Raspberry PI with NetBSDでGPIOを使う

おおしまやすし

GPIO

- General Purpose Input/Output: 汎用入出力
- コンピュータのもっとも低レベルな外部インターフェース
- ビット単位で端子がある

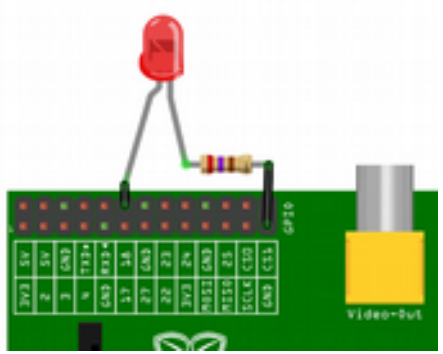
gpio(4)

- NetBSD 4.0以降でサポートされたgpio用デバイスドライバ
- ピン単位で入出力方向/モード指定、ピン入出力の実際
- 詳しくはman 4 gpio

bcmgpio(4)

- Raspberry PIのGPIOインターフェース
- NetBSD 7系からサポート
- SPIやI2C, シリアル通信とも共用
- デフォルトで何も使ってないピンは8個
- 26ピンヘッダのgpio0〜7と実際のCPUのピンに付いてるgpioピン番号はバラバラなので注意。コネクタのピン番号とも異なる。

GPIO名	ピン	端子番号	GPIO名	ピン	端子番号
GPIO0	17	11	GPIO1	18	12
GPIO2	27	13	GPIO3	22	15
GPIO4	23	16	GPIO5	24	18
GPIO6	25	22	GPIO7	4	7



- 今更なので詳しくはインターネットで

bcmgpio(4)

dmesgより:

```
bcmgpio0 at obio0: GPIO [0...31]
gpio0 at bcmgpio0: 32 pins
bcmgpio1 at obio0: GPIO [32...53]
gpio1 at bcmgpio1: 22 pins
```

2つのgpioデバイスがあって32bitと22bitある

gpioctl(8)

- NetBSDのgpio制御コマンド
- 基本的な使い方

```
# gpioctl gpio0                 # gpio0のピン数を取得
32
# gpioctl gpio0 17 set in       # pin17を入力に
# gpioctl gpio0 17              # pin17の状態を表示
0
# gpioctl gpio0 24 set out      # pin24を出力に
# gpioctl gpio0 24 1            # pin24をHigh(1)に
```

gpio(4) 注意

- RPI標準カーネルにて通常(マルチユーザ環境から) gpioctl gpio0 でピン数が0??

  ```
  # gpioctl gpio0
  0
  ```

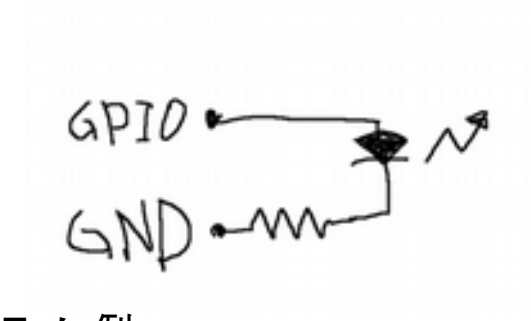
- gpioctl(8)のman pageを読もう

  > Only pins that have been configured at securelevel 0, typically during system startup, are accessible once the securelevel has been raised.

- つまり通常のマルチユーザモード(securelevelは1になる)では、ピンのモード設定できない
  - 変更するには/etc/gpio.confに記述。起動時しか変更できない。
    これは不便。ハードウェアをゴネゴネしてるときは結構ポートのモードも変えたりする。頻繁に触る時には option INSECURE 付き(securelevel =-1) のカーネルでやった方がいいかも。
    完成したら通常は途中で変わることもないのでセキュリティ上はいいけど
  - しかしi386/amd64のGENERICは(別の事情で)securelevel -1なのでいつでもできる(これ罠だよなー)
  - RPIカーネルはsecurelevel 0で起動するがrc.d/securelevelで1する。

Lチカ

- 適当なGPIOポートにLEDをつなげてON/OFFする
- 回路

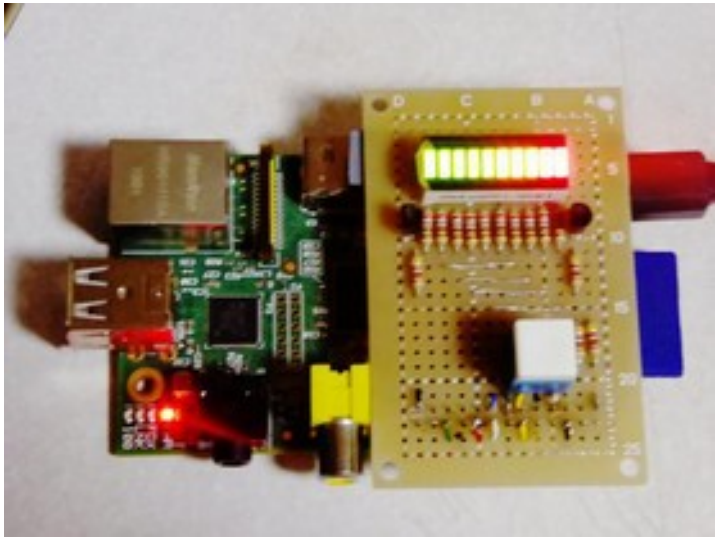


プログラム例:

```
struct gpio_req req;
  :
fd=open(“/dev/gpio0”,O_RDWR);
 :
req.gp_pin = 17;  /* GPIO0の場合  */
req.gp_value = GPIO_PIN_HIGH; /*点灯*/
 :
ioctl(fd, GPIOWRITE, &req);
```

タイマーにしてみる

- 10連レベルメータ(要はLED10個)+スイッチ1個
- そのまま使えるGPIOポートが8個しかなく足りないので1ポート使ってトランジスタで切り替えて5ポート2バンクに
- プログラムはSIGALRMで駆動してるだけ



1つ点灯
→次の点滅
→次の点灯
→全部点灯したら全部点滅
→途中でスイッチ押されたら最初からやり直し

これから

- iic(4)やspi(4)も認識してるんだがこれどう使えばいいんの? userlandからだと分解能キツくない?

```
bcmspi0 at obio0 intr 54: SPI
spi0 at bcmspi0: SPI bus
bsciic0 at obio0 intr 53: BSC0
iic0 at bsciic0: I2C bus
bsciic1 at obio0 intr 53: BSC1
iic1 at bsciic1: I2C bus
```

- ユーティリティ的な用途にディスプレイとか各種センサー繋げたい