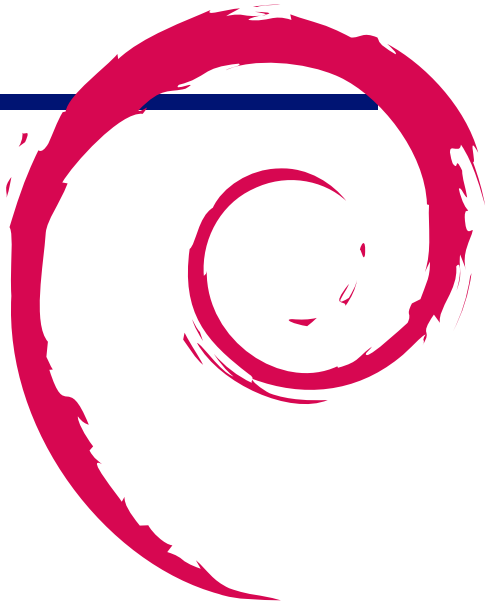
.Deb

銀河系唯一のDebian専門誌

2012 年 10 月 20 日

特集: Haskellパッケージ

特集: mtrack



1 はじめに

上川 純一

今月の Debian 勉強会へようこそ。これから Debian の世界にあしを踏み入れるという方も、すでにどっぷりとつかっているという方も、月に一回 Debian について語りませんか?

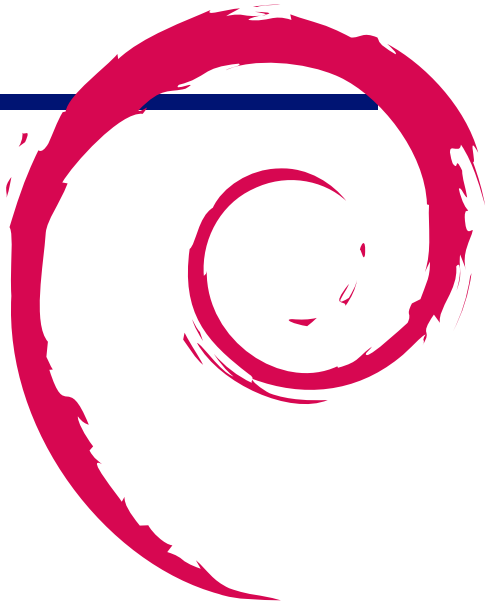
Debian 勉強会の目的は下記です。

- Debian Developer (開発者) の育成。
- 日本語での「開発に関する情報」を整理してまとめ、アップデートする。
- 場の提供。
  - 普段ばらばらな場所にいる人々が face-to-face で出会える場を提供する。
  - Debian のためになることを語る場を提供する。
  - Debian について語る場を提供する。

Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりと作るスーパーハッカーになった姿を妄想しています。情報の共有・活用を通して Debian の今後の能動的な展開への土台として、「場」としての空間を提供するのが目的です。

デビアン勉強会

目次



2 事前課題

上川 純一

今回の事前課題は以下です: 次から適当に選んで答えてください:

1. Debian での関数型プログラミング言語の利用経験とその時に感じた事柄(Lisp, Emacs Lisp, OCaml, Haskell 等)
2. 関数型プログラミング言語初心者に向けたお勧め書籍とそのウリの紹介,
3. 関数型言語で実装されている、使ってみたいソフトをあげてください

この課題に対して提出いただいた内容は以下です。

2.1 岩松 信洋

(1) erlang と Haskell を少々。
(2) Real World Haskell。プログラミング Erlang。
(3) Erlang の ビルドシステムである rebar をもうちょっと理解したい。

2.2 yamamoto

(1) 関数型プログラミング言語の利用経験 — なし (2) お勧め書籍 — 読んだこと、ありません (3) 使ってみたいソフト — 特になし
でも、 haskell はなんか面白そうだし、学んでみようかな? とか考えています。

2.3 alice.ferrazzi

2.4 鈴木崇文

(1)Debian での関数型プログラミング言語の利用経験とその時に感じた事柄(Lisp, Emacs Lisp, OCaml, Haskell 等) Erlang・・・ さわり程度ですが、 Riak などの実用レベルのソフトウェアで使用されている点や、分散環境や無停止に言語レベルで対応されている点が面白かったです。 Riak から情報を取ったり入れたりするだけならば比較的簡単に操作できました。 Haskell・・・ さわり程度ですが、 Haskell 好きな人が多いため学ぶには良い言語だと感じました。
(2) 関数型プログラミング言語初心者に向けたお勧め書籍とそのウリの紹介 Erlang はなかなか本がなかったです。 Haskell は「 すごい Haskell たのしく学ぼう! 」が読みやすいように感じますが、まだ理解できてません。
(3) 関数型言語で実装されている、使ってみたいソフトをあげてください Riak xmonad

2.5 吉野 (yy_y_ja_jp)

(1) 最近 Haskell を触っています. cabal-debian をもう少し知りたいです.

2.6 キタハラ

(1) Haskell 入門書のサンプルを動かした程度、 apt-get で簡単に導入できて感動したような記憶が・・・。 (2) Haskell の入門書を 2 冊ほど読みましたが、共に monad で挫折した、最近の本は読んでいない。 (3) 特になし。

2.7 dictoss(杉本 典充)

(1) emacs.el を書くくらいでなんとなく使っている感じです。シングルクォーテーションは閉じなくていい場合があるのでそれに戸惑うことがあります。

2.8 野首

elisp でちょっと major mode と shinbum module を書いてみたことがあるぐらいです。関数型プログラミングというレベルに至りませんでした。

2.9 @Lost_dog_

(1) Haskell:型安全のありがたみが分かった (2)『 すごい Haskell たのしく学ぼう!』は関数型の雰囲気を俯瞰できる。本気で勉強するなら、もっと王道のテキストを選んだほうがよいかも。(3) yi-editor

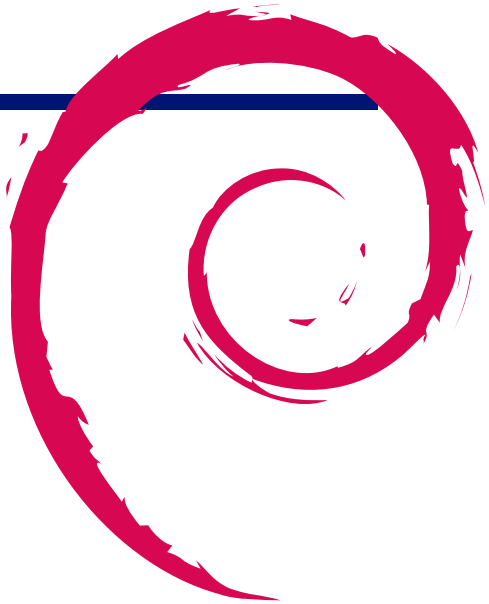
2.10 日比野 啓

(1) Haskell, OCaml ともに Debian には多数のパッケージがあってすばらしいです。

(2)

- **プログラミング Haskell**
  - Graham Hutton (著), 山本 和彦 (翻訳)]
  最初に読むならこれです。関数プログラミングのトピックを平易に解説しながが Haskell を試していきます。
- **すごい Haskell たのしく学ぼう!**
  - Miran Lipovaa (著), 田中 英行 (翻訳), 村主 崇行 (翻訳)
  「 プログラミング Haskell」の次はこれだと思います。より複雑な型の機能をも含めて Haskell の解説が進んでいきます。
- **Real World Haskell 実戦で学ぶ関数型言語プログラミング**
  - Bryan O'Sullivan (著), John Goerzen (著), Don Stewart (著), 山下 伸夫 (翻訳), 伊東 勝利 (翻訳), 株式会社タイムインターメディア (翻訳)
  実際に Haskell を現場で利用してる人たちが書いた本としての価値がある本です。 Haskell にもいろいろなライブラリがあり、その利用例として参考になると思います。少し古いのが問題点です。

(3) OCaml で作られている定理証明器 Coq をうまく使えるようになりたいです。あと Haskell でいろいろ作る側にまわりたい。



3 最近の Debian 関連のミーティング報告

上川 純一

3.1 東京エリア Debian 勉強会 91 回目報告

第 91 回東京エリア Debian 勉強会でした。参加者は中尾さん、山田泰志さん、やまねひできさん、やまもとひろゆきさん、北原さん、日比野さん、吉田@板橋さん、杉本さん、野島さん、岩松さん、 alice ferazzi さん、上川でした。

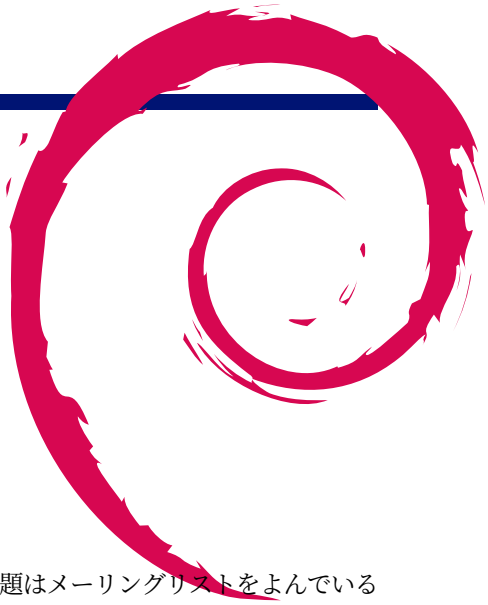
岩松さんが Debconf の紹介をしました。あつい。日本で開催するならどうしようかという話題でしばし盛り上がりました。 AARM64 の話題とか、 Clang で Debian をコンパイルしてみるプロジェクトについて語ってました。

野島さんが Debian においての共有ライブラリについて紹介しました。シンボルベースのバージョンをつかって shlibs ファイルに相当するものを作成する仕組みについて説明しました。それで本当に良いのか、いい感じに議論が紛糾。

野島さんが簡単にできそうな Debian 貢献について紹介しました。翻訳とか、 Debianart プロジェクトとか、適切なライセンスであるかのレビューだとか。

上川が C++11 を Debian で使う現状について語りました。なんで C++ やねんというリアクションもいい感じでした。実装面では、 Debian experimental に入ってきた libc++ がこれからどうなるのか楽しみです。

その後近所の居酒屋に場所を移して Debian の１９歳誕生日を祝いました。



4 Debian Trivia Quiz

岩松 信洋

ところで、みなさん Debian 関連の話題においついていますか? Debian 関連の話題はメーリングリストをよんでいると追跡できます。ただよんでいるだけではりあいがないので、理解度のテストをします。特に一人だけでは意味がわからないところもあるかも知れません。みんなで一緒に読んでみましょう。

今回の出題範囲は `debian-devel-announce@lists.debian.org` や `debian-devel@lists.debian.org` に投稿された内容と Debian Project News からです。

問題 1. 9/29 に行われた Debian 6.0 のアップデートは何回目でしょうか。

☐ A 5
☐ B 6
☐ C 7

問題 2. DMUA フィールドがなくなり、 Debian Maintainer のアップロードが変更されます。今後、アップロードの際にどのように作業する必要があるか?

☐ A FTP master に電話
☐ B スポンサーに dak の処理を依頼する
☐ C 専用アップローダにアップロード

問題 3. IRC 経由で VCS リポジトリを監視するサービスで終了したものは?

☐ A ICPO
☐ B NPA
☐ C CIA

問題 4. Debian Policy メンテナに新しく入ったのは誰か?

☐ A Kei Hibino
☐ B Colin Watson
☐ C Charles Plessy

問題 5. Checksums-SHA1,SHA256 の取り扱いが変更になったが、どう変更されたか?

☐ A 今まで無視されていました。ごめんね。
☐ B オプションだったので、必須としました。
☐ C これらは廃止し、 SHA-512 のみにします。

5 Haskell 周辺の Debian packaging

日比野

この記事では、プログラミング言語 Haskell のパッケージと、そのパッケージがどのように依存関係解決されて Debian 化されているのかについて紹介します。

5.1 HackageDB

http://hackage.haskell.org/ は Haskell で書かれたライブラリやツールのパッケージ配布サイトで、 Perl での CPAN[*1]のようなものです。ここで配布されている Haskell のパッケージ群は HackageDB と呼ばれ、それぞれのパッケージはしばしば Hackage と呼ばれます。次のような URL でそれぞれの Hackage の情報を参照できます。

```
http://hackage.haskell.org/package/<hackage の名前>
```

各 Hackage のソースコードの tarball には、バージョン情報やライブラリ、ビルドツールの依存関係情報が含まれています。

language-objc パッケージの例:

```
Name:          language-objc
Version:       0.4.2.5
...
Library
...
    Build-Depends: base       >= 3 && < 5,
                   filepath   >= 1.1 && < 1.4,
                   process    == 1.1.*,
                   directory  >= 1.1 && < 1.3,
                   array      == 0.4.*,
                   containers >= 0.4     && < 0.6,
                   newtype    == 0.2.*,
                   pretty     == 1.1.*
...
    Build-Tools:    happy, alex
```

5.2 Cabal ライブラリと cabal-install

Cabal [*2] は依存関係にしたがってパッケージを Hackage のサイトから取ってきたり、パッケージのソースツリーと手元の環境からパッケージを build したりするライブラリです。

cabal-install[*3]は Cabal ライブラリをコマンドラインから呼び出すためのツールです。 Hackage の名前は cabal-install ですが、コマンドの名前は cabal です。簡単な利用方法は例えば以下のような感じになります。

[*1] http://www.cpan.org/
[*2] http://hackage.haskell.org/package/Cabal
[*3] http://hackage.haskell.org/package/cabal-install

```
$ cabal unpack language-objc ## hackage.haskell.org からの取得と展開
$ cd language-objc-0.4.2.5
$ cabal configure            ## 依存関係の検査
$ cabal build                ## コンパイル
$ cabal copy                 ## インストール
$ cabal register             ## インストール管理情報を処理系に登録
```

あるいは

```
$ cabal install language-objc  ## 再帰的にインストールを全て実行
```

Cabal ライブラリ, cabal-install のどちらも Haskell で書かれていて HackageDB に置かれています。現在の事実上標準の Haskell 処理系は Glasgow Haskell Compiler(GHC) ですが、 Cabal ライブラリは GHC のソースツリーに含まれる形で配布されています。通常の開発環境で利用されるのはこの含まれている Cabal ライブラリです。

5.3 debian ライブラリと cabal-debian

やはり Haskell で書かれているもので debian[*4] ライブラリと cabal-debian[*5]というツールがあります。

debian ライブラリは Debian ソースパッケージの情報をハンドリングするためのライブラリです。このライブラリを使って作られているのが cabal-debian で、次のように Hackage のソースを Debian のソースパッケージに変換することができます。

```
$ cabal unpack language-objc
$ cd language-objc-0.4.2.5
$ cabal-debian --debianize
```

次のようにうまく依存関係の情報が変換されます。

```
$ cat debian/control
Source: haskell-language-objc
Priority: extra
Section: haskell
...
Build-Depends: debhelper (>= 7.0),
               haskell-devscripts (>= 0.8),
               cdbs,
               ghc,
               ghc-prof,
               libghc-newtype-dev (>= 0.2) | libghc-newtype-dev (<< 0.3),
               libghc-newtype-prof (>= 0.2) | libghc-newtype-prof (<< 0.3),
               libghc-syb-dev (>= 0.3) | libghc-syb-dev (<< 0.4),
               libghc-syb-prof (>= 0.3) | libghc-syb-prof (<< 0.4),
               happy,
               alex
Build-Depends-Indep: ghc-doc,
                     libghc-newtype-doc (>= 0.2) | libghc-newtype-doc (<< 0.3),
                     libghc-syb-doc (>= 0.3) | libghc-syb-doc (<< 0.4)
...
```

ライブラリのパッケージだと、そのまま利用できる程度のものが自動的に生成されます。実行ファイルがあるものについては rules にインストール先を書く必要があります。

現在の Debian では libghc-¡hackage の名前¿-dev,prof,doc という名前で Debian パッケージが作られます。 libghc-*-dev は開発用ライブラリ、 libghc-*-prof はプロファイル情報取得用のライブラリ、 libghc-*-doc はライブラリドキュメントです。

Haskell の場合は haddock というツールでソースコード内の特定の形式のコメントがドキュメントに変換されます。libghc-*-doc はここから作られます。

5.4 haskell-devscripts

cabal-debian で作られた Debian のソースパッケージは cdbs 用の Makefile (hlibrary.mk) を利用するように構成されています。 hlibrary.mk は haskell-devscripts という Debian パッケージに含まれています。

[*4] http://hackage.haskell.org/package/debian
[*5] http://hackage.haskell.org/package/cabal-debian

```
$ cat debian/rules
#!/usr/bin/make -f
include /usr/share/cdbs/1/rules/debhelper.mk
include /usr/share/cdbs/1/class/hlibrary.mk

# How to install an extra file into the documentation package
#binary-fixup/libghc-language-objc-doc::
#        echo "Some informative text" > debian/libghc-language-objc-doc/usr/share/doc/libghc-language-objc-doc/AnExtraDocFile
```

hlibrary.mk から利用されてるコマンド群 (dh_haskell_*) も haskell-devscripts に含まれています。各コマンドは基本的には GHC のインストール情報管理用のファイル (debian/*/var/lib/ghc/package.conf.d/*.conf) を利用しつつ、*.substvars を出力します。

dh_haskell_depends　Haskell のライブラリの依存関係を Debian の依存関係に変換します。

dh_haskell_extra_depends　データのパッケージや実行プログラムのパッケージといった、ライブラリの依存関係では解決できない依存関係を Debian の依存関係に変換します。

dh_haskell_provides　Haskell のライブラリの Debian 仮想パッケージも含めた提供情報を計算します。

dh_haskell_shlibdeps　Haskell のライブラリが依存しているライブラリアーカイブ (*.a) を提供しているパッケージを検索し Debian の依存関係として出力します。shlibdeps なのにライブラリアーカイブ (*.a) のみ検索するのは、現状の Debian の GHC 環境だと、ライブラリパッケージを共有ライブラリにはコンパイルしていないからです。

5.5 おわりに

HackageDB の依存関係管理と Debian パッケージの関係について書いてみました。関連のツールもよくできているので、Debian 化も簡単にできそうに見えたことと思います。この機会に気になる Hackage を Debian 化してみてはどうでしょうか。

6　レゴでなめこ収穫機を作ってみた

本庄 弘典

6.1　はじめに

PC 上で動作させた python スクリプトからレゴ マインドストームを操作し、 iPhone や Android で人気のアプリ『 なめこ栽培キット』の収穫機を作りました。

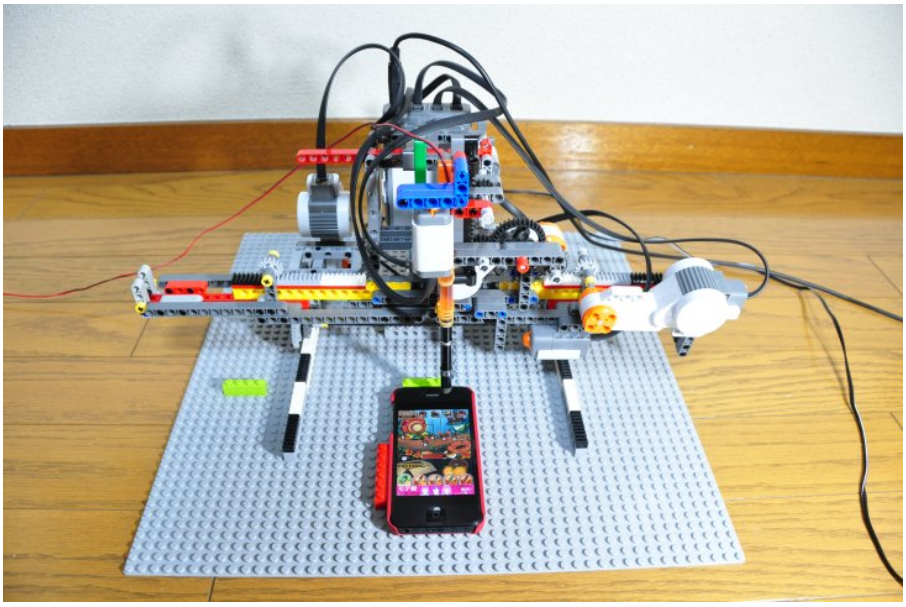


図 1　なめこロボ

6.2　『 なめこ栽培キット』とは

原木になめこの餌を注入し、生えてきたなめこを指でなぞって収穫する iPhone および Android のゲームです。株式会社ビーワークスからリリースされています。

- 正式名称は『 おさわり探偵なめこ栽培キット』。
- iPhone 版での最新作は『 おさわり探偵なめこ栽培キット Deluxe』。
- たまにレアなめこが生える。

6.3 レゴ マインドストームの紹介

株式会社レゴが販売している教育用ロボットで、次のような特徴を持っています。

- レゴブロックを使って組み立てられる
- レゴ テクニックシリーズのパーツを使用する
- モーターや各種センサーを制御できる
- ARM7 のコンピュータユニットから制御する
- ET ロボコンで使われている

6.3.1 レゴ マインドストームの購入方法

ET ロボコンの委員をやっている先生に購入方法を伺ったところ、「(株) アフレルから買ってください」と言われました。

- 教育用レゴ マインドストーム 正規代理店(株) アフレル
- `http://www.afrel.co.jp/`
- 現在価格 39,900 円

こちらでは ET ロボコン用のセット販売などを行っているようですが、特に ET ロボコンにこだわらない場合、Amazon.co.jp から現在価格 34,000 円で購入できます。

また一部のパーツは個別に購入することが可能です。筆者は次のお店を利用しました。

- ホビーショップ デジラ `http://www.dgla.jp/shop/`
- LEGO パーツ専門店 ハック フィン `http://huckfinn-lego.com/`
- レゴ パーツ販売ショップ「ブリッカーズ」 `http://www.brickers.jp/`

マインドストームにはいくつか種類がありますが、最新のものは『レゴ マインドストーム NXT 2.0』で、筆者もこちらを購入しました。

6.4 nxt-python

今回の目的はなめこ栽培キットの自動収穫機であるため、マインドストームを自立動作させる必要がありません。nxt-python は PC とマインドストーム NXT を USB や Bluetooth で接続し、PC 側で実行したスクリプトに従ってマインドストームをコントロールするための python モジュールです。

nxt-python は次の場所で公開されています。

- `http://code.google.com/p/nxt-python/`

Debian sid には python-nxt というパッケージがあり、こちらを導入することで nxt-python が利用できます。

```
$ sudo apt-get install python-nxt
```

6.4.1 サンプルコード

次のコードはポート B に接続されたモーターをパワー 100、360 度回転させるサンプルコードです。

```
#!/usr/bin/python

import nxt.locator
from nxt.motor import *

b = nxt.locator.find_one_brick()
m_left = Motor(b, PORT_B)
m_left.turn(100, 360)
```

なおドキュメントは見つからなかったため、ソースコードを読みながら使い方を調べました。

6.4.2 BaseMotor::turn(power, tacho_units)

turn() は power にパワーを、 tacho_units に角度を指定してモーターを回すメソッドです。

- 回りきるまで制御は帰ってこない。
- デフォルトでは回りきったところでブレーキをかける。
- 途中でモーターが回転しなくなると例外を飛ばす。
- power は-100 ～ 100 を指定可能。
- power に負の値を指定すると回転が逆になる。
- power に-10 ～ 10 くらいの値を指定しても力が弱くてモーターは回らない。

6.4.3 Motor::weak_turn(power, tacho_units)

turn() は無理矢理モーターを手で止めると怪我をするくらい強く回しますが、 weak_turn() はモーターを弱々しく回します。

- 手で止めると簡単にモーターは止まる。
- このとき例外は飛ばさない。
- 手で軽く回すと再び回り始める。
- tacho_units 分回ると制御が帰る。
- 使いどころが分からない。

6.4.4 Motor::run(power=100)

run() は指定されたパワーでモーターを回しつづけます。 run() を実行したのち、制御はすぐに戻ってきます。制御が戻ってきてもモーターは回りつづけます。

6.4.5 Motor::brake(), Motor::idle()

brake() および idle() はモーターを止めるメソッドです。 brake() はモーターを止めて動かないようブレーキをかけますが、 idle() は惰性で回りつづけ、緩やかに止まります。

6.4.6 Motor::get_tacho()

モーターは何度回ったかをカウントしており、 get_tacho() メソッドでこのカウンタを取得できます。 get_tacho() は次のの三つの値を返します。

- .tacho_count
- .block_tacho_count
- .rotation_count

6.4.7 Motor::reset_position(relative)

relative に True を渡すと.block_tacho_count を、 False を渡すと.rotation_count をリセットします。

6.4.8 Touch::is_pressed()

タッチセンサーが押されていれば True を、それ以外は False を返します。

6.4.9 その他

turn() メソッドはデフォルトの動作でブレーキをかけてくれるのですが、指定した数値どおりに動いてはくれません。しかし tacho は正確な値に見えるので、これを頼りに位置指定を行う関数を定義して利用しました。

```
def absturn(m, p):
    d = 1
    c = m.get_tacho().rotation_count
    q = p - c
    if q &lt; 0:
        d = -1
        q = abs(q)
    m.turn(30*d, q)
```

6.5 作成されたもの

実際に動かしているコードを gist で公開しました。

- https://gist.github.com/3897500

動画は YouTube で見られます。

- http://www.youtube.com/watch?v=uIEyWBs68c8
- 「 レゴ なめこ」で検索すると上の方に出てきます。

6.6 最後に

機械制御はなかなか思ったとおりに動いてくれないことが多く、モーターが正確な位置に動いてくれないなどの苦労に見舞われましたが、なんとか軽減することができました。

これを機会に皆さんもレゴ マインドストーム NXT の世界を覗いてみてはいかがでしょうか。

7 xserver-xorg-input-mtrack

岩松

Apple 社の Macbook Pro の タッチパッドは 2011 年 (2010 年?) 以降からマルチタッチに対応したトラックパッドを使っています。以前のトラッパッドでもマルチタッチができましたが、一部の機能しか使うことができないようです(pre-multitouch と呼ばれるようです)。 pre-multitouch では xserver-xorg-input-synaptics ドライバで動作していたのですが、最新の Macbook pro / air / Magic Trackpad では動作しません。これらの環境でマルチタッチを使うためには xserver-xorg-input-mtrack / multitouch といった他のドライバを利用する必要があります。今回、これらの情報をまとめました。

7.1 synaptics と mtrack

今まで使われてきた synaptics では、プロトコル A(図 3) が主体です。これはタッチ ID を持っていないため、トラッキングコンタクトを管理できません。

mtrack では、タッチ ID をサポートしたプロトコル「 プロトコル B (図 3) 」を使います。これによりより細かいタッチパッドの制御ができるようになっています。また、「 プロトコル B」を Linux カーネルから受信し、それを X ドライバにわかりやすい(人間にとってわかりやすい) 形にデータを形成するライブラリ mtdev を使います。

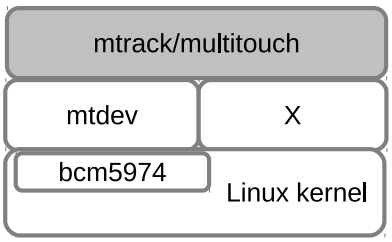


図 2 関係図

```
ABS_MT_POSITION_X x[0]
ABS_MT_POSITION_Y y[0]
SYN_MT_REPORT
ABS_MT_POSITION_X x[1]
ABS_MT_POSITION_Y y[1]
SYN_MT_REPORT
SYN_REPORT
```

図 3 プロトコル A

```
ABS_MT_SLOT 0
ABS_MT_TRACKING_ID 45
ABS_MT_POSITION_X x[0]
ABS_MT_POSITION_Y y[0]
ABS_MT_SLOT 1
ABS_MT_TRACKING_ID 46
ABS_MT_POSITION_X x[1]
ABS_MT_POSITION_Y y[1]
SYN_REPORT
```

図 4　プロトコル B

7.2　multitouch と mtrack

現在、 Macbook Pro / Air 用のタッチパッドをサポートしているドライバは multitouch と mtrack の 2 つがあります。 multitouch が先に作られましたが、基本的な設定しかできないシンプルな設定だったため、 mtrack という multitouch からフォークしたドライバが作成されました。こちらはタッチの圧力やジェスチャーのサポートが行われているため、使っているユーザが多いようです。

7.3　Debian で使う

Debian では既にパッケージ化されており、 APT でインストールできます。

```
$ sudo apt-get install xserver-xorg-input-mtrack
```

インストールされると、図 5 の内容の xorg 設定ファイルが /usr/share/X11/xorg.conf.d/以下にインストールされ、xorg.conf に記述しなくても動作するようになります。

```
Section "InputClass"
    MatchIsTouchpad "true"
    Identifier "Multitouch Touchpad"
    Driver "mtrack"
EndSection
```

図 5　mtrack のデフォルト設定

インストールした段階では、デフォルトの設定で動作します。以下によく使われる設定項目を表 1 示します。
以下に有効な設定を載せておきます。

- トラックパッドのシングルタップを無効にする
  トラックパッドに触っても(シングルタップしても）何も起きなくなります。

  ```
  Option "TapButton1" "0"
  Option "TapButton2" "0"
  Option "TapButton3" "0"
  ```

- 2 本指スクロールの動きをを OS X と同じにする

  ```
  Option "ScrollUpButton" "5"
  Option "ScrollDownButton" "4"
  ```

7.3.1　その他の情報

現在の mtrack ドライバは synaptics のように設定値を動的に変更できません。 xorg の設定を変更したい場合には、ファイルを変更し X サーバを再起動させる必要があります。これでは細かい設定等を行う時に大変なので常に設定を変更できるようにするためのパッチを作成し、アップストリームに送りました。 https://github.com/BlueDragonX/xf86-input-mtrack/pull/41

パッチを適用し、以下の設定を行なって X サーバを立ち上げると動的に設定を変更できるようになります。

```
Option "SHMConfig" "true"
```

Debian パッケージへの対応ですが、アップストリームで適用されれば更新したパッケージをアップロードする予定です。

肝心の設定用のツールですが、適当に作ったので後日公開します。

7.4 Debian でマルチタッチを使う場合にはどうしたらいいのか

大統一 Debian 勉強会での赤部さんの発表[*6] にもあったように、 Debian ではまだマルチタッチを提供するツール等が十分ではありません。 X や GTK+ などでは既にマルチタッチは対応していますが、それを使うアプリケーションがなく、 Ubuntu で採用されている utouch [*7] 関連のライブラリもまだパッケージになっていない状態です(ITP はされています）。よって Debian ではまだ iPad や Android タブレット相当の操作はできないと思われます。

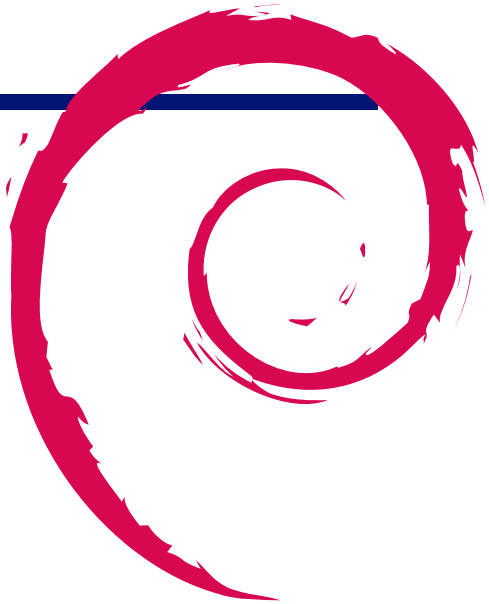
7.5 まとめ

mtrack ドライバでマルチタッチの制御はできるようになっていますが、アプリケーションが追いついていないのが現状です。ユニバーサルオペレーティングシステムを目指す以上、マルチタッチは避けて通れない機能なので早くサポートされてほしいものです。

[*6] http://gum.debian.or.jp/download/debian-gum-presentation.akabe.pdf
[*7] https://wiki.ubuntu.com/Multitouch

項目	内容	デフォルト値
TrackpadDisable	トラックパッド機能の動作内容と無効化	0
Sensitivity	トラックパッドのスピード	1
FingerHigh	指がタッチとして検知される圧力	5
FingerLow	指がリリースとして検知される圧力	5
IgnoreThumb	親指であるとわかるタッチを無視するか	False
IgnorePalm	手の平であるとわかるタッチを無視するか	False
DisableOnThumb	親指がさわっているとき全てのトラックパッドを無効にするか	False
DisableOnPalm	手の平ががさわっているとき全てのトラックパッドを無効にするか	False
ThumbRatio	親指の幅/長比率	70
ThumbSize	親指の最小限のサイズ	25
PalmSize	手の平の最小限のサイズ	10
BottomEdge	トラックパッドの無視する領域をパーセンテージで指定。	10
ButtonEnable	トラックパッドの物理ボタンを無効にするか	True
ClickFinger1	1 本指でのクリック動作	3
ClickFinger2	2 本指でのクリック動作	2
ClickFinger3	3 本指でのクリック動作	0
TapButton1	1 本指でのダブルタップ動作	1（ クリック&コピー）
TapButton2	2 本指でのダブルタップ動作	3（ ペースト）
TapButton3	3 本指のダブルタップ動作	2（ 右クリック）
TapButton4	4 本指でのダブルタップ動作	0（ 無効）
MaxTapTime	タップを認識する最大時間（ 値を小さくすると早くタップする必要がある。）	120
ScrollDistance	スクロールを有効にするために動かす距離	150
ScrollUpButton	2 本指でのスクロールアップ	4（ スクロールアップ）
ScrollDownButton	2 本指でのスクロールダウン	5（ スクロールダウン）
ScrollLeftButton	2 本指でのスクロールレフト	6（ スクロールレフト）
ScrollRightButton	2 本指でのスクロールライト	7（ スクロールライト）
SwipeDistance	スワイプを有効にするために動かす距離	700
SwipeUpButton	スワイプ（ 3 本指）アップ	8（ ALT + →/進む）
SwipeDownButton	スワイプ（ 3 本指）アップ	9（ ALT + ←/戻る）
SwipeLeftButton	スワイプ（ 3 本指）レフト	10（ 不明）
SwipeRightButton	スワイプ（ 3 本指）ライト	11（ 不明）
Swipe4Distance	スワイプを有効にするために動かす距離	700
Swipe4UpButton	スワイプ（ 4 本指）アップ	8
Swipe4DownButton	スワイプ（ 4 本指）ダウン	9
Swipe4LeftButton	スワイプ（ 4 本指）レフト	10
Swipe4RightButton	スワイプ（ 4 本指）ライト	11
AxisXInvert	X 軸を逆にするか	false
AxisXInvert	Y 軸を逆にするか	false

表 1　よく使う mtrack の設定項目



8 索引

Debian 勉強会資料

2012 年 10 月 20 日　　初版第 1 刷発行

東京エリア Debian 勉強会（ 編集・ 印刷・ 発行）