WiiRemoteComponents

ユーザーマニュアル

平成 22 年 5 月 6 日

芝浦工業大学 水川研究室
〒135-8548
東京都江東区豊洲 3-7-5
芝浦工業大学研究棟 11Q32 水川研究室
TEL：03-5859-8209
FAX：03-5859-8201
Mail：shibaura.hri.goiken@gmail.com

指導教員：水川 真
作成者：
田中 基雅
藤田 恒彦
鷹栖 尭大

目次

1. はじめに

本書は，任天堂株式会社より販売されている Wii リモコン(以下，WiiRemote)及びその拡張コントローラのコンポーネントの使用マニュアルです.

WiiRemote とその各種拡張コントローラを利用するためのモジュールを RT-Component 化した「WiiRemoteComponents」の仕様，利用方法を記述しました.

1.1. WiiRemote の概要

WiiRemote は，任天堂株式会社より販売されている家庭用ゲーム機「Wii」を操作するインタフェースです．WiiRemote には，各種ボタンと加速度センサ，CMOS センサと赤外線 LED を用いたポインティング機能等があります.

Wii と WiiRemote は Bluetooth によって通信を行っており，Bluetooth を利用する事により，WiiRemote-PC 間の通信が可能になります.

1.2. 本書の構成

本書は以下の構成で記述します.

章番号	タイトル	構成
2.	WiiRemoteComponents	開発した RT-Components の機能概要を説明します.
3.	WiiRemoteComponents をご利用いただく前に準備いただくこと	開発した RT-Component を利用するに当たり準備をしていただくものを説明します.
4.	WiiRemoteCompoents の利用	開発した RT-Component が利用可能になるまでの流れを説明します.
5.	動作環境・開発環境	開発に当たり確認した動作環境・開発環境を説明します.
6.	ライセンス	本 RT-Components のライセンスを説明します.
7.	連絡先	本 RT-Components に関する問い合わせ先を記します.

2. WiiRemoteComponents

「WiiRemoteComponents」は，WiiRemote とその各種拡張コントローラを利用するための RT-Component 群です．WiiRemote との通信を行う「WiiRemoteAgent」と，拡張コントローラごとの「Wii～Translator」から構成されます. 本章では, WiiRemoteComponents の仕様と概要示します.

2.1. WiiRemoteAgent

2.1.1. WiiRemote 通信概要

WiiRemote は，Bluetooth により PC と通信することが可能となります．Bluetooth を利用し WiiRemote を PC と接続した場合，HID(Human Interface Device)として認識されます．(通常のマウスなどと同じ扱いです．）そのため，データ通信は WiiRemote で規定している ReportID により行います.

2.1.2. コンポーネント概要

WiiRemoteAgent は WiiRemote と通信を行う RT-Component です．WiiRemoteAgent の外観を 図 2-1 に示します． WiiRemoteAgent は各種 Translator と接続することを想定し，WiiRemote から取得した生データを TimedOctetSeq 型で出力をしています．(0 バイト目が ReportID となっており，何のデータか解析することも可能です．）そのため，データ解析部分は各種 Translator が行っています．入力ポートは WiiRemote の LED および振動の制御を行います．WiiRemoteAgent のポートの仕様一覧を表 2-1 に示します．サービスポートは WiiRemote の拡張コントローラのキャリブレーション，拡張コントローラの状態，バッテリ残量（単位：パーセント）を提供しています．サービスポートの仕様を 表 2-2 に示します.

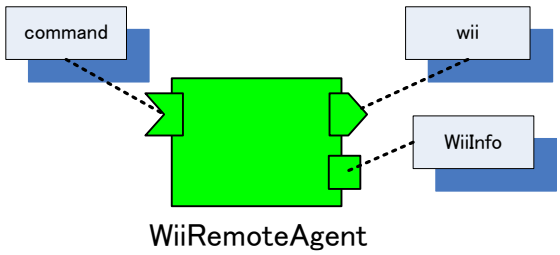


図 2-1 WiiRemoteAgent 外観

表 2-1 WiiRemoteAgent　インポート仕様一覧

インタフェースタイプ	ポート名	データタイプ	詳細
Out Port(Data Port)	wii	TimedOctetSeq	WiiRemote から受信した生データ
In Port(Data Port)	command	TimedOctet	WiiRemote の LED と振動を制御するポートです．最下位ビットに“1”が(XXXXXXX1)ある場合振動します．また，上位 4 ビットの各ビットが LED の点灯パターンになっています．たとえば，入力“10100001“（0xA1）の場合，LED が交互に点灯し，振動します.
Service Provider	WiiInfo	WiiInfo	表 2-2 を参照

表 2-2 WiiInfo.idl 仕様一覧

WiiInfo		
メソッド	戻り値	詳細
get_wii_mote_calib(void)	Calibration (sequence<octet>)	WiiRemote 本体のキャリブレーションデータの取得
get_wii_ext_calib(void)	Calibaration	WiiRemmote に接続されたキャリブレーションデータの取得
get_balance_calib(void)	Calibaration	BalanceBoard のキャリブレーションデータの取得
get_wii_ext_state(void)	Extention (sequence<octet>)	拡張コントローラの状態の取得コマンド
get_battery_value(void)	octet	バッテリの残存容量の取得コマンド

※Calibration/Extention は生データを送信しています．利用の際には解析を行ってください

また，各種拡張コントローラの設定はコンフィギュレーションにより行っています．コンフィギュレーションの詳細を表 2-3 に示します．“otherExt”は今後発売された拡張コントローラの実装を行いたい時に利用していただくことを想定しております．そのため，現在は未実装となっております．また，現在複数の拡張コントローラを接続することは不可能となっております．クラスの実装は付録に示します.

表 2-3 WiiRemoteAgent Configuration 一覧

変数名	データタイプ	初期値	詳細
ClassicController	bool	0	利用する拡張コントローラの値を“1”にすることで利用することが可能となります．ただし，OtherExt は WiiRemote の拡張コントローラが今後発売された場合に利用するために準備したものであり現在は未実装です.
Nunchaku	bool	0	
BalanceBoard	bool	0	
MotionPlus	bool	0	
OtherExt	bool	0	

注意点：WiiRemoteAgent は Activate 時にコンフィギュレーションを設定した拡張コントローラおよび WiiRemote のキャリブレーションデータを取得しています．WiiRemoteAgent の仕様としてキャリブレーションデータの取得に成功するまで状態遷移を行わない仕様としています.

2.2. WiiRemoteTranslator

2.2.1. WiiRemote 機能概要

WiiRemote はボタンデータおよび，3 軸の加速度センサにより WiiRemote の姿勢等を取得することができます.

2.2.2. コンポーネント概要

WiiRemoteTranslator の外観を図 2-2 に示します.

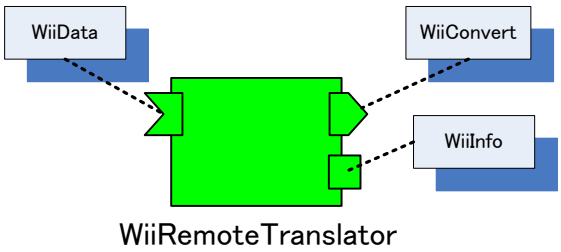


図 2-2 WiiRemoteTranslator の外観

WiiRemoteAgent から受信したデータを WiiRemote のボタンの状態，及び加速度データに変換するコンポーネントです． WiiRemoteTranslator のポートの仕様一覧を表 2-4 に，アウトポートの仕様を表 2-6 に示します．また，WiiRemoteTranslator にはコンフィグレーションによりボタン・加速度の状態をコンソールに表示することができます．WiiRemoteTranslator Configuration 仕様を表 2-5 に示します.

表 2-4 WiiRemoteTranslator のポートの仕様一覧

インタフェースタイプ	ポート名	データタイプ	詳細
In Port(Data Port)	WiiData	TimedOctetSeq	WiiRemoteAgent の wii ポートから取得した生データ
Out Port(Data Port)	WiiConvert	TimedShortSeq	表 2-6 を参照
Service Consumer	WiiInfo	WiiInfo	表 2-2 を参照

表 2-5 WiiRemoteTranslator Configuration 仕様

変数名	データタイプ	初期値	詳細
debug	bool	0	値を”1”に設定すると，コントローラの状態をコンソールで確認することができます

表 2-6 WiiRemoteTranslator アウトポート仕様一覧

WiiConvert(データタイプ：TimedShortSeq)	
配列番号	詳細
[0]	十字キー・上ボタンの状態(BOOL 値)
[1]	十字キー・右ボタンの状態(BOOL 値)
[2]	十字キー・下ボタンの状態(BOOL 値)
[3]	十字キー・左ボタンの状態(BOOL 値)
[4]	A ボタンの状態(BOOL 値)
[5]	B ボタンの状態(BOOL 値)
[6]	＋ボタンの状態(BOOL 値)
[7]	Home ボタンの状態(BOOL 値)
[8]	－ボタンの状態(BOOL 値)
[9]	1 ボタンの状態(BOOL 値)
[10]	2 ボタンの状態(BOOL 値)
[11]	X 軸方向の加速度(＊図 2-3)
[12]	Y 軸方向の加速度(＊図 2-3)
[13]	Z 軸方向の加速度(＊図 2-3)

WiiRemoteTranslator における座標系を図 2-3 に示します.

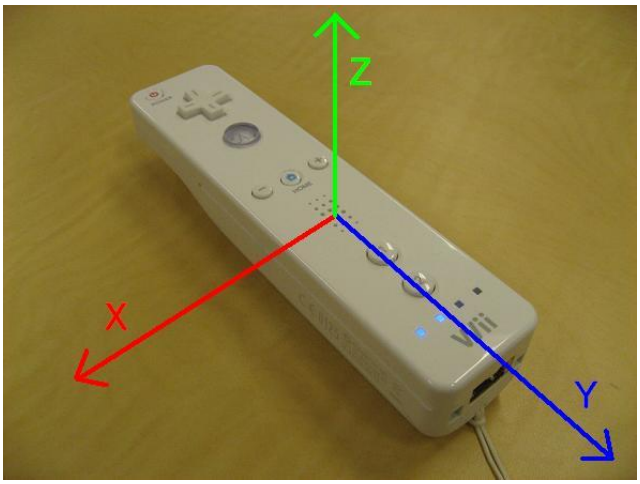


図 2-3 WiiRemoteTranslator における座標系

注意点：WiiRemoteTranslator は Activate 時に WiiRemote のキャリブレーションデータを取得しています. WiiRemoteTranslator の仕様としてキャリブレーションデータの取得に成功するまで状態遷移を行わない仕様としています.

2.3. WiiNunchakuTranslator

2.3.1. ヌンチャク機能概要

Wii Remote の拡張コントローラであるヌンチャクは，WiiRemote 同様，3 軸加速度センサを内蔵しています. また，その他に 2 つのボタンとアナログスティックを備えています.

2.3.2. コンポーネント概要

WiiNunchakuTranslator の外観を図 2-4 に示します.

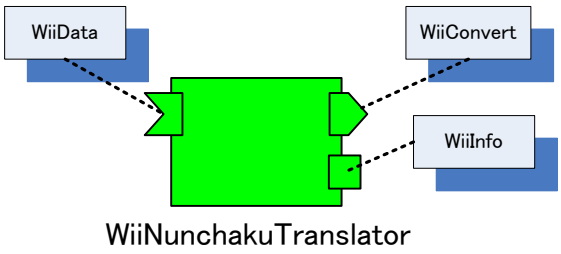


図 2-4 WiiNunchakuTranslator の外観

WiiRemoteAgent から受信したデータを WiiRemote のボタンの状態，加速度，及びヌンチャクのボタンデータ，加速度データに変換するコンポーネントです.

WiiNunchakuTranslator のポートの仕様一覧を表 2-7 に，アウトポートの仕様を表 2-9 に示します．また，WiiNunchakuTranslator にはコンフィグレーションによりボタン・加速度の状態をコンソールに表示することができます． WiiNunchakuTranslator Configuration 仕様を表 2-8 に示します.

表 2-7 WiiRemoteTranslator のポートの仕様一覧

インタフェースタイプ	ポート名	データタイプ	詳細
In Port(Data Port)	WiiData	TimedOctetSeq	WiiRemoteAgent の wii ポートから取得した生データ
Out Port(Data Port)	WiiConvert	TimedShortSeq	表 2-9 を参照
Service Consumer	WiiInfo	WiiInfo	表 2-2 を参照

表 2-8 WiiNunchakuTranslator Configuration 仕様

変数名	データタイプ	初期値	詳細
debug	bool	0	値を”1”に設定すると，コントローラの状態をコンソールで確認することができます

表 2-9 WiiNunchakuTranslator アウトポート仕様一覧

WiiConvert(データタイプ：TimedShortSeq)	
配列番号	詳細
[0~13]	WiiRemoteTranslator を参照(全 Translator 共通)
[14]	C ボタンの状態(BOOL 値)
[15]	Z ボタンの状態(BOOL 値)
[16]	アナログスティックの X 方向入力値 (解放時 0，最大まで左に傾けると-100，同右で 100)
[17]	アナログスティックの Y 方向入力値 (解放時 0，最大まで左に傾けると-100，同右で 100)
[18]	X 軸方向の加速度(＊図 2-5)
[19]	Y 軸方向の加速度(＊図 2-5)
[20]	Z 軸方向の加速度(＊図 2-5)

WiiNunchakuTranslator における座標系を図 2-5 に示します.

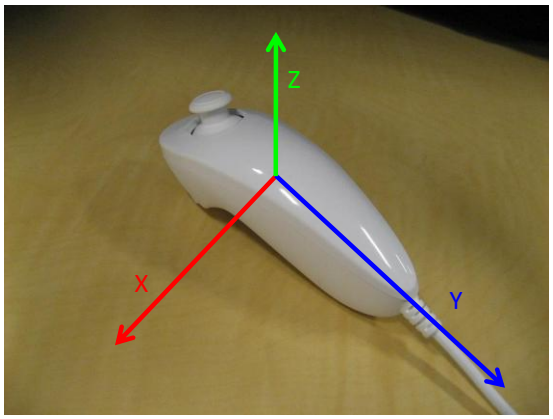


図 2-5 WiiNunchakuTranslator における座標系

注意点：WiiNunchakuTranslator は Activate 時にコンフィギュレーションを設定した拡張コントローラおよび WiiRemote のキャリブレーションデータを取得しています. WiiNunchakuTranslator の仕様としてキャリブレーションデータの取得に成功するまで状態遷移を行わない仕様としています.

2.4. WiiClassicControllerTranslator

2.4.1. クラシックコントローラ機能概要

Wii クラシックコントローラは, ボタンやアナログスティックを備えた従来のゲームコントローラ型の拡張コントローラです.

[1] クラシックコントローラ

クラシックコントローラからは，各種ボタンとアナログスティック，L ボタンと R ボタンのアナログ値を取得する事ができます．ボタン配置を図 2-6 及び図 2-7 に示します.

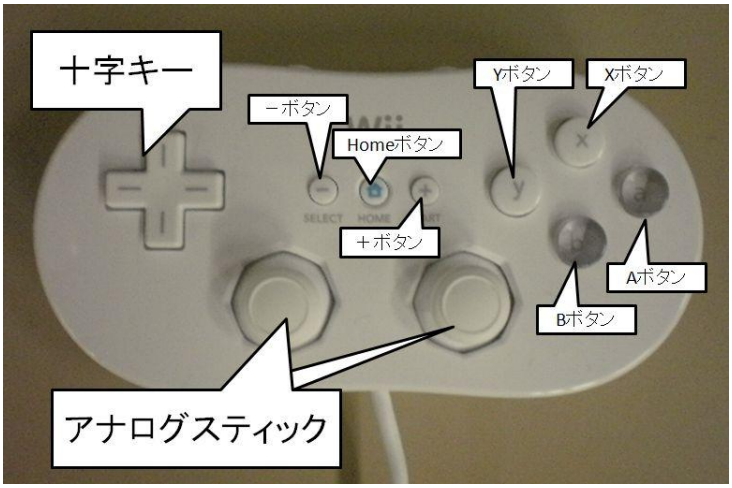


図 2-6 クラシックコントローラボタン配置(1)

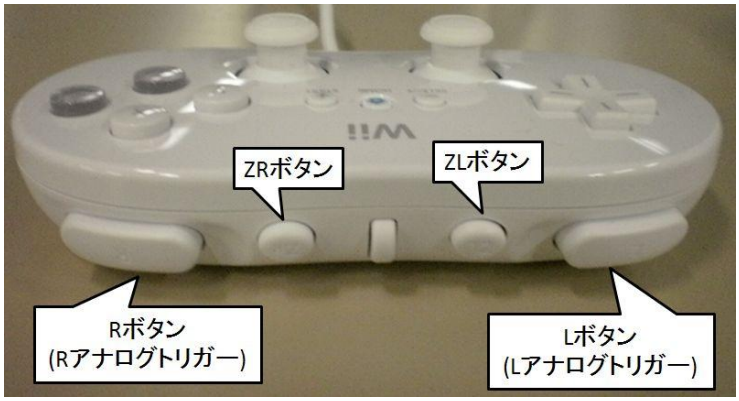


図 2-7 クラシックコントローラボタン配置(2)

[2] クラシックコントローラ PRO

クラシックコントローラ PRO は，新型のクラシックコントローラで，一部のボタン配置が変更され，L ボタンと R ボタンのアナログ入力が無くなっています. ボタン配置を図 2-8 及び図 2-9 に示します.

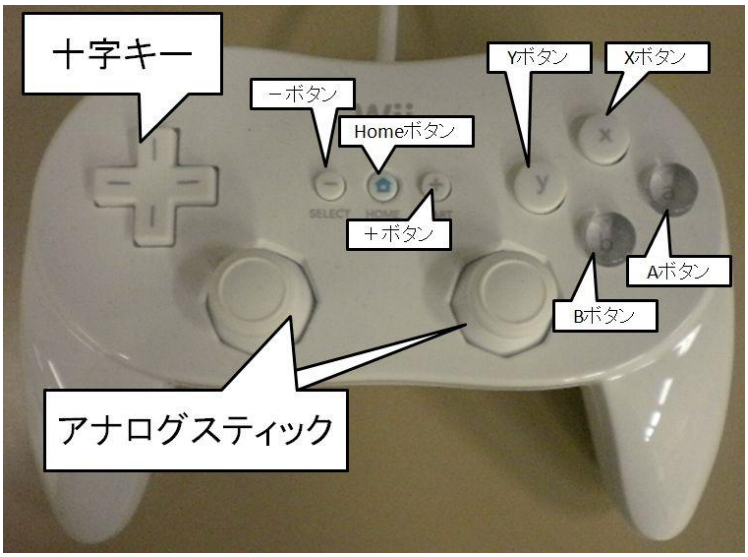


図 2-8 クラシックコントローラ PRO ボタン配置(1)

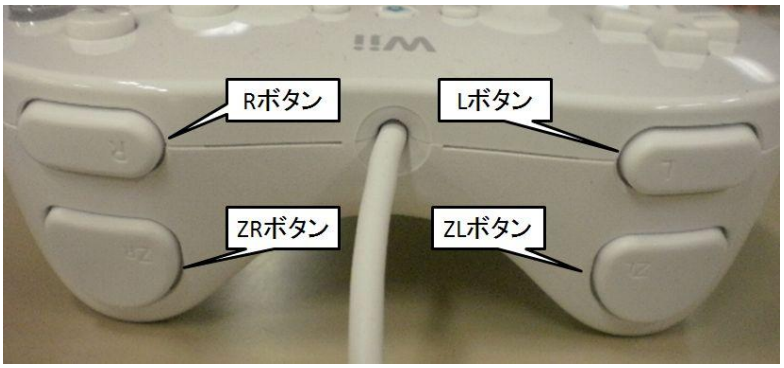


図 2-9 クラシックコントローラ PRO ボタン配置(2)

2.4.2. コンポーネント概要

WiiClassicControllerTranslator の外観を図 2-10 に示します.

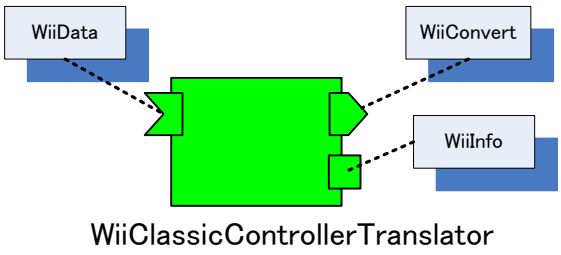


図 2-10 WiiClassicContorllerTranslator 外観

WiiRemoteAgent から受信したデータを WiiRemote ボタンの状態，加速度，及びクラッシクコントローラの各ボタン・アナログ値に変換するコンポーネントです.

WiiClassicControllerTranslator のポートの仕様一覧を表 2-10 に，アウトポートの仕様を表 2-12 に示します．また，WiiClassicControllerTranslator にはコンフィグレーションによりボタンの状態をコンソールに表示することができます．WiiClassicControllerTranslator Configuration 仕様を表 2-11 に示します.

表 2-10 WiiClassicControllerTranslator のポートの仕様一覧

インタフェースタイプ	ポート名	データタイプ	詳細
In Port(Data Port)	WiiData	TimedOctetSeq	WiiRemoteAgent の wii ポートから取得した生データ
Out Port(Data Port)	WiiConvert	TimedShortSeq	表 2-12 を参照
Service Consumer	WiiInfo	WiiInfo	表 2-2 を参照

表 2-11 WiiClassicControllerTranslator Configuration 仕様

変数名	データタイプ	初期値	詳細
debug	bool	0	値を”1”に設定すると，コントローラの状態をコンソールで確認することができます

表 2-12 WiiClassicControllerTranslator アウトポート仕様一覧

WiiConvert(データタイプ：TimedShortSeq)	
配列番号	詳細
[0]～[13]	WiiRemoteTranslator を参照(全 Translator 共通)
[14]	A ボタンの状態(BOOL 値)
[15]	B ボタンの状態(BOOL 値)
[16]	X ボタンの状態(BOOL 値)
[17]	Y ボタンの状態(BOOL 値)
[18]	－ボタンの状態(BOOL 値)
[19]	Home ボタンの状態(BOOL 値)
[20]	＋ボタンの状態(BOOL 値)
[21]	十字キー・上ボタンの状態(BOOL 値)
[22]	十字キー・下ボタンの状態(BOOL 値)
[23]	十字キー・左ボタンの状態(BOOL 値)
[24]	十字キー・右ボタンの状態(BOOL 値)
[25]	L ボタンの状態(BOOL 値)
[26]	R ボタンの状態(BOOL 値)
[27]	ZL ボタンの状態(BOOL 値)
[28]	ZR ボタンの状態(BOOL 値)
[29]	L ボタンのアナログ入力値 (最小値 0，最大値 100，PRO の場合は解放で 0，押下で 100)
[30]	R ボタンのアナログ入力値 (最小値 0，最大値 100，PRO の場合は解放で 0，押下で 100)
[31]	左アナログスティックの X 方向入力値 (解放時 0，最大まで左に傾けると-100，同右で 100)
[32]	左アナログスティックの Y 方向入力値 (解放時 0，最大まで下に傾けると-100，同上で 100)
[33]	右アナログスティックの X 方向入力値(-100～100) (解放時 0，最大まで左に傾けると-100，同右で 100)
[34]	右アナログスティックの Y 方向入力値(-100～100) (解放時 0，最大まで下に傾けると-100，同上で 100)

注意点：WiiClassicControllerTranslator は Activate 時にコンフィギュレーションを設定した拡張コントローラおよび WiiRemote のキャリブレーションデータを取得しています. WiiClassicControllerTranslator の仕様としてキャリブレーションデータの取得に成功するまで状態遷移を行わない仕様としています.

2.5. WiiBlanceBoardTranslator

2.5.1. バランスボード機能概要

WiiBalanceBoard は本体 4 隅に圧力センサを搭載しており，体重や，重心位置を取得できます.

2.5.2. コンポーネント概要

WiiBalanceBoardTranslator の外観を図 2-11 に示します.

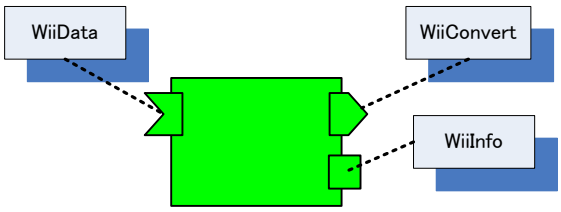


図 2-11 WiiBalanceBoardTranslator の外観

WiiRemoteAgent から受信したデータを解析し，圧力センサの生データ，4 隅にかかる重量データ，重心座標(x,y の 2 次元)に変換するコンポーネントです.

WiiBalanceBoardTranslator のポートの仕様一覧を表 2-13 に，アウトポートの仕様を表 2-15 に示します．また，WiiBalanceBoardTranslator にはコンフィグレーションにより重心位置やセンサの状態をコンソールに表示することができます．WiiBalanceBoardTranslator Configuration 仕様を表 2-14 に示します.

表 2-13 WiiBalanceBoardTranslator のポートの仕様一覧

インタフェースタイプ	ポート名	データタイプ	詳細
In Port(Data Port)	WiiData	TimedOctetSeq	WiiRemoteAgent の wii ポートから取得した生データ
Out Port(Data Port)	WiiConvert	TimedDoubleSeq	表 2-15 を参照
Service Consumer	WiiInfo	WiiInfo	表 2-2 を参照

表 2-14 WiiBalanceBoardTranslator Configuration 仕様

変数名	データタイプ	初期値	詳細
debug	bool	0	値を”1”に設定すると，コントローラの状態をコンソールで確認することができます

表 2-15 WiiBalanceBoardTranslator のアウトポート仕様一覧

WiiBalanceBoardTranslator (データタイプ：TimedDoubleSeq)	
配列番号	詳細
[0]	電源ボタンの状態(BOOL 値)
[1]	X 軸の重心座標(図 2-12)
[2]	Y 軸の重心座標(図 2-12)
[3]	左上にかかる重量データ
[4]	右上にかかる重量データ
[5]	右下にかかる重量データ
[5]	左下にかかる重量データ
[7]	左上センサの生データ
[8]	右上センサの生データ
[9]	右下センサの生データ
[10]	左下センサの生データ

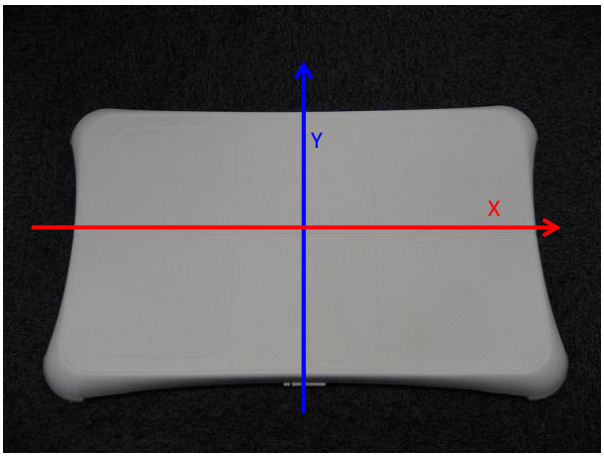


図 2-12 WiiBalanceBoardTranslator における座標系

注意点：WiiBalanceBoardTranslator は Activate 時に BalanceBoard のキャリブレーションデータを取得しています．WiiBalanceBoardTranslator の仕様としてキャリブレーションデータの取得に成功するまで状態遷移を行わない仕様としています．

2.6. WiiMotionPlusTranslator

2.6.1. モーションプラス機能概要

WiiMotionPlus では，ジャイロセンサにより yaw，pitch，roll のスピードを出力します．WiiMotionPlus の yaw，pitch，roll の各軸を図 2-13 に示します．

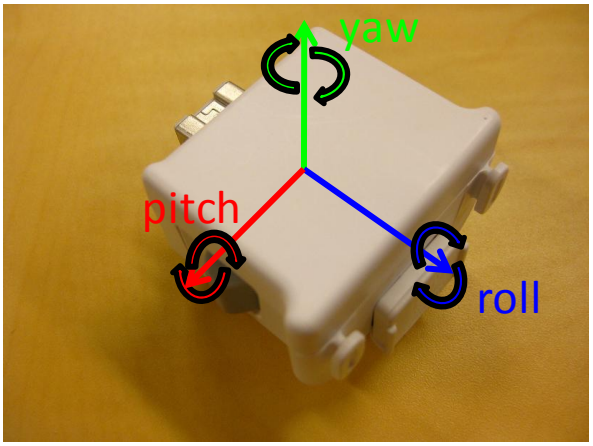


図 2-13 WiiMotionPlus の回転軸

2.6.2. コンポーネント概要

WiiMotionPlusTranslator は，WiiRemoteAgent から出力された生データ(ReportID “0x35”)を利用しています．WiiRemoteTranslator の外観を図 2-14 に示します．

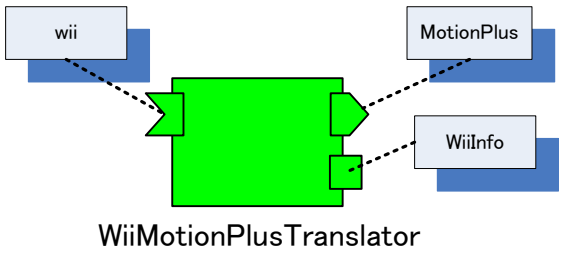


図 2-14 WiiMotionPlusTranslator 外観

WiiRemoteAgent から受信したデータを WiiRemote のボタン，加速度および MotionPlus のデータへ変換するコンポーネントです．WiiMotionPlusTranslator のポートの仕様一覧を表 2-16 に，アウトポートの仕様を表 2-18 に示します．また，WiiMotionPlusTranslator にはコンフィグレーションによりボタンやセンサの状態をコンソールに表示することができます．WiiMotionPlusTranslator Configuration 仕様を表 2-17 に示します．

表 2-16 WiiMotionPlus のポートの仕様一覧

インタフェースタイプ	ポート名	データタイプ	詳細
In Port(Data Port)	WiiData	TimedOctetSeq	WiiRemoteAgent の wii ポートから取得した生データ
Out Port(Data Port)	WiiConvert	TimedShortSeq	表 2-18 を参照
Service Consumer	WiiInfo	WiiInfo	表 2-2 を参照

表 2-17 WiiMotionPlusTranslator Configuration 仕様

変数名	データタイプ	初期値	詳細
debug	bool	0	値を"1"に設定すると, コントローラの状態をコンソールで確認することができます

表 2-18 WiiMotionPlusTranslator アウトポート仕様一覧

WiiConvert(データタイプ：TimedShortSeq)	
配列番号	詳細
[0]～[13]	WiiRemoteTranslator を参照(全 Translator 共通)
[14]	MotionPlus の yaw の生データ
[15]	MotionPlus の pitch の生データ
[16]	MotionPlus の roll の生データ
[17]	MotionPlus の yaw を速度に変換した値
[18]	MotionPlus の pitch を速度に変換した値
[19]	MotionPlus の roll を速度に変換した値

※キャリブレーションデータの使用方法が現在不明なため 10 回分のデータを平均して利用しています.

注意点：WiiMotionPlusTranslator では MotionPlus のキャリブレーションデータの決定を行うため 11 回分の生データを利用しています．そのため，起動直後にアウトポートからデータが出力されないことがあります. (キャリブレーションが終了するまで WiiRemote を動かさないでください．）また，WiiRemoteAgent から WiiRemote のキャリブレーションデータを取得できるまで, OnExecute へ状態遷移を行わない仕様としております. そのため, RT System Editor 上で状態遷移をしたか確認できるまで時間がかかることがあります.

3. WiiRemoteComponents をご利用の前にご準備頂きたい事

3.1. 準備して頂く製品

WiiRemoteCompoents を利用するためには，任天堂株式会社より販売されている WiiRemote または BalanceBoard が必要になります. また, PC と通信をするため Bluetooth モジュールが必要になります.

本書では，プリンステクノロジー株式会社より販売されている PTM-UBT3S を利用した場合の説明をします．(ただし，WiiRemoteAgent は株式会社東芝製の Bluetooth の StackDriver が利用可能な製品のみサポートを行います．Windows が標準で準備をしている Driver は利用不可能です．)

3.2. ダウンロードして頂くファイル

WiiRemoteComponents は Windows 用のデバイスドライバ開発キット(WDK)を利用しています．必要なソースコード・ライブラリ・DLL は以下に示す 6 つです．ただし，WiiRemoteAgent は最新版の WDK Ver.7 に対応できておりません．修正を行っておりますが，修正終了まで WDK Ver.6 を利用してください.

- guiddef.h
- hidpi.h
- hidsdi.h
- hidusage.h
- hid.lib
- setupapi.dll

以上の 6 つは以下に記載する方法で入手することが可能です.

(1) Microsoft Connect にサインインします．(未登録の場合は登録をして下さい)
(https://connect.microsoft.com/default.aspx)
(2) WindowsDriverKit
(https://connect.microsoft.com/site148) に行きます.
(3) ダウンロードページの, WDK の Archive より, WDK Ver.6 をダウンロードします.
(4) インストールしたフォルダ(WDK)の中から必要なライブラリ及びヘッダファイルをコピーします.

4. WiiRemoteComponents の利用

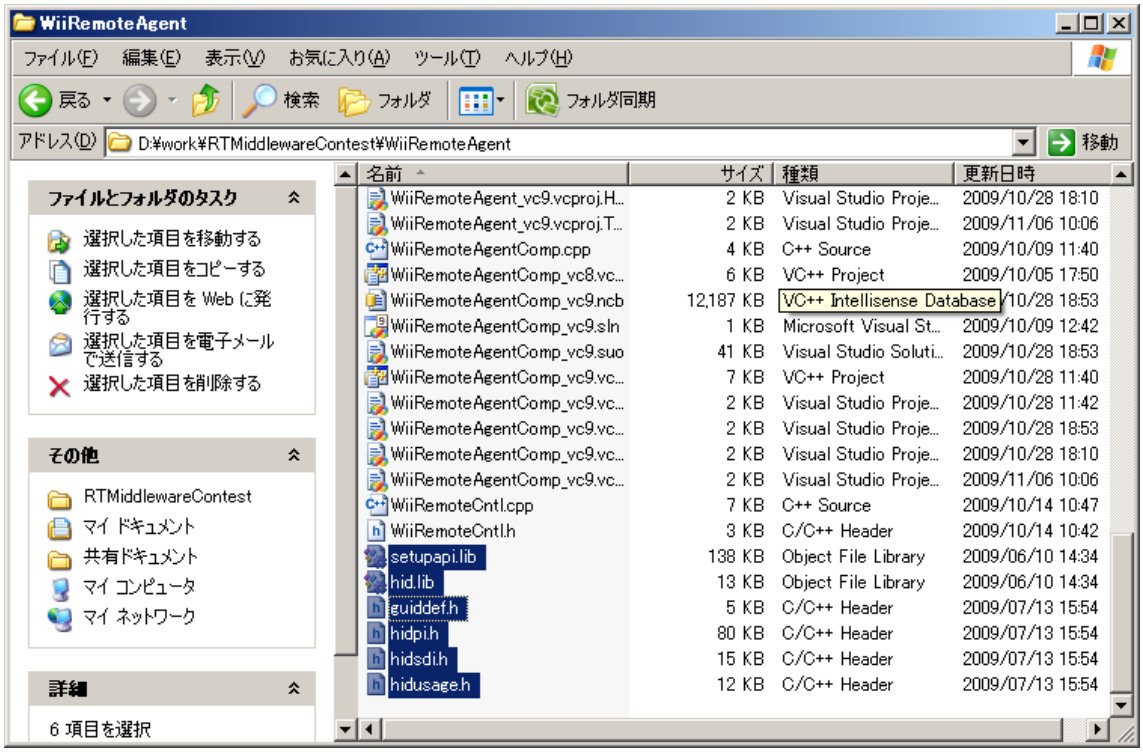
4.1. ビルド方法

以下は，OpenRTM-aist-1.0.0Release および Windows デバイスドライバ開発キット(WDK)をインストール済みであることを前提として説明いたします.

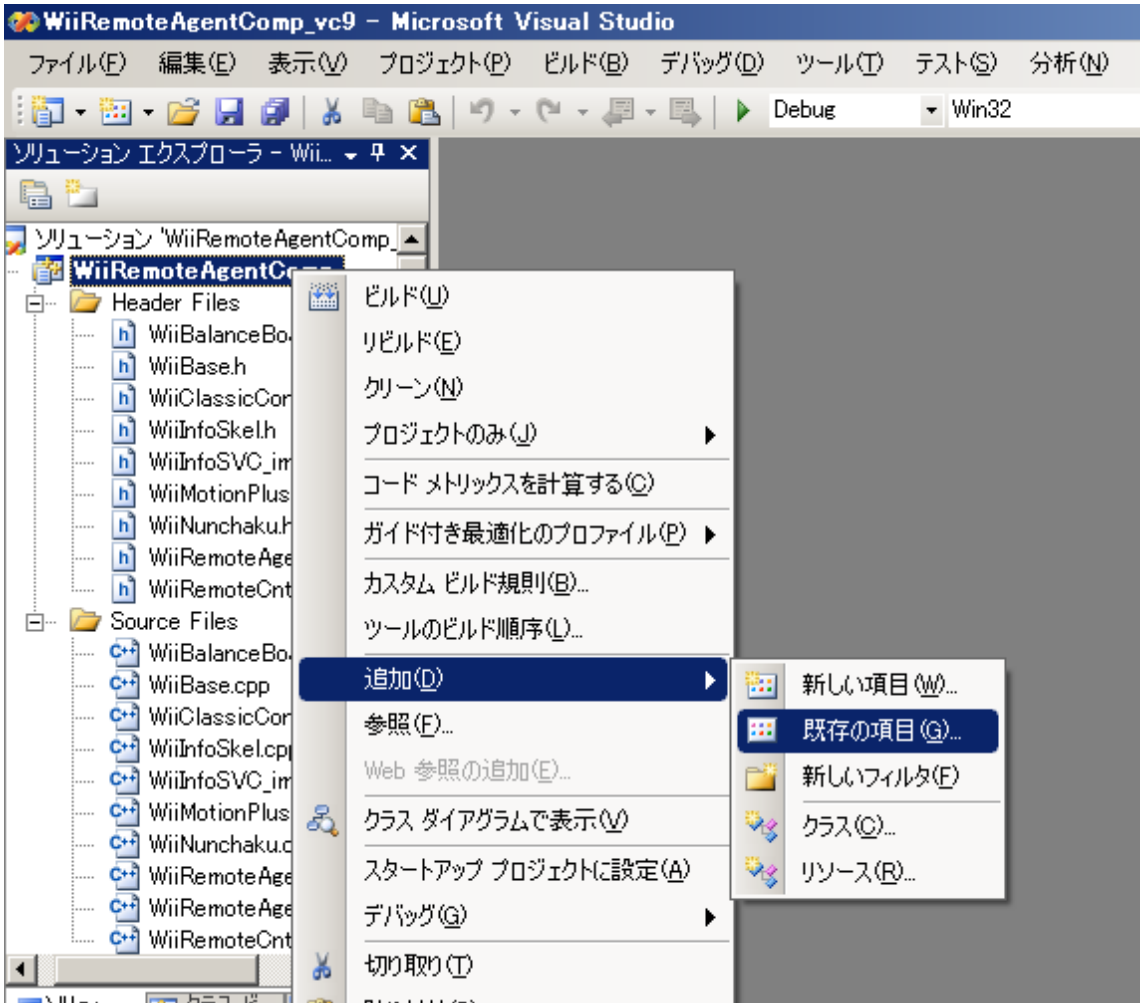
4.1.1. WiiRemoteAgent のビルド方法

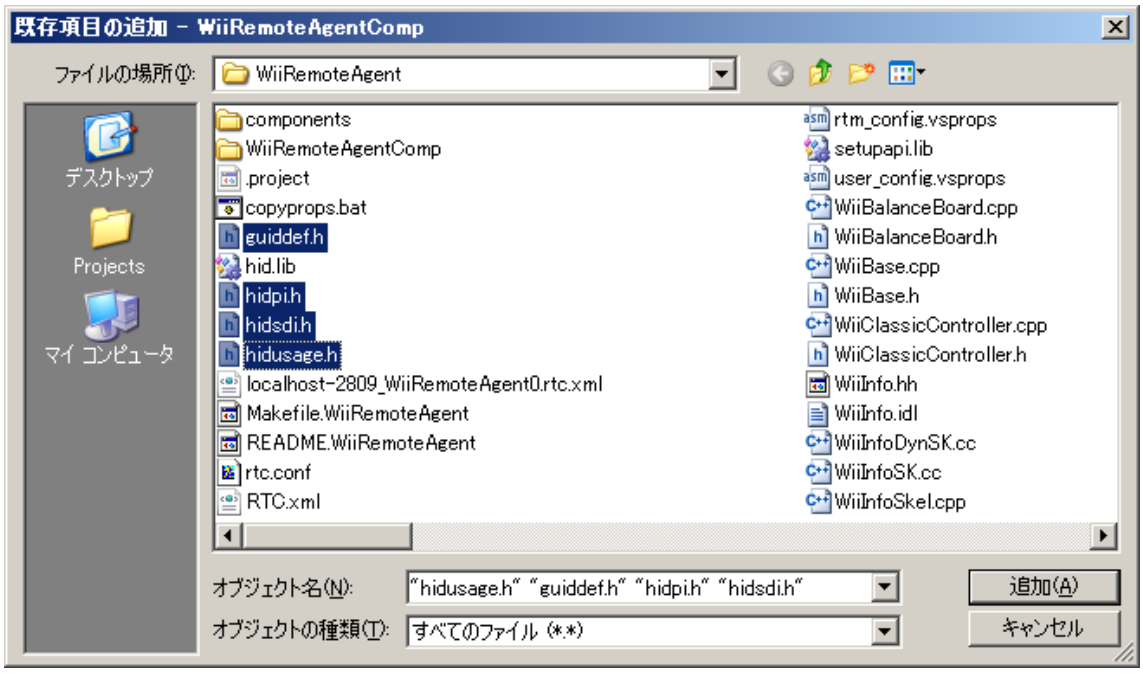
(1) 本プロジェクトを適当な場所に解凍します.

(2) (1)で解凍した“WiiRemoteComponents”のうち“WiiRemoteAgent”のプロジェクトファイルを開きます.「3.2.ダウンロード頂くファイル」で示した 6 つのファイルをプロジェクトファイルにコピーします.



(3) Visual Studio 2008 で“WiiRemoteAgent”のフォルダ内にある“WiiRemoteAgent_vc9.sln”を開きます．その後，(2)で追加したヘッダファイルをプロジェクトに追加します.


WiiRemoteAgentComp_vc9 - Microsoft Visual Studio
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) ツール(T) テスト(S) 分析(N)
Debug
Win32
ソリューション エクスプローラ - Wii...
ソリューション 'WiiRemoteAgentComp_
WiiRemoteAgentC
Header Files
WiiBalanceBo
WiiBase.h
WiiClassicCor
WiiInfoSkel.h
WiiInfoSVC_im
WiiMotionPlus
WiiNunchaku.h
WiiRemoteAge
WiiRemoteCnt
Source Files
WiiBalanceBo
WiiBase.cpp
WiiClassicCor
WiiInfoSkel.cp
WiiInfoSVC_im
WiiMotionPlus
WiiNunchaku.c
WiiRemoteAge
WiiRemoteAge
WiiRemoteCnt
ビルド(U)
リビルド(E)
クリーン(N)
プロジェクトのみ(J)
コード メトリックスを計算する(C)
ガイド付き最適化のプロファイル(P)
カスタム ビルド規則(B)...
ツールのビルド順序(L)...
追加(D)
参照(F)...
Web 参照の追加(E)...
クラス ダイアグラムで表示(V)
スタートアップ プロジェクトに設定(A)
デバッグ(G)
切り取り(T)
新しい項目(W)...
既存の項目(G)...
新しいフィルタ(F)
クラス(C)...
リソース(R)...


既存項目の追加 - WiiRemoteAgentComp
ファイルの場所(I):
WiiRemoteAgent
デスクトップ
Projects
マイ コンピュータ
components
WiiRemoteAgentComp
.project
copyprops.bat
guiddef.h
hid.lib
hidpi.h
hidsdi.h
hidusage.h
localhost-2809_WiiRemoteAgent0.rtc.xml
Makefile.WiiRemoteAgent
README.WiiRemoteAgent
rtc.conf
RTC.xml
rtm_config.vsprops
setupapi.lib
user_config.vsprops
WiiBalanceBoard.cpp
WiiBalanceBoard.h
WiiBase.cpp
WiiBase.h
WiiClassicController.cpp
WiiClassicController.h
WiiInfo.hh
WiiInfo.idl
WiiInfoDynSK.cc
WiiInfoSK.cc
WiiInfoSkel.cpp
オブジェクト名(N):
"hidusage.h" "guiddef.h" "hidpi.h" "hidsdi.h"
オブジェクトの種類(T):
すべてのファイル (*.*)
追加(A)
キャンセル

(4) ビルドを行います．(正常にビルドが行われない場合，(2)で追加したファイルに間違いがないことを確認してください．間違いがない場合，利用している WDK のバージョンを明記の上，連絡先メールアドレスまで連絡をお願いします．)

4.1.2. 各種 Translator のビルド方法

本説明では，WiiRemoteTranslator を利用する場合を想定し説明をします．他の Translator も基本的に同じ手順によりビルドを行うことが可能です．

(1) 4.1.1.の(1)で解凍した“WiiRemoteComponents”のうち“WiiRemoteTranslator”のフォルダを開きます．

(2) Visual Studio 2008 で“WiiRemoteTranslator”のフォルダ内にある“WiiRemoteTranslator_vc9.sln”を開きます．

(3) ビルドを行います．(“WiiRemoteAgent”と異なりライブラリなどを利用しておりません．)

4.2. WiiRemoteComponents の利用方法

(1) Wii Remote のボタン“1”，“2”を同時押しまたは電池の横にある赤いボタンを押し SYNC をします．(BalanceBoard は電池の横にある赤い SYNC ボタンを押してください．)

(2) Bluetooth の付属ソフトを利用し WiiRemote と PC を接続してください．

(3) WiiRemoteAgent を実行してください．

(4) 必要な Translator のコンポーネントを実行してください．

(5) WiiRemoteAgent のコンフィギュレーションを設定し，利用する拡張コントローラを設定してください．

(6) コンフィギュレーションの設定に対応した Translator を起動してください．

(7) 起動した Translator で動作確認を行う場合，Translator の Configuration の debug を“1”に設定してください．これにより CUI に状態表示が行われます．(各コンポーネントの接続は 0 を参照してください．)

※注意点:WiiRemoteAgent で複数の WiiRemote を接続することはできません．また，RT-Component と接続する WiiRemote は Bluetooth の付属ソフトに登録されている最初の WiiRemote となります．WiiRemote を交換する場合，付属ソフトに登録されている WiiRemote を削除し，使用する WiiRemote を再登録してください．

4.3. WiiRemoteCommponents の動作確認

4.2 にて起動したコンポーネントの接続を行います．WiiRemoteComponents で開発した Translator はインポートに WiiRemoteAgent からの出力を接続し，サービスポート(コンシューマ)は WiiRemoteAgent のサービスポート(プロバイダー)に接続する形になります．接続した図を図 4-1 に示します．接続完了後，Translator の Configuration である”debug”を”1”に設定し，アクティベイトを行い Translator の CUI を確認してください．サンプルとして ClassicController の Translator を図 4-2 に示します．

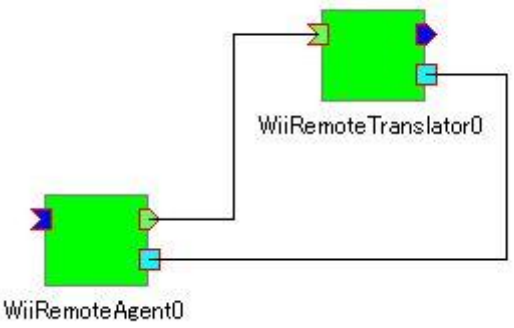


図 4-1 動作確認システム(WiiRemoteTranslator を利用した場合)

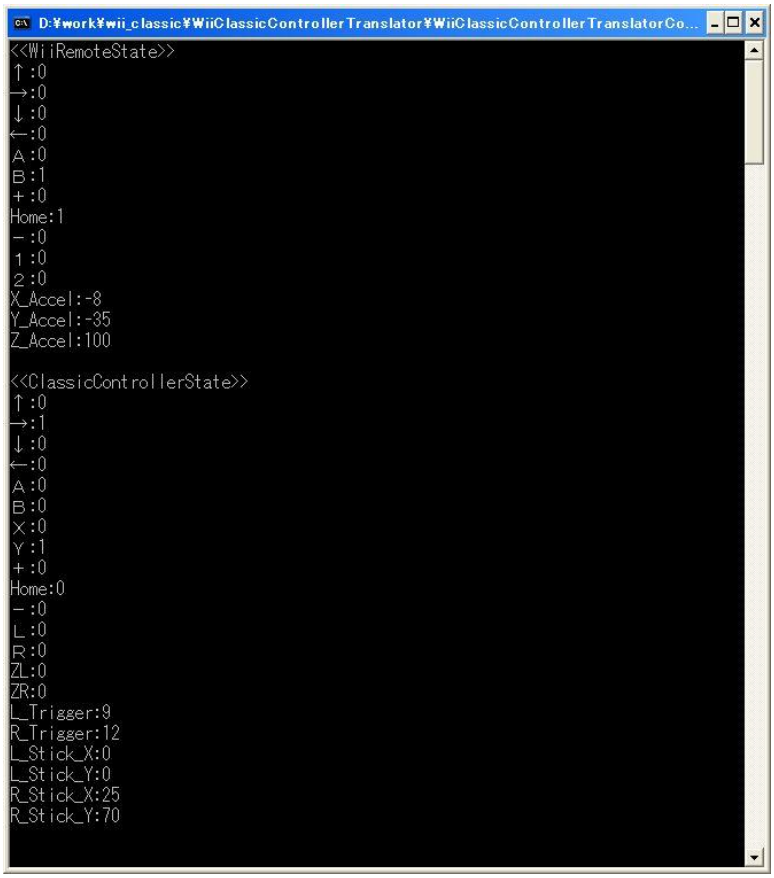


図 4-2 WiiClassicControllerTranslator のデバッグモード

5. 動作環境・開発環境

- Windows XP (Visual Studio 2008 でコンパイル)
- RT-Middlware (OpenRTM-aist-1.0.0Release)
- WiiRemote，BalanceBoard，ClassicController，ClassicControllerPro，Nunchaku，WiiMotionPlus
- WDK(Ver.6001.18000)

6. ライセンス

マイクロソフト株式会社より提供されている Windows 用のデバイスドライバ開発キット （WDK) のソースコード及びライブラリを除き, WiiRemoteComponents の著作権は, 芝浦工業大学水川研究室に帰属します.

API("hidpi"等)のソースコード及びライブラリ・DLL の著作権は, 開発元のマイクロソフト株式会社にあります.

芝浦工業大学水川研究室が著作権を持つ RT コンポーネントは, 非営利目的での使用及び改変自由です.

商用利用の場合は別途ご相談下さい.

なお，本作品を使用して発生した，いかなる損害についても責任を負いません.

7. 連絡先

芝浦工業大学　水川研究室
〒135-8548
　東京都江東区豊洲 3-7-5 芝浦工業大学　研究棟 11Q32　水川研究室
TEL：03-5859-8209　　FAX：03-5859-8201
E-Mail：shibaura.hri.goiken<at>gmail.com
指導教員：水川　真
作成者：田中　基雅
　　鷹栖　尭大
　　藤田　恒彦
URL：http://www.hri.ee.shibaura-it.ac.jp/

参考文献

[1] 任天堂株式会社「Wii」
http://www.nintendo.co.jp/wii/

[2] “WiiRemote プログラミング”
白井 暁彦ほか(オーム社)

[3] Wii Brew
http://wiibrew.org/wiki/Main_Page

[4] Wii Remote wiki
http://wiki.wiimoteproject.com/Main_Page

[5] CWiid
http://abstrakraft.org/cwiid/

[6] kako homepage
http://www.kako.com/index.html

8. 付録

- WiiRemoteAgent の機能拡張方法

WiiRemoteAgent は，デザインパターンのストラテジーを適用しています．各種クラスの実装は図 8-1 に示す通りとなっています．そのため，各種クラスはインタフェースと共通で利用するメソッドを実装した CWiiRemoteCntl クラスを継承する形で実装を行っています．新たに機能拡張を行う場合，CWiiRemoteCntl クラスを継承しインタフェースの実装を行ってください．これにより，容易に機能拡張を行うことが可能となっております．

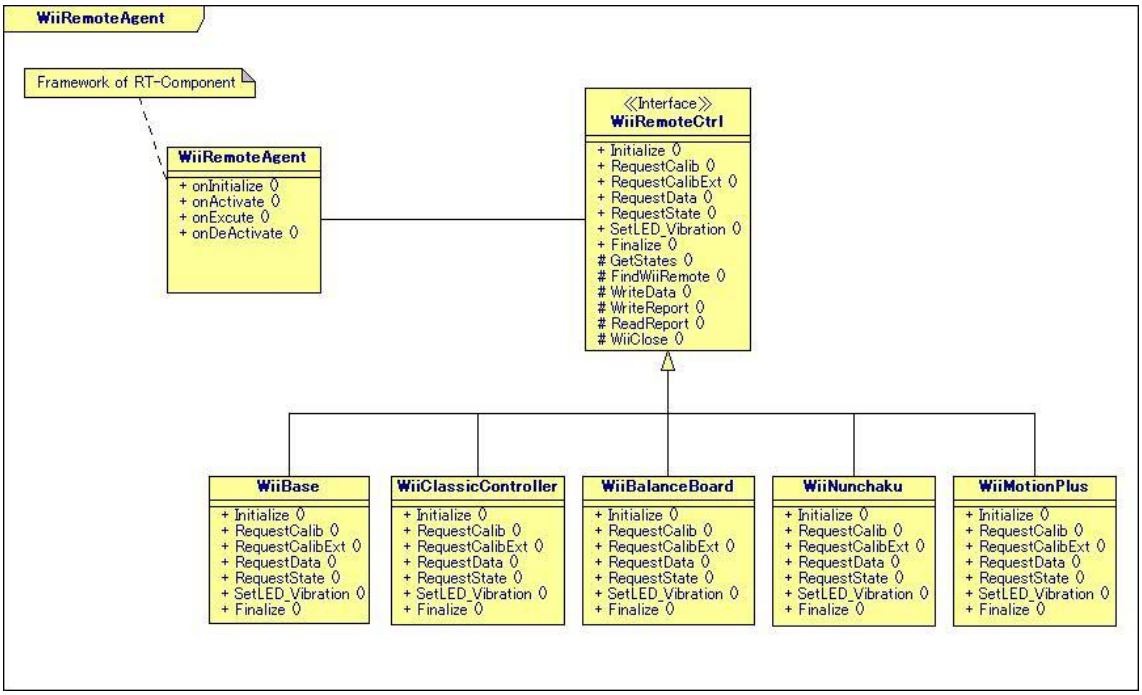


図 8-1 WiiRemoteAgent のクラス図

拡張コントローラのクラスを実装後，インスタンスを生成するため OnActivate に赤字で示した部分を追加してください．

≪WiiRemoteAgent.cpp　173 行目付近(OnActivate 処理内)≫

```
else if(m_OtherExt == true)
{
                wiimote = new “作成クラスのコンストラクタ”;
                //現在未対応．エラー状態に遷移させる
                //std::cout << "Unsupported Option" << std::endl;
                //return RTC::RTC_ERROR;
}
```

以上により，機能拡張を行うことが可能となります．