



# **MANUAL**

*Version 1.5.1*

*FOR THE INNCO.DLL v1.5.1*

*TO CONTROL THE*

## **INNCO DEVICE**

*(MAST, TURNTABLE AND TURNUNIT-TURNDEVICE)  
WITHOUT HARDWARELIMITS*

*innco systems GmbH*

*SCHWARZENFELD  
GERMANY*

*COPYRIGHT RESERVED  
ACCORDING TO  
GERMAN AND EU LAW*



1.1.COMMAND DESCRIPTION FOR THE INNCO.DLL.....	3
1.1.2.SELECT A DEVICE.....	4
1.2.1.THE NEW COMMAND FOR GET STATUS OF BUSY, POSITION AND POLARISATION LIKE AN GLOBALVARIABLE.....	5
1.2.2.MAST FUNCTIONS.....	6
1.2.3.TURNTABLE FUNCTION .....	9
1.2.3.TURN DEVICE FUNCTIONS (TURNUNIT).....	11
1.2.3.ANTENNA STAND FUNCTIONS .....	17
1.3.THE ERROR RETURN CODE.....	20
1.3.1.THE RETUN CODE OF THE STATUS COMANNDNS.....	21
1.4.F&Q .....	21
1.4.1.Operating method with multi-threading .....	21
1.4.2.Operating method without multi-thread .....	22
1.4.3.This is a option to use the innco.dll .....	23
1.4.4.This is an another option to use the innco.dll.....	24
1.5.WHAT IS THE DIFFRENT BETWEEN VERSION 1.2 AND 1.3.....	25
1.6. PROGRAMMING WITH THE INNCO.DLL.....	25



## 1.1.COMMAND DESCRIPTION FOR THE INNCO.DLL

### NOTE !

Because of fast return of the commands you must observe the following notes for an accurate function of our \*.dll.  
Please observe the RS232 is not so fast then a personal computer.  
You must set between the commands a little delay 1/50 second.

### NOTE !

Very seldom it is necessary ConnectMast( ) as well as ConnectTurntable( ) send over again about the device is not ready for receiving.  
Is the returncode not zero send the command after a delay of approximately 1/10 second Sleep(100) again.

### NOTE !

Between a set-command and a status-command there must be a little delay.  
The device needs 1/10 seceond delay or a Sleep(100) proceeding the set-command before  
the status commands reflect the action of the set-command.  
This applies to the following status commands:

Busyls( )  
IsMastMoving( )  
IsTurntableMoving( )



### 1.1.2.SELECT A DEVICE

Through the ConnectMast ( ) command e.g. for comport '1' is at this session the comport 1 with the joint device occupied from the calling thread until you send command the DisconnectMast( ) or ConnectTurntable() or ConnectMast( ) with a another comport number.

#### **With multi-threading.**

The best way to use the innco.dll, take a thread for every device. You can synchronous use every device ( the comport is whereas the session fixed at the device ) e.g. mast1 = com1 = thread01, mast2 = com2 = thread02, turntable = com3 = thread03. You can deallocate through e.g. the DisconnectMast( ) command the joint.

Every thread this joint port and device are absolute independent. The thread03 can't not use the port or the device from thread01. The thread02 can't not use the port or the device from thread03  
etc.

#### **Without multi-threading**

After the connect the port and the joint device through e.g. the CommandMast ( ) can you use all commands for the mast. If you change the device send e.g. the ConnectTurntable( ) with the new comport. Now you can use all turntable command . If you again change the device send e.g. the ConnectMast( ) with the comport. Now you can use all mast command. If you again change the device send e.g. the ConnectTurntable( ) with the comport. Now you can use all turntable command etc.



## 1.2.2.MAST FUNCTIONS

### ***int ConnectMast( int \*hMast, char Port);***

This command connect the mast

Port = the number of comport e.g. '1'  
hMast = the LUN number default = 1

return = error code  
call RetVal = ConnectMast( &hMast, Port)

### ***int SetMastHeight( int hMast,float Height, float Speed);***

This command sets a new position and speed for the mast and start movement for the selected device 1.2.

Height = is a float value between the minimum and maximum position  
The Height unit is metre  
Speed = is a float value between the minimum and maximum speed(cm/s)  
hMast = the LUN number default = 1

return = error code  
call RetVal = SetMatsHeight(hMast,Height,Speed)

### ***int GetMastHeight( int hMast, float \*Height );***

This command gets the current position of the mast for the selected device 1.2.

Height = is a float value of the current position  
The Height unit is metre  
hMast = the LUN number default = 1

return = error code  
call RetVal = GetMastHeight(hMast,&Height)

### ***int IsMastMoving( int hMast, int \*Flag );***

This command gets the current status of the mast for the selected device 1.2.

Flag = is an integer value as follows: moving = 1, static = 0  
hMast = the LUN number default = 1

return = error code  
call RetVal = IsMastMoving(hMast,&Flag)



\*\*\*\*\*

**NOTE!**

**This function can't provide a value from a device without hardware limits**

**This function is exclusive for device with hardware limit.**

***int GetMastHardwareLimits( int hMast, float \*Min, float \*Max) ;***

This command get return the hardware limits of position on the mast for the selected device 1.2.

Min = the lowest position of the mast (metre)  
Max = the highest position of the mast (metre)  
hMast = the LUN number default = 1

return = error code  
call RetVal = GetMastHardwareLimits(hMast,&Min,&Max)

\*\*\*\*\*

***int GetMastSpeedLimits( int hMast, float \*MinSpeed, float \*MaxSpeed) ;***

This command gets the speed limits on the mast for the selected device 1.2.

MinSpeed = the lowest speed of the mast (m/s)  
MaxSpeed = the highest speed of the mast (m/s)  
hMast = the LUN number default = 1

return = error code  
call RetVal = GetMastSpeedLimits(hMast,&MinSpeed,&MaxSpeed)

***int SetMastPolarisation( int hMast, int Polarisation) ;***

This command sets the antenna polarization to horizontal or vertical and starts positioning of the antenna on the mast for the selected device 1.2.

Polarisation = is a int value vertical = 1, horizontal = 0  
hMast = the LUN number default = 1

return = error code  
call RetVal = SetMastPolarisation(hMast,Polarisation)



***int GetMastPolarisation( int hMast, int \*Polarisation );***

This command gets the current polarisation of the antenna on the mast for the selected device 1.2.

Polarisation = is a int value vertical = 1, horizontal = 0  
the antenna is in a position between 1 degree and 89 degree = -1

hMast = the LUN number default = 1

return = error code  
call RetVal = GetMastPolarisation(hMast,&Polarisation)

***int MastStop( int hMast );***

This command stops the movement of the mast for the selected device 1.2.

hMast = the LUN number default = 1

return = error code  
call RetVal = MastStop(hMast)

***int DisconnectMast( int hMast);***

This command close the current comport for the selected device 1.2.

hMast = the LUN number default = 1  
or hTurntable

return = error code  
call RetVal = DisconnectMast(hMast)

***int ReferenceMast( int hMast );***

This command reference the mast for the selected device 1.2.  
!!! before start the process of measuring the mast must be referenced!!!

hMast = the LUN number default = 1

return = error code  
call RetVal = ReferenceMast(hMast)



### 1.2.3.TURNTABLE FUNCTION

***int ConnectTurntable( int \*hTurntable, char Port);***

This command connect the turntable.

Port = the number of comport e.g. '1'  
hTurntable = the LUN number default = 1

return = error code  
call RetVal = ConnectTurntable(&hTurntable,Port)

***int SetTurntablePosition( int hTurntable, float Angle, float Speed );***

This command sets a new position and speed for the turntable and starts movement of the selected device1.2.

Angle = is a float value between the minimum and maximum angle (degree)  
Speed = is a float value between the minimum and maximum speed (turns per second)  
hTurntable = the LUN number default = 1

return = error code  
call RetVal = SetTurntablePosition(hTurntable, Angle, Speed)

***int GetTurntablePosition(int hTurntable, float \*Angle);***

This command gets the current position of the turntable of the selected device1.2.

Angle = is a float value of the current angle (degrees)  
hTurntable = the LUN number default = 1

return = error code  
call RetVal = GetTurntablePosition(hTurntable,&Angle)

***int IsTurntableMoving( int hTurntable, int \*Flag );***

This command gets the current movement status of the turntable of the selected device1.2.

Flag = is a int value in move = 1, not in move = 0  
hTurntable = the LUN number default = 1

return = error code  
call RetVal = IsTurntableMoving(hTurntable,&Flag)



***int TurntableStop( int hTurntable);***

This command stops the movement of the turntable of the selected device1.2.

hTurntable = the LUN number default = 1  
return = error code  
call RetVal = TurntableStop(hTurntable)

\*\*\*\*\*

**NOTE!**

**This function can't provide a value from a device without hardware limits**

**This function is exclusive for device with hardware limit.**

***int GetTurntableHardwareLimits( int hTurntable, float \*MinAngle, float \*MaxAngle);***

This command returns the hardware limits of angle on the turntable of the selected device1.2.

Min = the lowest angle of the turntable (degrees)  
Max = the highest angle of the turntable (degrees)  
hTurntable = the LUN number default = 1  
return = error code  
call RetVal = GetTurntableHardwareLimits  
( hTurntable,&MinAngle,&MaxAngle)

\*\*\*\*\*

***int GetTurntableSpeedLimits( int hMast, float \*MinSpeed, float \*MaxSpeed);***

This command returns the speed limits on the turntable of the selected device1.2.

MinSpeed = the lowest speed of the turntable (turns pre second)  
MaxSpeed = the highest speed of the turntable (turns per second)  
hTurntable = the LUN number default = 1  
return = error code  
call RetVal = GetTurntableSpeedLimits(hMast,&MinSpeed,&MaxSpeed)

***int ReferenceTurntable( int hTurntable);***

This command references the turntable of the selected device1.2.

hTurntable = the LUN number default = 1  
return = error code  
call RetVal = ReferenceTurntable(hTurntable)



### 1.2.3.TURN DEVICE FUNCTIONS (TURNUNIT)

#### ***int ConnectTD( int \*hTD, char Port);***

This command connect the mast

Port               = the number of comport e.g. '1'  
hTD                = the LUN number default = 1  
  
return             = error code  
call   RetVal = ConnectTD( &hTD, Port)

#### ***int SetTDAxis1( int hTD, float Position, float Speed);***

This command sets a new position and speed for the mast and start movement for the selected device 1.2.

Position           = is a float value between the minimum and maximum position  
                    The position unit is degree  
Speed              = is a float value between the minimum and maximum speed(cm/s)  
                    **IS SPPED "0" SPEED IS MAX HARDWARELIMIT**  
hTD                = the LUN number default = 1  
  
return             = error code  
call   RetVal = SetTDAxis1(hTD,Position,Speed)

#### ***int SetTDAxis2( int hTD, float Position, float Speed);***

This command sets a new position and speed for the mast and start movement for the selected device 1.2.

Position           = is a float value between the minimum and maximum position  
                    The position unit is degree  
Speed              = is a float value between the minimum and maximum speed(cm/s)  
                    **IS SPPED "0" SPEED IS MAX HARDWARELIMIT**  
hTD                = the LUN number default = 1  
  
return             = error code  
call   RetVal = SetTDAxis2(hTD,Position,Speed)



***int GetTDAxis1( int hTD, float \*Position );***

This command gets the current position of the mast for the selected device 1.2.

Position       = is a float value of the current position  
                  The Height unit is metre  
hTD             = the LUN number default = 1  
  
return          = error code  
call     RetVal = GetTDAxis(hTD,&Position)

***int GetTDAxis2( int hTD, float \*Position );***

This command gets the current position of the mast for the selected device 1.2.

Position       = is a float value of the current position  
                  The Height unit is metre  
hTD             = the LUN number default = 1  
  
return          = error code  
call     RetVal = GetTDAxis(hTD,&Position)

***int IsTDMoving( int hTD, int \*Flag );***

This command gets the current status of the mast for the selected device 1.2.

Flag            = is an integer value as follows: moving = 1, static = 0  
hTD             = the LUN number default = 1  
  
return          = error code  
call     RetVal = IsTDMoving(hTD,&Flag)



### 1.2.3. Antenna Stand



\*\*\*\*\*

**NOTE!**

**This function can't provide a value from a device without hardware limits**

**This function is exclusive for device with hardware limit.**

*int GetTDHardwareLimits1( int hTD, float \*Min, float \*Max) ;*

This command get return the hardware limits of position on the mast for the selected device 1.2.

Min = the lowest position of the mast (metre)  
Max = the highest position of the mast (metre)  
hTD = the LUN number default = 1

return = error code  
call RetVal = GetTDHardwareLimits1(hTD,&Min,&Max)

*int GetTDHardwareLimits2( int hTD, float \*Min, float \*Max) ;*

This command get return the hardware limits of position on the mast for the selected device 1.2.

Min = the lowest position of the mast (metre)  
Max = the highest position of the mast (metre)  
hTD = the LUN number default = 1

return = error code  
call RetVal = GetTDHardwareLimits2(hTD,&Min,&Max)

\*\*\*\*\*



***int GetTDSpeedLimits1( int hTD, float \*MinSpeed, float \*MaxSpeed) ;***

This command gets the speed limits on the mast for the selected device 1.2.

MinSpeed                 = the lowest speed of the mast (cm/s)  
MaxSpeed                 = the highest speed of the mast (cm/s)  
hTD                       = the LUN number default = 1  
  
return                    = error code  
call     RetVal = GetTDSpeedLimits1(hTD,&MinSpeed,&MaxSpeed)

***int GetTDSpeedLimits2( int hTD, float \*MinSpeed, float \*MaxSpeed) ;***

This command gets the speed limits on the mast for the selected device 1.2.

MinSpeed                 = the lowest speed of the mast (cm/s)  
MaxSpeed                 = the highest speed of the mast (cm/s)  
hTD                       = the LUN number default = 1  
  
return                    = error code  
call     RetVal = GetTDSpeedLimits2(hTD,&MinSpeed,&MaxSpeed)

***int TDStop( int hTD);***

This command stops the movement of the mast for the selected device 1.2.

hTD                       = the LUN number default = 1  
  
return                    = error code  
call     RetVal = TDStop(hMast)

***int DisconnectTD( int hTD);***

This command close the current comport for the selected device 1.2.

hTD                       = the LUN number default = 1  
  
return                    = error code  
call     RetVal = DisconnectTD(hTD)



***int ReferenceTD( int hTD);***

This command reference the mast for the selected device 1.2.  
!!! before start the process of measuring the mast must be referenced!!!

hTD                   = the LUN number default = 1

return               = error code

call     RetVal = ReferenceTD(hTD)

### 1.2.3.ANTENNA STAND FUNCTIONS

***int ConnectAS( int \*hAS, char Port);***

This command connect the mast

Port = the number of comport e.g. '1'  
hMast = the LUN number default = 1

return = error code  
call RetVal = ConnectAS( &hAS, Port)

***int IsASMoving( int hAS, int \*Flag );***

This command gets the current status of the mast for the selected device 1.2.

Flag = is an integer value as follows: moving = 1, static = 0  
hAS = the LUN number default = 1

return = error code  
call RetVal = IsASMoving(hAS,&Flag)

***int SetASPolarisation( int hAS, int Polarisation );***

This command sets the antenna polarization to horizontal or vertical and starts positioning of the antenna on the mast for the selected device 1.2.

Polarisation = is a int value vertical = 1, horizontal = 0  
hAS = the LUN number default = 1

return = error code  
call RetVal = SetASPolarisation(hAS,Polarisation)

***int GetASPolarisation( int hAS, int \*Polarisation );***

This command gets the current polarisation of the antenna on the mast for the selected device 1.2.

Polarisation = is a int value vertical = 1, horizontal = 0  
the antenna is in a position between 1 degree and 89 degree = -1

hAS = the LUN number default = 1

return = error code  
call RetVal = GetASPolarisation(hAS,&Polarisation)



***int ASStop( int hAS );***

This command stops the movement of the mast for the selected device 1.2.

hAS               = the LUN number default = 1

return           = error code  
call   RetVal = ASStop(hAS)

***int DisconnectAS( int hAS );***

This command close the current comport for the selected device 1.2.

hAS               = the LUN number default = 1

return           = error code  
call   RetVal = DisconnectAS(hAS)

***int ReferenceAS( int hAS );***

This command reference the mast for the selected device 1.2.

!!! before start the process of measuring the mast must be referenced!!!

hAS               = the LUN number default = 1

return           = error code  
call   RetVal = ReferenceAS(hAS)





### 1.3.THE ERROR RETURN CODE

- #1 This is Not A Turntable  
    ->on ConnectTurntable()  
This Is Not A Mast  
    ->on ConnectMast()
- #2 Command Parameter Error
- #3 The Device Is Not referenced
- #4 Read Error  
    ->Get Mast - Get Turntable  
    ->The Device Is Not Reverence
- #5 No Connect To Device  
    -> on this port have not found a device the innco.dll  
    -> or the device have not respond of the call from the innco.dll
- #6 Speed Is Out Of Range or Command Parameter Error  
\*\*\*\*\*

#### NOTE!

#### **This error code is not in use in the v1.3**

- ~~#7 Position Is Out Of Range or Command Parameter Error~~

\*\*\*\*\*

- #8 Can`t Assigned Lun
- #9 Can't Open Port  
    -> port address dont know the pc-system  
    -> or port is used from another program



### **1.3.1.THE RETURN CODE OF THE STATUS COMMANDS**

BusyIs( )

- 1 = Axis 1 is busy
- 2 = Axis 2 is busy
- 0 = Not axis is busy

PositionIs( )

- A float value is the position
- 1 = this device is not referenced

PolarisationIs( )

- 1 = Vertical
- 0 = Horizontal
- 1 = Busy

## **1.4.F&Q**

### **1.4.1.Operating method with multi-threading**

You connect with a thread line a com-port with a device and connect with another thread line a another com-port with a another device and you can control simultaneous all device.

Every thread form you used a copy of our innco.dll and think he and the innco.dll are alone, he connect the port if not used from another program thread and make what you want.



### 1.4.2. Operating method without multi-thread

- Connect with ConnectMast() e.g.(PORT 1)
- Set the mast to the position with SetMastHeight()  
attention the mast must be referenced  
with ReferenceMast() otherwise the mast functions do nothing.  
After connect before set.
- Confirm the position, moving, or stop with IsMastMoving(), GetMastPosition() and MastStop().
- Switch to turntable with ConnectTurntable() e.g.(PORT 2)  
the conntact to the mast is lost !

#### NOTE !

The mainprogram must select the port through the command ConnectMast( x , port number ) what it want the mainprogram must know the port of the device. With this number select the mainprogram the port and the joint device. After the connect can the mainprogram use all commands for the selected device.

- Set the Turntable to the position with SetTurntablePosition()
- Switch to the mast with ConnectMast() (PORT 1)  
the connection to the turntable is lost !
- Confirm the position, moving, or stop with IsMastMoving(), GetMastPosition() and MastStop().
- Switch to turntable with ConnectTurntable() (PORT 2)  
the connection to the mast is lost !
- Confirm the position, moving, or stop with IsTurntableMoving(), GetTurntablePosition() and TurntableStop().

A new SetMastHeight(), SetTurntablePosition() and SetMastPolarisation() can the device understand not before the movement is concluded.



## NOTE !

To switch the port and the joint device is it necessary to send the Connect\*() command.  
If you use the \*.dll via separate threads (with multi threading 4.1. ) you can every port and the joint decive simultaneous use without new connect the port and without switching between the ports.

Every thread is independence from the another threads from the mainprogram.

### 1.4.3.This is a option to use the innco.dll

\*\*\*\*\*

(pseudo code instruction)

```
1:    Connect Turntable (Port1)
2:    Reference Turntable()
3:    Wait 100ms
4:    is turntable moving? Yes, GOTO 3
5:    Set Turntable Position (X)
6:    Wait 100ms
7:    get current turntable position
8:    is turntable moving? Yes, GOTO 6
9:    Connect Mast (Port 2)
10:   Reference Mast()
11:   Wait 100ms
12:   is mast moving? Yes, GOTO 11:
13:   set mast height (X)
14:   Wait 100ms
15:   get mast height
16:   Is mast moving?, Yes, GOTO 14:
..
..

1000: Disconnect Turntable ( Port 1)
1001: Disconnect Mast( Port 2) Disconnect Mast ( Port 3)
*****
```



#### 1.4.4. This is another option to use the innco.dll

\*\*\*\*\*

(pseudo code instruction)

```
1:    Connect Turntable (Port1)
2:    Reference Turntable()

3:    Connect Mast (Port 2)
4:    Reference Mast()

5:    Connect Turntable (Port1)
6:    Wait 100ms
7:    is turntable moving? Yes, GOTO 6
8:    Set Turntable Position (X)

9:    Connect Mast (Port 2)
10:   Wait 100ms
11:   is mast moving? Yes, GOTO 10:
12:   set mast height (X)
13:   Wait 100ms
14:   get mast height
15:   is mast moving? Yes, GOTO 13:

16:   Connect Turntable (Port1)
17:   Get Turntable Position (X)

18:   Connect Mast (Port 2)
19:   Get mast height
..
1000: Disconnect Turntable ( Port 1) Disconnect Mast( Port 8)
1001: Disconnect Mast( Port 2) Disconnect Turntable ( Port 15)
*****
```

#### NOTE !

It is not necessary to disconnect all com ports at terminating your application. The innco.dll disconnect the last connection before he connect the next port. The command Disconnect\*() is necessary to make available the com port for use through the stranger application and if you close your application, for an error-free terminating only the last connection.

E.g. ConnectTurntable(port1) is the first command before you can talk with the port and the joint turntable, to talk with the mast must you ConnectMast(port2). Before you can again talk with the turntable must you ConnectTurntable(port1) and now can you use the turntable. It is not necessary to wait is the moving conclusion.

Every Connect\*() command close all open port (from this thread was are formerly) before the new connection will be starting. You must not control which port is disconnected except by the terminating of the application.



## **1.5.WHAT IS THE DIFFRENT BETWEEN VERSION 1.2 AND 1.3**

The \*.dll control in the version 1.2 the input value from the function

SetMastHeight and SetTurntablePosition

and return the error code #7 “position is out of range”

The version 1.3 do not control the input value from the

SetMastHeight and SetTurntablePosition

You can't not use below function for devices without hardware limit.

The min value is '0' and the max value is '0' by devices without hardware limit.

GetTurntableHardwareLimits and GetMastHardwareLimits

## **1.6. PROGRAMMING WITH THE INNCO.DLL**

*Use the below files in your C++ MFC project:*

*INNCO.DLL*

*INNCO.h*

*INNCO.LIB*

*in your project.*

*Look into the DllTestExe.c for a  
light programming with the \*.dll.*



### **Support for the INNCO.DLL:**

innco systems GMBH  
Erlenweg 12  
92521 Schwarzenfeld / Germany

Email: [info@inncosystems.com](mailto:info@inncosystems.com)

Phone: +49 (0) 9435 301659 0  
Fax: +49 (0) 9435 301659 99