

3. まずは実験してみよう

デザイン後送

3.1

WiiRemoteとPCをBluetoothで接続する

ここでは前節で紹介したいくつかのBluetoothスタックを使って、実際にWiiRemoteをPCに接続する方法をステップバイステップで解説していきます。すでにお使いのBluetooth環境で問題なく接続できている読者は読み飛ばして、次節のツールを使った実験に進んでいただいてもかまいませんが、初めて実験するときや新しく買ったBluetoothアダプタを利用する場合は、ぜひ関連する解説を一読することをお勧めします。ドライバ付属のウィザードだけではどうしても接続できない「ちょっとしたコツ」が必要となることがあります。

Bluetoothをインストールする際の注意

現在、多くのBluetoothホストアダプタはUSBインタフェースを採用しています。インストールは製品付属のマニュアルをよく読んで行ってください。一般的なUSBメモリなどと異なり、Bluetoothホストアダプタは、初めて挿入する「前に」、付属ソフトウェアのインストールが必要になる場合がほとんどです。

何も考えずに、ハードウェアをUSBポートに挿入してしまうと、自動でドライバを設定されたりして、ソフトウェアのセットアップで詰まることになります。まずは、製品に同梱されているマニュアルを一読しましょう。多くの場合、ソフトウェアCD-ROMのインストーラを使って、ソフトウェアをセットアップすることになるはずです。

Bluetooth機器のドライバはサービスとしてインストールされるものが多いので、インストール後は必ず再起動します。またインストール時のオプションで、Bluetoothヘッドホンなどの音声関係のサービスや、ファイル転送関係のサービスを選択できる場合があります。もしWiiRemoteだけで利用するのであれば、これらのサービスは使用しないので、追加インストールしたり、サービスを自動起動する設定にする必要はありません（メモリやリソースの節約になります）。

無事に再起動したら、タスクバーやデスクトップにあるBluetoothアダプタのアイコンから、Bluetoothスタックのさまざまなサービスを利用できるようになります。

次項から、実際にいくつかのBluetoothスタックでのペアリングの方法を見ていきましょう。

POINT ▶▶▶

すべての Bluetooth 機器は「ペアリング」という接続認証を行う必要があります。これはさまざまな機器が混在する近距離無線通信において、適切な周辺機器が、適切なユーザーに確認された上で、接続されることを保証するためにあります。携帯電話などの機器は、勝手に接続されると電話帳の閲覧や発信などが行えてしまうため、接続に PIN (パスコード) が必要になります。一方、WiiRemote の場合には、害のない、ただのユーザーインターフェースなので、PIN は設定されていません。基本はホスト側とタイミングを合わせて同期ボタンを押すだけでペアリングできます。

PTM-UBT3S (東芝製スタック) でのペアリング

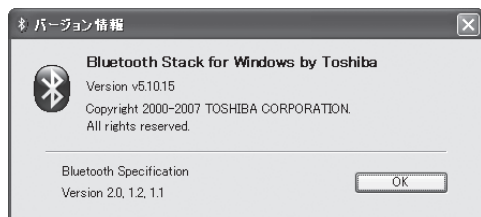
ここでは手軽に入手できる USB 外付けタイプの Bluetooth アダプタ「PTM-UBT3S」(プリンストンテクノロジー)を使って、ステップバイステップで WiiRemote と PC のペアリングを解説します。

筆者が購入した PTM-UBT3S に同梱されていたスタックは東芝製でした。タスクバーの時計の近くにある Bluetooth のアイコンをダブルクリックすると「Bluetooth 設定」というウインドウが起動します。メニューの「ヘルプ→バージョン情報」を選ぶことで「Bluetooth Stack for Windows by Toshiba Version v5.10.15」といったバージョン情報を確認することができます。

図3-1 東芝製 Bluetooth スタックのメイン画面



図3-2 東芝製 Bluetooth スタックのバージョン情報



この東芝製スタックはDELLの一部の機種にも採用されており、WiiRemote登場当初から安定した接続ができていたことで有名でした。さっそく、WiiRemoteと接続してみましょう。

1 ペ어링の開始

「Bluetoothの設定」のメニューから「新しい接続」をクリックすると「新しい接続の追加ウィザード」が起動します。「エクスプレスモード(おすすめ)」のまま、次に進みましょう。

「Bluetooth機器を探しています」と表示されたら、すかさずWiiRemoteの1ボタンと2ボタンを同時に押し、押したままの状態をキープします。すると、プレイヤーインジゲーター(WiiRemote下部にある4個の青色LED)が点滅し、外部からの接続要求を数秒間の間、受け入れることを意味する「接続認証待ち」の状態になります。

図3-3 ここでWiiRemoteを同期モードにする



2 使用するBluetooth機器としてWiiRemoteを選択

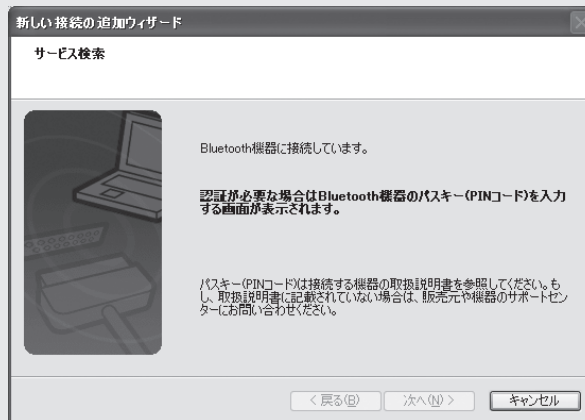
次に「使用するBluetooth機器を選択してください」というダイアログが表示され、そこに図3-4のように「Nintendo RVL-CNT-01」が現れます。これがWiiRemoteです。表示されたら「次へ」をクリックしてください。

図3-4 「Nintendo RVL-CNT-01」つまりWiiRemoteが表示された



「Bluetooth機器に接続しています」というダイアログが表示されたら、WiiRemoteを見て青色LEDが点滅していることを確認します。もし点滅していなかったら、再度1ボタンと2ボタンを押したままにして、接続認証待ちの状態を維持してください。

図3-5 このときWiiRemoteは接続認証待ち(LED点滅)であること



PINコードは必要ないので、自動的に接続完了の状態になるはずです。これで無事接続できました。途中何度か1ボタンと2ボタンを押したままにするところがありましたが、WiiRemoteの裏側にある電池フタ内部にある「Sync」ボタンでも同じ操作ができます(まっ

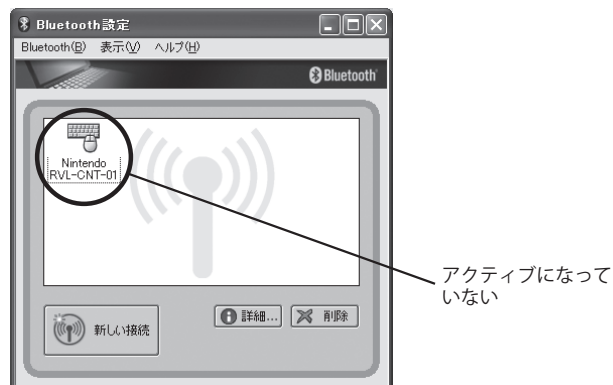
た同じ機能です。

図 3-6 Bluetooth 機器として WiiRemote が接続された



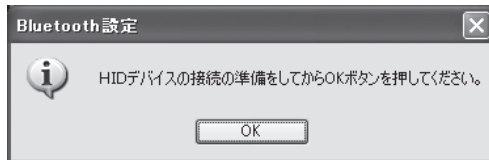
WiiRemote には節電機能があり、しばらくの間、通信やボタン操作がないと、自動的に接続を終了し、電源断の状態になります。この切断状態では、Bluetooth 管理ソフトウェアでの表示は図 3-7 のようになります。

図 3-7 WiiRemote は登録されているが切断されている



この状態は「WiiRemote は登録されているが切断されている」という状態です。ダブルクリックすると、「HID デバイスの接続の準備をしてから OK ボタンを押してください」と表示されます。ここで再度 WiiRemote の 1、2 ボタンを同時に押して、接続認証待ちの状態（4 つの青色 LED が点滅）にしてから、OK ボタンを押せば再接続できます。

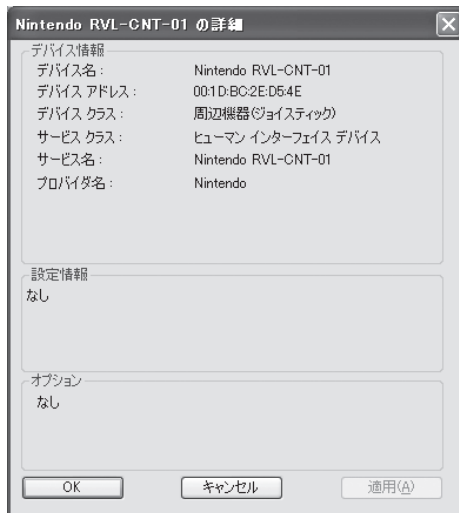
図3-8 WiiRemote の 1、2 ボタンを押して再接続



なお複数のWiiRemoteが混在する可能性がある場合、不要な接続設定は右クリックでメニューを表示し「削除」しておくとういでしょう。うまく接続できないときも、いったん設定を削除して、最初のステップからやり直してみてください。

また、右クリックのメニューから「詳細」を表示できます。ここにはWiiRemoteの個体番号「デバイスアドレス」(ネットワークカードのMACアドレスに相当する固有のID)などが表示されています。複数のWiiRemoteが混在する環境だと、個々のWiiRemoteを見分けるにはこのデバイスアドレスが頼りです。下4桁をシールなどにメモして貼っておくとよいでしょう。

図3-9 登録されたWiiRemote 接続の詳細情報



Lenovo ThinkPad でのペアリング

次に、代表的なノート PC「ThinkPad」に標準搭載の Bluetooth アダプタでの接続例を紹介します。ここでは筆者が使用している Windows XP 搭載 ThinkPad X61 を例にしています。

「コントロールパネル」には「Bluetooth 設定」というアイコンがあり、「診断」タブを見ると「Broadcom Corporation」、「ファームウェアリビジョン Version 2.1.211.299」とありました。

図 3-10 ThinkPad に搭載されている Broadcom 製 Bluetooth スタック



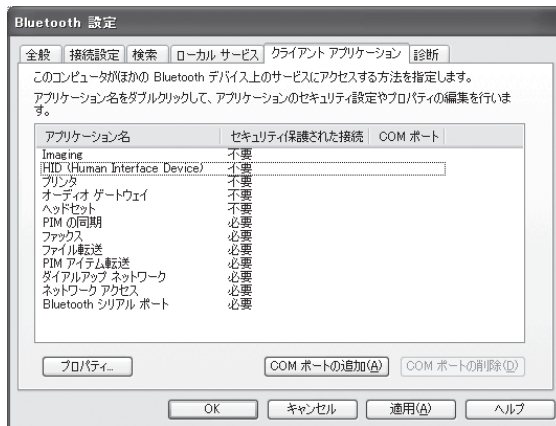
もしこの段階で正しく表示されていない場合、

- Bluetooth アダプタが故障している
- 無線機能がハードウェアスイッチやソフトウェアスイッチで OFF になっている
- BIOS で有効にされていない
- デバイスが存在しない

といった可能性があります。特に BIOS やソフトウェアスイッチでの有効化は見落としがちなので、確認してみるとよいでしょう（無線の状態は「Fn」＋「F5」キーで確認することができます）。

次に「クライアントアプリケーション」のタブをクリックしてみましょう。これから利用する「HID (Human Interface Device)」のセキュリティ保護が「不要」になっていることを確認してください。

図3-11 「クライアントアプリケーション」タブでHIDの設定を確認

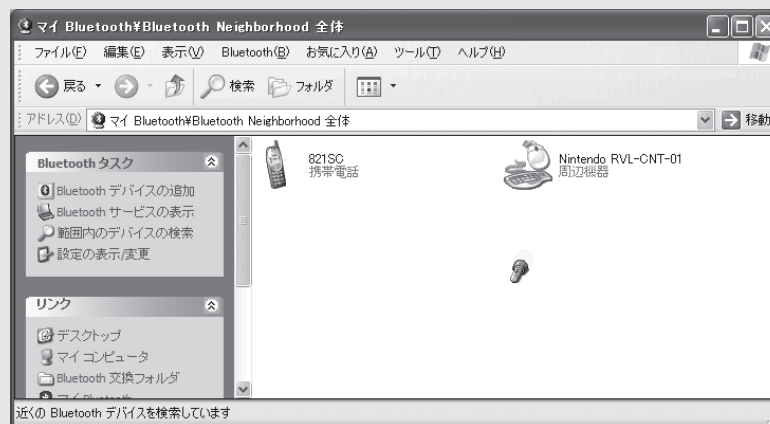


無事に作動しているようなら実際に接続してみましょう。

1 ペ어링可能なデバイスの検索

「マイコンピュータ」に「マイ Bluetooth」というアイコンがあるはずなので、ダブルクリックして開いてください。最初は何も表示されないのですが、WiiRemote の 1 ボタンと 2 ボタンを押して、ペアリング可能な状態にしたところで「範囲内のデバイスの検索」を実行してみてください。

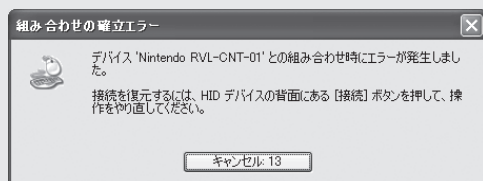
図3-12 範囲内のデバイスの検索



近くに WiiRemote や携帯電話など、ペアリング可能なデバイスが存在するとアイコンが表示されます。このアイコンにマウスポインタを近づけたり、プロパティを見るとデバイスアドレスを見ることができます。

さて、ここでつい、そのアイコンをダブルクリックして「デバイスの接続」を実行してしまいがちなのですが、この方法では永遠に WiiRemote とのペアリングを実現することはできません。

図 3-13 エラーが表示され先に進めない



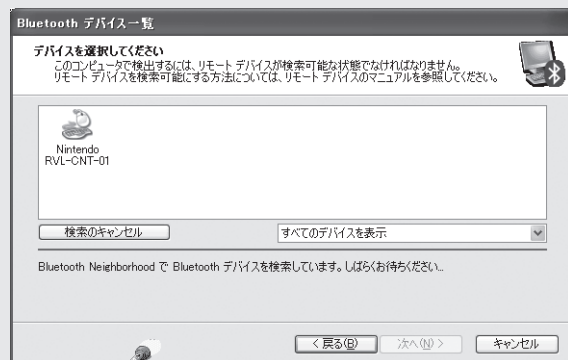
この流れでいくと、ペアリングのための PIN コードを必須として要求されるのですが、WiiRemote には「空白のパスワード」が設定されているため、空白のパスワードを受け付けないこのスタックでは認証ができないのです。これはおそらく Broadcom のスタックにおけるバグか仕様ミスなのですが、ちょっとしたコツで接続する方法があります。

2 「Bluetooth セットアップウィザード」で接続設定を追加

まず一度、検索結果の「個々のデバイスアイコンではない空白部分」をクリックしてください。すると左側のタスクの一覧に「Bluetooth デバイスの追加」というタスクが現れます。これをクリックすると「Bluetooth セットアップウィザード」が起動しますので、「次へ」ボタンをクリックして検索を開始します。WiiRemote が現れたらそのアイコンをクリックして「次へ」のボタンを押します。ここから WiiRemote の 1 ボタンと 2 ボタンを押したままにして指を離さないでください。

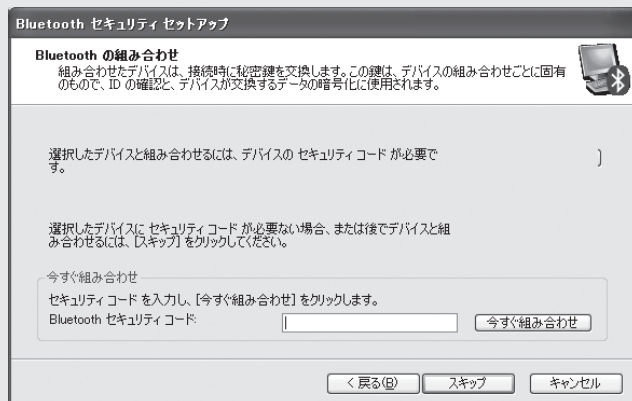
PIN を要求する画面が表示されますが、WiiRemote の 1、2 ボタンを片手で押し続けた

図 3-14 表示された「RVL-CNT-01」をクリックして「次へ」



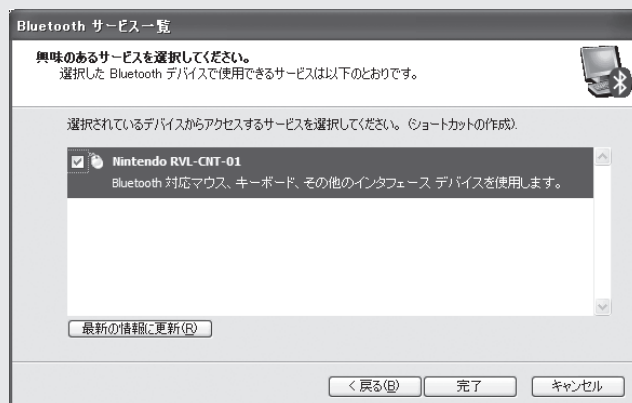
まま「スキップ」をクリックします。そのまま3秒ほど待ってみてください。

図 3-15 1、2 ボタンを押したまま「スキップ」し、3 秒ほど待つ



するとエラー画面は表示されず、無事、WiiRemoteのHIDサービスを選択する画面が表示されます。チェックボックスをONにして「完了」してください。

図 3-16 このサービス選択画面が表示されるまで待つのがコツ



この流れで無事Broadcomのスタックも利用できるようになります。

PINが要求される画面で1、2 ボタンを押したままにしておくのがポイントです。再接続の場合も同じ手順で接続できますが、一度ペアリングに成功していれば、マイコンピュータ内の「マイ Bluetooth」から、接続したいアイコンをクリックして「選択したBluetoothデバイスの追加」として、1、2 ボタンを押したままにしておけば、比較的簡単に再接続できます。

この後「マイ Bluetooth」ウィンドウには何も表示されませんが、「範囲内のデバイスの表示」というBluetoothタスクをクリックしてみてください。「マイ Bluetooth¥Bluetooth Neighborhood

全体」という場所にはリンク状態を示す「→←」が付加された WiiRemote のアイコンが表示されているはずです。

切断する場合は、デバイスのアイコンをクリックして「無効化」を選んでください。再接続は、デバイスのアイコンをクリックして Bluetooth タスクから「選択した Bluetooth デバイスの追加」を選ぶと、ウィザードが再度起動します。

POINT ▶▶▶

上記の手順と同じく、1、2 ボタンを押した状態をキープして「スキップ」を選んだ後、そのままボタンを押し続けて HID サービスが表示されるまで 3 秒ほど待つのを忘れないようにしてください。

以上で、ThinkPad に搭載されている Broadcom 製のエクスペローラ統合型のスタックにおける接続方法の解説は終わりです。長年の実績と堅牢さで人気の ThinkPad ですが、筆者が使用している環境はいささか古いのかもしれません（システムのデバイスマネージャから Bluetooth デバイス内部のドライバのプロパティを見ると、メーカーは確かに Broadcom ですが、日付が「2006/12/19」、バージョンが「5.1.0.2900」です）。

Broadcom 製 Bluetooth スタック、ちょっといい話

Broadcom は Bluetooth 業界では大手で、WiiRemote 自体にもそのチップが搭載されています。Broadcom が供給するドライバソフトウェアは周辺機器として販売されている Bluetooth アダプタや多くのノート PC に採用されています。ドライバのバージョンも日々進化しており、WiiRemote が登場した当初と比べると下層のドライバには何の不満もないのですが、ときおり翻訳上の表現や GUI の動作で意味不明なところもあります（そもそも WiiRemote を接続するテストをスタック開発者が標準的に行っているとは思えませんが……）。

また、今のところ、HID サービスが列挙されるまで「1 ボタンと 2 ボタンを押した状態をキープしないとタイムアウトする」というテクニックは共通のようです。

その他、筆者が体験したおもしろい Broadcom 製 Bluetooth スタックの経験としては、デバイスの列挙時に文字化けを起こし、ボタンも表示されず、「何をしてよいのかわからない」というケースがありました。こんなときは「Alt + C」（「Alt」キーを押しながら「C」）や「Alt + S」、「Alt + N」など「スキップ」や「次へ」にあたるショートカットを試してみるとよいでしょう。

ASUS EeePC S101 でのペアリング

続いて、最近話題のネットブックの代表ともいえる ASUS 製「EeePC 901」でのペアリングを紹介します。内蔵されている Bluetooth アダプタは ThinkPad の例と同じく Broadcom 製ですが、バージョンが異なり、より Windows のエクスペローラに統合されています。接続はできるのですが、ちょっとしたコツがあるので、初期設定から解説します。

まずは、初めて Bluetooth で接続する場合、コントロールパネルの「システム」から「デバイスマネージャ」を起動し、どのような Bluetooth デバイスがインストールされているのか確認しておきましょう。

図3-17 デバイスマネージャ「Bluetooth デバイス」でドライバの詳細を確認



1 Bluetooth の初期設定

この PC で初めて Bluetooth を使用する場合、初期設定が必要になります。コントロールパネルの「Bluetooth 設定」をダブルクリックすると、初期設定のためのウィザードが起動します。

まず、設定する Bluetooth サービスにアクセスするためのアイコンをスタートメニューやプログラムメニュー、マイコンピュータに追加するかどうかを聞かれますが、ここでのチェックはお好みでかまいません。「次へ」ボタンをクリックしましょう。

「WIDCOMM Bluetooth Software 5.1.0.5500」と小さく表示されていますが、この PC の

Bluetooth ホストとしての名前とコンピュータの種類を設定します。ここでは「Eee」という名前の「ラップトップ」としました。

図 3-18 デバイスの名称と種類を設定

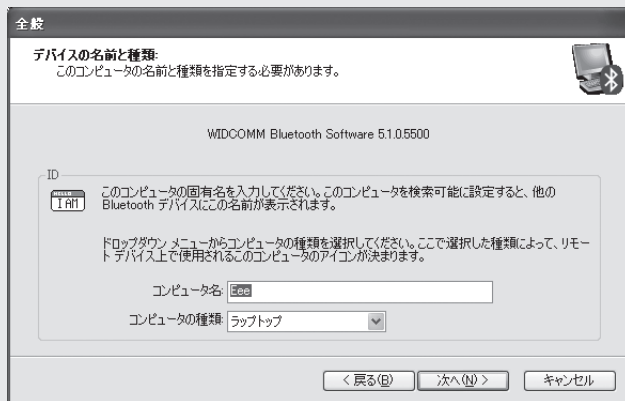
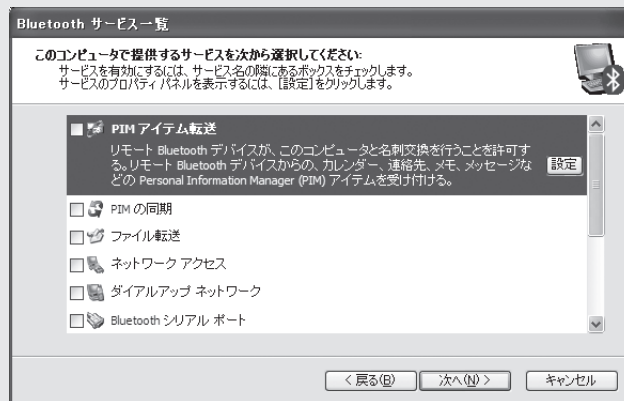


図 3-19 「次へ」をクリックしてサービスの設定に進む



続いてサービスの設定に入ります。

図 3-20 すべてのチェックを外し、「次へ」



ここでは WiiRemote の接続に必要なサービスはありませんので、すべてのチェックを外した状態で「次へ」に進みます。サービスは後でもコントロールパネル「Bluetooth の設定」から「ローカルサービス」で設定することができるので、心配はいりません。

2 | WiiRemote との接続

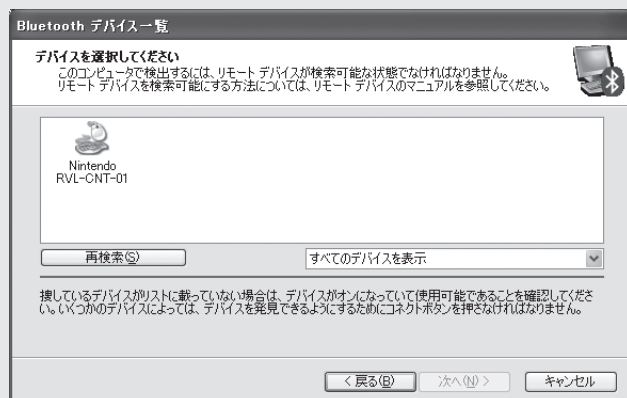
さあ、初期設定は終わっています。ウィザードはそのまま続いてデバイスとの接続を行うことができます。WiiRemote を準備して「次へ」進みます。WiiRemote の 1 ボタン、2 ボタンを同時に押して、接続待機モード（青色 LED4 つが点滅）の状態にします（2 つのボタンからは手を離さないほうがよいでしょう）。タイミングが合わないときはウィザードの「再検索」を押してみてください。

図 3-21 WiiRemote を手元に準備して「次へ」



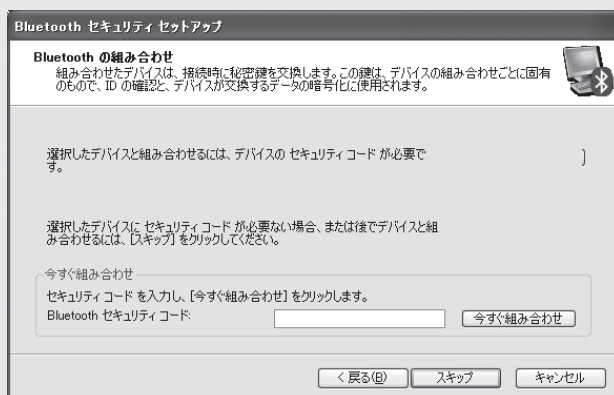
マウスとキーボードのアイコンとともに「Nintendo RVL-CNT-01」が発見されたら、すばやくクリックして「次へ」を押します。このとき、つい WiiRemote のボタンから手を離してしまうことが多いので気をつけてください。

図 3-22 WiiRemote が検出されたらアイコンをクリックして「次へ」



ここから少しすばやく手を動かす必要があります。ダイアログが表示され、ペアリングに入ります。「今すぐ組み合わせ」と表示されていますが、実は PIN を持たない WiiRemote はこの流れでは接続できません。ここでは「スキップ」を選択します。すると、数秒間何も表示されないの、この間 1 ボタン、2 ボタンを押したままにしてください。

図 3-23 セキュリティコードは「スキップ」する



無事に WiiRemote のサービスの列挙に成功すると、図 3-24 のように「Nintendo RVL-CNT-01 : Bluetooth 対応マウス、キーボード、その他のインタフェースデバイスを使用します。」と表示されます。1 ボタン、2 ボタンを押さえたままなので大変かもしれませんが、

チェックして「次へ」を押しましょう。

図 3-24 使用するサービスが表示されるのでチェックして「次へ」

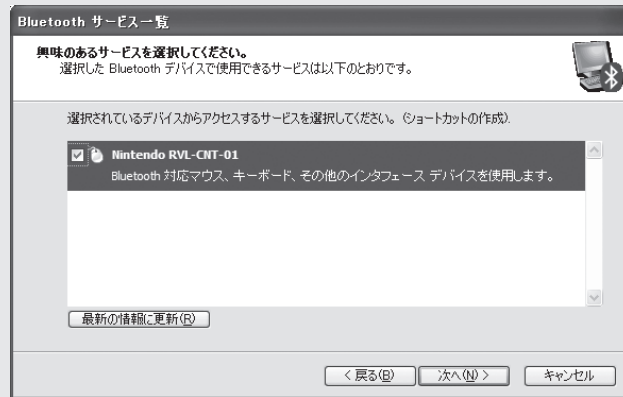


図 3-25 無事に接続が完了



これで無事接続完了です。WiiRemote のボタンから手を離しても大丈夫です。

「マイ Bluetooth¥Bluetooth Neighbourhood 全体」というウィンドウに、WiiRemote を表す「Nintendo RVL-CNT-01」が表示されているはずです。

図3-26 接続中なので「→ ←」が表示されている



ダブルクリックすると、ステータスが表示されます。ステータスには信号強度やアクティビティが表示されます。WiiRemote のボタンを押すと受信データが増える様子を見ることができます。なお、他のスタックのように MAC アドレスを見ることはできないようです。

図3-27 ステータス表示：ボタンを押すと「受信」が増える



右クリックで「無効化」を選ぶと切断状態になります。再度接続するときは「マイコンピュータ」の「マイ Bluetooth」から「Bluetooth デバイスの追加」というウィザード形式のものを選んでください。

図3-28 マイコンピュータの「マイ Bluetooth」では「Bluetooth デバイスの追加」を選択



再接続であっても「デバイスの追加」ウィザードを選ぶ、という点にだけ気をつけてください。接続の流れ自体は再接続でも全く変わりません。1、2 ボタンを押したままにするのを忘れずに！ 接続されている場合は「Bluetooth Neighborhood 全体」に表示されます。

POINT ▶▶▶

「Bluetooth デバイスの検索」や「Bluetooth Neighborhood 全体」を選んでもよさそうなのなのですが、テストした環境ではペアリングの「スキップ」にバグがあるようで、先に進むことができません。

さて、無事に Bluetooth 接続の流れが理解できたでしょうか？ 最後に紹介した ThinkPad と EeePC の例は Broadcom のスタックにおけるバグのようなものがあり、ペアリングに「ボタンを押したままにする」、「再接続時もウィザードから」などのコツが必要でしたが、一度覚えてしまえば難なく利用できます。特に、ネットブックでは Visual Studio を使った開発などはあまり現実的ではありませんが、標準で Bluetooth アダプタを内蔵しているので、展示やプロジェクトなどでは便利に使えることもあるでしょう（何より安価です！）。

ここで紹介した Bluetooth 接続がうまくいかなかった方は、できればこのステップで使えそうなものを調達しておくことをお勧めします。無事に Bluetooth 接続に成功した人は、そのまま次の節の実験に進んでください。

その他のスタックとして BlueSoleil が上げられます（BlueSoleil は、WiiRemote 登場当初は欧米で最も動作実績のあるスタックでした）。専用の接続アプリケーションが特徴的で、特に難なく接続できるので解説しません。

また、Windows Vista において、一時期 WiiRemote が接続できず、諦めていた人もいたかもしれませんが、SP1 以降から動作することが報告されています。諦めずに試してみるとよいでしょう。

ちなみに Mac OS X や Ubuntu などの Linux では OS の標準の機能だけで問題なく接続可能です。

3.2 WiinRemote で実験

世界最速公開のツール「WiinRemote」

WiinRemote は WiiRemote を PC で利用できるフリーウェアです。2006 年 12 月 2 日、Wii 本体が発売されたその日に公開され、世界で最も有名になりました。tokkyo さんによって開発され、「おなかすいた族」にて公開されています。WiinRemote を使うと、WiiRemote で Windows のカーソルを動かしたり、加速度センサーや赤外線センサーの状態を観察することができます。

おなかすいた族

URL <http://onakasuita.org/wii/>

2007 年 1 月 13 日に公開された「WiinRemote_v2007.1.13.zip」が現在のところ最終版で、Borland Delphi 6 によるソースコードも公開されています。最も早く公開されただけでなく、さまざまな機能が安定して利用できるのも、現在でも多くのユーザーに利用されているツールです。

WiinRemote の入手とインストール

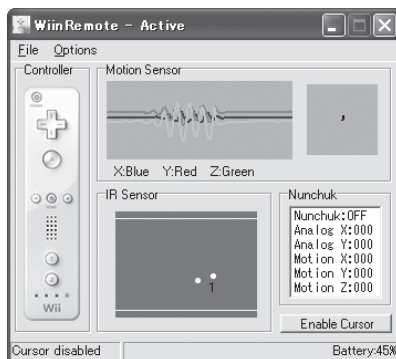
WiinRemote は前述のように、「おなかすいた族」にて無料で配布されています。「ダウンロード」の最も新しいバージョン「WiinRemote_v2007.1.13.zip」をクリックしてダウンロードします。ダウンロードしたファイルを解凍します。場所はどこでもよく、インストールは不要です。あとは「WiinRemote.exe」を実行するだけです。

図3-29 WiinRemoteをダウンロードして解凍



前節で、無事に WiiRemote と PC が Bluetooth 接続できていれば、図 3-30 のように加速度の値を示すグラフが表示されます。WiiRemote をブンブンと振って、動きが変化することを確認しましょう。

図3-30 WiinRemote が起動した状態



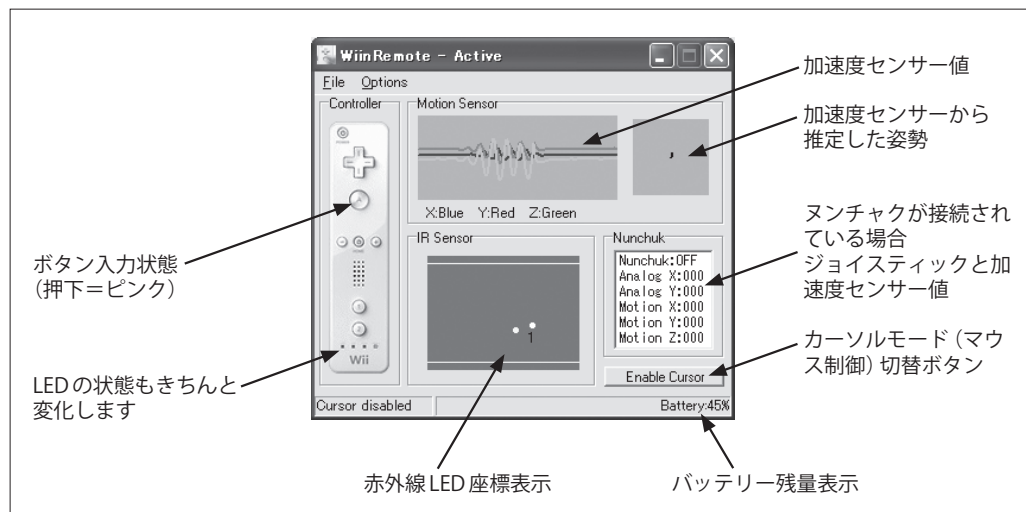
WiinRemote を使いこなそう

WiinRemote にはさまざまな機能があります。WiiRemote の基本的な入力情報を確認するだけでなく、マウスの代わりとしてある程度の操作ができるようになっています。

POINT ▶▶▶

まず、マウス入力機能の On/Off 切り換えのショートカット (「Ctrl + S」) を覚えておきましょう。いろいろ試す際にあわせています。

図3-31 WiinRemote : 各部解説



左側の WiiRemote の写真が、現在押されているボタンを表示しています。ボタンが押されるとピンクで表示されます。

● Motion Sensor

加速度センサーのリアルタイムの値が表示されます。Blue、Red、Green がそれぞれ WiiRemote の X、Y、Z 軸に割り当てられています。次のセクションではこれを使ってちょっとした物理の実験を行います。右側に表示されている黒い線が、推定された WiiRemote の「姿勢」を表しています。

● IR Sensor

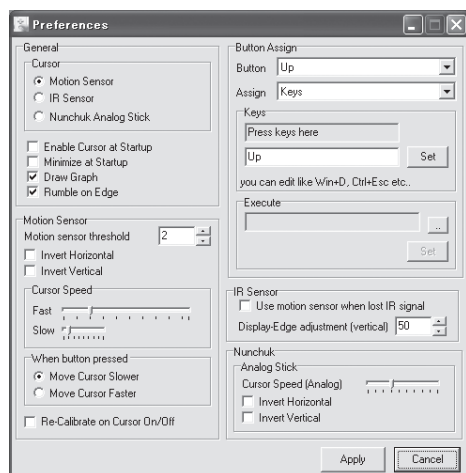
赤外線センサーが取得した座標値を表示しています。いま手元に純正のセンサーバーがなくても、WiiRemote を使って周りを探してみてください。何か反応するものがあるかもしれません。身近なものでは、たとえば、太陽や白熱電球などが赤外線を発しています。

● Nunchuk

ヌンチャクコントローラーが接続されている場合「Nunchuk」のエリアに加速度センサーとアナログスティックの値が表示されます。

「Options」メニュー→「Preferences」、もしくは「Ctrl」+「P」キーで設定画面が表示されます。

図3-32 WiinRemote：設定画面



左から、「General」の「Cursor」で、カーソルモード時のマウスポインタを何で操作するかを選択できます。デフォルトは加速度センサーによる傾きですが、これはあまり操作しやすいものではありません。赤外線で操作する「IR Sensor」もしくはヌンチャクのアナログスティックで操作する「Nunchuk Analog Stick」をぜひ試してみてください。その他のチェックボックスの機能は以下の通りです。

表3-1 WiinRemoteのGeneral設定

項目	意味
Enable Cursor at Startup	カーソルモードを起動時から使用
Minimize at Startup	起動時に最小化
Draw Graph	グラフを描画
Rumble on Edge	マウスカーソルが画面端に来たときにバイブレーターを振動

「Motion Sensor」項目は、加速度センサーでマウスを操作するときに必要なパラメータです。

表3-2 WiinRemoteのMotion Sensor設定

項目	意味
Motion sensor threshold	加速度センサーの微小な値をどこまで無視するか
Invert Horizontal	水平方向を逆転
Invert Vertical	垂直方向を反転
Cursor Speed	カーソル移動速度の最速と最小（左にすると遅い）
When button pressed	Bボタンが押されているときにカーソルが速い/遅いを選択
Re-Calibrate on Cursor On/Off	カーソルモードの切り替え時にキャリブレーションを実行

「Button Assign」では、WiiRemoteの各ボタンに機能を割り当てられます。シンプルですがなかなか強力な機能で、たとえばButtonを「HOME」、Assignを「Keys」にして、「Keys」にある「Press Keys Here」をクリックして「Alt」キーを押してから「F4」キーを押してみましょう。その下に「Alt + F4」と表示されたら「Set」を押します（最後に「Apply」キーを押して設定保存）。これで「HOME」ボタンを押したら、そのウィンドウを閉じる機能ができました。同じような手順で「+」「-」キーに音量を割り当てたり、「1」「2」キーに好きなアプリケーションを割り当てることができます（「Execute」で実行ファイルを選んでから「Set」するのを忘れずに）。

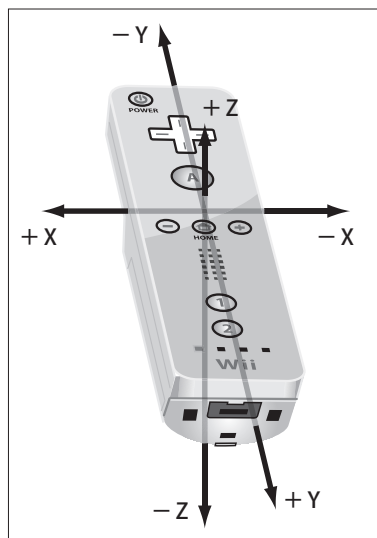
「IR Sensor」では「赤外線が見えないときは加速度センサーを使う」というチェックボックスと、LEDが視界の外に出た場合の安定性を調整するための「見失った時の調整枠を表示（垂直方向）」があります。

「Nunchuk」ではアナログスティックをカーソルモードで使用した場合の速度と、上下左右方向の逆転用チェックボックスがあります。

重力を観察してみよう

さて、さきほど紹介した「Motion Sensor」では、加速度のリアルタイム値を3色{X,Y,Z} = {青, 赤, 緑} で表示しています。WiiRemoteのX、Y、Z軸とはそれぞれ、ボタンの付いている面を

図3-33 WiiRemoteの3軸加速度センサー



机の上に置いたとして、側面のボタンが付いていない方向がX、赤外線センサーが付いている方向がY、ボタン面の上下方向がZとなります。

そして、よく見ると、3つの色の線はいつも同じレベルにはありません。ボタン面を上にしているときは緑が他に比べて少し低く(①)、右側を下にする形で横に寝かせると青が低く(②)、そして赤外線センサー部分を下にして立たせると赤が低く(③)になります。

図3-34 WiinRemoteで重力を観察



これはいったい何でしょう？ そうです！これが地球上のすべての物質に働く「万有引力」です。普段は直接私たちの目には見えない重力加速度が、各軸の下向きに働く加速度として観察できているのです。当たり前のことなのですが、ちょっとした感動が味わえませんか？ これを利用することで、WiiRemoteの傾きも検出できます。加速度センサーグラフの右側ではそのようにして姿勢を推定しているようです。ただし、気をつけなければなりません。人間がWiiRemoteを強く振ると、重力よりも大きな値を入力することになります。そんなとき、この姿勢推定はどんな様子になるのでしょうか？ 実際に観察してみてください。

PowerPointでプレゼンテーション

さて、シンプルでパワフルな「WiinRemote」ですが、実用的に使うために、少し設定してみましょう。ここではプレゼンテーションソフト「PowerPoint」での利用を想定してみます。

まずは「HOME」ボタンでPowerPointを起動できるようにしましょう。「Ctrl」+「P」キーで設定画面を表示させて「Button Assign」の「Button」を選んで「Home」に変更、そしてすぐ下の「Assign」を「Execute」に変更します。さらに下のExecuteに「C:\Program Files\Microsoft Office\OFFICE11\POWERPNT.EXE」をファイルブラウザなどを使って設定し、最後にSetを押します。「Apply」を押していったん設定画面を閉じて、HOMEボタンを押すとPowerPointが起動するのを確認しましょう。表示されたら、好きなプレゼンテーションファイルを開いておくとういでしょう。

よく聞く“キャリブレーション”って何？

「キャリブレーション」とは、デバイス系制御ではよく出てくる用語です。センサーや測定器など、ある信号の入力と測定の対象と出力される値との関係を、再現性のある基準に合わせて決定付ける作業です。デバイス制御系のプログラミングでは、デバイスの「生(なま)の値」(raw data)を使用できるデータに変換するための操作ともいえます。

この変換は用途や特性によりさまざまな変換が実施されます。初期値や原点をセットすればいいものから、較正曲線といった二次曲線を利用する方法、ある値域だけを利用するバンドパスフィルタを組み合わせた方法など、その調整方法についてはさまざまです。

なお昔は「較正」とも表記されていましたが、近年では計量法で「校正」と表記されています。しかし、正式には「校正」の概念には「調整」が含まれないそうです。そのため、最近では「キャリブレーション」のままカタカナで表記されることが多いようです。

ユーザーインターフェースなどで「キャリブレーションする」といった場合は、たいてい「原点のリセットなどを行ってそこを基準にする」という行動であることが多いです。たとえば、タブレットPCなどのタッチパネルを初めて使うとき画面に表示された「+」をクリックしてください、というメッセージが数回出ることがありますが、内部では4点の表示上の位置とタッチパネルデバイスから送られてくる信号を適合させるための「キャリブレーション」を行っています。

WiiRemoteは計測機器ではなく、ゲーム用のインターフェースですから、測量や重力の再現性は、そこそこ求められますが、測定器ほどではありません。再現性の不備や、経年変化などで誤りがあったとしても、同じような体験が再現できればいいわけです。

そのためWiiRemoteには、加速度センサーの重力に対する補正値が保存されているようです。また赤外線センサーの強度は距離、すなわち使用する部屋の空間に依存しますので、Wii本体側で4段階に設定することができるようです。その他、必要になるキャリブレーションの仕組みはアプリケーション開発者側で考えて実装する必要があります。もちろんキャリブレーションの必要のない、極力少ない手順で安定して利用できる仕組みが実現できれば、すばらしいです。

同じ要領で、以下のように割り当ててみてください。

表3-3 PowerPoint用の Button Assign 例

Button	Assign	機能
Up	Up	スライド戻し (変更なし)
Down	Down	スライド送り (変更なし)
Right	Keys [PgDn]	ページ送り
Left	Keys [PgUp]	ページ巻き戻し
A	Mouse->Left Click	マウスクリック (変更なし)
B	Cursor On/Off & Speed	カーソルモード & 速度切替 (変更なし)
Plus	Mixer->Volume Up	再生音量アップ
Minux	Mixer->Volume Down	ビデオ等の再生音量ダウン
1	Keys [Esc]	いざというときのための終了
2	Keys [F5]	プレゼンテーション再生

どうでしょう？ いつものPowerPointが、よりカッコよく操作できるようになりました。この応用で、特にプログラムを書かなくてもさまざまなアプリケーションでWiiRemoteが使えるようにカスタマイズすることができます。

ここではフリーウェア「WiinRemote」を使って、WiiRemoteの接続後の動作確認と、PowerPointをWiiRemoteで操作するための設定を解説しました。この応用で、特にプログラムを書かなくても、さまざまなアプリケーションをWiiRemoteで操作できるように、GUIだけでボタンの割り当てなどを設定することができます。

WiinRemoteはシンプルですが、スタートが早かったことで多くの人々に支えられ、多くのスタックで安定動作することが特徴です。インストールも不要でサイズも小さいので、いざというときの動作確認のために持ち歩いておくに便利です。

しかし、ボタンの割り当てなど、その固有の設定を保存して後で読み込んだり、切り換えたりすることには向いていません。またジョイスティックの代わりの信号を出すことも、残念ながらできません。そのような用途には次節で解説する「GlovePIE」が適しているでしょう。

3.3

GlovePIEで実験

GlovePIEとは

GlovePIEは最も広く使われている、WiiRemoteをサポートするコントローラーエミュレーターです。Carl Kennerによって開発されています。GlovePIEとは「Glove Programmable Input Emulator」の意味で、もともとヴァーチャルリアリティのためのデータグローブ製品「5DT」を使って、さまざまなゲームをプレイするためにジョイスティックやマウスをエミュレーションするために開発されていたようです。その成長の過程でWiiRemoteをサポートし、有名になりました。特徴として、すべてのエミュレーションを専用のスクリプトで記述します。たとえば「Aボタンをショット、Bボタンをボムに」といったゲームそれぞれの割り当てを、自分で書くことができるのです。

たとえば、「DOOM」などの一人称シューティングゲームでよく使われる [W] [A] [S] [D] キーがそれぞれ前後左右の移動キーに割り当てられている場合、GlovePIEスクリプトでは、データグローブの位置を使って

```
W = glove.z > -50 cm
S = glove.z < -70 cm
A = glove.x < -10 cm
D = glove.x > 10 cm
```

と表現することができます。他にもジョイスティックやマウス、キーボードも複数のキー入力を連続したマクロとして扱うことなどもでき、SAPI（音声出力API）や、電子音楽で使われるMIDIやOpenSound Control（OSC）などの出力もサポートしており、非常に高機能なツールといえます。

GlovePIE のインストール

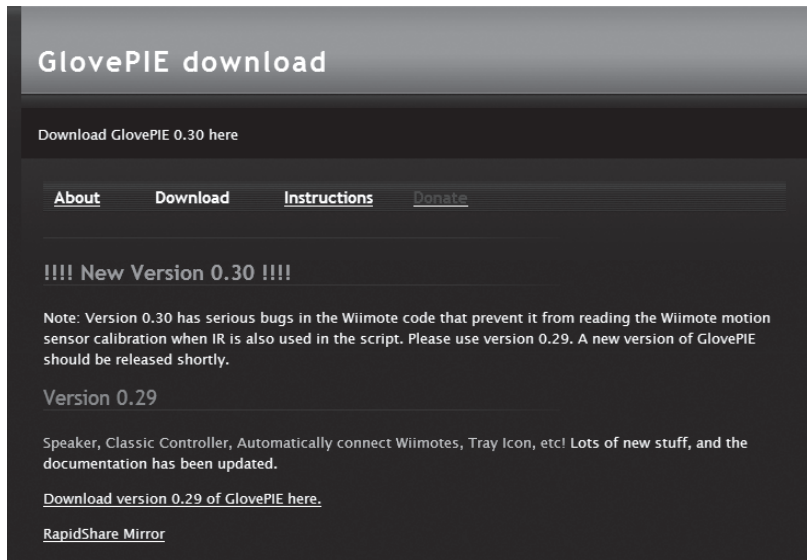
GlovePIE のインストールは非常に簡単です。公式ホームページに行き、最新版をダウンロードし、アーカイブを展開するだけです。

GlovePIE download

URL http://carl.kenner.googlepages.com/glovepie_download

最新版はバージョン 0.30 なのですが、深刻なバグ（赤外線センサー使用時に加速度センサーキャリブレーションに不具合）を理由に公開が中止されています。ジェスチャー認識などたくさんの新機能が盛り込まれているようなのですが、「公開されるまでの間、バージョン 0.29 をお使いください」と書かれたまま、ずいぶん長い時間が経ってしまっています。バージョン 0.29 の公開は 2007 年 1 月 4 日と少々古いのですが、特に問題はないので、こちらをダウンロードしましょう。

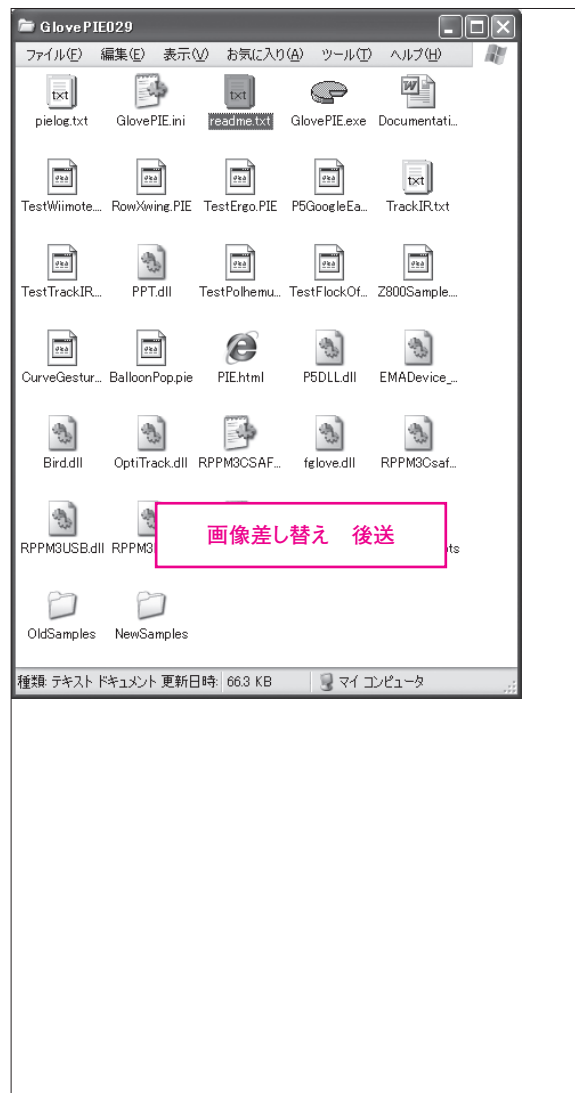
図 3-35 GlovePIE のホームページ：ダウンロードに関して



ダウンロードのリンクも帯域の制限などで 4 か所ありますので、確実にダウンロードできているかどうか確認しながら「GlovePIE029.zip」をダウンロードしてください。

ダウンロードした ZIP ファイルを展開すると、たくさんのファイルが現れます。実行ファイルは「GlovePIE.exe」ですが、まずはライセンス関係を確認するために「readme.txt」を開きましょう。

図 3-36 GlovePIE ver.0.29を展開したところ



ライセンスに関しては、以下のように記述されています。

This software is copyright © Carl Kenner, except for scripts by other
By using this software you agree to obey the following license conditions:

* You can't make money using this software as part of a baseball simulation. This is for contractual reasons. But you can make as much money as you like using it for anything else.

* You may not use this software directly or indirectly for any military purpose. This includes, but is not limited to, training, research and development, controlling military hardware, directing military personel, or troop entertainment. You may not use this software anywhere on a military base or vessel. This applies to all versions of PIE.

* You may not export this software to Israel, or use it in Israel (including the occupied territories), until Israel has ended its occupation of the West Bank, Gaza Strip, Lebanon, Syria, and anywhere else it may occupy. If you try to run it in Israel it will give you an error.

* Missionaries may not use this software. It may not be used for any missionary purpose. Or any other genocidal purpose.

* You may not use this software to cheat at online or multiplayer games. What constitutes cheating depends on the game and the server. Just using a different input device shouldn't be considered cheating, but complex scripted actions to make things easier may be considered cheating. Don't get GlovePIE banned, or you will hurt everyone who wants to play with a VR glove. But feel free to cheat at single player!

意識すると、以下のようになります。

このソフトウェアのコピーライトは©Carl Kennerです。他の著者によるスクリプトを除きます。このソフトウェアを使用することにより、以下のライセンス条項に従うことに同意します。

- あなたは、このソフトウェアを野球シミュレーションの一部のようにして金を儲けることができませんが、これは契約上の理由です。他の何かのために使うことで、好きなだけお金を

儲けることができます。

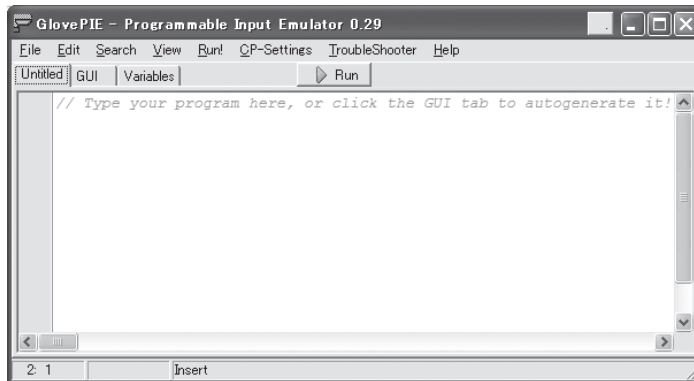
- 直接または間接的に、このソフトウェアを軍用目的に使用してはいけません。これはトレーニング、研究開発、軍用ハードウェアの制御、部隊指示、部隊の娯楽などを含み、限定されません。このソフトウェアを軍用基地もしくは軍用艦内で使用してはいけません。これはPIEのすべてのバージョンに該当します。
- このソフトウェアを、イスラエルがウエストバンク（ヨルダン川西岸）のその占領、ガザ地区、レバノン、シリアと他の占領区の占有を終えるまで、イスラエルに輸出もしくはイスラエル（占有された領土を含む）で使用してはいけません。もしあなたがこのソフトウェアをイスラエルで実行しようとするば、あなたにエラーを与えます。
- 宣教師は、このソフトウェアを使用してはいけません。いかなる伝道目的、もしくは他のいかなる大量虐殺目的にも使われてはいけません。
- オンラインまたはマルチプレイヤーゲームでいかさまをするために、このソフトウェアを使用はしてはいけません。不正行為を意味するところは、ゲームとサーバーに依存します。普段と異なる入力デバイスを使うことは不正行為とされるべきではありませんが、ものごとをより簡単にするための、複雑なスクリプト化されたアクションは不正行為になるでしょう。GlovePIEを禁止させられないようにしてください。もしくは、VRグローブでみんなを傷つける遊びをしたいなら、シングルプレイヤーで不正行為を遠慮なくやってください！

本書を読んでいらっしゃる方で、このライセンス条文が問題になる人はまずいないでしょう。もしこのソフトを使ってお金持ちになった、なんていい話があったら、ホームページの「Donate」から、作者のCarl Kenner氏に募金をするとういかもしれません。新しい機能についてのアイデアなども、募金とともに募集しています。

GlovePIEの基本操作

まず、3.1節の流れに沿って、WiiRemoteをBluetooth接続してください。無事接続が終わったら「GlovePIE.exe」を起動します。

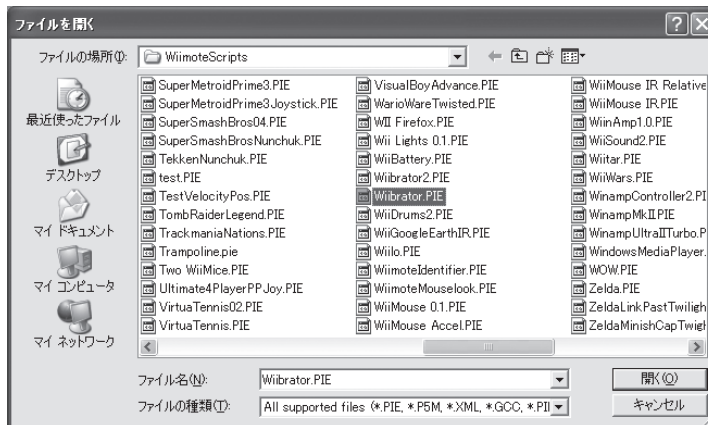
図3-37 GlovePIEを起動したところ



この後、サンプルスクリプトを読み込んで、実際に WiiRemote を使って操作する実験を行います。ここで注意です。スクリプトが想定外の動作をすることで、正常な操作ができなくなる可能性があります。そうなったときにあわてないために、「Shift + P + I + E」もしくは「Alt + R」→「S」で「スクリプトの停止」ができることを覚えておくといよいでしょう（「Shift + PIE」は動作しないこともあります）。

まずは、最も簡単な WiiRemote を使うサンプルスクリプトを試してみましょう。「File」メニューから「Open」を選んで、「WiimoteScripts」※2 にある「Wiibrator.PIE」をロードします。

図3-38 Wiibrator.PIE をロード



※2：このディレクトリには大量のおもしろそうなファイル名が並んでいます。PIE スクリプトというテキストファイルで、さまざまなゲームのエミュレータや、Windows 上で WiiRemote を使うためのサンプルが用意されています。メモ帳などのテキストエディタで開いたり編集することもできますが、この際なので「.PIE」という拡張子を GlovePIE に関連づけてしまうのもよいでしょう。

スクリプトがロードされると、以下のように表示されているはずです。

コード3-1 Wiibrator.PIEをロードしたところ

```
//Wiibrator
//By deceased

// Does not control mouse so feel free to surf ^^

// D pad up turns it on
// D pad down turns it off
if wiimote.Up
    wiimote.Rumble = 1
endif
if wiimote.Down
    wiimote.Rumble = 0
endif
```

このスクリプトは十字キーを使ってバイブレーターをOn/Offするだけのスクリプトなので、安心して実行できます。「F9」もしくは、メニューの[Run!]から[Run!]を選んで実行してみましょう。

緑の三角矢印が「Run」から「Stop」に変わり、スクリプト表示エリアがグレーになったら実行中です。接続したWiiRemoteの十字キーの上(POWERボタン側)を押してみましょう。バイブレーターが作動します。あわてず十字キーの下(Aボタン側)を押すと、止まります。何度でも繰り返し遊んでみてください。ひととおり遊んだら、GlovePIEウィンドウ内の「Stop」を押してスクリプトを止めましょう。

さて、このスクリプトでは何が行われているのでしょうか？「//」で始まる行はコメントなので、実質6行のスクリプトです。まず、PIEスクリプトは非常に動的でユーザーフレンドリーなスクリプトで、ほとんどのケースで初期化コードが不要です。一番上から実行され、一番下までいくと、また上から実行されます。この場合も宣言や初期化などしなくても

```
if wiimote.Up
```

とすることで、WiiRemoteが初期化され、現在のボタンの状態をwiimote.Upで取得することができます。ここがTrue(=真)になったときにwiimote.Rumble = 1となりバイブレーターを駆動し、endifでif文を抜けます。下を押した場合、つまりif wiimote.Downの場合も同様に、バイブレーターの振動を意味するwiimote.Rumbleを0にすることで振動を止めます。

たとえば、ここで新しくシンプルなスクリプトを作成してみましょう。「File」メニューから

「New」を選びます。「Clear text box without saving changes? (保存しなくてもいいですか?)」という質問が出たら「Yes」で問題ありません。新しいスクリプトとして、以下をテキストボックスに書きます。

コード3-2 **GlovePIE** Aボタンを押している間だけバイブレーターを鳴らす

```
wiimote.Rumble = wiimote.A
```

たった一行のプログラムですが、Aボタンを押している間だけバイブレーターが鳴ります。GlovePIEはスクリプト記述についても支援機能があり、「wiimote.」といったように予約されたクラス名とピリオドまで書くと、自動的にその先のプロパティ名などを表示してくれます。これでマニュアルがなくても簡単にスクリプトが書けますし、またシンタックスエラー（文字の書き損じによる間違い）も大きく減らすことができます。

書き終わったら、まずはシンタックスを確認しましょう。メニューの「Run!」から「Check for errors」を選びます。何か間違いがあると、該当する行がピンク色に変わります。間違いがなければ何も起きませんので、「F9」で実行しましょう。無事に、Aボタンを押している間だけバイブレーターが鳴れば成功です。Aボタンを連打などとしてみて、バイブレーターの鳴り具合を確認してみるとよいでしょう。

POINT ▶▶▶

もし挙動がおかしい、反応がないといったときは、接続を確認するために、WiinRemoteなど確実に動くツールを起動してみるとよいでしょう。Bluetoothが時間切れで自動切断されている場合もあります。単純なスクリプトなので、挙動がおかしいときは、単にGlovePIEやPCを再起動してみるのも解決策になるときもあります。

バイブレーターを“暴走”させてみよう

バイブレーターに関する注意を、実験を通して確認してみましょう。バイブレーターを起動した状態でスクリプトを止めると何が起きるでしょう？ 上記のスクリプトを実行し、Aボタンを押した状態で、GlovePIE（スクリプト）を停止させます。するとスクリプトは終了しているのに、バイブレーターは鳴り続けます。大変です。これは暴走状態です。この状態でしばらく放っておくと、あっという間に電池が切れてしまいます。しかし、スクリプトを再度起動しても、Aボタンを連打しても、バイブレーターは止まらないかもしれません（これはバグではないかと見えています）。

こういうときは、あわてず騒がず、GlovePIEを終了します。スクリプトの保存が必要なら保存しておきます。そして、再度GlovePIEを起動し「Wiibrator.PIE」などのバイブレーターを使用するスクリプトを読み込んで、Runすればバイブレーターの暴走は止まります。

3.4**GlovePIEで作る
「高機能赤外線マウス」**

この節では、GlovePIEを使って高機能な赤外線マウスを作成していきます。実践的な開発を通して、WiiRemote開発のコツやGlovePIEの強力なスクリプティング機能とGUIによる支援機能をステップバイステップで学ぶことができます。

高機能赤外線マウスを設計する

まずは、これから作成する赤外線マウスについて、簡単に仕様を決めておきましょう。基本動作としては、WiiRemoteをセンサーバーや赤外線光源に向けて、マウスのようにして使うタイプのものにします。せっかく作るのですからファイル操作やPowerPointのプレゼンテーションに実際に使える高機能なものを目指します。

表 3-4 想定する赤外線マウスの機能

WiiRemote 側操作	割り当てる操作
赤外線	マウスポインタの移動
十字キー	カーソルキー
A ボタン	マウス左ボタン
A ボタン・ダブルクリック	Enter キー
B ボタン	マウス右ボタン
B ボタン・ダブルクリック	Delete キー
A + B ボタン同時押し	デスクトップを表示
+ ボタン	アプリケーション切り替え「Alt + Tab」
- ボタン	アプリケーション終了「Alt + F4」
Home ボタン	スクリプト終了
1 ボタン	GlovePIE 最小化／最大化
2 ボタン長押し	PowerPoint を起動
2 ボタン・シングルクリック	Esc キー（プレゼンテーション終了）
2 ボタン・ダブルクリック	プレゼンテーション開始「F5」

WiiRemote ならでは、たくさんあるデジタルボタンを活用して、長押し・ダブルクリック・A + B ボタン同時押しなどのコンビネーションを使った使いやすい操作を盛り込んでいます。もちろんこのセクションの体験を通して、ご自身のアイデアで新しい機能を盛り込んでいくことも可能です。

ボタンアクションの実装

ボタン、キーに動作を割り当てる

まずは手始めに「Home ボタンで終了」を実装してみましょう。新しいスクリプトとして以下を記述して実行してみてください。GlovePIE スクリプトでは大文字・小文字は無視されます。

コード 3-3 **GlovePIE** 「Home ボタンで終了」の実装

```
if Wiimote.Home then
  ExitScript
end if
```

記述したら WiiRemote を接続後に Run して、Home ボタンで終了できるかどうか試してみてください。「ExitScript」とは実行中の GlovePIE スクリプトを終了させるコマンドです。詳しくは、GlovePIE と同じディレクトリにあるマニュアル (Documentation.rtf) に記載されています。基本的なコマンドはこのセクションで解説していきますので、今は「そういう便利なものがあるんだ」という理解でよいでしょう (本節の最後にまとめて紹介します)。

さて、次は十字キーにカーソルキーを割り当てましょう。先ほどのように if 文を使うと明確に条件やその後の処理を記述できますが、今回はボタンイベントに対するキーアクションを大量に設定しなければならないので、もっと簡単な記述方式で書き直してみます。

コード 3-4 **GlovePIE** 十字キーにカーソルキーを割り当てる

```
//WiiRemote IR mouse

//Key Binds
ExitScript = Wiimote.Home
Key.Up      = Wiimote.Up
Key.Down    = Wiimote.Down
Key.Left    = Wiimote.Left
Key.Right   = Wiimote.Right
```

このように「=」でつなぐことで、キー入力に割り当てるアクションが 1 行で表現できます。また「//」で始まる行はコメント行です (日本語は文字化けします)。

次は + ボタンでアプリケーションを切り換え、- ボタンでアプリケーション終了のようにします。「Alt + Tab」のように、複数のキーを使う場合は以下のような表記をします。

コード 3-5 **GlovePIE** + ボタンでアプリケーションの切り換え、- ボタンでアプリケーション終了

```
Key.Alt+Tab = Wiimote.Plus
Key.Alt+F4  = Wiimote.Minus
```

簡単ですね! (もちろん「Key.Alt + Key.Tab」と表記してもかまいません)

そして、次は「WiiRemote の A ボタンと B ボタンを同時に押したときに、デスクトップを表示する」というアクションです。

コード 3-6 **GlovePIE** 「WiiRemote の A ボタンと B ボタンを同時に押したときに、デスクトップを表示する」実装

```
Key.Windows+D = Wiimote.A and Wiimote.B
```

次は、WiiRemote の A + B ボタンのダブルクリックを使って Enter や Delete を入力できるようにします。DoubleClick() という関数を使うことで、ダブルクリックを判定できます。

コード 3-7 **GlovePIE** A + B ボタンのダブルクリックを使って Enter や Delete を入力する

```
Key.Enter = DoubleClicked(Wiimote.A)
Key.Delete = DoubleClicked(Wiimote.B)
```

ここで実際に動作を試すために、メモ帳などを実行してからスクリプトを Run してみて、+ ボタンでアプリケーションを切り換え、十字ボタンやダブルクリックを試してみるとよいでしょう。最後に Home ボタンでスクリプトを停止します。

1つのボタンに複数の動作を割り当てる

次は、「2 ボタン」にいろいろな機能を割り当ててみます。長押し、シングルクリック、ダブルクリックで、それぞれ PowerPoint の起動、「Esc」キー、「F5」キーに割り当てます。

コード 3-8 **GlovePIE** 2 ボタンで PowerPoint を起動

```
Execute("C:\Program Files\Microsoft Office\OFFICE11\POWERPNT.EXE")
= HeldDown(Wiimote.Two, 1s)
Key.Escape = SingleClicked(Wiimote.Two)
Key.F5 = DoubleClicked(Wiimote.Two)
```

HeldDown(対象, 秒数) とすることで、長押しを検出できます。単位は秒です。「0.5」などでもよいでしょう。Execute("実行ファイル名") で任意のアプリケーションを起動できます。シングルクリック、ダブルクリックも同様に関数を使って検出することができるので、今までと同じように割り当てます。

さて、実際に 2 ボタンを長押しし、PowerPoint が立ち上がったら、今度はダブルクリックでプレゼンテーションを起動、さらにシングルクリックでプレゼンテーション終了、一ボタンで PowerPoint 終了、と試してみてください。「Esc」キーは他のアプリケーションなどでもよく使うので、ここに割り当てておくのは便利そうです。

一ボタンを押して終了すると「保存しますか?」と聞かれますが、十字キーの右、そして「A ボタン」のダブルクリックで「いいえ」を選択することができます。もう簡単な操作ならなんでも WiiRemote でできそうですね!

最後に、左右のマウスボタンをそれぞれ A、B ボタンに、そして 1 ボタンには GlovePIE の表示切り替え機能を割り当てます。

コード3-9 **GlovePIE** 左右のマウスボタン、1ボタンへの割り当て

```
UnMinimizePie = HeldDown(wiimote.One, 1s)
MinimizePie   = wiimote.One

Mouse.LeftButton = Wiimote.A
Mouse.RightButton = Wiimote.B
```

ここまでの作業でスクリプトは以下のようにになっているはずです。

コード3-10 **GlovePIE** WiiRemote ボタンアクション

```
//WiiRemote IR mouse (part for buttons)

//Key Binds
ExitScript = Wiimote.Home
Key.Up     = Wiimote.Up
Key.Down   = Wiimote.Down
Key.Left   = Wiimote.Left
Key.Right  = Wiimote.Right

//combination
Key.Alt+Tab = Wiimote.Plus
Key.Alt+F4  = Wiimote.Minus
Key.Windows+D = Wiimote.A and Wiimote.B

//Double Clicks
Key.Enter = DoubleClicked(Wiimote.A)
Key.Delete = DoubleClicked(Wiimote.B)

//Multifunctions on Two-Button
Execute("C:\Program Files\Microsoft Office\OFFICE11\POWERPNT.EXE") = ¥
    HeldDown(Wiimote.Two, 1s)
Key.Escape = SingleClicked(Wiimote.Two)
Key.F5     = DoubleClicked(Wiimote.Two)

//Hide and Show by One-Button
UnMinimizePie = HeldDown(wiimote.One, 1s)
MinimizePie   = wiimote.One

//Mouse Buttons are linking to A and B
Mouse.LeftButton = Wiimote.A
Mouse.RightButton = Wiimote.B
```


いかがでしょうか？ GlovePIE スクリプトの短い記述だけで、かなり高機能なツールが作れることが実感できたでしょうか？

PowerPoint がインストールされていない場合は、他のアプリケーションで試してみてください。たとえば、`Execute("mspaint.exe")` とすることで「ペイント」が起動します。「PrintScreen」キーと貼り付け（「Ctrl+V」）でもおもしろいでしょう。

コード 3-11 **GlovePIE** ペイントで試す場合のスクリプト

```
Execute("mspaint") = HeldDown(Wiimote.Two, 1s)
Key.PrintScreen = SingleClicked(Wiimote.Two)
Key.Ctrl+V      = DoubleClicked(Wiimote.Two)
```

デバッグ機能・関数・フィルタの活用

さて、ボタンアクションを一通り使いこなせるようになって、だんだん楽しくなってきたところでしょう！ しかし、まだこの状態では、マウスポインタは相変わらず動きません。センサーバーの赤外線に WiiRemote を向けて、自在にマウスポインタが操作できたらどんなに楽しいでしょうか。

デバッグ機能

はやる気持ちを抑えて、まずは「デバッグ機能」を学びましょう。

先ほどのスクリプト（コード 3-10）の一番下に以下の記述を足してみましょう。

コード 3-12 **GlovePIE** デバッグ用の表示 (1)

```
Debug = mouse.x
```

実行すると、GlovePIE の「Run ボタン」の右側に、何やら数字が表示されるようになったはずです。これは現在のマウスの X（横方向）の値です。ここでマウスを動かしてみましょう。値は左から右にいくにつれ、ゼロから 1 への小数をとるはず（これを値域 [0,1] と表現します）。`Debug=` とすることで、デバッグ用に内部の値を表示する機能です。

続いて、以下の `DebugPrint()` 関数も試してみましょう。

コード3-13 **GlovePIE** デバッグ用の表示 (2)

```
Debug = mouse.x
DebugPrint("X="+mouse.x+" Y="+mouse.y)
```

これで別のウィンドウが開き、時系列でより多くのデータを読むことができます。用途に合わせて使い分けるとよいでしょう。

さて、それではついにセンサーバーを使って、このデバッグウィンドウに赤外線センサーの値を表示してみましょう。先ほどのテストコード(コード3-13)を以下のように書き換えます。

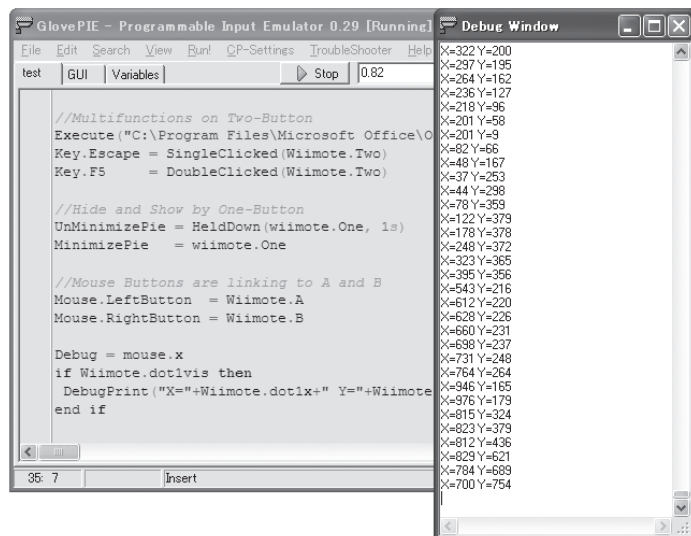
コード3-14 **GlovePIE** デバッグ用の表示 (3)

```
Debug = "mx="+mouse.x+" my="+mouse.y
if Wiimote.dot1vis then
  DebugPrint("X="+Wiimote.dot1x+" Y="+Wiimote.dot1y)
end if
```

「dot1vis」が「最初の1点目が見えるかどうか」、「dot1x」と「dot1y」はその座標を表しています。

さっそく実験してみましょう。最低でも1点の赤外線光源が見えればよいので、近くにセンサーバーがない場合は、太陽光や電球光源などを探してから実行しましょう。光源として使えるかどうか不安なときは、あらかじめWiinRemoteを使って確認すると便利です。

図3-39 別ウィンドウ表示に赤外線センサーの値が表示される



赤外線光源に向けたとき、デバッグウィンドウに値が書き足されていく様子が見えれば成功です！ 値が書き足されない場合は、WiinRemoteを使って赤外線が正しく認識できているか試してみましょう。ウィンドウが見つからないときは、GlovePIEのウィンドウの裏側に隠れていることがあるので調節してみてください。

さて、原理的にはこのdot1x、dot1yをマウスのx、yに割り当てれば完成、ということなのですが、実際にはそう単純ではありません。

コード3-15 **GlovePIE** このままではマウスポインタは常に右下に行ってしまう

```
Mouse.x = Wiimote.dot1x  
Mouse.y = Wiimote.dot1y
```

実は、このままでは画面上のマウスポインタとWiiRemoteの赤外線センサーが取得する値とのスケールが合っていません。さらに、動作方向の極性(正負)が合っていません。操作のイメージとしては、WiiRemoteを上に向けたらマウスポインタが上に、左に向けたらマウスポインタが左にいて欲しい感じがします(多くのWii用ゲームがそうであるように)。そのため、スケーリングをして、さらにある軸の正負を反転させる必要があります。

WiinRemoteやデバッグ出力で観察できるように、WiiRemoteの赤外線センサーは水平方向Xは[0,1023]の値域、垂直方向Yは[0,767]の値域をとります。WiiRemoteをセンサーバーに向かって上から下に向けると、WiinRemoteから見える赤外線の点像は上部から下部、値Xは0からはじまって767に向かって増加していきます。また、左から右に向けたとき、WiinRemoteから見える赤外線の点像は右側から左側へ、値Xは1023から始まって0に向かって減少していきます。

対して、左辺側のマウスポインタの座標は左上を(0,0)、右下を(1,1)とする[0,1]の値域となっています。これら異なる値域をスマートにつなぐ関数が「MapRange(x, a,b , c,d)」です。xという値域[a,b]をとる元の入力値を値域[c,d]に変換できます。よって、以下のように関数を記述することで、正しい操作感が得られるようになります。

コード3-16 **GlovePIE** 正しく変換される式

```
Mouse.x = MapRange(Wiimote.dot1x, 1023, 0 , 0,1)  
Mouse.y = MapRange(Wiimote.dot1y, 0, 767 , 0,1)
```

さて、これで完成！ と思いきや、実行してみると……どうでしょう？ 何か違和感はありませんか？ 人によっては感じないかもしれませんが、実際にこの赤外線マウスを使って、フォルダを

開き、ファイルを移動して……といった操作を試してみてください。すると、ポインタが高速にブルブルふるえて非常に操作しづらいはずです。この「ふるえ」にはいろんな原因があります。1つは赤外線センサーのノイズ(特に「ドリフトノイズ」と呼ばれることもあります)、センサーの内部がわかるわけではないので、ここでは単に「ノイズ」としておきましょう)、それから人間の手のものが持っている微少なふるえです。マウスの場合は机の上の摩擦で気にならないのですが、WiiRemoteの場合は空中で、しかもかなり高速に高精度の計測を行うので、このような人間の手のふるえの存在も理解しながら実装していく必要があります。信号処理の知識はこのようなケースで非常に役に立つでしょう。この場合は、急速な動きである高周波成分を除去する「ローパスフィルタ」が効果的ではないでしょうか。

難しい話はさておき、ローパスフィルタは平均をとることで実現できます。GlovePIE スクリプトには、こういった便利な関数が数多く実装されています。

コード3-17 **GlovePIE** 平均をとる関数

```
Smooth(x, [ExtraFrames, [DeadbandDistance]])
```

Smooth 関数は値 x を過去の [ExtraFrames] の件数分だけ使って平均を算出します。もし [DeadbandDistance] 以上の変更がなければ、すべてにおいて変更がなかったものとして判断します(省略時のデフォルト [DeadbandDistance] は 0)。

先ほどの Mouse への代入の式において、Smooth 関数を取り入れてみましょう。ExtraFrames は 3 にしておきます。これで過去 3 フレームの値を使って平均をとります。デバッグ出力も、以下のようにすっきり整理します。

コード3-18 **GlovePIE** ローパスフィルタを実装

```
if Wiimote.dotlvis then
  Mouse.x = MapRange(Smooth(Wiimote.dotlx, 3), 1023, 0, 0.1)
  Mouse.y = MapRange(Smooth(Wiimote.dotly, 3), 0, 767, 0.1)
  Debug = "mx="+Mouse.x+" my="+Mouse.y+" IRX="+Wiimote.dotlx+" IRY="+Wiimote.dotly
end if
```

どうでしょうか？ ぐっと安定感のあるマウスポインタになったのではないのでしょうか？ 信号処理の理論では、適切な ExtraFrames を算出する方法もありますが、今回の場合は経験的に入れても全く問題ないでしょう。3～10 程度で試してみるとよいでしょう(値が大きくなるとスムーズにはなりますが、その分処理が重たくなります)。

さて、これで高機能赤外線マウスは完成です。ここまでのすべてのスクリプトを掲載します。


コード3-19 **GlovePIE** 高機能赤外線マウス (完成版)

```
//WiiRemote IR mouse (basic)

//Key Binds
ExitScript = Wiimote.Home
Key.Up      = Wiimote.Up
Key.Down    = Wiimote.Down
Key.Left    = Wiimote.Left
Key.Right   = Wiimote.Right

//combination
Key.Alt+Tab = Wiimote.Plus
Key.Alt+F4  = Wiimote.Minus
Key.Windows+D = Wiimote.A and Wiimote.B


//Double Clicks
Key.Enter   = DoubleClicked(Wiimote.A)
Key.Delete  = DoubleClicked(Wiimote.B)

//Multifunctions on Two-Button
Execute("C:\Program Files\Microsoft Office\OFFICE11\POWERPNT.EXE") = 
HeldDown(Wiimote.Two, 1s)

Key.Escape = SingleClicked(Wiimote.Two)
Key.F5      = DoubleClicked(Wiimote.Two)

//Hide and Show by One-Button
UnMinimizePie = HeldDown(wiimote.One, 1s)
MinimizePie   = wiimote.One

//Mouse Buttons are linking to A and B
Mouse.LeftButton = Wiimote.A
Mouse.RightButton = Wiimote.B

//Assign Infrared as mouse input
if Wiimote.dotlvis then
    Mouse.x = MapRange(Smooth(Wiimote.dotlx, 3), 1023, 0, 0,1)
    Mouse.y = MapRange(Smooth(Wiimote.dotly, 3), 0, 767, 0,1)
    Debug = "mx="+Mouse.x+" my="+Mouse.y+" IRX="+Wiimote.dotlx+" 
    IRY="+Wiimote.dotly
end if
```

さて、実際に使い込んでボタンアサインや細かなパラメータなどを変更してみるとよいでしょう。なお、このスクリプトは赤外線が見えないときはマウス制御を奪いませんので、マウスと協

調作業することもできます。実行したままタスクバーにしまっておけば（最小化ボタンの左にタスクバー格納ボタンがあります）、何かと便利でカッコいいです。

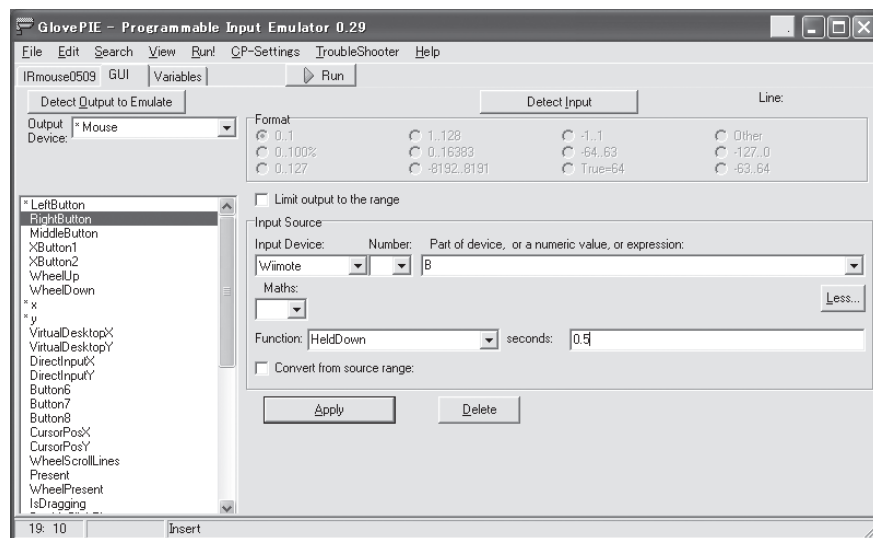
A + B ボタン同時押しでデスクトップを表示し、+ ボタンでアプリケーションを切り換え、B ボタンで右クリック、十字キーで細かな作業をし、A ボタンのダブルクリックで決定、2 ボタンで Esc……といった感じで、慣れると快感になってきます。スクリプトは Home ボタンを押せばいつでも終了できます。

GUIをつかったスクリプト記述支援

先ほどの高機能赤外線マウスでは、最後に赤外線の座標系をマウスの座標系に合わせるために、新しい MapRange という関数を使いました。GlovePIE は非常に高機能なスクリプト環境を装備しているので、そもそも「mouse」といったキーワードや便利な関数、コマンドなどを、付属のマニュアルだけで探してくるのは（しかも「未完成」と明記されています……）、なかなか骨の折れる作業です。

GlovePIE の新機能として、「GlovePIE GUI」が実装されています。グラフィカルな環境で、入出力の関係を結びつけると自動的にスクリプトが作成され、接続関係なども管理してくれるようです。マニュアルをよく読んでいくと「WiiRemote 関係の機能は GUI を使わないほうがよい」と書

図3-40 GlovePIE GUI



かれていますのですが、筆者が試した感覚では、WiiRemote→キーボード・マウスなどに使うのであれば十分使えますし、最終的にテキストで表現されるスクリプトがすべてなので、GlovePIEの機能をすばやく調べる上ではかなり役立つ環境ですので、紹介しておきます。

練習のために、先ほど最後に作成したWiiRemote→マウスの変換式をGUIで作成してみましょう。まずは先ほど作成した「高機能赤外線マウス」を保存して、いったんGlovePIEを起動し直してください。新しいGlovePIEが起動したら、メニューの「Edit」の下あたりにある「GUI」というタブをクリックしてください。ボタンが2つありますが「Detect Output Emulate」を押すと、使用可能な出力デバイスがコードされるので、「Output Device」から「Mouse」を選んでください。下にたくさんの操作可能なプロパティが現れるのでここから「x」を選び「Edit Manually」ボタンを押します。

「Input Device」というプルダウンがあるので、ここから「Wiimote」を選びます。「Number」は複数のWiiRemoteを接続した場合に使用しますのでここでは空白のままで、その隣の「Part of device, or a numeric value, or expression:」に「dot1x」と書いてみてください（すでに入力支援機能が働いているはずです）。プルダウンから選択するのもよいでしょう。

さらに「More」というボタンを押して、詳細を表示します。「Function:」というプルダウンで「Smooth」を選ぶと「Frames」が現れるので「3」を設定し、最後に「Convert from source range」から「min」と「max」をそれぞれ、デフォルトで入っている「0,1023」から「1023,0」に書き換えて、最後に「Apply」という大きなボタンを押して数秒待ちます。

GUIタブから「Untitled」と表示されている通常のスクリプトタブに戻ります。すると、

```
Mouse.x = MapRange(Smooth(Wiimote.dot1x, 3), 0,1023, 0,1)
```

というように、先ほどと全く同じスクリプトが自動生成されていることが確認できます。もちろん実行もできますが、同じ手順を再度「GUI」タブに戻って「mouse」→「y」→「Edit Manually」→「Wiimote」……というように選んで完成させてみてください。GUIで作成したスクリプトは必ず文末に現れるようです（GlovePIE自体は手続き型言語ではないので、順番が問題になることはあまりありません）。

今回は新規スクリプトで実験しましたが、既存のスクリプトでも十分に利用できます。新機能を追加するとき、新しくキーボードに割り当てたいが名称がわからないとき、適切な値域を扱いたいが基本的な値を調べるのが面倒なとき……、などに非常に役に立ちます。WiiRemoteへの出力（たとえばLEDやバイブレーター、スピーカーなど）もまだサポートされていませんが、MIDIなどはかなり整備されているようなので、今後、アップデートされるようであれば機能強化が期待されるところです。

サンプルスクリプトを読もう

さて、高性能赤外線マウスを自力で完成することができたあなたは、もう他のPIEスクリプトを読むのが難しくなくなっているはずです。「WiimoteScripts」フォルダの中にあるさまざまな例を読んでみることをお勧めします。

PCゲームのキーボード・マウスのエミュレーションが多いのですが、より簡単に操作できるよう、さまざまなスクリプティングテクニックが読み取れます。他にもこのディレクトリには、WiiRemoteの基本機能に対する解説的なスクリプト、たとえばバッテリー残量を表示する「WiiBattery.PIE」や、MIDIと組み合わせた簡易ドラムセット「WiiDrums2.PIE」、ヌンチャクまで使った本格ギター「Wiitar」、加速度から距離を求めようとする「TestVelocityPos.PIE」など参考になるものがあります（ならないものも、たくさんあります！）。

ウィンドウの境界を振動で通知

バイブレーターは最初に試したとおり Wiimote.Rumble で制御できますので、マウスポインタが現在のウィンドウの境界にぶつかったら、振動するようなスクリプトを追加してみます。

コード3-20 **GlovePIE** ウィンドウの境界でバイブレーターを鳴らす

```
var.hit = false;
var.hit = mouse.CursorPosX<Window.Left or
    mouse.CursorPosX>Window.Left+Window.Width
var.hit = var.hit or mouse.CursorPosY<Window.Top or
    mouse.CursorPosY>Window.Top+Window.Height
Wiimote.Rumble =var.hit;
```

変数 var.hit は現在のマウスポインタの位置と、現在のウィンドウ境界を比べてポインタが外にいる場合 true になります。これを Wiimote.Rubmle につなげています。

これを応用すれば、弱視の方や何らかの理由でマウスが使えないかた（たとえば産まれたばかりの赤ん坊を抱っこしている、など）でもエクスプローラのようなGUIが触りやすくなるかもしれません。

LEDでタイマーを作る

学会などでのプレゼンテーションで、なかなか時間内に収まらなかったり、逆に時間が気になりすぎて、堂々と話ができなかったりしませんか？ このスクリプトは、手元に持っているであ

ろう WiiRemote の LED を使って、残り時間を表現します。

コード3-21 **GlovePIE** LEDによるタイマー表示

```
Wiimote.Led1 = HeldDown(true, 5s);  
Wiimote.Led2 = HeldDown(true, 10s);  
Wiimote.Led3 = HeldDown(true, 20s);  
Wiimote.Led4 = HeldDown(true, 1 minutes);
```

この例では WiiRemote の青色 LED が左から順に、5 秒、10 秒、20 秒、1 分……というように点灯して最後は 4 つ全てが点灯します。Windows のプログレスバーのようなイメージですね。あらかじめプレゼンテーションの構成とともに、適切なラップタイムを設定しておくのもよいでしょう。

スピーカーを鳴らす

最後にスピーカーを使ってみます。GlovePIE で使える WiiRemote のスピーカー機能は非常に低レベルな機能しか提供されていません。周波数とボリュームを設定し、それが適切な長さだけ再生されるように自分で管理する必要があります。WAV ファイルを再生できるような機能もそのうち出てくるのかもしれませんが……。このスクリプトはその構造を理解するための目的で書かれています。1 ボタンを押すたびにさまざまな音（ノイズ？）が鳴ります。

コード3-22 **GlovePIE** スピーカーを鳴らす

```
if Wiimote.One then  
  Wiimote.Volume = 1.00  
  Wiimote.SampleRate = 3640 Hz  
  while ( true )  
    if var.f<=0 then  
      var.f=360  
    end if  
    // wait 10ms  
    debugprint("f="+var.f + "Cos:"+cos(var.f)+"  
      Freq:"+abs(500*cos(var.f)+1000));  
    Wiimote.Frequency =abs(500*cos(var.f)+1000)  
    // Wiimote.Frequency = 261.62  
    var.f=var.f-10;  
  end while  
else  
  wait 1000ms  
  Wiimote.Frequency = 0Hz  
end if
```

コメントアウトされていますが、いくつかの周波数（たとえば 261.62Hz）などでキレイに聞こえる音が存在します。インターネットを探すと音階と周波数の関係についてのデータがあるので、その周波数を使えば音階を表現することも可能でしょう（やはり、音質はそこそこですが……）。

コマンドラインによる起動自動化

PIE スクリプトを作り込んでいくと、今度は GlovePIE から読み込んで実行……という流れが面倒になってくるはずです。アート作品の展示などに使う場合は、ぜひとも自動起動させたいところです。GlovePIE はコマンドラインからの起動もサポートしています。

起動時のオプションは、GlovePIE.exe に続いて以下の 3 種類の方法でファイル名を指定します。

- **filename.PIE**

起動時に開いておく PIE スクリプトを指定。「Run」メニューから「Run!」を選ばない限り実行されない

- **-filename.PIE**

上記と同様、起動後は自動でスクリプトが開始する

- **/r:filename.EXE**

PIE が起動したときに自動開始する、他の「.EXE」形式の実行ファイルを指定する。たとえばゲームと PIE スクリプトを同時起動など

空白などが入る場合はクォーテーション「"」で囲むのを忘れずに。拡張子「.PIE」は含んでも含まなくてもよいようです。

たとえば、以下のようなバッチファイルを書いておけば、すぐに作成した赤外線マウスが使えます。

コード 3-23 **GlovePIE** 自動起動させるバッチファイル

```
startIRmouse.bat

"C:\YGlovePIE029\YGlovePIE.exe" -"C:\YGlovePIE029\IRmouse"
```

もちろんIRmouse.PIEは指定した場所に保存しておいてください。

GlovePIEの可能性はGlovePIEスクリプトの使いこなしにあるといっても過言ではありません。この節で学んだとおり、GlovePIEによるスクリプティングだけでもかなりのことができます。またGlovePIEは、MIDIだけでなくジョイスティックやデータグローブ、OSCやSpeechAPIといった他のインタラクション技術でも使われているような、基本的なユーザーインタフェース周辺機器のエミュレータやコンバータとしてのポテンシャルもとても高いソフトウェアです。かなり、さまざまな言語の仕様を取り込んでいるので、プログラミング言語マニアとしても勉強になります。

これでこの章はおしまいです。Bluetooth接続やWiinRemoteやGlovePIEなど、基本となるツールの使い方はしっかりと理解できたでしょうか？ この先のプログラミング編に入る前に「わかったつもり」を脱しておいてください。また、ここでの知識は実際に新しいハードウェアを購入したり、新しい作品を作ったり、展示を行う上での改造を行ったり、といったときに必ず役に立つものです。

