

# RaspberryPiで電流測ってみました

さとう

LILO&東海道らぐオンラインミーティング  
2020-05-020



※ 一部、過去発表と重複があります

Page 1

# 最近やってること

- 例えば覆面算をPythonで解いてみたりしてます。

```
$ python3 s2.1.py
```

```
  s e n d  
+ m o r e  
-----  
m o n e y
```

```
$ python3 banana.py
```

```
  バ ナ ナ  
+ バ ナ ナ  
-----  
シ ナ モ ン
```

# 最近やってること

- 例えば覆面算をPythonで解いてみたりしてます。

```
$ python3 s2.1.py
```

```
  s e n d  
+ m o r e  
-----  
m o n e y
```

```
  9 5 6 7  
+ 1 0 8 5  
-----  
1 0 6 5 2
```

```
$ python3 banana.py
```

```
  バ ナ ナ  
+ バ ナ ナ  
-----  
シ ナ モ ン
```

```
  8 7 7  
+ 8 7 7  
-----  
1 7 5 4
```

※出典: [パズル遊びへの招待・オンライン版](#)  
ウィキペディア「[覆面算](#)」

# 錆びかけたラズパイであれこれ

- 実はRaspbrryPi4 まだ買ってません(汗

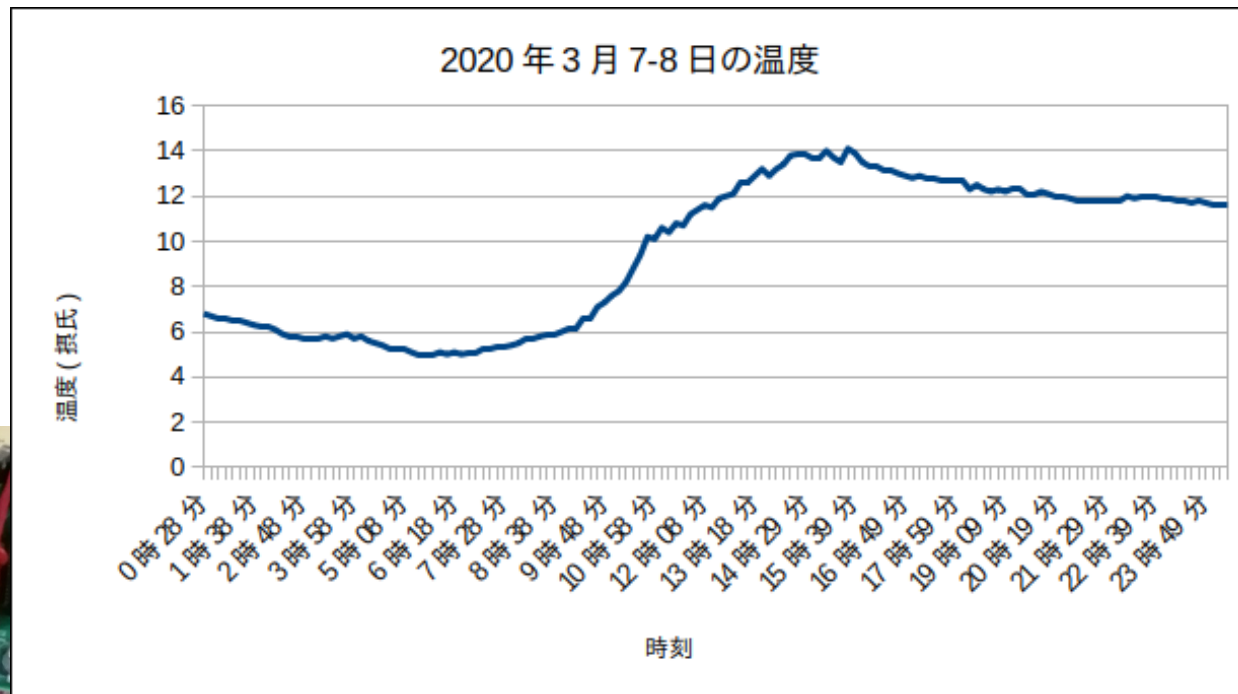
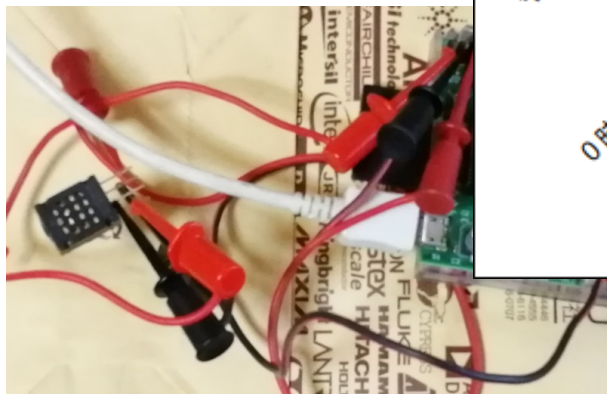


# センサー繋いでいます。

- センサーに興味が出てきています。
- まず、温度センサー買いました。



# 温度センサーでこのとおり



# 少し前に

- 電圧/電流センサ買いました!
- Adafruit  
INA219 High Side DC Current Sensor Breakout
- 接続はI2C
- ArudinoとRaspberryPiの  
ドキュメントあり





# そんなの買ってどうするの

- 「ちょっと」電流を測って見たかった。
- はい、電流計なら持ってます。

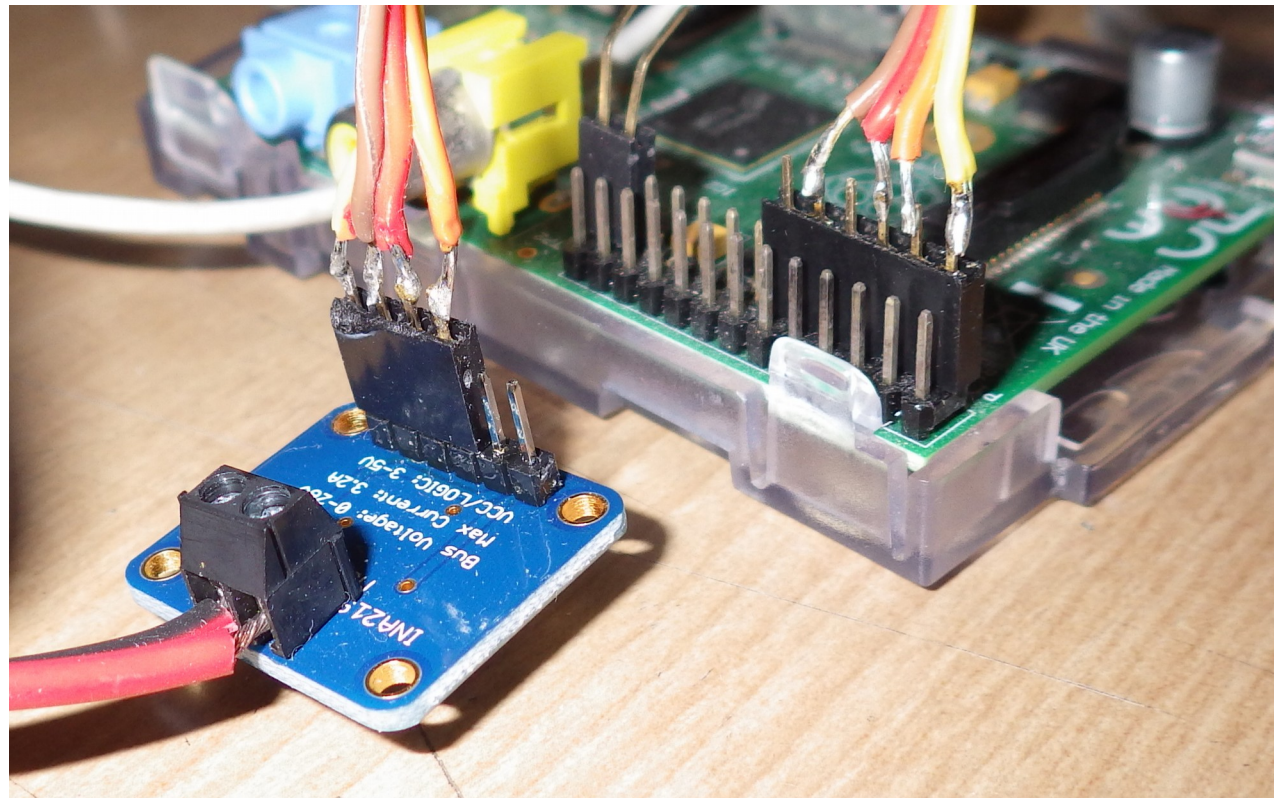




# でも、

- 「表示する電流計」はあっても、「記録してくれる」電流計はあんまりないみたい。
- ないことはないけど、ハードウェア開発の専用機器に飛んでゆきそう(安くても万円単位)  
(クランプ式電流センサくらい知っておけ!とのツッコミありました)
- なにより、作れるなら作ってみたい!

# ラズパイに繋ぐとこんなふうです



# 動作確認。

- あれ、見えてない...

```
$ sudo i2cdetect -y 1
```

```
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
```

```
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
70:  -- -- -- -- -- -- -- --
```

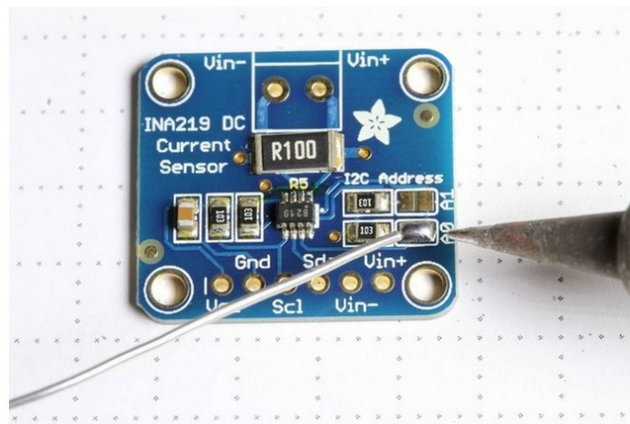
# ドキュメント読みなさい。

- <https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout/assembly>
- ジャンパをショートさせて、I2Cのアドレスを指定します

## Addressing the Boards

If more than one INA219 breakout board is used, each board must be assigned a unique address. This is done with the address jumpers on the right edge of the board. The I2C base address for each board is 0x40. The binary address that you program with the address jumpers is added to the base I2C address.

To program the address offset, use a drop of solder to bridge the corresponding address jumper for each binary 1 in the address.



Up to 4 boards may be connected. Addressing is as follows:

**Board 0:** Address = 0x40 Offset = binary 00000 (no jumpers required)

**Board 1:** Address = 0x41 Offset = binary 00001 (bridge A0 as in the photo above)

# やりなおし。

- 今度はばっちりです。

```
$ sudo i2cdetect -y 1
```

```
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
```

```
00:  --  - - - - -
```

```
10:  - - - - -
```

```
20:  - - - - -
```

```
30:  - - - - -
```

```
40:  - - 41 - - - - -
```

```
50:  - - - - -
```

```
60:  - - - - -
```

```
70:  - - - - -
```

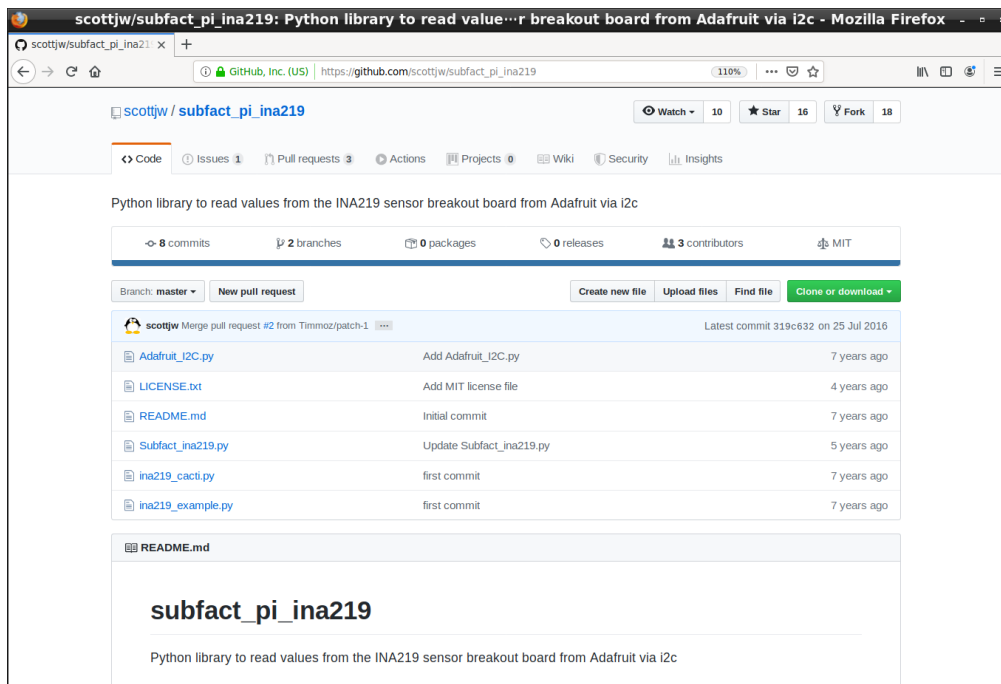
# やはり専用デバイスは楽です。

- 市販のI2Cデバイスは、ラズパイでは「そのままでは」動かないものもある、そうです。
- ラズパイに内蔵された「プルアップ抵抗」( $1.8\text{k}\Omega$ )は、おそらく「オマケ」。
- ちゃんと動かすためにはそれなりの回路を



# ソフトは

- これもPythonで作ってくれている人がおりました。
- 感謝!



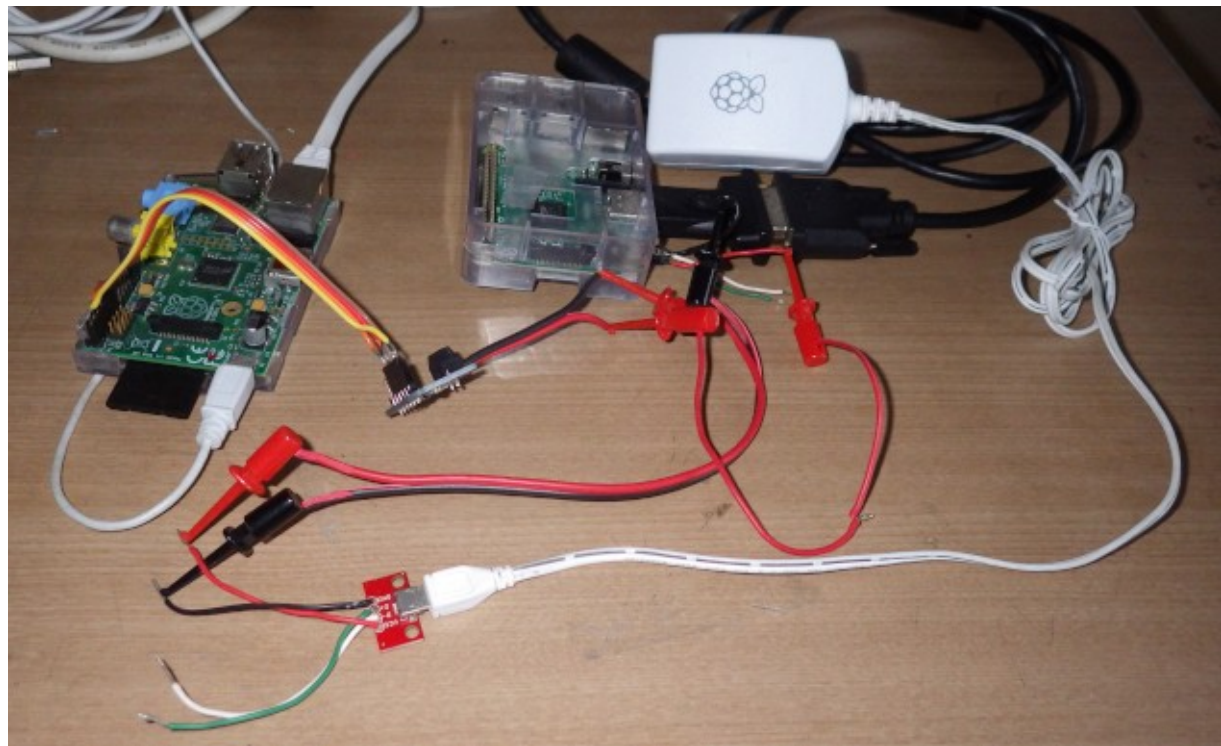
# で、ソフトも少し書き換えてみました

- 0.1秒ごとに電流を測定して、規定回数(500回)測定し、まとめて標準出力するように

```
#!/usr/bin/python
from Subfact_ina219 import INA219
from time import sleep
ina = INA219()
elist = []
for x in range(500):
    elist.insert(x,ina.getCurrent_mA())
    sleep(0.1)
for y in elist:
    print y
```

# ラズパイを起動させて

- 電流の変化を追ってみました
- 接続はこんな感じです



# 取れたデータを

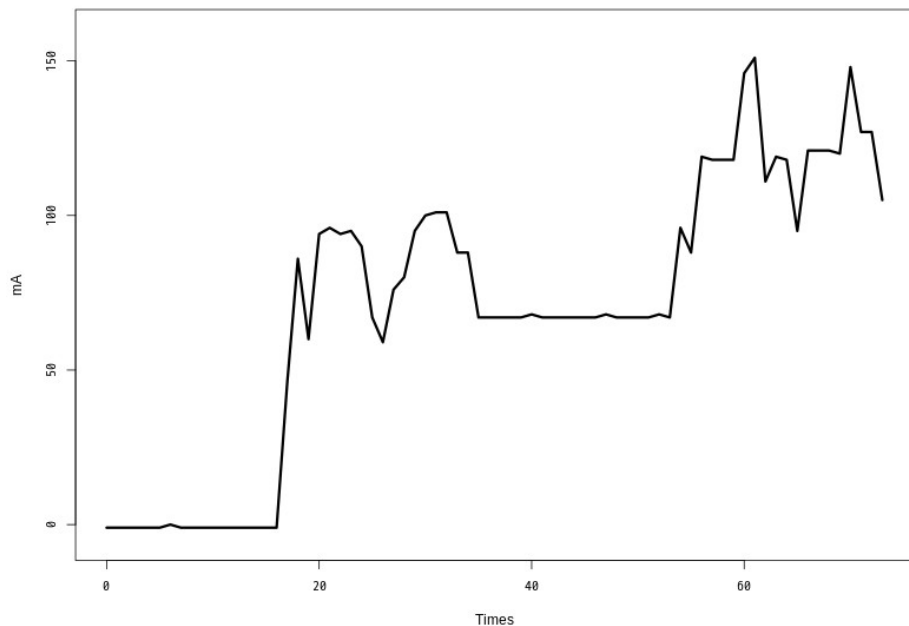
- Rでこんな風処理しています。

```
#!/usr/bin/env Rscript
args = commandArgs(trailingOnly=TRUE)
mydata=read.table(args, header=T)
png(filename=paste(args,".png",sep=""), width=800,height=600)
plot (
  mydata$Times, mydata$Number,
  type="l", xlab="Times", ylab="mA", ylim=c(0,260), lwd=3,
  main="ラズパイB2の電源ONからLXDE、そして電源OFFまで"
)
dev.off()
```

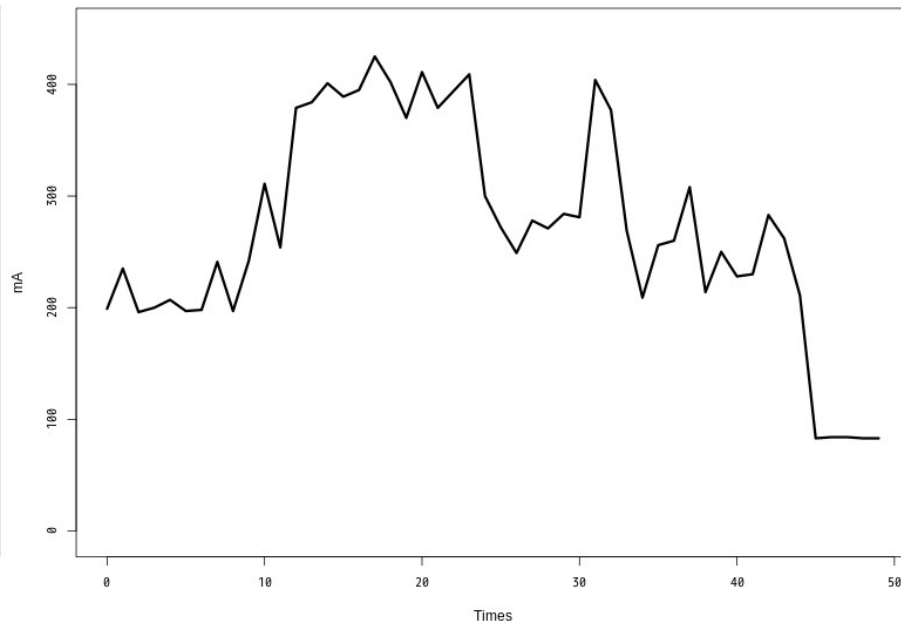
# ラズパイ起動時とか終了時とか

- グラフにしてみました

ラズパイB2の起動

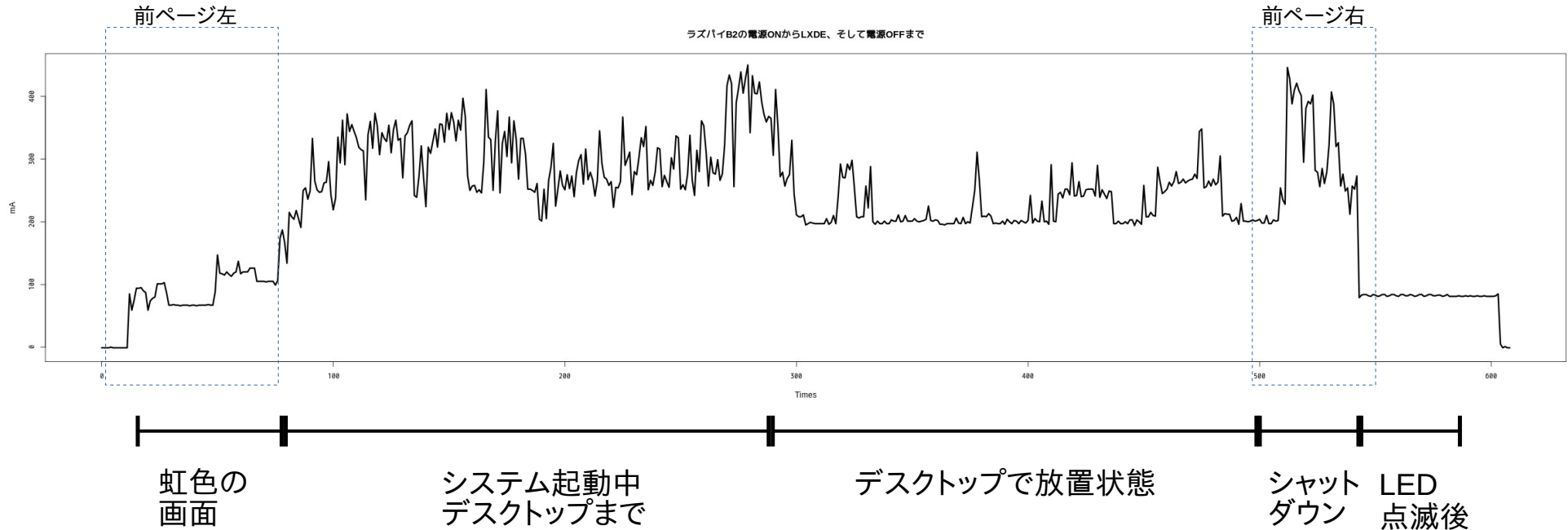


ラズパイB2のLXDEでシャットダウン



# はじめからしまいまで

- 動作状態はだいたいこんなふうなはず



# 最近、話すのが下手に

- ご清聴に感謝します。

続