

# できる！ BIT BANGモード

**Y.TOMOHARA**

# 自己紹介

## TOMO

- @tomozh
- <http://ore-kb.net>

福岡市内の会社でソフト開発やってます

- 主にVisual C++, H8, 最近Android

趣味でプログラミングと電子工作やってます

- マイコンは、PIC→AVR
- プログラミングは、MSX BASIC→ASM(Z80)→C→C++

# アジェンダ

PCインタフェースの昔と今

FTシリーズについて

Bit Bangモードとは何か

非同期Bit Bangモード

- LED点滅
- SW入力

同期Bit Bangモード

- I2Cで温度センサ

はまりどころ

参考文献

# PCインタフェースのむかし

プリンタポート、シリアルポートを使って比較的手軽に制御ができていた

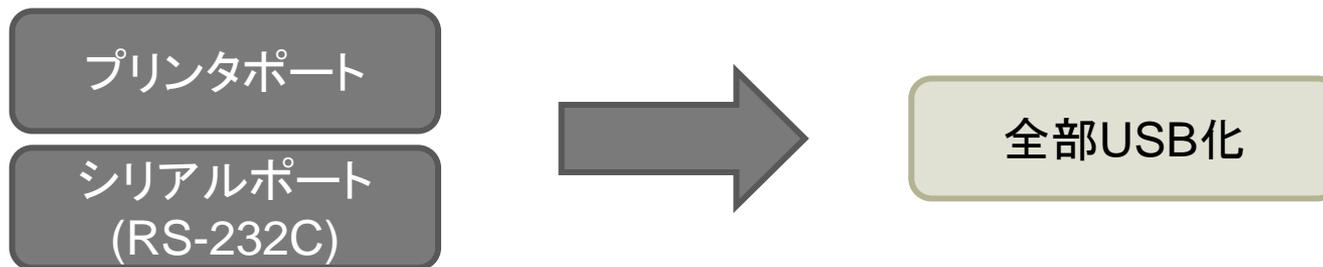
プリンタポート

シリアルポート  
(RS-232C)



# PCインタフェースのいま

昔出来ていたことが出来なくなった！



- デバイスドライバ
- USBプロトコルの実装
- USBベンダIDの取得
- Vistaから、WHQL取得してないドライバはインストール時にやたら派手な警告が出るようになった

アマチュアに優しくない昨今のPC事情に絶望した！

そこに救世主が現れた！



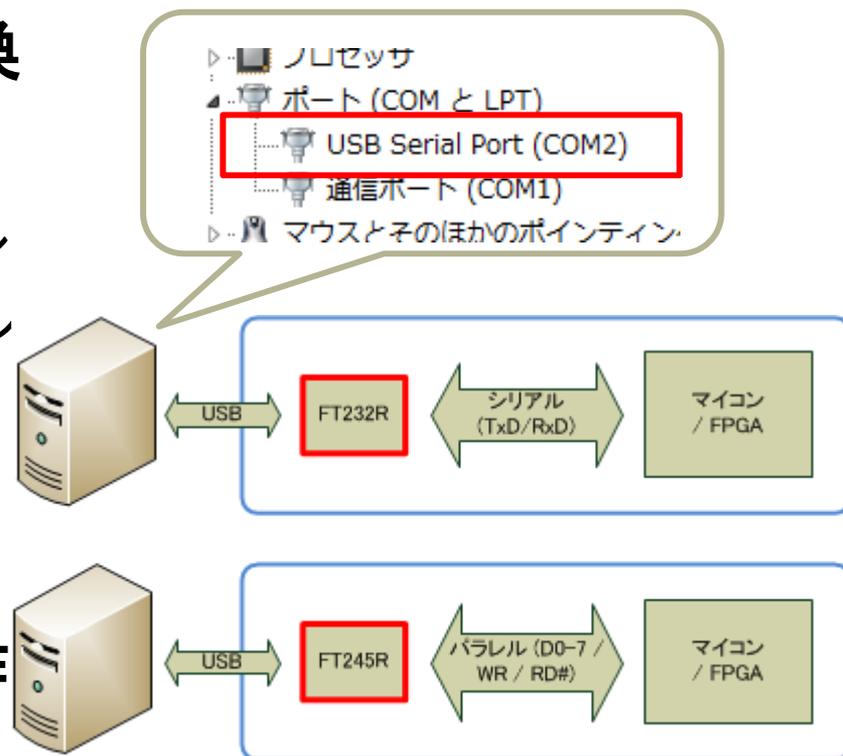
FTDIのFTシリーズ

# FTシリーズについて

## FTDI社のUSB-シリアル/USB-パラレル変換IC

- FT232: USB-シリアル
- FT245: USB-パラレル
- FT2232: 両方

PCからはシリアルポートに見える(専用のドライバで直接操作することも可能)



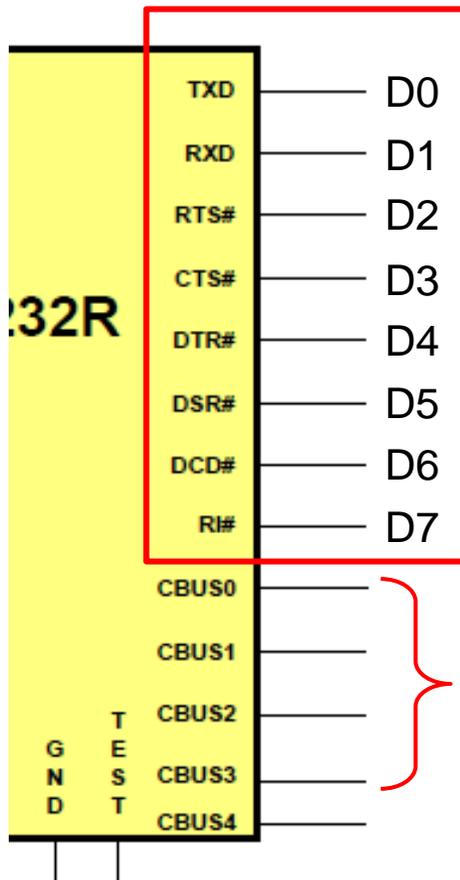
# FTシリーズ比較表

	FT232/245B	FT232/245R	FT2232D/C/L	FT232/2232H
EEPROM	外付け	内蔵	外付け	外付け
非同期Bit Bang	○	○	○	○
同期Bit Bang	—	○	○	○
CBUS BitBang	—	FT232のみ	○	○
USB I/F	USB2.0 FS	USB2.0 FS	USB2.0 FS	USB2.0 HS
OSC	外付け	内蔵	外付け	外付け
単価 @秋月電子	—	¥350 (FT232RL)	¥500 (FT2232D)	—

↑  
オススメ



# BIT BANGモードとはなに か？



外部接続用の端子が  
関数一発で  
汎用の入出力端子になる！！

FT232/2232専用 (Bシリーズ, FT245R不可)  
FTProgでEEPROMに設定すればここも使える！  
(しかし上のやつとは排他)

# 2種類のモード

Bit Bangには2種類のモードがある！

## 非同期Bit Bangモード

- 最初(FT232B)からある

## 同期Bit Bangモード

- FT232Rで追加されてやっと使い物ちょっといい感じになった

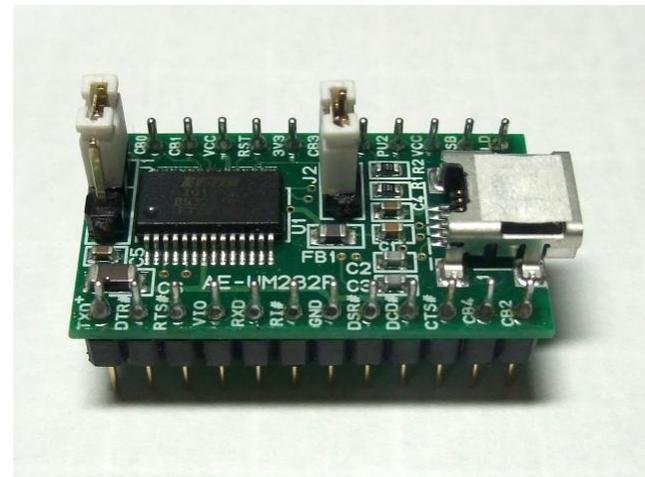
# BIT BANGモードを使うには (ハード編)

## Bit Bang対応のFTDIチップ

- FT232RLがおすすめ
- コネクタ付きのモジュールがもっとおすすめ



FT232RL



秋月電子 AE-UM232R  
(¥950)

# BIT BANGモードを使うには (ソフト編)

## VC++ (Visual Studio)の開発環境

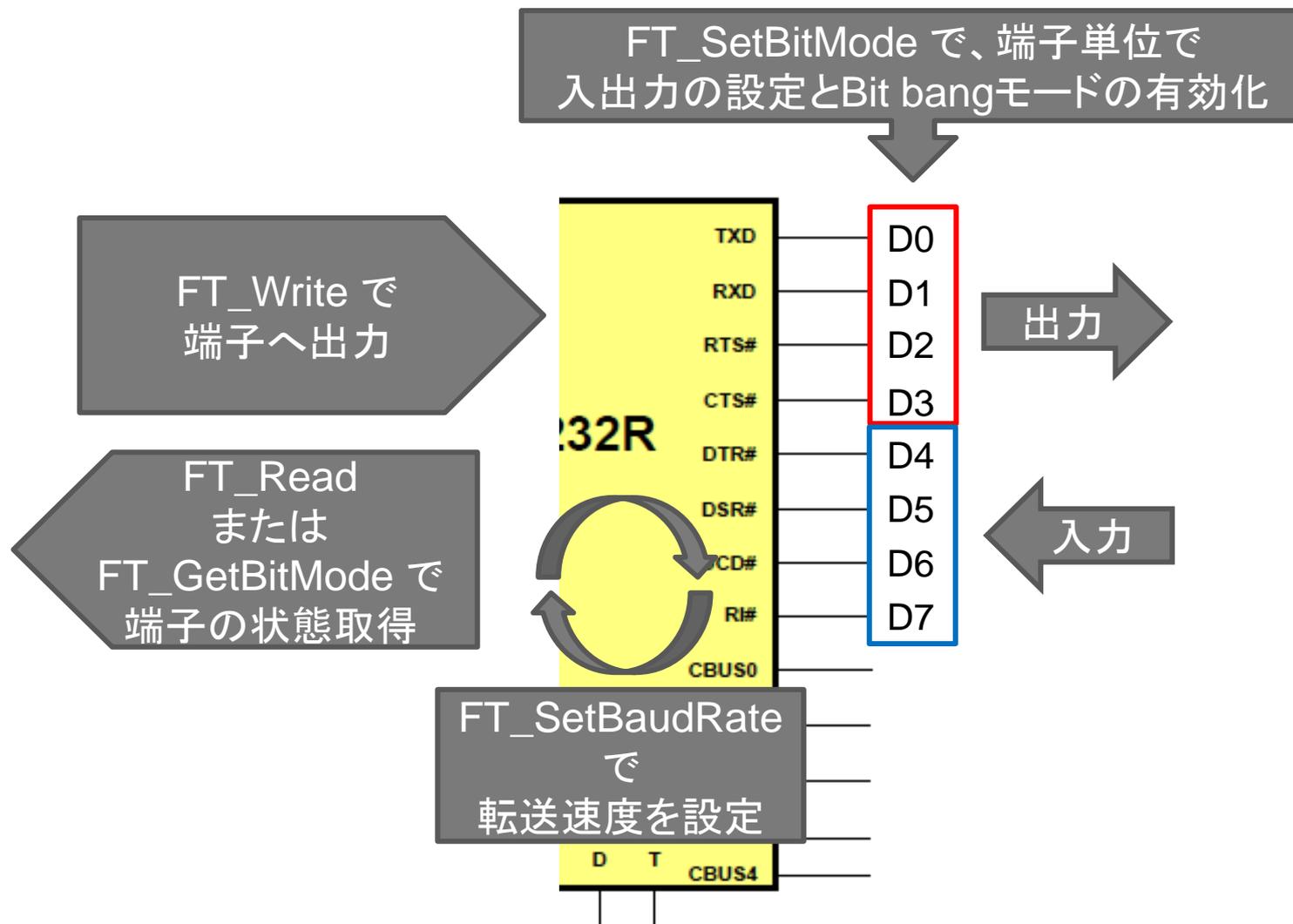
- 基本はC/C++
- C#のWrapperがFTDIのサイトで公開されている
- Linux, mac等でも開発できる(らしい)

## FTD2XXライブラリ

- ftd2xx.h
- ftd2xx.lib
- ftd2xx.dll

FTDIのサイトからダウンロードできます

# おおざっぱな流れ



# 非同期**BIT BANG**モード

**FT\_Write**で設定されたバイト列を、端子から等間隔で出力する

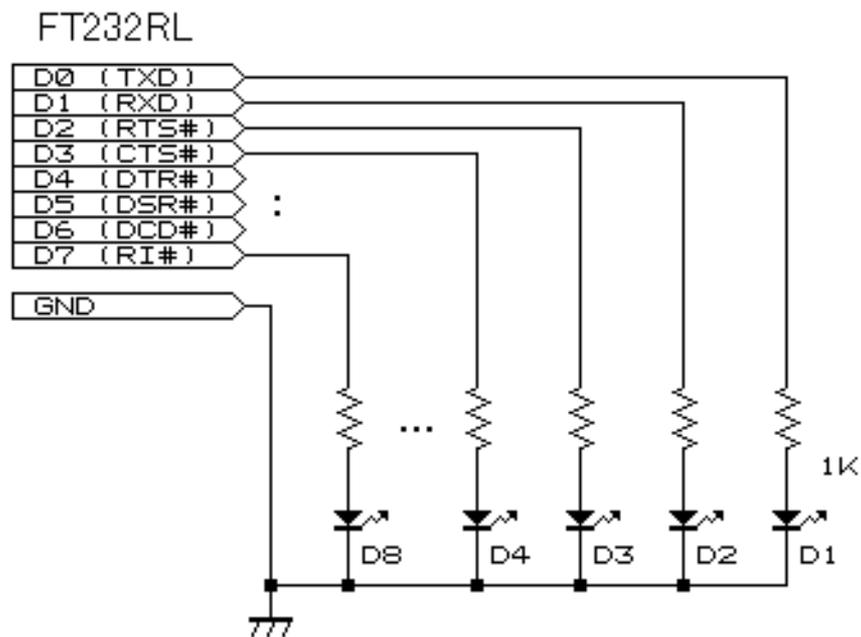
(**FT\_SetBitMode**で出力設定された端子のみ)

**端子の状態を等間隔で入力する**

- 受信したデータは、**FT\_Read**で取得
- **FT\_GetBitMode**を使えば最新の状態を取得できる

**上記2つを非同期で行う**

# 簡單！LED点滅



# ASYNCPINPUT.CPP(1)

```
#include <windows.h>
#include "..¥lib¥ftd2xx.h"
#pragma comment(lib, "..¥¥lib¥¥ftd2xx.lib")
:
FT_HANDLE      hFt;

// 1番目のデバイスをオープン
FT_Open(0, &hFt);

// 明示的にリセット
FT_SetBitMode(hFt, 0x00, FT_BITMODE_RESET);

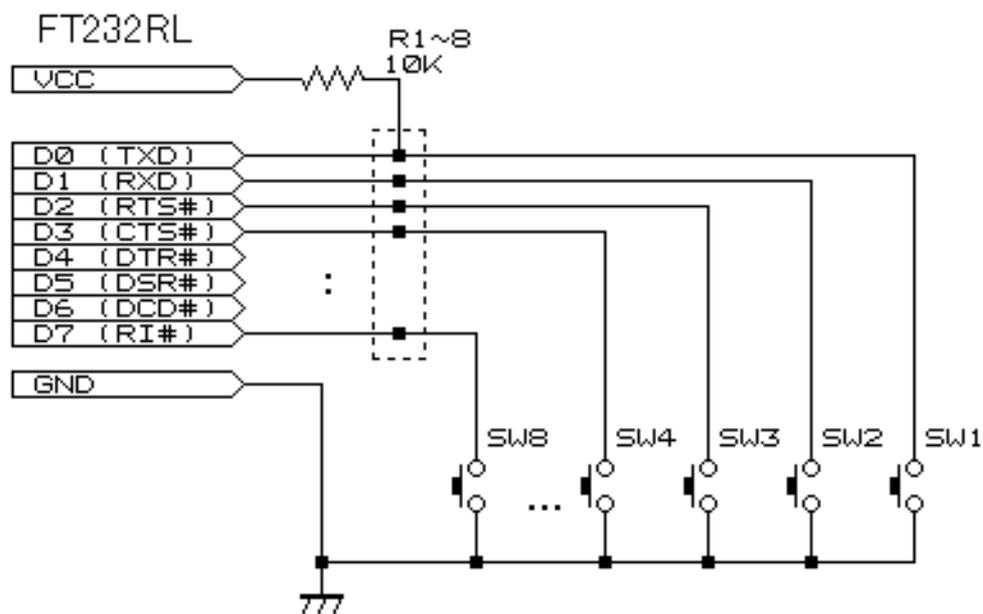
// 非同期BitBang (b0-b7:出力)
FT_SetBitMode(hFt, 0xff, FT_BITMODE_ASYNC_BITBANG);
```

# ASYNCPINPUT.CPP(2)

```
while(1)
{
    unsigned char data;
    unsigned long written;          // 出力バイト数

    // 右へ...
    for(int i=0; i<8; i++)
    {
        data = (1<<i);
        // dataの値をポートに出力
        FT_Write(hFt, &data, 1, &written);
        Sleep(200);
    }
    :
    左に流す処理は割愛
    :
}
```

# 簡単！スイッチ入力



# ASYNCOUOUTPUT.CPP(1)

```
#include <windows.h>
#include "..¥lib¥ftd2xx.h"
#pragma comment(lib, "..¥¥lib¥¥ftd2xx.lib")
:
FT_HANDLE      hFt;

// 1番目のデバイスをオープン
FT_Open(0, &hFt);

// 明示的にリセット
FT_SetBitMode(hFt, 0x00, FT_BITMODE_RESET);

// 非同期BitBang (b0-b7:出力)
FT_SetBitMode(hFt, 0xff, FT_BITMODE_ASYNC_BITBANG);
```

LED点滅と同じ！

# ASYNCOUPTPUT.CPP(2)

```
while(1)
{
    unsigned char data;

    // ポートの値をdataに格納
    FT_GetBitMode(hFt, &data);

    printf("Data=%02X¥n", data);

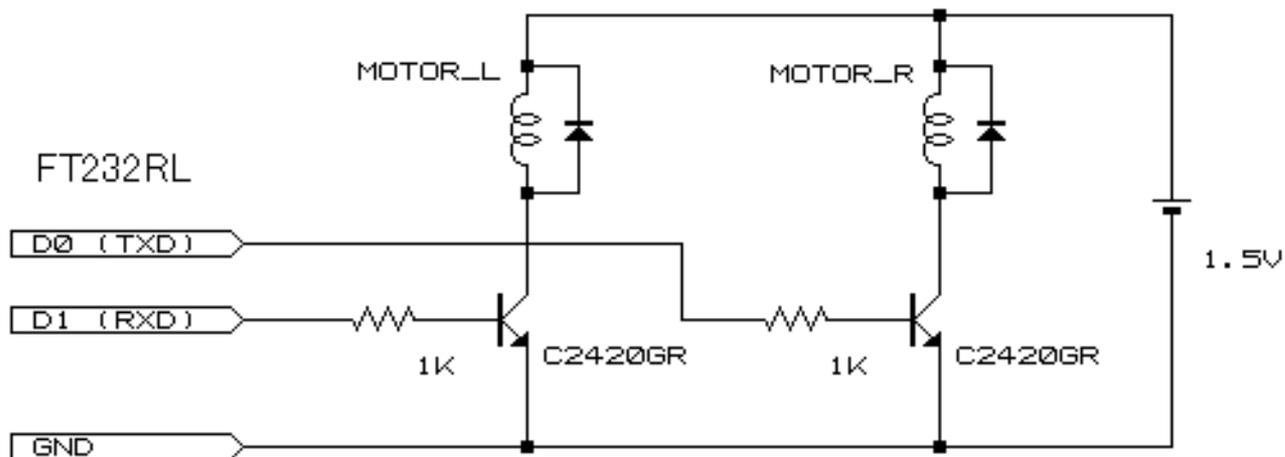
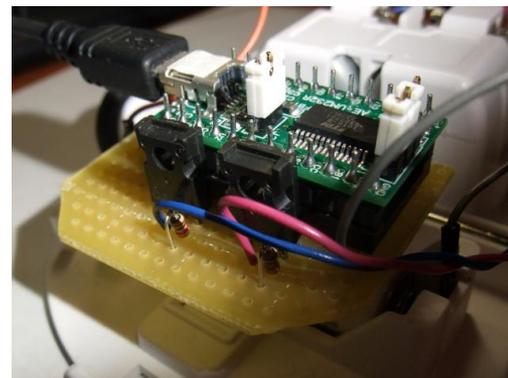
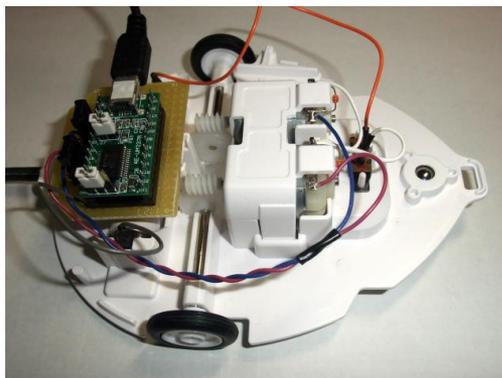
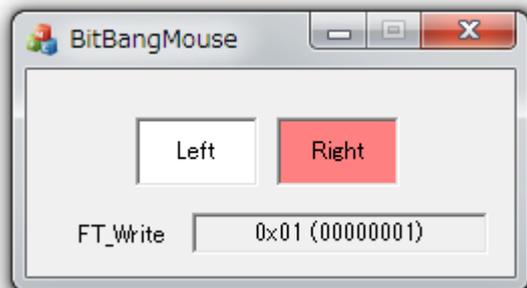
    Sleep(200);
}
```



```
d:\Develop¥FTDI¥BitBangSample¥bin¥AsyncInput.exe
Data=FF
Data=FF
Data=FB
Data=F9
Data=FF
Data=FF
Data=F8
Data=FF
Data=FB
Data=FF
Data=FF
Data=FF
```

押したスイッチのビットが0になる

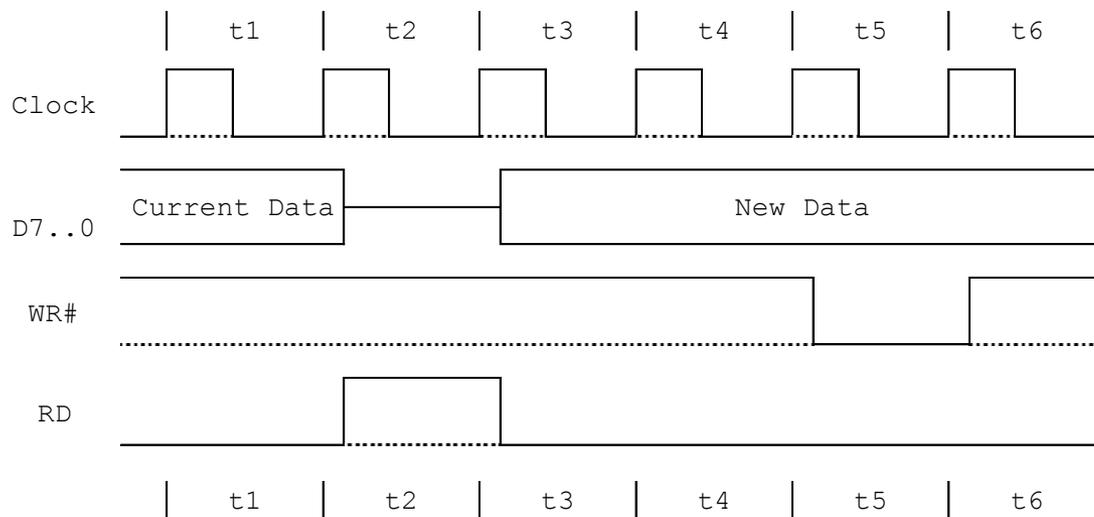
# 応用！壁づたいネズミ・改



# 同期BIT BANGモード

基本的な動きは非同期Bit Bangモードと同じ

端子情報の入力と出力を同じタイミングで行う(厳密には、入力→出力の順)



Application Note AN\_232R-01 for the FT232R and FT245R  
Bit Bang Modes Version 2.02  
3 Synchronous Bit Bang Mode

# SYNCINOUT.CPP(1)

Includeは省略

```
FT_HANDLE      hFt;

// 1番目のデバイスをオープン
FT_Open(0, &hFt);

// 転送速度を設定
FT_SetBaudRate(hFt, 9600);

// タイムアウト設定 (Recv:1000ms, Send:1000ms)
FT_SetTimeouts(hFt, 1000, 1000);

// 明示的にリセット
FT_SetBitMode(hFt, 0x00, FT_BITMODE_RESET);

// 同期BitBang (b0-b7:出力)
FT_SetBitMode(hFt, 0xff, FT_BITMODE_SYNC_BITBANG);
```

# SYNCINOUT.CPP(2)

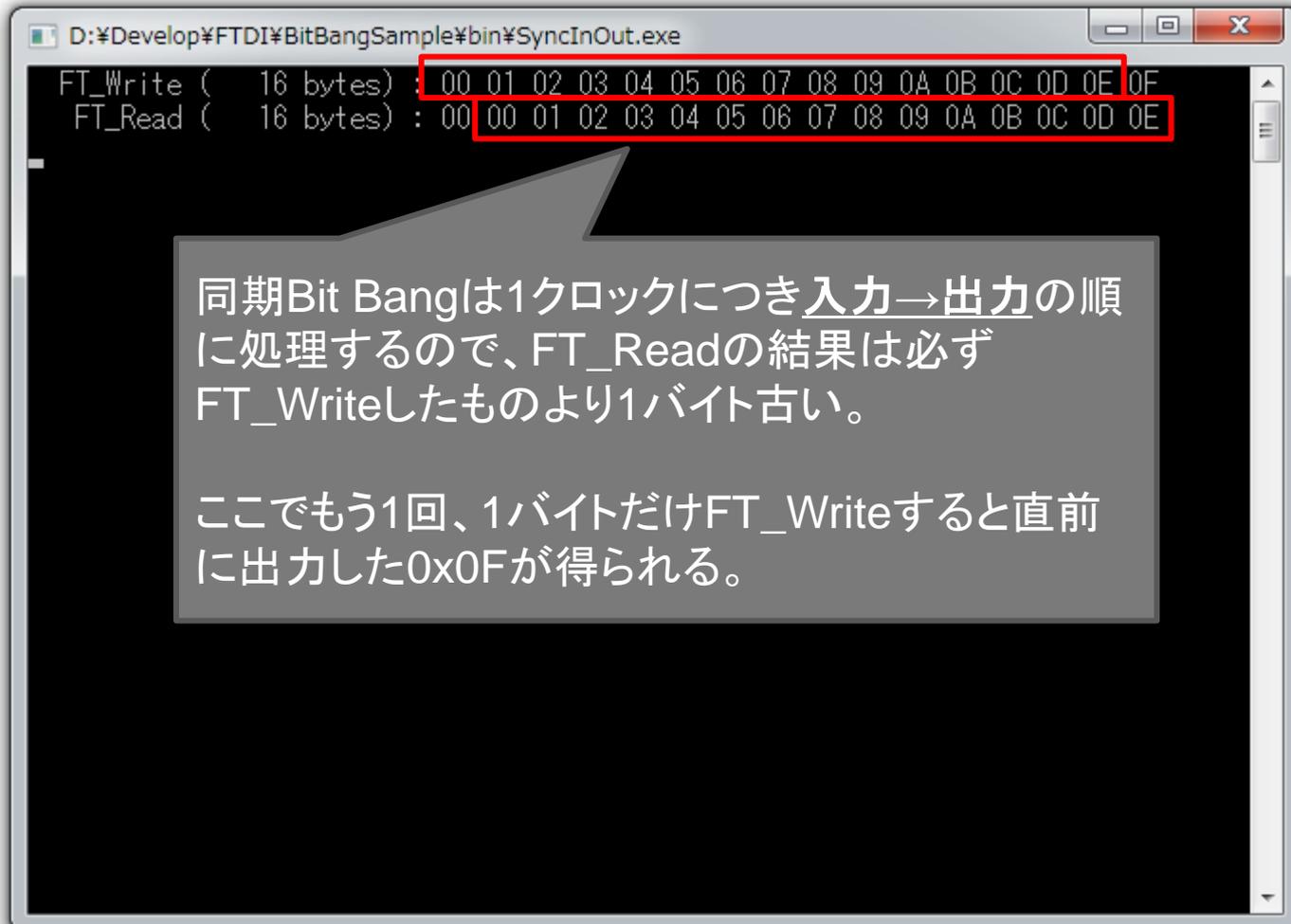
```
const int DATA_LEN = 16;
unsigned char writeData[DATA_LEN]; // 送信データ
unsigned char readData[1024];      // 受信データ
unsigned long bytesWritten;        // 出力バイト数
unsigned long bytesRead;           // 入力バイト数

for(int i = 0; i < DATA_LEN; i++)
    writeData[i] = i % 256;

FT_Write(hFt, writeData, DATA_LEN, &bytesWritten);
dump("FT_Write", writeData, bytesWritten);

FT_Read(hFt, readData, sizeof(readData), &bytesRead);
dump("FT_Read", readData, bytesRead);
```

# SYNCINOUT.CPP(3)



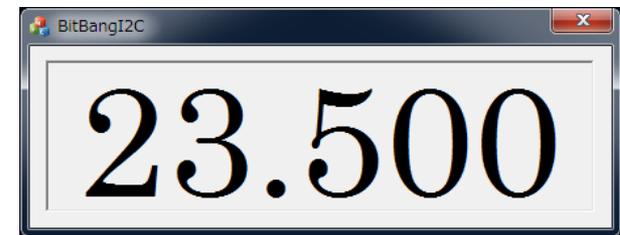
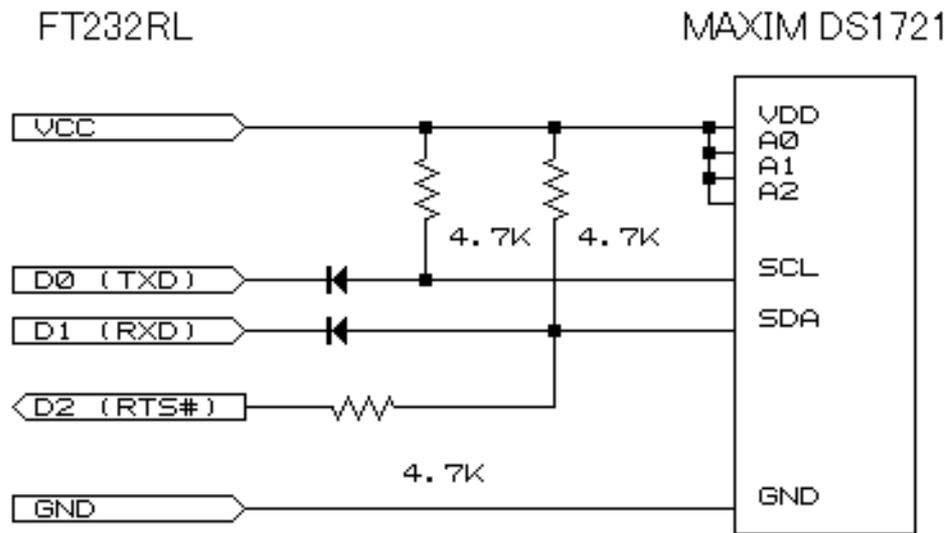
```
D:\Develop\FTDI\BitBangSample\bin\SyncInOut.exe
FT_Write ( 16 bytes) : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
FT_Read ( 16 bytes) : 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E
```

同期Bit Bangは1クロックにつき入力→出力の順に処理するので、FT\_Readの結果は必ずFT\_Writeしたものより1バイト古い。

ここでもう1回、1バイトだけFT\_Writeすると直前に出力した0x0Fが得られる。

# 同期BIT BANGでI2C温度計

同期Bit Bangモードで簡易I2C通信をやってみる



# 概要

バイト配列にSDA,SCLの波形を作り込んでおき、FT\_Writeでまとめて送信

同時に、SDA,SCLの一連の変化が受信バッファに格納されるので、FT\_Readで読み取ってI2Cの受信データに変換

同期Bit Bangなので、

N番目の受信データ=N-1番目の送信データ  
が成り立つのでビット位置の関連づけができる

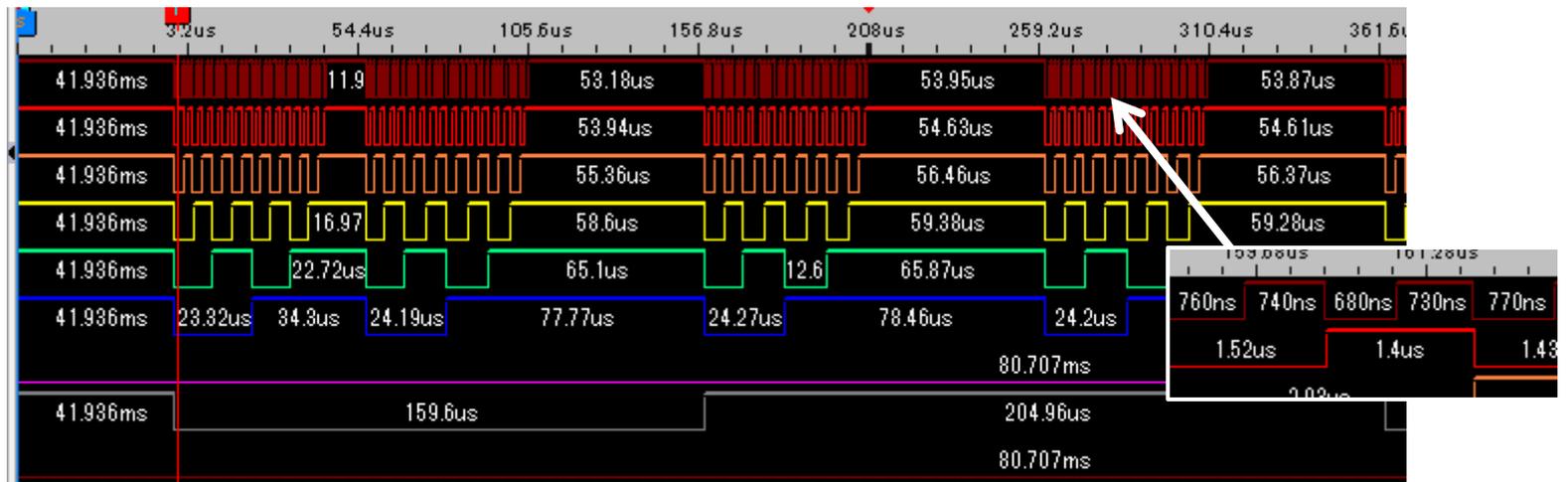
# はまりどころ(1/5)

変更した設定は、ソフトを終わらせても残っている(全部かは知らない)

- FT\_SetBitMode、SetLatencyTimerなど、初期値を期待して省略すると思わぬトラブルになるので、明示的に初期化しましょう。

# はまりどころ(2/5)

絶え間なくFT\_Writeしても出力に間隔が空くことがある

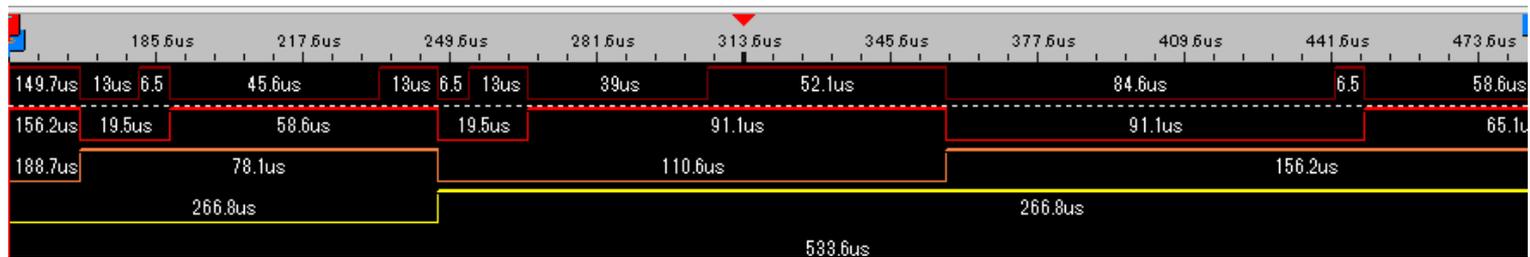


FT232RL

921,600bps設定/1024バイトのデータを連続送信し、D0-7を観測  
データは00h~FFhの循環

# はまりどころ(3/5)

基本的に等間隔で送信されない



FT232RL

9,600bps設定/1024バイトのデータを連続送信し、D0-7を観測  
データは00h~FFhの循環

# はまりどころ(4/5)

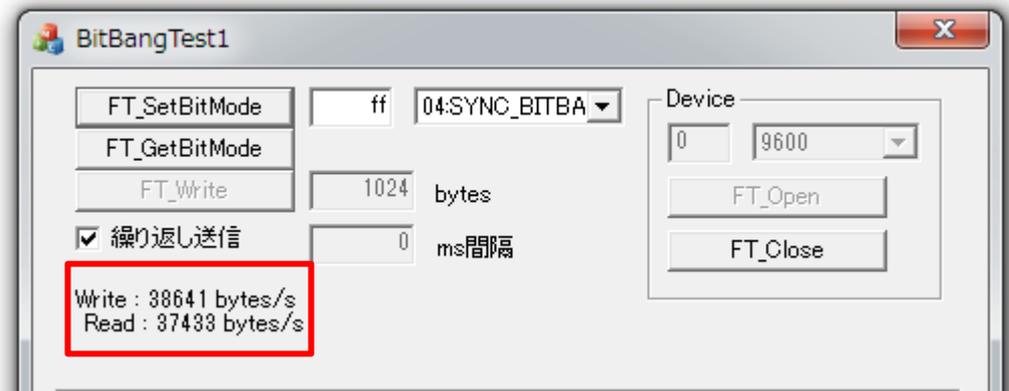
## FT\_Writeはブロッキングする

- FT\_Writeで送信している間に別スレッドでFT\_Readしても、FT\_Writeが完了するまで読み込まれなかった。
- しかし、FT\_Writeはなるべくまとめて送信した方が高速

# はまりどころ(5/5)

## Bit Bangでの転送速度が仕様書とかなり違う

- 仕様書にはBit Bangモード時の転送間隔はFT\_SetBaudRateの値 × 16と記載されているが、実測では× 4ぐらいになる



FT232RL

9,600bps設定時に約38,000bytes/s

# 読むべきドキュメント類

<http://www.ftdichip.com/Support/FTDocuments.htm>

## **FT\_000071 : Software Application Development D2XX Programmer's Guide**

- D2XXプログラマーズガイド

## **AN232BM-01**

- Bit Bangモードの解説 (FT232/245B)

## **AN\_232R-01**

- Bit Bangモードの解説 (FT232/245R)

## **AN232B-03**

- スループット改善の方法について

# その他情報

## 西日本常盤商行: USB Module

- D2XXやBit Bangモードの解説
- [http://www.tokiwa-west.co.jp/USB\\_Module/USB.htm](http://www.tokiwa-west.co.jp/USB_Module/USB.htm)

## ヒューマンデータ: USB関連参考資料

- 資料多数、評価ボードの販売も行っている
- <http://www.hdl.co.jp/USB/FTDI/index.html>

## ore-kb.net: Bit-Bangモードの使い方

- 手前味噌ですが、自分のところにも
- <http://ore-kb.net/hard/BitBang/>

ご清聴ありがとうございました！