SmartDongle and PowerBASIC

The SmartDongle can be used under Powerbasic with a little code and using the VB dll supplied by the company. There are VB and C examples of source code available on the company website, which list the calls to the DLL (usk_vb.dll).

The major calls required for the PB user are the reading, writing and 'login' modes, plus the ability to change the colour of the LED to indicate that a process is going on. The code examples below are Public Domain and carry no warranty - they are used at your own risk, but have been tested and used in a commercial situation.

1) Declares:-

```
DECLARE FUNCTION VBRead LIB "usk_vb.dll"  ALIAS "VBRead"  (BYVAL P1 AS
STRING, BYVAL P2 AS STRING, BYVAL address AS LONG, BYREF Buffer AS
ASCIIZ*128, BYVAL SIZE AS LONG ) AS LONG

DECLARE FUNCTION VBWrite LIB "usk_vb.dll" ALIAS "VBWrite" (BYVAL P1 AS
STRING, BYVAL P2 AS STRING, BYVAL address AS LONG, BYVAL Buffer AS
STRING, BYVAL SIZE AS LONG ) AS LONG

DECLARE FUNCTION SDReset LIB "usk_vb.dll" ALIAS "Reset" () AS LONG

DECLARE FUNCTION LedGreen LIB "usk_vb.dll" ALIAS "LedGreen" () AS LONG

DECLARE FUNCTION LedRed LIB "usk_vb.dll" ALIAS "LedRed" () AS LONG

DECLARE FUNCTION LedOff LIB "usk_vb.dll" ALIAS "LedOff" () AS LONG

DECLARE FUNCTION GetSerialNumber LIB "usk_vb.dll" ALIAS
"GetSerialNumber" (BYREF SerNum AS ASCIIZ*100) AS LONG
```

The GetError routine in the DLL cannot easily be used in PB. It is declared as a FUNCTION, but has no return value. A FUNCTION is given below that avoids this problem.

2) Reading the device

Use VBWrite to perform this task. Examples are below:-

a)
```
FUNCTION LoginToSmartDongle (P1 AS STRING, P2 AS STRING) AS LONG
    LOCAL errorflag AS LONG
    LOCAL errorstring AS STRING
    errorflag = VBRead(P1, P2, 0, CHR$(0), 0)
    IF errorflag<>0 THEN GetErrorString errorflag
    IF errorflag>1 THEN errorflag=1
    FUNCTION=errorflag
    END FUNCTION
```

It is used as:-

```
                    IsKeyPresent= LoginToSmartDongle (P1,P2)
                    IF IsKeyPresent<>0 THEN
                       MSGBOX "Failed to access a valid
SmartDongle.",%MB_ICONINFORMATION,"Smartdongle"
                       EXIT SELECT
                    END IF
```

b) Data may be read by supplying an address and a length of data to retrieve:-

```
stringlength=128
address=128
TrialTextOut=ReadSmartDongle (address,stringlength)
```

where:-

```
FUNCTION ReadSmartDongle (address AS LONG,stringlength as long) AS
STRING
LOCAL TrialTextOut1 AS ASCIIZ*(stringlength+1)
LOCAL errorflag AS LONG
    TrialTextOut1=""
    errorflag =  VBRead (P1, P2, address, TrialTextOut1, stringlength)
    IF errorflag<>0 THEN GetErrorString errorflag
    TrialTextOut1=REMOVE$(TrialTextOut1,CHR$(0))
    IF LEN(TrialTextOut1)>0 THEN FUNCTION =TrialTextOut1
END FUNCTION
```

3) Writing to the device

Use VBWrite to write to the device:-

```
FUNCTION WriteToSmartDongle (TrialText AS STRING,address AS
LONG,stringlength as long) AS LONG
    LOCAL errorflag AS LONG
    LOCAL TrialTextIn AS ASCIIZ*(stringlength+1)

    TrialText=LEFT$(TrialText+SPACE$(128),stringlength)
    TrialTextIn=TrialText
    errorflag = VBWrite(P1, P2, address, TrialTextIn,stringlength)
'write data to dongle

    IF errorflag<>0 THEN
        GetErrorString errorflag
        FUNCTION =0:EXIT FUNCTION
    END IF

END FUNCTION
```

4) Getting device serial number:-

```
FUNCTION GetSerNum () AS STRING
LOCAL SerNum AS ASCIIZ*100
LOCAL errorflag AS LONG
    errorflag=GetSerialNumber (SerNum)
    IF errorflag<>0 THEN GetErrorString errorflag
    FUNCTION=SerNum
END FUNCTION
```

used as -

```
LOCAL SerNum AS ASCIIZ*128
SerNum=GetSerNum ()
```

5) Changing LED display:-

LEDRed() will change the LED to red
LEDGreen() will change the LED to green
LEDOff() will turn the LED off

The example code below will perform a task - as it does so the LED will flash red, then stay a steady green when the task is finished.

```
                CASE %IDC_BUTTON1 'write
                    stringlength=128
                    FOR n=1001 TO 1023
                        LEDRed()
                        CONTROL SET TEXT CB.HNDL,
1000,"Writing"+STR$(n-1000)
                        address=128*(n-1001)
                        CONTROL GET TEXT CB.HNDL,n TO TrialText
                        WriteToSmartDongle TrialText,address
                        LEDOff()
                        DIALOG REDRAW CB.HNDL
                    NEXT
                    CONTROL SET TEXT CB.HNDL, 1000,""
                    LEDGreen()
```

6) Error routine:-

```
FUNCTION GetErrorString ( errorflag AS LONG) AS LONG    'errorString AS
STRING,
    LOCAL message AS STRING
'from sd_uskerr.h for SmartDongle API version 3.2
'www.smartdongle.com
'   MicroWorks, Inc.
'   2808 North Cole Road
'   Boise, ID 83704

    SELECT CASE errorflag
        CASE 0' USK_OK  0
            message="OK"
        CASE 1 'USK_ERR_OPEN                              1
            message="Can't open dongle"
        CASE 2 'USK_ERR_IS_OPEN                           2
            message="Dongle is already open."
        CASE 3 'USK_ERR_MEM                               3
            message="Memory allocation error "
        CASE 4 'USK_ERR_EEPROM_SIZE                       4
            message="Deprecated (4)"
        CASE 5 'USK_ERR_5                                 5
            message="Reserved (5)"
        CASE 6 'USK_ERR_SECURITY_STATE_TIMEOUT            6
            message="Timeout waiting for state change "
        CASE 7 'USK_ERR_KEY_STATE_TIMEOUT                 7
            message="Timeout waiting for state change "
        CASE 8 'USK_ERR_TWC_RESET_TIMEOUT                 8
            message="Timeout waiting for TWC to reset "
        CASE 9 'USK_ERR_BUSY_TIMEOUT                      9
            message="Timeout waiting for task_loop to be flagged as
inactive."
        CASE 10 'USK_ERR_NOT_ENABLED                      10
            message="The Dongle is Not in ENABLED Mode "
        CASE 11 'USK_ERR_EEPROM_READ_TIMEOUT              11
            message="Timeout Waiting For Read Flag(s) to be set "
        CASE 12 'USK_ERR_EEPROM_READ                      12
            message="An EEPROM Read ERROR Occurred "
        CASE 13 'USK_ERR_EEPROM_WRITE                     13
```

```
            message="An EEPROM Write ERROR Occurred "
        CASE 14 'USK_ERR_USK_VALIDATE                    14
            message="Dongle failed to validate against user keys "
        CASE 15 'USK_ERR_HOST_VALIDATE                   15
            message="Host failed to validate against dongle keys "
        CASE 16 'USK_ERR_DISABLE                         16
            message="An error occurred when attempting to lock
SmartDongle "
        CASE 17 'USK_ERR_DATA_FORMAT                     17
            message="Data format Error "
        CASE 18 'USK_ERR_IOCTL_GET_PIPE_INFO             18
            message="IOCTL_GET_PIPE_INFO Error "
        CASE 19 'USK_ERR_IOCTL_GET_DEVICE_DESCRIPTOR     19
            message="IOCTL_GET_DEVICE_DESCRIPTOR Error "
        CASE 20 'USK_ERR_IOCTL_GET_CONFIG_DESCRIPTOR     20
            message="IOCTL_GET_CONFIG_DESCRIPTOR Error "
        CASE 21 'USK_ERR_IOCTL_REGISTER_NOTIFY_EVENT     21
            message="IOCTL_REGISTER_NOTIFY_EVENT Error "
        CASE 22 'USK_ERR_IOCTL_RESET_DEVICE              22
            message="IOCTL_RESET_DEVICE Error "
        CASE 23 'USK_ERR_IOCTL_RESET_PIPE                23
            message="IOCTL_RESET_PIPE Error "
        CASE 24 'USK_ERR_IOCTL_GET_DRIVER_REVISION       24
            message="IOCTL_GET_DRIVER_REVISION Error "
        CASE 25 'USK_ERR_IOCTL_GET_STRING_DESCRIPTOR     25
            message="IOCTL_GET_STRING_DESCRIPTOR Error "
        CASE 26 'USK_ERR_26                              26
            message="Deprecated (26)"
        CASE 27 'USK_ERR_IOCTL_SEND_KEY                  27
            message="IOCTL_SEND_KEY Error "
        CASE 28 'USK_ERR_IOCTL_GET_KEY                   28
            message="IOCTL_GET_KEY Error "
        CASE 29 'USK_ERR_IOCTL_WRITE_PORT                29
            message="IOCTL_WRITE_PORT Error "
        CASE 30 'USK_ERR_IOCTL_READ_PORT                 30
            message="IOCTL_READ_PORT Error "
        CASE 31 'USK_ERR_IOCTL_READ_EEPROM               31
            message="IOCTL_READ_EEPROM Error "
        CASE 32 'USK_ERR_IOCTL_WRITE_EEPROM              32
            message="IOCTL_WRITE_EEPROM Error "
        CASE 33 'USK_ERR_IOCTL_GET_SECURITY_STATE        33
            message="IOCTL_GET_SECURITY_STATE Error "
        CASE 34 'USK_ERR_IOCTL_RESET_SECURITY_STATE      34
            message="IOCTL_RESET_SECURITY_STATE Error "
        CASE 35 'USK_ERR_IOCTL_GET_KEY_STATE             35
            message="IOCTL_GET_KEY_STATE Error "
        CASE 36 'USK_ERR_IOCTL_RESET_KEY_STATE           36
            message="IOCTL_RESET_KEY_STATE Error "
        CASE 37 'USK_ERR_IOCTL_SEND_HOST_VERIFY_REQUEST  37
            message="IOCTL_SEND_HOST_VERIFY_REQUEST Error "
        CASE 38 'USK_ERR_IOCTL_REQUEST_PAGE              38
            message="IOCTL_REQUEST_PAGE Error "
        CASE 39 'USK_ERR_IOCTL_GET_PAGE_FLAGS            39
            message="IOCTL_GET_PAGE_FLAGS Error "
        CASE 40 'USK_ERR_IOCTL_GET_TWC_FLAG              40
            message="IOCTL_GET_TWC_FLAG Error "
        CASE 41 'USK_ERR_IOCTL_GET_BUSY_FLAG             41
```

```
    message="IOCTL_GET_BUSY_FLAG Error "
CASE 42 'USK_ERR_IOCTL_GET_EEPROM_ERROR_FLAG     42
    message="IOCTL_GET_EEPROM_ERROR_FLAG Error "
CASE 43 'USK_ERR_43                              43
    message="Reserved (43)"
CASE 44 'USK_ERR_44                              44
    message="Reserved (44)"
CASE 45 'USK_ERR_IOCTL_QTSDONGLE_SEND_PORT       45
    message="IOCTL_QTSDONGLE_SEND_PORT Error "
CASE 46 'USK_ERR_IOCTL_QTSDONGLE_GET_FLAGS       46
    message="IOCTL_QTSDONGLE_GET_FLAGS Error "
CASE 47 'USK_ERR_IOCTL_SET_MODE                  47
    message="IOCTL_SET_MODE Error "
CASE 48 'USK_ERR_IOCTL_GET_MODE                  48
    message="IOCTL_GET_MODE Error "
CASE 49 'USK_ERR_NODEV                           49
    message="NODEV Error "
CASE 54 'USK_ERR_MUTEX_DEV_CREATE                54
    message="MUTEX_DEV_CREATE Error "
CASE 55 'USK_ERR_MUTEX_DEV_TIMEOUT               55
    message="MUTEX_DEV_TIMEOUT Error "
CASE 56 'USK_ERR_MUTEX_DEV_ABANDONED             56
    message="MUTEX_DEV_ABANDONED Error "
CASE 57 'USK_ERR_MUTEX_DEV_UNKNOWN               57
    message="MUTEX_DEV_UNKNOWN Error "
CASE 58 'USK_ERR_MUTEX_API_CREATE                58
    message="MUTEX_API_CREATE Error "
CASE 59 'USK_ERR_MUTEX_API_TIMEOUT               59
    message="MUTEX_API_TIMEOUT Error "
CASE 60 'USK_ERR_MUTEX_API_ABANDONED             60
    message="MUTEX_API_ABANDONED Error "
CASE 61 'USK_ERR_MUTEX_API_UNKNOWN               61
    message="MUTEX_API_UNKNOWN Error "
CASE 63 'USK_ERR_BAD_ARGUMENTS                   63
    message="BAD_ARGUMENTS Error "
CASE 65 'USK_ERR_SID_RETRIEVE                    65
    message="DSID_RETRIEVE Error "
CASE 66 'USK_ERR_ACL_INIT                        66
    message="ACL_INIT Error "
CASE 67 'USK_ERR_ACE_ADD                         67
    message="ACE_ADD Error "
CASE 68 'USK_ERR_SD_INIT                         68
    message="SD_INIT Error "
CASE 69 'USK_ERR_SD_SET                          69
    message="SD_SET Error "
CASE 80 'USK_ERR_80                              80
    message="Reserved (80)"
CASE 81 'USK_ERR_IOCTL_QTSDONGLE_SET_STRICT      81
    message="IOCTL_QTSDONGLE_SET_STRICT Error "
CASE 82 'USK_ERR_REQUIRES_DRIVER_2_3_0           82
    message="REQUIRES_DRIVER_2_3_0 Error "
CASE 83 'USK_ERR_REQUIRES_DONGLE_1_50            83
    message="REQUIRES_DONGLE_1_50 Error "
CASE 84 'USK_WARN_MODE_PAGE_BOUNDARY             84
    message="WARN_MODE_PAGE_BOUNDARY Error "
CASE 85 'USK_WARN_CRYPT_PAGE_BOUNDARY            85
    message="WARN_CRYPT_PAGE_BOUNDARY Error "
```

```
        CASE 86 'USK_RESET_COMEREADY                        86
            message="RESET_COMEREADY Error "
        CASE 255 'USK_ERR_UNKNOWN                           255
            message="Unknown Error "
        CASE ELSE
           message= "Unknown Error (2)"
    END SELECT

    MSGBOX message,%MB_ICONINFORMATION,"SmartDongle"

END FUNCTION
'================================================
```

7) Operational tips:-

a) 'Login' to check that the device is present. If it is not, put up a warning message and leave the program.

b) Check for the device presence at strategic points in the program in case it is removed.

c) use the memory to store user information - remember that reading is relatively slow. Include the data that is used on the program printouts.

d) 'lock' a specific key to a user (or company) and to an individual program e.g. put a reference number in the program code and put this in the device. Read the device and compare the numbers - if different, put up a warning message and leave the program. A company may put copies of the program on several machines but they will not run if the device is not present. Location specific data on the device can give added protection as although the program will run, if it is in a different location, all the printouts will give the 'registered' address.

e) A program may be leased for a specific time - store the end date in the device and check it every time the program is run and periodically in case the program is running continuously.

f) A program may be leased for a specific number of operations. Put the counter on the device and decrement this each time the program is run. You may need to decrement say every two hours in case the program is left running continuously.

8) Example code to program the device (not the best code, but it works):-

```
'SmartDongle utility by Iain Johnstone
'Released to the Public Community as-is with warranty - use at your own
risk.
'INSTALL THE DONGLE DRIVER BEFORE USING THIS SOFTWARE!!!!!!
'see http://www.smartdongle.com/developer-tools-and-drivers - then
select the appropriate package:-
'SmartDongle x86 Driver Installer (MSI)
'or
'SmartDongle x64 Driver Installer (MSI)
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'Code:-
'Change the PI and P2 values in PBMAIN from the demo values to your own
'values as issued by SmartDongle. If incorrect
'values are used the program will throw you out!
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'Useage:-
'This example will take data from the screen and write it to the
'dongle. It will also read it back
'for checking. The textboxes are initially filled with short data.
```

```
'
'Login will open the dongle and return the serial number.
'
'Write will save the data to the dongle. The LED will flash during this
process (which is a bit slow) - under
'Vista the counter may stop functioning but the LED continues to flash
'as the data is written.
'
'Clear screen will clear all data textboxes
'
'Check will read the dongle and then fill the textboxes
'
'Save will save the data to a text file (*.DNG) so that a record is
'kept if a dongle is 'lost' and another
'can be written without entering the data again
'
'Load will read the file and display the data in the textboxes.
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'Instructions:-
'1. Click on Login - serial number displayed
'2. Enter required text
'3. Click on Save - save file
'4. Click on Write - LED will flash as data is written
' (under Vista you may get a 'Program not responding'
'message - ignore it!)
'5. Click on Clear screen
'6. Click on Check memory - dongle is read then data displayed.
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'This code is used at your own risk - it is available to demonstrate
usage under PowerBASIC.
'Please modify this code for your own needs.
'PB9 compiler
#COMPILE EXE
'#DEBUG ERROR ON
'#DEBUG DISPLAY ON
#DIM ALL
'------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
'   ** Includes **
'------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
#INCLUDE ONCE "WIN32API.INC"
#INCLUDE ONCE "COMMCTRL.INC"
'------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------

'------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
```

```
'   ** Constants **
'-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
--------------------------
#PBFORMS BEGIN CONSTANTS
%IDD_DIALOG1   =  101
%IDC_TEXTBOX1  = 1001
%IDC_TEXTBOX2  = 1002
%IDC_TEXTBOX3  = 1003
%IDC_TEXTBOX4  = 1004
%IDC_TEXTBOX5  = 1005
%IDC_TEXTBOX6  = 1006
%IDC_TEXTBOX7  = 1007
%IDC_TEXTBOX8  = 1008
%IDC_TEXTBOX9  = 1009
%IDC_TEXTBOX10 = 1010
%IDC_TEXTBOX11 = 1011
%IDC_TEXTBOX12 = 1012
%IDC_TEXTBOX14 = 1013
%IDC_TEXTBOX15 = 1014
%IDC_TEXTBOX16 = 1015
%IDC_TEXTBOX17 = 1016
%IDC_TEXTBOX18 = 1017
%IDC_TEXTBOX20 = 1018
%IDC_TEXTBOX21 = 1019
%IDC_TEXTBOX22 = 1020
%IDC_TEXTBOX24 = 1021
%IDC_TEXTBOX25 = 1022
%IDC_TEXTBOX26 = 1023
%IDC_TEXTBOX27 = 1024

%IDC_BUTTON1   = 5001
%IDC_BUTTON2   = 5002
%IDC_BUTTON3   = 5003
%IDC_BUTTON4   = 5004
%IDC_BUTTON5   = 5005
%IDC_BUTTON6   = 5006
#PBFORMS END CONSTANTS
'-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
--------------------------
GLOBAL P1, P2 AS STRING
'-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
--------------------------
'   ** Declarations **
'-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
--------------------------
DECLARE CALLBACK FUNCTION ShowDIALOG1Proc()
DECLARE FUNCTION ShowDIALOG1(BYVAL hParent AS DWORD) AS LONG
```

```
'####################################################################
###########
DECLARE FUNCTION VBRead LIB "usk_vb.dll"  ALIAS "VBRead"  (BYVAL P1 AS
STRING, BYVAL P2 AS STRING, BYVAL address AS LONG, BYREF Buffer AS
ASCIIZ*128, BYVAL SIZE AS LONG ) AS LONG
DECLARE FUNCTION VBWrite LIB "usk_vb.dll" ALIAS "VBWrite" (BYVAL P1 AS
STRING, BYVAL P2 AS STRING, BYVAL address AS LONG, BYVAL Buffer AS
STRING, BYVAL SIZE AS LONG ) AS LONG
DECLARE FUNCTION SDReset LIB "usk_vb.dll" ALIAS "Reset" () AS LONG
DECLARE FUNCTION LedGreen LIB "usk_vb.dll" ALIAS "LedGreen" () AS LONG
DECLARE FUNCTION LedRed LIB "usk_vb.dll" ALIAS "LedRed" () AS LONG
DECLARE FUNCTION LedOff LIB "usk_vb.dll" ALIAS "LedOff" () AS LONG
DECLARE FUNCTION GetSerialNumber LIB "usk_vb.dll" ALIAS
"GetSerialNumber" (BYREF SerNum AS ASCIIZ*100) AS LONG
'====================================================================
==============

'--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
-------------------------

'--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
-------------------------
'   ** Main Application Entry Point **
'--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
-------------------------
FUNCTION PBMAIN()
     LedOff()
'   /* SmartDongle  Demo Keys */
   P1 = "0xec6cc589aefd1e75"
   P2 = "0xfcec0a6a82747b3f"

    LEDGreen ()  'turn on LED to green
    DongleScreen %HWND_DESKTOP
END FUNCTION
'--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
-------------------------



'functions
'====================================================================
====================================================================
====================================================================
=========================
FUNCTION LoginToSmartDongle (P1 AS STRING, P2 AS STRING) AS LONG
    LOCAL errorflag AS LONG
    LOCAL errorstring AS STRING
    errorflag = VBRead(P1, P2, 0, CHR$(0), 0)
    IF errorflag<>0 THEN GetErrorString errorflag
```

```
    IF errorflag>1 THEN errorflag=1
    FUNCTION=errorflag
END FUNCTION
'-------------------------------------------------------------------
--------------
FUNCTION WriteToSmartDongle (TrialText AS STRING,address AS LONG) AS
LONG
    LOCAL errorflag,stringlength AS LONG
    LOCAL TrialTextIn AS ASCIIZ*128

    TrialText=LEFT$(TrialText+SPACE$(128),128)
    TrialTextIn=TrialText
    stringlength=128
    errorflag = VBWrite(P1, P2, address, TrialTextIn,stringlength)
'write data to dongle

    IF errorflag<>0 THEN
        GetErrorString errorflag
        FUNCTION =0:EXIT FUNCTION
    END IF

END FUNCTION
'-------------------------------------------------------------------
--------------
FUNCTION ReadSmartDongle (address AS LONG) AS STRING
LOCAL TrialTextOut1 AS ASCIIZ*128
LOCAL errorflag,stringlength AS LONG

    TrialTextOut1=""
    stringlength=128
    errorflag =  VBRead (P1, P2, address, TrialTextOut1, stringlength)
    IF errorflag<>0 THEN GetErrorString errorflag
    TrialTextOut1=REMOVE$(TrialTextOut1,CHR$(0))
    IF LEN(TrialTextOut1)>0 THEN FUNCTION =TrialTextOut1

END FUNCTION
'-------------------------------------------------------------------
--------------
FUNCTION GetSerNum () AS STRING
LOCAL SerNum AS ASCIIZ*100
LOCAL errorflag AS LONG
    errorflag=GetSerialNumber (SerNum)
    IF errorflag<>0 THEN GetErrorString errorflag
    FUNCTION=SerNum
END FUNCTION
'===================================================================
=========
FUNCTION GetErrorString ( errorflag AS LONG) AS LONG    'errorString AS
STRING,
    LOCAL message AS STRING
'from sd_uskerr.h for SmartDongle API version 3.2
'www.smartdongle.com
'   MicroWorks, Inc.
'   2808 North Cole Road
'   Boise, ID 83704
```

```
    SELECT CASE errorflag
        CASE 0' USK_OK  0
            message="OK"
        CASE 1 'USK_ERR_OPEN                              1
            message="Can't open dongle"
        CASE 2 'USK_ERR_IS_OPEN                           2
            message="Dongle is already open."
        CASE 3 'USK_ERR_MEM                               3
            message="Memory allocation error "
        CASE 4 'USK_ERR_EEPROM_SIZE                       4
            message="Deprecated (4)"
        CASE 5 'USK_ERR_5                                 5
            message="Reserved (5)"
        CASE 6 'USK_ERR_SECURITY_STATE_TIMEOUT            6
            message="Timeout waiting for state change "
        CASE 7 'USK_ERR_KEY_STATE_TIMEOUT                 7
            message="Timeout waiting for state change "
        CASE 8 'USK_ERR_TWC_RESET_TIMEOUT                 8
            message="Timeout waiting for TWC to reset "
        CASE 9 'USK_ERR_BUSY_TIMEOUT                      9
            message="Timeout waiting for task_loop to be flagged as
inactive."
        CASE 10 'USK_ERR_NOT_ENABLED                      10
            message="The Dongle is Not in ENABLED Mode "
        CASE 11 'USK_ERR_EEPROM_READ_TIMEOUT              11
            message="Timeout Waiting For Read Flag(s) to be set "
        CASE 12 'USK_ERR_EEPROM_READ                      12
            message="An EEPROM Read ERROR Occurred "
        CASE 13 'USK_ERR_EEPROM_WRITE                     13
            message="An EEPROM Write ERROR Occurred "
        CASE 14 'USK_ERR_USK_VALIDATE                     14
            message="Dongle failed to validate against user keys "
        CASE 15 'USK_ERR_HOST_VALIDATE                    15
            message="Host failed to validate against dongle keys "
        CASE 16 'USK_ERR_DISABLE                          16
            message="An error occurred when attempting to lock
SmartDongle "
        CASE 17 'USK_ERR_DATA_FORMAT                      17
            message="Data format Error "
        CASE 18 'USK_ERR_IOCTL_GET_PIPE_INFO              18
            message="IOCTL_GET_PIPE_INFO Error "
        CASE 19 'USK_ERR_IOCTL_GET_DEVICE_DESCRIPTOR      19
            message="IOCTL_GET_DEVICE_DESCRIPTOR Error "
        CASE 20 'USK_ERR_IOCTL_GET_CONFIG_DESCRIPTOR      20
            message="IOCTL_GET_CONFIG_DESCRIPTOR Error "
        CASE 21 'USK_ERR_IOCTL_REGISTER_NOTIFY_EVENT      21
            message="IOCTL_REGISTER_NOTIFY_EVENT Error "
        CASE 22 'USK_ERR_IOCTL_RESET_DEVICE               22
            message="IOCTL_RESET_DEVICE Error "
        CASE 23 'USK_ERR_IOCTL_RESET_PIPE                 23
            message="IOCTL_RESET_PIPE Error "
        CASE 24 'USK_ERR_IOCTL_GET_DRIVER_REVISION        24
            message="IOCTL_GET_DRIVER_REVISION Error "
        CASE 25 'USK_ERR_IOCTL_GET_STRING_DESCRIPTOR      25
            message="IOCTL_GET_STRING_DESCRIPTOR Error "
```

```
CASE 26 'USK_ERR_26                                26
    message="Deprecated (26)"
CASE 27 'USK_ERR_IOCTL_SEND_KEY                    27
    message="IOCTL_SEND_KEY Error "
CASE 28 'USK_ERR_IOCTL_GET_KEY                     28
    message="IOCTL_GET_KEY Error "
CASE 29 'USK_ERR_IOCTL_WRITE_PORT                  29
    message="IOCTL_WRITE_PORT Error "
CASE 30 'USK_ERR_IOCTL_READ_PORT                   30
    message="IOCTL_READ_PORT Error "
CASE 31 'USK_ERR_IOCTL_READ_EEPROM                 31
    message="IOCTL_READ_EEPROM Error "
CASE 32 'USK_ERR_IOCTL_WRITE_EEPROM                32
    message="IOCTL_WRITE_EEPROM Error "
CASE 33 'USK_ERR_IOCTL_GET_SECURITY_STATE          33
    message="IOCTL_GET_SECURITY_STATE Error "
CASE 34 'USK_ERR_IOCTL_RESET_SECURITY_STATE        34
    message="IOCTL_RESET_SECURITY_STATE Error "
CASE 35 'USK_ERR_IOCTL_GET_KEY_STATE               35
    message="IOCTL_GET_KEY_STATE Error "
CASE 36 'USK_ERR_IOCTL_RESET_KEY_STATE             36
    message="IOCTL_RESET_KEY_STATE Error "
CASE 37 'USK_ERR_IOCTL_SEND_HOST_VERIFY_REQUEST    37
    message="IOCTL_SEND_HOST_VERIFY_REQUEST Error "
CASE 38 'USK_ERR_IOCTL_REQUEST_PAGE                38
    message="IOCTL_REQUEST_PAGE Error "
CASE 39 'USK_ERR_IOCTL_GET_PAGE_FLAGS              39
    message="IOCTL_GET_PAGE_FLAGS Error "
CASE 40 'USK_ERR_IOCTL_GET_TWC_FLAG                40
    message="IOCTL_GET_TWC_FLAG Error "
CASE 41 'USK_ERR_IOCTL_GET_BUSY_FLAG               41
    message="IOCTL_GET_BUSY_FLAG Error "
CASE 42 'USK_ERR_IOCTL_GET_EEPROM_ERROR_FLAG       42
    message="IOCTL_GET_EEPROM_ERROR_FLAG Error "
CASE 43 'USK_ERR_43                                43
    message="Reserved (43)"
CASE 44 'USK_ERR_44                                44
    message="Reserved (44)"
CASE 45 'USK_ERR_IOCTL_QTSDONGLE_SEND_PORT         45
    message="IOCTL_QTSDONGLE_SEND_PORT Error "
CASE 46 'USK_ERR_IOCTL_QTSDONGLE_GET_FLAGS         46
    message="IOCTL_QTSDONGLE_GET_FLAGS Error "
CASE 47 'USK_ERR_IOCTL_SET_MODE                    47
    message="IOCTL_SET_MODE Error "
CASE 48 'USK_ERR_IOCTL_GET_MODE                    48
    message="IOCTL_GET_MODE Error "
CASE 49 'USK_ERR_NODEV                             49
    message="NODEV Error "
CASE 54 'USK_ERR_MUTEX_DEV_CREATE                  54
    message="MUTEX_DEV_CREATE Error "
CASE 55 'USK_ERR_MUTEX_DEV_TIMEOUT                 55
    message="MUTEX_DEV_TIMEOUT Error "
CASE 56 'USK_ERR_MUTEX_DEV_ABANDONED               56
    message="MUTEX_DEV_ABANDONED Error "
CASE 57 'USK_ERR_MUTEX_DEV_UNKNOWN                 57
    message="MUTEX_DEV_UNKNOWN Error "
CASE 58 'USK_ERR_MUTEX_API_CREATE                  58
```

```
            message="MUTEX_API_CREATE Error "
        CASE 59 'USK_ERR_MUTEX_API_TIMEOUT              59
            message="MUTEX_API_TIMEOUT Error "
        CASE 60 'USK_ERR_MUTEX_API_ABANDONED            60
            message="MUTEX_API_ABANDONED Error "
        CASE 61 'USK_ERR_MUTEX_API_UNKNOWN              61
            message="MUTEX_API_UNKNOWN Error "
        CASE 63 'USK_ERR_BAD_ARGUMENTS                  63
            message="BAD_ARGUMENTS Error "
        CASE 65 'USK_ERR_SID_RETRIEVE                   65
            message="DSID_RETRIEVE Error "
        CASE 66 'USK_ERR_ACL_INIT                       66
            message="ACL_INIT Error "
        CASE 67 'USK_ERR_ACE_ADD                        67
            message="ACE_ADD Error "
        CASE 68 'USK_ERR_SD_INIT                        68
            message="SD_INIT Error "
        CASE 69 'USK_ERR_SD_SET                         69
            message="SD_SET Error "
        CASE 80 'USK_ERR_80                             80
            message="Reserved (80)"
        CASE 81 'USK_ERR_IOCTL_QTSDONGLE_SET_STRICT     81
            message="IOCTL_QTSDONGLE_SET_STRICT Error "
        CASE 82 'USK_ERR_REQUIRES_DRIVER_2_3_0          82
            message="REQUIRES_DRIVER_2_3_0 Error "
        CASE 83 'USK_ERR_REQUIRES_DONGLE_1_50           83
            message="REQUIRES_DONGLE_1_50 Error "
        CASE 84 'USK_WARN_MODE_PAGE_BOUNDARY            84
            message="WARN_MODE_PAGE_BOUNDARY Error "
        CASE 85 'USK_WARN_CRYPT_PAGE_BOUNDARY           85
            message="WARN_CRYPT_PAGE_BOUNDARY Error "
        CASE 86 'USK_RESET_COMEREADY                    86
            message="RESET_COMEREADY Error "
        CASE 255 'USK_ERR_UNKNOWN                       255
            message="Unknown Error "
        CASE ELSE
           message= "Unknown Error (2)"
    END SELECT

    MSGBOX message,%MB_ICONINFORMATION,"SmartDongle"

END FUNCTION
'=================================================================
========
'-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-------------------------
'   ** CallBacks **
'-----------------------------------------------------------------
-----------------------------------------------------------------
-----------------------------------------------------------------
-------------------------
CALLBACK FUNCTION DongleScreenProc()
LOCAL IsKeyPresent, n, errorflag,stringlength, address AS LONG
LOCAL TrialTextOut AS ASCIIZ*128
```

```
LOCAL TrialText, temp, filename, start AS STRING
LOCAL SerNum AS ASCIIZ*128



    SELECT CASE AS LONG CB.MSG
        CASE %WM_INITDIALOG
            ' Initialization handler

        CASE %WM_NCACTIVATE
            STATIC hWndSaveFocus AS DWORD
            IF ISFALSE CB.WPARAM THEN
                ' Save control focus
                hWndSaveFocus = GetFocus()
            ELSEIF hWndSaveFocus THEN
                ' Restore control focus
                SetFocus(hWndSaveFocus)
                hWndSaveFocus = 0
            END IF

        CASE %WM_COMMAND
            ' Process control notifications
            SELECT CASE AS LONG CB.CTL
            'Textboxes below copy data fron the portrait to the
landscape boxes to save retyping - allows for alterations.
                CASE %IDC_TEXTBOX2
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX2 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX17, temp

                CASE %IDC_TEXTBOX3
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX3 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX18, temp

                CASE %IDC_TEXTBOX5
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX5 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX20, temp

                CASE %IDC_TEXTBOX6
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX6 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX21, temp

                CASE %IDC_TEXTBOX7
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX7 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX22, temp

                CASE %IDC_TEXTBOX14
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX14 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX24, temp

                CASE %IDC_TEXTBOX15
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX15 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX25, temp

                CASE %IDC_TEXTBOX16
```

```
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX16 TO temp
                    CONTROL SET TEXT CB.HNDL,%IDC_TEXTBOX26, temp

                CASE %IDC_BUTTON1 'write
                    stringlength=128
                    FOR n=1001 TO 1023
                        LEDRed()
                        CONTROL SET TEXT CB.HNDL,
1000,"Writing"+STR$(n-1000)
                        address=128*(n-1001)
                        CONTROL GET TEXT CB.HNDL,n TO TrialText
                        WriteToSmartDongle TrialText,address
                        LEDOff()
                        DIALOG REDRAW CB.HNDL
                    NEXT
                    CONTROL SET TEXT CB.HNDL, 1000,""
                    LEDGreen()

                CASE %IDC_BUTTON2 'clear
                   FOR n=1001 TO 1023
                        CONTROL SET TEXT CB.HNDL,n,""
                    NEXT

                CASE %IDC_BUTTON3 'read
                    LEDRed()
                    stringlength=128
                    FOR n=1001 TO 1023
                        CONTROL SET TEXT CB.HNDL,
1000,"Reading"+STR$(n-1000)
                        address=128*(n-1001)
                        TrialTextOut=ReadSmartDongle (address)
                        CONTROL SET TEXT CB.HNDL,n,TRIM$(TrialTextOut)
                    NEXT
                    CONTROL SET TEXT CB.HNDL, 1000,""
                    LEDGreen()

                CASE %IDC_BUTTON4 'login
                    IsKeyPresent= LoginToSmartDongle (P1,P2)
                    IF IsKeyPresent<>0 THEN
                       MSGBOX "Failed to access a valid
SmartDongle.",%MB_ICONINFORMATION,"Smartdongle"
                       EXIT SELECT
                    END IF

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    SerNum=GetSerNum ()
                    CONTROL SET TEXT CB.HNDL, %IDC_TEXTBOX27, SerNum

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

                CASE %IDC_BUTTON5 'save
                    CONTROL GET TEXT CB.HNDL,%IDC_TEXTBOX10 TO start
'make filename from data in TB10
                    DISPLAY SAVEFILE CB.HNDL,,, "Save data to file", "",
"Dongle data" + CHR$(0) + "*.DNG" + CHR$(0), start,
"DNG", %OFN_CREATEPROMPT TO filename
```

```
                    'save file
                    OPEN filename FOR OUTPUT AS #1
                    FOR n=1001 TO 1023
                        address=128*(n-1001)
                        CONTROL GET TEXT CB.HNDL,n TO TrialText
                        WRITE #1,TrialText
                    NEXT
                    CLOSE #1

                CASE %IDC_BUTTON6 'load
                    DISPLAY OPENFILE CB.HNDL,,, "Load data from file",
"", "Dongle data" + CHR$(0) + "*.DNG" + CHR$(0), "*.DNG",
"DNG", %OFN_PATHMUSTEXIST TO filename
                    'load file & fill boxes
                    OPEN filename FOR INPUT AS #2
                    FOR n=1001 TO 1023
                        address=128*(n-1001)
                        INPUT #2,TrialText
                        CONTROL SET TEXT CB.HNDL,n,TrialText
                    NEXT
                    CLOSE #2

            END SELECT
    END SELECT
END FUNCTION
'--------------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-------------------------

'--------------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-------------------------
'   ** Dialogs **
'--------------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-------------------------
FUNCTION DongleScreen(BYVAL hParent AS DWORD) AS LONG
    LOCAL lRslt AS LONG

#PBFORMS BEGIN DIALOG %IDD_DIALOG1->->
    LOCAL hDlg  AS DWORD

    DIALOG NEW hParent, "SmartDongle Tool", 19, 49, 599, 388, %WS_POPUP
OR %WS_BORDER OR %WS_DLGFRAME OR %WS_SYSMENU OR %WS_CLIPSIBLINGS
OR %WS_VISIBLE OR %DS_MODALFRAME OR %DS_3DLOOK OR %DS_NOFAILCREATE
OR %DS_SETFONT, _
        %WS_EX_CONTROLPARENT OR %WS_EX_LEFT OR %WS_EX_LTRREADING
OR %WS_EX_RIGHTSCROLLBAR, TO hDlg
    CONTROL ADD FRAME,   hDlg, -1, "Portrait", 3, 78, 591, 138
    CONTROL ADD FRAME,   hDlg, -1, "Landscape", 3, 219, 591, 138

    CONTROL ADD LABEL,   hDlg, -1, "User number", 6, 6, 144, 12
```

```
    CONTROL ADD LABEL,   hDlg, -1, "HeaderTitle", 7, 144, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "HeaderAddress1", 7, 162, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Customer name", 7, 60, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headercompany1", 7, 88, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headercompany2", 7, 106, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headerwebsite", 7, 124, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "User code", 330, 6, 105, 12
    CONTROL ADD LABEL,   hDlg, -1, "BMP Name", 330, 24, 105, 12
    CONTROL ADD LABEL,   hDlg, -1, "Program number", 6, 24, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Logo height", 330, 40, 105, 12
    CONTROL ADD LABEL,   hDlg, -1, "Logo width", 6, 40, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "HeaderPhone / Address2", 9, 180,
141, 12
    CONTROL ADD LABEL,   hDlg, -1, "HeaderFax / Phone", 9, 196, 138, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headeremail", 327, 126, 108, 12
    CONTROL ADD LABEL,   hDlg, -1, "HeaderTitle", 6, 285, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "HeaderAddress1", 7, 301, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headercompany1", 7, 229, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headercompany2", 7, 247, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headerwebsite", 7, 265, 144, 12
    CONTROL ADD LABEL,   hDlg, -1, "Headeremail", 327, 267, 108, 12
    CONTROL ADD LABEL,   hDlg, -1, "Serial number", 447, 366, 63, 12
    CONTROL ADD LABEL,   hDlg, -1, "HeaderFax / Phone", 6, 337, 138, 12
    CONTROL ADD LABEL,   hDlg, -1, "HeaderPhone / Address2", 6, 320,
141, 12


    CONTROL ADD LABEL,   hDlg, 1000, "", 390, 366, 40, 12


    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX1, "11945472", 156, 3, 150,
15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX2, "Consulting Engineers",
156, 141, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX3, "HeaderAddress1", 156,
159, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX4, "Cust name", 156, 57, 435,
15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX5, "Headercompany1", 156, 87,
435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX6, "Headercompany2", 156,
104, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX7, "www.", 156, 123, 150, 14
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX8, "hydro", 441, 3, 150, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX9, "Logo.bmp", 441, 21, 150,
15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX10, "123456", 156, 21, 150,
15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX11, "120", 441, 38, 150, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX12, "456", 156, 38, 150, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX14, "01", 156, 177, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX15, "01", 156, 195, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX16, "@", 441, 123, 150, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX17, "Consulting EngineersL",
156, 282, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX18, "HeaderAddress1L", 156,
300, 435, 15
```

```
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX20, "Headercompany1L", 156,
228, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX21, "Headercompany2L", 156,
245, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX22, "www.L", 156, 264, 150,
14
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX24, "01", 156, 318, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX25, "01", 156, 336, 435, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX26, "@L", 441, 264, 150, 15
    CONTROL ADD TEXTBOX, hDlg, %IDC_TEXTBOX27, "SN", 513, 363, 78, 15

    CONTROL ADD BUTTON,  hDlg, %IDC_BUTTON4, "Log in",         6, 363,
54, 18
    CONTROL ADD BUTTON,  hDlg, %IDC_BUTTON1, "Write",         66, 363,
54, 18
    CONTROL ADD BUTTON,  hDlg, %IDC_BUTTON2, "Clear screen", 126, 363,
54, 18
    CONTROL ADD BUTTON,  hDlg, %IDC_BUTTON3, "Check memory", 186, 363,
54, 18
    CONTROL ADD BUTTON,  hDlg, %IDC_BUTTON5, "Save",         246, 363,
54, 18
    CONTROL ADD BUTTON,  hDlg, %IDC_BUTTON6, "Load",         306, 363,
54, 18

#PBFORMS END DIALOG

    DIALOG SHOW MODAL hDlg, CALL DongleScreenProc TO lRslt

#PBFORMS BEGIN CLEANUP %IDD_DIALOG1
#PBFORMS END CLEANUP

    FUNCTION = lRslt
END FUNCTION
'-------------------------------------------------------------------------
---------------------------------------
```

Queries about the above may be directed to the author - IainJohnstone@mstdrain.co.uk.