

# ウィンドウズのウィンドウをコマンドラインで操作する

ツイッタのTLで見かけた『民安おぺれーたあ』ってなんぞや？と見てみるとVOICEROIDにメッセージを送ってしゃべらせるアプリ(というかブリッジ的なもの?) ニコニコ生放送とかでコメントを読ませるのに使らしい。

いろいろなアプリがあるようで、眺めてみると、どうもウィンドウを捕まえてそこからテキストボックスにメッセージを書き込んで再生ボタンを押す、ってなことをやってるようです。

久しくこの手のアプリを作っていないので、リハビリを兼ねて真似っこして見る事にしました。

2016/09/14

実際に動くものを[echoseika,sasarasay 関連](#)のページで公開しました。

2012/09/28 書き始め

## アプリのウィンドウハンドルを捕まえる

手元には「VOICEROID+民安ともえ」がありますので、まずこれをプログラムから捕まえる事をやってみます。開発環境は Visual Studio 2010 でC#を使ってみます。

<http://www.atmarkit.co.jp/fdotnet/dotnettips/233enumwin/enumwin.html> を参照してウィンドウ一覧を出せるか試してみました。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;

namespace echotammy
{
    class echotummy
    {
        static void Main(string[] args)
        {
            Process[] ps = Process.GetProcesses();
            foreach (Process pitem in ps)
            {
                if (pitem.MainWindowHandle != IntPtr.Zero)
                {
                    Console.WriteLine("[{0}:{1}] {2}", pitem.Id,
pitem.ProcessName, pitem.MainWindowTitle);
                }
            }

            return;
        }
    }
}
```

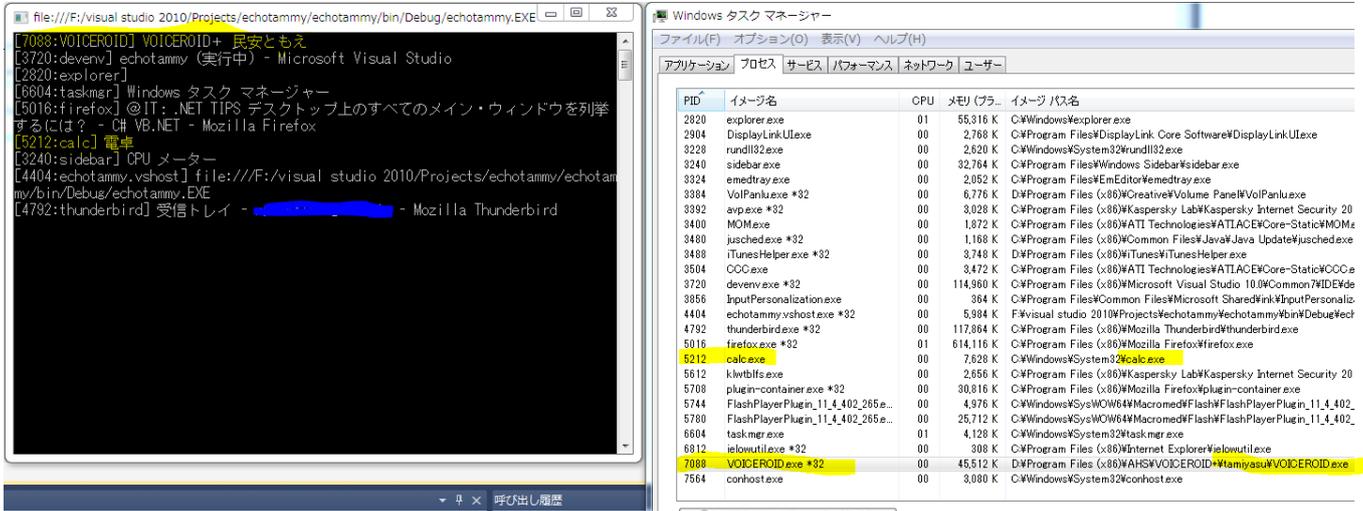
```

}
}
}

```

VS2010でコンソールアプリケーションを選んだ際の雛形にそのまま書き込んでます。

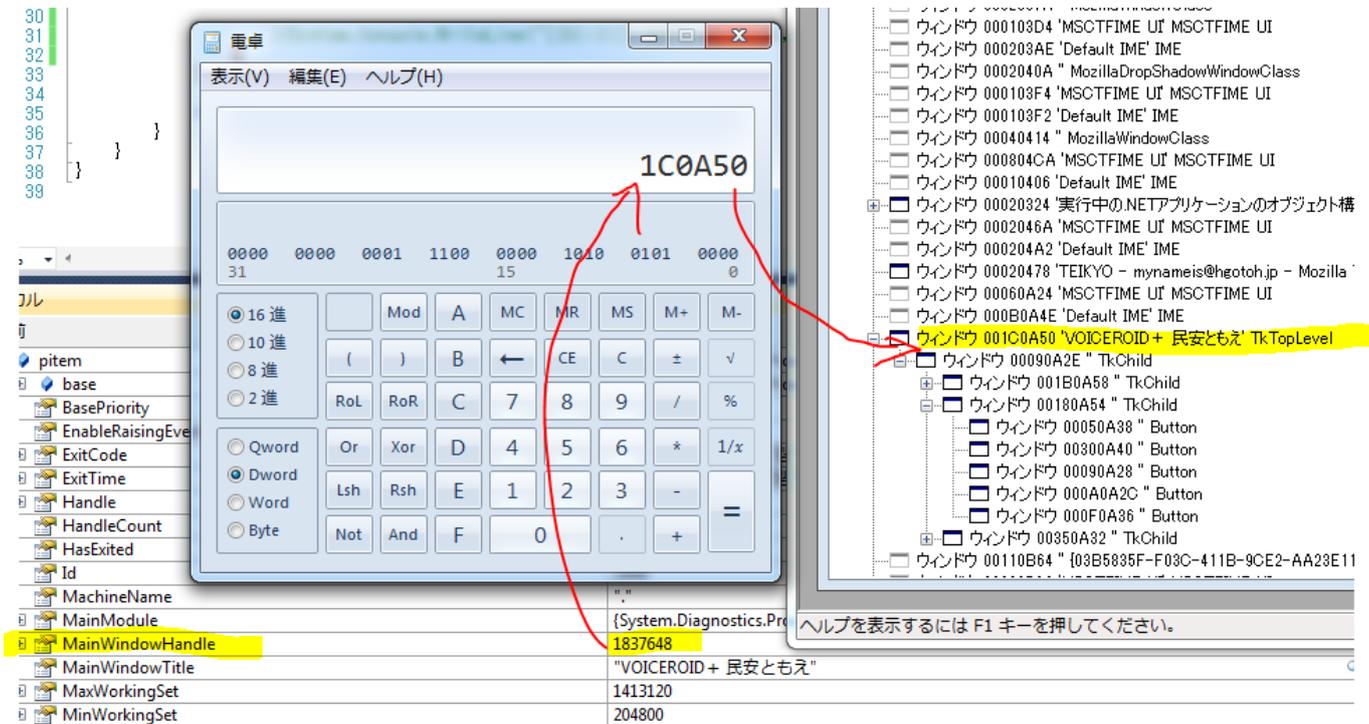
民安さんを起動してから実行します。結果は、ご覧の通り。



64bitのWindows7で実行してありますが32bitアプリのウィンドウも捕まえられる模様です。

とりあえず、ウィンドウタイトルの“VOICEROID 民安ともえ”でアプリ特定する事にします。まあ、ウィンドウタイトルが同じ別アプリも引っ掛けてしまうのももう少し絞らないといけないのですが後で対応しましょう。

そしてウィンドウハンドルをSpy++で確認してみると、



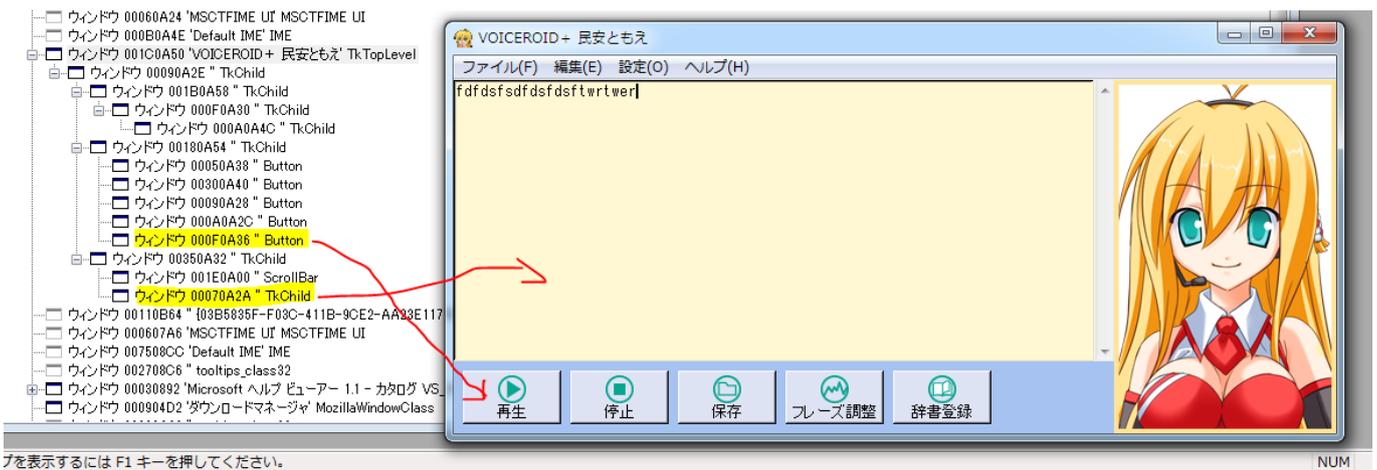
無事ウィンドウハンドルを取れたことがわかりました。

## アプリのコントロールウィンドウを捕まえる

“VOICEROID 民安ともえ” のウィンドウにはテキストを入力するボックスがあって、再生の為のボタンがあって、その他色々コントロールが配置されています。今回の意図では、少なくとも「テキストボックス」「再生ボタン」の2コントロールをいじれるようにならなくてはなりません。

このコントロールは、アプリのウィンドウの子ウィンドウとして存在するので、各コントロールのウィンドウ(ウィンドウハンドル)を捕まえる必要があります。

<http://www.atmarkit.co.jp/fdotnet/dotnettips/605managedspy/managedspy.html> を参考にしました。VisualStudioについてくる Spy++ のウィンドウ検索でファインダーツールを使うと、わりあい楽に見つけられます。



ファインダーツールを調べたいコントロールの上へドラッグ&ドロップすれば、ウィンドウハンドルが判明します。すると、2つのコントロールが存在する箇所がわかりました。

テキストボックスのコントロールはクラスTkChildで、親ウィンドウ下の子ウィンドウ下の3つ目の子ウィンドウ下の2つ目の子ウィンドウ

再生ボタンはクラスButtonで、親ウィンドウ下の子ウィンドウ下の2つ目の子ウィンドウ下の5つ目の子ウィンドウ

この2つのウィンドウハンドルを捕まえればいいわけです。

さて、どうやって実際のウィンドウハンドルを捕まえるかですが..... C#というか.NET Framework だけでは取得できない模様です。Windows API を呼び出しせねばなりません。

<http://blogs.itmedia.co.jp/mtaneda/2012/02/c-ba41.html> をみて呼び出しを試みます。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Runtime.InteropServices;

namespace echotammy
{
    class echotummy
    {
```

```
delegate void WNDENUMPROC(IntPtr hwnd, int lParam);

[DllImport("user32.dll")]
private static extern int EnumChildWindows(IntPtr hWnd, WNDENUMPROC
lpEnumFunc, int lParam);

static void CALLBACKFUNC(IntPtr hwnd, int lParam)
{
    Console.WriteLine("{0}", hwnd.ToString("X8"));
}

static void Main(string[] args)
{
    string titleStr = "VOICEROID 民安ともえ";

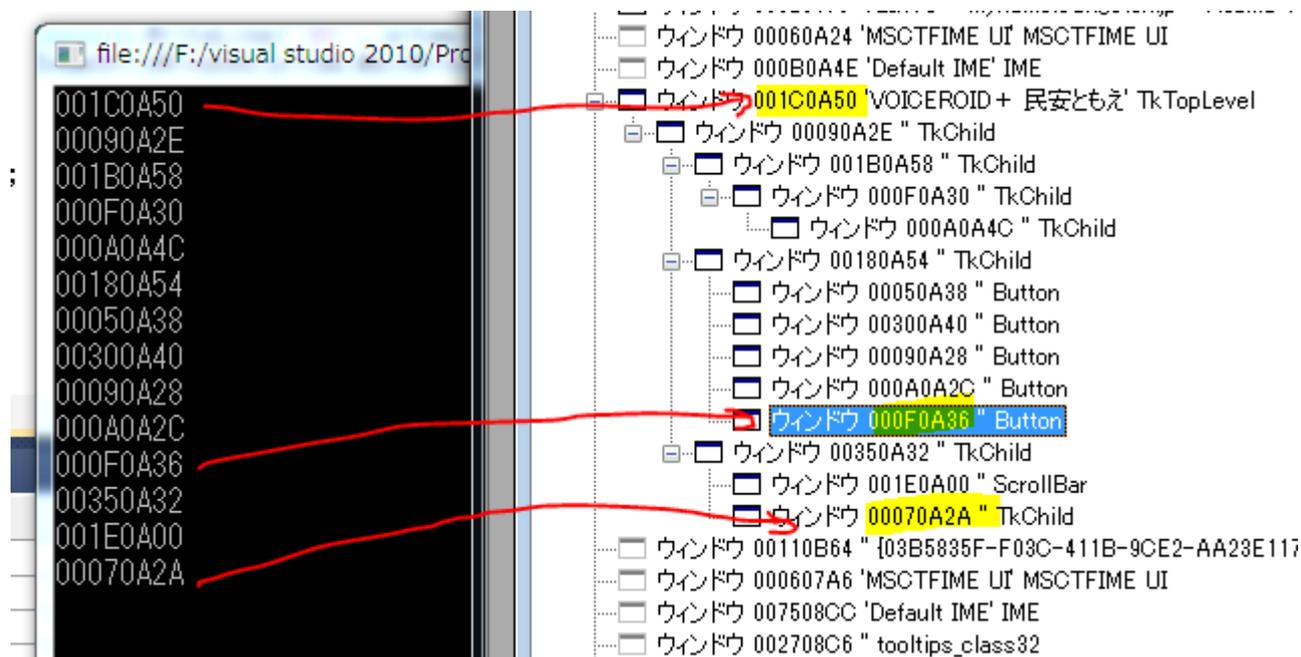
    Process[] ps = Process.GetProcesses();
    WNDENUMPROC callback = CALLBACKFUNC;

    foreach (Process pitem in ps)
    {
        if ((pitem.MainWindowHandle !=
IntPtr.Zero)&&(pitem.MainWindowTitle == titleStr))
        {
            Console.WriteLine("{0}",
pitem.MainWindowHandle.ToString("X8"));
            EnumChildWindows(pitem.MainWindowHandle, callback, 0);
        }
    }

    return;
}
}
```

EnumChildWindows()を使ってウインドウハンドルを列挙してみました。"DllImport("user32.dll)"だと32bitのプラットフォームオンリーじゃないの?と思ったのですが、64bitのプラットフォームには32bit版 user32.dll と 64bit版 user32.dll があって実行時に適宜選択されるそうです。

で、実行結果はこれ。



今回はうまい具合にSpy++の表示と同じ並びで取り出せてるみたいですが、何かUIに変更があるとこの並びも変わるでしょうし、そもそも毎回この並びになる保証もありません。別の手がいるな。

Webをうろろしたところ、まさにやりたいことが

[http://blog.goo.ne.jp/masaki\\_goo\\_2006/e/7421fd026d0aa6b375fbbd8dedeea881](http://blog.goo.ne.jp/masaki_goo_2006/e/7421fd026d0aa6b375fbbd8dedeea881) で見つかったのであとはWindowsSDKのマニュアルを見てコードを書き換えます。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Runtime.InteropServices;

namespace echotammy
{
    class echotummy
    {
        [DllImport("user32.dll")]
        private static extern IntPtr GetWindow(IntPtr hWnd, uint uCmd);

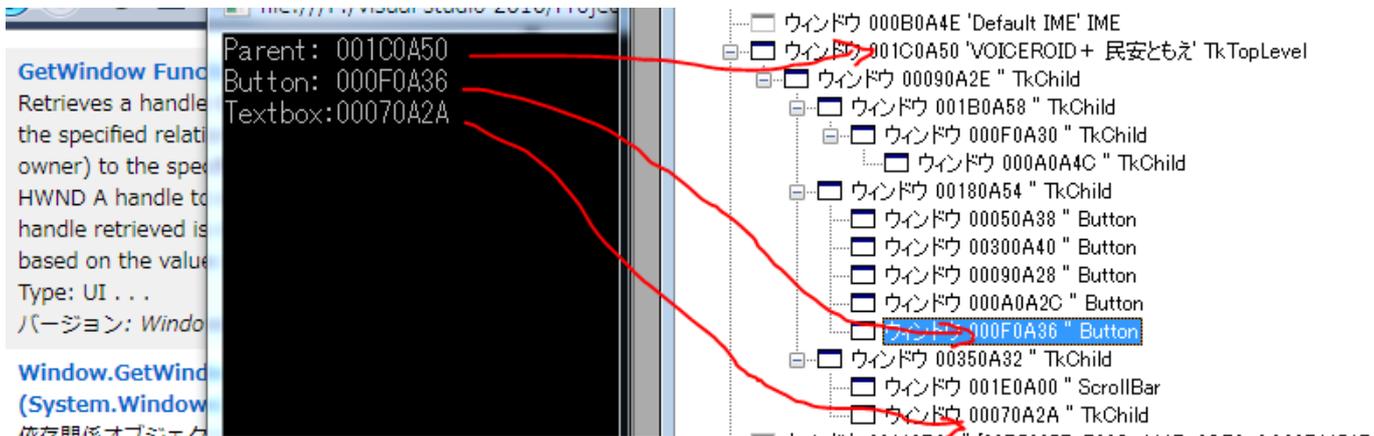
        const uint GW_HWNDNEXT = 2;
        const uint GW_CHILD = 5;

        static void Main(string[] args)
        {
            string titleStr = "VOICEROID 民安ともえ";

            Process[] ps = Process.GetProcesses();

            foreach (Process pitem in ps)
            {
                if ((pitem.MainWindowHandle !=
                    IntPtr.Zero)&&(pitem.MainWindowTitle == titleStr))
```





これではメッセージを送るのみとなりました。

## 民安さんにしゃべくりしてもらおう

### 再生ボタンを押してみる

以下はボタンが押せましたよ版。実行前に民安さんを起動してテキストボックスにテキストを入力して置いてください。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Runtime.InteropServices;

namespace echotammy
{
    class echotummy
    {
        [DllImport("user32.dll")]
        private static extern IntPtr GetWindow(IntPtr hWnd, uint uCmd);

        [DllImport("user32.dll")]
        public static extern int PostMessage(IntPtr hWnd, uint Msg, int wParam, int lParam);

        const uint GW_HWNDNEXT = 2;
        const uint GW_CHILD = 5;
        const uint WM_NULL = 0;

        const string titleStr = "VOICEROID 民安ともえ";

        static void Main(string[] args)
        {
            Process[] ps = Process.GetProcesses();
```

```

        foreach (Process pitem in ps)
        {
            if ( (pitem.MainWindowHandle !=
IntPtr.Zero)&&(pitem.MainWindowTitle.Equals(titleStr)) )
            {
                IntPtr hWndButton, hWndTextbox, hWndWorkPtr;

                hWndWorkPtr = pitem.MainWindowHandle;

                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_CHILD); // 子ウイ
ンドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_CHILD); // 1番目孫
ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 2番
目孫ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_CHILD); // 2番目孫
ウインドウで1番目の子ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 2番
目孫ウインドウで2番目の子ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 2番
目孫ウインドウで3番目の子ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 2番
目孫ウインドウで4番目の子ウインドウ
                hWndButton = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 2番目
孫ウインドウで5番目の子ウインドウ

                hWndWorkPtr = pitem.MainWindowHandle;

                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_CHILD); // 子ウイ
ンドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_CHILD); // 1番目孫
ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 2番
目孫ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 3番
目孫ウインドウ
                hWndWorkPtr = GetWindow(hWndWorkPtr, GW_CHILD); // 3番目孫
ウインドウで1番目の子ウインドウ
                hWndTextbox = GetWindow(hWndWorkPtr, GW_HWNDNEXT); // 3番
目孫ウインドウで2番目の子ウインドウ

                PostMessage(hWndButton, WM_NULL, 0, 0);

                //Console.WriteLine("Parent: {0}",
pitem.MainWindowHandle.ToString("X8"));
                //Console.WriteLine("Button: {0}",
hWndButton.ToString("X8"));
                //Console.WriteLine("Textbox:{0}",
hWndTextbox.ToString("X8"));
            }
        }

```

```

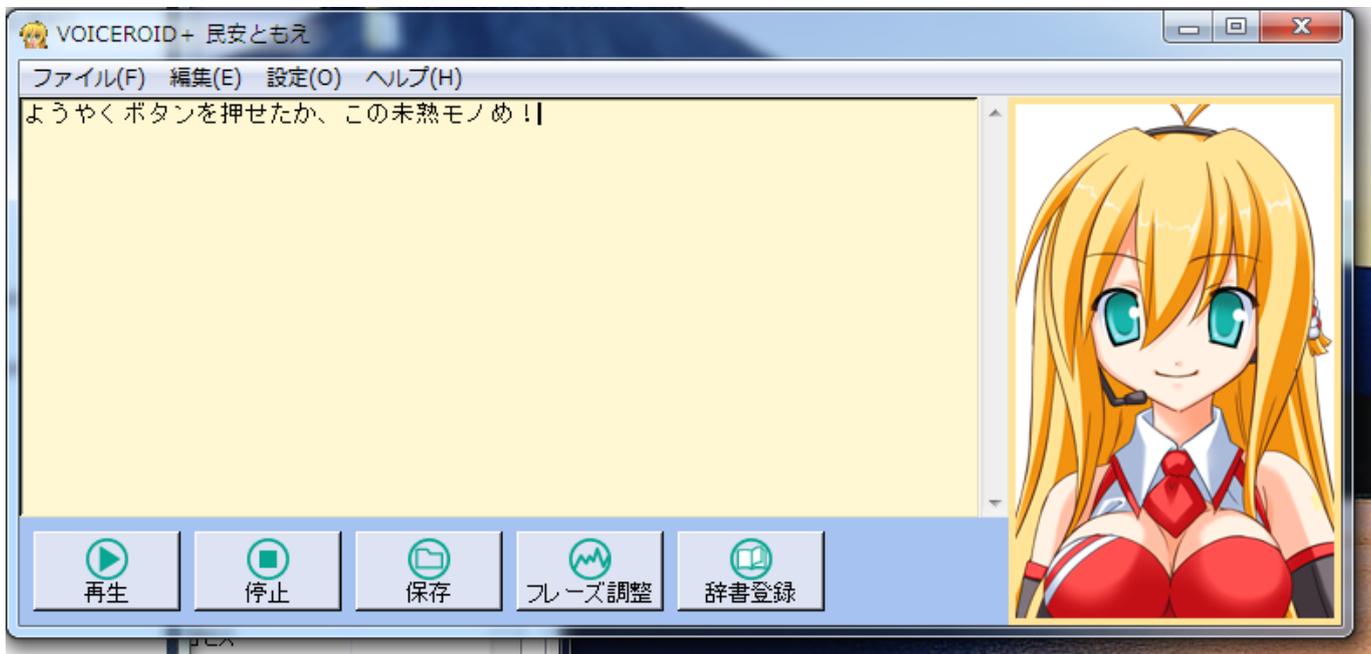
return;
}
}
}

```

再生ボタンのウインドウハンドルも捕まえたし、あとはBM\_CLICKメッセージ送って (・ω)んばんじゃーい！だと思ってたんですがねえ。

ウキウキしながらテキストボックスにメッセージを入れてSpy++で監視していたところ、そもそもBM\_CLICKが再生ボタンに送られてきていない。SetFocus()したらBM\_CLICKが来たけどそもそも処理してない模様。

なんとなくだけど、親コントロールがマウスカースルの位置とボタンクリックを検出して処理をしているっぽい。「あきらめたら以下省略」という事でグーグル先生をこき使う。すると <http://codetter.com/?p=623> でPythonをつかって同じような事をしている人がいまして「WM\_NULLだけでいけるみたい」のコメントが。...やった！！ 先生！民安さんがお話してくれたよ！



技術資料, Windows, voicerooid, tool

From: <https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link: <https://wiki.hgotoh.jp/documents/csharp/code-005>

Last update: 2024/11/02 13:39

