

実践IoTハウス

<http://www.pepolinux.com>

Twitter@kujiranodanna

山内

IOTやっていますか？

- ◆ IoT【 Internet of Things 】とは インターネット オブシングス
- ◆ コンピュータなどの情報・通信機器だけでなく、世の中に存在する様々な物体(モノ)に通信機能を持たせ、インターネットに接続したり・・・自動制御、遠隔計測などを行うこと。
- ◆ 引用：<http://e-words.jp/w/IoT.html>

IOTハウスやっていますか？

- ◆ IOTハウスとは・・・インターネット経由でなんか家の中の物を動かす
- ◆ インターネットに繋がってたら家の物が勝手に動く？少し気持ち悪いかも
- ◆ スマートハウスとは・・・スマートフォンでなんか家の中の物を動かす
- ◆ 個人的な解釈ですが
- ◆ さて皆さんは？

IOTハウスやっています

- ◆ 出先から部屋の温度を確認してクーラー運転する
- ◆ 夕方に外が暗くなると玄関や部屋の明かりを点ける
- ◆ 夜に外の湿度を元に家庭菜園の水やりをする
- ◆ 防犯センサーがはたらくと動画をメールする
- ◆ 火災感知器が発報するとメールと電話(ワンギリ)で知らせる
- ◆ などRaspberry PiでIoTハウスを実践しています。
- ◆ 引用: <https://ja.osdn.net/projects/pepolinux/>

IOTハウス

Home Electronics & Security



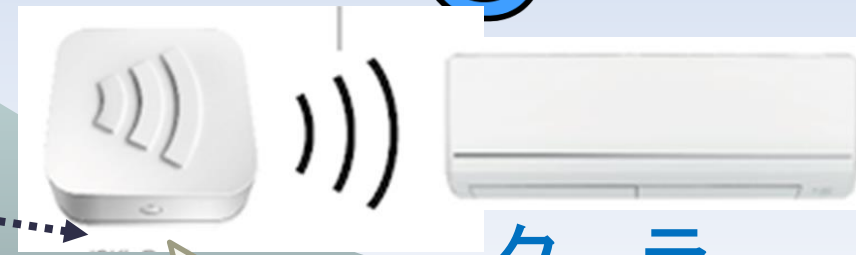
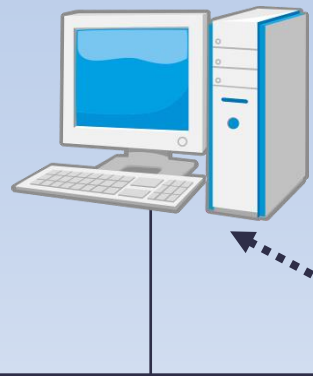
実践IoTハウス

Raspberry PiでIoTハウスを実践しています。



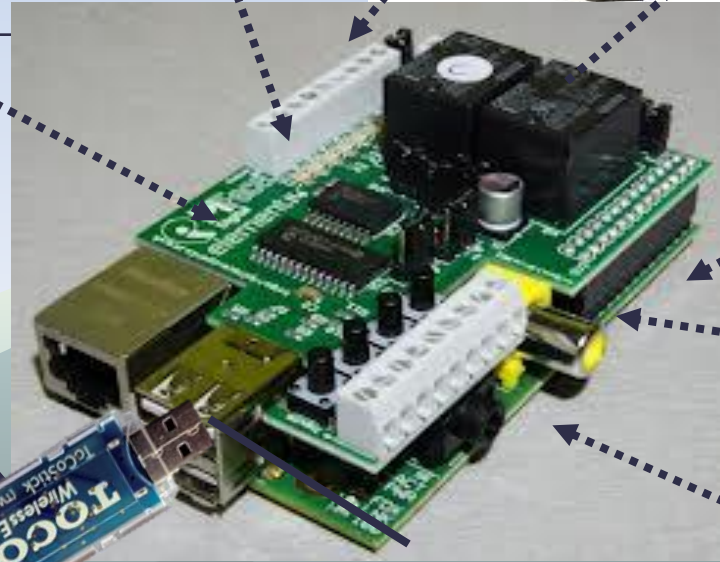
実践IoTハウス

Home Electronics & Security



クーラー

IRKit
学習型赤外線リモコン



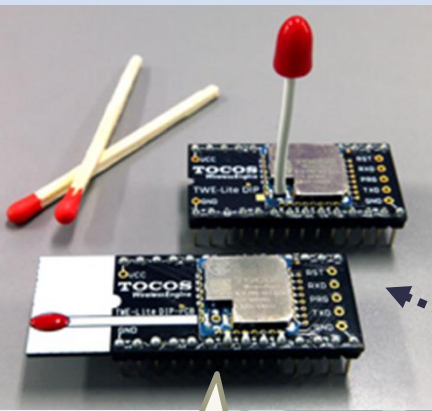
火災報知器発報で動画メール
と電話同時通知

Raspberry Pi + Piface
Gmail連携で動物監視
外部機器の監視制御

リモートハンド



Tocos無線モジュール
離れた物を監視制御



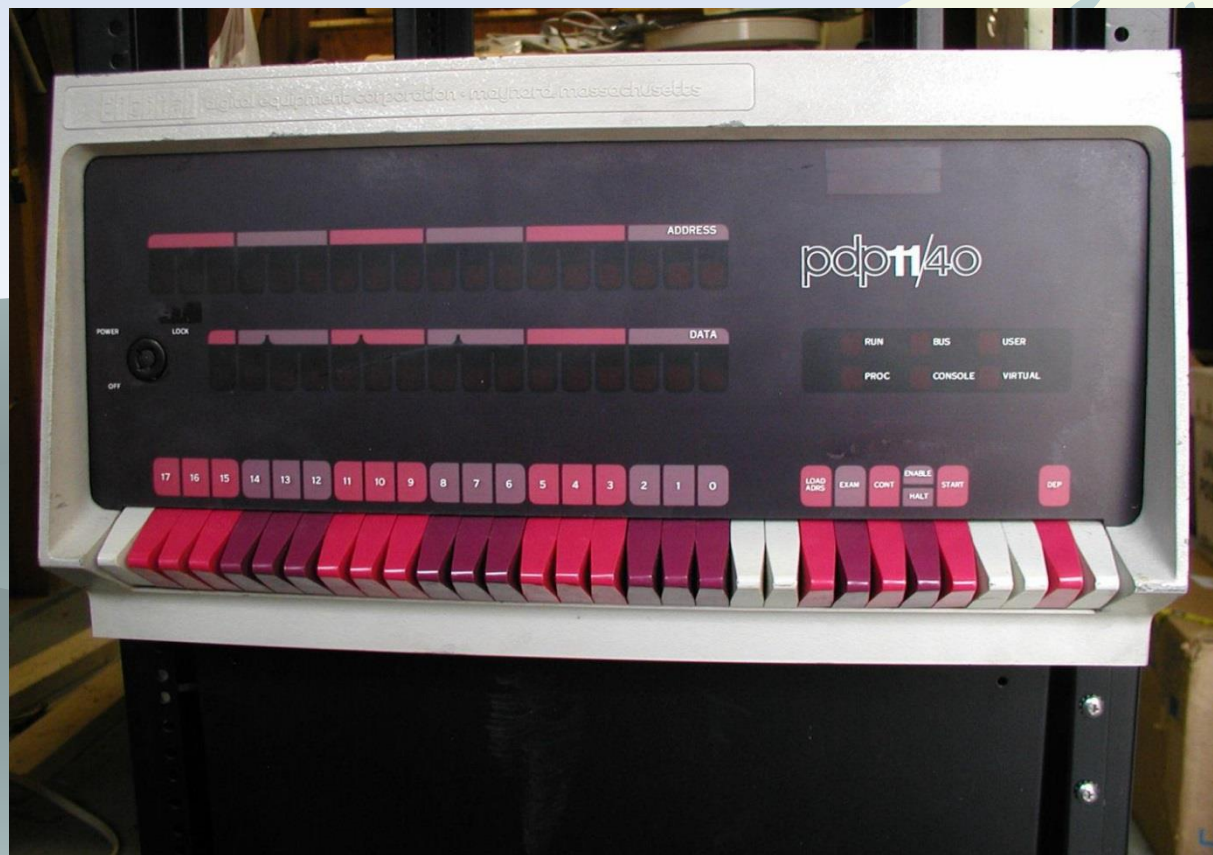
Raspberry Piは持っているけど

- ◆ 勢いで買ってしまった
- ◆ 使い道に悩んでいる
- ◆ まだLチカ
 - ◆ 40年前はミニコン全盛期、Lチカの事をランプパカパカ
 - ◆ あこがれのUnix、PDP11のピアノスイッチ...
- ◆ 机や押入れで眠らせている

Raspberry piで遊んだ後はPifaceを買ってホーム
エレクトロニクスとホームセキュリティで実用しよう

少しだけPDP11

出典 <http://www.vintagecomputer.net/digital/pdp11-40/>



少しだけレガシーシステム 40年前



磁気DISK

5MB

磁気TAPE

50MB

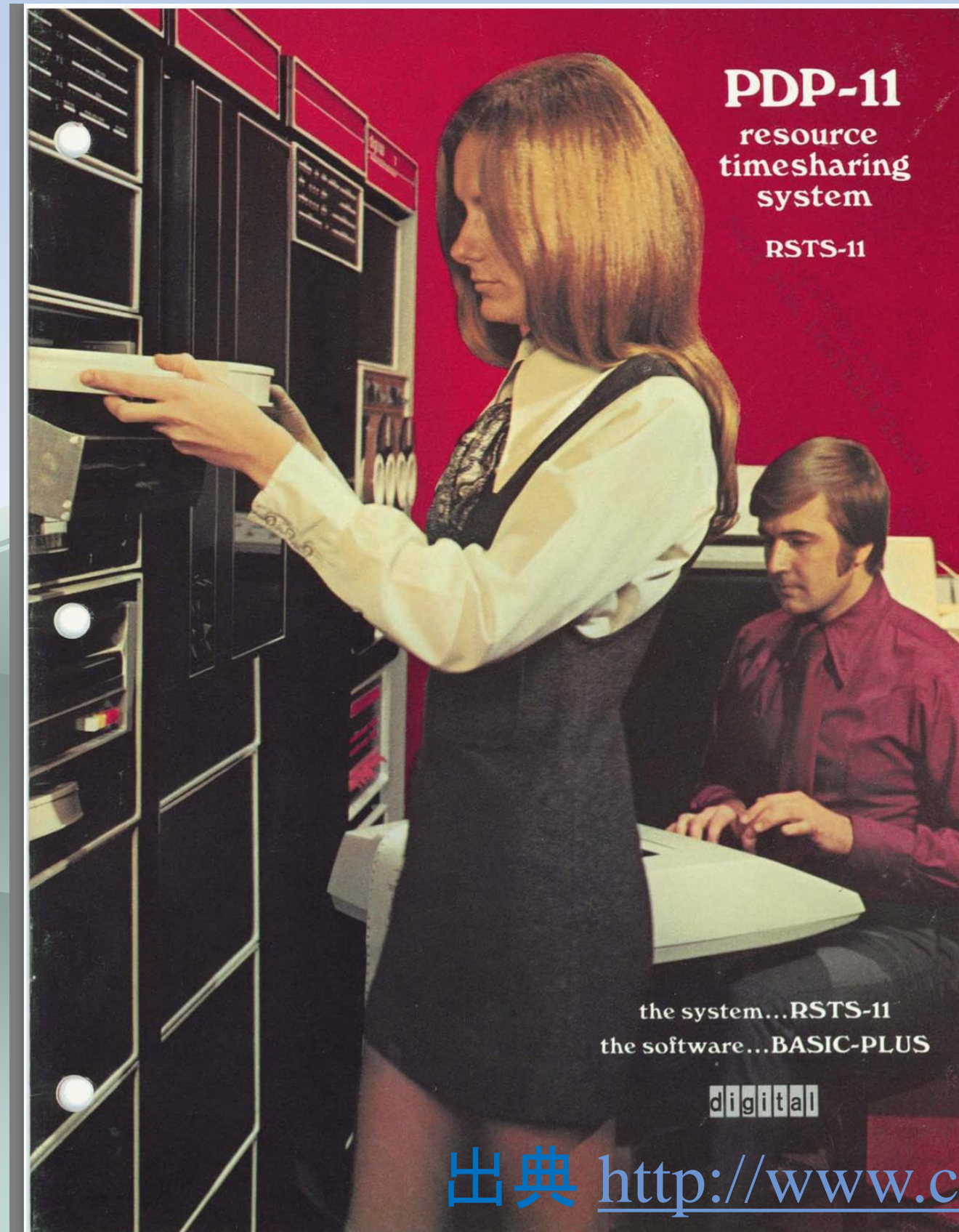
タイプライター

ラインプリンタ

パンチカード

紙テープ

少しだけレガシーシステム



レガシーシステムなら幾らでもお話 出来るのですが

- ◆ ランイプリンタ・レーザープリンタ
- ◆ シリアルプリンタ・母型・ドットインパクト
- ◆ カートリッジ磁気DISK
- ◆ 磁気テープ
- ◆ フロッピーDISK
- ◆ コア・ワイヤーメモリ
- ◆ モデム・通信制御・プロトコル
- ◆ 昔はCEやネットワークSEやってみました

三都物語＋北海道＋東京＋名古屋＋広島

kof2002→osc2011kobe→osc2011kyoto→kof2011→kof2012→osc2013Kyoto
→kof2013→osc2014hokkaido→kof2014→osc2015Tokyo→osc2015Nagoya→
osc2015kyoto→kof2015→osc2016nagoya→kof2016→osc2016hirosima



OSDN

PepoLinux 1CDLinux開発

Remote-Hand VMware Player ▶ デバイス ▶

PepoLinux Boot Menu

PepoLinuxリモートハンドイメージ

- 1 PepoLinux normal
- 2 PepoLinux unionfs
- 3 PepoLinux few memo メンテナンス対象機
- 4 PepoLinux HDD/pepolinux unionfs not used
- 5 PepoLinux HDD/pepolinux unionfs used
- 6 PepoLinux debug mode without any auto probes 社内PCより公衆網経由にてユーザーのネットワーク機器のリモートメンテナンスを行う。

Press [Tab] to edit options
Automatic boot in 8 seconds.

IP-VPN

トからCOMポート経由にてルーターの設定をダウンロードできる

HUB

COM

モジュール

被メンテナンス対象機

現地保守員は持参のモバイルPCへ携帯カード+携帯電話を接続PepoLinuxを立ち上げ後の操作はリモートよりメンテナンスを行う。

携帯電話網 公衆電話網

Welcome to PepoLinux loginterm

この仮想マシンに入力先を切り替えるには、Ctrl+G を押してください。

vmware

シリアル通信ソフトepicon開発

epicon - もっと早くもっと

sourceforge.jp/projects/pepolinux/wiki/epicon

Web スライス ギャ... おすすめサイト 今日の番組表 [地上...]

その他のブックマーク

最初に

epiconとはLinuxで使えるシリアル・コンソール用の通信ソフトです。SwitchやRouterなどシリアルポートでConfigを設定するネットワーク機器はメーカーや機種を問わず使えると思います。Cisco製ルータなどの設定を自動化する際、事前に作成したテキスト・データをコピーしてシリアル・コンソール画面へ貼り付けコンフィグの流し込みを行います。この時、文字と改行の送出デレイを挿入しコンフィグ・データの取りこぼしを防ぐことが必要です。ルータのシリアル・コンソールとして重要な文字と改行の送出デレイをepiconはサポートしているのでコンフィグを安心してコピー&ペーストが出来ます。この他、簡易telnet、zmodemなどのファイル転送ソフト、shell、マクロ、外部ソフトの起動などCUIだが多機能でコンパクトな作りとなっています。

epiconを使ったリモート保守イメージ

PepoLinuxリモートハンドイメージ

被メンテナンス対象機

リモートからCOMポート経由にてルータの設定をダウンロードできる

COM

被メンテナンス対象機

社内PCより公衆網経由にてユーザーのネットワーク機器のリモートメンテナンスを行う。

モデム

被メンテナンス対象機

IP-VPN

被メンテナンス対象機

HUB

被メンテナンス対象機

現地保守員は持参のモバイルPCへ携帯カード+携帯電話を接続PepoLinuxを立ち上げ後の操作はリモートよりメンテナンスを行う。

携帯電話網 公衆電話網

最近の更新 (Recent Changes)

2011-10-28

- chobit_prog pepogmail_send
- chobit_prog pepodioid

2011-10-25

- chobit_prog init.d_pepodiod

2011-10-20

- chobit_prog pepodioidctl

2011-10-17

- epicon

2011-10-02

- chobit_prog podcats_get

最新リリース情報

Remot-Hand2VPN (0.3)	2009-04-27
Remote-Hand (v0.2)	2011-06-03
acm-FOMA (v0.26.FOMA.10)	2011-05-28
chobit_prog (pepogmail_send)	2011-10-28
epicon (5.0)	2011-10-17
kernel-source (linux-2.6.38.6-pepo)	2011-06-02
pepolinux (0.49)	2011-10-19

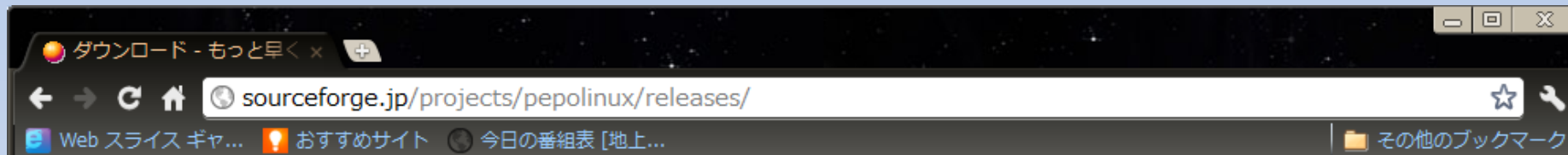
Wikiガイド (Guide)

- Wikiの文法
- リンクの種類と文法
- ブロックプロセッサ
- 拡張文法
- サイドバー
- プロジェクト Wikiでの広告設定

サイドバー (Side Bar)

- このサイドバーについて
- このサイドバーの編集

acm-FOMA Driver開発



- USB接続が出来るFOMA携帯電話用のLinux Kernel-2.6以降のdriver
- /drivers/usb/class/cdc-acm.cをdocomoのFOMAのdriverとして改造、2007.3.31初リリース
- 当初FOMAの機種毎にコード追加してリリースを、余りに早い新機種ラッシュそれに対応すべく2007.11.17に機種別コードを不要へ
- 従来からMODEMモードのみでしたが、2011.5.28 v0.26.10からはOBEXとATcommandの3モードをサポート出来るように



Remote-Hand開発

Remote-Hand2VPNとは、VMware Playerと1CDLinuxをベースにGUIでOpenVPNの設定を手軽に行いVPN越しにRemote-Handを実現させる、VMware Applianceです。

OpenVPNの情報サイトは数多あるが、ドキュメントを読み進めないと簡単に構築できないのではないのでしょうか。ややネットワークの知識があるがWebUIでサーバ/クライアントの鍵の生成・廃止・ダウンロードなどが5分程度で設定出来るようにしました。

それとOpenVPNは会社から自宅PCなどへ簡単に接続が出来てしまう優れたものである反面、セキュリティ上企業にとって大変危険です、これらをよく理解して正しい目的で使いましょう。

サーバPC (WindowsXP)とクライアントPCを用意、VMware Playerが事前にインストールされていて、満足に操作出来ることを前提として、下記にリモートハンドの使用説明をします。

Remote-Hand2VPN接続構成図v0.3

サーバ側 (左):
WindowsXP pro
VMware Player
Openvpn Server
eth0:dhcp-client
eth1:host-only/dhcp-client
eth2:host-only
eth2@tap0:bridge interface
br0:bridge(VPN IP interface)

クライアント側 (右):
WindowsXP pro
VMware Player
Openvpn Client
eth0:dhcp-client
eth1:host-only/dhcp-client
eth2:host-only or bridge
tap0:(VPN IP interface)

- OpenVPNはブリッジモード(TAP)
- iptablesでフィルタリング可能な構成(初期:ICMPのみ許可)
- Client→Serverセグメント全体と通信可
- オプションでServer→Clientセグメント全体と通信可

Powered by [PepoLinux](#)

最近の更新 (Recent Changes)

- 2011-10-28
 - [chobit_prog pepogmail_send](#)
 - [chobit_prog pepodiod](#)
- 2011-10-25
 - [chobit_prog init.d.pepodiod](#)
- 2011-10-20
 - [chobit_prog pepodioctl](#)
- 2011-10-17
 - [epicon](#)
- 2011-10-02
 - [chobit_prog podcats_get](#)

最新リリース情報

Remot-Hand2VPN (0.3)	2009-04-27
Remote-Hand (v0.2)	2011-06-03
acm-FOMA (v0.26.FOMA.10)	2011-05-28
chobit_prog (pepogmail_send)	2011-10-28
epicon (5.0)	2011-10-17
kernel-source (linux-2.6.38.6-pepo)	2011-06-02
pepolinux (0.49)	2011-10-19

Wikiガイド (Guide)

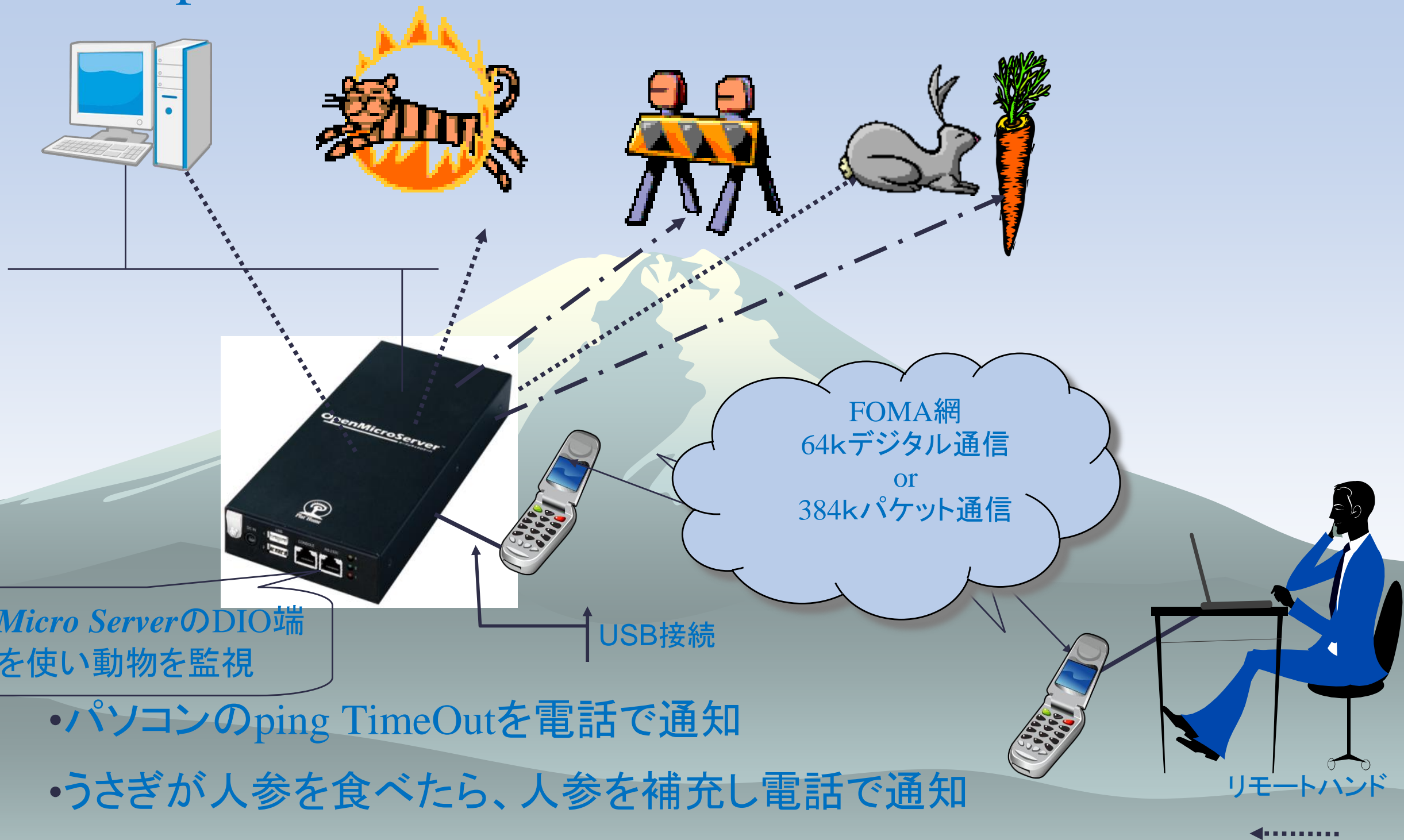
- [Wikiの文法](#)
- [リンクの種類と文法](#)
- [ブロックプロセッサ](#)
- [拡張文法](#)
- [サイドバー](#)
- [プロジェクト Wikiでの広告設定](#)

サイドバー (Side Bar)

- [このサイドバーについて](#)
- [このサイドバーの編集](#)

PepoLinux

OpenMicroServer + FOMA Remote-Hand



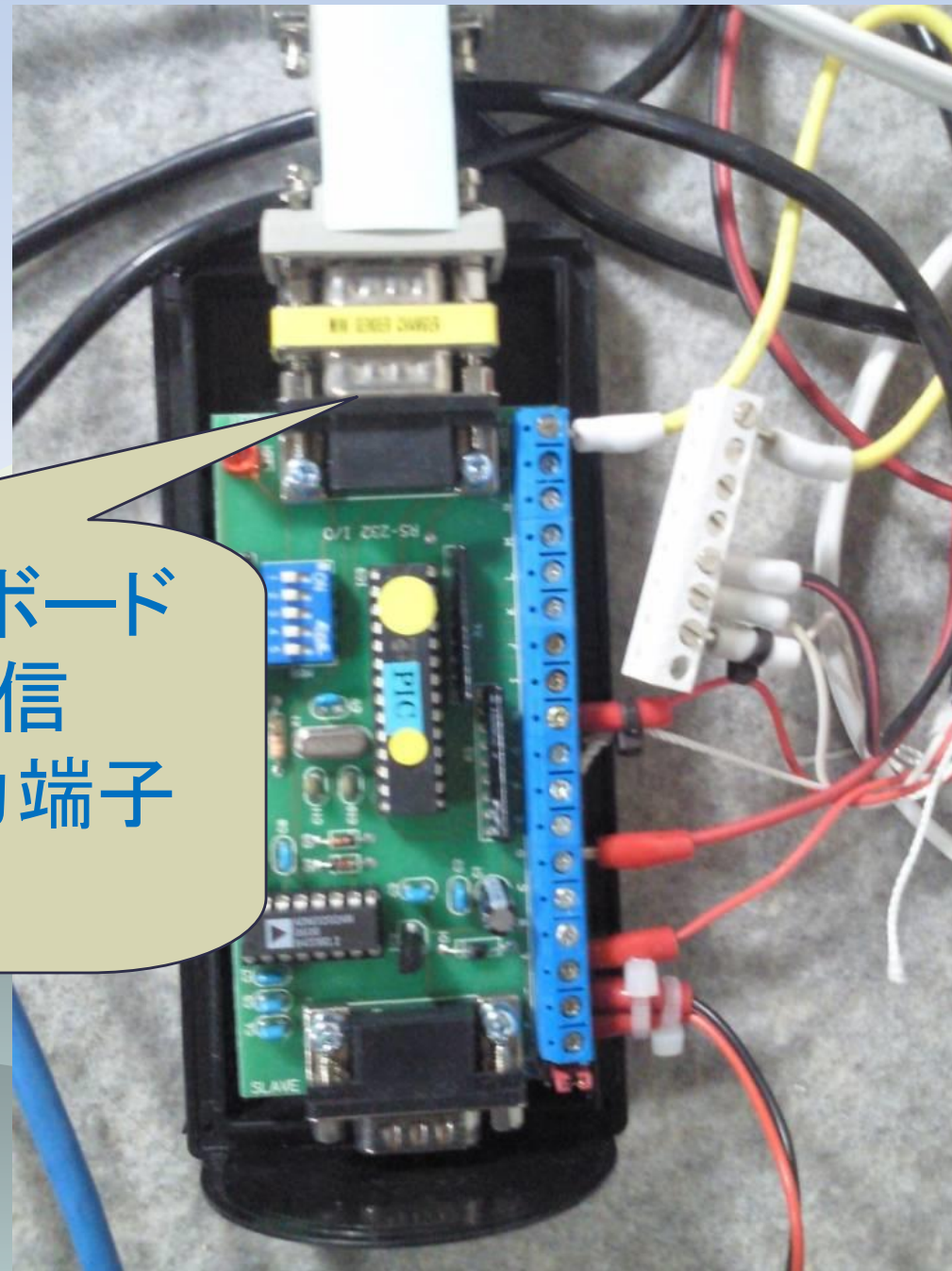
- パソコンのping TimeOutを電話で通知
- うさぎが人参を食べたら、人参を補充し電話で通知
- トラが檻を破って脱走したら、パトライトを点灯
- うさぎやトラをどうやって認識させる

OpenMicroServerでRemote-Hand

- OpenMicroserverはDIOを4ビット内蔵
 - Digital信号(0v,3.3v)出力でAC100駆動させるには別途Driver(TR,SCR,FLS等)必要
- 如何せんとも遅いmips系CPU(400MHz)
- 値段が6万位とお高い割に結構しょぼい
- 2011年早々製造中止
- 次機種にはDIOは内蔵されていない
- という事でOpenMicroserverの代替えを探す事に

DIOシリアルボード

市販のDIOシリアルボード
9600bps非同期通信
5v14本の入力／出力端子
3,000円程度



PepoLinux

Vmplayer+PepoLinux+DIO+FOMA=Remote-Hand



Vmplayer+PepoLinux+シリアルDIOボードで動物を監視

FOMA網
64kデジタル通信
or
384kパケット通信

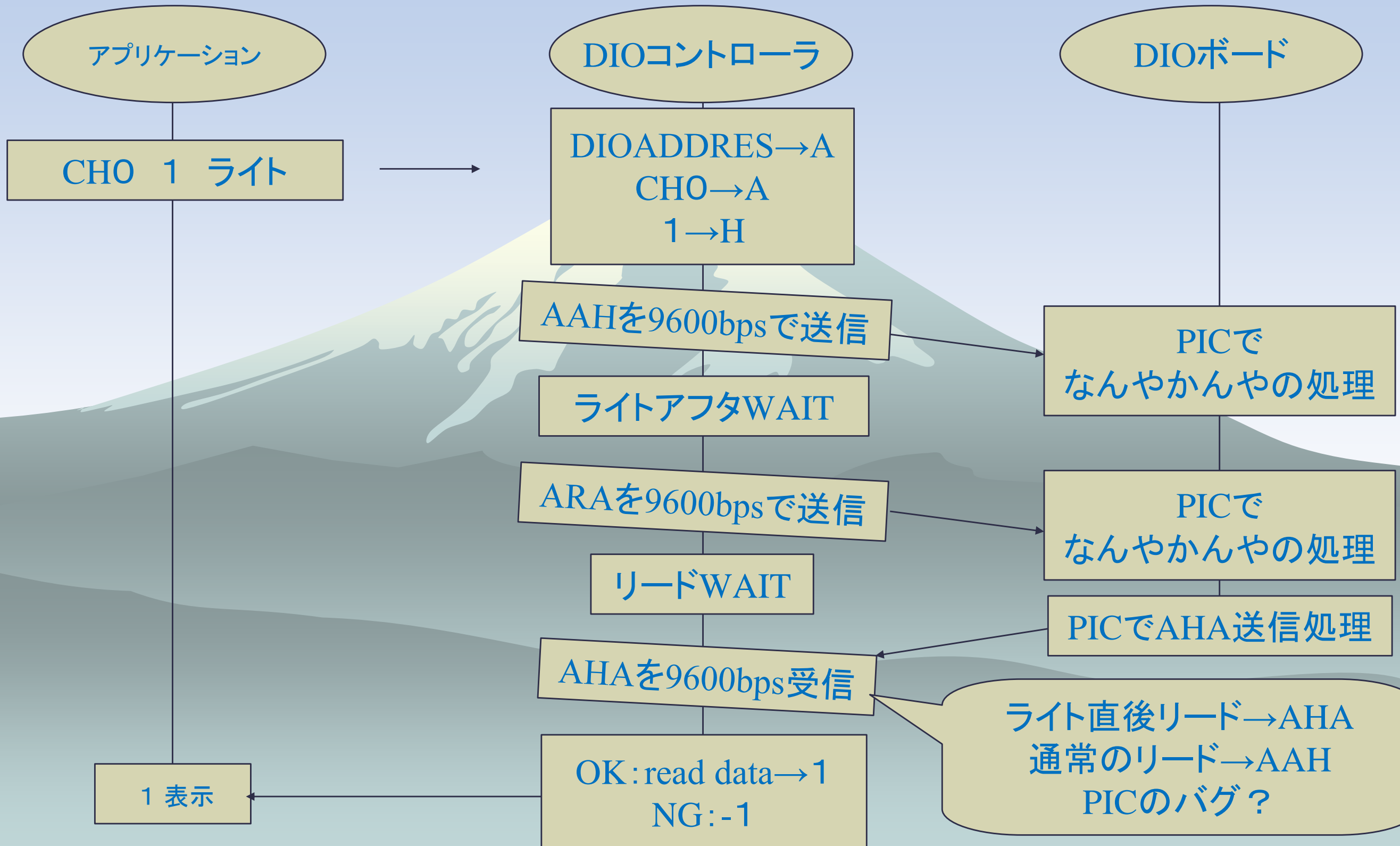
シリアル接続

USB接続

リモートハンド

- トウが檻を破って脱走したらパトライトを点灯させ電話通知
- うさぎが人参を食べたら、人参を補充しMailで通知
- Serverのping TimeOutをMailと電話で同時通知
- うさぎやトラをどうやって認識させる？

シリアル経由だと結構ややこしい



ところでRaspberry Pi

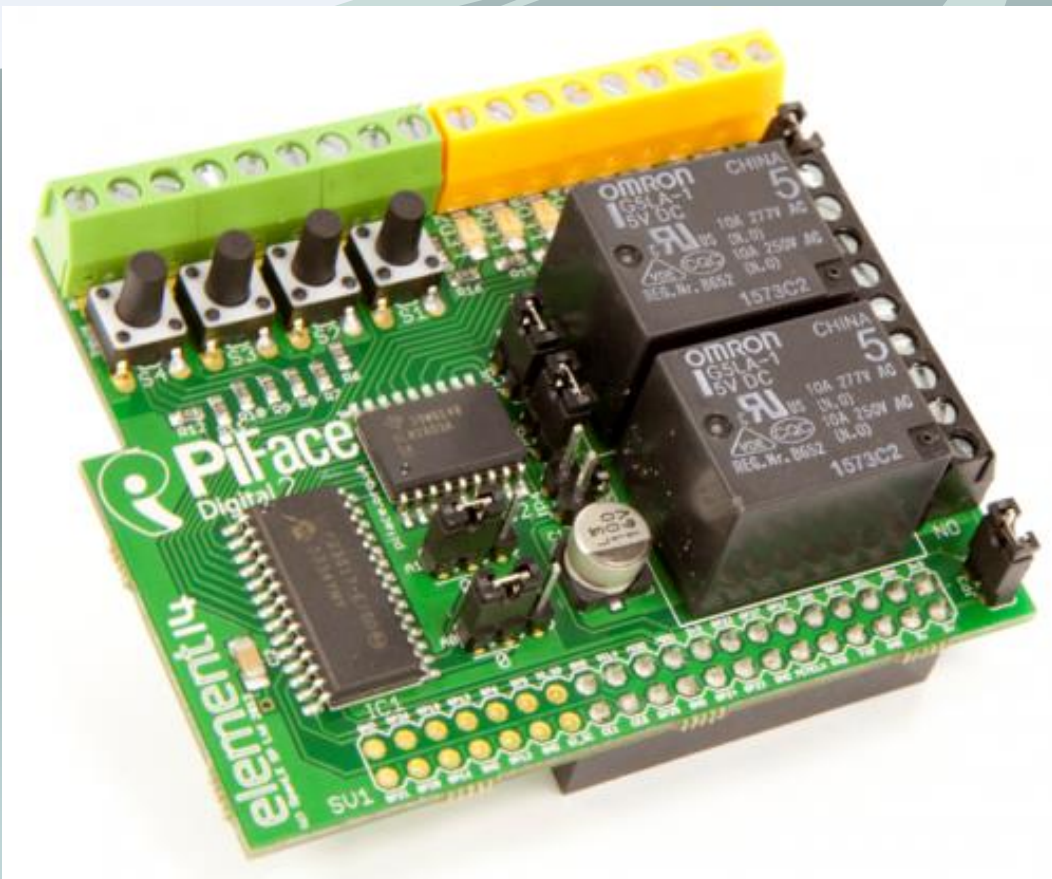
◆ raspberry piとは・・・

Vmplayer + PepoLinux +
シリアルDIO代替



Raspberry Piの拡張ボードPiface

- ◆ PifaceとはデジタルI/O拡張ボード Raspberry Pi用の入出力8ビットの拡張ボード
- ◆ 5Vリレー×2の出力でAC100V/DC24V5A迄外部スイッチング



Raspberry Pi + Piface = Remote-Hand



Raspberry Pi and Case : £29.99

PIFACE : £19.99

Samsung 8GB Debian1 : £8.33

USB to Micro USB Cable 2m :
£1.66

Royal Mail Airmail : £10.98

Total : £75.94

発注後1W 2013年2月末入手
OS当初Pidora、現在Raspbian

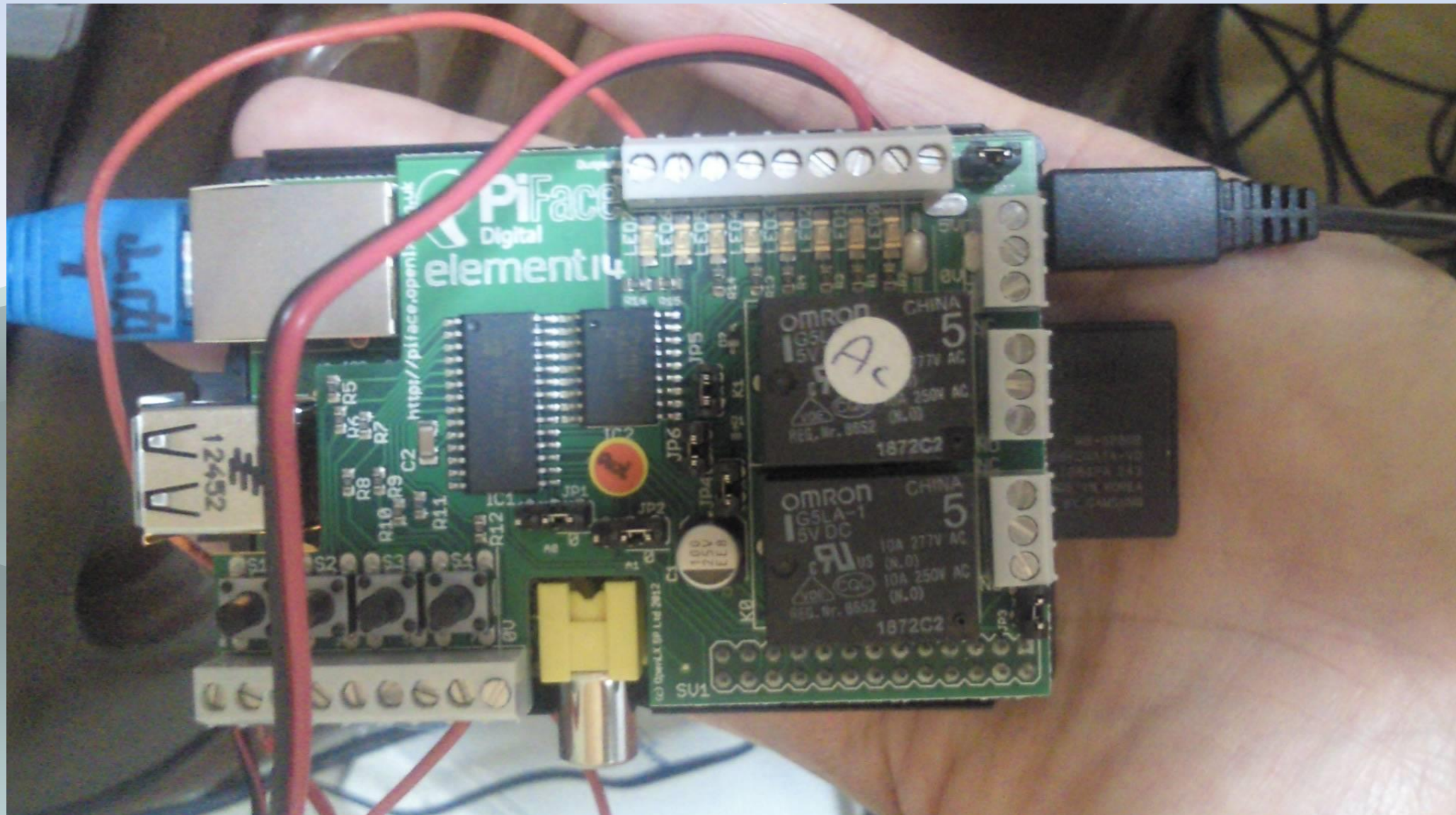
2014年のXmasプ
レゼント



DIOコアプログラム

- pepopiface : DIOの読み書きをマルチユーザー動作
- pepopifaced : DIOから1秒毎に入力信号の変化に合わせたアクションを実行
- pepogmail4dio : gmail経由でDIOの入出力操作と情報を返信
- pepogmail4pic : gmail経由でUSB接続カメラの動画を数秒記録して添付返信

pepopiface(≒piface(Gpio))を制御



pepopiface(≒piface(Gpio))を制御

```
** Welcome to pepopiface Version-0.6 Copyright  
Yamauchi.Isamu compiled:Sep 20 2015 **
```

```
usage:pepopiface port:0-8 [0|1] [timer:0-300000ms]
```

```
# pepopiface 0 1 → 1 port0へwrite
```

```
# pepopiface 0 1 1500 → port0へ1.5秒 1 write
```

```
# pepopiface 0 → 1
```

```
# pepopiface 8 → 00ff 上位8:出力・下位8:入力
```


pepopifaceはlibpifaceを使うが難あり pfio.cへpatch

pepopifaceコマンドはpiface(Gpio)を制御する
libpifaceに難あり

pfio_init()を最初にcallが必要、しかし毎回出力ポートが初期化

patchを作成 `../piface/c/src/piface/pfio.c`

```
// spi_write(GPIOA, 0x00); // turn on port A
spi_write(IODIRA, 0); // set port A as an output
spi_write(IODIRB, 0xFF); // set port B as an input
spi_write(GPPUB, 0xFF); // turn on port B pullups

// initialise all outputs to 0
// int i;
// for (i = 1; i <= 8; i++)
//     pfio_digital_write(i, 0);
//
```

libpifaceを少し

<https://docs.google.com/folder/d/0B-UAZ9CyJCLGQjJ3RDlqa2pqaDg/edit?pli=1>

`pfio_digital_read(pin_number)`

Returns the binary value of the specified input pin.

`pfio_digital_write(pin_number, value)`

Sets the binary value of the specified output pin to the given value.

`pfio_read_input()`

Returns the input port information as a list. Index 2 contains the port data (binary status of the input pins).

`pfio_read_output()`

Returns the output port information as a list. Index 2 contains the port data (binary status of the input pins).

pifaceを正しい込むと不具合が

- ◆ マルチプロセス環境ではpifaceで競合
- ◆ 点けたはずの灯りが点かない
- ◆ 消したはずの灯りが消えてない
- ◆ 排他制御が必要

排他制御セマフォ

- ◆ semaphoreセマフォはプロセス間でリソースの排他制御
- ◆ Raspbian Linuxとpepopifaceで実装
- ◆ セマフォはflagの加算/減算処理
- ◆ flagがマイナスでロック
- ◆ flagがプラスでアンロック
- ◆ 出典:
https://ja.osdn.net/projects/pepolinux/wiki/chobit_prog%20semaphore

排他制御セマフォ

```
void mysem_lock(int sid){
    struct sembuf mysemop[1];
    mysemop[0].sem_num = 0;
    mysemop[0].sem_op =
        LOCK;
    mysemop[0].sem_flg =
        SEM_UNDO;
    if(semop(sid, mysemop, 1)
        == -1){
        perror("semop: semop
        lock-1 failed");
        exit(1);
    }
}
```

LOCK

```
void mysem_unlock(int sid){
    struct sembuf mysemop[1];
    mysemop[0].sem_num = 0;
    mysemop[0].sem_op =
        UNLOCK;
    mysemop[0].sem_flg =
        SEM_UNDO;
    if(semop(sid, mysemop, 1)
        == -1){
        perror("semop: semop
        unlock failed");
        exit(1);
    }
}
```

UNLOCK

pepopifacedはdaemon

- ◆ 定周期でpepopiface 8を投げ入力8ポートを監視
- ◆ たとえば入力ポート0の変化によりコマンドを起動

High→Low : dio0high2lowコマンドを実行

Low→High : dio0low2highコマンドを実行

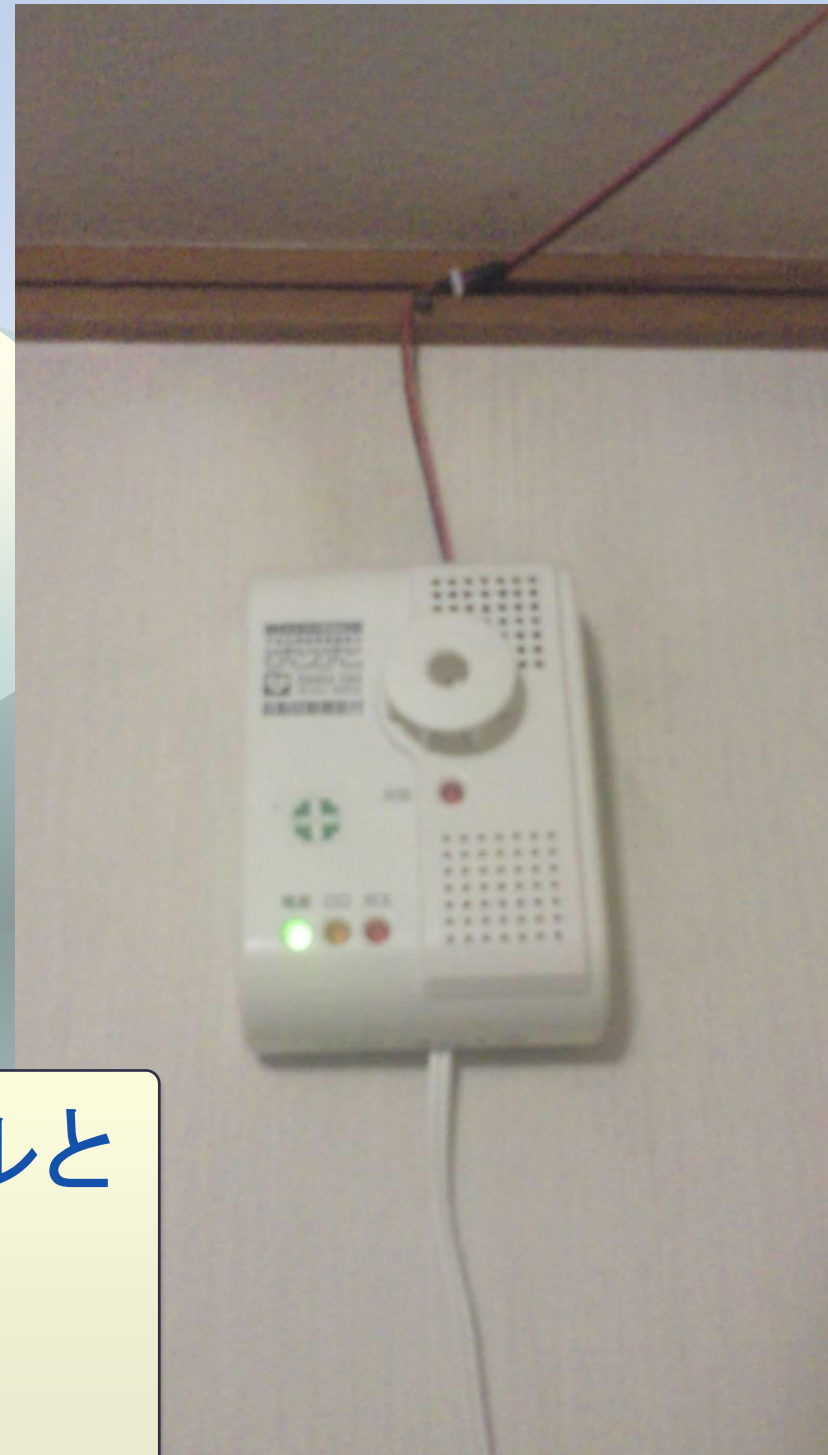
Management DI(Digital Input) -1

Setting first action to the digital input

火事です	: high	Action:low→high	NONE	Timer:	ms	none	Log display
Phone:		Email:		回数:	0	2015/05/24 18:00:09 ~	
防犯センサー	: high	Action:low→high	NONE	Timer:	ms	none	Log display
Phone:		Email:		回数:	1542	2015/05/24 18:00:10 ~ 2015/07/25 10:47:50	
出力1と接続	: high	Action:low→high	入力3と接続high	Timer:	ms	none	Log display
Phone:		Email:		回数:	8	2015/07/03 05:02:16 ~ 2015/07/06 17:42:16	
Input4	: high	Action:low→high	出力2high	Timer:	ms	none	Log display
Phone:		Email:		回数:	-	-	
Input5	: high	Action:low→high	窓の電気high	Timer:	ms	none	Log display
Phone:		Email:		回数:	-	-	
Input6	: high	Action:low→high	窓の電気low	Timer:	ms	none	Log display
Phone:		Email:		回数:	-	-	
Input7	: high	Action:low→high	Output4high	Timer:	ms	none	Log display
Phone:		Email:		回数:	-	-	
			Output4low	Timer:	ms	none	Log display
			散水ポンプhigh	Timer:	ms	none	Log display
			散水ポンプlow	Timer:	ms	none	Log display
			ななちゃん電気high	Timer:	ms	none	Log display
			ななちゃん電気low	Timer:	ms	none	Log display
			扇風機high	Timer:	ms	none	Log display
			扇風機low	Timer:	ms	none	Log display
			玄関あかりhigh	Timer:	ms	none	Log display
			玄関あかりlow	Timer:	ms	none	Log display
			あかりsend	Timer:	ms	none	Log display
			クーラー運転send	Timer:	ms	none	Log display
			クーラー停止send	Timer:	ms	none	Log display

防災システム

大阪ガス・ピコピコ火災報知器（無電圧A接点）



火災報知器が発報するとメールと
電話で通知

防犯システム

人感センサーが働くと動画
メールと電話で通知



人感センサー
無線受信器



人感センサー
無線送信器



センサーライト
+ 無線送信器

動物監視(ペット)



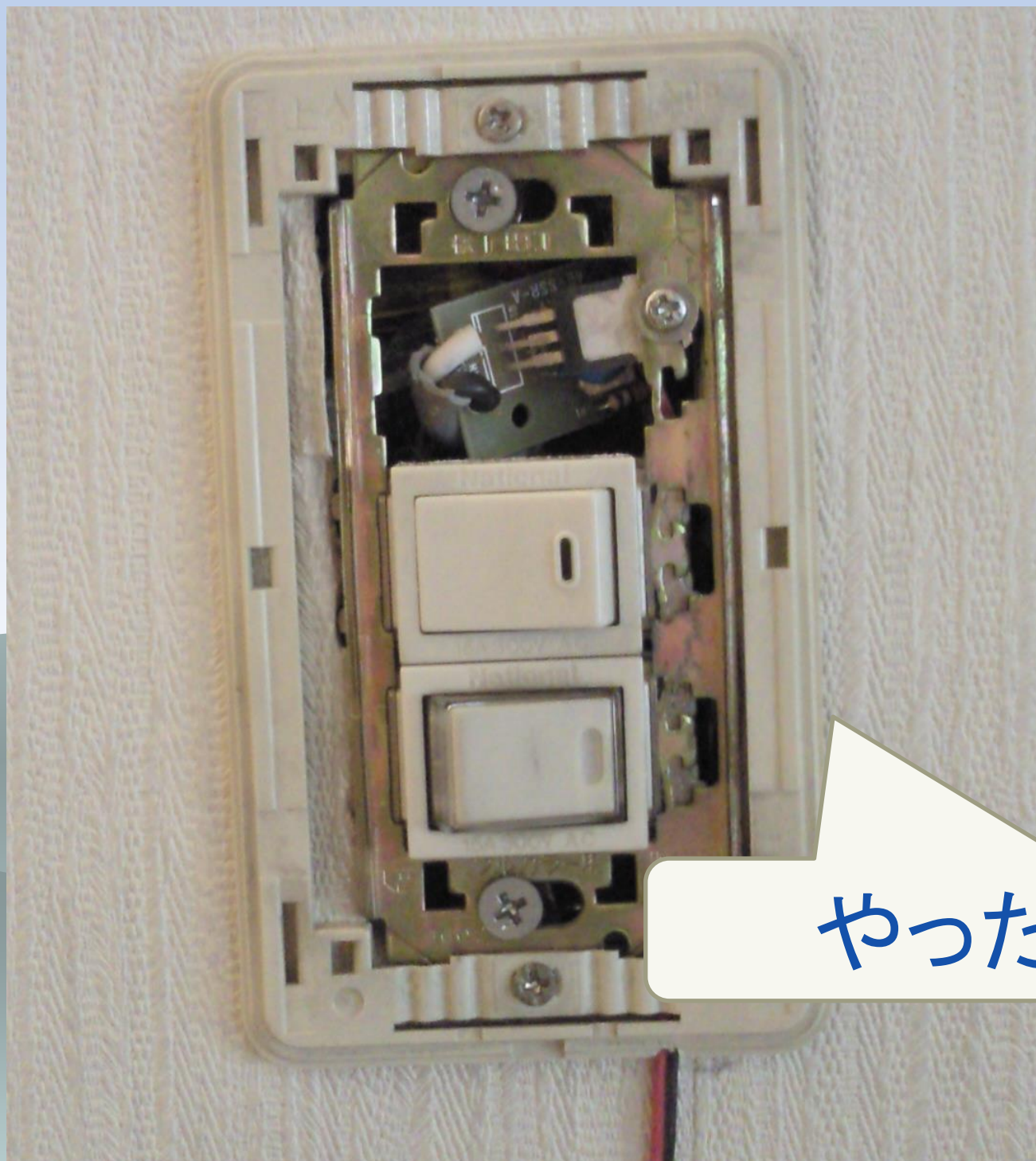
Gmail経由カメラで動画
子供達の顔を確認

ソリッドステートスイッチ AC100Vコンセント



DIO出力0-5VでAC100V/5Aをオフ/オンさせるソ
リッドステートスイッチ内蔵コンセントBOX2,000円
程度

壁スイッチ改造



やったらだめよ！

実践IoTハウス Home Electronics & Security

Remote-Hand raspberry_pi ver:0.21.2016.11.9

Setting automated process

URL:

Event 1 Start * 00-59 * 00-23 * 1-31 * 1-12 * *:Every time

Folder name Podcasts Entry

id	Trigger	Action	Delay	Repeat
1	CPU_Temphigh	CPUファンhigh	1800(ms)	* min
2	屋外照度low	窓の電気high		* min
3	屋外照度low	窓の電気low	3000	
4	屋外照度high	窓の電気low		ms
5	屋外照度low	ななちゃん電灯low	3000(ms)	
6	屋外照度high	ななちゃん電灯low		ms *
7	屋外照度low	玄関電灯low	3000(ms)	* *
8	屋外照度low	玄関電灯low	3000(ms)	* *
9	屋外照度high	玄関電灯low		ms *
10	Enable	ななろく天井電灯		ms 30
11	Enable	ななろく天井電灯	3000(ms)	02
12	USB_Temphigh	扇風機high	9000(ms)	* min
13	I2C_Humlow	散水ポンプhigh	6000(ms)	10 min * 20 hour * * Day * Month * none
14	Enable	none		ms * min * * hour * * Day * Month * none

Callout 1: 定時処理
podcastを定時取得
デバイスに保存

Callout 2: 定時処理
定時に電灯や散水ポンプを
オン/オフ
照度・温度・湿度の条件で
オン/オフ

家庭菜園の自動散水は湿度の条件 オン・オフ



Setting TWE-LITE Wireless DI & AI

TI1	low	散水スイッチ	none	▼
TI2	high	Input10	none	▼
TI3	high	Input11	none	▼
AI1	high	1568 屋外照度	Slice 1300	none ▼ Analog input-1 Graph
AI2	high	4080 Input14	Slice 3000	none ▼ Analog input-2 Graph
AI3	low	4080 Input15	Slice 4500	none ▼ Analog input-3 Graph
AI4	high	4080 Input16	Slice 2000	none ▼ Analog input-4 Graph

CPU_Temperature	low	45.4°C	Slice 45	none ▼	CPU Temperature Graph
USB_Temperature	low	20.8°C	Slice 25	none ▼	USB Temperature Graph
I2C_Temperature	low	15.5°C	Slice 30	none ▼	I2C Temperature Graph
I2C_Humidity	high	92.1%	Slice 90	none ▼	I2C Humidity Graph

2	屋外照度low	▼	窓の電気high	▼	3000(ms	*	min 5	▼	18-2hour	*
3	屋外照度low	▼	窓の電気low	▼	3000(ms	*	min 5	▼	00-0hour	*
4	屋外照度high	▼	窓の電気low	▼	ms	*	min 5	▼	05-0hour	*
5	屋外照度low	▼	ななちゃん電灯low	▼	3000(ms	*	min 5	▼	00-0hour	*
6	屋外照度high	▼	ななちゃん電灯low	▼	ms	*	min 5	▼	05-0hour	*
7	屋外照度low	▼	玄関電灯low	▼	3000(ms	*	min 5	▼	17-2hour	*
8	屋外照度low	▼	玄関電灯low	▼	3000(ms	*	min 5	▼	00-0hour	*
9	屋外照度high	▼	玄関電灯low	▼	ms	*	min 5	▼	05-0hour	*
10	Enable	▼	ななろく天井電灯	▼	ms	30	min *	▼	17	hour
11	Enable	▼	ななろく天井電灯	▼	3000(ms	02	min *	▼	00	hour
12	USB_Temphigh	▼	扇風機high	▼	9000(ms	*	min 2	▼	*	hour
13	I2C_Humlow	▼	散水ポンプhigh	▼	6000(ms	10	min *	▼	20	hour

玄関電灯は簡易照度計(100均電卓)の条件でオン・オフ



Setting TWE-LITE Wireless DI & AI

TI1	low	散水スイッチ	none	▼	
TI2	high	Input10	none	▼	
TI3	high	Input11	none	▼	
AI1	high	1568 屋外照度	Slice 1300	none ▼	Analog input-1 Graph
AI2	high	4080 Input14	Slice 3000	none ▼	Analog input-2 Graph
AI3	low	4080 Input15	Slice 4500	none ▼	Analog input-3 Graph
AI4	high	4080 Input16	Slice 2000	none ▼	Analog input-4 Graph

CPU_Temperature	low	45.4°C	Slice 45	none ▼	CPU Temperature Graph
USB_Temperature	low	20.8°C	Slice 25	none ▼	USB Temperature Graph
I2C_Temperature	low	15.5°C	Slice 30	none ▼	I2C Temperature Graph
I2C_Humidity	high	92.1%	Slice 90	none ▼	I2C Humidity Graph

Gmail経由で制御 ユーザーとキーワード登録

Remote-Hand raspberry_pi ver:0.17 2015.7.24

Ethernet Setting Routing Setting modem Setting Web Setting DIO Setting ping_DO Setting ping_mail Setting ping_tel Setting DIO Control1 DIO Control2 Mail Setting Auto Process Server Control

Setting system Email

Setting operation in Gmail

Gmail User:

Gmail Password:

Mail Address: Allow Email address

Key Word: Subject(keyword)

Mail Check Timer: New Email check interval(Sec)

Jitter: EmailArrival correction(Sec)

Gmail経由でコマンド
pifaceの1番を
5秒間オンにする
Subject: dio2 1 1 5000

実践IoTハウス

Gmail経由監視・とらが脱走したらメール

Remote-Hand raspberry_pi ver:0.17 2015.7.24

Ethernet Setting Routing Setting modem Setting Web Setting DIO Setting ping_DO Setting ping_mail Setting ping_tel Setting DIO Control1 DIO Control2 Mail Setting Auto Process Server Control

Management DI(Digital Input)-2

Setting second action to the digital input

火事です : high Action:low→high NONE Timer: ms none
Phone: Email: 回数: 0 2015/05/24 18:00:09 ~

防犯センサー : high Action:low→high NONE Timer: ms none
Phone: Email: 1673 2015/05/24 18:00:10 ~ 2015/07/27 14:07:07

出力1と接続 : high Action:low→high Timer: ms high
Phone: Email: 16 2015/07/0

Input high Timer: ms
Phone: Email: -

Input high Timer: ms
Phone: Email: -

Input high Timer: ms
Phone: Email: -

Input high Timer: ms
Phone: Email: -

Input high NONE Timer: ms
Phone: Email: Count: -

2階 Input high NONE Timer: ms
Phone: Email: Count: -

Input high NONE Timer: ms
Phone: Email: Count: -



Gmail経由で制御 キーワードでコマンドリスト送付

==input list==

1:火事です=high
2:Input2=high
3:Input3=high
4:Input4=high
5:Input5=high
6:Input6=high
7:Input7=high
8:Input8=high 9:散水スイッチ=low 10:Input10=high
11:Input11=high 13:屋外照度=high 14:Input14=high
15:Input15=low
16:Input16=high
17:CPU_Temp=low
18:USB_Temp=none
19:I2C_Temp=low
20:I2C_Hum=low

==output list==

1:CPUファン=low 2:火事テスト=low 3:窓の電気=low
4:Output4=low
5:Output5=low 6:ななちゃん電灯=low 7:扇風機=low
8:Output8=low 9:ななろく天井電灯 10:クーラー運転
11:クーラー停止
12:Output12
13:Output13
14:Output14 15:散水ポンプ=low 16:Output16=none
17:玄関電灯=low

==Other Command==

18:CPU Temperature
19:USB Temperature
20:No1 Camera Movie
21:No1 Camera Picture
22:No2 Camera Movie
23:No2 Camera Picture
24:No3 Camera Movie
25:No3 Camera Picture
26:Module Camera Movie
27:Module Camera Picture
28:I2C
Temperature&Humidity
29:CPU
Temperature&Graph
30:USB
Temperature&Graph
31:I2C
Temperature&Graph
32:I2C Humidity&Graph

Raspberry Pi + Webカメラ + gmailで動画

- ◆ Raspberry Piは非力なマシン 320 × 240 1pps迄
- ◆ `avconv -f video4linux2 -s 320x240 -r 1 -i /dev/vide0 -t 10 y remote-hand.mp4`
- ◆ USB接続の安価なWebカメラはavconvが時々Hung Up
- ◆ Hung Upでも次のコマンドで再開させる
Script→pepomp4ctl
- ◆ これでRaspberry PiでRemote-Handが完成！

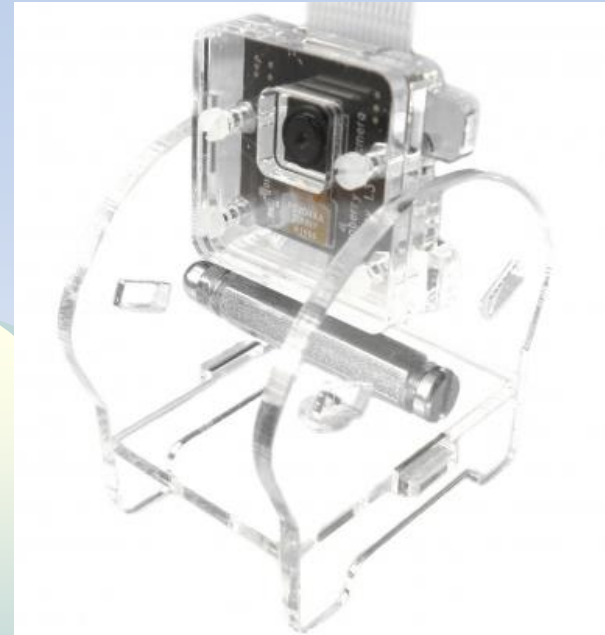
Raspberry Pi 2は強力

Webカメラ 320 × 240 10ppsでも余裕



やっぱりRaspberry Piはモジュールカメラ

```
raspistill -t 1000 -vf -hf -w 2592 -h 1944 -o ${IMAGE}
```



高画質な写真や
動画が撮れます

IRKit

IRKitとはネットワーク対応学習型赤外線リモコン



IRKitとコラボ

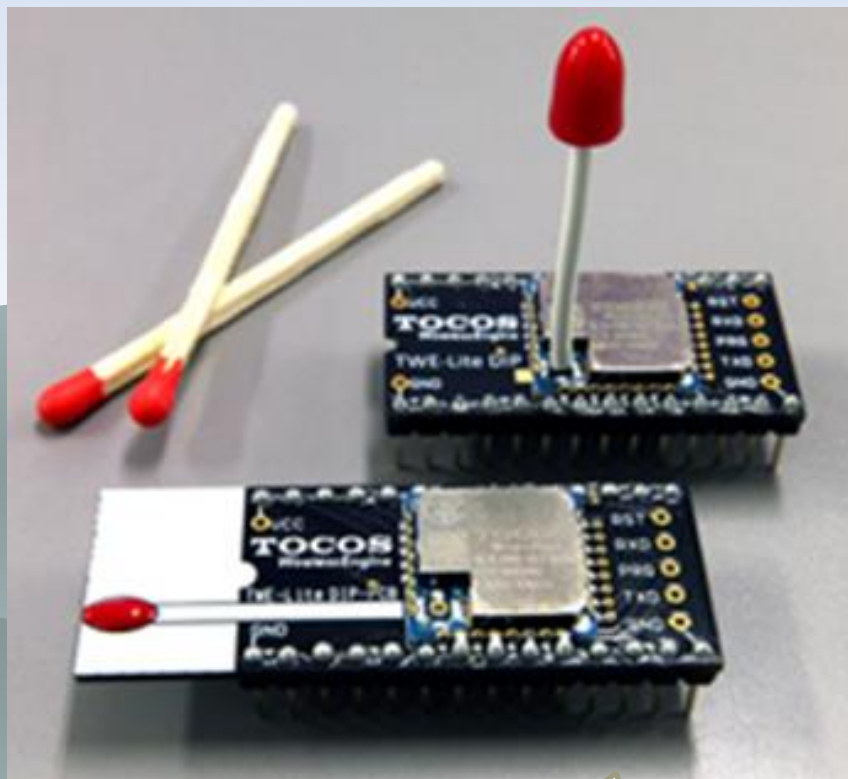
IRKitとはネットワーク対応学習型赤外線リモコン

Raspberry piでIRデータ登録・送信

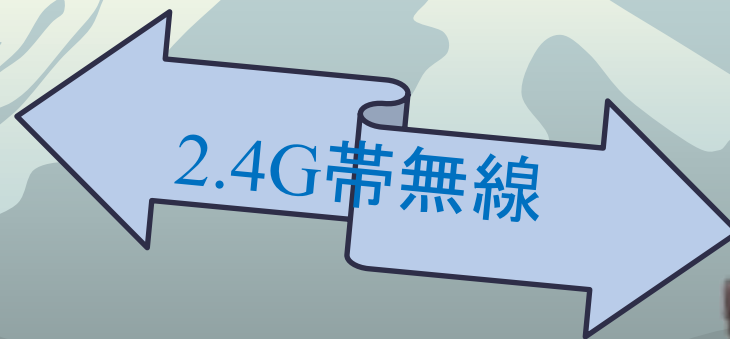


Tocos無線モジュール

- Tocosからmono-wirelessに変わったけど
- 簡単にすぐに使える無線モジュール
- TWE-Lite DIP (トワイライト・ディップ) は AI・PWM・I2C・DIO
- 当初DIO (デジタルイン・アウト) のみ使用
- 現在はI2C温湿度センサーAM2321やアナログ入力も使用



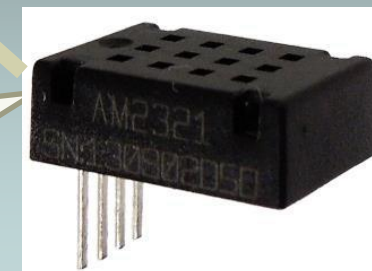
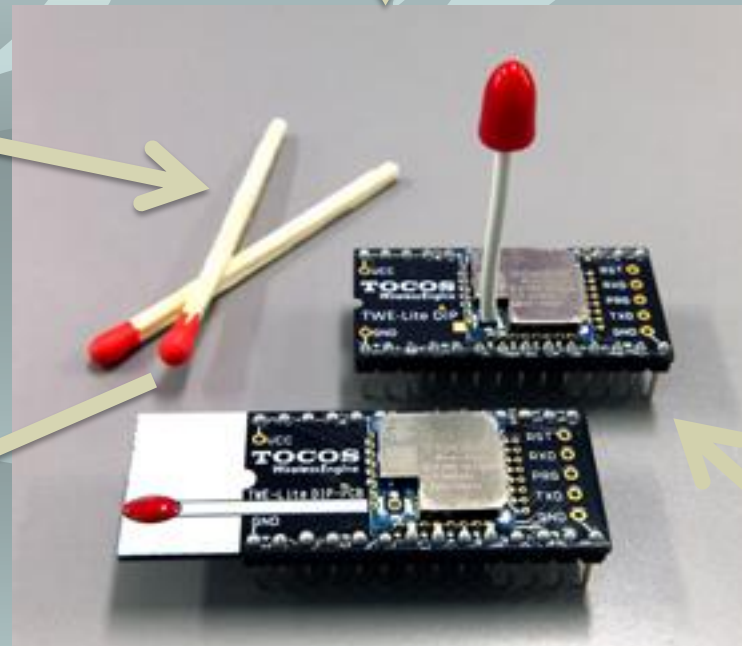
TWE-Lite DIP



USB Dongle
ToCoStick

Tocos無線モジュール

wirelessでDIOが使える
2階の電気・火災報知器や
庭の自動散水が無線で制御



AM2315温度湿度センサー
小指の爪サイズ

Tocos無線モジュールを制御

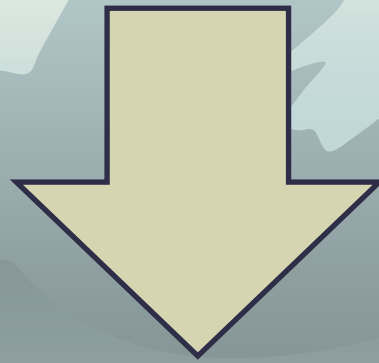
- USB dongle ToCoStickはシリアル制御
- シリアル制御コマンドは稚拙のepiconで
- `/usr/local/bin/epicon -s 115200 -d 5 -ql /dev/ttyUSB0 -c ${CMD}`
- 無線は水もの? ...時々不通状態→リトライ
- `:7881150175810000380026C9000C04220000FFFFFFFFFFFFFFA7`なデータが1秒毎に定期受信→どういうアプリを組むの?
- DOの状態確認が出来ない→動作モードを変更(定期送信を行わない)、DI-DOを1ビット削りループ接続、確認用へ
- 送信時にチェックSUMを付加、安直なscriptでは計算2秒程→Cで...
TWE-Lite Ver1.6からは送信は自動付加・受信のみチェック
- `pepotocosctl`コマンド完成

Tocos無線モジュールを制御

- # epicon -ql /dev/ttyUSB0 -s 115200
- + + + ←エコー無し
- --- CONFIG/TOCOS TWELITE DIP APP V1-06-16/SID=0x81010b68/LID=0x00 ---
- a: set Application ID (0x67720102)
- i: set Device ID (121=0x79)
- c: set Channels (18)
- x: set Tx Power (03)
- t: set mode4 sleep dur (1000ms)
- y: set mode7 sleep dur (10s)
- f: set mode3 fps (32)
- z: set PWM HZ (1000)
- o: set Option Bits (0x00000020)
- b: set UART baud (38400)
- p: set UART parity (N)
- ---
- S: save Configuration
- R: reset to Defaults

Tocos無線モジュールを制御

- 無線は水もの？・・・時々不通でデジタル出力がraspberrypi側と不一致
- 点けた筈の電灯が点かない
- 消した筈の電灯が点きっぱなし
- 2.4G帯が混雑？



- 2.4G帯3チャンネルを1CHに固定→混雑緩和
- 出力3ビット1回のコマンドで終わらせる→混雑緩和
- ポーリングタイマーを2秒→4秒→混雑緩和
- raspberrypi側でDOのオンオフ情報を保持させて、数秒間隔で同期させるdaemonで改善→無限リトライ

Tocos無線モジュールを制御

- ToCoStickとraspberrypiはシリアル通信、epiconを使う
- `/usr/local/bin/epicon -s 115200 -ql /dev/ttyUSB0 -c command_file`
- コマンドファイル
- `#!/bin/sh`
- `echo -en ":7888AA015C0000X"¥r¥n`
- `msleep 20`
- `read -s -t 1 RD || RD="-1"`
- `#echo $RD >/dev/stderr`
- `echo -en ":7888AA015C03020004X"¥r¥n`
- `msleep 20`
- `read -s -t 1 RD || RD="-1"`
- `msleep 1500`
- `echo -en ":7888AA025C0006X"¥r¥n`

5CはAM2321の
I2Cアドレス

TOCOS TWE-LiteとToCoStick(トコスティック)で 温湿度計AM2321センサーのデバック中 入力データフォーマット(外部マイコン→TWE)

- 1: 1バイト: 無線デバイスのアドレス (0xDB: 自分自身, 0x00 ~ 0x7F, 0x00:親機, 0x78: 全子機)
- 2: 1バイト: 0x88
- 3: 1バイト: 要求番号(応答メッセージにそのまま出力される識別子)
- 4: 1バイト: コマンド (0x1: 書き込み, 0x2: 読み出し, 0x4: 書き込み & 読み出し)
- 5: 1バイト: I2Cアドレス (1010100 なら 0x54)
- 6: 1バイト: I2Cコマンド (最初のコマンドバイト)
- 7: 1バイト: データサイズ (無い時は 0)
- 8: Nバイト: データ (データサイズが0のときは、本フィールドは無し)
- 9: 1バイト: チェックサム
- :DB88010123230055
- ^1^2^3^4^5^6^7^9

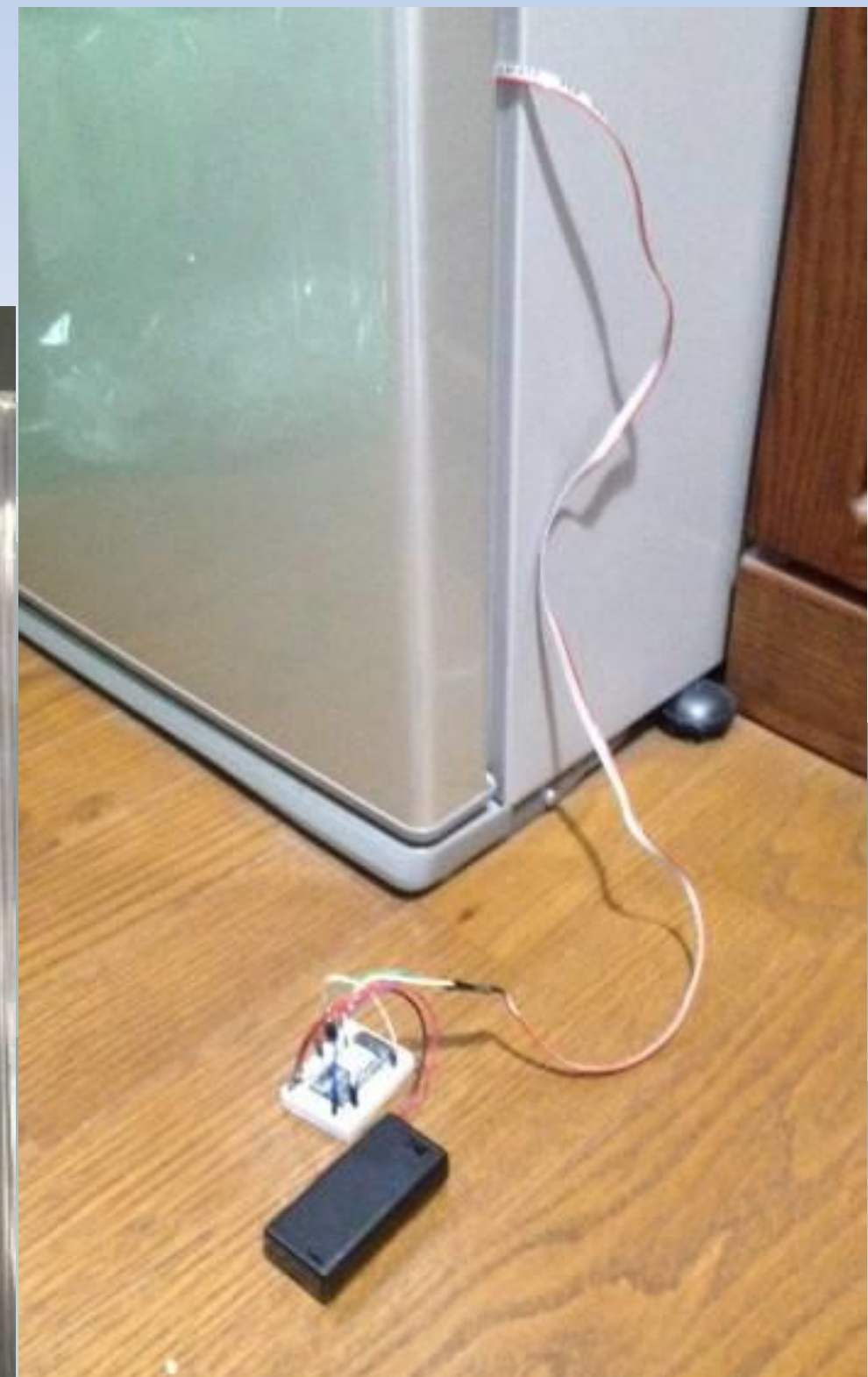
TOCOS TWE-LiteとToCoStick(トコスティック)で 温湿度計AM2321センサーのデバック中 出力データフォーマット(TWE→外部マイコン)

- 1: 1バイト: 無線デバイスのアドレス
- 2: 1バイト: 0x89
- 3: 1バイト: 要求番号
- 4: 1バイト: コマンド (0x1: Write, 0x2: Read, 0x4: Write&Read)
- 5: 1バイト: 0:FAIL, 1:SUCCESS
- 6: 1バイト: データサイズ (無い時は 0)
- 7: Nバイト: データ (データサイズが0のときは、本フィールドは無し)

TOCOS TWE-LiteとToCoStick(トコスティック)で 温湿度計AM2321センサーのデバック中

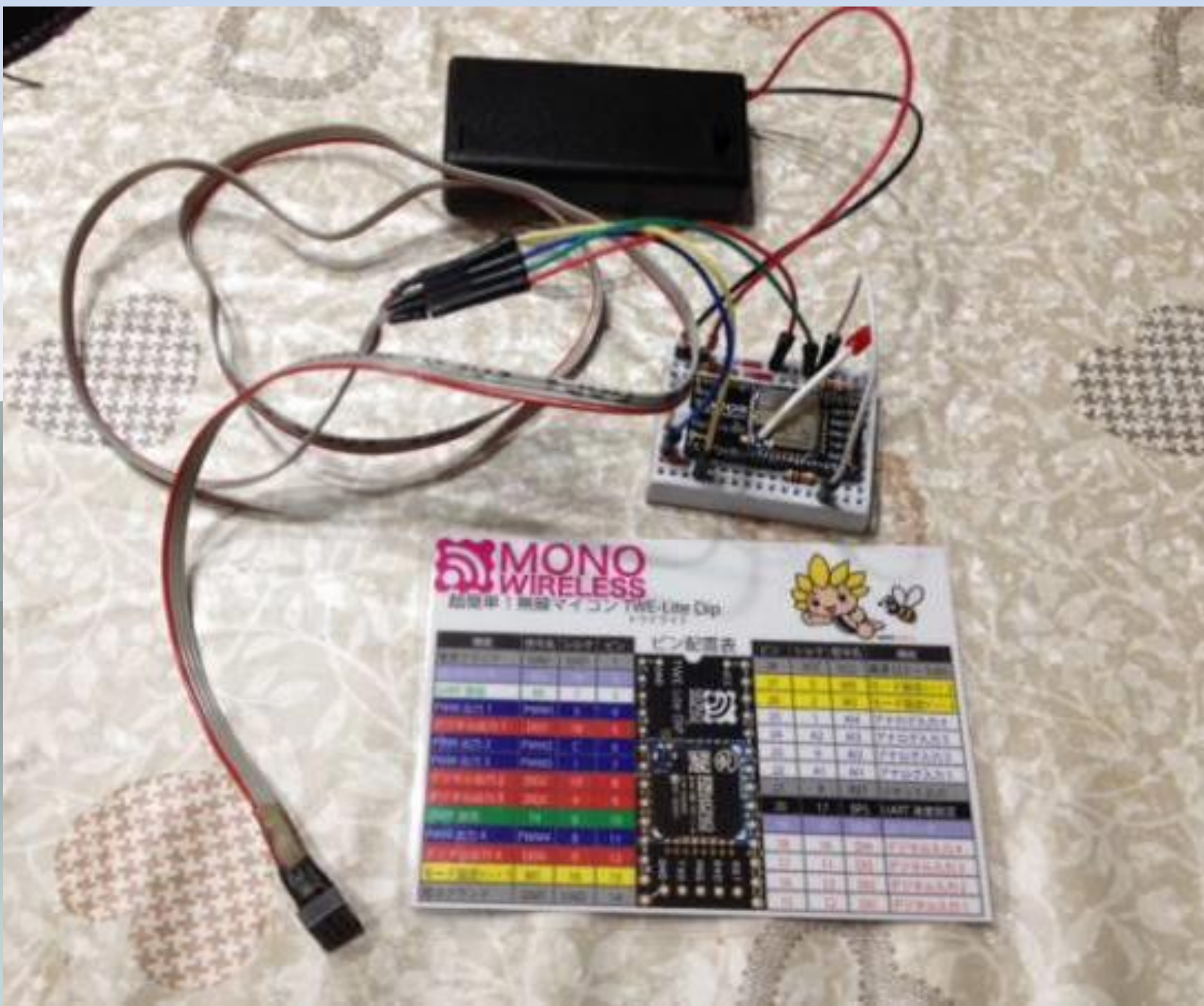
- センサーのWakeUP送信
- :7888AA015C0000X
- 温湿度レジスタから4byte読み取り送信
- :7888AA015C03020004X
- 応答
- :7889AA01000054
- 1500ms経過後に読み取り送信・・・4byteと思って読み取りCountを4にするとダメ
- :7888AA025C00006X
- 応答
- :7889AA0201060304**022A00F0**29
- 何処に温度と湿度があるんやろか
- **022A**湿度 $\Rightarrow 02 \times 256 + 2 \times 16 + 10/10 \Rightarrow 55.4\%$
- **00F0**温度 \Rightarrow 1バイト目80以上でマイナス、2バイト目は下位
- 今回はプラスF $\Rightarrow 15$
- $00 \times 256 + 15 \times 16 + 0/10 = 24.0^{\circ}\text{C}$

TOCOS TWE-LiteとToCoStick(トコスティック)で 温湿度計AM2321センサーのデバック中 冷蔵庫の温度測定に挑戦



TOCOS TWE-LiteとToCoStick(トコスティック)で 温湿度計AM2321センサーのデバック中

TWE-Lite内蔵ワイヤレス制御電源コンセント



TOCOS TWE-LiteとToCoStick(トコスティック)で簡易照度センサー(100均電卓)デバック中

```
[root@tmp]# ./new_pepotocosctl 6
```

```
RD :788115017E8100979C003C67000D291B00085CFFFFFFFC6E
```

AD値の復元には以下のように計算 $AD1[mV] = (5C * 4 + 0) * 4$ FCの下位2ビットが補正值 ADポートは未使用(VCC接続)の場合、対応する e1 ~ e4 値は $0xFF \Rightarrow 1472$



100均電卓をTocosアナログ入力1に接続

TOCOS TWE-LiteとToCoStick(トコスティック)で簡易照度センサー(100均電卓)デバック中

The screenshot shows a web interface for a Raspberry Pi project. The interface is divided into several sections:

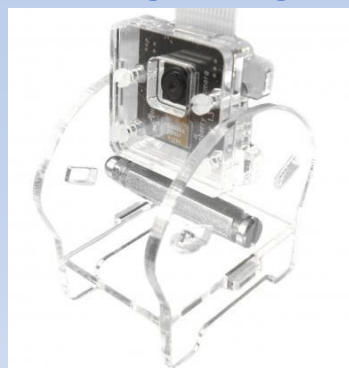
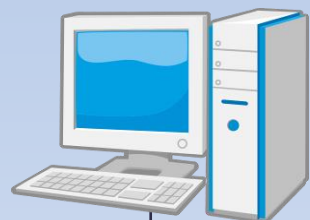
- Input Settings:** A table with 8 rows, each representing an input pin (Input2 to Input8). Each row has a status indicator (all are 'high'), the pin name, and a dropdown menu (all are 'none').
- Setting TWE-LITE Wireless DI & AI:** A table with 4 rows, each representing a sensor (TI1, TI2, TI3, AI1, AI2, AI3, AI4). Each row has a status indicator (TI1, AI1 are 'low'; TI2, TI3, AI2, AI3, AI4 are 'high'), a label (e.g., '散水スイッチ', '屋外照度'), a value, and a dropdown menu (all are 'none').
- Temperature and Humidity Settings:** A table with 4 rows, each representing a sensor (CPU_Temperature, USB_Temperature, I2C_Temperature, I2C_Humidity). Each row has a status indicator (all are 'low'), a label, a value, and a dropdown menu (all are 'none').
- Interface:** A dropdown menu set to 'piface', and buttons for 'Run' and 'Clear'.
- Graph:** A line graph titled 'I2C analog input-1' showing voltage over time. The y-axis ranges from 0.0 to 5.0 k. The x-axis shows time from Mon 12:00 to Tue 12:00. The graph shows a red line that rises to approximately 4.0 k during the day and drops to approximately 1.0 k at night.

Callout boxes provide additional information:

- 昼間100均電卓の太陽電池は2000mv以上4080表示** (During the day, the solar battery of the 100-yen digital multimeter is displayed above 2000mv up to 4080).
- 100均電卓の太陽電池の電圧をグラフ化スライスを決定** (Determine the slice for graphing the voltage of the solar battery of the 100-yen digital multimeter).

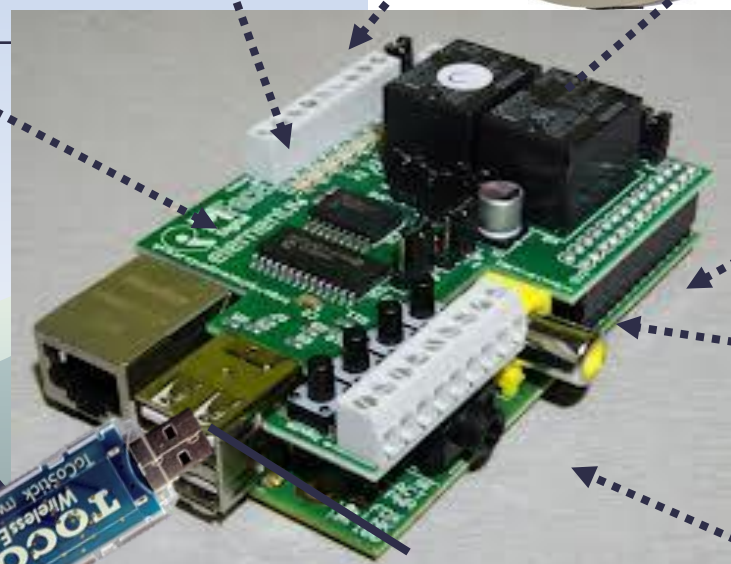
実践IoTハウス

Home Electronics & Security



クーラー

IRKit
学習型赤外線リモコン



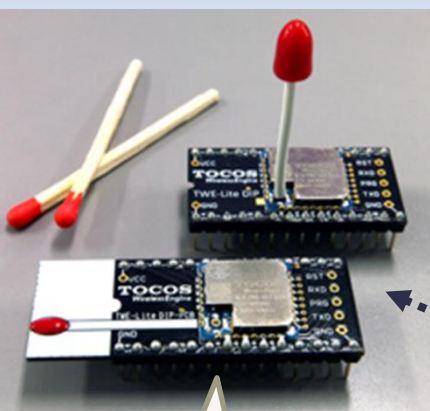
火災報知器発報で動画メール
と電話同時通知

Raspberry Pi + Piface
Gmail連携で動物監視
外部機器の監視制御



リモートハンド

Tocos無線モジュール
離れた物を監視制御



実践IoTハウス

Home Electronics & Security

Management DI(Digital Input) -1

Setting first action to the digital input

火事です : high Action: low→high NONE Timer: ms none Log display

防犯センサー : high Action: low→high NONE Timer: ms none Log display

出力1と接続 : high Action: low→high NONE Timer: ms none Log display

Input4 : high Action: low→high NONE Timer: ms none Log display

Update Log

- Update 2015/07/25 10:47:50 #1542
- Update 2015/07/25 10:47:06 #1541
- Update 2015/07/25 10:46:50 #1540
- Update 2015/07/25 08:39:52 #1529
- Update 2015/07/25 08:39:37 #1528
- Update 2015/07/25 08:38:56 #1527
- Update 2015/07/25 08:38:44 #1526
- Update 2015/07/25 08:38:31 #1525
- Update 2015/07/25 08:37:11 #1524
- Update 2015/07/25 08:36:59 #1523
- Update 2015/07/25 08:36:23 #1522
- Update 2015/07/25 08:36:12 #1521
- Update 2015/07/25 08:36:02 #1520
- Update 2015/07/25 08:35:48 #1519
- Update 2015/07/25 08:35:37 #1518

入力high→low→high
22イベントとログ表示
36アクション×2設定

USB or modules
カメラで動画

温度・湿度・アナログ入力の
グラフとスライスでデジタル化

Setting digital output terminal name

Server-Synchronized at 2016/11/08 15:20:36

Output1	low	none	Timer	ms	CPUファン	none
Output2	low	none	Timer	ms	火事テスト	none
Output3	low	none	Timer	ms	窓の電気	none
Output4	low	none	Timer	ms	Output4	none
Output5	low	none	Timer	ms	Output5	none
Output6	low	none	Timer	ms		
Output7	high	none	Timer	ms		
Output8	low	none	Timer	ms		

入出力端子へ別名設定

Setting IRKit

IR1 none Timer ms なんとなく天井電灯 IR_data Real

IR2 none Timer ms クーラー運転 IR_data Real

IRKit赤外線データ
登録&出力

Setting digital input terminal name

Input1	high	火事です	none
Input2	high	Input2	none
Input3	high	Input3	none
Input4	high	Input4	none
Input5	high	Input5	none
Input6	high	Input6	none
Input7	high	Input7	none
Input8	high	Input8	none

Tocos無線モジュール
DIO & AI

Setting TWE-LITE Wireless DI & AI

T11	low	散水スイッチ	none
T12	high	Input10	none
T13	high	Input11	none
AI1	high	1568 屋外照度	Slice 1300 none Analog input-1 Graph
AI2	high	4080 Input14	Slice 3000 none Analog input-2 Graph
AI3	low	4080 Input15	Slice 4500 none Analog input-3 Graph
AI4	high	4080 Input16	Slice 2000 none Analog input-4 Graph

CPU_Temperature	low	45.4°C	Slice 45 none CPU Temperature Graph
USB_Temperature	low	20.8°C	Slice 25 none USB Temperature Graph
I2C_Temperature	low	15.5°C	Slice 30 none I2C Temperature Graph

実践IoTハウス

イベントログを表示

Management DI(Digital Input) -1

Setting first action to the digital input

火事です : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: 0 2015/05/24 18:00:09 ~

防犯センサー : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: 1542 2015/05/24 18:00:10 ~ 2015/07/15 10:47:50

出力1と接続 : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: 8 2015/07/03 05:02:16 ~ 2015/07/06 17:42:16

Input4 : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: -

Input5 : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: -

Input6 : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: -

Input7 : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: -

Input8 : high Action:low→high NONE Timer: ms none Log display
 Phone: Email: 回数: -

2階が火事です

dio1high Update Log - Googl...

about:blank

```

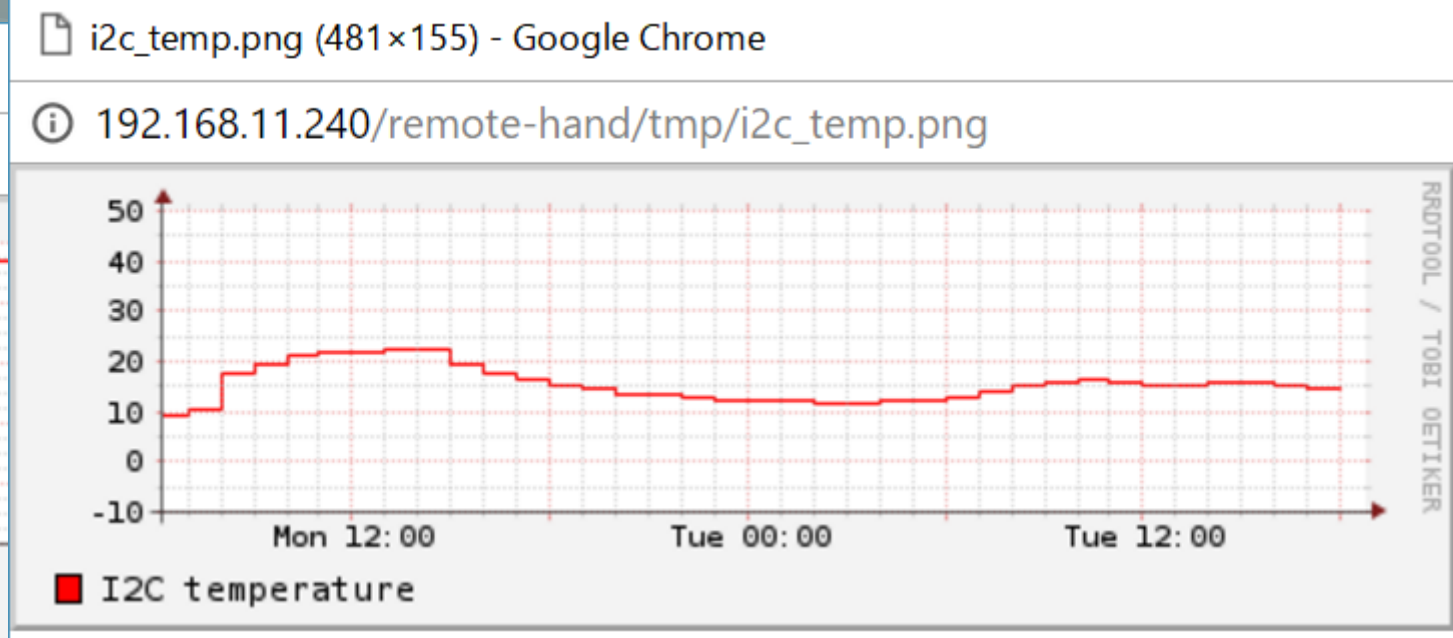
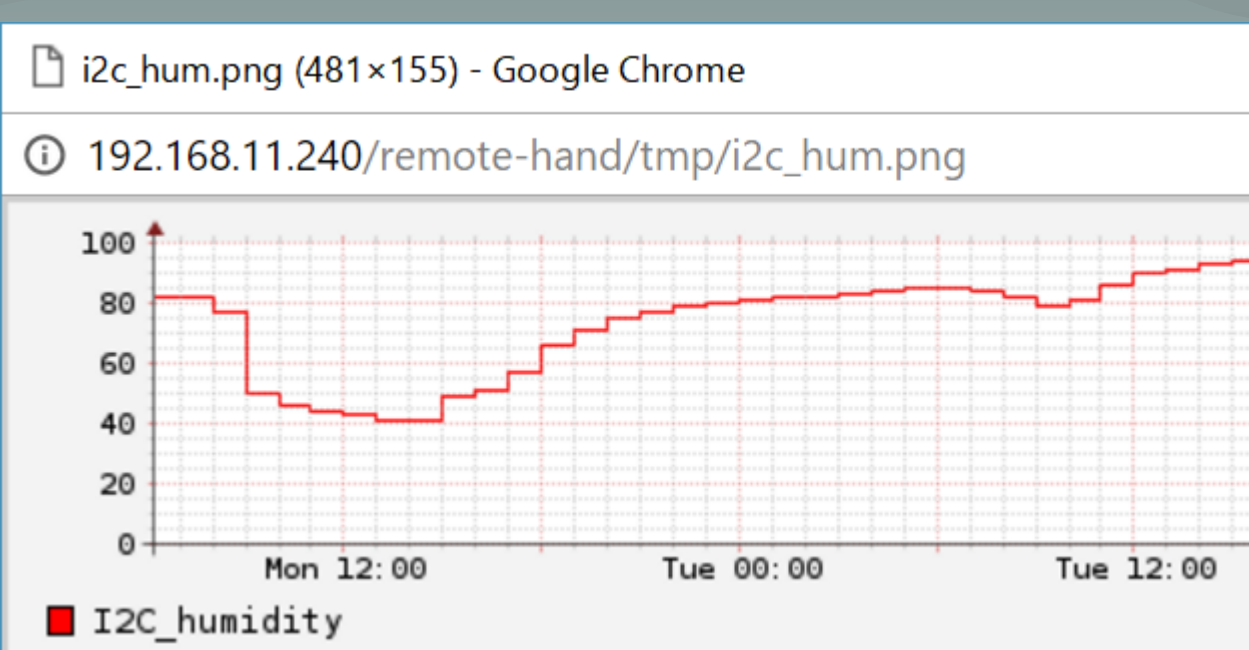
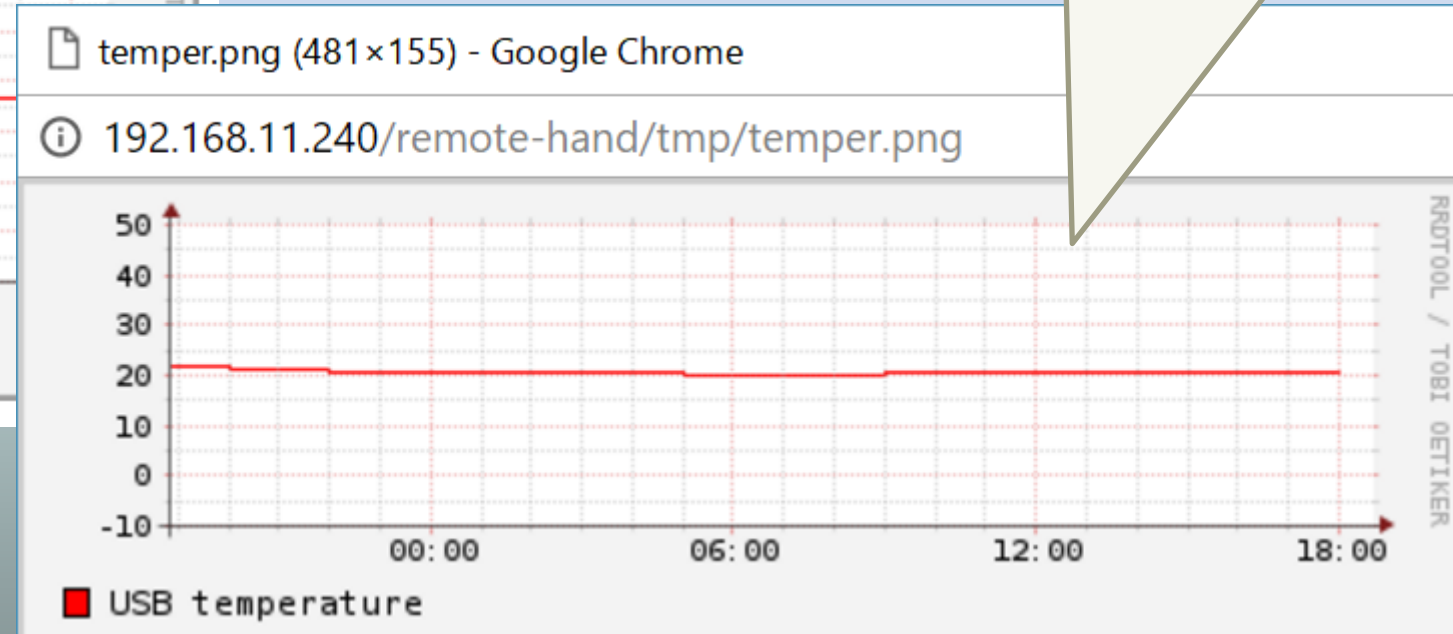
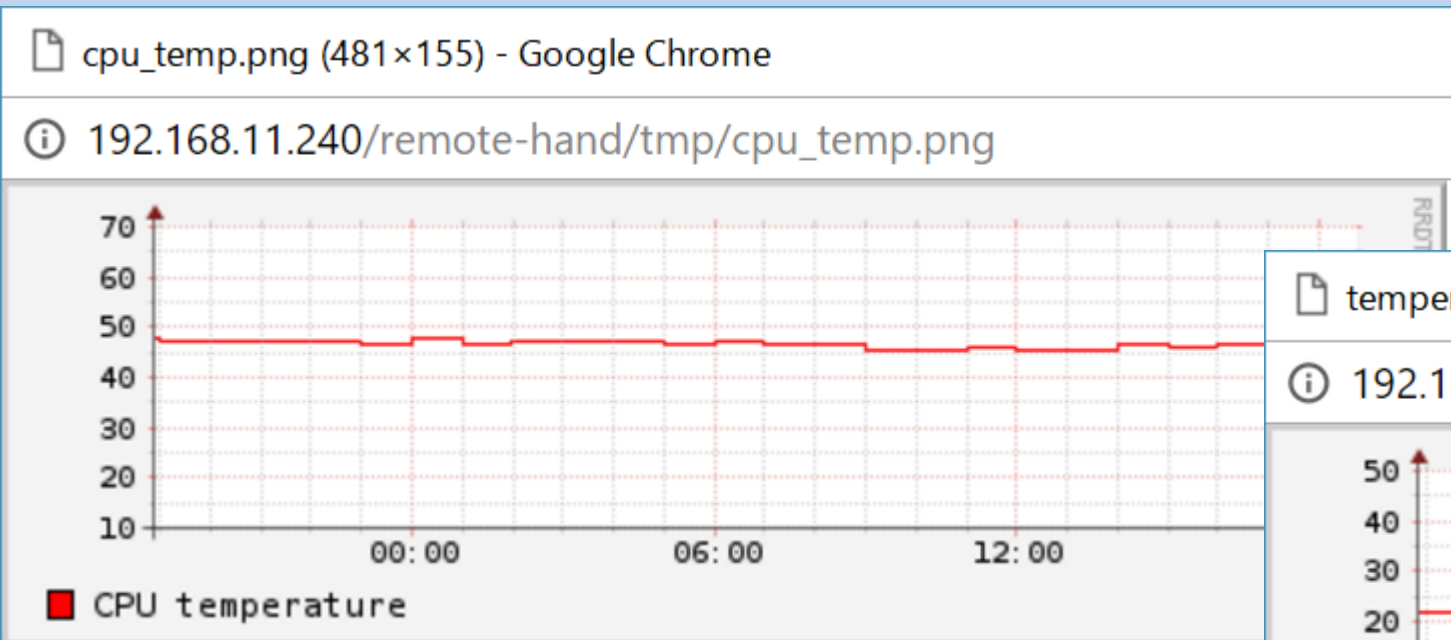
dio1high Update Log
Update 2015/02/22 18:44:05 #1559
Update 2015/02/22 18:43:55 #1558
Update 2015/02/22 18:43:42 #1557
Update 2015/02/22 18:43:30 #1556
Update 2015/02/22 18:43:19 #1555
Update 2015/02/22 18:43:09 #1554
Update 2015/02/22 18:31:12 #1553
Update 2015/02/22 18:31:02 #1552
Update 2015/02/22 18:30:51 #1551
Update 2015/02/22 18:30:40 #1550
Update 2015/02/22 18:29:59 #1549
Update 2015/02/22 14:25:17 #1548
Update 2015/02/22 14:25:05 #1547
Update 2015/02/22 14:24:54 #1546
Update 2015/02/22 14:20:01 #1545
Update 2015/02/22 14:19:40 #1544
Update 2015/02/22 13:28:26 #1543
Update 2015/02/22 13:28:00 #1542
Update 2015/02/22 13:27:44 #1541
Update 2015/02/22 13:14:28 #1540
Update 2015/02/22 13:14:16 #1539
Update 2015/02/22 13:09:02 #1538
Update 2015/02/22 12:54:01 #1537
Update 2015/02/22 12:53:51 #1536
Update 2015/02/22 12:53:30 #1535
Update 2015/02/22 12:52:52 #1534
Update 2015/02/22 12:51:58 #1533
Update 2015/02/22 12:49:07 #1532
Update 2015/02/22 12:47:45 #1531
Update 2015/02/22 12:43:17 #1530
Update 2015/02/22 12:35:53 #1529
Update 2015/02/22 11:24:33 #1528
Update 2015/02/22 11:23:59 #1527
Update 2015/02/22 11:23:23 #1526
Update 2015/02/22 11:22:56 #1525
Update 2015/02/22 11:22:31 #1524
Update 2015/02/22 11:15:31 #1523
  
```

入力毎に2つのイベントhigh、low
のカウンターとログを表示

実践IoTハウス

グラフ表示

RRDtoolでCPU温度
などグラフ表示



スマホde実践IoTハウス

Home Electronics & Security

Remote-Hand raspberry_pi ver:0.31 2016.11.8

Control digital output

lo	CPUファン	High/Low
lo	火事テスト	High/Low
hi	窓の電気	High/Low
lo	Output4	High/Low
lo	Output5	High/Low
hi	ななちゃん	High/Low
hi	扇風機	High/Low
lo	Output8	High/Low
Ready	ななろく天	Send
Ready	クーラー運	Send
Ready	クーラー停	Send
none	Output12	Send
none	Output13	Send
none	Output14	Send
lo	散水ポンプ	High/Low
no	Output16	High/Low
hi	玄関電灯	High/Low

Digital input information

hi	火事です
hi	Input2
hi	Input3
hi	Input4
hi	Input5
hi	Input6
hi	Input7
hi	Input8
lo	散水スイッ
hi	Input10
hi	Input11
lo	屋外照度

Ready	クーラー停	Send
none	Output12	Send
none	Output13	Send
none	Output14	Send
lo	散水ポンプ	High/Low
no	Output16	High/Low
hi	玄関電灯	High/Low

Digital input information

hi	火事です
hi	Input2
hi	Input3
hi	Input4
hi	Input5
hi	Input6
hi	Input7
hi	Input8
lo	散水スイッ
hi	Input10
hi	Input11
lo	屋外照度
hi	Input14
lo	Input15
hi	Input16
lo	CPU_Tempe
lo	USB Tempe
lo	I2C Temper
lo	I2C Humidit

43.3°C CPU Temperature Graph

USB Temperature 18.8°C USB Temperature Graph

I2C Temperatur 11.5°C I2C Temperature Graph

I2C Humidity 90.3% I2C Humidity Graph

Update

Logout

©2016-2018 pepolinux.com 2016/11/11 05:13:52

Javascript+c+shell Script

- ◆ フルスクラッチ
- ◆ cgi bash Script 13000行
- ◆ Helper bash Script 3100行
- ◆ Javascript 3000行
- ◆ c 2000行
- ◆ 気がつけば約20000行余り

Raspberry Piはflashを酷使 webアクセスlogをtmpfsへ

```
[root@tmp]# df
```

```
Filesystem 1K-blocks Used Available Use% Mounted on  
/dev/root 7254872 2724852 4138444 40% /  
devtmpfs 185960 0 185960 0% /dev  
tmpfs 38048 288 37760 1% /run  
tmpfs 5120 0 5120 0% /run/lock  
tmpfs 76080 0 76080 0% /run/shm  
/dev/mmcblk0p1 57214 23118 34096 41% /boot  
tmpfs 102400 3248 99152 4% /www
```

Raspberry Piはflashを酷使 /var/spool, /var/logをtmpfsへ

```
[root@var]# ll
```

```
total 102428
```

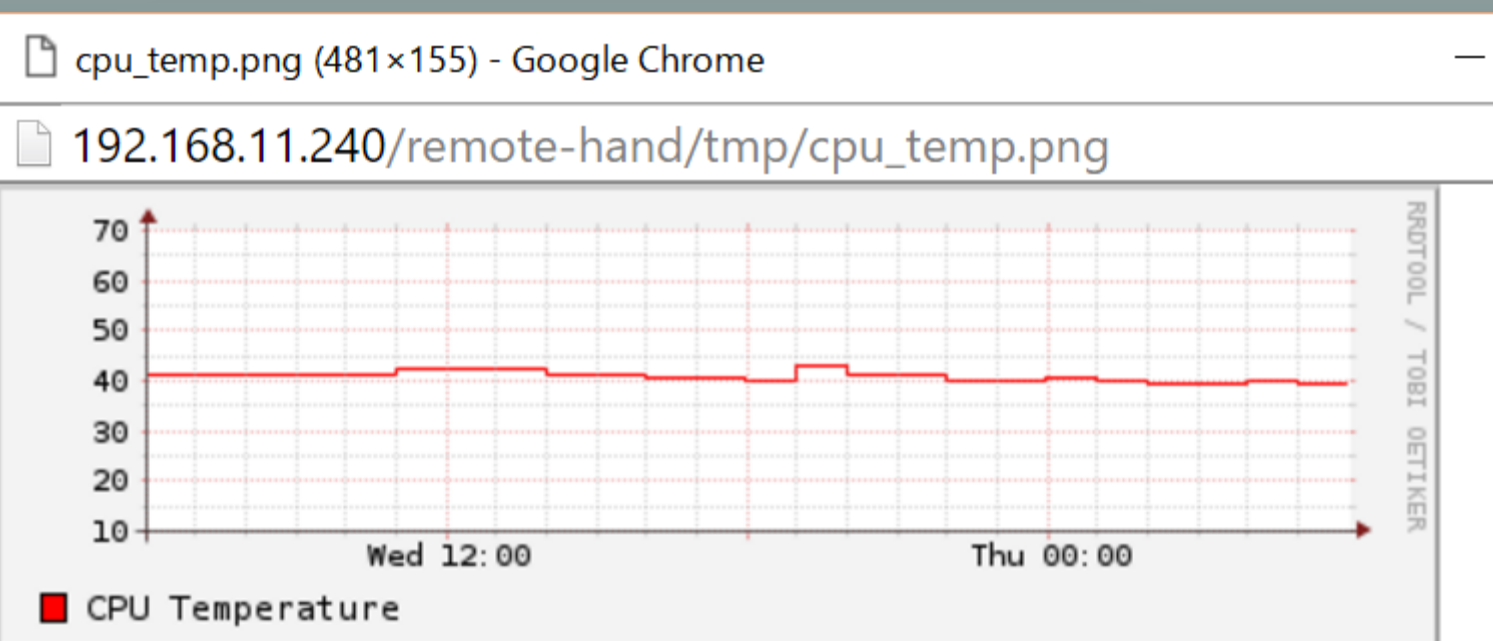
```
lrwxrwxrwx 1 root root 8 Nov 8 15:11 log -> /www/log
```

```
lrwxrwxrwx 1 root root 9 Nov 8 15:13 mail -> /www/mail
```

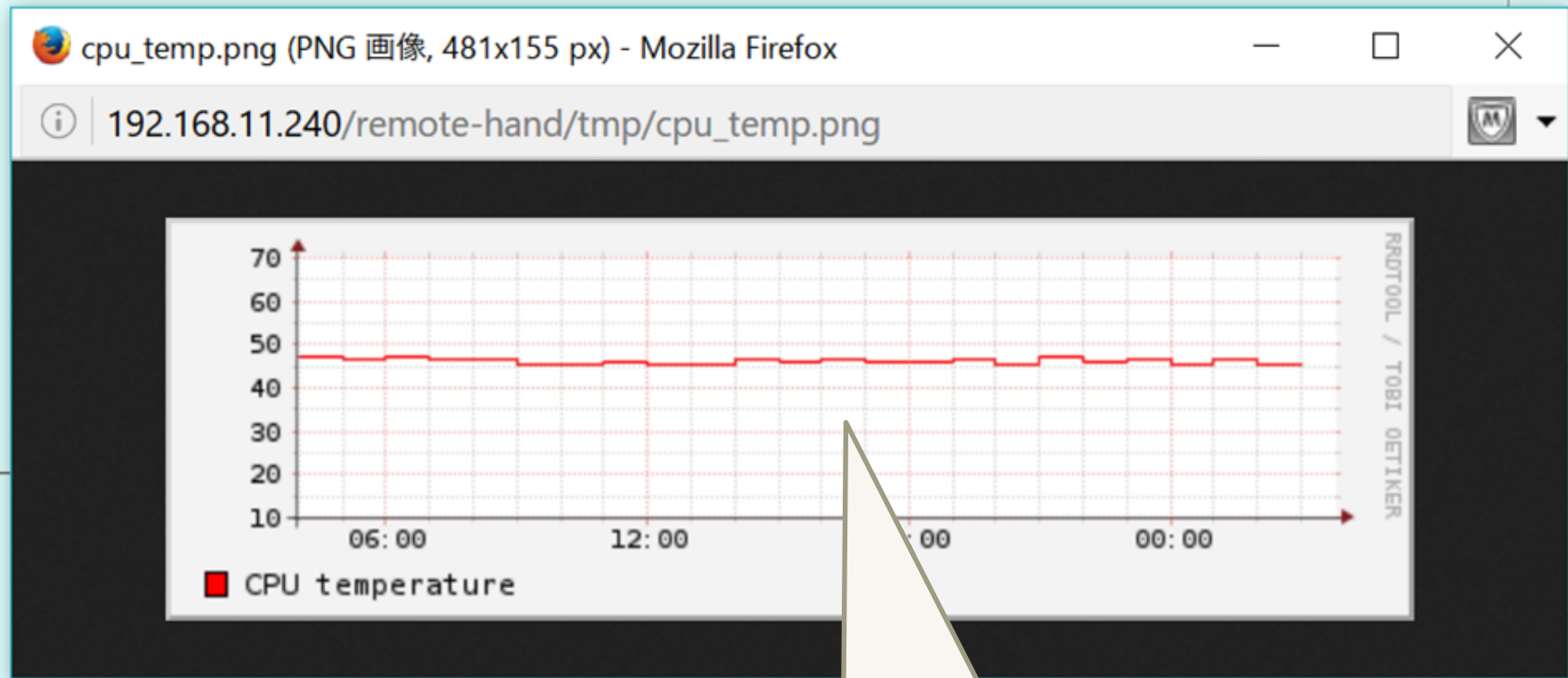
```
lrwxrwxrwx 1 root root 10 Nov 8 15:11 spool -> /www/spool
```


Raspberry pi ハングアップ

- ◆ CPU温度をリアル表示させたところ、何と58°C付近
- ◆ 急遽CPUファンを取り付け、50°C前後で推移
- ◆ 無線LANチップが高温になるので効果絶大
- ◆ CPUファンを回しっぱなしでは芸が無いので、pifaceのリレーでオンオフ
- ◆ CPUファンを運転してからは50°Cを超えることは無い
- ◆ 新VerではCPU温度条件でファンを回す



Raspberry piのCPUはスライス温度45°Cでファン制御



Input	Level	Name	Value
Input1	high	火事です	none
Input2	high	Input2	none
Input3	high	Input3	none
Input4	high	Input4	none
Input5	high	Input5	none
Input6	high	Input6	none
Input7	high	Input7	none
Input8	high	Input8	none

TI	Level	Name	Value
TI1	low	散水スイッチ	none
TI2	high	Input10	none
TI3	high	Input11	none

AI	Level	Value	Name	Slice	Value	Level
AI1	low	320	屋外照度	1300	none	
AI2	high	4080	Input14	3000	none	
AI3	low	4080	Input15	4500	none	
AI4	high	4080	Input16	2000	none	

CPU_Temperature **low** 45.4°C Slice 45 none

CPU温度45°C一定

CPU_Temperature	low	45.4°C	Slice 45	none	CPU Temperature Graph
USB_Temperature	low	19.7°C	Slice 25	none	USB Temperature Graph
I2C_Temperature	low	11.1°C	Slice 30	none	I2C Temperature Graph
I2C_Humidity	low	73.9%	Slice 90	none	I2C Humidity Graph

Interface piface Run Clear

Raspberry pi無線LANダウン

- ◆ 無応答時にUSB無線LANチップを抜き差しで復旧
- ◆ 無線LANチップの高温が原因では無かった
- ◆ 定期的にgatewayへping、failしたら無線LANをrestart

```
if ! ping -c $GATEWAY 2>&1 > $PING_LOG;then
```

```
ifdown wlan0
```

```
ifup wlan0
```

```
Fi
```

- ◆ これでもダメ、2.4GHZ帯は限界か
- ◆ やっぱりRaspberry piは有線で使おう

実践IoTハウス

Home Electronics & Security

Remote-Hand raspberry_pi ver:0.25 2016.5.4

Ethernet Setting Routing Setting Web Setting DIO Setting ping_DO Setting ping_mail Setting ping_tel Setting DIO Control1 DIO Control2 Mail Setting Auto Process Server Control

Setting system Email

Setting operation in Gmail

Gmail User: dokodemodo@gmail.com

Gmail Password:

Mail Address: all@pepolinux.local Allow Email address

Key Word: dio Subject(keyword)

Mail Check Timer: 30 New Email check interval(Sec)

Jitter: 30 EmailArrival correction(Sec)

Entry ▼ Run Clear

メールアドレス・キーワード・
チェック間隔を設定

Update Logout

©2016-2018 pepolinux.com 2016/05/12 06:40:12 37.9°C Graph

実践IoTハウス

Home Electronics & Security

Remote-Hand raspberry_pi ver:0.25 2016.5.4

Ethernet Setting Routing Setting Web Setting DIO Setting ping_DO Setting ping_mail Setting ping_tel Setting DIO Control1 DIO Control2 Mail Setting Auto Process Server Control

Server configuration and control

Date: Time:

Host name:

domain:

IP address eth0: / 24 ▼

default gateway:

nameserver1:

nameserver2:

root password:

web user:

web password:

wireless ssid: key:

IP address wlan0: / 24 ▼

Stop Restart Initial setting&Restart

IPアドレスなどネットワーク設定と現在のデータ保存

IPアドレス・メール設定の後リスタートで反映

©2016-2018 pepolinux.com 2016/05/12 06:41:54 34.7°C

実践IoTハウス

Remote-Hand raspberry_pi ver:0.32 2016.11.18

- Ethernet Setting
- Routing Setting
- Web Setting
- DIO Setting
- ping_DO Setting
- ping_mail Setting
- ping_tel Setting
- DIO Control1
- DIO Control2
- Mail Setting
- Auto Process
- Server Control

Setting DIO & IRKit & Tocos

Setting digital output terminal name

Server-Synchronized at 2016/11/19 06:17:47

Output1	low	none	ms	CPUファン			none
Output2	low	none	ms	火事テスト			none
Output3	high	none	ms	窓の電気			
Output4	low	none	ms	Output4			
Output5	low	none	ms	Output5			
Output6	high	none	ms	ななちゃん電灯			
Output7	high	none	ms	扇風機			
Output8	low	none	ms	Output8			none

入出力端子の名前設定と現在データ表示

Setting IRKit

IR1	none	ms	ななろく天井電灯		none	IR_data	Ready	none
IR2	none	ms	クーラー運転		none	IR_data	Ready	none
IR3	none	ms	クーラー停止		none	IR_data	Ready	none
IR4	none	ms	Output12		none	IR_data	none	none
IR5	none	ms	Output13		none	IR_data	none	none
IR6	none	ms	Output14		none	IR_data	none	none

IRKit_IP192.168.11.16 none Search_Set

Setting Tocos wireless ttyUSB1 I2C_Temperature&Humidity 06:17 14.6°C 94.9%

TO1	low	ms	散水ポンプ		none
TO2	none	ms	Output16		none
TO3	high	ms	玄関電灯		none

Setting Modem Device ttyUSB0

実践IoTハウス

Setting Modem Device

Voice control Mail:

Web Camera1 Web Camera2 Web Camera3 Module Camera

Server

Setting digital input terminal name

Input1	<input type="text" value="high"/>	<input type="text" value="火事です"/>	<input type="text" value="none"/>
Input2	<input type="text" value="high"/>	<input type="text" value="Input2"/>	<input type="text" value="none"/>
Input3	<input type="text" value="high"/>	<input type="text" value="Input3"/>	<input type="text" value="none"/>
Input4	<input type="text" value="high"/>	<input type="text" value="Input4"/>	<input type="text" value="none"/>
Input5	<input type="text" value="high"/>	<input type="text" value="Input5"/>	<input type="text" value="none"/>
Input6	<input type="text" value="high"/>	<input type="text" value="Input6"/>	<input type="text" value="none"/>
Input7	<input type="text" value="high"/>	<input type="text" value="Input7"/>	<input type="text" value="none"/>
Input8	<input type="text" value="high"/>	<input type="text" value="Input8"/>	<input type="text" value="none"/>

入出力端子の名前設定と現在データ表示

Setting TWE-LITE Wireless DI & AI

TI1

TI2

TI3

AI1

AI2

AI3

AI4

CPU Temperature

USB Temperature

I2C Temperature

I2C Humidity

Interface

実践IoTハウス

Home Electronics & Security

Remote-Hand raspberry_pi ver:0.25 2016.5.4

Ethernet
Setting

Routing
Setting

Web
Setting

DIO
Setting

ping_DO
Setting

ping_mail
Setting

ping_tel
Setting

DIO
Control1

DIO
Control2

Mail
Setting

Auto
Process

Server
Control

Management DI(Digital Input)-1

Setting first action to the digital input

火事です **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message:

Input2 **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message:

Input3 **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message: Count: -

Input4 **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message: Count: -

Input5 **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message: Count: -

Input6 **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message: Count: -

Input7 **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message: Count: -

Input8 **high** Action:low→high NONE Alt none Timer: ms none Log display

Phone: Email: Message: Count: -

入力イベントに対する処理の
設定-1とログ表示

実践IoTハウス

Phone: _____	Email: _____	Message: _____	Count: _____				
Input11	high	Action:low→high	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
火事です	high	Action:high→low	Phone	Alt	none	Timer: _____ ms	none
Phone: 090123456789	Email: _____	Message: _____	Count: 7	2016/02/29 17:13:03	~	2016/02/29 18:26:39	
Input2	high	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
Input3	high	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
Input4	high	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
Input5	high	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
Input6	high	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
Input7	high	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
Input8	high	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: _____	-			
散水スイッチ	low	Action:high→low	NONE	Alt	none	Timer: _____ ms	none
Phone: _____	Email: _____	Message: _____	Count: 46	2016/04/19 06:45:11	~	2016/05/12 04:44:15	

入力イベントに対する処理の設定-2

実践IoTハウス

Home Electronics & Security

Remote-Hand raspberry_pi ver:0.25 2016.5.4

- Ethernet Setting
- Routing Setting
- Web Setting
- DIO Setting
- ping_DO Setting
- ping_mail Setting
- ping_tel Setting
- DIO Control1
- DIO Control2
- Mail Setting
- Auto Process
- Server Control

Setting ping monitoring and digital output action

Setting IP address and the digital Output for monitoring

IP1:	192.168.11.1	DO:	窓の電気high	Timer:	50000	ms	Entry
IP2:		DO:	none	Timer:		ms	Entry
IP3:		DO:	none	Timer:		ms	Entry
IP4:		DO:	none	Timer:		ms	Entry

Ping monitoring interval: 5 Min

Run Clear

Update Logout

PING監視設定・タイムアウト
するとPifaceの出力端子を
ON/OFF

実践IoTハウス

Home Electronics & Security

Remote-Hand raspberry_pi ver:0.25 2016.5.4

Ethernet Setting Routing Setting Web Setting DIO Setting ping_DO Setting ping_mail Setting ping_tel Setting DIO Control1 DIO Control2 Mail Setting Auto Process Server Control

Setting ping monitoring and e-mail

Setting IP address and FAIL at which monitoring

IP1: 192.168.11.1	Email1: dokokanodare@gmail.com	Entry ▼
IP2: <input type="text"/>	Email2: <input type="text"/>	Entry ▼
IP3: <input type="text"/>	Email3: <input type="text"/>	Entry ▼
IP4: <input type="text"/>	Email4: <input type="text"/>	Entry ▼

Ping monitoring interval: 5 Min ▼

Run Clear

Update Logout

PING監視設定・タイムアウト
するとメールを送る

実践IoTハウス

Remote-Hand Raspberry_pi_××.zipで公開中

- ◆ なんとか容量1Gbyte程度
- ◆ Remote-Hand Raspberry_piは全てオープンソース下記からダウンロード
- ◆ <https://osdn.jp/projects/pepolinux/releases/p14620>



ご清聴有難う御座いました

残り時間少ないですがブースも見てね

PepoLinux

おしまい

<http://www.pepolinux.com>

Twitter@kujiranodanna

山内