

チャレンジ偏光観測 気分は天文学者

—2019年南米皆既日食、ものづくり編—

2019/10/05

田中辰治

自己紹介&やりたいこと

- ・ 職無、金無、ストレス無。プレミアムエブリディは天文ボランティアに全力投球?
- ・ 星より機材、日食より工作。金なくとも時間有、地上最悪上野の空で天文やるぜ
- ・ 美しい写真は他人にまかせた。サイエンスしたい、おじさんの自由研究は偏光観測だー!

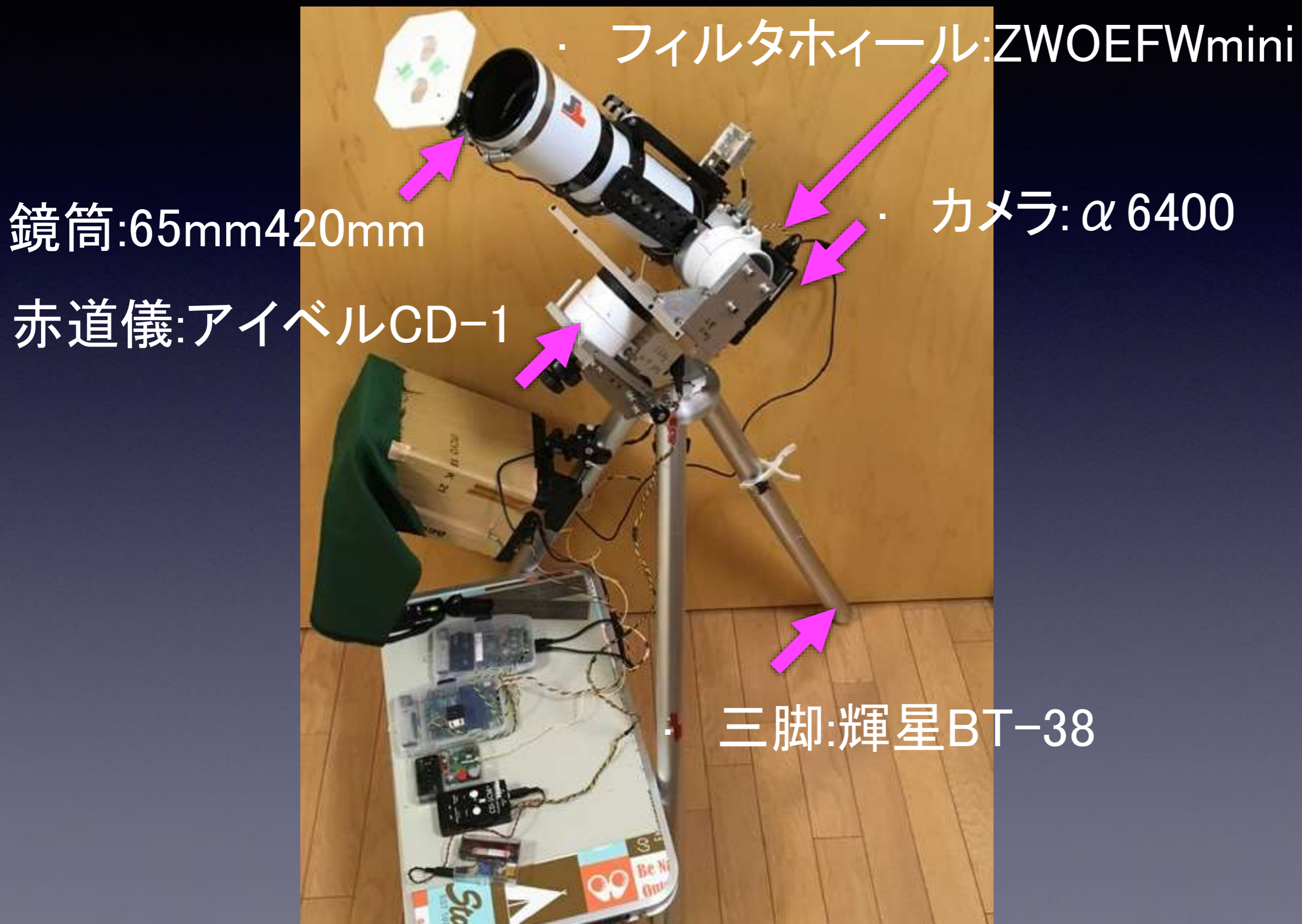


マンション廊下、戦う冷やしカメラと5nmフィルタ

偏光観測とは

- ・ Kコロナ(太陽自身による)の明るさ(エネルギー量)を測りたいが、太陽周りのチリで反射した光(Fコロナ)が混ざってる→**偏光フィルタで分離可!**(プールのキラキラを消すあのフィルタ)
- ・ **無偏光、0度、60度、120度偏光の4つを段階露出、しかも複数セット。(5段x4フィルタx5セット)**
- ・ フラット、ダーク等校正用データも含め数百枚必要。→**効率的な自動化が必須(PC,マイコン等)。**

主要機材



鏡筒:65mm420mm

赤道儀:アイベルCD-1

・ フィルタホイール:ZWOEFWmini

・ カメラ:α 6400

・ 三脚:輝星BT-38

コントローラ比較と選択

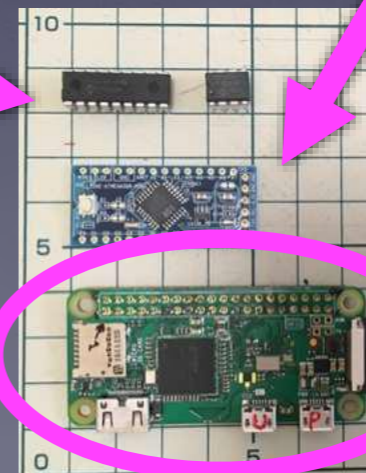
他にも種々環境があるが私に扱えるのは下記くらい

	PIC	Arduino	RaspberryPi
形態(代表例)	IC(6-100pin)	FRISKサイズ基板<	FRISKサイズ基板<
歴史	1975~	2005~	2012~
開発言語	アセンブラ、C	C風	"Linux"環境
デジタルI/O例	16(16F88)	14(Uno)	26
価格(例)	200円<	1000円<	1000円<
長所	安い、小さい	拡張性、開発容易	"Unix"資産 楽USBホスト
短所	開発難、要書込機	拡張すると高価	電源ぶち切厳禁

[https://ja.wikipedia.org/wiki/PIC_\(コントローラ\)](https://ja.wikipedia.org/wiki/PIC_(コントローラ))

<https://ja.wikipedia.org/wiki/Arduino>

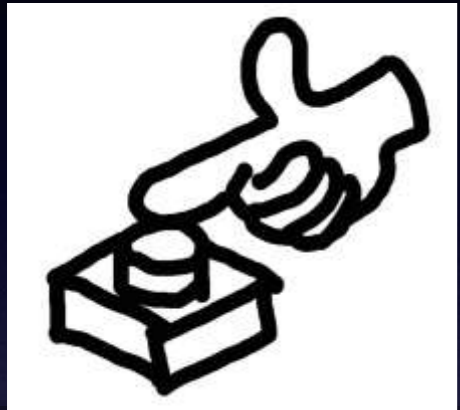
https://ja.wikipedia.org/wiki/Raspberry_Pi



一番小さい
ゼロWを使用

ハードウェア

5段階 * 4フィルタ * 5回繰り返す。



100枚/76秒

コンピュータ
RPI



リモート端子へ
撮影指示



5段階撮影

SONY
α 6400

INDIコマンド
回転終了



フィルタホイール
ZWO EFWmini

INDIコマンド
フィルタ変更

撮影終了

ストロボ接点から
パルス

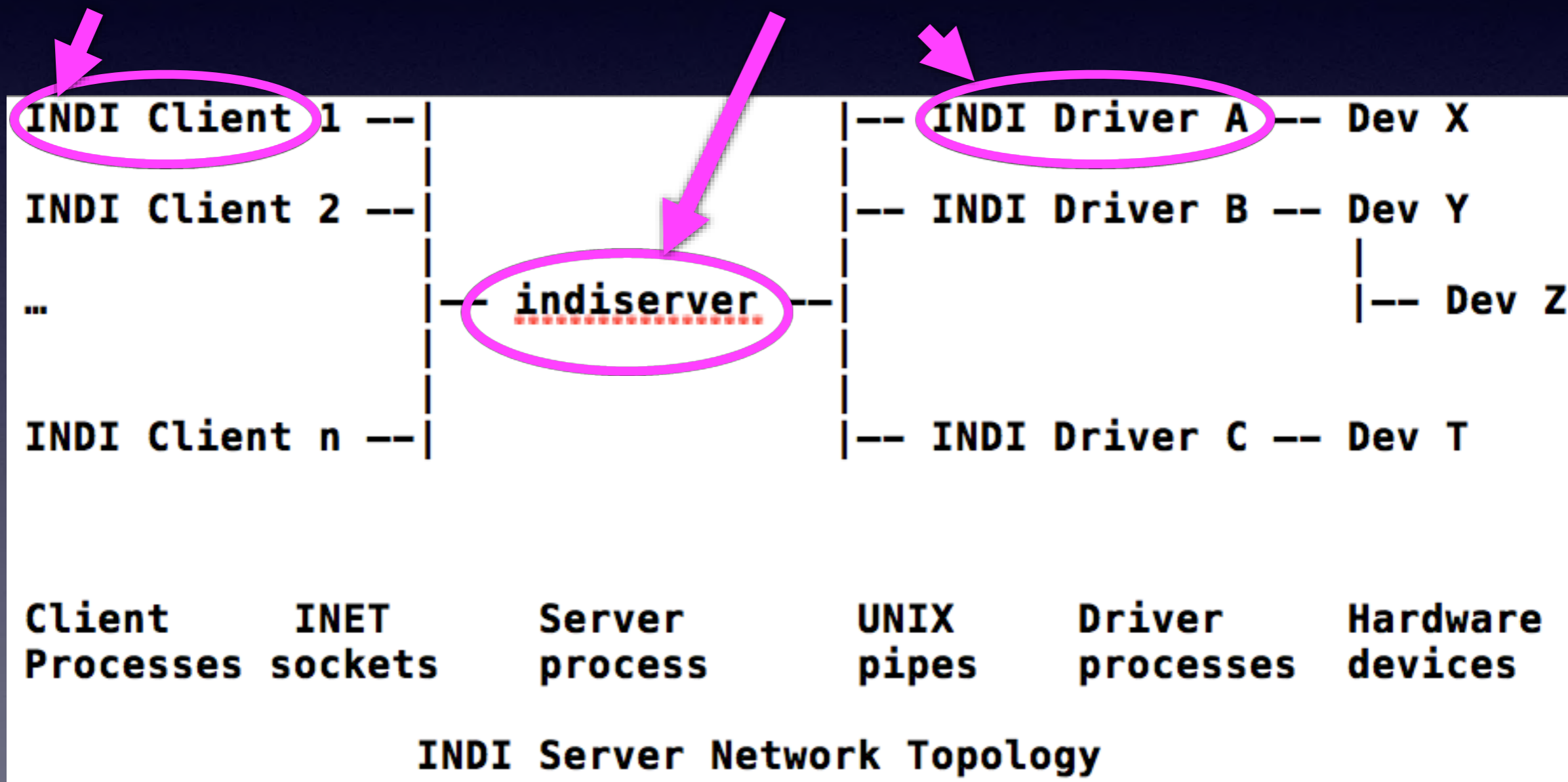


カウンタIC
74HC4017

プログラムは
bashとPython3

INDI ?

- INDIはWindowsのASCOMみたいなUNIX上の共通プラットフォーム(かな?)
- 赤道儀、カメラ、ドーム、ピント調整、フィルターホイールなどを制御
- クライアント(アプリケーション)、サーバ、ドライバ(デバイス)間をやりとり。



ソフトウェア？ (たったこれだけ)

カメラに撮影指示。撮影完了はICで知る。

LED点灯プログラムをコールし、スイッチ入力待ち。スイッチ押すと、メインにもどる

```
#!/usr/bin/env python3

#raspi0のための小さい基板用
#20190228
#SW_PINをハードウェアでプルアップ、pi.PUD_DOWNをやめた
#論理を逆転
#20190227作成
#SW_PIN 5->17
#LED_PIN 同一
import wiringpi as pi, time

SW_PIN = 17
LED_PIN = 22

pi.wiringPiSetupGpio()
#pi.pullUpDnControl( SW_PIN, pi.PUD_DOWN )
pi.pinMode( SW_PIN, pi.INPUT )
pi.pinMode( LED_PIN, pi.OUTPUT )
pi.digitalWrite( LED_PIN, pi.HIGH )

while True:
    if ( pi.digitalRead( SW_PIN ) == pi.LOW ):
        pi.digitalWrite( LED_PIN, pi.LOW )
        break
    else:
        pi.digitalWrite( LED_PIN, pi.HIGH )
        time.sleep(0.01)
```

Python3:
LED、スイッチ制御

```
#!/usr/bin/env bash
export LANG=en_US
#start INDI server, and wait three sec
sudo indserver indi,asi,wheel &
sleep 3s
#setup of EFW
indi_setprop "ASI EFW.CONNECTION.CONNECT=On"
indi_setprop "ASI EFW.CONNECTION.DISCONNECT=Off"
#back ground
jobs

#set filter1
indi_setprop "ASI EFW.FILTER.SLOT.FILTER.SLOT.VALUE=1"
# (loop if the last char(-1), of message is NOT 1)
while [[ $files: -1 != '1' ]]
do
    files=$(indi_getprop "ASI EFW.FILTER.SLOT.FILTER.SLOT.VALUE")
    echo $files: -1
done

#def EFW function as F_ROT() to change filter #1->2->3->4)
#call rem_sw8n26.py(RasPi trig. DSLR) #1 got proper Filter#

function F_ROT()
{
    i=1
    # final ver.
    while [ $i -lt 5 ]
    # temp ver.
    # while [ $i -lt 3 ]
    do
        indi_setprop "ASI EFW.FILTER.SLOT.FILTER.SLOT.VALUE=$i"
        while [[ $files: -1 != $i ]]
        do
            files=$(indi_getprop "ASI EFW.FILTER.SLOT.FILTER.SLOT.VALUE")
            echo $files: -1
        done
        #call Python script rem_sw8n26.py n.b->required full path for systemd
        sudo python3 /home/pi/rem_sw8n26.py

        #echo $i
        i=$((i+1))
    done
    #set filter first # for next trig
    indi_setprop "ASI EFW.FILTER.SLOT.FILTER.SLOT.VALUE=1"
}

#main
while :
do
    #LED_ON, and waiting for push-SW ON(call python script sw_7802.py)
    #n.b->required full path for systemd
    sudo python3 /home/pi/sw_7802.py
    #start timer
    SECONDS=0
    #call FIL_ROT function
    final ver.
    for i in {0..4}; do
    # temp ver.
    # for i in {0..1}; do
        F_ROT
    done
    #get time
    echo $SECONDS
done
```

bash:
メイン(フィルタ制御)

```
#!/usr/bin/env python3
import wiringpi as pi, time

#複数行コメント始
#lighter bdでRasPi0W用
#17リングカウンタ使用。
#*2EVを1セット(6shots:1/1000-1sec)。
#X接点のライズエッジを6回数えブラケット撮影を完了する。
#GPIOピンの定義。
REM_PIN EOSのリモート端子。RasPiからの撮影指示。LoでEOSをトリガー(ここではLoは200ms)
EOSのX接点(3.3Vプルアップ)シャッタの開閉毎にLo/Hを繰り返す。パルス幅は当然シャッタースピード依存。
EOC_PIN カウント完了。ブラケット撮影終了。
MSR_PIN 4017のMRピン。最初Hi。Loの間クロック有効。
最初にREM_PINをHiにしておく。そうしないとなんか変。2Vくらいになってる。

#複数行コメントここまで
...

REM_PIN = 27
EOC_PIN = 4
MSR_PIN = 6

pi.wiringPiSetupGpio()
pi.pinMode( REM_PIN, pi.OUTPUT )
pi.pinMode( EOC_PIN, pi.INPUT )
pi.pinMode( MSR_PIN, pi.OUTPUT )

pi.digitalWrite( REM_PIN, pi.HIGH )
time.sleep(0.1)
pi.digitalWrite( MSR_PIN, pi.HIGH )
time.sleep(0.1)
pi.digitalWrite( MSR_PIN, pi.LOW )
time.sleep(0.1)
pi.digitalWrite( REM_PIN, pi.LOW )
time.sleep(0.2)
pi.digitalWrite( REM_PIN, pi.HIGH )

while True:
    if ( pi.digitalRead(EOC_PIN) == pi.HIGH ):
        pass
        break
    else:
        pass
        time.sleep(0.01)
```

Python3:カメラ制御

eclipse.sh

rem_sw8n26.py

Python3:カメラ制御

モニタ、キーボードレス自動起動

WindowsのAUTOEXEC.BAT同様、電源ONで起動するsystemd使用。

/usr/lib/systemd/system/**eclipse.service**というテキストファイルに

[Unit]

Description=eclipse shoting raspberry pi by GPIO button input

Wants=network.target

[Service]

ExecStart=/home/pi/eclipse.sh

Restart=on-failure

RestartSec=10s

[Install]

WantedBy=multi-user.target

を書いて

sudo systemctl start **eclipse**

sudo systemctl enable **eclipse**

で準備完了

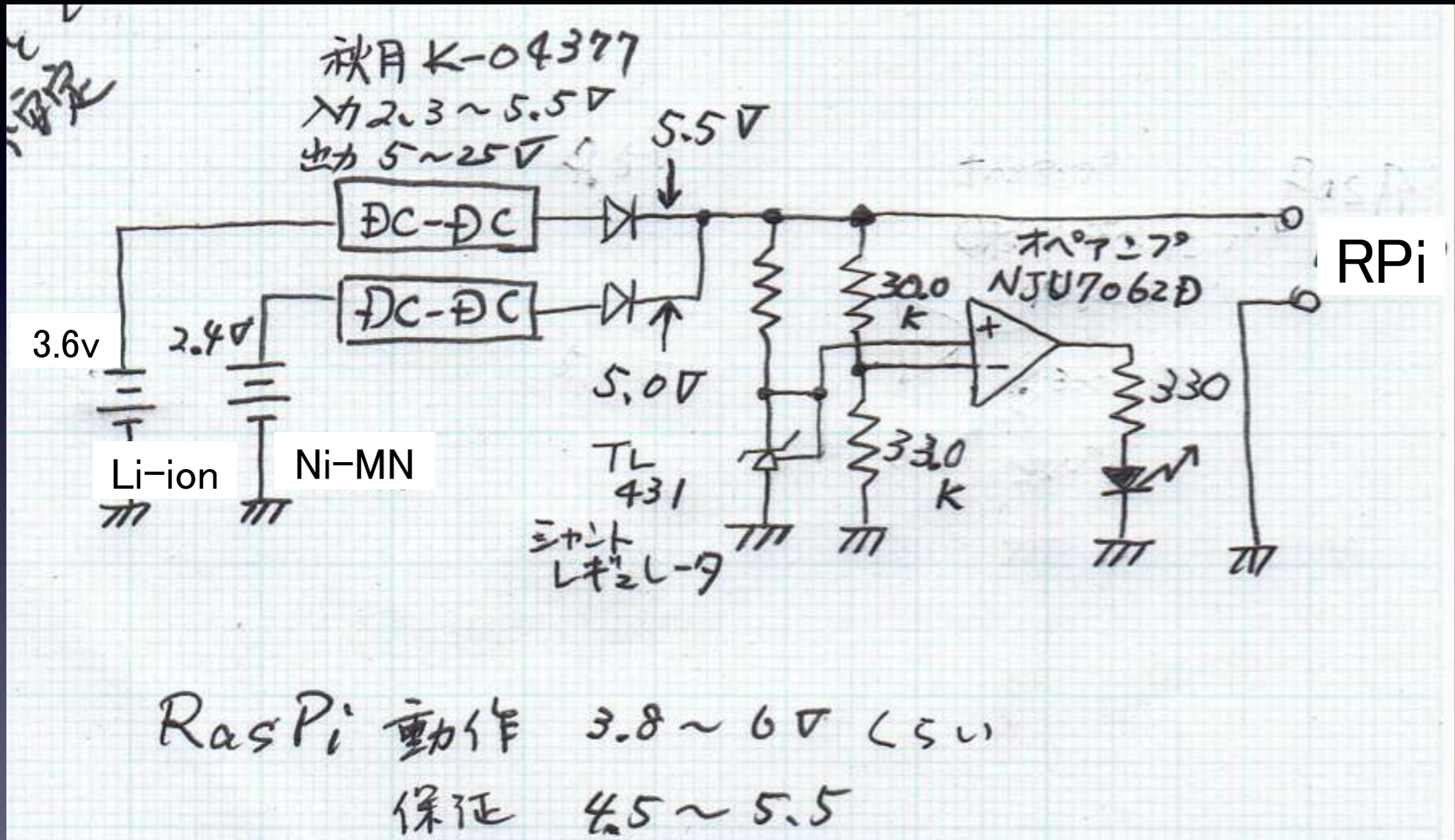
依存関係とかいろいろ設定
できる。でもまねたら動いた
のでよし。

<http://hammmm.hatenablog.com/entry/2016/11/14/231337>

<http://enakai00.hatenablog.com/entry/20130917/1379374797>

<http://hendigi.karaage.xyz/2016/11/auto-boot/>

RPi用バックアップ電源

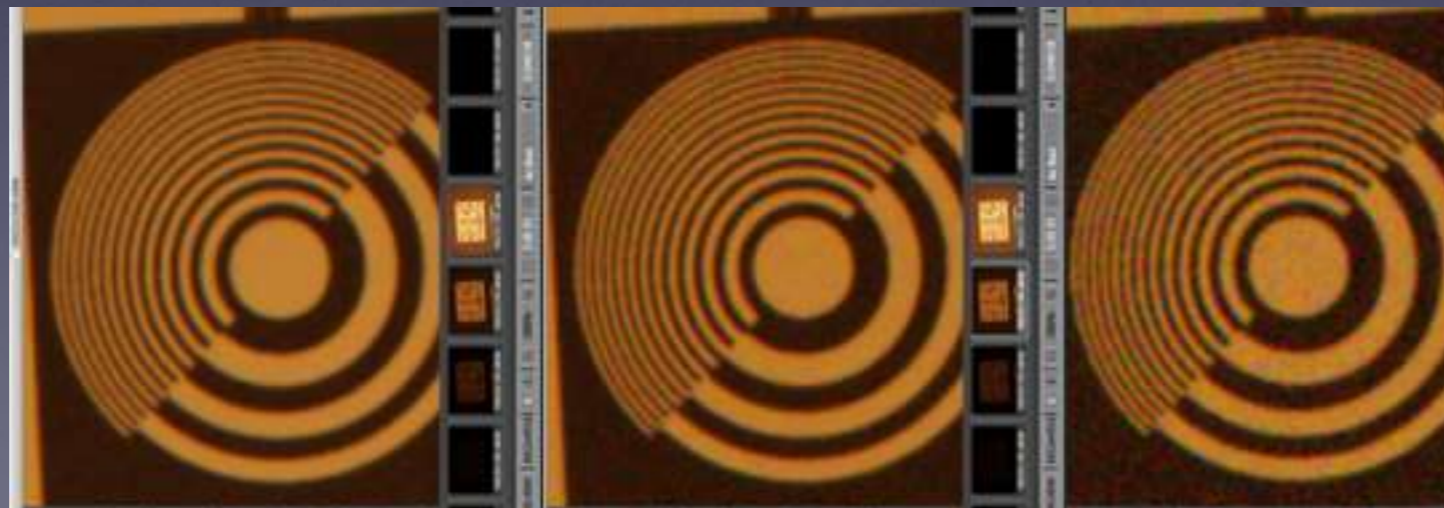


<http://scw.asahi-u.ac.jp/~sanozemi/Sakuhin/NiMH01/NiMH01.html>

RPiは電源ぶち切り禁止。万ーのためバックアップ必要

カメラの選定とブレ確認

- ・ 現有カメラ(キャノンX7i)はミラーショック大。
- ・ “ミラーアップ”すると時間かかる(しかも都度アップ)。
- ・ やっぱミラーレスだよな。1/1000~1/4秒(5段階露出)、14bit RAW、APS-CのSONY α 6400を選定。
- ・ 10m先のテストパターンでブレ確認し、問題はなさそう。



1/4(ISO200)

1/15(ISO800)

1/60(ISO3200)

カメラストロボ接点改造



74HC4017AP使用

論理図
真理値表
タイミングチャート
はデータシート参照

シューアダプタとリモコンで撮影完了を知るケーブル作成

偏光フィルタと無偏光フィルタ

- ・ 偏光／無偏光が素通しでは透過光が異：露出差
- ・ 偏光／無偏光でガラス厚差あると：差あるとピントずれ



- ・ 無偏光をNDフィルタ使用：透過量同一化
- ・ 厚さは2.244mm対2.204mm：ピント差13um



ピント用外付け7インチモニタ

7インチモニタ

段ボールに植毛紙

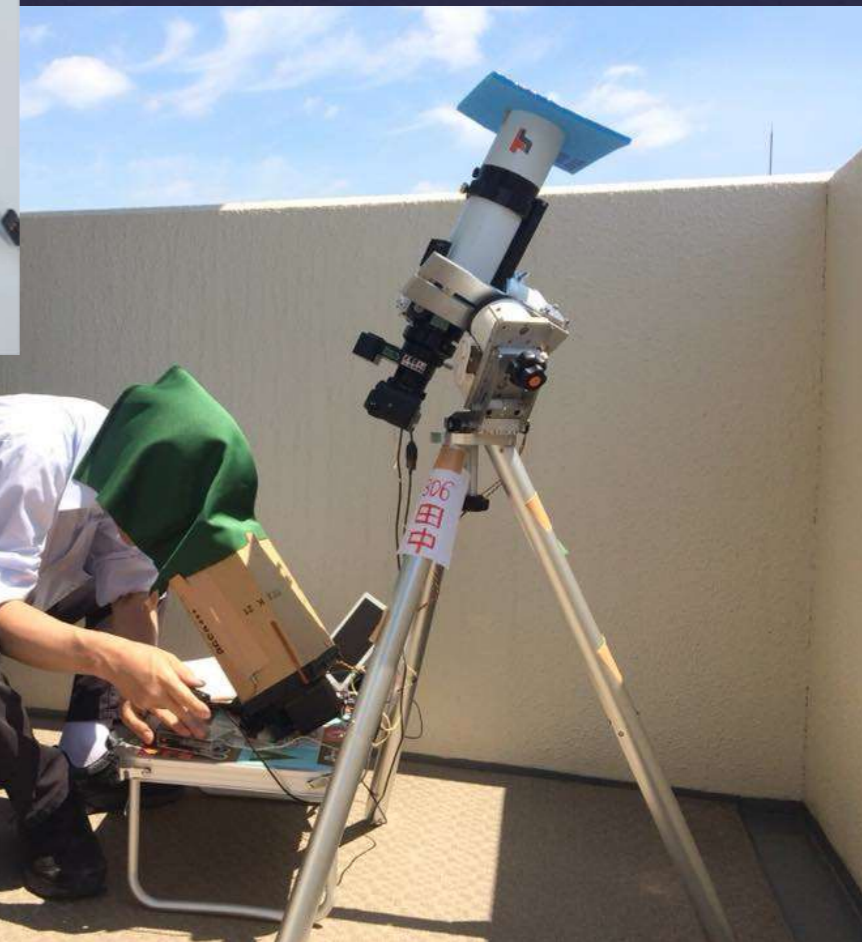
冠布

組み上げた状態

バッテリー

クランプ

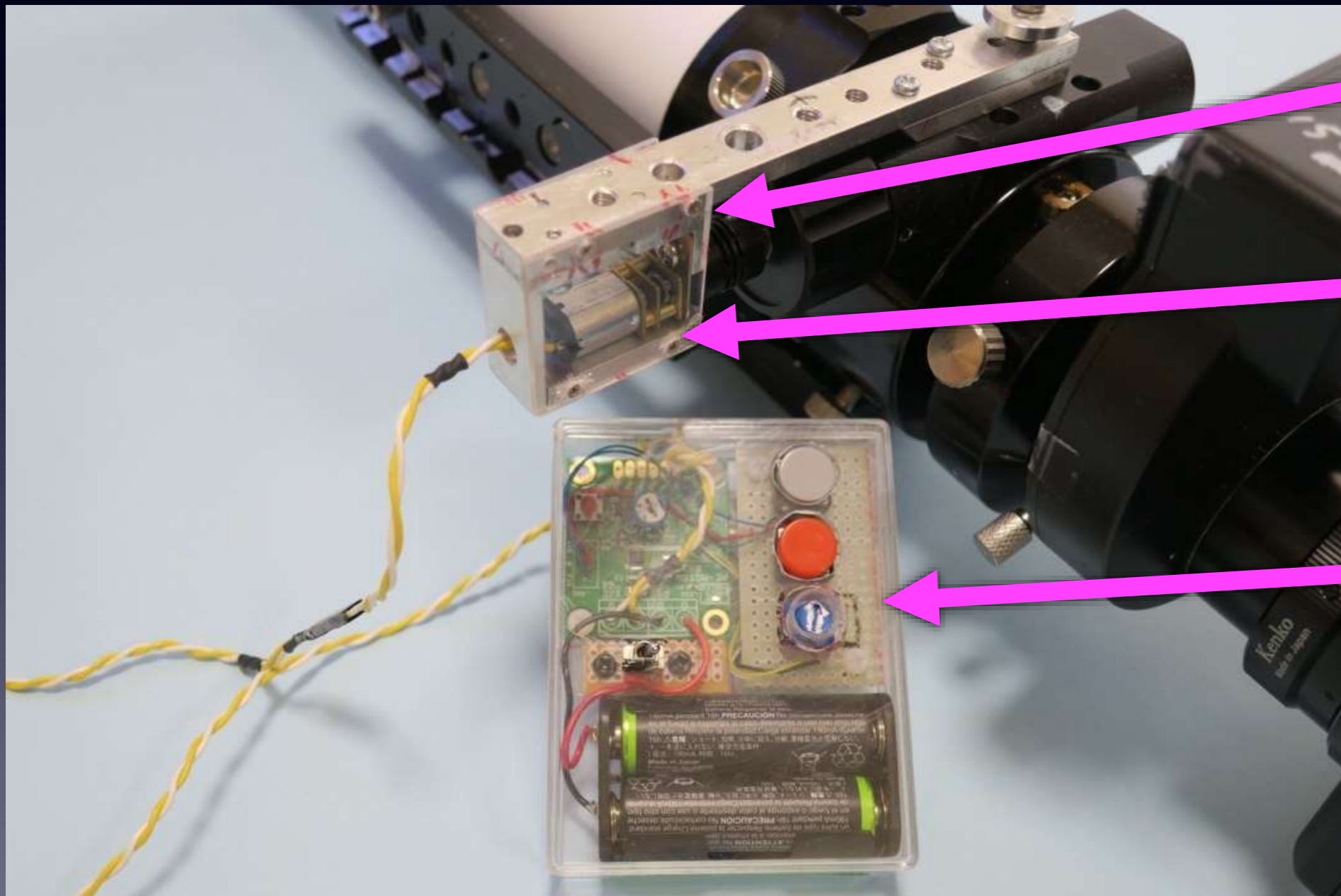
2X拡大レンズ



https://www.amazon.co.jp/gp/product/B077D2HNTW/ref=ppx_yo_dt_b_search_asin_image?ie=UTF8&psc=1

ピント合わせの電動化

ピント合わせ時にカメラが動いて合わせにくいので電動化

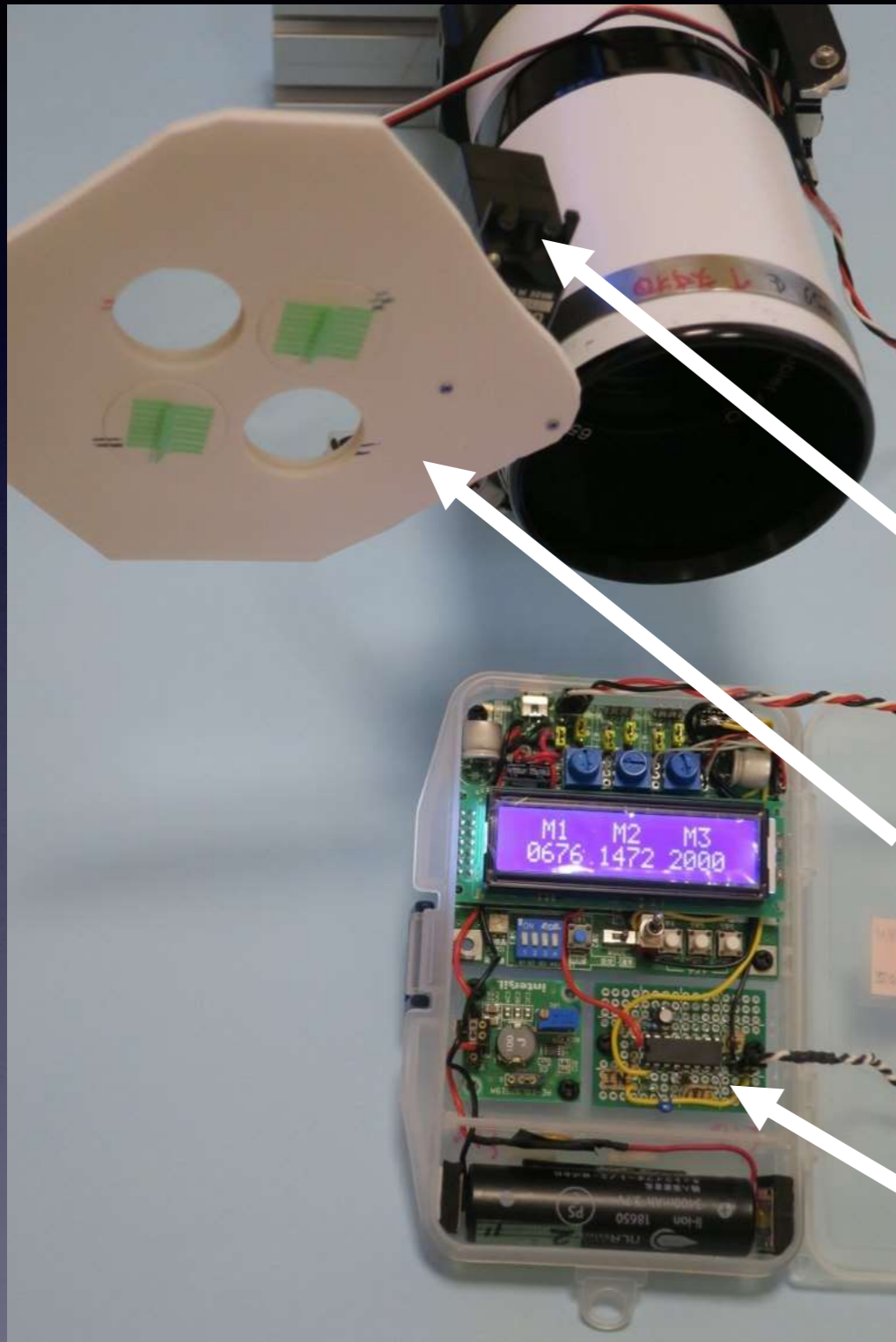


ケース:アルミと
プラスチック

DCギアモータ

秋月のDCモータド
ライバキット

日食フィルタの開閉自動化



秋月電子の3サーボ・アクチュエーター・キットで日食フィルタ開閉を電動化し、さらに日食フィルタを開くとラズパイが撮影指示するよう連動した。(74HC221APの立上りエッジ利用)

サーボモータ

日食フィルタはブタの鼻(ハルトマンプレート風)

ボタンの代わりに74HC221AP

74HC221AP使用

論理図
真理値表
タイミングチャート
はデータシート参照

できあがった装置



↑
RPi Zero W基板

↑
サーボ用基板

↑
ピン用基板

成果？(実はいんちき)

DARK/FLAT/位置合/露出合無1セット



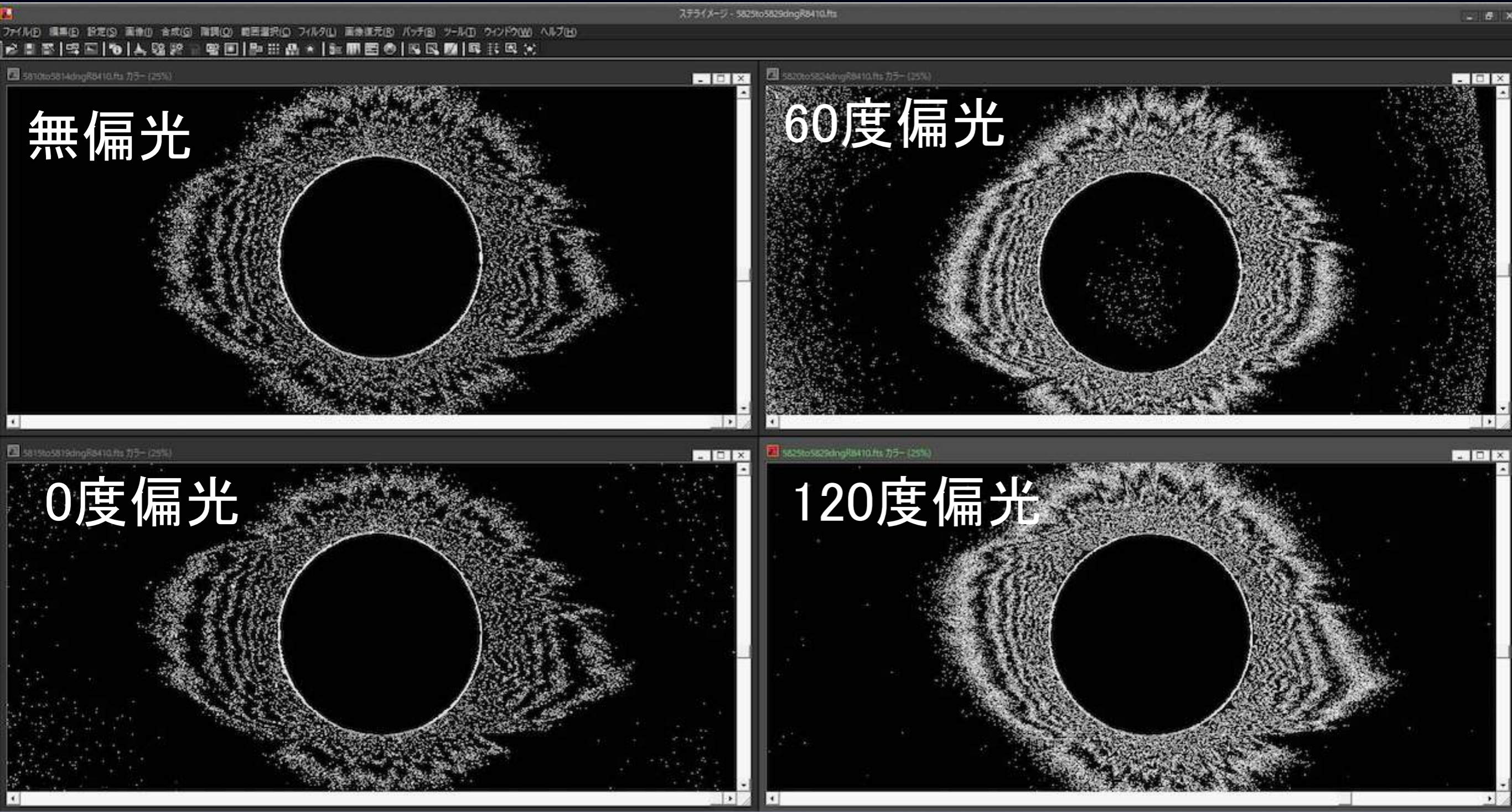
いんちきのいいわけ1

DARK/FLATの向きがおかしくなった



いんちきのいいわけ2

キッチンリフォームで画像処理どころではなかった



今後の計画(ものづくり依存症)

- ・ GPSで自動化 (めざせノータッチエクリップス!)
- ・ 基板一体化 (配線ダラケ対策)
- ・ フィルタと保存の高速化 (敵は2023豪州にあり)
- ・ Python3で統一化 (頭の体操:認知症対策)
- ・ 軽量化(荷物重量制限と腰痛対策)

その前に画像解析ししっかりやれよ!



ありがとうございました 旅のご酒飲帳

Whisky

Pisco

CERVEZA

