

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using Microsoft.FlightSimulator.SimConnect;
using System.Runtime.InteropServices;

namespace fsxBox
{
    public partial class Form1 : Form
    {
        //-----
        // variables globales
        //-----
        const int WM_USER_SIMCONNECT = 0x0402;
        SimConnect simconnect;
        SerialPort serialPort;
        string strBuffer = "";
        string strPort = "";
        bool cockpit = true;
        bool lightP = false;
        bool lightN = false;
        bool battery = false;
        bool alternator = false;
        //-----
        // définitions objets SimConnect
        //-----
        [StructLayout(LayoutKind.Sequential, CharSet = CharSet.Ansi, Pack = 1)]
        struct strcBool
        {
            public bool valeur;
        }
        //-----
        enum EVENTS
        {
            ENGINE,
            MIXTURE,
            PROPELLER,
            MAGNETO,
            PUSH,
            VIEW_OUT,
            VIEW_IN,
            AILERON,
            ELEVATOR,
            GEAR,
        }
    }
}

```

```

    LIGHT_P_TOGGLE,
    LIGHT_N_TOGGLE,
    BATTERY_TOGGLE,
    ALTERNATOR_TOGGLE,
    AVIONICS_TOGGLE,
    FLOAT_RETRACT,
    FLOAT_EXTEND
}
//-----
enum DEFINITIONS
{
    LIGHT_P,
    LIGHT_N,
    BATTERY,
    ALTERNATOR,
    AVIONICS,
    FLOAT
}
//-----
enum DATA_REQUESTS
{
    REQ_LIGHT_P,
    REQ_LIGHT_N,
    REQ_BATTERY,
    REQ_ALTERNATOR,
    REQ_AVIONICS
}
//-----
enum GROUPID
{
    FLAG = 1
}
//-----
// evenements SimConnect
//-----
protected override void DefWndProc(ref Message m)
{
    if (m.Msg == WM_USER_SIMCONNECT)
    {
        if (simconnect != null)
        {
            simconnect.ReceiveMessage();
        }
    }
    else
    {
        base.DefWndProc(ref m);
    }
}
}

```

```

//-----
void simconnect_OnRecvOpen(SimConnect sender, SIMCONNECT_RECV_OPEN data)
{
    displayText("Connecté à FSX");
}
//-----
void simconnect_OnRecvException(SimConnect sender, SIMCONNECT_RECV_EXCEPTION data)
{
    displayText("Exception reçue : " + data.dwException);
}
//-----
void simconnect_OnRecvQuit(SimConnect sender, SIMCONNECT_RECV data)
{
    displayText("FSX s'est arrêté");
    closeConnection();
}
//-----
void simconnect_OnRecvSimobjectDataBytype(SimConnect sender, SIMCONNECT_RECV_SIMOBJECT_DATA_BYTYPE data)
{
    bool arg;
    switch ((DATA_REQUESTS)data.dwRequestID)
    {
        case DATA_REQUESTS.REQ_LIGHT_P:
            arg = ((strcBool)data.dwData[0]).valeur;
            if (arg != lightP)
            {
                simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.LIGHT_P_TOGGLE,
                    (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
            }
            break;
        case DATA_REQUESTS.REQ_LIGHT_N:
            lightN = ((strcBool)data.dwData[0]).valeur;
            if (lightN == lightP)
            {
                simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.LIGHT_N_TOGGLE,
                    (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
            }
            break;
        case DATA_REQUESTS.REQ_BATTERY:
            arg = ((strcBool)data.dwData[0]).valeur;
            if (arg != battery)
            {
                simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.BATTERY_TOGGLE,
                    (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
            }
            break;
        case DATA_REQUESTS.REQ_ALTERNATOR:
            alternator = ((strcBool)data.dwData[0]).valeur;
            if (alternator == battery)
    }
}

```

```

        {
            simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.ALTERNATOR_TOGGLE,
                                           (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
        }
        break;
    case DATA_REQUESTS.REQ_AVIONICS:
        arg = ((strcBool)data.dwData[0]).valeur;
        if (!arg)
        {
            simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.AVIONICS_TOGGLE,
                                           (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
        }
        break;
    }
}
//-----
private void initComms()
{
    try
    {
        // initialisation : ouverture, fermeture et exceptions
        simconnect.OnRecvOpen += new SimConnect.RecvOpenEventHandler(simconnect_OnRecvOpen);
        simconnect.OnRecvQuit += new SimConnect.RecvQuitEventHandler(simconnect_OnRecvQuit);
        simconnect.OnRecvException += new SimConnect.RecvExceptionHandler(simconnect_OnRecvException);
        // liste des définitions
        simconnect.AddToDataDefinition(DEFINITIONS.LIGHT_P, "LIGHT PANEL", "bool",
                                       SIMCONNECT_DATATYPE.INT32, 0.0f, SimConnect.SIMCONNECT_UNUSED);
        simconnect.AddToDataDefinition(DEFINITIONS.LIGHT_N, "LIGHT NAV", "bool",
                                       SIMCONNECT_DATATYPE.INT32, 0.0f, SimConnect.SIMCONNECT_UNUSED);
        simconnect.AddToDataDefinition(DEFINITIONS.BATTERY, "ELECTRICAL MASTER BATTERY", "bool",
                                       SIMCONNECT_DATATYPE.INT32, 0.0f, SimConnect.SIMCONNECT_UNUSED);
        simconnect.AddToDataDefinition(DEFINITIONS.ALTERNATOR, "GENERAL ENG MASTER ALTERNATOR:1", "bool",
                                       SIMCONNECT_DATATYPE.INT32, 0.0f, SimConnect.SIMCONNECT_UNUSED);
        simconnect.AddToDataDefinition(DEFINITIONS.AVIONICS, "AVIONICS MASTER SWITCH", "bool",
                                       SIMCONNECT_DATATYPE.INT32, 0.0f, SimConnect.SIMCONNECT_UNUSED);
        // définitions des structures de variables
        simconnect.RegisterDataDefineStruct<strcBool>(DEFINITIONS.LIGHT_P);
        simconnect.RegisterDataDefineStruct<strcBool>(DEFINITIONS.LIGHT_N);
        simconnect.RegisterDataDefineStruct<strcBool>(DEFINITIONS.BATTERY);
        simconnect.RegisterDataDefineStruct<strcBool>(DEFINITIONS.ALTERNATOR);
        simconnect.RegisterDataDefineStruct<strcBool>(DEFINITIONS.AVIONICS);
        // liste des evenements : EventID
        simconnect.MapClientEventToSimEvent(EVENTS.ENGINE, "THROTTLE_SET");
        simconnect.MapClientEventToSimEvent(EVENTS.MIXTURE, "MIXTURE_SET");
        simconnect.MapClientEventToSimEvent(EVENTS.PROPELLER, "AXIS_PROPELLER_SET");
        simconnect.MapClientEventToSimEvent(EVENTS.MAGNETO, "TOGGLE_ALL_STARTERS");
        simconnect.MapClientEventToSimEvent(EVENTS.PUSH, "TOGGLE_PUSHBACK");
        simconnect.MapClientEventToSimEvent(EVENTS.VIEW_OUT, "VIEW_MODE");
        simconnect.MapClientEventToSimEvent(EVENTS.VIEW_IN, "VIEW_MODE_REV");
    }
}

```

```

simconnect.MapClientEventToSimEvent(EVENTS.AILERON, "AILERON_SET");
simconnect.MapClientEventToSimEvent(EVENTS.ELEVATOR, "ELEVATOR_SET");
simconnect.MapClientEventToSimEvent(EVENTS.GEAR, "GEAR_SET");
simconnect.MapClientEventToSimEvent(EVENTS.LIGHT_P_TOGGLE, "PANEL_LIGHTS_TOGGLE");
simconnect.MapClientEventToSimEvent(EVENTS.LIGHT_N_TOGGLE, "TOGGLE_NAV_LIGHTS");
simconnect.MapClientEventToSimEvent(EVENTS.BATTERY_TOGGLE, "TOGGLE_MASTER_BATTERY");
simconnect.MapClientEventToSimEvent(EVENTS.ALTERNATOR_TOGGLE, "TOGGLE_MASTER_ALTERNATOR");
simconnect.MapClientEventToSimEvent(EVENTS.FLOAT_RETRACT, "RETRACT_FLOAT_SWITCH_DEC");
simconnect.MapClientEventToSimEvent(EVENTS.FLOAT_EXTEND, "RETRACT_FLOAT_SWITCH_INC");
simconnect.MapClientEventToSimEvent(EVENTS.AVIONICS_TOGGLE, "TOGGLE_AVIONICS_MASTER");
// liste des requetes
simconnect.OnRecvSimobjectDataBytype += new SimConnect.RecvSimobjectDataBytypeEventHandler(simconnect_OnRecvSimobjectDataBytype);
}
catch (COMException ex)
{
    displayText(ex.Message);
}
}
//-----
public Form1()
{
    InitializeComponent();
    setButtons(true, false, true);
    cbPort.Items.AddRange(SerialPort.GetPortNames());
    if (cbPort.Items.Count > 0)
    {
        cbPort.SelectedIndex = cbPort.Items.Count - 1;
    }
}
//-----
private void closeConnection()
{
    if (simconnect != null)
    {
        simconnect.Dispose();
        simconnect = null;
    }
    if (serialPort.IsOpen) serialPort.Close();
}
//-----
private void displayText(string message)
{
    listBox.Items.Insert(0, message + "\n");
}
//-----
private void setButtons(bool bConnect, bool bDisconnect, bool port)
{
    bConnexion.Enabled = bConnect;
    bDeconnexion.Enabled = bDisconnect;
}

```

```

    cbPort.Enabled = port;
}
//-----
private void bConnexion_Click(object sender, EventArgs e)
{
    simconnect = null;
    string c = cbPort.SelectedItem.ToString();
    if (c.Equals(""))
    {
        displayText("Port com absent");
        return;
    }
    serialPort = new SerialPort(c, 9600);
    serialPort.DataReceived += new SerialDataReceivedEventHandler(serialPort_DataReceived);
    try
    {
        serialPort.Open();
        simconnect = new SimConnect("Essai", this.Handle, WM_USER_SIMCONNECT, null, 0);
        initComms();
        setButtons(false, true, false);
    }
    catch (COMException ex)
    {
        displayText("erreur de connexion avec FSX : " + ex.Message);
        if (serialPort.IsOpen) serialPort.Close();
    }
}
//-----
private void bDeconnexion_Click(object sender, EventArgs e)
{
    closeConnection();
    setButtons(true, false, true);
    displayText("Déconnecté de FSX");
}
//-----
private void serialPort_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    strBuffer = serialPort.ReadExisting();
    this.Invoke(new EventHandler(lecturePort));
}
//-----
private void lecturePort(object sender, EventArgs e)
{
    if (strBuffer.Length < 1) return;
    strPort += strBuffer;
    int l = strPort.Length;
    if (strPort[l - 1] != '.' && l < 200) return;
    string nbrStr = "0";
    bool debut = false;
}

```

```

bool neg = false;
for (int f = 0; f < strPort.Length; f++)
{
    if (strPort[f] == 32) // char = espace : début séquence
    {
        nbrStr = "0";
        debut = true;
    }
    if (strPort[f] == 45) // char = signe négatif
    {
        neg = true;
    }
    else if (strPort[f] > 47 && strPort[f] < 58)
    {
        nbrStr += (char)strPort[f];
    }
    else if (strPort[f] == 88 && debut) // char = X : fin de la sequence joystick x
    {
        uint val = Convert.ToUInt32(nbrStr) * 12000 / 512;
        if (!neg) val = UInt32.MaxValue - val;
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.AILERON,
            val, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

        debut = false;
        neg = false;
    }
    else if (strPort[f] == 89 && debut) // char = Y : fin de la sequence joystick y
    {
        uint val = Convert.ToUInt32(nbrStr) * 9000 / 512;
        if (!neg) val = UInt32.MaxValue - val;
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.ELEVATOR,
            val, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

        debut = false;
        neg = false;
    }
    else if (strPort[f] == 77 && debut) // char = M : fin de la sequence moteur
    {
        uint val = Convert.ToUInt32(nbrStr) * 16383 / 1023;
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.ENGINE,
            val, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

        debut = false;
        neg = false;
    }
    else if (strPort[f] == 82 && debut) // char = R : fin de la sequence richesse
    {
        uint val = (Convert.ToUInt32(nbrStr) * 16383 / 1023);
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.MIXTURE,
            val, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

        debut = false;
        neg = false;
    }
}

```

```

}
else if (strPort[f] == 80 && debut)    // char = P : fin de la sequence pas d'hélice
{
    uint val = Convert.ToUInt32(nbrStr) * 16383 / 512;
    if (neg) val = UInt32.MaxValue - val;
    simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.PROPELLER,
        val, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

    debut = false;
    neg = false;
}
else if (strPort[f] == 65 && debut)    // char = A : fin de la sequence bouton A
{
    lightP = Convert.ToUInt32(nbrStr) == 1;
    simconnect.RequestDataOnSimObjectType(DATA_REQUESTS.REQ_LIGHT_P, DEFINITIONS.LIGHT_P, 0, SIMCONNECT_SIMOBJECT_TYPE.USER);
    simconnect.RequestDataOnSimObjectType(DATA_REQUESTS.REQ_LIGHT_N, DEFINITIONS.LIGHT_N, 0, SIMCONNECT_SIMOBJECT_TYPE.USER);
    debut = false;
    neg = false;
}
else if (strPort[f] == 66 && debut)    // char = B : fin de la sequence bouton B
{
    battery = Convert.ToUInt32(nbrStr) == 1;
    simconnect.RequestDataOnSimObjectType(DATA_REQUESTS.REQ_BATTERY, DEFINITIONS.BATTERY, 0, SIMCONNECT_SIMOBJECT_TYPE.USER);
    simconnect.RequestDataOnSimObjectType(DATA_REQUESTS.REQ_ALTERNATOR, DEFINITIONS.ALTERNATOR, 0, SIMCONNECT_SIMOBJECT_TYPE.USER);
    simconnect.RequestDataOnSimObjectType(DATA_REQUESTS.REQ_AVIONICS, DEFINITIONS.AVIONICS, 0, SIMCONNECT_SIMOBJECT_TYPE.USER);
    debut = false;
    neg = false;
}
else if (strPort[f] == 67 && debut)    // char = C : fin de la sequence bouton C
{
    bool val = Convert.ToUInt32(nbrStr) == 1;
    if (val)
    {
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.FLOAT_EXTEND,
            (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.FLOAT_EXTEND,
            (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
    }
    else
    {
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.FLOAT_RETRACT,
            (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.FLOAT_RETRACT,
            (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
    }
    debut = false;
    neg = false;
}
else if (strPort[f] == 68 && debut)    // char = D : fin de la sequence bouton D (GEAR)
{

```



```

        uint val = Convert.ToUInt32(nbrStr);
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.GEAR,
                                       val, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

        debut = false;
        neg = false;
    }
    else if (strPort[f] == 69 && debut)    // char = E : fin de la sequence push-bouton E (MAGNETO)
    {
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.MAGNETO,
                                       (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

        debut = false;
        neg = false;
    }
    else if (strPort[f] == 70 && debut)    // char = F : fin de la sequence push-bouton F (PUSH BACK)
    {
        simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.PUSH,
                                       (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);

        debut = false;
        neg = false;
    }
    else if (strPort[f] == 71 && debut)    // char = G : fin de la sequence push-bouton G (VIEW/INIT)
    {
        if (cockpit)
        {
            simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.VIEW_OUT,
                                           (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
        }
        else
        {
            simconnect.TransmitClientEvent((uint)SimConnect.SIMCONNECT_OBJECT_ID_USER, EVENTS.VIEW_IN,
                                           (uint)0, GROUPID.FLAG, SIMCONNECT_EVENT_FLAG.GROUPID_IS_PRIORITY);
        }
        cockpit = !cockpit;
        debut = false;
        neg = false;
    }
}
strPort = "";
}
//-----
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (bDeconnexion.Enabled) closeConnection();
}
//-----
}
}

```