

DLL for SELECTION of BELT DRIVEN FANS

**For integration into OEM CAE/design programs
running under 32 bit PC operating systems**

DLL Version 2

(Rel. 2.5.1)

INTERFACE SPECIFICATIONS

Rev. 3

Details on I/O calls to NICOTRA.DLL

The following pages contain, for each exported function of the nicotra.dll file, subroutine definition, compiler directives and calling list variable definitions, according to Microsoft Fortran language and Microsoft Fortran Power Station Compiler conventions.

All character variables in input shall be in C styled, null-terminated format.

Integer(2) or Integer*2 variables are the same as VB Integer and C short int variables.

Integer, Integer (4) or Integer*4 variables are the same as VB Long and C long int variables.

Real(4) variables are the same as VB Single and C float variables.

Real(8) variables are the same as VB Double and C double variables.

SUBROUTINE DEFINITIONS

subroutine GET_INI_CONFIG(Nserie,Serie)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_INI_CONFIG
!MS$ ATTRIBUTES ALIAS:'_GET_INI_CONFIG':: GET_INI_CONFIG
!MS$ ATTRIBUTES REFERENCE :: Nserie
!MS$ ATTRIBUTES REFERENCE :: Serie
      Integer*4 Nserie
      Character(10) Serie(100)
      .....
```

Subroutine GET_PRODUCTS(List)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_PRODUCTS
!MS$ ATTRIBUTES ALIAS:'_GET_PRODUCTS':: GET_PRODUCTS
!MS$ ATTRIBUTES REFERENCE :: List
      Character(30000) List
      .....
```

Subroutine GET_ARC_VERSION(LG,List)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_ARC_VERSION
!MS$ ATTRIBUTES ALIAS:'_GET_ARC_VERSION':: GET_ARC_VERSION
!MS$ ATTRIBUTES REFERENCE :: LG
!MS$ ATTRIBUTES REFERENCE :: List
      Integer*2 LG
      Character(20000) List
      .....
```

Subroutine GET_ACCESSORY(KEY,LG,List)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_ACCESSORY
!MS$ ATTRIBUTES ALIAS:'_GET_ACCESSORY':: GET_ACCESSORY
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE :: LG
!MS$ ATTRIBUTES REFERENCE :: List
```

BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

Character(16) KEY ! null character in position 16 if KEY is 15 chars long
Character(20000) List
Integer*2 LG
.....

Subroutine GET_STANDARDS_FANALONE(KEY,Z1,Z2,OUT)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_STANDARDS_FANALONE
!MS$ ATTRIBUTES ALIAS: '_GET_STANDARDS_FANALONE':: GET_STANDARDS_FANALONE
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE :: Z1,Z2
!MS$ ATTRIBUTES REFERENCE :: OUT
    Character(16) KEY ! null character in position 16 if KEY is 15 chars long
    Integer(2) z1,z2 ! not used, only for compatibility reasons
    Real(8) OUT(0:13)
    .....
```

Subroutine GET_CALCULATION_FANALONE(S1,S2,IN,KEY,Z1,Z2,OUT)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_CALCULATION_FANALONE
!MS$ ATTRIBUTES ALIAS: '_GET_CALCULATION_FANALONE':: GET_CALCULATION_FANALONE
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE:: Z1,Z2
!MS$ ATTRIBUTES REFERENCE:: S1,S2
!MS$ ATTRIBUTES REFERENCE :: IN
!MS$ ATTRIBUTES REFERENCE :: OUT
    Character(16) KEY ! null character in position 16 if KEY is 15 chars long
    Integer(2) s1,s2,z1,z2 !! not used, only for compatibility reasons
    Real(8) IN(0:12)
    Real(8) OUT(0:31)
    .....
```

Subroutine GET_CALCULATION_MULTIFANS(S1,S2,IN,KEY,Z1,Z2,OUT,Vent)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_CALCULATION_MULTIFANS
!MS$ ATTRIBUTES ALIAS: '_GET_CALCULATION_MULTIFANS':: GET_CALCULATION_MULTIFANS
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE:: Z1,Z2
!MS$ ATTRIBUTES REFERENCE:: S1,S2
!MS$ ATTRIBUTES REFERENCE :: IN
!MS$ ATTRIBUTES REFERENCE :: OUT
!MS$ ATTRIBUTES REFERENCE :: Vent
```

Character(16) KEY ! null character in position 16 if KEY is 15 chars long
Integer(2) s1,s2,z1,z2
Real (8) IN(0:12) ! input array
Real (8) OUT ! error flag
Type (Fans) Vent(0:30) ! output array of structures

TYPE Fans

```
SEQUENCE
Character(18) tipo
Character(12) wrn
Integer      nrec
Real(4)      Eta
Real(4)      Pvol
Real(4)      Ptot
Real(4)      Pstat
Real(4)      Pdin
```

BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

```

Real(4)      Ngiri
Real(4)      W
Real(4)      Wel
Real(4)      vper
Real(4)      Vel
Real(4)      lwam
Real(4)      lwaa
Real(4)      Ro
Real(4)      NB
Real(4)      Ns
Real(4)      NMAX
Real(4)      WMAX
Real(4)      diam
Real(4)      miner
Real(4)      fl1
Real(4)      fl2
Real(4)      fl3
Real(4)      fl4
Real(4)      fl5
Real(4)      NG
Real(4)      Cw
Real(4)      CN
Real(4)      cps
Real(4)      SPLm
Real(4)      fr1m
Real(4)      fr2m
Real(4)      fr3m
Real(4)      fr4m
Real(4)      fr5m
Real(4)      fr6m
Real(4)      fr7m
Real(4)      fr8m
Real(4)      SPLa
Real(4)      fr1a
Real(4)      fr2a
Real(4)      fr3a
Real(4)      fr4a
Real(4)      fr5a
Real(4)      fr6a
Real(4)      fr7a
Real(4)      fr8a
Real(4)      esp
Real(4)      Q
END TYPE

```

.....

Subroutine GET_CALCULATION_BELT(S1,S2,IN,KEY,BELT,ERRORMESSAGE,Z1,Z2,OUT)

!MS\$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_CALCULATION_BELT

!MS\$ ATTRIBUTES ALIAS: '_GET_CALCULATION_BELT':: GET_CALCULATION_BELT

!MS\$ ATTRIBUTES REFERENCE:: S1,S2

!MS\$ ATTRIBUTES REFERENCE :: IN

!MS\$ ATTRIBUTES REFERENCE :: KEY,BELT,ERRORMESSAGE

!MS\$ ATTRIBUTES REFERENCE:: Z1,Z2

!MS\$ ATTRIBUTES REFERENCE :: OUT

Character(16) KEY

Character(30) BELT,BELTx

Character(80) ERRORMESSAGE

Integer(2) S1,S2 ! not used, only for compatibility reasons

BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

```
Real(8) IN(0:23)      ! dim max
Integer(2) z1,z2      ! not used, only for compatibility reasons
Real(8) OUT(0:19)     ! dim max
.....
```

Subroutine

GET_CALCULATION_MULTI_BELT(S1,S2,IN,KEY,BELT,ERRORMESSAGE,Z1,Z2,OUT,Tras)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_CALCULATION_MULTI_BELT
!MS$ ATTRIBUTES ALIAS: '_GET_CALCULATION_MULTI_BELT':: GET_CALCULATION_MULTI_BELT
!MS$ ATTRIBUTES REFERENCE:: S1,S2
!MS$ ATTRIBUTES REFERENCE :: IN
!MS$ ATTRIBUTES REFERENCE :: KEY,BELT,ERRORMESSAGE
!MS$ ATTRIBUTES REFERENCE:: Z1,Z2
!MS$ ATTRIBUTES REFERENCE :: OUT
!MS$ ATTRIBUTES REFERENCE :: Tras
```

```
Character(16) KEY
Character(30) BELT
Character(80) ERRORMESSAGE
Integer(2) S1,S2
Real(8) IN(0:23)      ! dim max
Integer(2) z1,z2
Real(8) OUT
Type (trasmiss) Tras(30)
```

TYPE trasmiss

SEQUENCE

```
Integer Pm
Integer Ngmot
Integer Ngole
Integer dmot
Integer dven
Integer NgventN
Real(4) Ngvent
Real(4) Wa      ! power, max trasmissible
Real(4) Wr      ! power, design power = Wmot*Cc (Cc=1.3)
Real(4) Wass
Real(4) Wmot
Real(4) heff
Real(4) Qn
Integer Ieff
Integer L
Integer gamma
Real(4) Cl
Real(4) Cg
Real(4) u
Integer cbven
Integer cbmot
Integer damot
Integer daven
Real(4) heffm
Real(4) Ts
Real(4) depth
Real(4) fn
Real(4) Qst
Real(4) CradMot
Real(4) HMot
Real(4) EspMot
```

BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

END TYPE

Subroutine GET_GRAPH_POWER(KEY,Z1,Z2,OUT)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_GRAPH_POWER
!MS$ ATTRIBUTES ALIAS: '_GET_GRAPH_POWER':: GET_GRAPH_POWER
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE:: Z1,Z2
!MS$ ATTRIBUTES REFERENCE :: OUT
      Character(16) KEY ! null character in position 16 if KEY is 15 chars long
      Integer(2) z1,z2 ! not used, only for compatibility reasons
      Real(8) OUT(0:136)
```

Subroutine GET_GRAPH_RPM(KEY,Z1,Z2,OUT)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_GRAPH_POWER
!MS$ ATTRIBUTES ALIAS: '_GET_GRAPH_POWER':: GET_GRAPH_POWER
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE:: Z1,Z2
!MS$ ATTRIBUTES REFERENCE :: OUT
      Character(16) KEY ! null character in position 16 if KEY is 15 chars long
      Integer(2) z1,z2 ! not used, only for compatibility reasons
      Real(8) OUT(0:136) ! output array
```

Subroutine GET_GRAPH_ETA(KEY,Z1,Z2,OUT)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_GRAPH_POWER
!MS$ ATTRIBUTES ALIAS: '_GET_GRAPH_POWER':: GET_GRAPH_POWER
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE:: Z1,Z2
!MS$ ATTRIBUTES REFERENCE :: OUT
      Character(16) KEY ! null character in position 16 if KEY is 15 chars long
      Integer(2) z1,z2 ! not used, only for compatibility reasons
      Real(8) OUT(0:136) ! output array
```

Subroutine PRESS_DB_CONST(KEY,Db, Vmn, Vmx, Py, Vx, Punti,Erl)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: PRESS_DB_CONST
!MS$ ATTRIBUTES ALIAS: '_PRESS_DB_CONST':: PRESS_DB_CONST
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE:: Punti,Erl
!MS$ ATTRIBUTES REFERENCE :: Db,Vmn,Vmx,Py,Vx
      Character(16) KEY ! null character in position 16 if KEY is 15 chars long
      Integer(2) Punti,Erl
      Real (4) Db,Vmn,Vmx,Py(0:100),Vx(0:100)
```

Subroutine POINTS_DB_CONST(KEY,Db, Vmn, Vmx, Py, Vx, Punti, FLAG_INSTALLAZIONE, FLAG_LATO, RO_0, Erl)

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: POINTS_DB_CONST
```

BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

```
!MS$ ATTRIBUTES ALIAS: '_POINTS_DB_CONST':: POINTS_DB_CONST
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE :: Punti, Erl
!MS$ ATTRIBUTES REFERENCE :: Db, Vmn, Vmx, Py, Vx, RO_0
!MS$ ATTRIBUTES REFERENCE :: FLAG_INSTALLAZIONE, FLAG_LATO
```

```
Character(16) KEY ! null character in position 16 if KEY is 15 chars long
INTEGER(2) Punti
INTEGER(2) Erl
REAL(4) Db
REAL(4) Vmn, Vmx
REAL(4) Vx(0:100)
REAL(4) Py(0:100)
INTEGER(2) FLAG_INSTALLAZIONE
INTEGER(2) FLAG_LATO
REAL(4) RO_0
.....
```

Subroutine GET_PICTURE(KEY, hbm)

```
!MS$ ATTRIBUTES DLLEXPORT, STDCALL:: GET_PICTURE
!MS$ ATTRIBUTES ALIAS: '_GET_PICTURE':: GET_PICTURE
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE :: hbm
Character(16) KEY ! null character in position 16 if KEY is 15 chars long
Integer(4) hbm
.....
```

Subroutine GET_PICTURE_NAME (KEY, FileName)

```
!MS$ ATTRIBUTES DLLEXPORT, STDCALL:: GET_PICTURE_NAME
!MS$ ATTRIBUTES ALIAS: '_GET_PICTURE_NAME':: GET_PICTURE_NAME
!MS$ ATTRIBUTES REFERENCE :: KEY
character(16) KEY
character(255) FileName.....
```

Subroutine GET_WMF(KEY, hWM)

```
!MS$ ATTRIBUTES DLLEXPORT, STDCALL:: GET_WMF
!MS$ ATTRIBUTES ALIAS: '_GET_WMF':: GET_WMF
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE :: hWM
Character(16) KEY ! null character in position 16 if KEY is 15 chars long
Integer(4) hWM
.....
```

Subroutine GET_WMF_DIM_LIST(KEY, hWM, Filename, Sort, Letter, Dimen)

```
!MS$ ATTRIBUTES DLLEXPORT, STDCALL:: GET_WMF_DIM_LIST
!MS$ ATTRIBUTES ALIAS: '_GET_WMF_DIM_LIST':: GET_WMF_DIM_LIST
!MS$ ATTRIBUTES REFERENCE :: KEY
!MS$ ATTRIBUTES REFERENCE :: hWM
!MS$ ATTRIBUTES REFERENCE :: Filename
Character(16) KEY ! null character in position 16 if KEY is 15 chars long
Integer hWM
Character(255) Filename
Integer*2 Sort
Character(2) Letter(0:27)
Real*8 Dimen(0:27)
.....
```

BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

```
SUBROUTINE GET_NOISE_DATA(INLET_DATA_OUT, INLET_DATA_IN, OUTLET_DATA_OUT,  
OUTLET_DATA_IN, BO_DATA, FLAG_INSTALLAZIONE, ERL, VENT)
```

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_NOISE_DATA
```

```
!MS$ ATTRIBUTES ALIAS: '_GET_NOISE_DATA':: GET_NOISE_DATA
```

```
!MS$ ATTRIBUTES REFERENCE :: INLET_DATA_OUT, INLET_DATA_IN
```

```
!MS$ ATTRIBUTES REFERENCE :: OUTLET_DATA_OUT, OUTLET_DATA_IN
```

```
!MS$ ATTRIBUTES REFERENCE :: BO_DATA
```

```
!MS$ ATTRIBUTES REFERENCE :: FLAG_INSTALLAZIONE, ERL
```

```
!MS$ ATTRIBUTES REFERENCE :: VENT
```

```
Subroutine GET_CERT_DATA(KEY, ICERTGLOB, ICERTSPEC, ERRORE)
```

```
!MS$ ATTRIBUTES DLLEXPORT,STDCALL:: GET_CERT_DATA
```

```
!MS$ ATTRIBUTES ALIAS: '_GET_CERT_DATA':: GET_CERT_DATA
```

```
!MS$ ATTRIBUTES REFERENCE :: KEY
```

```
!MS$ ATTRIBUTES REFERENCE :: ICERTGLOB
```

```
!MS$ ATTRIBUTES REFERENCE :: ICERTSPEC
```

```
!MS$ ATTRIBUTES REFERENCE :: ERRORE
```


Application examples in Visual Basic

COMMON PUBLIC TYPES DECLARATIONS

Code lines with declarations of User-defined Types.

```
Public Type keytype
    KEY As String * 16
End Type
```

```
Public Type TypeLista
    Lista As String * 20000
End Type
```

```
Public Lista As TypeLista
```

```
Type Fans
    Tipo As String * 18
    wrn As String * 12
    nrec As Long
    Eta As Single
    Pvol As Single
    Ptot As Single
    Pstat As Single
    Pdin As Single
    Ngiri As Single
    w As Single
    Wel As Single
    vper As Single
    Vel As Single
    Lwam As Single
    Lwaa As Single
    Ro As Single
    NB As Single
    Ns As Single
    Nmax As Single
    Wmax As Single
    diam As Single
    miner As Single
    fl1 As Single
    fl2 As Single
    fl3 As Single
    fl4 As Single
    fl5 As Single
    NG As Single
    Cw As Single
    CN As Single
    cps As Single
    SPLm As Single
    fr1m As Single
    fr2m As Single
    fr3m As Single
    fr4m As Single
    fr5m As Single
    fr6m As Single
    fr7m As Single
    fr8m As Single
```

BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

SPLa As Single

fr1a As Single

fr2a As Single

fr3a As Single

fr4a As Single

fr5a As Single

fr6a As Single

fr7a As Single

fr8a As Single

esp As Single

Q As Single

End Type

Type ArrayofFans

fan(0 To 30) As Fans

End Type

Type trasmiss

Pm As Long

Ngmot As Long

Ngole As Long

dmot As Long

dven As Long

NgventN As Long

Ngvent As Single

Wa As Single

Wr As Single

Wass As Single

Wmot As Single

heff As Single

Qn As Single

Ieff As Long

l As Long

Gamma As Long

Cl As Single

Cg As Single

u As Single

cbven As Long

cbmot As Long

damot As Long

daven As Long

heffm As Single

Ts As Single

depth As Single

fn As Single

Qst As Single

CradMot As Single

HMot As Single

EspMot As Single

End Type



BDF DLL ver. 2

Completed 27/11/01

Printed 31/08/2005 15.01

```
Type arrayoftrasm
  tras(1 To 30) As trasmiss
End Type

Type BELT_type
  belt As String * 30
End Type

Type ERRORMESSAGE_type
  errormessage As String * 80
End Type
```

SOME EXAMPLES OF SUBROUTINE DECLARATIONS AND CALLS (VBasic)

Used-defined types declarations are given in the preceding pages.

GET_ACCESSORY

```
Declare Sub GET_ACCESSORY Lib "Nicotra.dll" (ByRef KEY As keytype, ByRef LG As Integer, ByRef List As TypeLista)
.....
Dim KEY As keytype, LG As Integer, Lista As TypeLista
.....
Call GET_ACCESSORY(KEY, LG, Lista)
.....
```

GET_STANDARDS_FANALONE

```
Declare Sub GET_STANDARDS_FANALONE Lib "Nicotra.dll" (ByRef KEY As keytype, ByRef Z1 As Integer, ByRef Z2 As Integer, ByRef out As Double)
.....
Dim idumm1 As Integer, idumm2 As Integer
Dim KEY As keytype, out(0 To 13) As Double
.....
Call GET_STANDARDS_FANALONE(KEY, idumm1, idumm2, out(0))
.....
```

GET_CALCULATION_FANALONE

```
Declare Sub GET_CALCULATION_FANALONE Lib "Nicotra.dll" (ByRef S1 As Integer, ByRef S2 As Integer, ByRef InData As Double, ByRef KEY As keytype, ByRef Z1 As Integer, ByRef Z2 As Integer, ByRef out As Double)
.....
Dim S1 As Integer, S2 As Integer, KEY As keytype, Z1 As Integer, Z2 As Integer
Dim InData(0 To 12) As Double, OUT(0 To 31) As Double
.....
Call GET_CALCULATION_FANALONE(S1, S2, InData(0), KEY, Z1, Z2, OUT(0))
.....
```

GET_CALCULATION_MULTIFANS

```
Declare Sub GET_CALCULATION_MULTIFANS Lib "Nicotra.dll" (ByRef S1 As Integer, ByRef S2 As Integer, ByRef InData As Double, ByRef KEY As keytype, ByRef Z1 As Integer, ByRef Z2 As Integer, ByRef out As Double, ByRef Ventil As ArrayofFans)
.....
Dim S1 As Integer, S2 As Integer, KEY As keytype, Z1 As Integer, Z2 As Integer
Dim InData(0 To 12) As Double, OUT As Double, ERRFL As Double
Dim vent As ArrayofFans
.....
Call GET_CALCULATION_MULTIFANS(S1, S2, InData(0), KEY, Z1, Z2, ERRFL, vent)
.....
```

GET_CALCULATION_BELT

```
Declare Sub GET_CALCULATION_BELT Lib "Nicotra.dll" (ByRef S1 As Integer, ByRef  
    S2 As Integer, ByRef in_ As Double, ByRef KEY As keytype, ByRef belt As  
    BELT_type, ByRef errormessage As ERRORMESSAGE_type, ByRef Z1 As Integer,  
    ByRef Z2 As Integer, ByRef out As Double)
```

```
Dim S1 As Integer, S2 As Integer, KEY As keytype, Z1 As Integer, Z2 As Integer  
Dim belt As BELT_type, errormessage As ERRORMESSAGE_type  
Dim in_b(0 To 23) As Double, OUT_b(0 To 19 * 120) As Double
```

```
.....  
Call GET_CALCULATION_BELT(S1, S2, in_b(0), KEY, belt, errormessage, Z1, Z2,  
    OUT_b(0) )  
.....
```

GET_CALCULATION_MULTI_BELT

```
Declare Sub GET_CALCULATION_MULTI_BELT Lib "Nicotra.dll" (ByRef S1 As Integer,  
    ByRef S2 As Integer, ByRef in_ As Double, ByRef KEY As keytype, ByRef belt  
    As BELT_type, ByRef errormessage As ERRORMESSAGE_type, ByRef Z1 As Integer,  
    ByRef Z2 As Integer, ByRef out As Double, ByRef trasm As arrayoftrasm)
```

```
Dim S1 As Integer, S2 As Integer, KEY As keytype, Z1 As Integer, Z2 As Integer  
Dim belt As BELT_type, errormessage As ERRORMESSAGE_type, err As Double  
Dim in_b(0 To 23) As Double, trasm As arrayoftrasm
```

```
.....  
Call GET_CALCULATION_MULTI_BELT(S1, S2, in_b(0), KEY, belt, errormessage, Z1,  
    Z2, err, trasm)  
.....
```

PRESS_DB_CONST

```
Declare Sub PRESS_DB_CONST Lib "Nicotra.dll" (ByRef KEY As keytype, Db As  
    Single, Vmn As Single, Vmx As Single, Py As Single, Vx As Single, Punti As  
    Integer, Erl As Integer)
```

```
Dim KEY As keytype, Db As Single, Vmin As Single, Vmax As Single, Punti as  
    Integer, Erl As Integer  
Dim Py_(0 To 100) As Single, Vx_(0 To 100) As Single
```

```
.....  
Call PRESS_DB_CONST(KEY, Db, Vmin, Vmax, Py_(0), Vx_(0), Punti, Erl)  
.....
```