



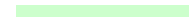
# DesignWaveの付録基盤を使ったWiiコントローラもどきの開発

---

筑波大学 情報学群 情報科学類 2年

柴田 泰晴

2008年 5月2日

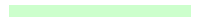




# 自己紹介(1)

---

- 名前
  - 柴田 泰晴 (yasuharu)
- 出身
  - 愛知県豊橋市
- 趣味
  - プログラミング、読書、昼寝

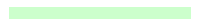




# 自己紹介(2)

---


- 主に使用する言語
  - しっかりとしたものを書くときには
    - C言語/C++
  - さっとものを作るときには
    - C#/Java
  - Webアプリケーションを作るときには
    - PHP



# 自己紹介(3)


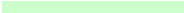
---

- 過去一ヶ月にやっていたこと
  - プロジェクト一覧などを管理するためのWebアプリケーションの開発
  - 正規表現を用いたWikiソフトの開発
  - BNF記法を用いた数式処理ソフトの開発
  - Linuxのソースコードを解説書に目を通しながら読む



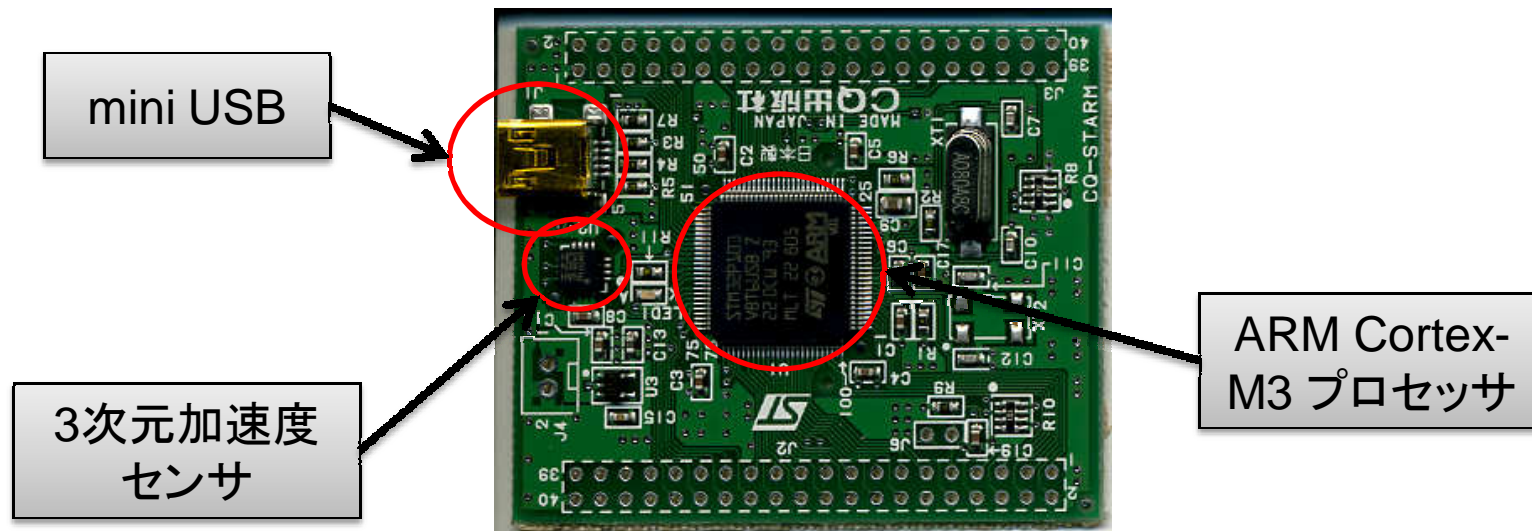
# 今回発表するものは？

---

- DesignWaveの付録基盤を使って、Wiiのコントローラのような操作をできるようにする
    - 傾けたり、振ったりして操作する
  - それを開発する手順について説明をする
  - 対象としては、マイコンを触るきっかけがほしいような人
  - 環境は、Windowsを想定
- 
- 

# DesignWaveの付録基盤とは？


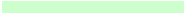
- CQ出版社から出版されているDesignWaveという雑誌の2008年5月号の付録基盤
- 特徴としては、





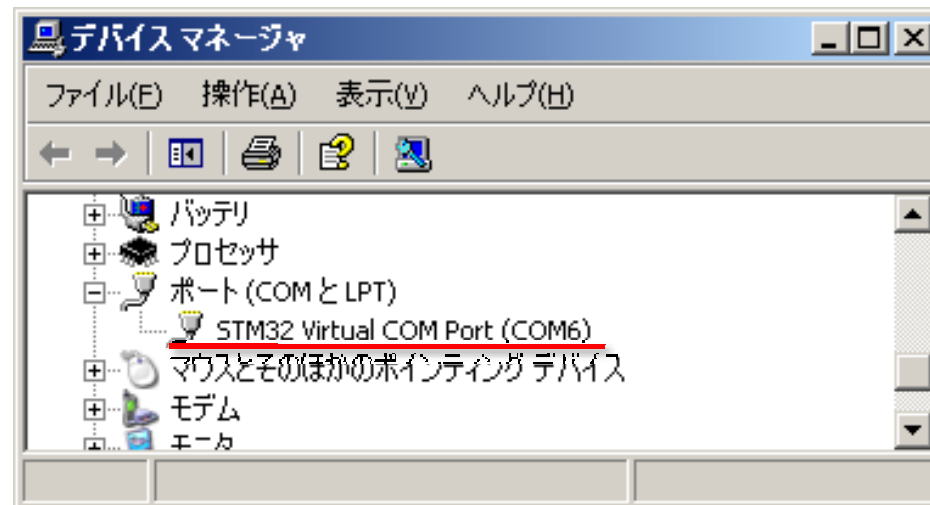
# コントローラの開発の方針

---

- マイコンから、3次元加速度センサの値を取得して、その値を見て動きを判断する
    - 受信した値は、X軸、Y軸、Z軸の加速度の値が入っている
  - このプレゼンでは、次のように説明をする
    - マイコンから値を読み込む方法の説明
    - 実際にプログラムで使う関数の説明
- 
- 

# 接続の確認

- 物理的な接続はUSB
- デバイスマネージャで確認すると、仮想COMポートと接続している
- つまり、COMポートを制御して、通信すればよい





# ところで、COMポートって何？

- COMポート (Communication Port) とは、シリアルインタフェースの総称
- RS-232CのポートのことをCOMポートと言うことがある
- また、シリアルポートと言うこともある

COMポート

RS-232Cのポート

シリアルポート

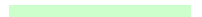




# COMポートの制御

---


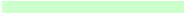
- COMポートの制御にはいくつか方法がある
  - Win32 API
  - Microsoft Communication Control
  - .NET Framewrok
  - 直接操作する
- 最後の方法を除いて、データを送受信するのは簡単





# 受信したデータの中身は？

---

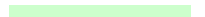
- それじゃあ、データを受信すれば3次元加速度センサの値が見られるんですね
    - もちろん、そんなわけではないです
  - マイコンもコンピュータなので、出力がどうなっているかは中のプログラムで決定される
  - 今回に関しては、雑誌中で「\*」を入力すると3次元加速度センサの値が出力されることがわかっているので、これを利用する
- 
- 



# もし、自分で作るとしたら？

---

- 今回は、雑誌の付録なので最初からマイコンにプログラムが書き込まれているが、普通は書き込まれていない
- そういうときは、マイコン用のプログラムを自分で書く必要がある

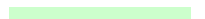




# 実際に実装しよう

---

- 昨年のWORD記事にあった、シューティングゲーム開発講座をもとにする
- Wiiのコントローラのような感じで操作をできるようにする
- COMポートの制御は、Win32 APIを用いて実装する



# 接続する

- CreateFile関数を使用して、COMポートと接続する

COMポートの番号を指定する「COMn」とも指定できるが、 $n \geq 10$ の時には「¥¥.¥COMn」と記述しないと接続に失敗する

```
HANDLE hCom = CreateFile(  
    “¥¥¥¥.¥¥COM6”,  
    GENERIC_READ | GENERIC_WRITE,  
    0,  
    NULL,  
    OPEN_EXISTING,  
    FILE_ATTRIBUTE_NORMAL,  
    NULL  
);
```

# データを送信する

---

- WriteFile関数を使用してデータを送信する

```
WriteFile(hCom, &Data, Length, &WriteByte, NULL);
```

- 今回は、接続が完了した段階で「\*」を送信する

# データを受信する(1)

- ClearCommError関数を使って、COMポートのバッファに溜まっているデータのサイズを調べる
- COMSTAT構造体のcbInQue変数にバッファに溜まっているデータのサイズが入っている

```
ClearCommError(hCom, &Errors, &ComStat);  
Count = ComStat.cbInQue;
```




# データを受信する(2)

- ReadFile関数を使ってデータを受信する

```
DWORD Read;  
ReadFile(hCom, Data, Count, &Read, NULL);
```

- 今回は、加速度センサの値が「(X軸,Y軸,Z軸)」の形式で送られてくるので、それを受信する

# 終了する



---

- CloseFile関数を使ってポートを閉じる


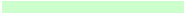
```
CloseHandle(hCom);
```





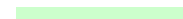
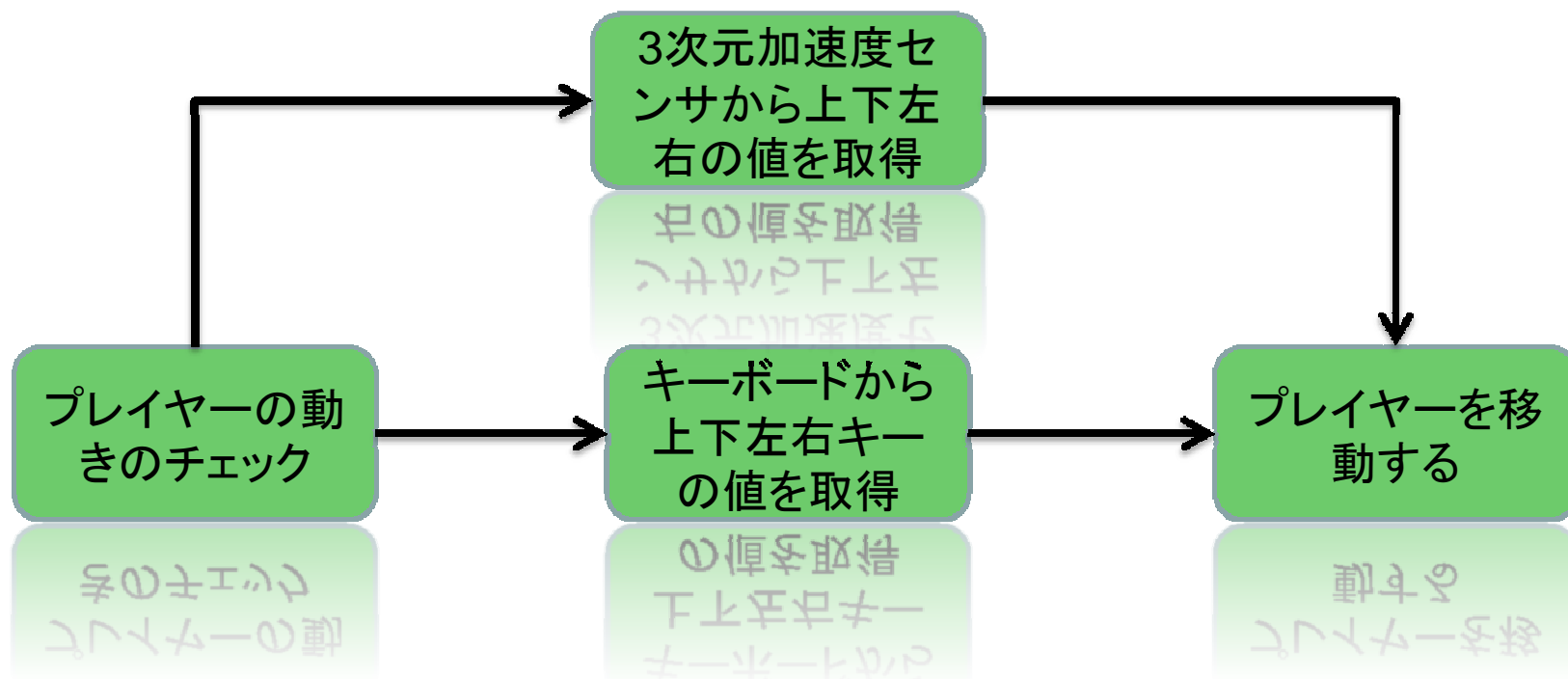
# その他の関数


---

- sscanf関数
    - 書式に従ってデータを取得する
  - SetCommTimeouts関数
    - 同期通信の場合、ずっと処理を待ってしまい、プログラムが止まってしまう可能性があるため、設定が必要
  - GetCommState / SetCommState関数
    - 通信速度の設定などを行う
- 
- 




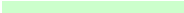
# 実装の方針





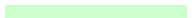
# 実際に動かしてみる

---

- 細かいコードの説明は省略して、今まで説明をした関数を元に実装を試してみた
  - 実装の方針としては、受信した加速度のデータを各軸ごとに見てやって、値の変化を見てコントローラの操作を行っている
- 
- 



# デモ





# まとめ

---

- 結局のところ何が言いたかったのか？
  - COMポートを使ったマイコン制御の方法を知ってもらう
  - マイコンに興味をもってもらって、実際に使えるようになってもらう
  - 実際に触ってみて、興味を持ってもらい、わからないところは調べてほしい

