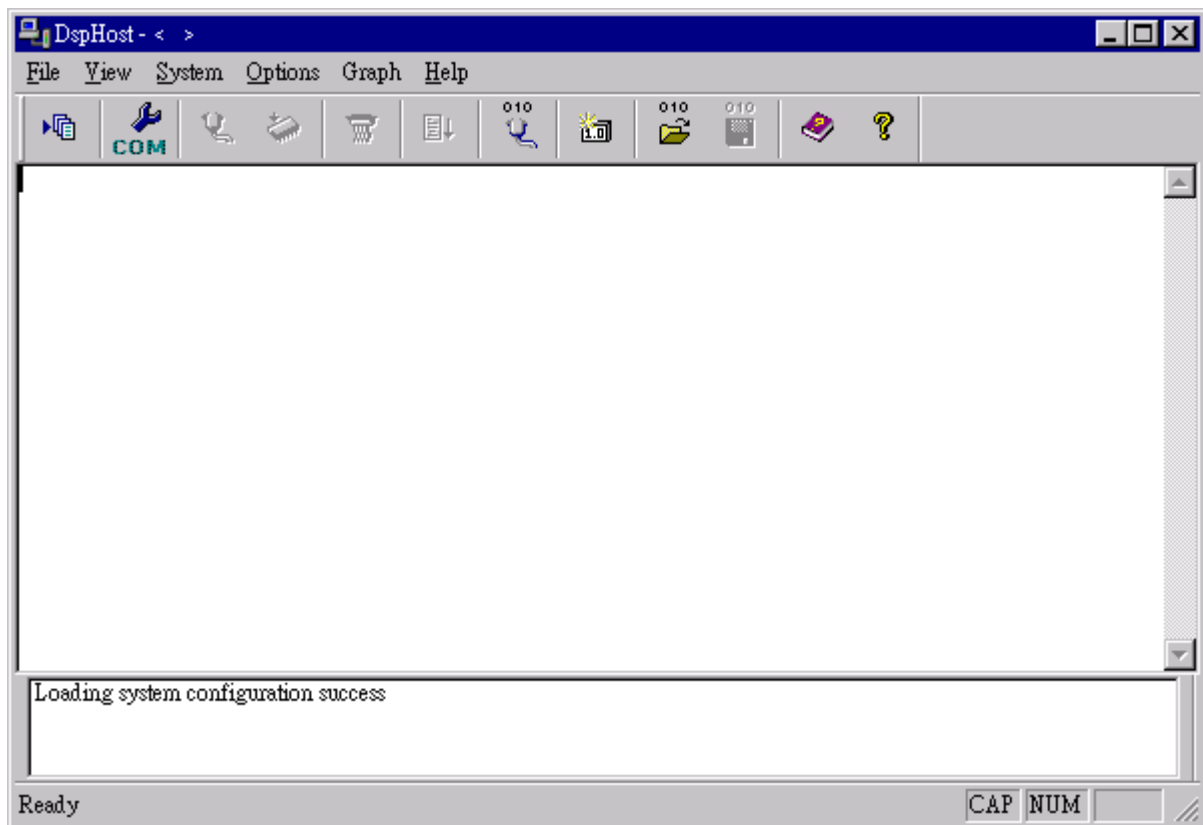


Figure 6.1 “DspHost” window

- 2) Click on “COM Port Settings...” under “System” (Fig. 6.2) or push the “COM” icon button to make correct settings (Fig. 6.3) based on the serial port you use.

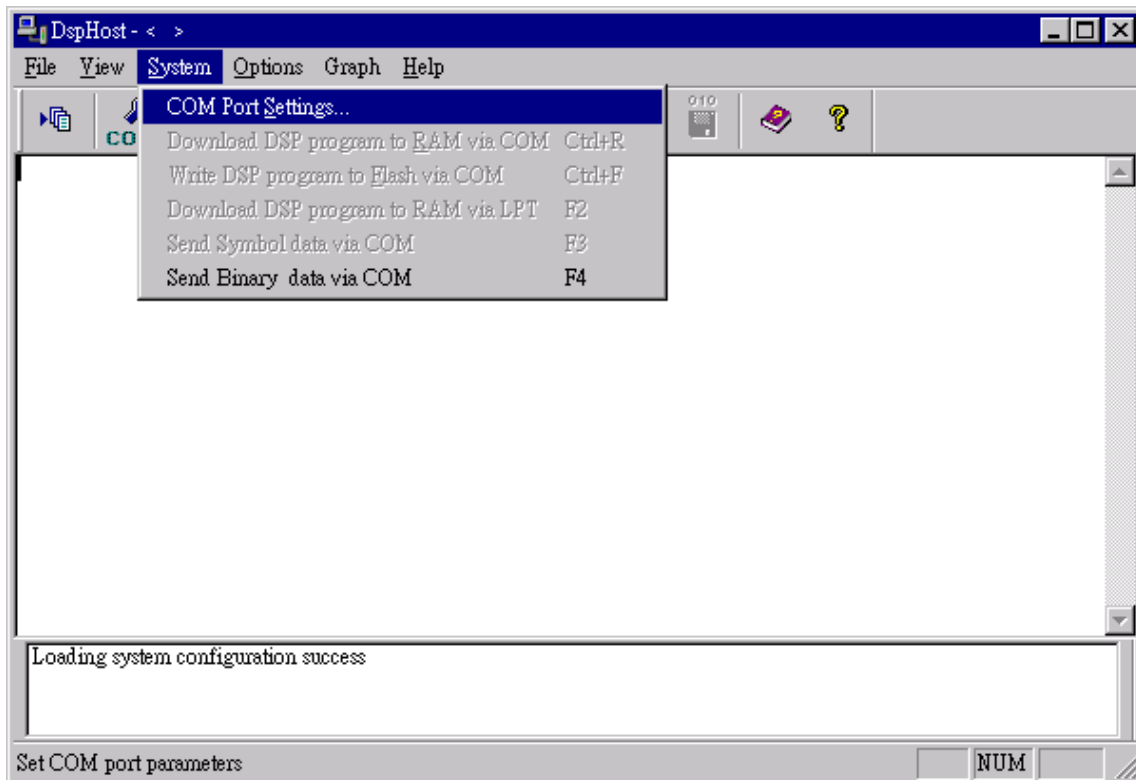


Figure 6.2 Use menu to select “COM Port Settings...”

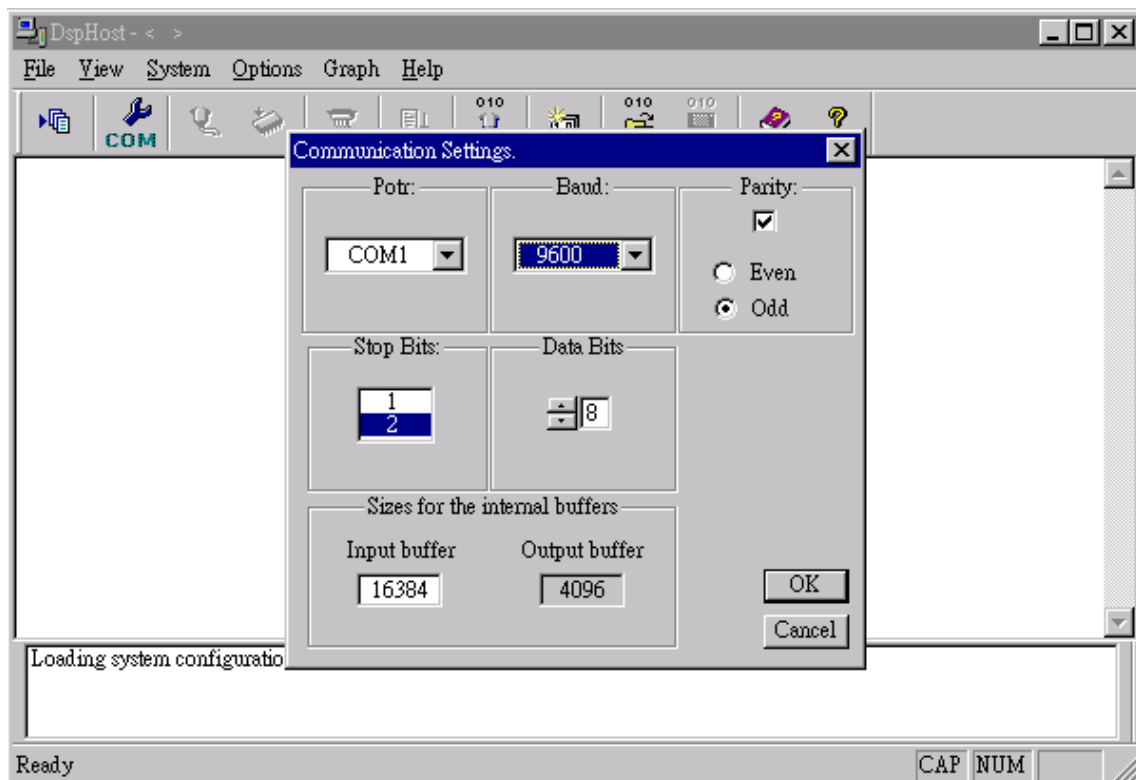


Figure 6.3 Example of COM1 Communication Settings

3) Download DSP code (firmware) to flash memory in LMDX controller box.

The error compensation data will not be saved automatically while updating DSP codes by use of “DspHost”. **So please make sure to take down the error compensation table data using “NKn” command before downloading any programs using “DspHost”.**

The DSP code files have a file extension of COD (*.cod)

The procedure to download programs to DSP board is described as follows:

(a) Make sure that the connection of RS-232 cable between PC and LMDX is OK.

Execute DspHost.exe. Turn on LMDX, normally the LMDX (monitor program) will send an asterisk (*) to the terminal program. If within 0.7 seconds at this moment backspace key is pressed, the LMDX monitor program will be activated and take over. See Fig. 6.4 After this, close DspHost, then execute DspHost again.

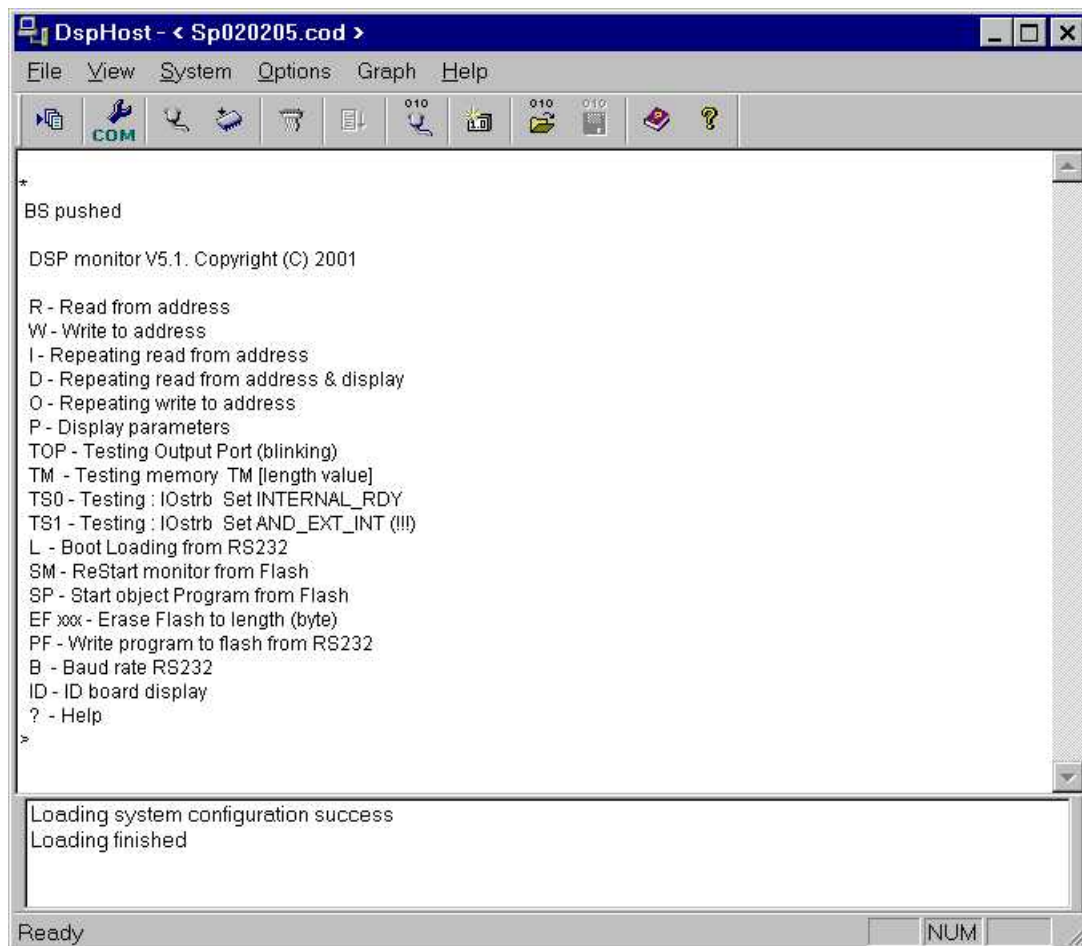


Figure 6.4 LMDX monitor program

(b) Click on “Select File...” in “File” (Fig. 6.5) to choose the firmware (Fig. 6.6) to download.

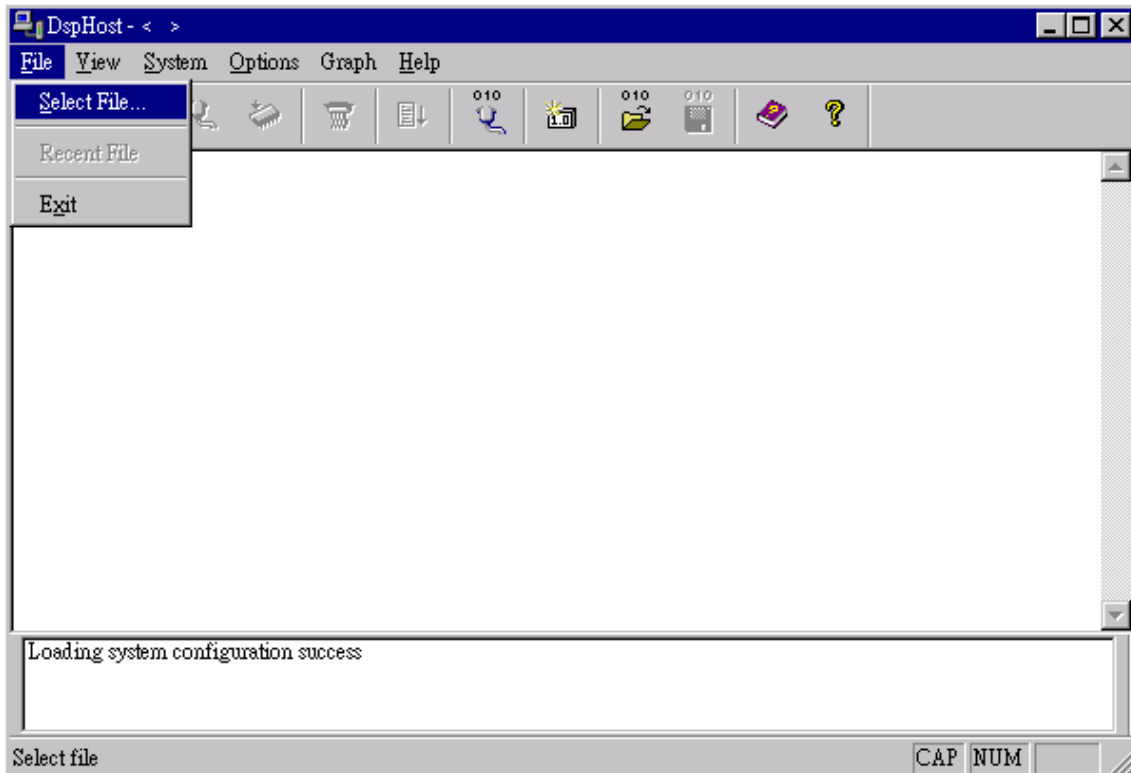


Figure 6.5 Use menu to select a firmware file

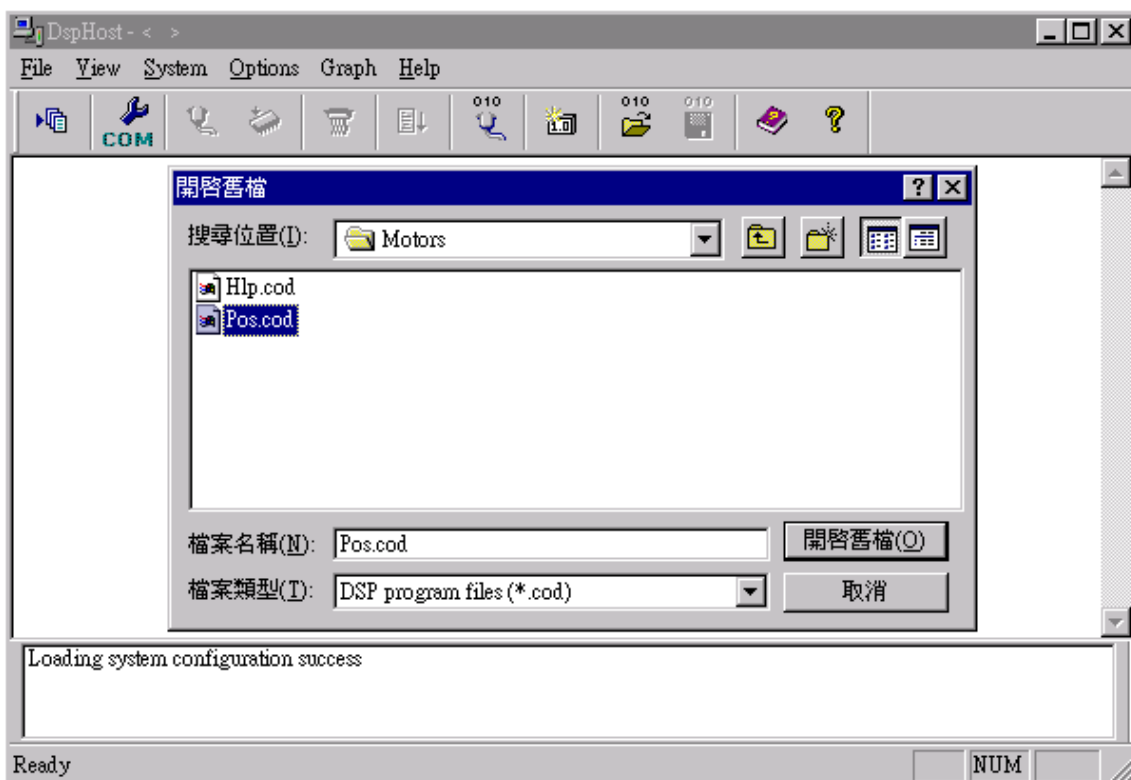


Figure 6.6 Select DSP program files (firmware) to open

(c) Click on “Write DSP program to E_lash via COM” in “System” (Fig. 6.7) or  to

download the chosen DSP program to LMDX controller box's flash memory.

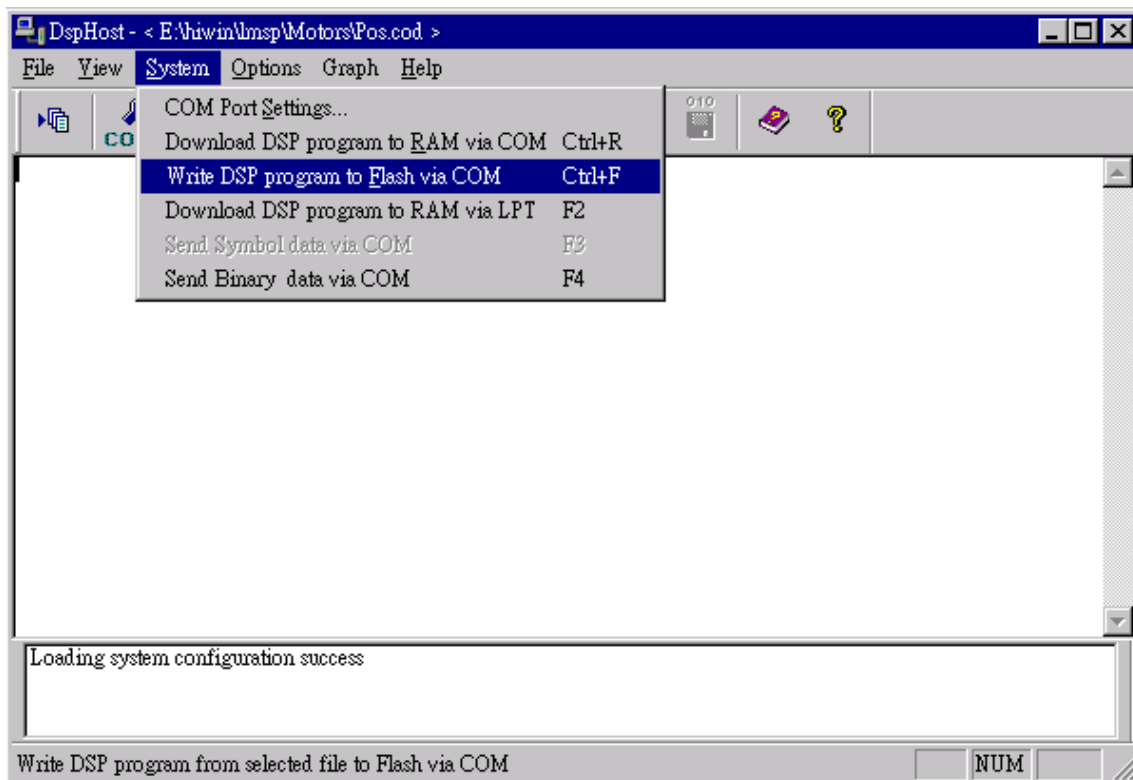


Figure 6.7 Write DSP program to flash memory via COM

- (d) After the download process (Fig.6.8), the program should be in the DSP flash memory. Now you can turn LMDX off and on again or you can issue the SP command to run the downloaded DSP code.

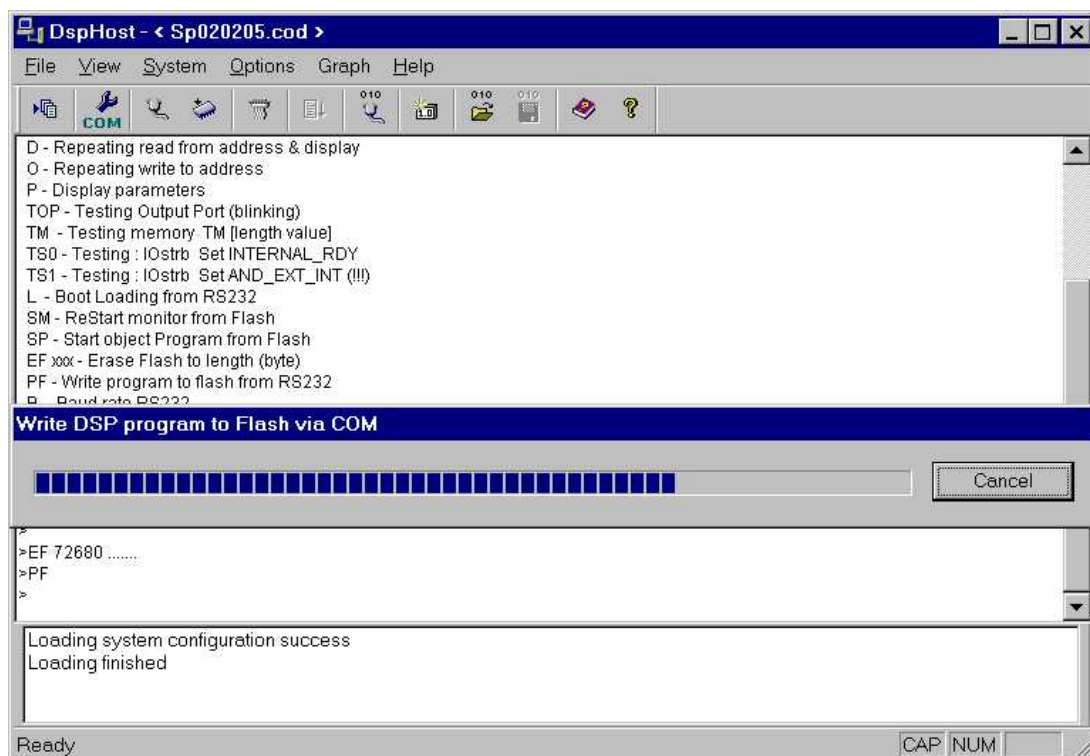


Figure 6.8 Download process of DSP program

4) Download motor parameters to the LMDX controller box.

(a) Click on “Select File...” in “File” (as in Fig. 6.5) to select the desired file (Fig. 6.9).

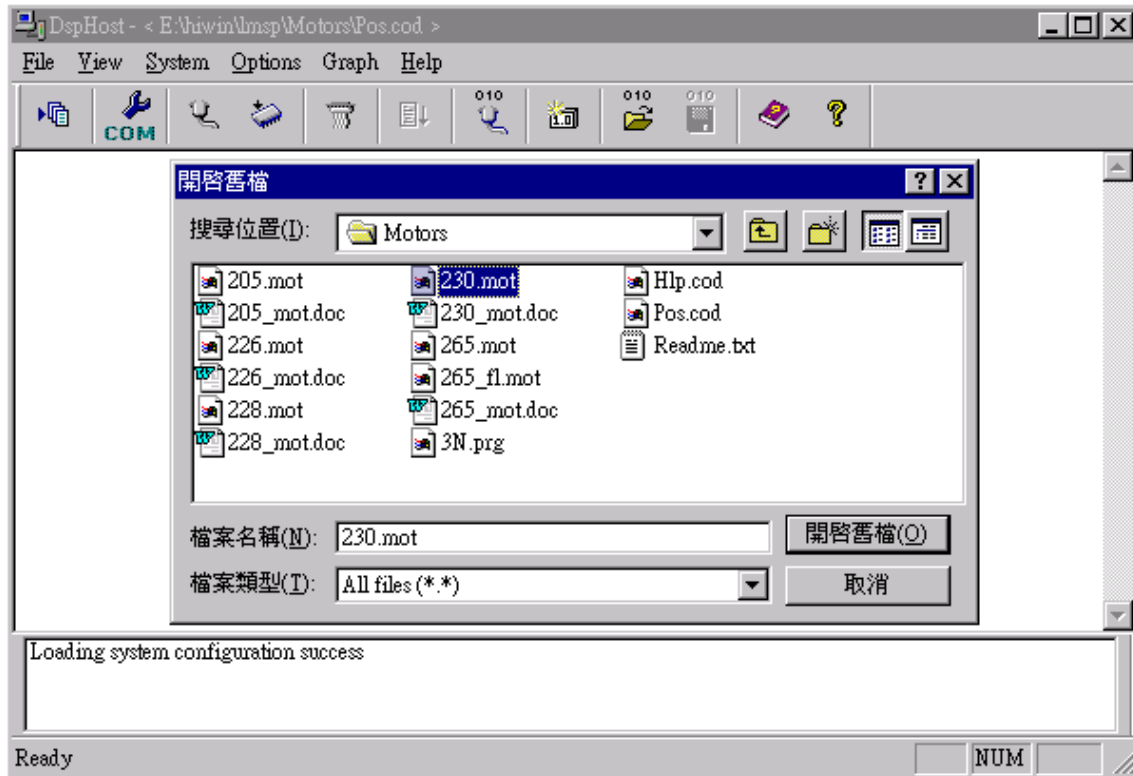


Figure 6.9 Select the file with desired motor parameters

(b) Click on “Send Symbol data via COM” in “System” to download motor parameters via RS-232 (Fig. 6.10).

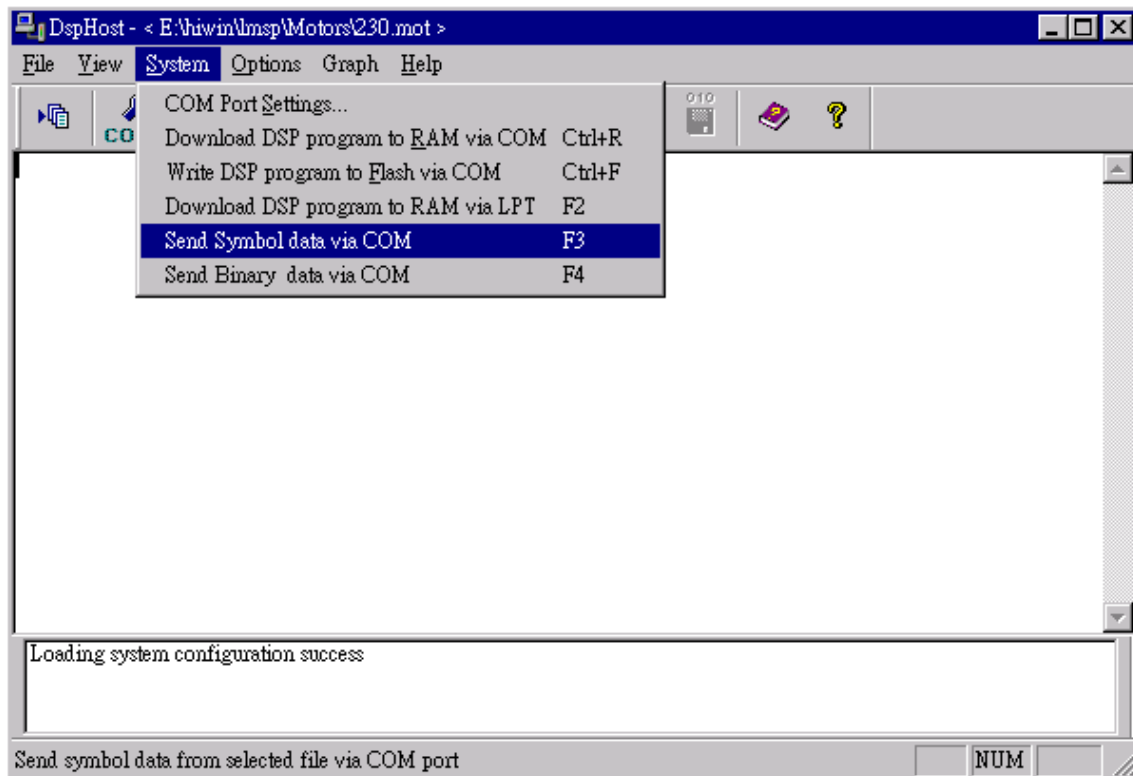


Figure 6.10 Download motor parameters to the LMDX controller box

5) Download user's motion programs to the LMDX controller box's DSP.

An example motion program "3n_p1.prg" is also included in the CD-ROM.

(a) Click on "Select File..." in "File" (Fig. 6.11) to select user's program file (Fig. 6.12).

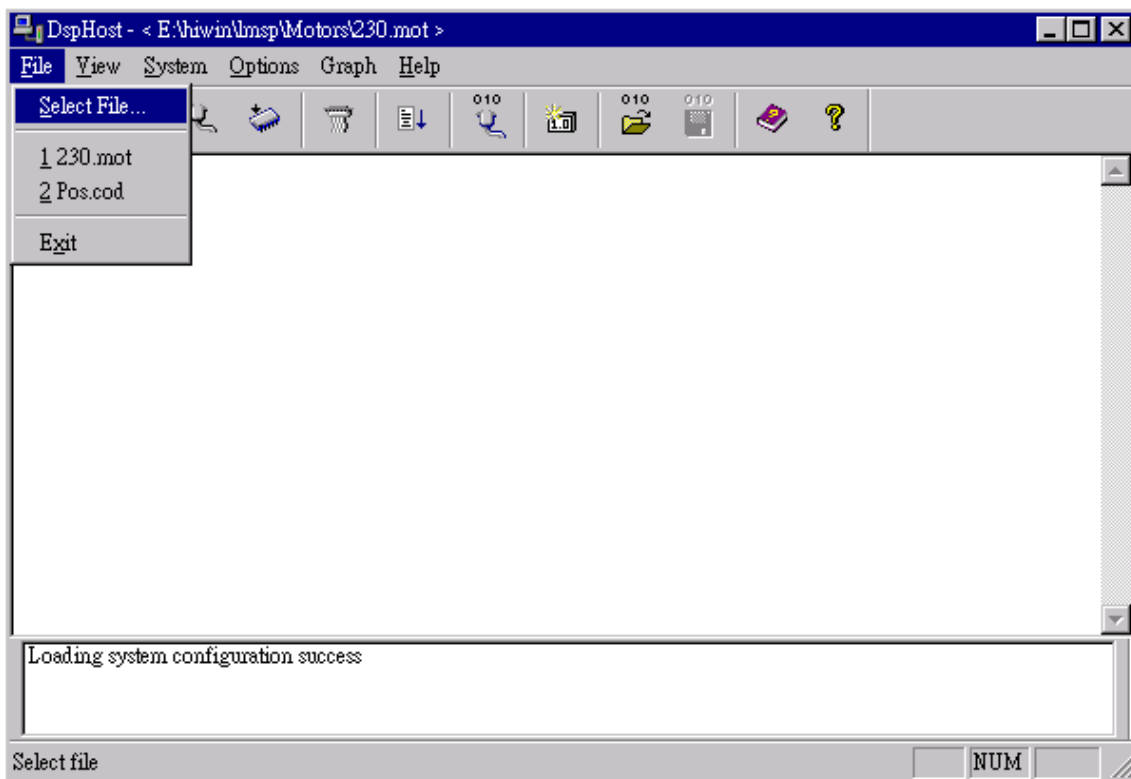


Figure 6.11 Use menu to select user's program file

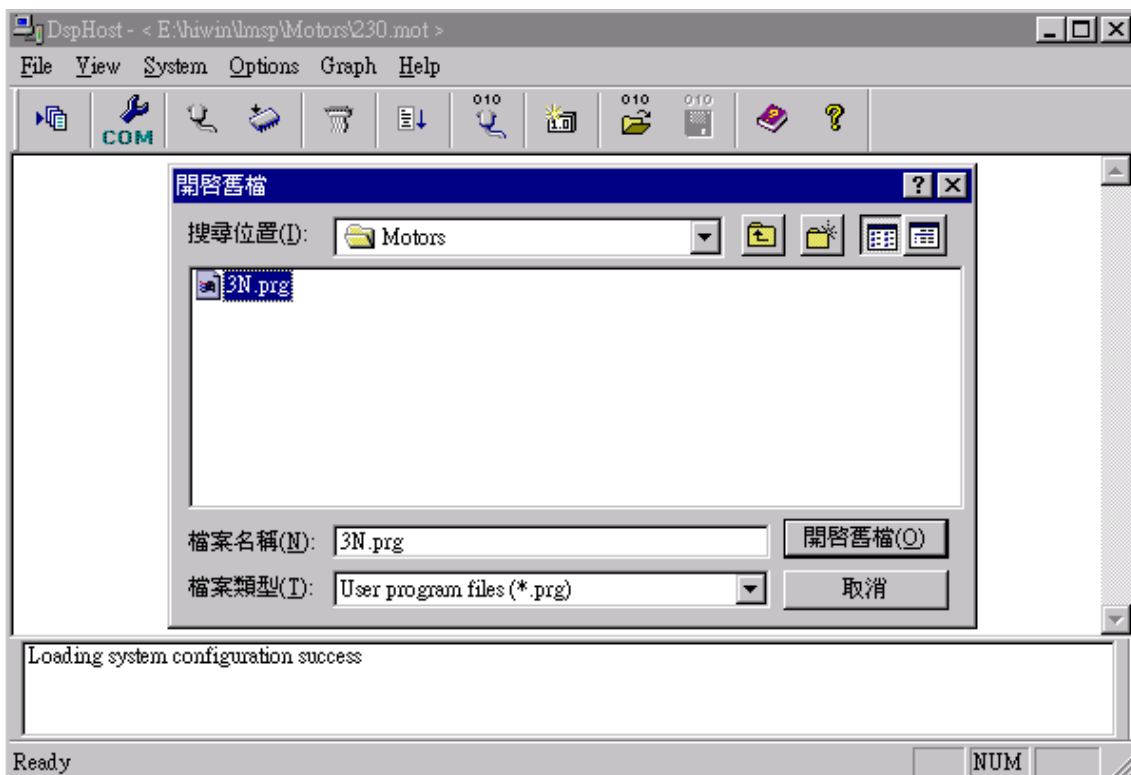


Figure 6.12 Select user's program file to open

- (b) Click on "Send Binary data via COM" in "System" to download user's program file via RS-232 (Fig. 6.13).

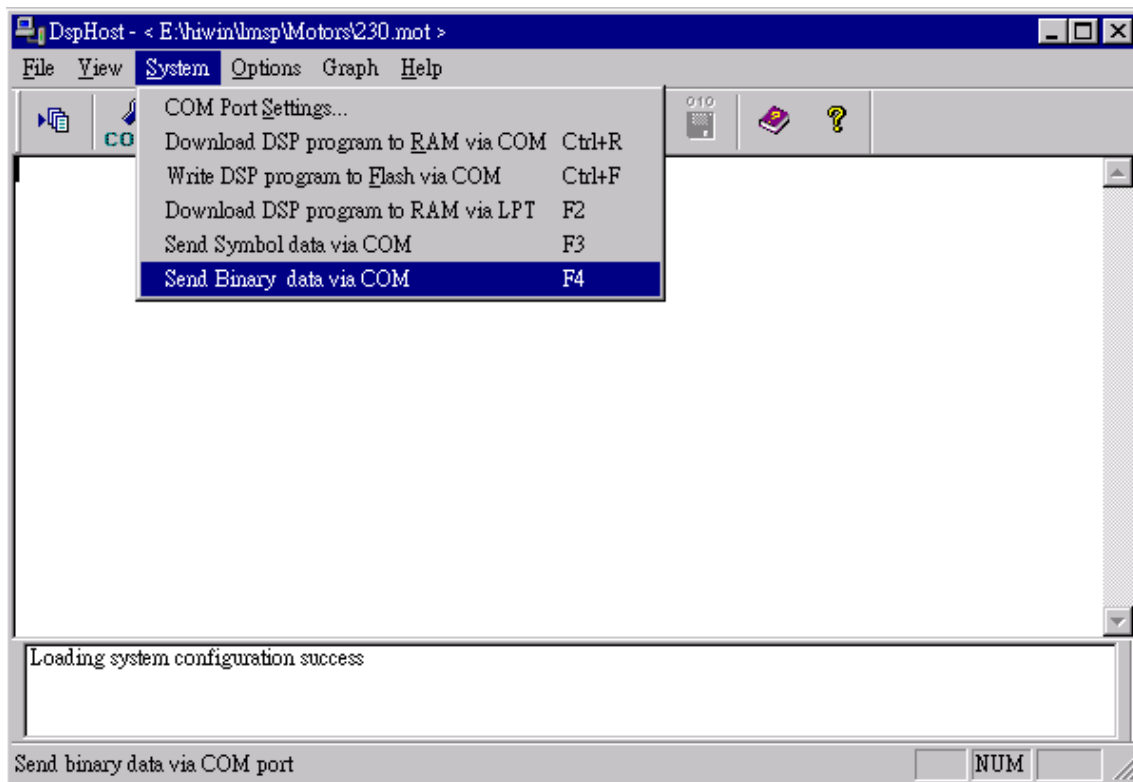


Figure 6.13 Download user's program file to DSP board

VII. HELP

If you have any questions concerning LMSP and/or LMDX controller box, *HIWIN* engineers are always there to assist you. Please take notes of your questions, error messages, firmware version, and serial number before contacting us. We can be reached as shown below.

HIWIN MIKROSYSTEM CORP.

No. 3, 6th Road

Taichung Industrial Park

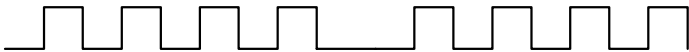
Taichung 407, Taiwan

TEL: 886-4-23550110 ext. 511, 512, 513

FAX: 886-4-23550123

APPENDIX A PULSE COMMAND FORMAT

There are two formats of pulse commands: CW/CCW and STEP/DIR. Only STEP/DIR is available in LMDX controller box. Its format is shown below.

	Positive direction	Negative direction
STEP		
DIR	