ROBOTIS CO., LTD. www.robotis.com

Servomotores DYNAMIXEL

Dr. Alejandro Aceves López

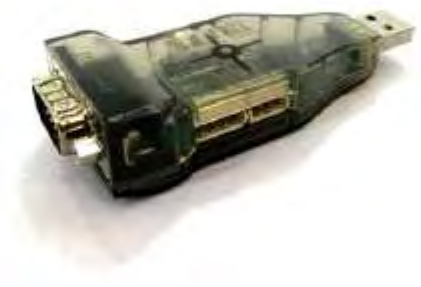
Seminario del grupo de investigación en Humanoides

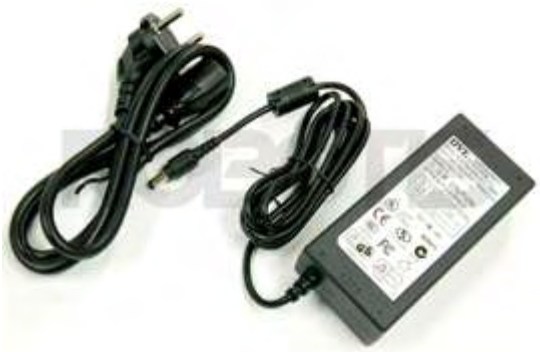
Agenda

1. Material necesario
2. Instalar USB2Dynamixel
3. Instalar RoboPlus y probar el AX-12
4. Conocer a fondo el AX-12
5. Usar TERMITE con AX-12
6. Usar el AX-12 con MATLAB

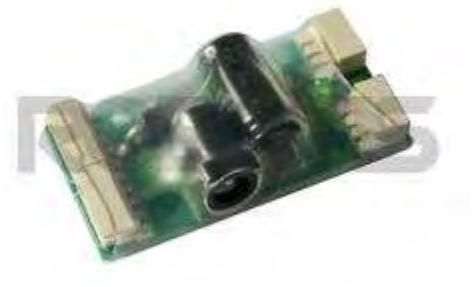
1.- Material necesario

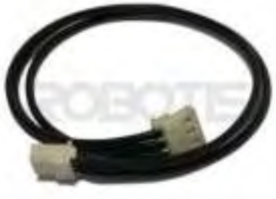
USB2Dynamixel





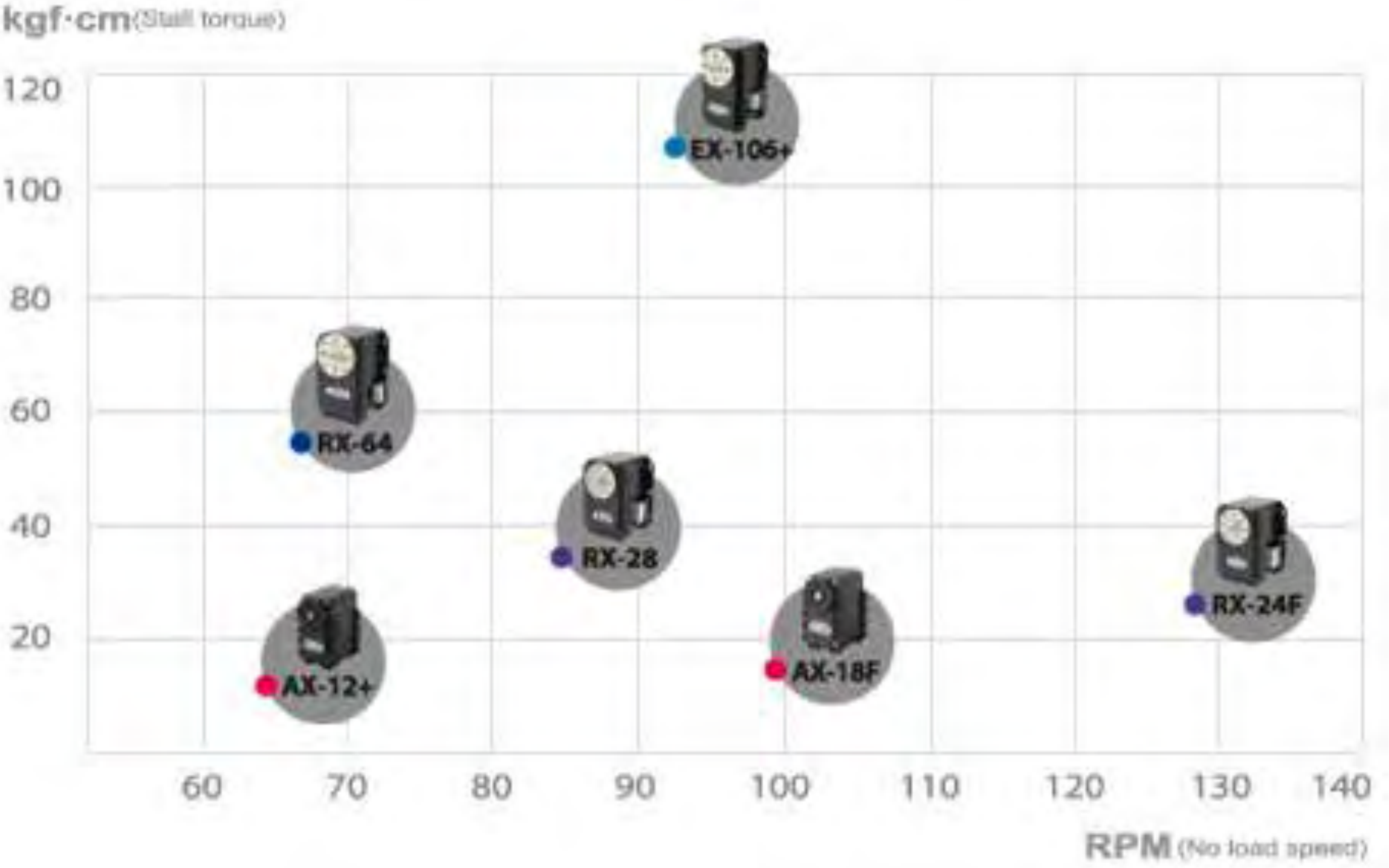
SMPS2Dynamixel



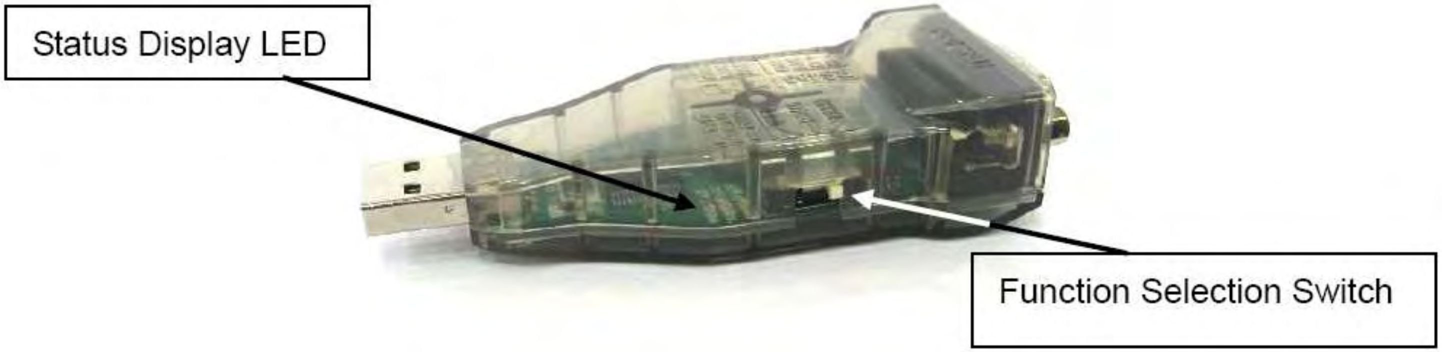


	AX-12	
Weight (g)	55	
Gear Reduction Ratio	1/254	
Input Voltage (V)	at 7V	at 10V
Final Max Holding Torque(kgf.cm)	12	16.5
Sec/60degree	0.269	0.196

	RX-64	
Weight (g)	125	
Dimension (mm)	40.2 x 61.1 x 41.0	
Gear Reduction Ratio	1/200	
Applied Voltage (V)	at 15V	at 18V
Final Reduction Stopping Torque (kgf.cm)	64.4	77.2
Speed (Sec/60 degrees)	0.188	0.157

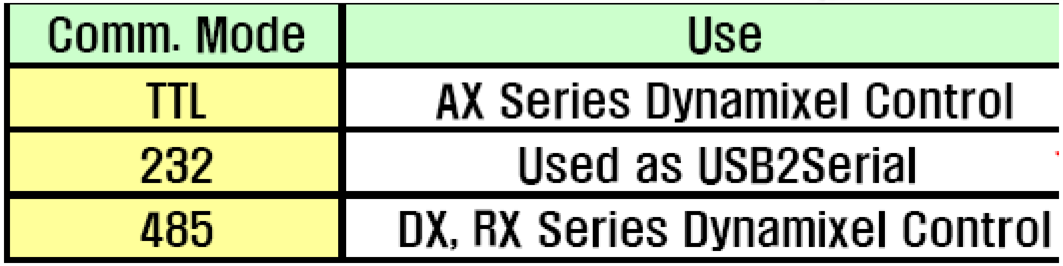

kgf·cm (Stall torque)
120
100
80
60
40
20
60
70
80
90
100
110
120
130
140
RPM (No load speed)
EX-106+
RX-64
RX-28
RX-24F
AX-12+
AX-18F

2.- Instalar USB2Dynamixel

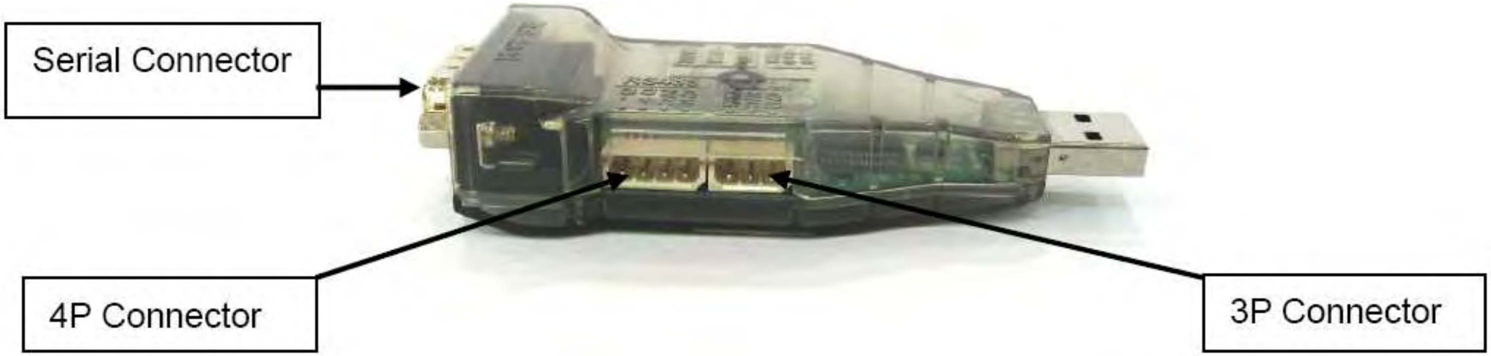


Status Display LED

Function Selection Switch



Comm. Mode	Use
TTL	AX Series Dynamixel Control
232	Used as USB2Serial
485	DX, RX Series Dynamixel Control

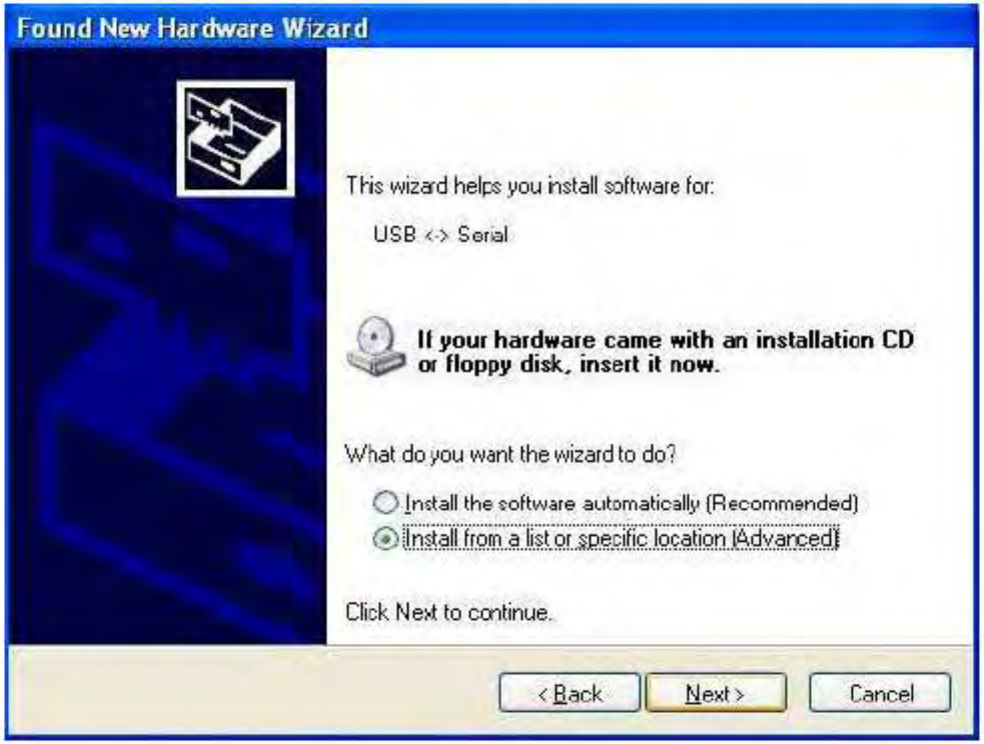


Serial Connector

4P Connector

3P Connector

< Step 1 > When USBtoSerial is connected to PC, the following “New Hardware” dialog window will appear, here, users select “Install from a list of specific location.”



Found New Hardware Wizard

This wizard helps you install software for:

USB <-> Serial

If your hardware came with an installation CD or floppy disk, insert it now.

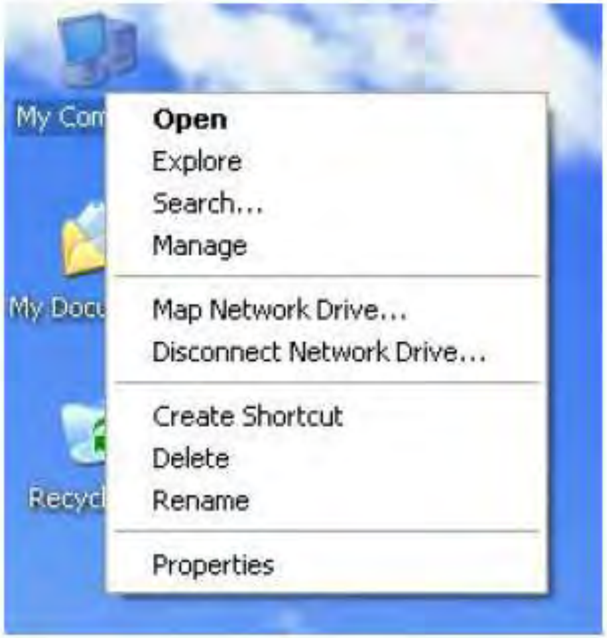
What do you want the wizard to do?

○ Install the software automatically (Recommended)

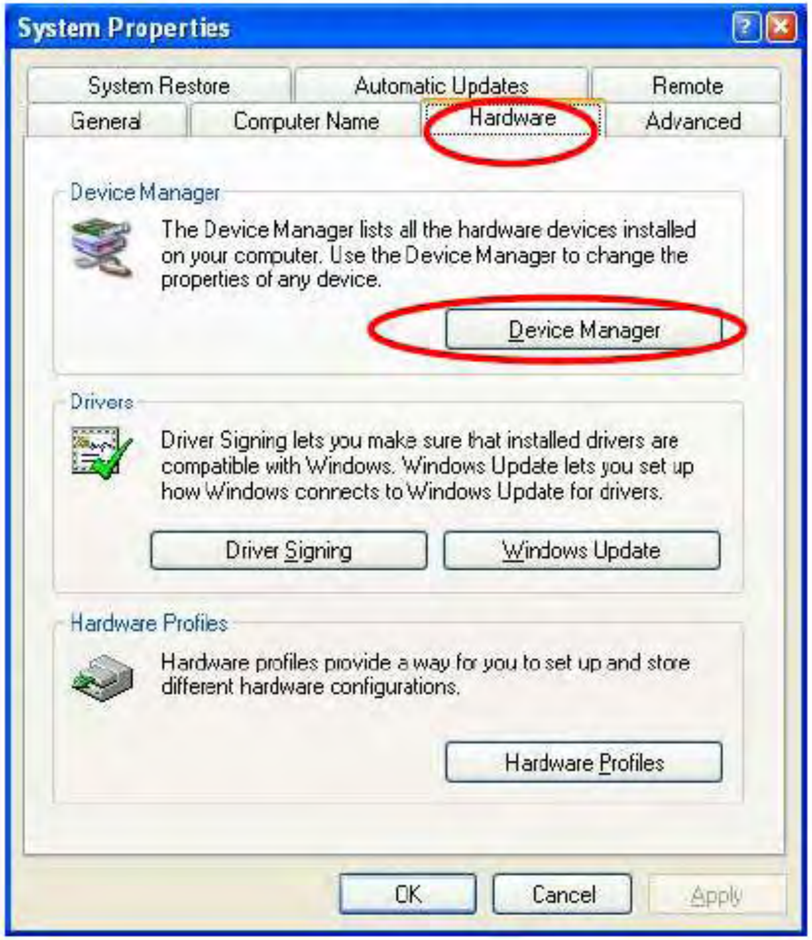
● Install from a list or specific location (Advanced)

Click Next to continue.

Right click the "My Computer" icon of Window desktop, and then select "Properties."



Select "Hardware" tab and select the "Device Manager"



Device Manager

The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

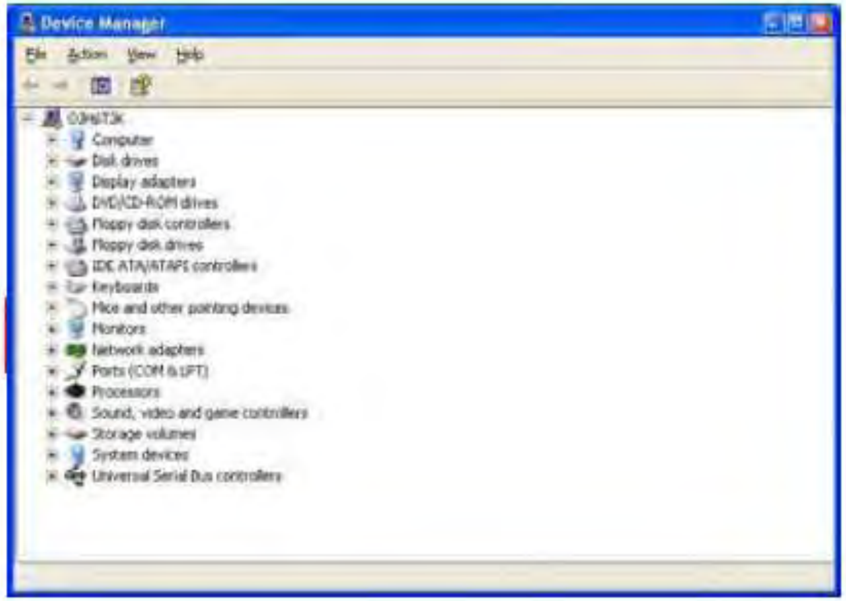
Drivers

Driver Signing lets you make sure that installed drivers are compatible with Windows. Windows Update lets you set up how Windows connects to Windows Update for drivers.

Hardware Profiles

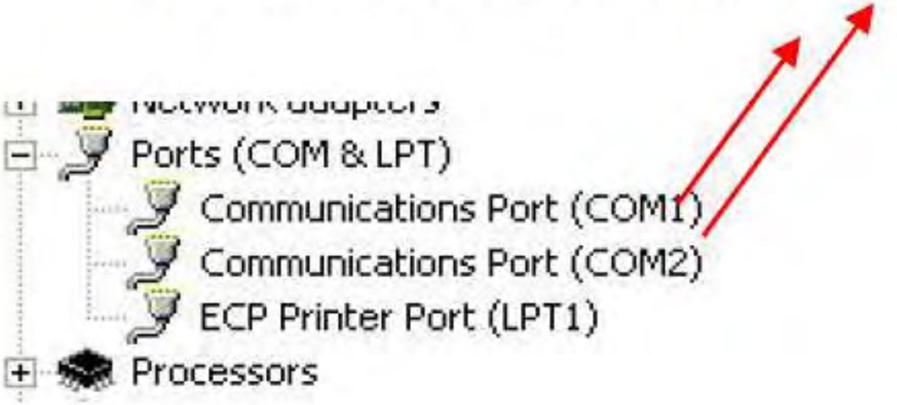
Hardware profiles provide a way for you to set up and store different hardware configurations.

Select Port (Com & LPT) from the list



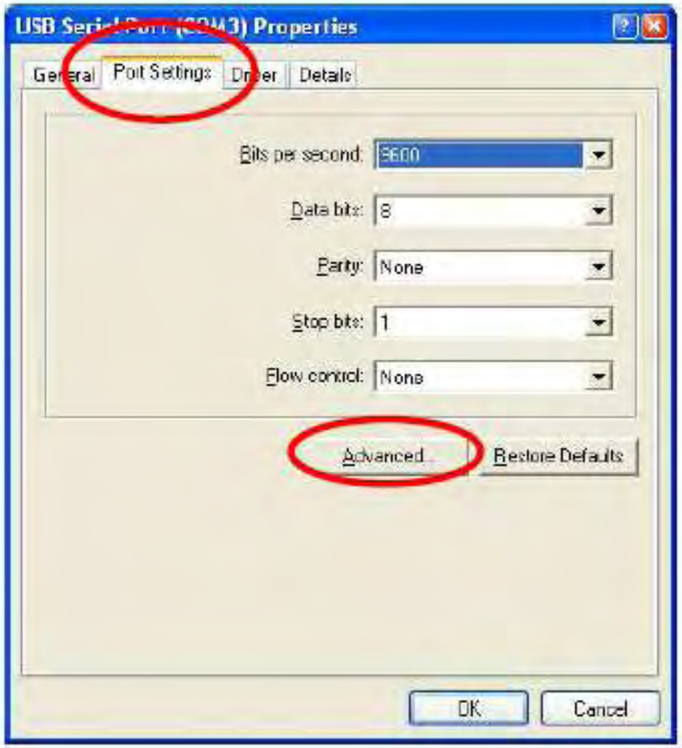
COMNT3K
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Floppy disk controllers
- Floppy disk drives
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- Processors
- Sound, video and game controllers
- Storage volumes
- System devices
- Universal Serial Bus controllers

Serial COM Ports that can be connected



- Network adapters
- Ports (COM & LPT)
  - Communications Port (COM1)
  - Communications Port (COM2)
  - ECP Printer Port (LPT1)
- Processors

< Step 6 > After checking driver information, change "Latency Timer" to 1ms. This change allows for the fast response time.



USB Serial Port (COM3) Properties

General | Port Settings | Driver | Details

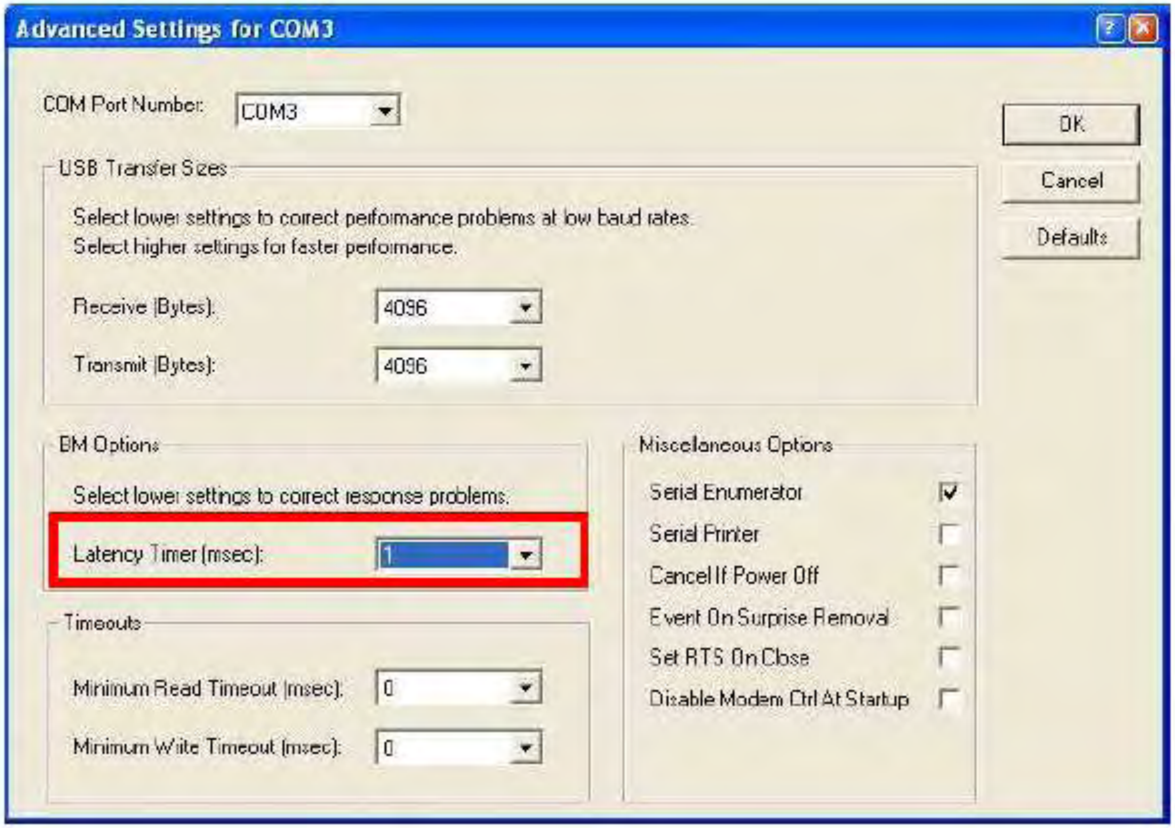
Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None



Advanced Settings for COM3

COM Port Number: COM3

USB Transfer Sizes

Select lower settings to correct performance problems at low baud rates.
Select higher settings for faster performance.

Receive (Bytes): 4096

Transmit (Bytes): 4096

BM Options

Select lower settings to correct response problems.

Latency Timer (msec): 1

Timeouts

Minimum Read Timeout (msec): 0

Minimum Write Timeout (msec): 0

Miscellaneous Options

Serial Enumerator ☑

Serial Printer □

Cancel If Power Off □

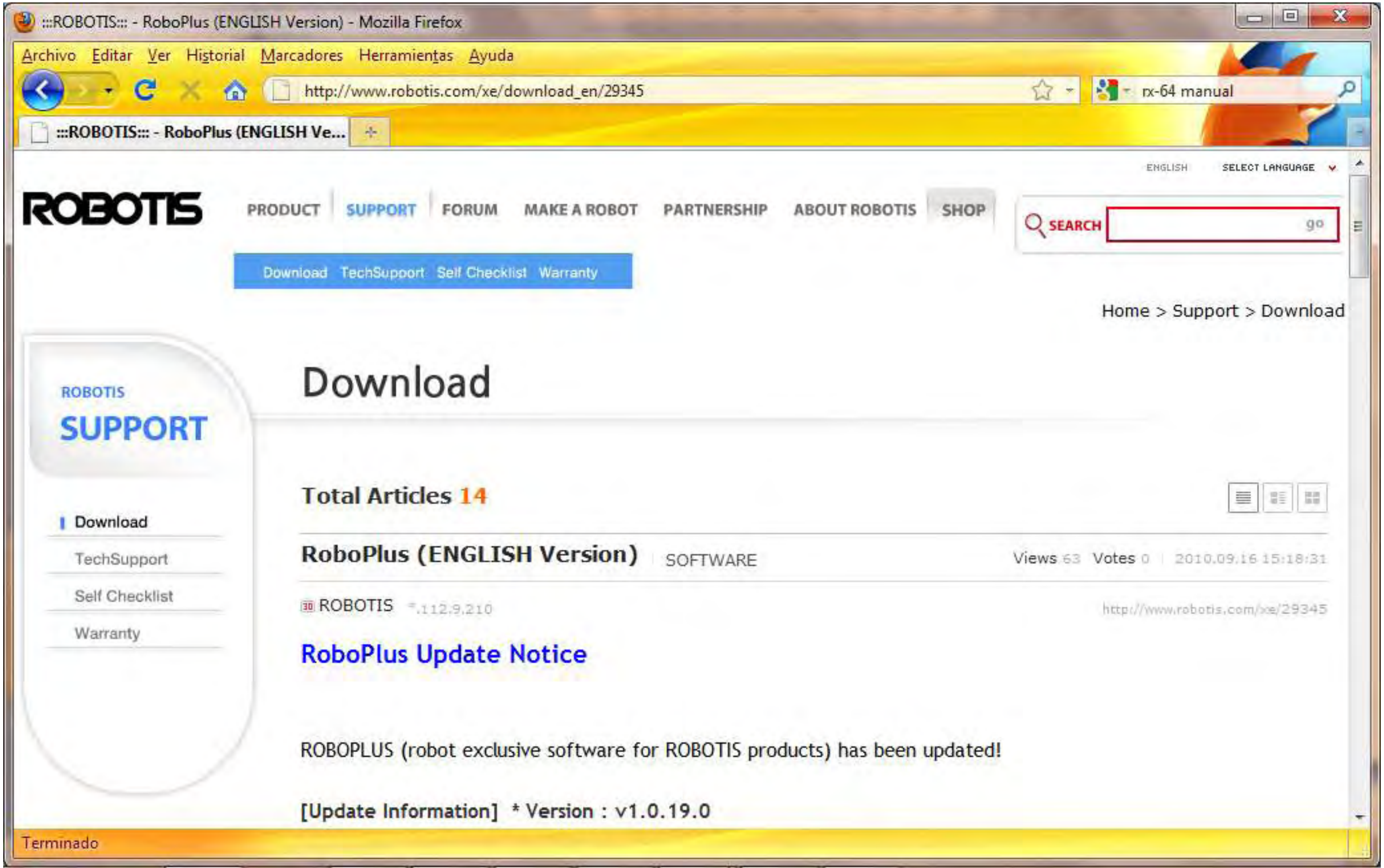
Event On Surprise Removal □

Set RTS On Close □

Disable Modem Ctrl At Startup □

3.- Instalar RoboPlus y probar AX12

Ingresar a: http://www.robotis.com/xe/download_en


:::ROBOTIS::: - RoboPlus (ENGLISH Version) - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
http://www.robotis.com/xe/download_en/29345
rx-64 manual
:::ROBOTIS::: - RoboPlus (ENGLISH Ve...
ENGLISH
SELECT LANGUAGE
ROBOTIS
PRODUCT
SUPPORT
FORUM
MAKE A ROBOT
PARTNERSHIP
ABOUT ROBOTIS
SHOP
SEARCH
go
Download TechSupport Self Checklist Warranty
Home > Support > Download
ROBOTIS
SUPPORT
Download
TechSupport
Self Checklist
Warranty
Terminado

Download

Total Articles 14

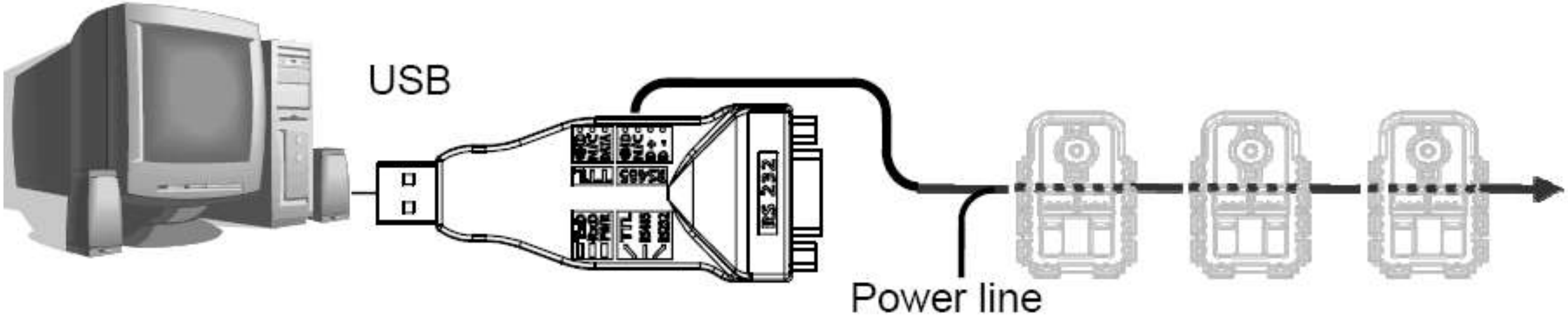
RoboPlus (ENGLISH Version) SOFTWARE Views 63 Votes 0 2010.09.16 15:18:31

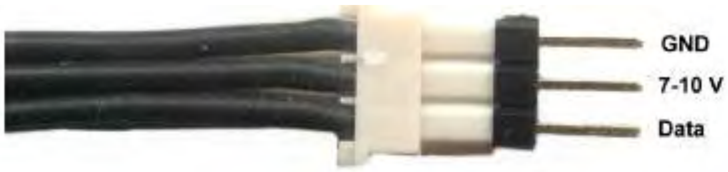
ROBOTIS *.112.9.210 http://www.robotis.com/xe/29345

RoboPlus Update Notice

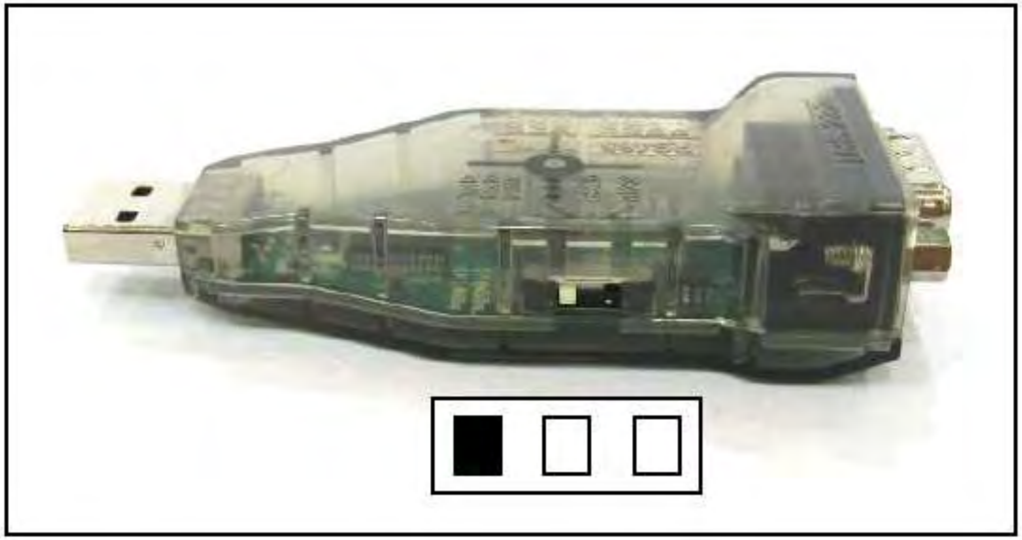
ROBOPLUS (robot exclusive software for ROBOTIS products) has been updated!

[Update Information] * Version : v1.0.19.0

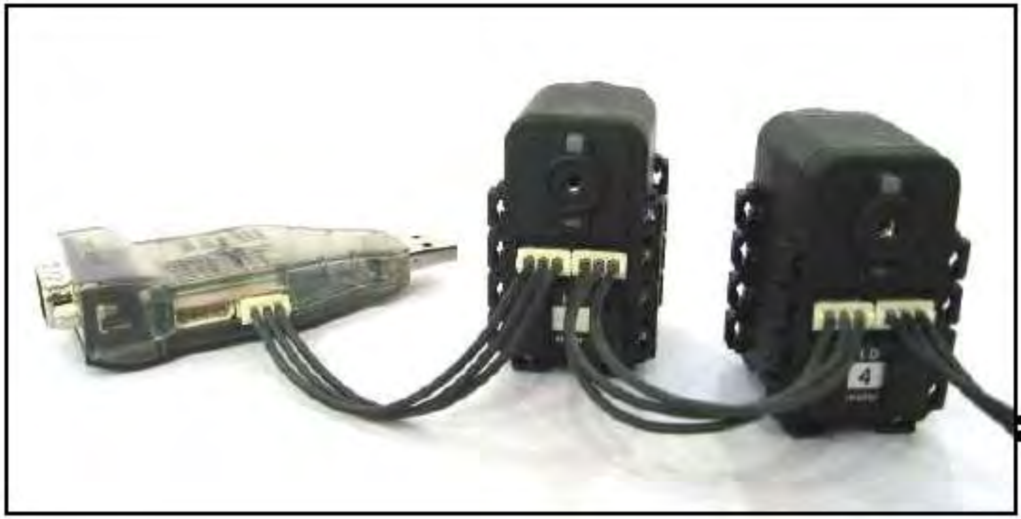

USB
TTL
RS485
RS 232
Power line


GND
7-10 V
Data

ID
14
ID
16
ID
18

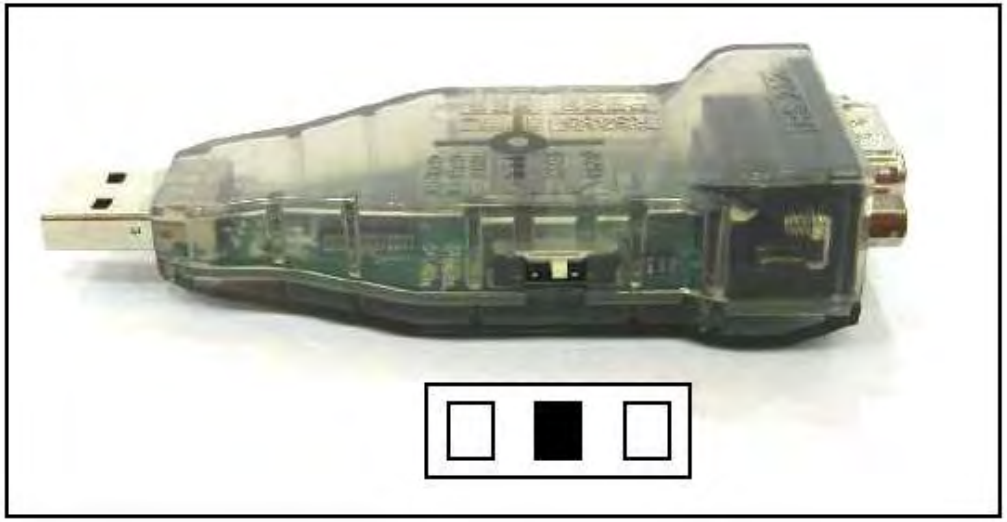


Set the function selection switch to TTL mode

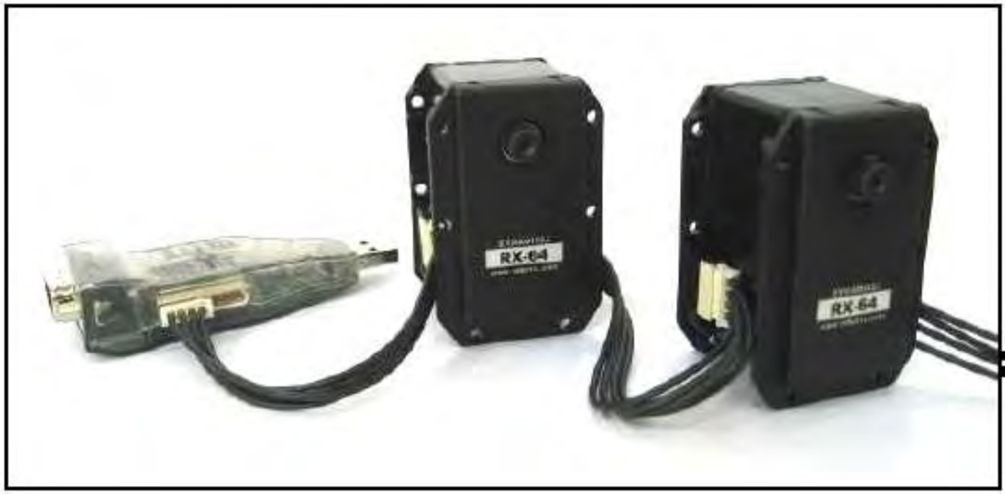


Dynamixels can be connected in series using a 3P cable.

Connect the power line to the last dynamixel.



Set the selection function switch to RS 485 mode

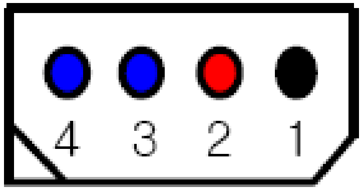


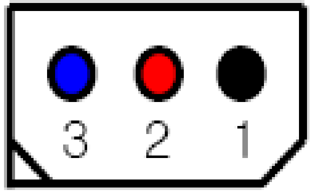
Dynamixels can be connected in series using a 4P cable.

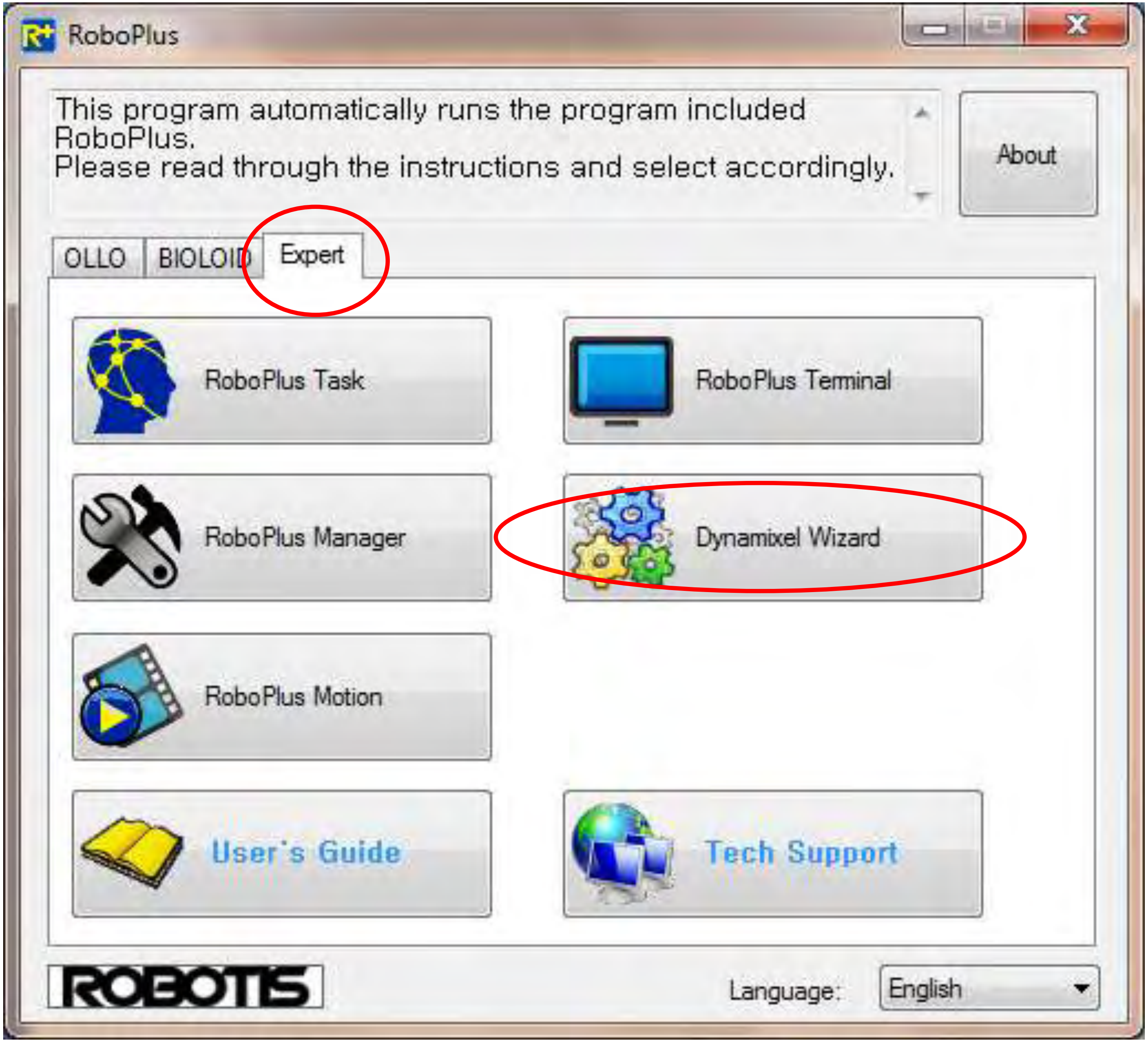
Connect the power line to the last dynamixel.

POWER
(RX-10: DC10 to 12V)
(RX-28: DC12 to 16V)
(RX-64: DC15 to 18V)
(DX-117: DC12 to 16V)

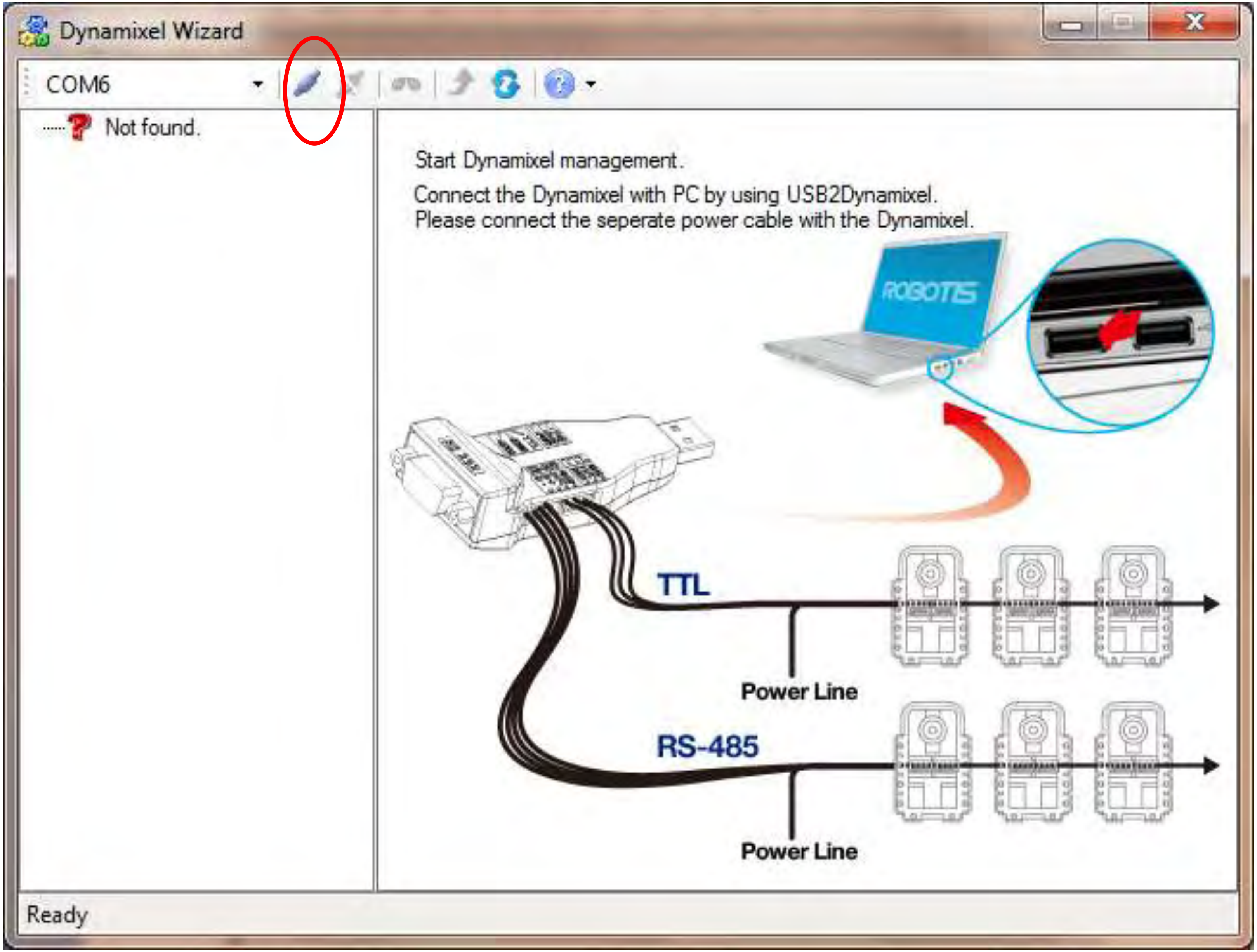
Pin Figure of 4P/3P Cable Connector

4 Pin Cable		
Pin No.	Signal	Pin Figure
1	GND	
2	POWER	
3	DATA + (RS-485)	
4	DATA - (RS-485)	

3 Pin Cable		
Pin No.	Signal	Pin Figure
1	GND	
2	POWER	
3	DATA (TTL)	



This program automatically runs the program included RoboPlus.
Please read through the instructions and select accordingly.



Dynamixel Wizard

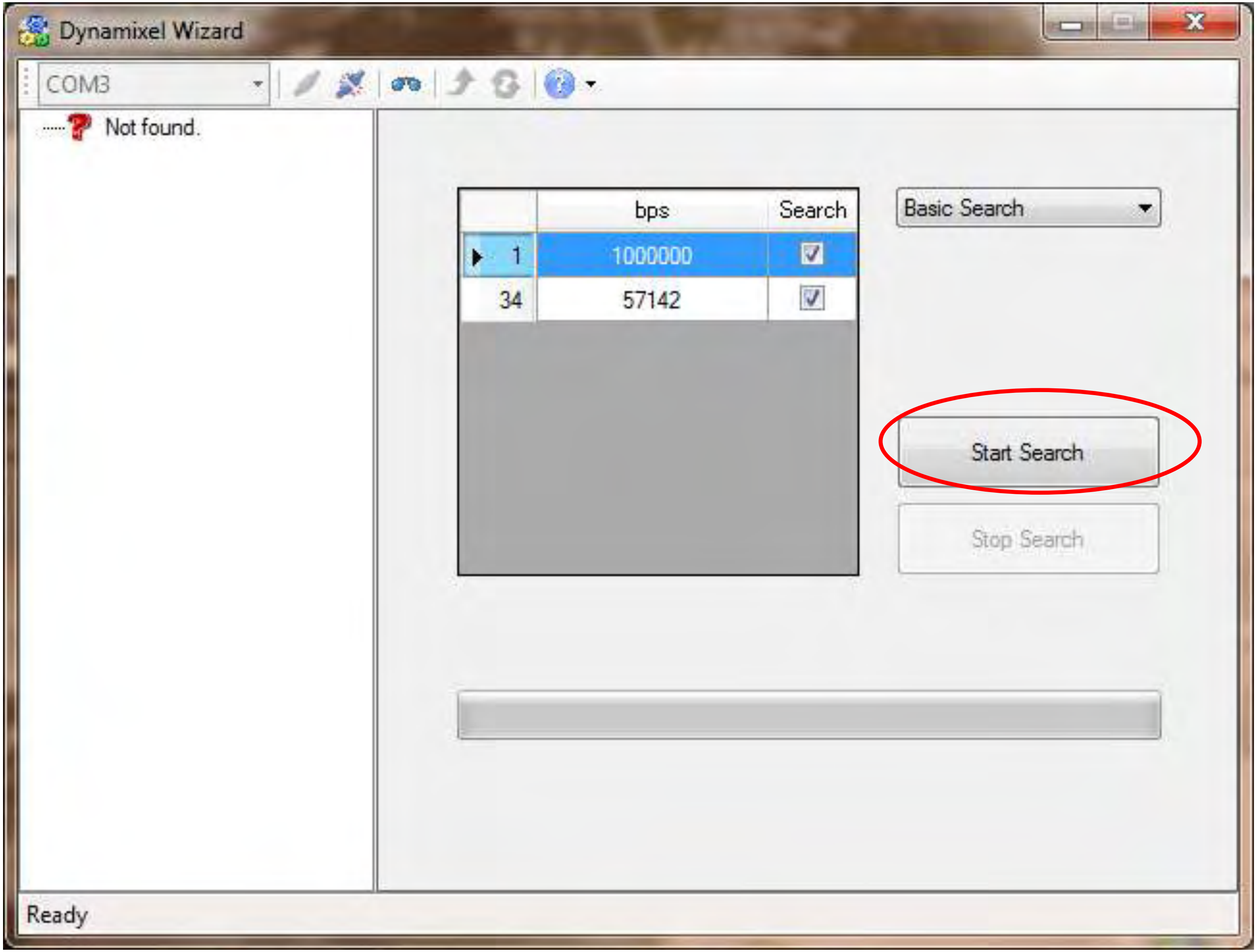
COM6

Not found.

Start Dynamixel management.

Connect the Dynamixel with PC by using USB2Dynamixel.
Please connect the seperate power cable with the Dynamixel.

Ready



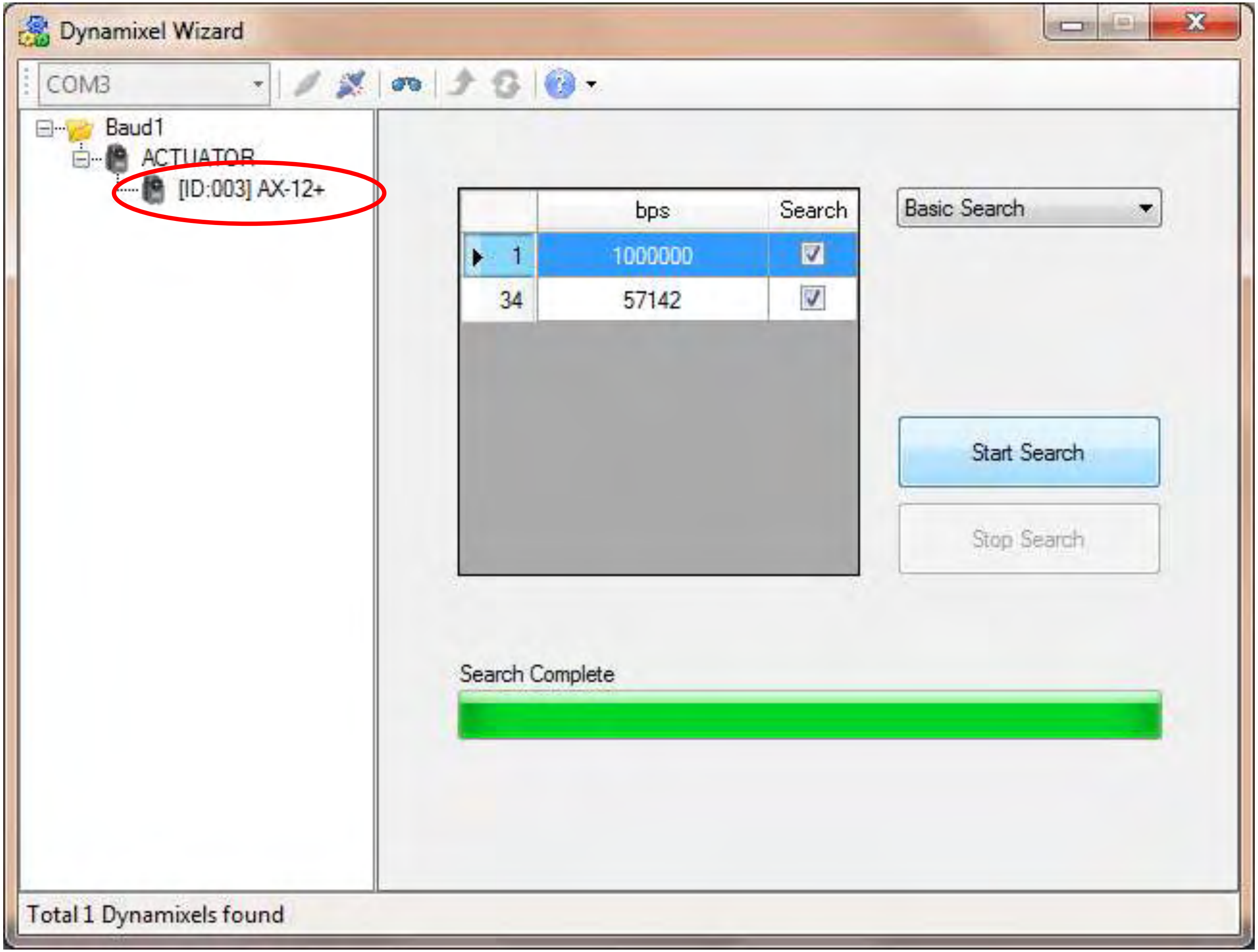
COM3

Not found.

	bps	Search
1	1000000	☑
34	57142	☑

Basic Search

Ready



Dynamixel Wizard

COM3

Baud1

ACTUATOR

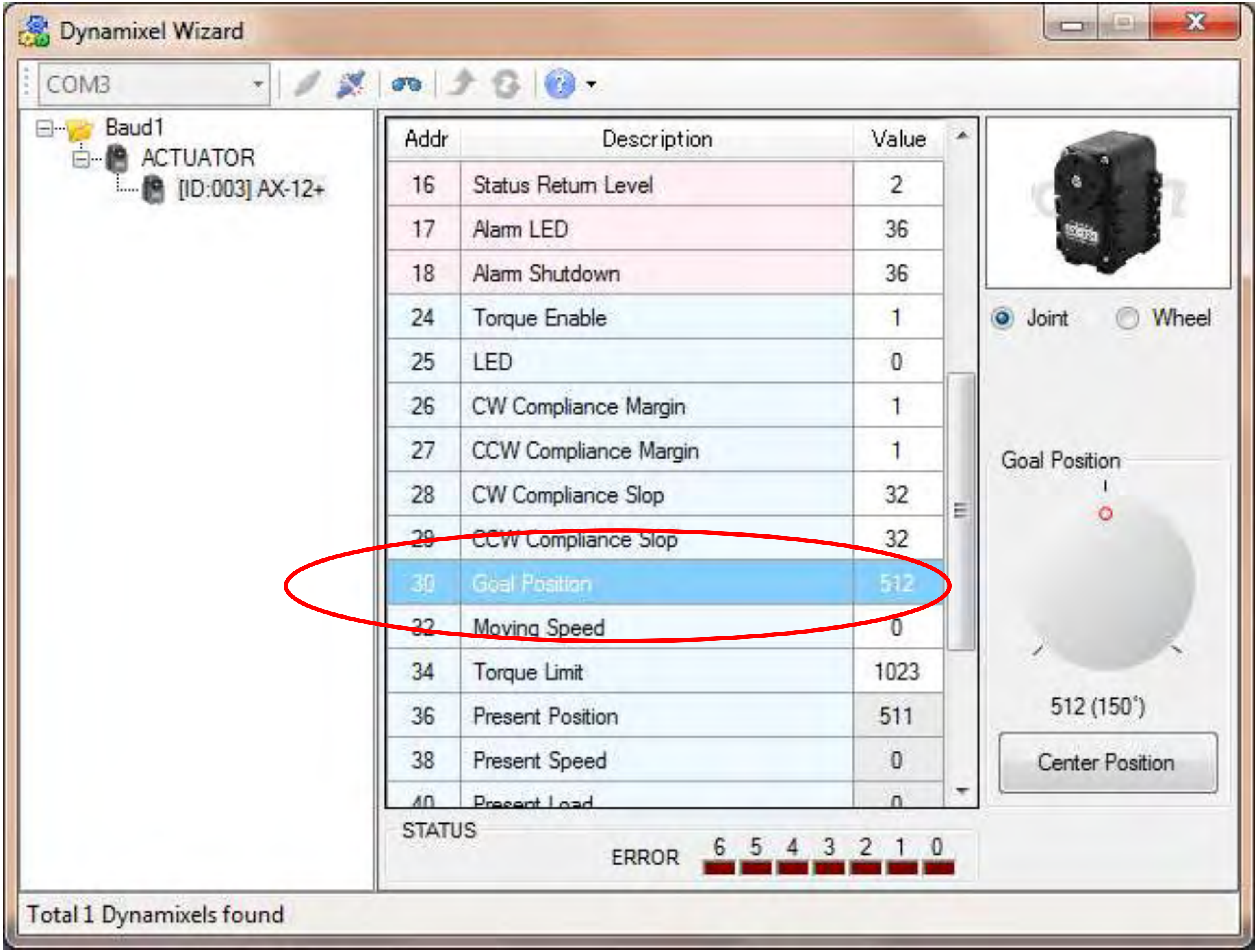
[ID:003] AX-12+

	bps	Search
▶ 1	1000000	☑
34	57142	☑

Basic Search

Search Complete

Total 1 Dynamixels found

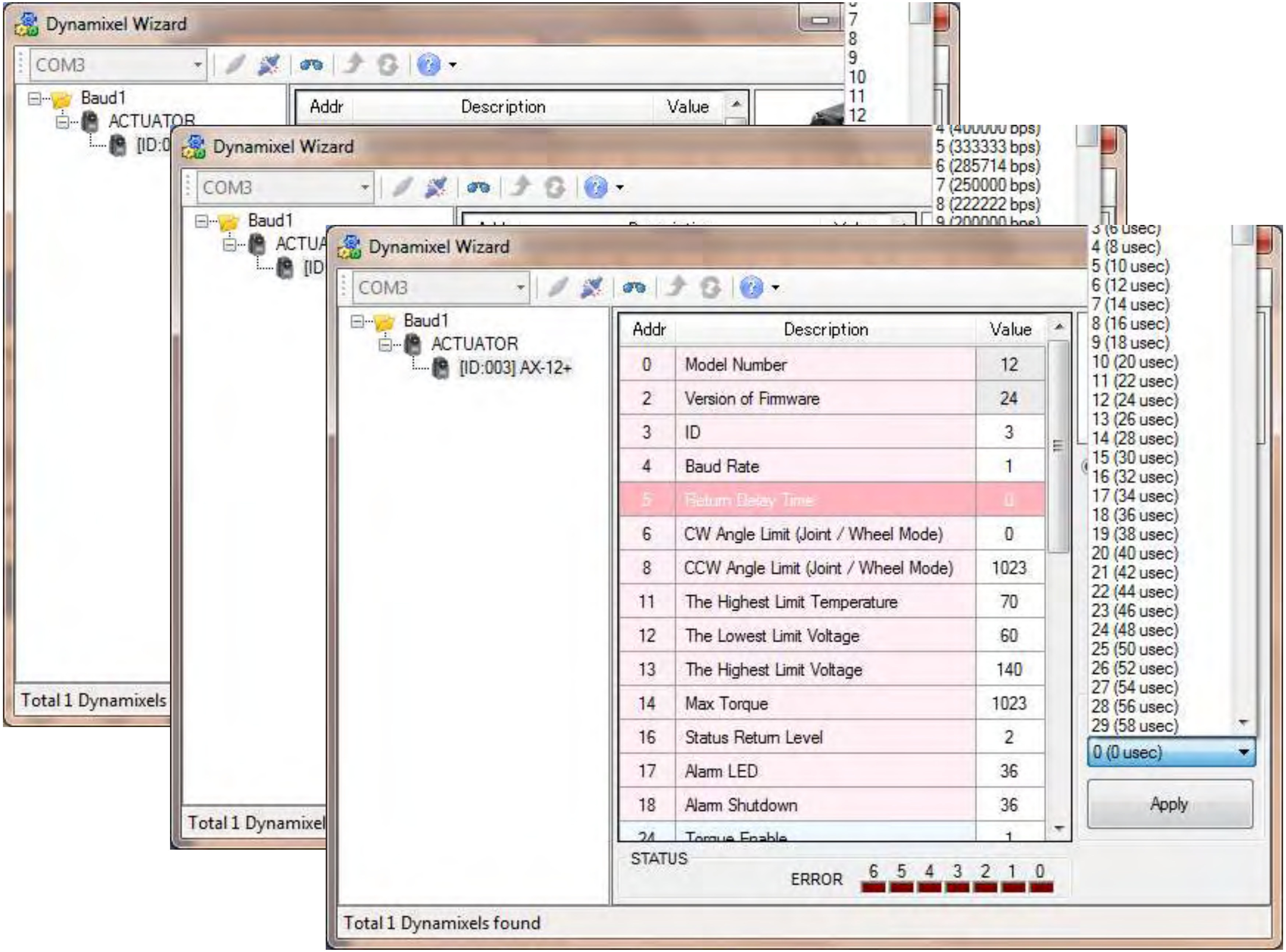


Addr	Description	Value
16	Status Return Level	2
17	Alarm LED	36
18	Alarm Shutdown	36
24	Torque Enable	1
25	LED	0
26	CW Compliance Margin	1
27	CCW Compliance Margin	1
28	CW Compliance Slop	32
29	CCW Compliance Slop	32
30	Goal Position	512
32	Moving Speed	0
34	Torque Limit	1023
36	Present Position	511
38	Present Speed	0
40	Present Load	0

● Joint ○ Wheel

Goal Position: 512 (150°)

Total 1 Dynamixels found

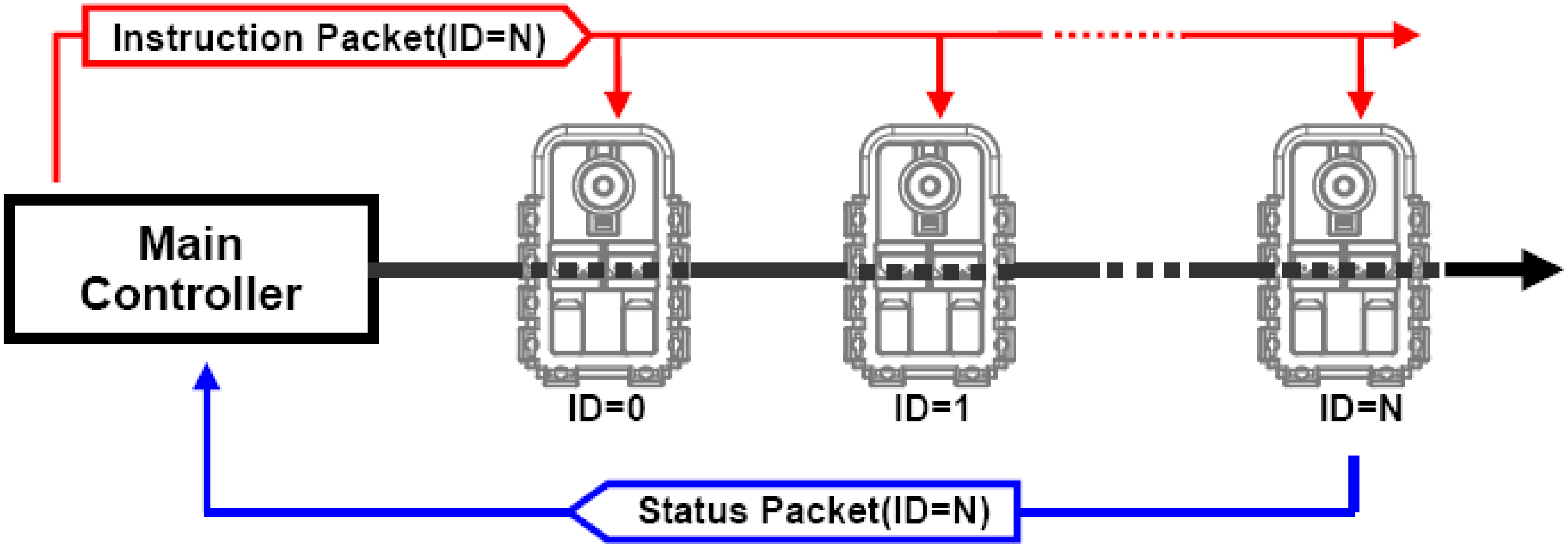


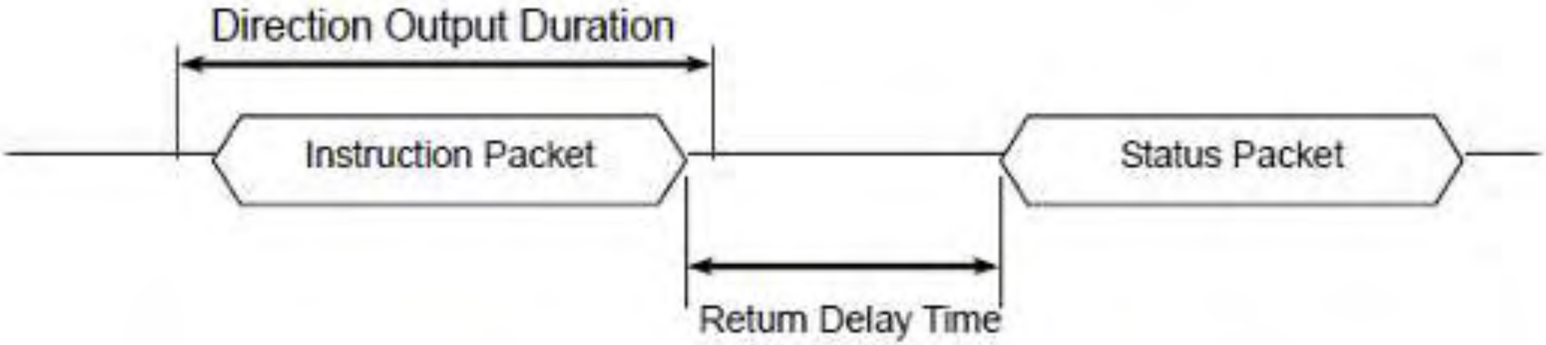
Addr	Description	Value
0	Model Number	12
2	Version of Firmware	24
3	ID	3
4	Baud Rate	1
5	Return Delay Time	0
6	CW Angle Limit (Joint / Wheel Mode)	0
8	CCW Angle Limit (Joint / Wheel Mode)	1023
11	The Highest Limit Temperature	70
12	The Lowest Limit Voltage	60
13	The Highest Limit Voltage	140
14	Max Torque	1023
16	Status Return Level	2
17	Alarm LED	36
18	Alarm Shutdown	36
24	Torque Enable	1

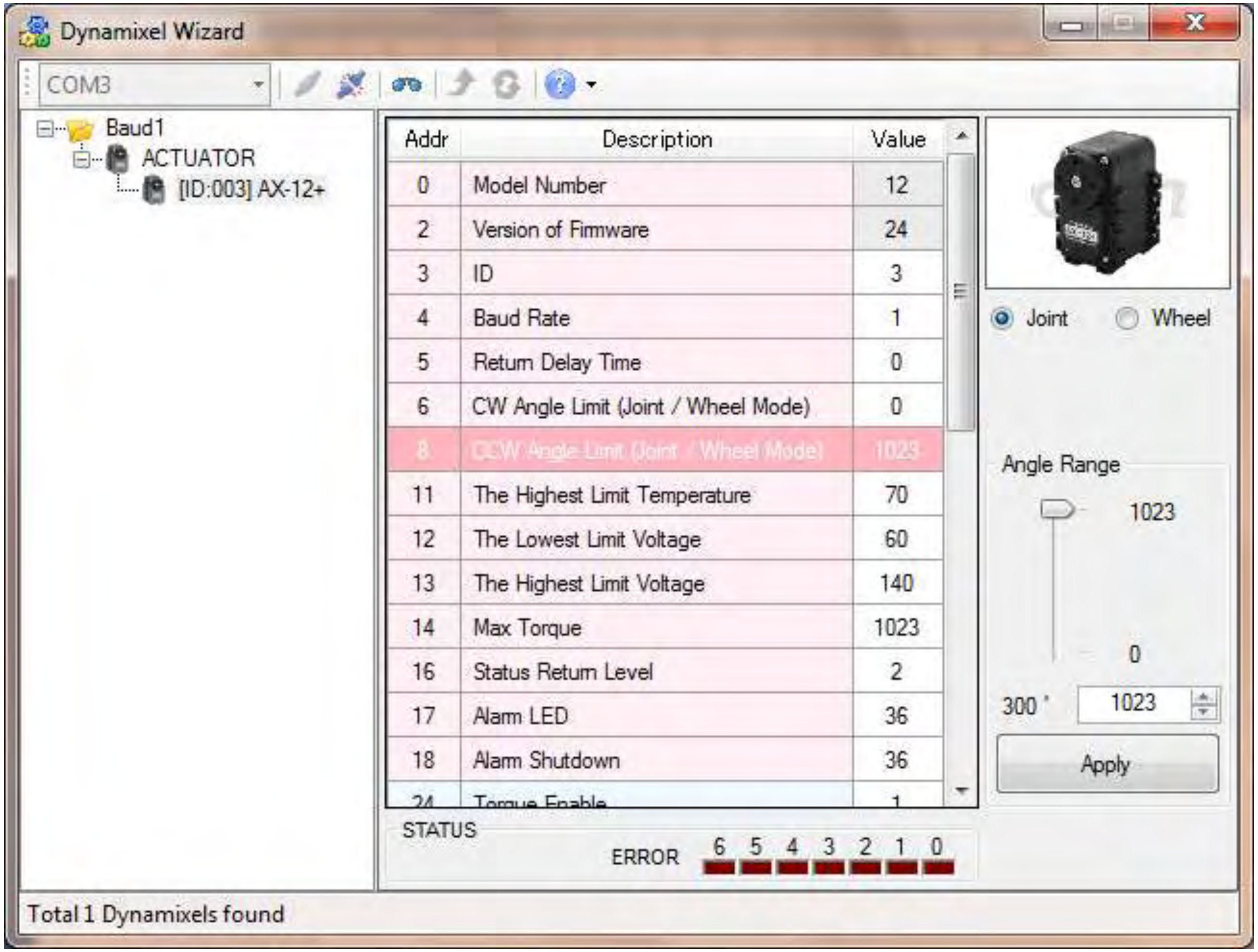
STATUS

ERROR 6 5 4 3 2 1 0

Total 1 Dynamixels found


Instruction Packet(ID=N)
Main Controller
ID=0
ID=1
ID=N
Status Packet(ID=N)


Direction Output Duration
Instruction Packet
Status Packet
Retum Delay Time



Addr	Description	Value
0	Model Number	12
2	Version of Firmware	24
3	ID	3
4	Baud Rate	1
5	Return Delay Time	0
6	CW Angle Limit (Joint / Wheel Mode)	0
8	CCW Angle Limit (Joint / Wheel Mode)	1023
11	The Highest Limit Temperature	70
12	The Lowest Limit Voltage	60
13	The Highest Limit Voltage	140
14	Max Torque	1023
16	Status Return Level	2
17	Alarm LED	36
18	Alarm Shutdown	36
24	Torque Enable	1

● Joint ○ Wheel

Angle Range

1023

0

300 ° 1023

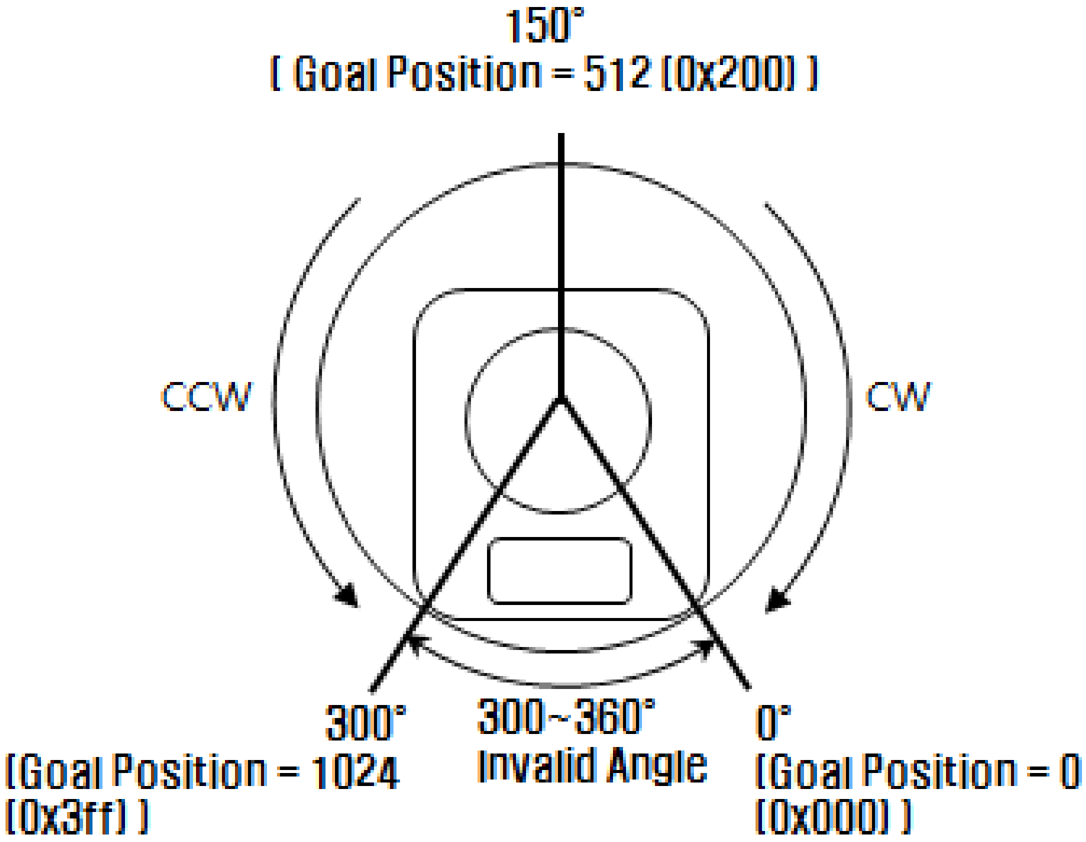
Total 1 Dynamixels found

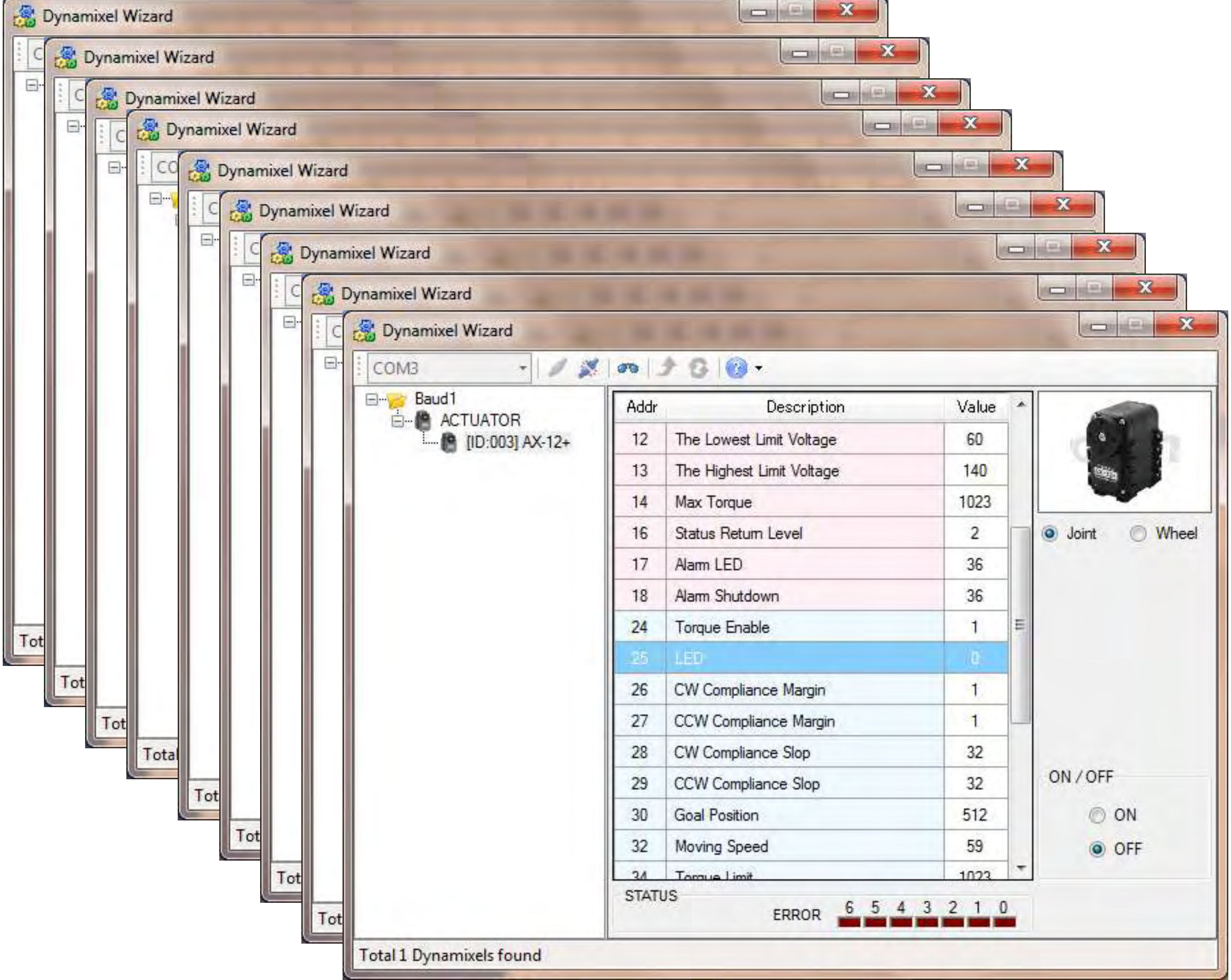
Goal Position

It is a position value of destination.

0 to 1023 (0x3FF) is available. The unit is 0.29 degree.

If Goal Position is out of the range, Angle Limit Error Bit (Bit1) of Status Packet is returned as ‘1’ and Alarm is triggered as set in Alarm LED/Shutdown.





Addr	Description	Value
12	The Lowest Limit Voltage	60
13	The Highest Limit Voltage	140
14	Max Torque	1023
16	Status Return Level	2
17	Alarm LED	36
18	Alarm Shutdown	36
24	Torque Enable	1
25	LED	0
26	CW Compliance Margin	1
27	CCW Compliance Margin	1
28	CW Compliance Slop	32
29	CCW Compliance Slop	32
30	Goal Position	512
32	Moving Speed	59
34	Torque Limit	1023

● Joint ○ Wheel

ON / OFF

○ ON

● OFF

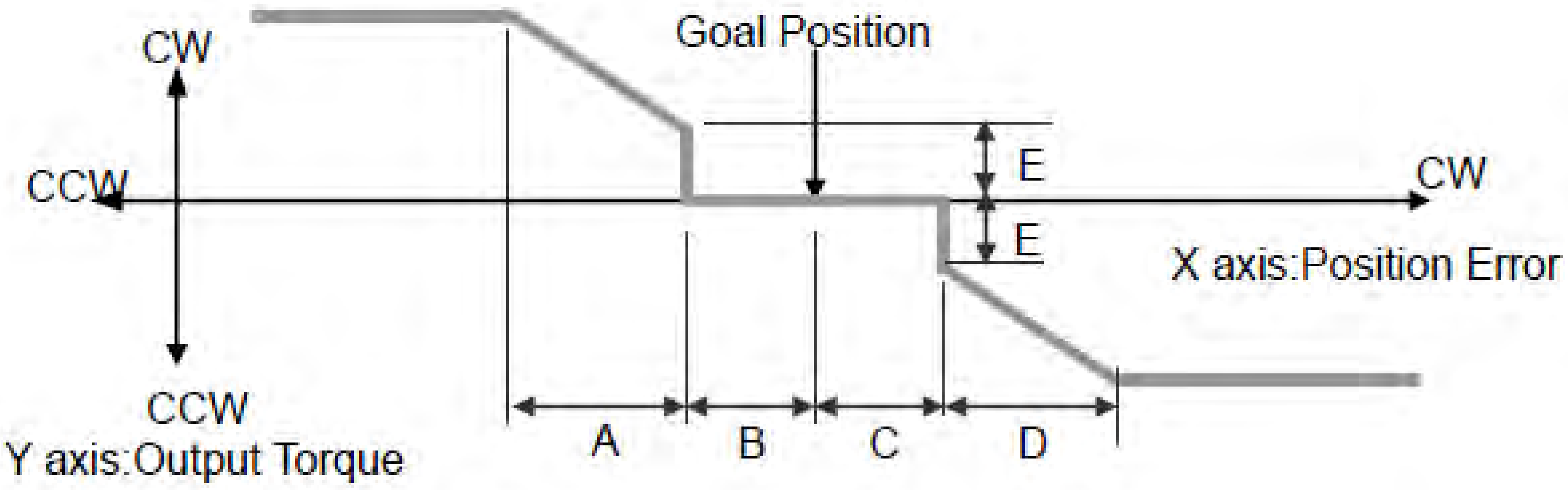
STATUS

ERROR 6 5 4 3 2 1 0

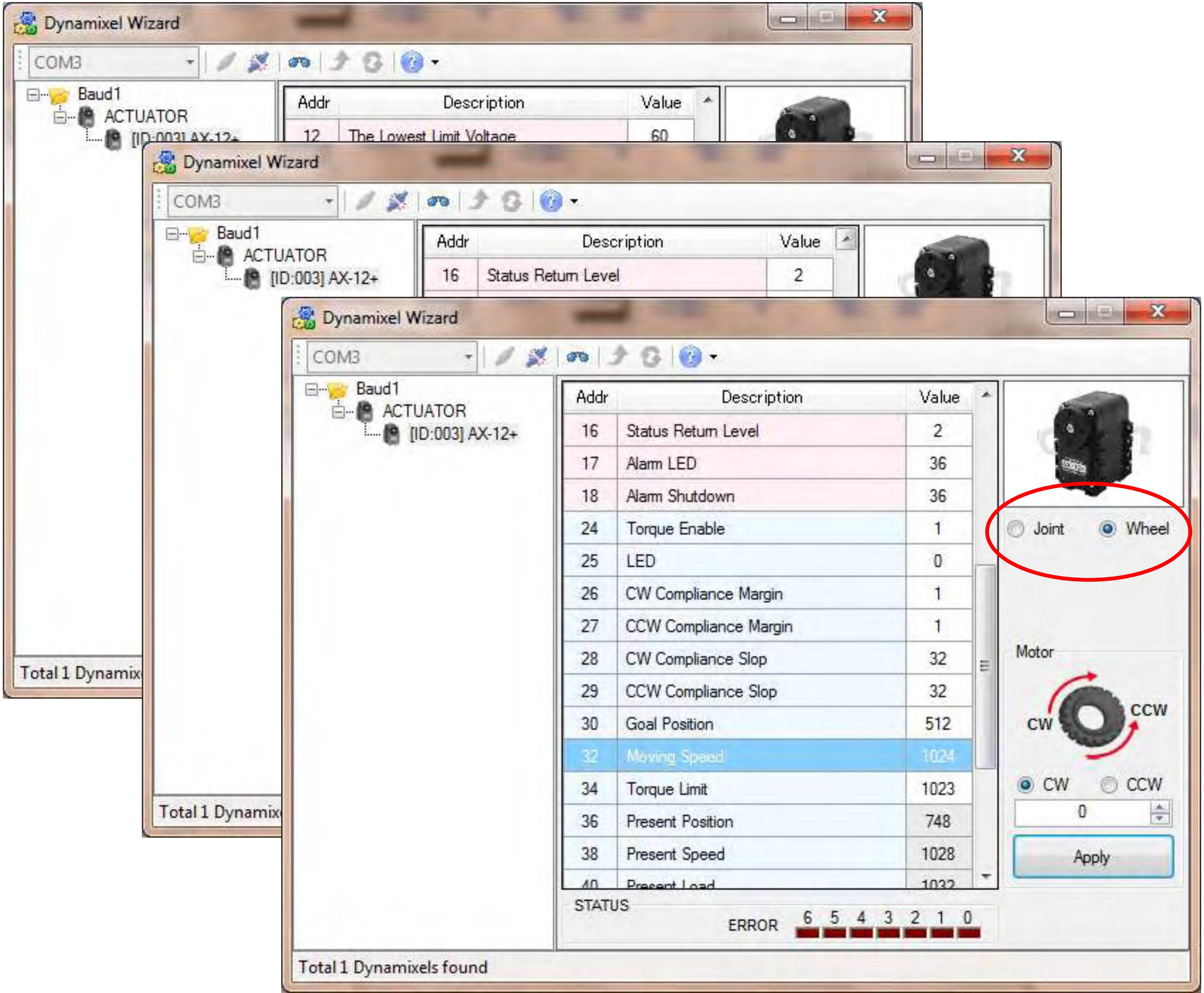
Total 1 Dynamixels found

Compliance

Compliance is to set the control flexibility of the motor.
The following diagram shows the relationship between the location and the motor.



A : CCW Compliance Slope(Address0x1D)
B : CCW Compliance Margin(Address0x1B)
C : CW Compliance Margin(Address0x1A)
D : CW Compliance Slope (Address0x1C)
E : Punch(Address0x30,31)



Dynamixel Wizard

COM3

Baud1
ACTUATOR
[ID:003] AX-12+

Addr	Description	Value
12	The Lowest Limit Voltage	60

Total 1 Dynamix

Dynamixel Wizard

COM3

Baud1
ACTUATOR
[ID:003] AX-12+

Addr	Description	Value
16	Status Return Level	2

Total 1 Dynamix

Dynamixel Wizard

COM3

Baud1
ACTUATOR
[ID:003] AX-12+

Addr	Description	Value
16	Status Return Level	2
17	Alarm LED	36
18	Alarm Shutdown	36
24	Torque Enable	1
25	LED	0
26	CW Compliance Margin	1
27	CCW Compliance Margin	1
28	CW Compliance Slop	32
29	CCW Compliance Slop	32
30	Goal Position	512
32	Moving Speed	1024
34	Torque Limit	1023
36	Present Position	748
38	Present Speed	1028
40	Present Load	1032

○ Joint ● Wheel

Motor

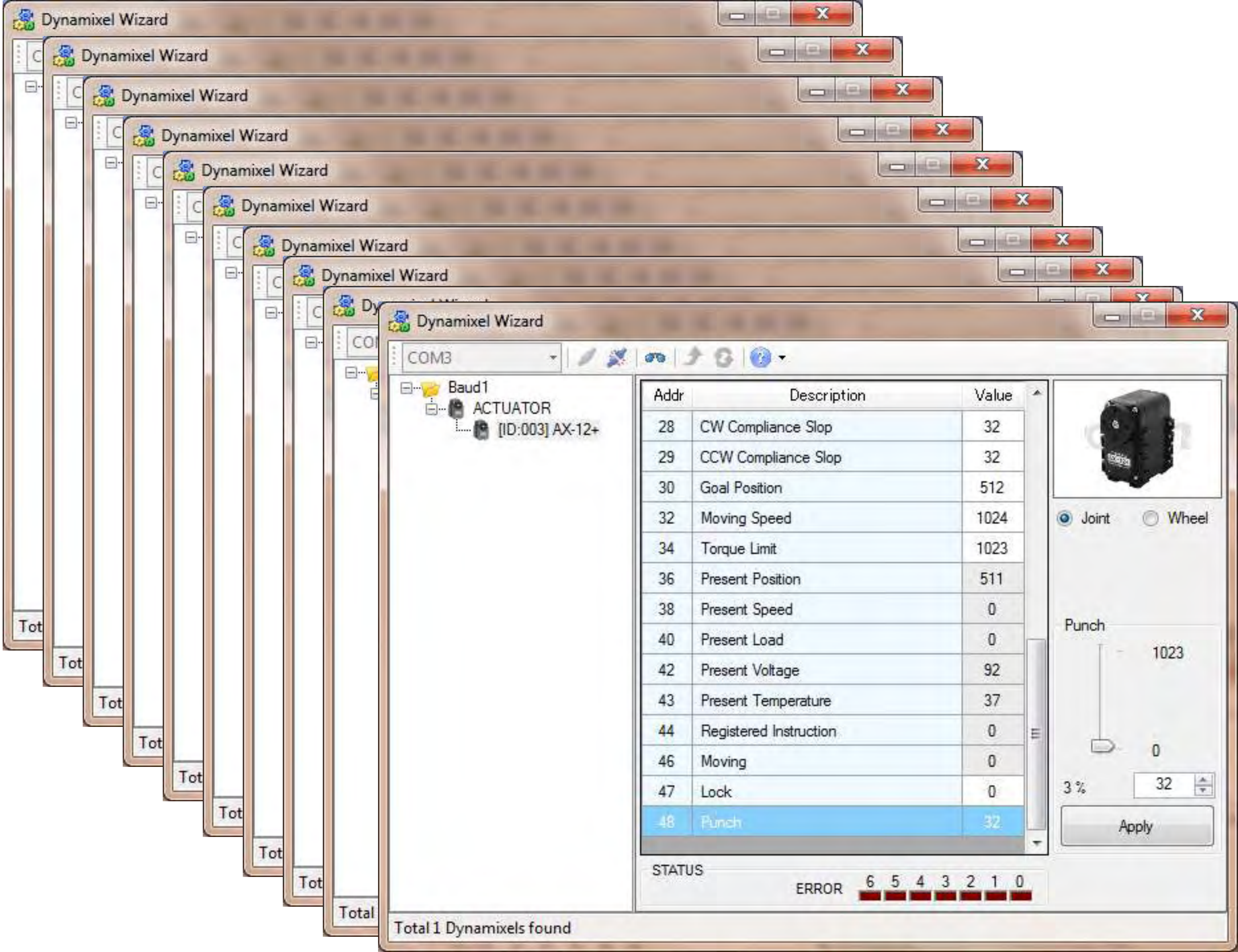
● CW ○ CCW

0

Apply

STATUS

Total 1 Dynamixels found



Addr	Description	Value
28	CW Compliance Slop	32
29	CCW Compliance Slop	32
30	Goal Position	512
32	Moving Speed	1024
34	Torque Limit	1023
36	Present Position	511
38	Present Speed	0
40	Present Load	0
42	Present Voltage	92
43	Present Temperature	37
44	Registered Instruction	0
46	Moving	0
47	Lock	0
48	Punch	32

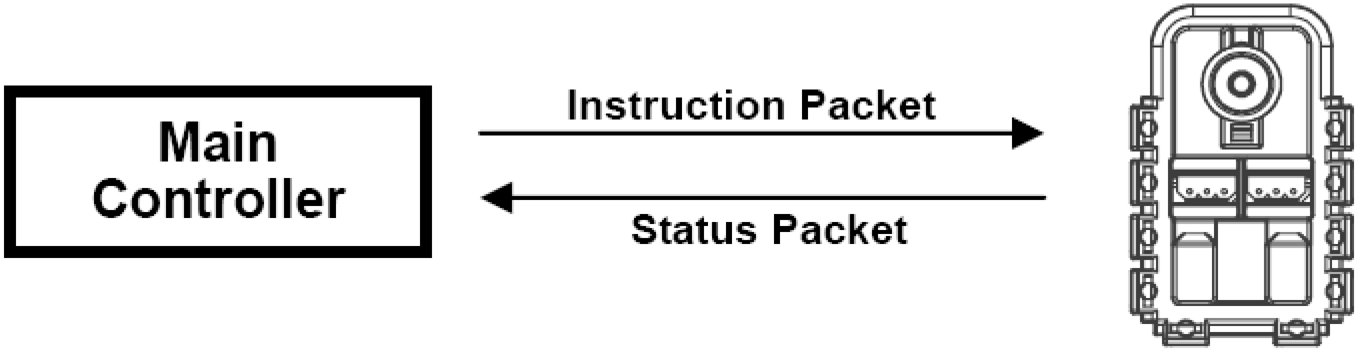
● Joint ○ Wheel

Punch: 1023 / 0

3 % 32

Total 1 Dynamixels found

4.- Conocer a fondo el AX-12



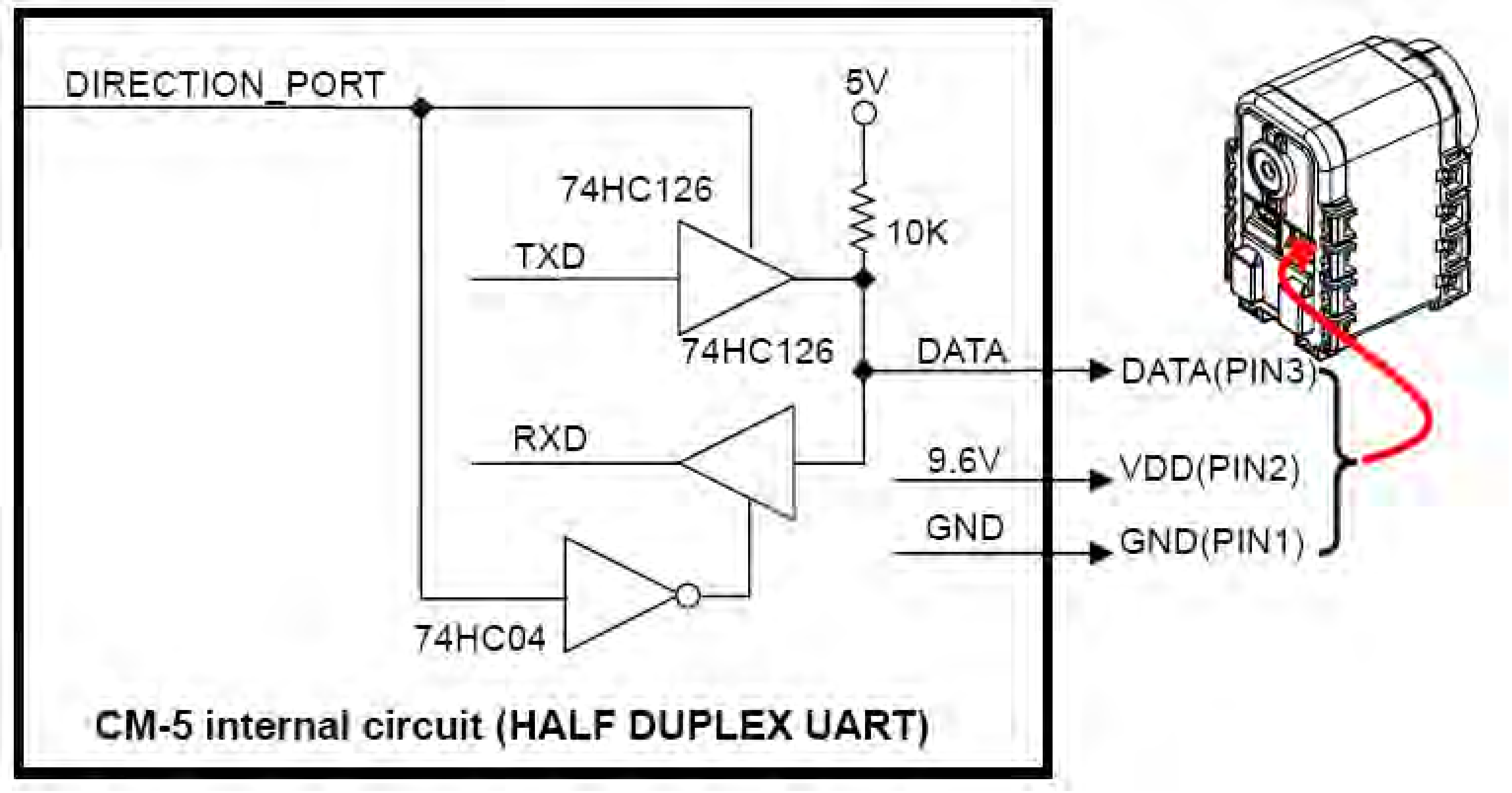
Main Controller

Instruction Packet

Status Packet

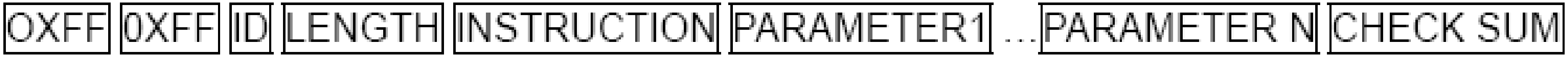
Connection to UART

To control the Dynamixel actuators, the main controller needs to convert its UART signals to the half duplex type.



Instruction Packet

The Instruction Packet is the packet sent by the main controller to the Dynamixel units to send commands. The structure of the Instruction Packet is as the following.



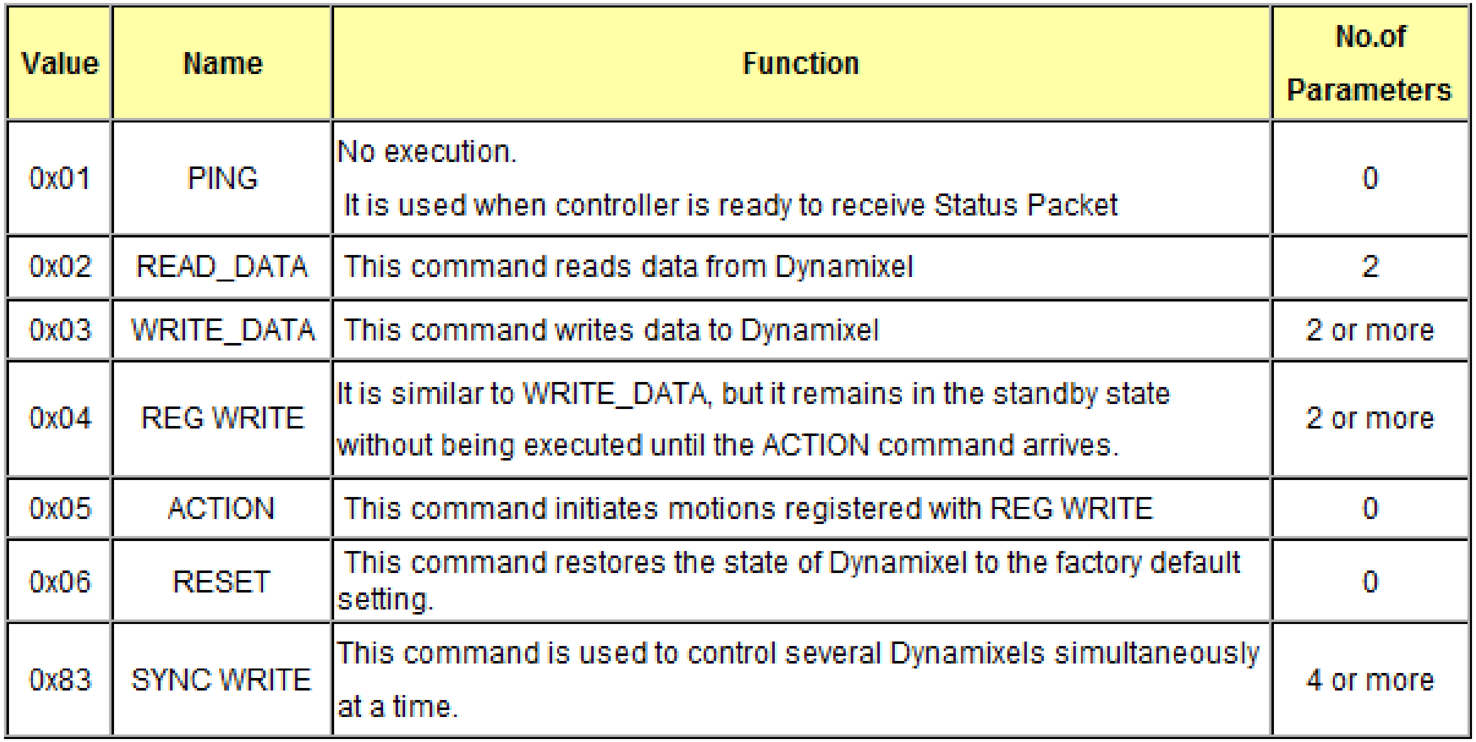
OXFF | 0XFF | ID | LENGTH | INSTRUCTION | PARAMETER1 | … | PARAMETER N | CHECK SUM

Start: The first two bytes are always 255. They indicate the start of an incoming packet.

ID: Only 0-254 may be used. A value of 254 indicates a broadcast to all devices on the network.

Length: If a single instruction is used with no data then the length value will be 2. If one data parameter is used then the length value will be 3.

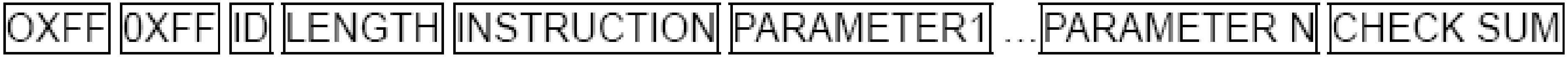
Instruction:



Value	Name	Function	No.of Parameters
0x01	PING	No execution. It is used when controller is ready to receive Status Packet	0
0x02	READ_DATA	This command reads data from Dynamixel	2
0x03	WRITE_DATA	This command writes data to Dynamixel	2 or more
0x04	REG WRITE	It is similar to WRITE_DATA, but it remains in the standby state without being executed until the ACTION command arrives.	2 or more
0x05	ACTION	This command initiates motions registered with REG WRITE	0
0x06	RESET	This command restores the state of Dynamixel to the factory default setting.	0
0x83	SYNC WRITE	This command is used to control several Dynamixels simultaneously at a time.	4 or more

Instruction Packet

The Instruction Packet is the packet sent by the main controller to the Dynamixel units to send commands. The structure of the Instruction Packet is as the following.



OXFF	0XFF	ID	LENGTH	INSTRUCTION	PARAMETER1	...	PARAMETER N	CHECK SUM

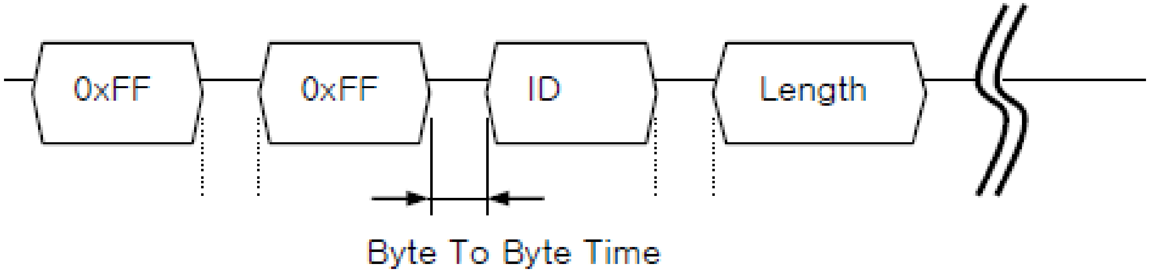
Parameter1-N : This is the optional data parameter(s). Some instructions like Reset and Ping do not have data parameters. Others have 1 or more data parameters.
Check Sum: The computation method for the 'Check Sum' is as the following.

$$\text{Check Sum} = \sim(\text{ID} + \text{Length} + \text{Instruction} + \text{Parameter1} + \dots \text{Parameter N})$$

If the calculated value is larger than 255, the lower byte is defined as the checksum value.
~ represents the NOT logic operation.

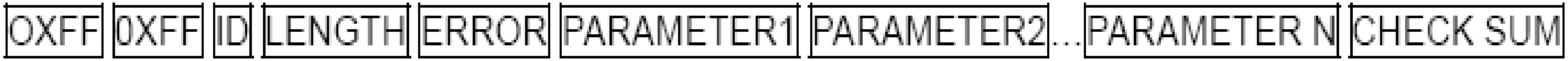
Byte to Byte Time

The delay time between bytes when sending an instruction packet. If the delay time is over 100ms, then the Dynamixel actuator recognizes this as a communication problem and waits for the next header (0xff 0xff) of a packet again.



Status Packet(Return Packet)

The Status Packet is the response packet from the Dynamixel units to the Main Controller after receiving an instruction packet. The structure of the status packet is as the following.



OXFF	0XFF	ID	LENGTH	ERROR	PARAMETER1	PARAMETER2	…	PARAMETER N	CHECK SUM

Start: The first two bytes are always 255. They indicate the start of an incoming packet.
ID: Will contain the ID of the device indicated by the packet just received.
Length: If a single instruction is used with no data then the length value will be 2.
Error: A single byte indicating the status of the last command packet.
PARAMETER0…N: Used if additional information is needed.
CHECK SUM: The computation method for the ‘Check Sum’ is as the following.

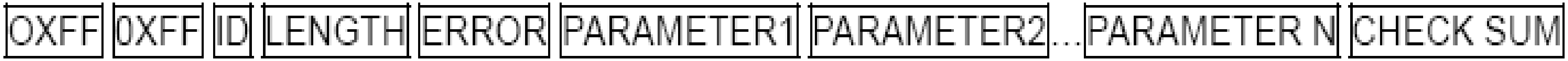
Check Sum = ~ (ID + Length + Instruction + Parameter1 + ... Parameter N)

If the calculated value is larger than 255, the lower byte is defined as the checksum value.

~ represents the NOT logic operation.

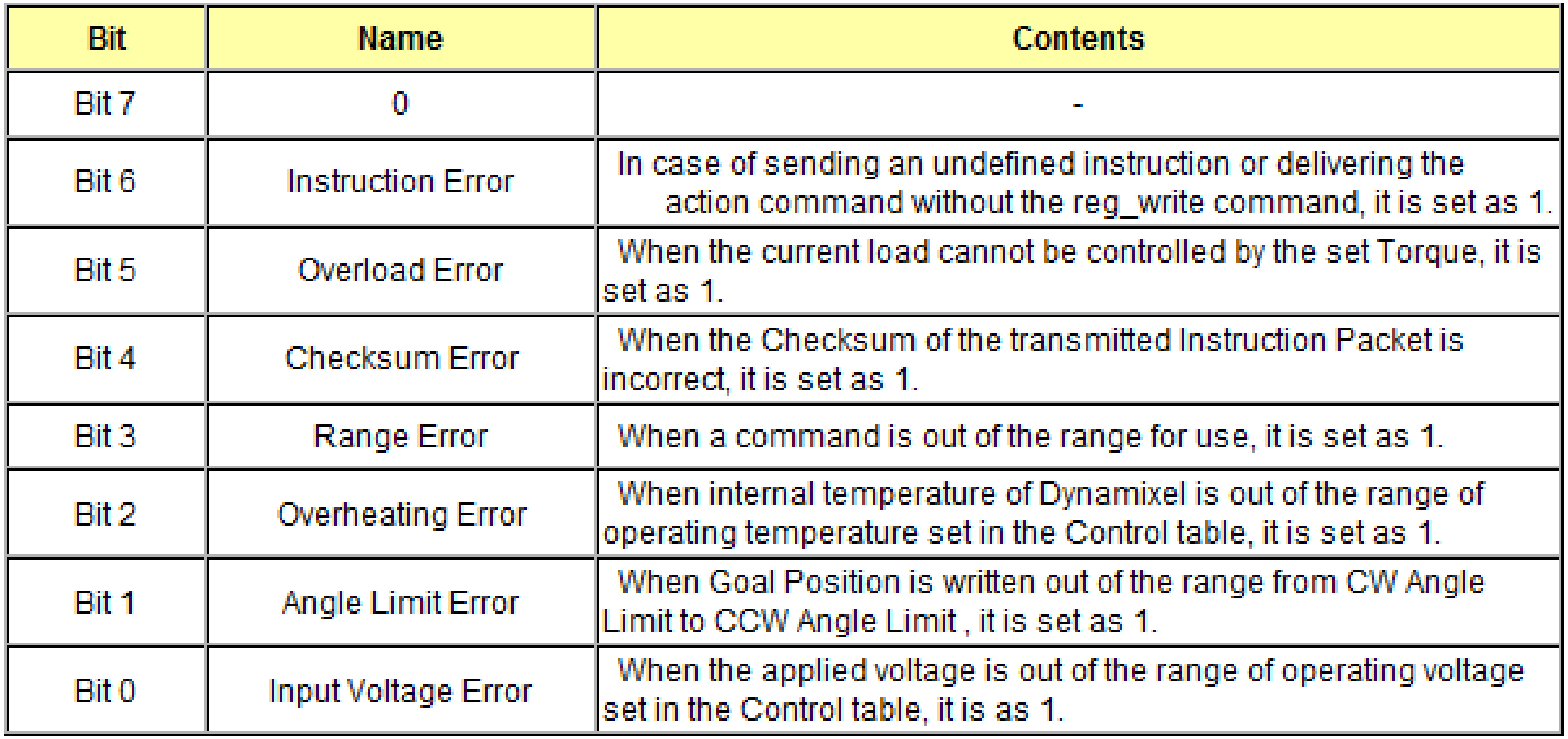
Status Packet(Return Packet)

The Status Packet is the response packet from the Dynamixel units to the Main Controller after receiving an instruction packet. The structure of the status packet is as the following.

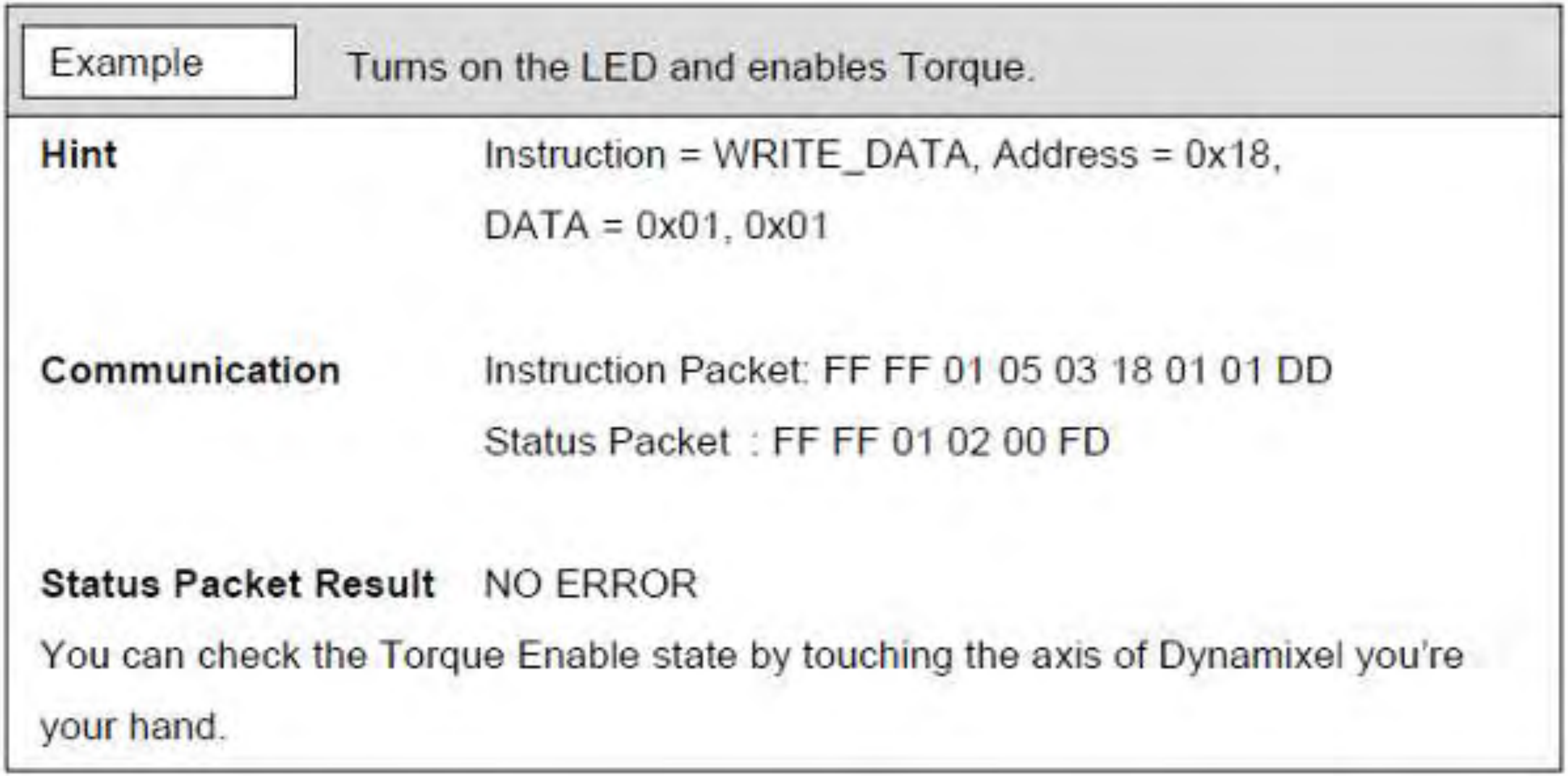


OXFF 0XFF ID LENGTH ERROR PARAMETER1 PARAMETER2…PARAMETER N CHECK SUM

Error:



Bit	Name	Contents
Bit 7	0	-
Bit 6	Instruction Error	In case of sending an undefined instruction or delivering the action command without the reg_write command, it is set as 1.
Bit 5	Overload Error	When the current load cannot be controlled by the set Torque, it is set as 1.
Bit 4	Checksum Error	When the Checksum of the transmitted Instruction Packet is incorrect, it is set as 1.
Bit 3	Range Error	When a command is out of the range for use, it is set as 1.
Bit 2	Overheating Error	When internal temperature of Dynamixel is out of the range of operating temperature set in the Control table, it is set as 1.
Bit 1	Angle Limit Error	When Goal Position is written out of the range from CW Angle Limit to CCW Angle Limit , it is set as 1.
Bit 0	Input Voltage Error	When the applied voltage is out of the range of operating voltage set in the Control table, it is as 1.



Example	Turns on the LED and enables Torque.

Hint	Instruction = WRITE_DATA, Address = 0x18,
	DATA = 0x01, 0x01

Communication	Instruction Packet: FF FF 01 05 03 18 01 01 DD Status Packet : FF FF 01 02 00 FD
Status Packet Result	NO ERROR

You can check the Torque Enable state by touching the axis of Dynamixel you're your hand.

Area	Address (Hexadecimal)	Name	Description	Access	Initial Value (Hexadecimal)
EEPROM	0 (0X00)	Model Number(L)	Lowest byte of model number	R	12 (0X0C)
	1 (0X01)	Model Number(H)	Highest byte of model number	R	0 (0X00)
	2 (0X02)	Version of Firmware	Information on the version of firmware	R	-
	3 (0X03)	ID	ID of Dynamixel	RW	1 (0X01)
	4 (0X04)	Baud Rate	Baud Rate of Dynamixel	RW	1 (0X01)
	5 (0X05)	Return Delay Time	Return Delay Time	RW	250 (0XFA)
	6 (0X06)	CW Angle Limit(L)	Lowest byte of clockwise Angle Limit	RW	0 (0X00)
	7 (0X07)	CW Angle Limit(H)	Highest byte of clockwise Angle Limit	RW	0 (0X00)
	8 (0X08)	CCW Angle Limit(L)	Lowest byte of counterclockwise Angle Limit	RW	255 (0XFF)
	9 (0X09)	CCW Angle Limit(H)	Highest byte of counterclockwise Angle Limit	RW	3 (0X03)
	11 (0X0B)	the Highest Limit Temperature	Internal Limit Temperature	RW	85 (0X55)
	12 (0X0C)	the Lowest Limit Voltage	Lowest Limit Voltage	RW	60 (0X3C)
	13 (0X0D)	the Highest Limit Voltage	Highest Limit Voltage	RW	190 (0XBE)
	14 (0X0E)	Max Torque(L)	Lowest byte of Max. Torque	RW	255 (0XFF)
	15 (0X0F)	Max Torque(H)	Highest byte of Max. Torque	RW	3 (0X03)
	16 (0X10)	Status Return Level	Status Return Level	RW	2 (0X02)
	17 (0X11)	Alarm LED	LED for Alarm	RW	36(0x24)
	18 (0X12)	Alarm Shutdown	Shutdown for Alarm	RW	36(0x24)

RAM	24 (0X18)	Torque Enable	Torque On/Off	RW	0 (0X00)
	25 (0X19)	LED	LED On/Off	RW	0 (0X00)
	26 (0X1A)	CW Compliance Margin	CW Compliance margin	RW	0 (0X00)
	27 (0X1B)	CCW Compliance Margin	CCW Compliance margin	RW	0 (0X00)
	28 (0X1C)	CW Compliance Slope	CW Compliance slope	RW	32 (0X20)
	29 (0X1D)	CCW Compliance Slope	CCW Compliance slope	RW	32 (0X20)
	30 (0X1E)	Goal Position(L)	Lowest byte of Goal Position	RW	-
	31 (0X1F)	Goal Position(H)	Highest byte of Goal Position	RW	-
	32 (0X20)	Moving Speed(L)	Lowest byte of Moving Speed	RW	-
	33 (0X21)	Moving Speed(H)	Highest byte of Moving Speed	RW	-
	34 (0X22)	Torque Limit(L)	Lowest byte of Torque Limit	RW	ADD14
	35 (0X23)	Torque Limit(H)	Highest byte of Torque Limit	RW	ADD15
	36 (0X24)	Present Position(L)	Lowest byte of Current Position	R	-
	37 (0X25)	Present Position(H)	Highest byte of Current Position	R	-
	38 (0X26)	Present Speed(L)	Lowest byte of Current Speed	R	-
	39 (0X27)	Present Speed(H)	Highest byte of Current Speed	R	-
	40 (0X28)	Present Load(L)	Lowest byte of Current Load	R	-
	41 (0X29)	Present Load(H)	Highest byte of Current Load	R	-
	42 (0X2A)	Present Voltage	Current Voltage	R	-
	43 (0X2B)	Present Temperature	Current Temperature	R	-
	44 (0X2C)	Registered	Means if Instruction is registered	R	0 (0X00)
	46 (0X2E)	Moving	Means if there is any movement	R	0 (0X00)
	47 (0X2F)	Lock	Locking EEPROM	RW	0 (0X00)
	48 (0X30)	Punch(L)	Lowest byte of Punch	RW	32 (0X20)
	49 (0X31)	Punch(H)	Highest byte of Punch	RW	0 (0X00)

PING

Function Does not command any operations. Used for requesting a status packet or to check the existence of a Dynamixel actuator with a specific ID.
Length 0X02
Instruction 0X01
Parameter NONE

WRITE_DATA

Function To write data into the control table of the Dynamixel actuator
Length N+3 (N is the number of data to be written)
Instruction 0X03
Parameter1 Starting address of the location where the data is to be written
Parameter2 1st data to be written
Parameter3 2nd data to be written
Parameter N+1 Nth data to be written

READ_DATA

Function Read data from the control table of a Dynamixel actuator
Length 0X04
Instruction 0X02
Parameter1 Starting address of the location where the data is to be read
Parameter2 Length of the data to be read

REG_WRITE

Function The REG_WRITE instruction is similar to the WRITE_DATA instruction, but the execution timing is different. When the Instruction Packet is received the values are stored in the Buffer and the Write instruction is under a standby status. At this time, the Registered Instruction register (Address 0x2C) is set to 1. After the Action Instruction Packet is received, the registered Write instruction is finally executed.

Length N+3 (N is the number of data to be written)

Instruction 0X04

Parameter1 Starting address of the location where the data is to be written

Parameter2 1st data to be written

Parameter3 2nd data to be written

Parameter N+1 Nth data to be written

ACTION

Function Triggers the action registered by the REG_WRITE instruction

Length 0X02

Instruction 0X05

Parameter NONE

The ACTION instruction is useful when multiple Dynamixel actuators need to move simultaneously. When controlling multiple Dynamixel actuator units, slight time delays can occur between the 1st and last units to receive an instruction. The Dynamixel actuator handles this problem by using the ACTION instruction.

Broadcasting

The Broadcast ID (0XFE) is used when sending ACTION instructions to more than two Dynamixel actuators. Note that no packets are returned by this operation.

RESET

Function Changes the control table values of the Dynamixel actuator to the Factory Default Value settings
Length 0X02
Instruction 0X06
Parameter NONE

SYNC WRITE

Function Used for controlling many Dynamixel actuators at the same time. The communication time decreases by the Synch Write instruction since many instructions can be transmitted by a single instruction. However, you can use this instruction only when the lengths and addresses of the control table to be written to are the same. Also, the broadcasting ID needs to be used for transmitting.
ID 0XFE
Length (L + 1) * N + 4 (L: Data length for each Dynamixel actuator, N: The number of Dynamixel actuators)
Instruction 0X83
Parameter1 Starting address of the location where the data is to be written
Parameter2 The length of the data to be written (L)

Parameter2	The length of the data to be written (L)	
Parameter3	The ID of the 1st Dynamixel actuator	Data for the 1st Dynamixel actuator
Parameter4	The 1st data for the 1st Dynamixel actuator	
Parameter5	The 2nd data for the 1st Dynamixel actuator	
…		
Parameter L+3	The Lth data for the 1st Dynamixel actuator	
Parameter L+4	The ID of the 2nd Dynamixel actuator	Data for the 2nd Dynamixel actuator
Parameter L+5	The 1st data for the 2nd Dynamixel actuator	
Parameter L+6	The 2nd data for the 2nd Dynamixel actuator	
…		
Parameter 2L+4	The Lth data for the 2nd Dynamixel actuator	
….		

5.- Usar TERMITE con AX-12

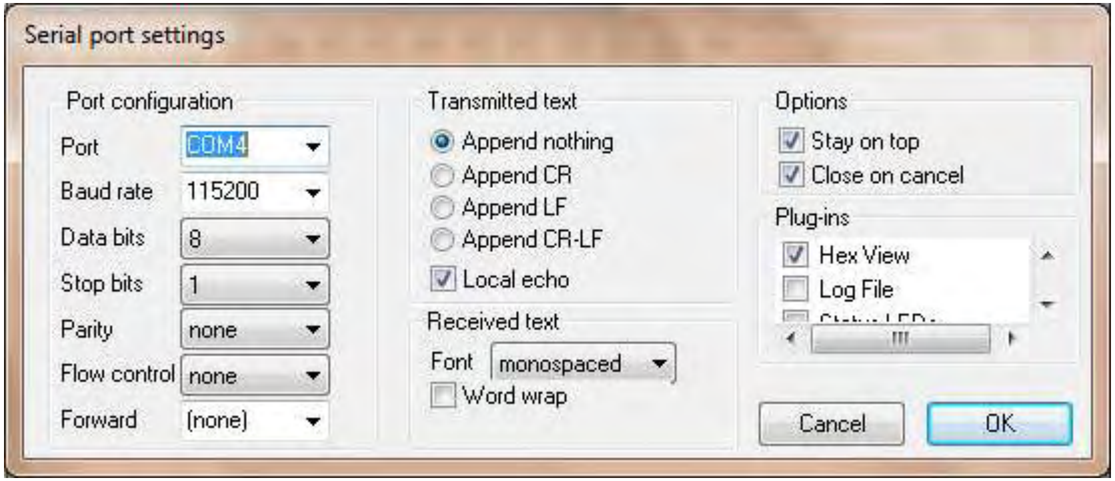
Termite: a simple RS232 terminal

Termite version 2.4 - complete setup (214 kiB)

Ingresar a la dirección, bajar el programa e instalarlo:

http://www.compuphase.com/software_termite.htm

- Asegurarse que el AX-12 se comunica a 115100bps
- Ejecutar TERMITE
- Configurar la comunicación según se muestra:



Serial port settings

Port configuration

Port: COM4
Baud rate: 115200
Data bits: 8
Stop bits: 1
Parity: none
Flow control: none
Forward: (none)

Transmitted text

● Append nothing
○ Append CR
○ Append LF
○ Append CR-LF
☑ Local echo

Received text

Font: monospaced
□ Word wrap

Options

☑ Stay on top
☑ Close on cancel

Plug-ins

☑ Hex View
□ Log File

- FF FF 03 05 03 1E 00 00 D6 //POSICION CERO

Escribir en la terminal: 0xffff0305031e0000d6

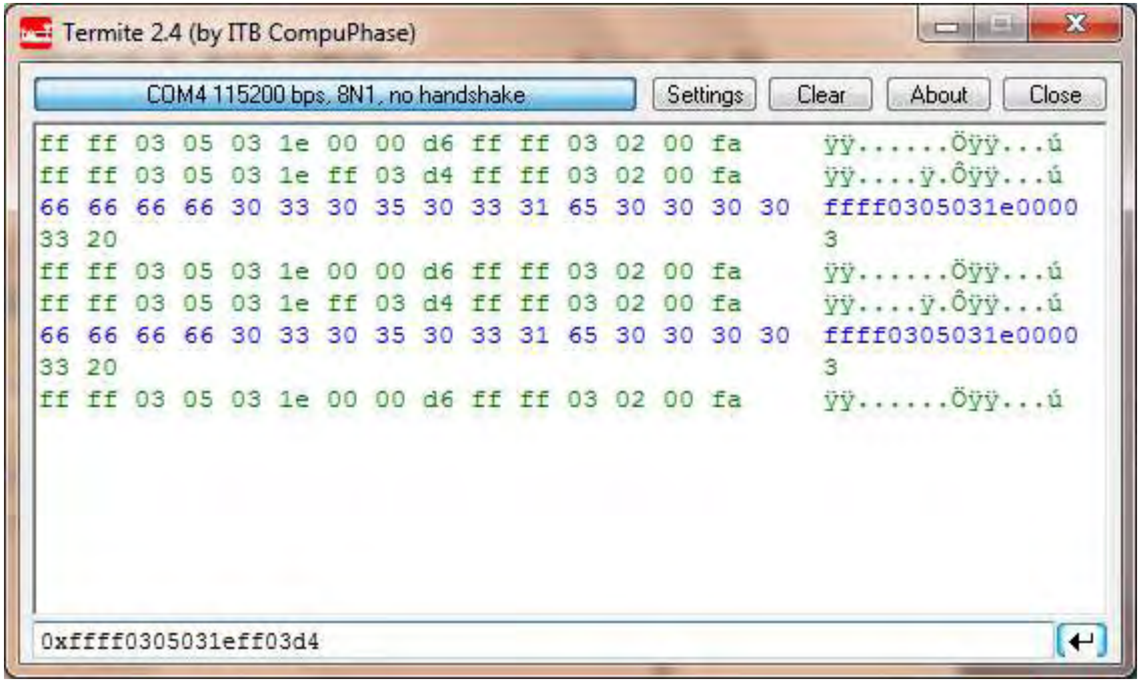
- FF FF 03 05 03 1E FF 03 D4 //POSICION 1023

Escribir en la terminal: 0xffff0305031eff03d4

- FF FF 00 04 02 2A 02 CD   //VER VOLTAJE Y TEMPERATURA

- FF FF 00 04 00 5C 23 7C   //VALORES OBTENIDOS DE VOLTAJE Y TEMPERATURA

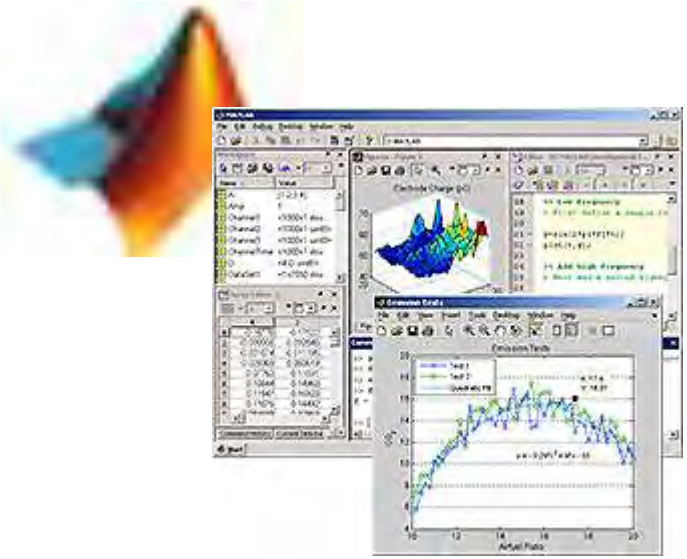
- FF FF 03 04 00 58 22 7E


Termite 2.4 (by ITB CompuPhase)
COM4 115200 bps, 8N1, no handshake
Settings
Clear
About
Close

```
ff ff 03 05 03 1e 00 00 d6 ff ff 03 02 00 fa       ÿÿ......Öÿÿ...ú
ff ff 03 05 03 1e ff 03 d4 ff ff 03 02 00 fa       ÿÿ....ÿ.Ôÿÿ...ú
66 66 66 66 30 33 30 35 30 33 31 65 30 30 30 30    ffff0305031e0000
33 20                                              3
ff ff 03 05 03 1e 00 00 d6 ff ff 03 02 00 fa       ÿÿ......Öÿÿ...ú
ff ff 03 05 03 1e ff 03 d4 ff ff 03 02 00 fa       ÿÿ....ÿ.Ôÿÿ...ú
66 66 66 66 30 33 30 35 30 33 31 65 30 30 30 30    ffff0305031e0000
33 20                                              3
ff ff 03 05 03 1e 00 00 d6 ff ff 03 02 00 fa       ÿÿ......Öÿÿ...ú
```

```
0xffff0305031eff03d4
```

6.- Usar el AX-12 con MATLAB



- Direct serial communication
- API interfase

USANDO EL PUERTO SERIAL CON MATLAB 7

Configurar el puerto serial

Para configurar el puerto serial, antes de enviar o recibir data, se debe inicializar el puerto. A continuación se presentan dos formas de hacerlo, una corta y otra con especificación de parámetros.

Forma corta

```
  PS=serial('COM4','BaudRate', 1000000,'Parity', 'none','DataBits',8,'StopBits', 1);
  fopen(PS);
```

Forma con especificación de parámetros

```
  PS=serial('COM1');
  set(PS,'Baudrate',9600);      % se configura la velocidad a 9600 Baudios
  set(PS,'StopBits',1);         % se configura bit de parada a uno
  set(PS,'DataBits',8);         % se configura que el dato es de 8bits, debe estar entre 5 y 8
  set(PS,'Parity','none');      % se configura sin paridad
  set(PS,'Terminator','CR/LF'); % Caracter con que finaliza el envío
  set(PS,'OutputBufferSize',1); % Número de bytes a enviar
  set(PS,'InputBufferSize' ,1); % Número de bytes a recibir
  set(PS,'Timeout',5);          % tiempo de espera en segundos antes de abortar
  fopen(PS);
```

Leer datos del puerto serial

Para leer datos binarios se usa la instrucción:

```
variable = fread(PS,1,'uint8');
```

Esto le dice al Matlab qué tipo de datos se interpretan los bits que se están recibiendo. Los tipos de datos pueden ser enteros sin signo, enteros con signo, de punto flotante, de 8 bits, de 16 bits, etc.

Cerrar el puerto

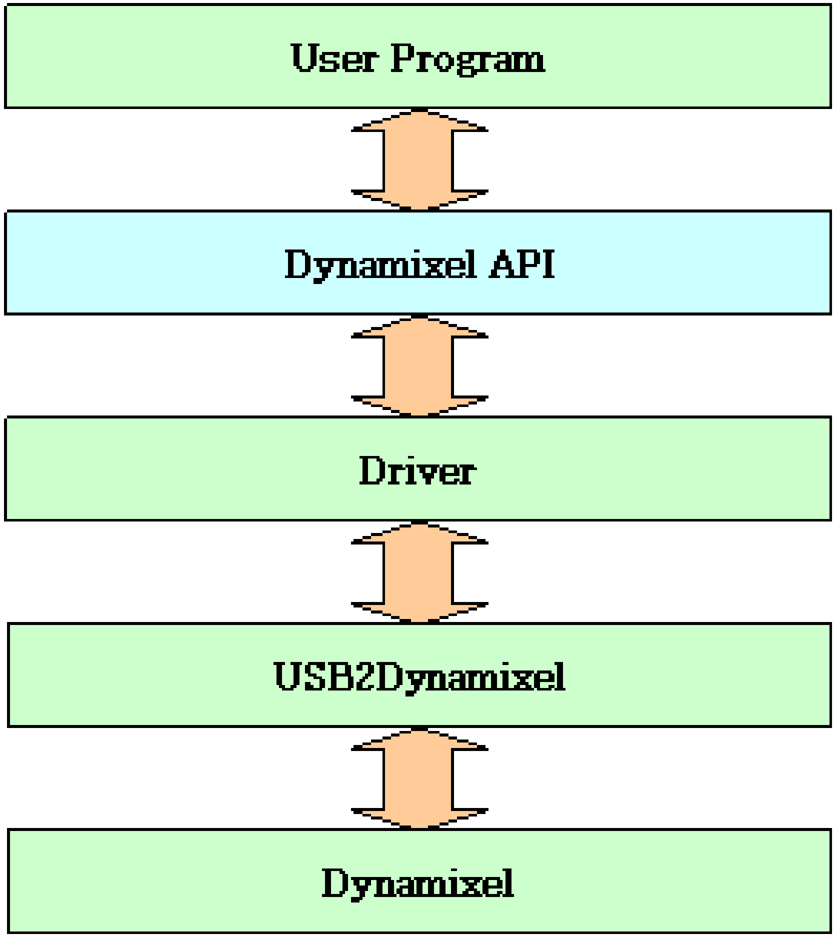
Una vez utilizado el puerto, éste se debe cerrar:

```
fclose(PS);
delete(PS);
clear PS;
```

```
function SerialLed()
    % Set the port paramenter
    s=serial('COM3', 'BaudRate', 1000000, 'Parity', ...
    'none', 'DataBits', 8, 'StopBits', 1);
    % open the port
    fopen(s);
    %-------- [LED 1 On ] ------------------------------
    %FF FF 01 04 03 19 01 DD
    a = [255, 255, 1, 4, 3, 25, 01, 221];
    %---------------------------------------------------
    %-------- [LED 1 Off ] -----------------------------
    %FF FF 01 04 03 19 00 DE
    %a = [255, 255, 1, 4, 3, 25, 00, 222];
    %---------------------------------------------------
    % display the values in a
    disp(a)
    % binary write
    fwrite(s, a);
    % Expecting a 6 byte status packet
    out=fread(s, 6);
    % Display status packet
    disp(out);
    % Clean up
    fclose(s);
    delete(s);
    clear s;
end
```

Dynamixel API

API stands for Application Programming Interface. This is a user programming solution. By using the Dynamixel API, you may readily read or deliver commands to the Dynamixels.



User Program

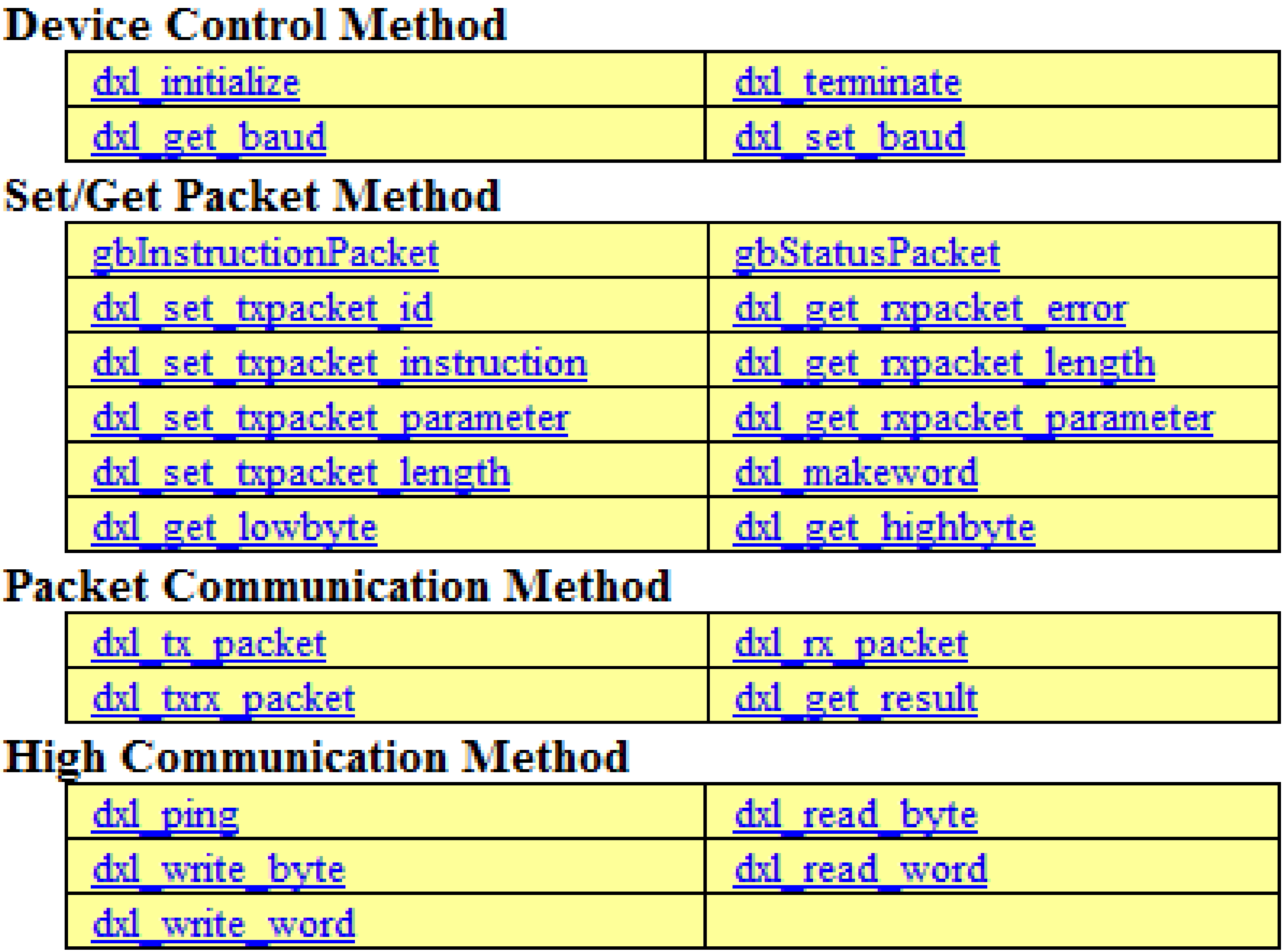
Dynamixel API

Driver

USB2Dynamixel

Dynamixel

Visual C++
Visual Basic
Visual C#
Java
Python
MATLAB
LABVIEW



Device Control Method

dxl_initialize	dxl_terminate
dxl_get_baud	dxl_set_baud

Set/Get Packet Method

gbInstructionPacket	gbStatusPacket
dxl_set_txpacket_id	dxl_get_rxpacket_error
dxl_set_txpacket_instruction	dxl_get_rxpacket_length
dxl_set_txpacket_parameter	dxl_get_rxpacket_parameter
dxl_set_txpacket_length	dxl_makeword
dxl_get_lowbyte	dxl_get_highbyte

Packet Communication Method

dxl_tx_packet	dxl_rx_packet
dxl_txrx_packet	dxl_get_result

High Communication Method

dxl_ping	dxl_read_byte
dxl_write_byte	dxl_read_word
dxl_write_word	

Register dynamixel.dll and dynamixel.h

You must register dynamixel.dll and dynamixel.h in Matlab if you want to use the API in your Matlab project.

SDK Source Location

C:\Program Files\ROBOTIS\USB2Dynamixel\bin\dynamixel.dll
C:\Program Files\ROBOTIS\USB2Dynamixel\import\dynamixel.h

MatLab Destination folder

C:\Program Files\MATLAB\R2009a\toolbox\matlab\winfun\win32

```
function example1(id)

loadlibrary('dynamixel','dynamixel.h');
libfunctions('dynamixel');
res = calllib('dynamixel','dxl_initialize');
if res == 1
    disp('Succeed to open USB2Dynamixel!');
    BaudRate=calllib('dynamixel','dxl_get_baud');
    % Indica velocidad de comunicación. El valor normal es 1 == 1Mbps.
    disp(sprintf('Baud Rate %d',BaudRate));

    calllib('dynamixel','dxl_ping',id);
    % Esta función sirve para saber si el servomotor id responde o no.
    StatusError=calllib('dynamixel','dxl_get_result');
    % Indica cual fue el resultado de la operación anterior.

    calllib('dynamixel','dxl_write_word',254,32,0);
    % El 254 significa que se envia la misma instrucción a todos los servomotores. El 32
    % significa escribiremos la velocidad. El 0significa la velocidad máxima. Ver manual.
    calllib('dynamixel','dxl_write_word',254,30,512);
    % El 254 significa que se envia la misma instrucción a todos los servomotores. El 30
    % significa escribiremos la posición. El 512 significa el valor deseado de posición.

    Plot_PresentPos=int32(calllib('dynamixel','dxl_read_word',id,36));

    calllib('dynamixel','dxl_terminate');
    % Con esta instruccion terinamos de usar el dispositivo USBDynamixel.
    disp('Succeed to close USB2Dynamixel!');
else
    disp('Failed to open USB2Dynamixel!');
end
unloadlibrary('dynamixel');
```