

GameController.framework

を調べてみました

<※2014.12.11>

GameController.frameworkの背景

- Apple社は、MFiアクセサリメーカーに対して、ゲームコントローラーの規格を策定しています。



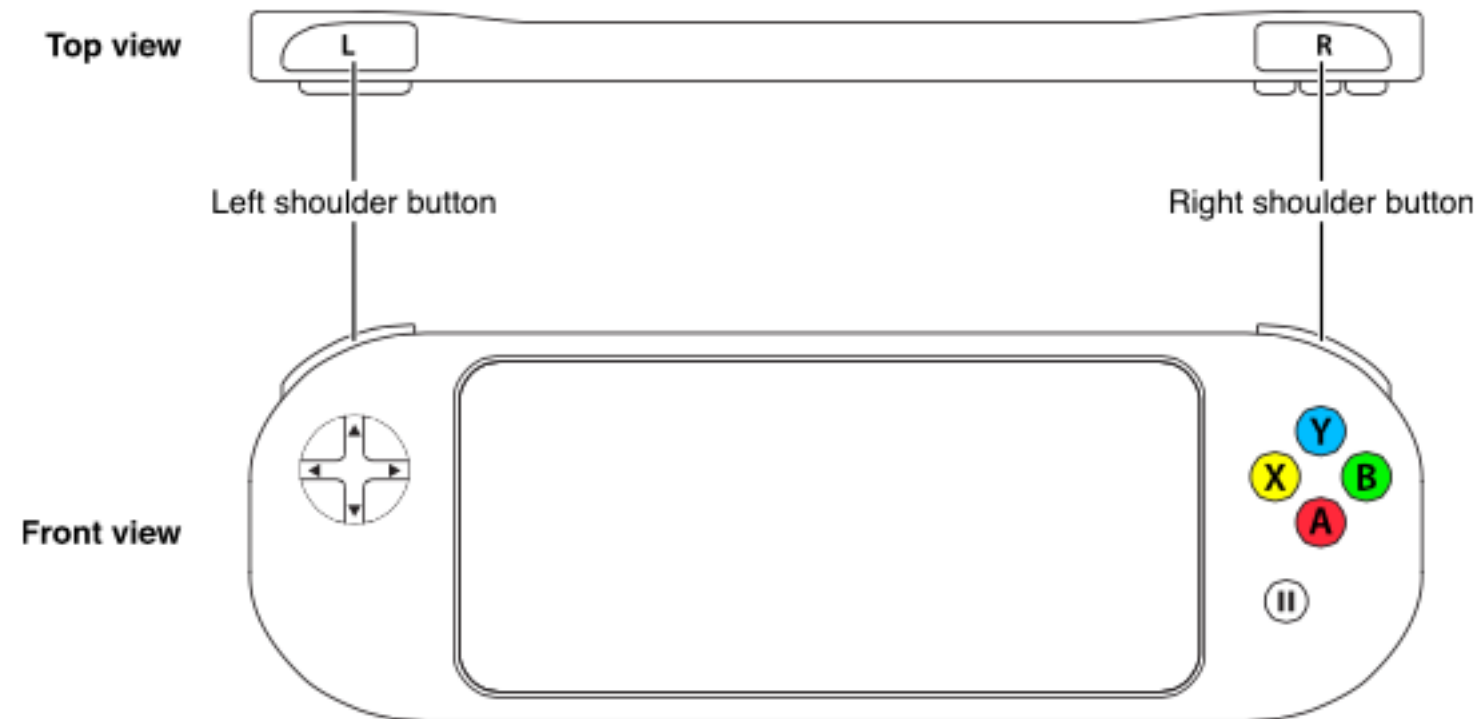
※Made for iPod/iPhone/iPad
のロゴマーク

- iOS7以降のアプリケーションプログラムは、**GameController.framework**を使用して、MFi規格に対応しているゲームコントローラーからの入力を読み取ることができます。

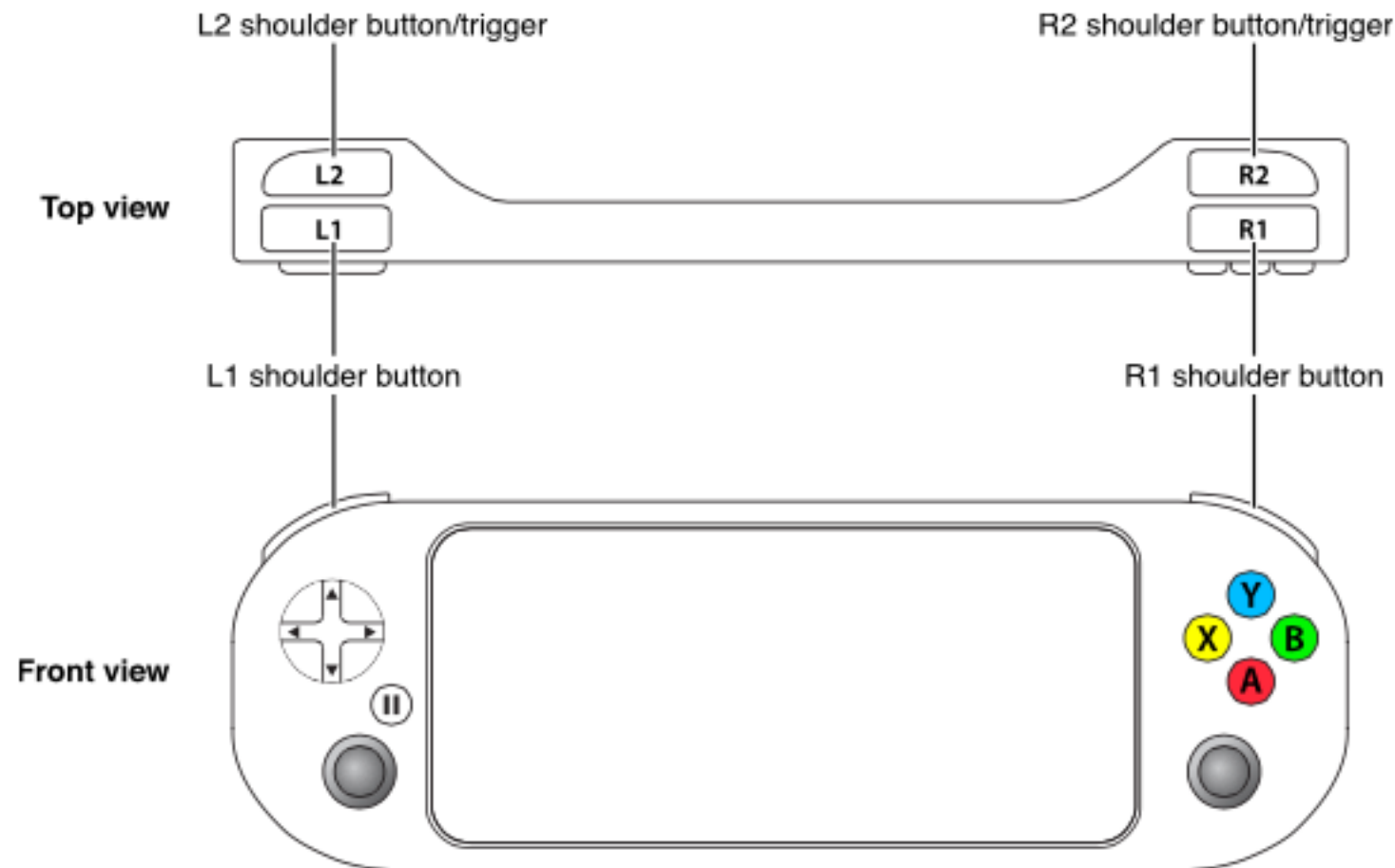
iOSゲームコントローラとは

1. iOS組み込み標準コントローラ
2. iOS組み込み拡張型コントローラ
3. スタンドアロン拡張型コントローラ

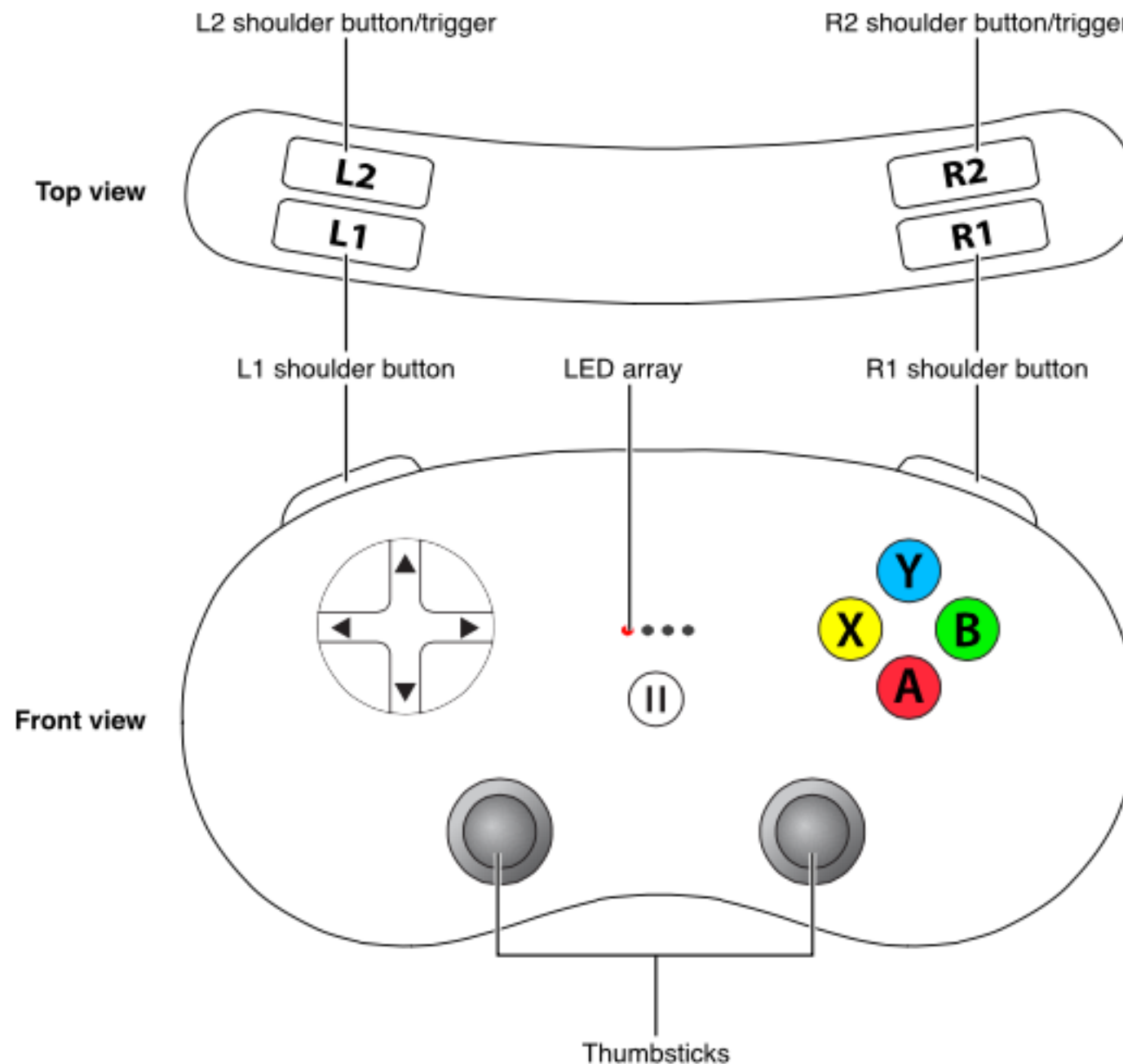
iOS組み込み標準型コントローラ



iOS組み込み拡張型コントローラ



スタンドアロン拡張型コントローラ



※形状は左図に限りません

スタンドアロン拡張型コントローラを 初期するには

1. Link Binary With LibrariesでGameController.frameworkを設定(XcodeプロジェクトのBuild Phasesで設定)
2. `#import <GameController/GameController.h>`
3. `NSNotificationCenter`を使用して
`GCControllerDidConnectNotification`を観測 はまりポイント有
4. notificationから`GCController(NSObject)`オブジェクトを取得
5. GCControllerオブジェクトから`GCExtendedGamepad`オブジェクトを取得

GCExtendedGamepad

- `@interface GCExtendedGamePad : NSObject`
- `@property (readonly) GCControllerDirectionPad *dpad`
- `@property (readonly) GCControllerButtonInput *buttonA`
- `@property (readonly) GCControllerButtonInput *buttonB`
- `@property (readonly) GCControllerButtonInput *buttonX`
- `@property (readonly) GCControllerButtonInput *buttonY`
- `@property (readonly) GCControllerDirectionPad *leftThumbstick`
- `@property (readonly) GCControllerDirectionPad *rightThumbstick`
- `@property (readonly) GCControllerButtonInput *leftShoulder`
- `@property (readonly) GCControllerButtonInput *rightShoulder`
- `@property (readonly) GCControllerButtonInput *leftTrigger`
- `@property (readonly) GCControllerButtonInput *rightTrigger`

Aボタン押下を検出

```
gamePad.buttonA.valueChangedHandler =  
^(GCControllerButtonInput *button, float value, BOOL  
pressed) {  
    NSLog(@"buttonAに関するハンドラ: %f, %d", value,  
pressed);  
};
```

```
2014-12-11 01:03:33.226 iOSApp[14109:2114140] buttonAに関するハンドラ: 0.176758, 1  
2014-12-11 01:03:33.241 iOSApp[14109:2114140] buttonAに関するハンドラ: 0.305664, 1  
2014-12-11 01:03:33.257 iOSApp[14109:2114140] buttonAに関するハンドラ: 0.474609, 1  
2014-12-11 01:03:33.271 iOSApp[14109:2114140] buttonAに関するハンドラ: 0.451172, 1  
2014-12-11 01:03:33.287 iOSApp[14109:2114140] buttonAに関するハンドラ: 0.286133, 1  
2014-12-11 01:03:33.317 iOSApp[14109:2114140] buttonAに関するハンドラ: 0.035156, 1  
2014-12-11 01:03:33.331 iOSApp[14109:2114140] buttonAに関するハンドラ: 0.000000, 0
```

R1ボタン押下を検出

```
gamePad.leftShoulder.valueChangedHandler =  
^(GCControllerButtonInput* button, float value, BOOL  
pressed) {  
    NSLog(@"R1に関するハンドラ: %f, %d", value,  
pressed);  
};
```

```
2014-12-11 01:06:33.966 iOSApp[14109:2114140] R1に関するハンドラ: 0.254883, 1  
2014-12-11 01:06:33.967 iOSApp[14109:2114140] R1に関するハンドラ: 0.537109, 1  
2014-12-11 01:06:33.978 iOSApp[14109:2114140] R1に関するハンドラ: 0.639648, 1  
2014-12-11 01:06:33.993 iOSApp[14109:2114140] R1に関するハンドラ: 0.655273, 1  
2014-12-11 01:06:34.027 iOSApp[14109:2114140] R1に関するハンドラ: 0.639648, 1  
2014-12-11 01:06:34.028 iOSApp[14109:2114140] R1に関するハンドラ: 0.431641, 1  
2014-12-11 01:06:34.059 iOSApp[14109:2114140] R1に関するハンドラ: 0.074219, 1  
2014-12-11 01:06:34.060 iOSApp[14109:2114140] R1に関するハンドラ: 0.000000, 0
```

十字キーさえも。

```
gamePad.dpad.valueChangedHandler =  
^(GCControllerDirectionPad *dpad, float xValue, float  
yValue) {  
    NSLog(@"十字キーに関するハンドラ: %f, %f", xValue,  
yValue);  
};
```

```
2014-12-11 01:08:58.149 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.101562, 0.000000  
2014-12-11 01:08:58.155 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.435547, 0.000000  
2014-12-11 01:08:58.183 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.505859, 0.000000  
2014-12-11 01:08:58.187 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.404297, 0.000000  
2014-12-11 01:08:58.219 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.097656, 0.000000  
2014-12-11 01:08:58.220 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.000000, 0.000000  
2014-12-11 01:08:58.819 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.000000, -0.258789  
2014-12-11 01:08:58.820 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.000000, -0.340820  
2014-12-11 01:08:58.853 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.000000, -0.325195  
2014-12-11 01:08:58.856 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.000000, -0.105469  
2014-12-11 01:08:58.864 iOSSApp[14109:2114140] 十字キーに関するハンドラ: -0.000000, -0.000000
```

右スティックぐりぐり

```
gamePad.rightThumbstick.valueChangedHandler =  
^(GCControllerDirectionPad *dpad, float xValue, float  
yValue) {  
    NSLog(@"右スティック処理: %f, %f", xValue,  
yValue);  
};
```

```
2014-12-11 01:12:24.278 iOSSApp[14109:2114140] 右スティック処理: 0.197266, 0.000000  
2014-12-11 01:12:24.279 iOSSApp[14109:2114140] 右スティック処理: 0.716797, 0.000000  
2014-12-11 01:12:24.282 iOSSApp[14109:2114140] 右スティック処理: 1.000000, 0.000000  
2014-12-11 01:12:24.777 iOSSApp[14109:2114140] 右スティック処理: 0.000000, 0.000000  
2014-12-11 01:12:27.777 iOSSApp[14109:2114140] 右スティック処理: -0.071289, 0.000000  
2014-12-11 01:12:27.778 iOSSApp[14109:2114140] 右スティック処理: -0.071289, 0.125977  
2014-12-11 01:12:27.778 iOSSApp[14109:2114140] 右スティック処理: -0.362305, 0.125977  
2014-12-11 01:12:27.782 iOSSApp[14109:2114140] 右スティック処理: -0.362305, 0.457031  
2014-12-11 01:12:27.782 iOSSApp[14109:2114140] 右スティック処理: -0.748047, 0.457031  
2014-12-11 01:12:28.277 iOSSApp[14109:2114140] 右スティック処理: -0.748047, 0.000000  
2014-12-11 01:12:28.277 iOSSApp[14109:2114140] 右スティック処理: 0.000000, 0.000000
```

昔、一時期こんな処理を書いていた

```
public void mainloop(){  
    Runtime run = Runtime.getRuntime();  
    while(true){  
        if(m_vsync == false) continue;  
        m_vsync = false;  
  
        //キー状態の取得  
        m_key_state = getKeypadState();  
        if(((m_key_state & (1 << Display.KEY_UP))!=0)){  
            if(m_cy > 0) m_cy-= m_vector;  
        }  
  
        if(((m_key_state & (1 << Display.KEY_DOWN))!=0)){  
            if(m_cy < 240) m_cy+= m_vector;  
        }  
  
        if(((m_key_state & (1 << Display.KEY_RIGHT))!=0)){  
            if(m_cx < 320) m_cx+= m_vector;  
        }  
  
        if(((m_key_state & (1 << Display.KEY_LEFT))!=0)){  
            if(m_cx > 0) m_cx-= m_vector;  
        }  
    }  
}
```

Blockでのハンドラ処理を見て、新しいなあと思いました

調べた感想

- ・ GameController.frameworkは素晴らしいフレームワーク。
- ・ 容易に既存アプリに組み込めた。
- ・ アイディアは、溢れる。

参考文献

- ・ Apple, Game コントローラープログラミングガイド
- ・ Game Controller Framework Reference

[https://developer.apple.com/library/mac/documentation/
GameController/Reference/GameController_RefColl/
index.html#classes](https://developer.apple.com/library/mac/documentation/GameController/Reference/GameController_RefColl/index.html#classes)