

Lecture des joysticks sous Windows

Note : *Les informations de cette note d'application sont le résultat de notre compréhension de l'utilisation de la librairie mmsystem.
Ce ne saurait être une vérité absolue et ne nous engage aucunement !*

La lecture des joysticks sous Windows utilise une dll standard :

- 'winmm.dll' sur les versions récentes
- 'mmsystem.dll' sur les versions plus anciennes (XP).

Fonctions les plus utiles :

La liste est non exhaustive.

- JoySetCapture : Capture d'un joystick
- JoyReleaseCapture : Libération d'un joystick
- joyGetDevCaps : Retourne les caractéristiques du joystick
- joyGetNumDevs : Nombre de joysticks connectés
- joyGetPos : Retourne les positions de 3 axes :
X, Y, Z et des boutons
- joyGetPosEx : Retourne les positions de 7 axes :
X, Y, Z, R, U, V, POV et des boutons
- les autres fonctions sont à découvrir sur le site de Microsoft

Pour tous les détails, il suffit de faire une recherche sur internet avec l'une de ces fonctions.

Delphi ou Lazarus

Sous Delphi ou Lazarus, il faut inclure l'unité MMSYSTEM.

Langage C

Les procédures pour interroger un joystick quelconque sous windows sont décrites sur le site de microsoft :

<https://learn.microsoft.com/en-us/windows/win32/api/joystickapi/>

Une version plus détaillée en français est disponible :

<https://tcharles.developpez.com/joystick/>



Nombre de joysticks

Jusqu'à 16 joysticks peuvent être utilisés en même temps.

Le nombre de joysticks connectés est obtenu en appelant la fonction
`UINT joyGetNumDevs();`

Windows 10 / XP	Les valeurs valides pour uJoyID vont de 0 (JOYSTICKID1) à 15
Windows NT 4.0	Les valeurs valides sont limitées à JOYSTICKID1 et JOYSTICKID2.

*Attention : la dll ne différencie pas les interfaces Joystick des interfaces HID.
Un périphérique composite pourrait utiliser plusieurs identifiants.*

Etapes à suivre

- **Trouver l'identifiant du joystick** : Par défaut, Windows ne donne pas d'information pour savoir quel joystick correspond à quel identifiant. Ça ne pose pas de problème avec un seul joystick, ce qui est le cas la plupart du temps. Alors uJoyID = 0. Mais si on a plusieurs joysticks, on ne sait pas dans quel ordre ils sont attribués ni si cet ordre sera permanent !
Voir notre exemple en Lazarus :
Trouver mon joystick parmi d'autres joysticks
- **Capturer le joystick** : JoySetCapture
- **Acquérir les déplacements et les boutons** :
Faire une boucle ou utiliser un timer pour acquérir les données du joystick : joyGetPos ou joyGetPosEx
- **Libérer le joystick** : JoyReleaseCapture

Trouver mon joystick parmi d'autres joysticks

On peut faire une interrogation de tous les joysticks présents avec la fonction **joyGetDevCaps** pour rechercher un joystick

- avec un **VID** et un **PID** particulier
- mais aussi avec un nombre d'**axes** ou de **boutons** particuliers, ce qui permet aussi d'éliminer les interfaces HID

Avec cette méthode, on ne peut pas distinguer 2 joysticks identiques !

```
program TryMyJoystick;
// Example in Lazarus free pascal

uses MMSYSTEM, SysUtils, Crt;
var uJoyID:Integer;

function FindMyJoystick(MyVID, MyPID: word; const MyAxes: integer = -1;
const MyButtons: integer = -1): integer;
var
  c,JoyVID, JOYPID, joyNumber,JoyAxes, JoyButtons: word;
  uJoyID: integer;
  InfosCaps : TJOYCAPS ;
begin
  // Get Number of connected joystick
  JoyNumber := joyGetNumDevs;
  // Initialize uJoyID
  uJoyID := -1;

  // check every joystick (remember Joystick #1's ID = 0, etc.)
  c := 0;
  while (c < JoyNumber - 1) do
  begin
    // try to read information from joyGetDevCaps
    if joyGetDevCaps(c, @InfosCaps, SizeOf(InfosCaps)) = JOYERR_NOERROR then
    begin
      JoyVID := InfosCaps.wMid; // Get Vendor ID
      JoyPID := InfosCaps.wPid; // Get Product ID
      JoyAxes := InfosCaps.wNumAxes; // Get Axes Quantity
      JoyButtons := InfosCaps.wNumButtons; // Get Buttons Quantity
      // is it my Joystick ?
      // Check if VID and PID are identical
      if (JoyVID = MyVID) and (JoyPID = MyPID)
      // Optionnally check number of Axes is identical
      and ((MyAxes < 0) or (JoyAxes = MyAxes))
      // Optionnally check number of Buttons is identical
      and ((MyButtons < 0) or (JoyButtons = MyButtons))
      then uJoyID := c; // Select This Joystick
    end;
    Inc(c); // Try Next Joystick
  end;

  Result := uJoyID;
end;

begin
  uJoyID:=FindMyJoystick($25c7,$0119,3,12); // VID , PID, 3 Axes, 12 Buttons
  writeln (format('Joystick #&d',[uJoyID]));
  Readln;
end.
```

Acquisition d'un joystick

```

program TryMyJoystick;
// Example in Lazarus free pascal

uses MMSYSTEM, SysUtils, Crt;

var uJoyID: Integer;
    MyJoy: TJoyInfoEx;
    JoyHandle: THandle;

function FindMyJoystick(MyVID, MyPID: word; const MyAxes: integer = -1;
...

procedure ReadJoystick(uJoyID: integer);
var
    Buttons: integer;
    AxeX, AxeY, AxeZ: integer;
    ErrorResult: MMResult;
begin
    if (uJoyID < -1)
    then ErrorResult := joyGetPosEx(uJoyID, @MyJoy)
    else ErrorResult := 255;

    if ErrorResult = JOYERR_NOERROR then
    begin
        AxeX := MyJoy.wxpos; // Read X position
        AxeY := MyJoy.wypos; // Read Y Position
        AxeZ := MyJoy.wzpos; // Read Z Position
        Buttons := MyJoy.wButtons; // Read Buttons
        writeln(format('X=%d, Y=%d, Z=%d, Buttons:%d ', [AxeX, AxeY, AxeZ, Buttons]));
    end;
end;

procedure CaptureJoystick(uJoyID: integer);
begin
    case JoySetCapture(JoyHandle, uJoyID, 0, False) of
        JOYERR_NOERROR:    Writeln('JOYERR_NOERROR ');
        MMSYSERR_NODRIVER: Writeln('MMSYSERR_NODRIVER');
        JOYERR_PARMS:      Writeln('JOYERR_PARMS ');
        JOYERR_NOCANDO:     Writeln('JOYERR_NOCANDO ');
        JOYERR_UNPLUGGED:   Writeln('JOYERR_UNPLUGGED');
        else
            Writeln('???');
        end;
    end;
end;

begin
    uJoyID := FindMyJoystick($25c7, $0119, 3, 12); // VID , PID, 3 Axes, 12 Buttons
    writeln(format('Joystick #d', [uJoyID]));
    if uJoyID >= 0 then
    begin
        MyJoy.dwSize := 65535;
        MyJoy.dwFlags := JOY_RETURNX OR JOY_RETURNX OR JOY_RETURNX OR JOY_RETURNX;
        // MyJoy.dwFlags := JOY_RETURNALL;
        CaptureJoystick(uJoyID);
        repeat
            ReadJoystick(uJoyID);
            delay(50); // pause 50 ms
        until keypressed;
        JoyReleaseCapture(uJoyID);
    end;
end.

```