



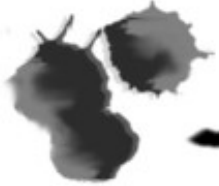
PureRuby でロボット制御

にしたかお <zophos@koka-in.org>
2009/9/19



胡瓜蔵

まずは自己紹介



- zophos, にし @ おおさか ,
 @@@@
- IRCNet %Ruby でチャチャ入れ
- るびまの編集




やられっぱなし



胡瓜蔵

やられっぱなしの歴史




- 2000 年頃：IRC フレームワーク “RICA”
→ \ay さんの rice
- 2004 年頃：付箋型 Wiki “poti”
→ ふしはらさんの wema
- 2005 年頃：Web ベースプレゼンツール
“Tunggudulu”
→ google docs

etc...



胡瓜胡

やられてないのもありますよ



- EXIF パーサ “ rexif ”

あきた

- 日本 Ruby の会 名刺ジェネレータ

あきた

- QR コードデコーダ “ libdecodeqr ”

あきつつあり



胡瓜蔵

本業は



- 某研究所勤務
- 「マルチカメラシステムを用いた三次元形状認識の研究」
- 要するにロボットの目
- 来年ラボがなくなりそう...

やっぱりやられっぱなし



胡瓜蔵

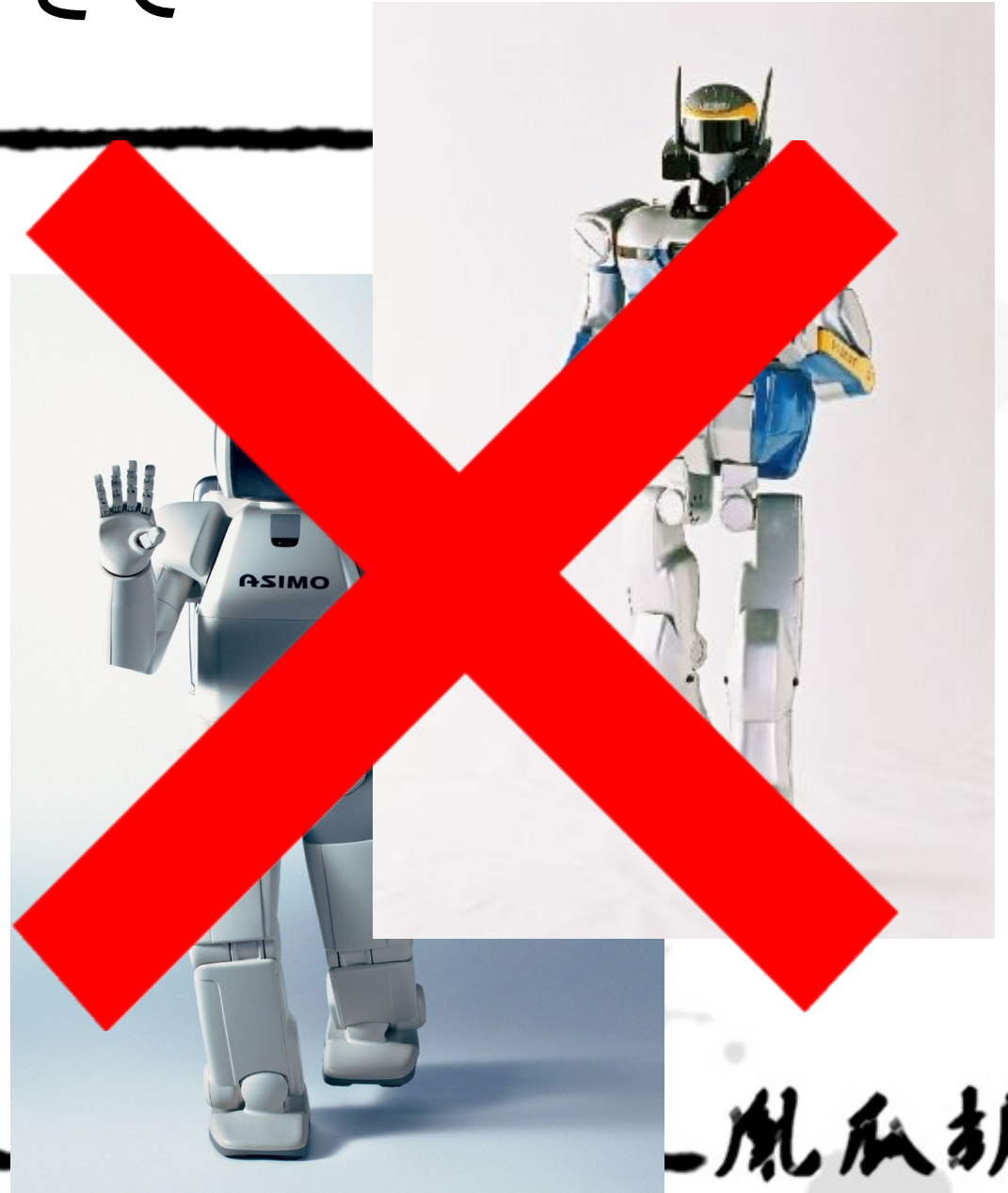
絶望した！

さて

ロボットですが

こんなんとか
こんなんとか

できません



胡瓜風

かわりに

これつかいます

ELEKIT MR-999



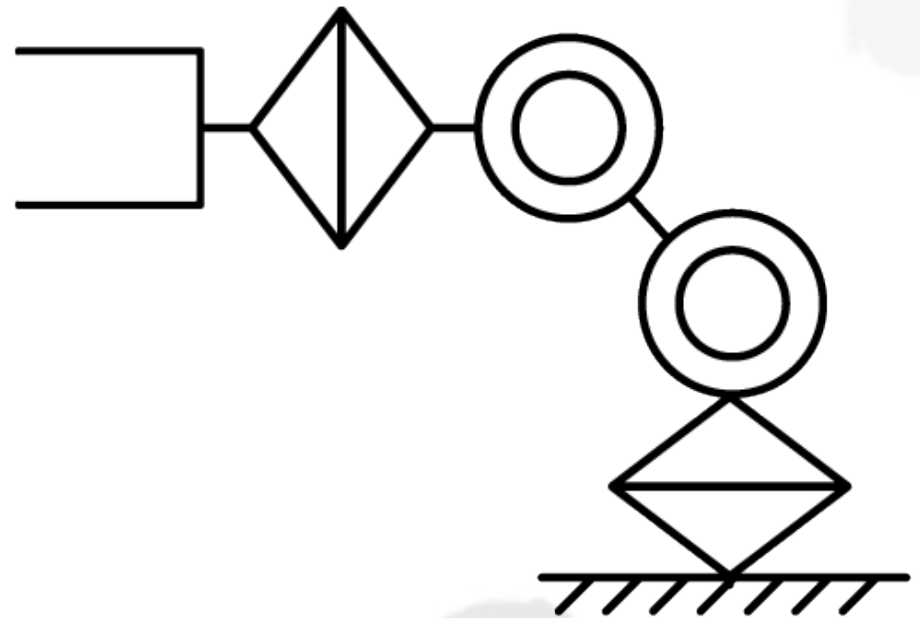
胡瓜鳳

ELEKIT MR-999

- 5 軸 4 自由度
 - 人間は 7 自由度
- 各関節は DC モータ
- 純正制御 I/F ユニット
IF-100



軟弱ロボット屋には魅
力たっぷり



胡瓜蔵

制御 I/F ユニット IF-100

- USB 接続
- .NET な専用ソフト
 - Mono では動作不可
- プロトコル非公開




好き勝手にけへんやん



胡瓜蔵

絶望した！

好き勝手してみる

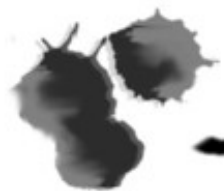


- Linux Box につないでみると
 - HID (Human Interface Devices) として認識
 - Morphy Planning の文字列が...
 - Vendor Id: 0x12ed, Product Id: 0x1003
- テクノキットの USB-IO
- これがわかれば勝ったも同然



胡瓜蔵

USB-IO



- 元はモルフィー企画
 - 現在はテクノキットと Km2Net の 2 社が引き継ぎ
 - Vendor Id, Product Id が違うだけ
- Port0 8bit, Port1 4bit の計 12bit DIO
- 標準 HID ドライバでアクセス可能
- 詳しくはぐぐれ



胡瓜蔵

ここまでできたら



- あとは出力 bit とモータの関係を調べるだけ
- 地道に一個ずつ叩いて行っても知れてるけど
- 制御用ソフトの出力見た方が楽だね

一般人は USB アナライザなんて持ってない



胡瓜蔵

snoopypro



<http://sourceforge.net/projects/usbsnoop/files/>

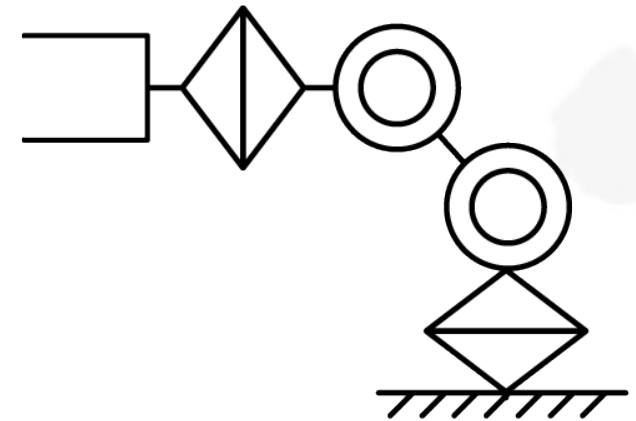
- Windows 用 USB パケットスニファ
- Windows 用ドライバのある USB デバイスならほとんどキャプチャ可能



胡瓜藏

解析結果

Port	Bit	Hex	Joint
0	0	0x01	5(+)
	1	0x02	5(-)
	2	0x04	3(+)
	3	0x08	3(-)
	4	0x10	1(+)
	5	0x20	1(-)
	6	0x40	2(+)
	7	0x80	2(-)
1	0	0x01	4(+)
	1	0x02	4(-)
	2	0x04	N/C
	3	0x08	



- Lo 出力で ON
- 出力直後に全 bit 読み取り
- 定期的にストロブ信号出力



胡瓜藏

さあ書くぞ



- USB-IO を Ruby で叩くのはやられてる
 - DLL 経由でほげ
- わざわざ DLL 作らんでもいけるんちゃうん
- どうせなら PureRuby で



胡瓜蔵

HID のアクセス (1)



- libusb を使うと portable
 - Ruby binding: 田中哲さんの ruby-usb
<http://www.a-k-r.org/ruby-usb/>
 - libusb-win32 は HID 関連にいろいろ難あり
- Linux の場合
 1. /dev/usb/lp0* あたりを open() して
 2. ioctl() でモゲ でいいはずすみませんちゃんと確認してません



胡瓜蔵

HID のアクセス (2)



- Win32API の場合

1. HidD_GetHidGuid() で HID を探す
2. SetupDiGetClassDevs() で HID クラスデバイスをリストアップ
3. SetupDiEnumDeviceInterfaces() と SetupDiGetDeviceInterfaceDetail() でパス名に変換
4. CreateFile() でデバイスを開く
5. HidD_GetAttributes() で Vendor Id, Product Id をキーに目的のデバイスを見つける



PureRuby しばり…



- どこまでが PureRuby?
→ 標準添付ライブラリまで by オレ

Ruby/DL は Ruby1.8 から標準添付っ



胡瓜蔵

Ruby/DL のおさらい



- Ruby から shared library をつかう仕組み
 - require 'dl/import' して
 - 必要なら require 'dl/struct' もして
 - 適当な module 内で extend DL::Importable して
 - load したい *.[(so)|(dll)] を dlopen して
 - 関数宣言を extern



たとえば

```
require 'dl/import'
```

```
module Win32
```

```
  extend DL::Importable  
  dload('user32.dll')
```

```
  extern(<<_EOS_  
int MessageBox(void *,  
                unsigned char *,  
                unsigned char *,  
                unsigned int)
```

```
  _EOS_  
  )
```

```
end
```

```
Win32.messageBox(nil,'world','hello',1)
```

Ruby/DL で HID アクセス



- C のヘッダファイルから
 - 構造体を struct, 共用体を union
 - Ruby/DL の知らない型を typealias
 - 関数プロトタイプを extern

数が多いとめんどくさい



VB 用定義ファイルから機械変換
ところどころに罫が...



胡瓜蔵

いろいろ書いた

- HID アクセスは IO っぽい RHid クラス
 - 一応他の方法やプラットフォームでの実装も考えて
- USB-IO 固有の機能は Usblo クラス
- 各関節は Joint クラス
- 関節の集合体として MR999 クラス
 - 動作の定義は MR999 クラスで

抽象化ですっきり



胡瓜蔵

そんな訳でデモ

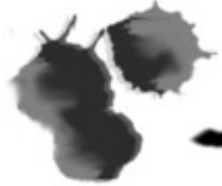


振ります



胡瓜蔵

欠陥



- 精密な動きができない
- 自分の姿勢がわからない
- PureRuby である必要がない
- ネギ臭い



胡瓜蔵

絶（略）

ちょっと真面目に



- Ruby で機械制御
 - 実時間制御は厳しい
 - 実行時間が読めない
 - GC が一番のネック
 - オブジェクト指向と機械制御は相性が良い
 - 時間制限が厳しくない制御用途

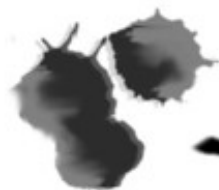


Ruby を産業用ロボットのプログラム言語に!!



胡瓜蔵

まとめ



- Ruby/DL は黒魔術
- Ruby で制御してる車には乗りたくない
- あとは keisuken がなんとかしてくれる



胡瓜蔵



<http://github.com/zophos/rhid>



胡瓜風