

# 改良したUSBオシロスコープについての追記

2023年8月

このUSBオシロスコープを紹介してから大分時間が経過した。最近、動作させようとしたがいろいろと忘却したこともあり、再始動に手間取ってしまった。このUSBオシロスコープは非常に汎用性があり、現時点でも有能な測定器であると思っているので、その再紹介とその「忘備録」を兼ねて本報告書を書くこととした。なを、本オシロスコープの詳細については後記している参考文献を参照すること。

本オシロスコープは2つの動作モードで使用できる。1つ目の動作モードは通常のオシロスコープとして (=モード1)。2つ目は積算機能を持たせ、雑音成分の除去機能を持たせての動作 (=モード2)。装置の立ち上げの初期画面で動作モードを選択する。

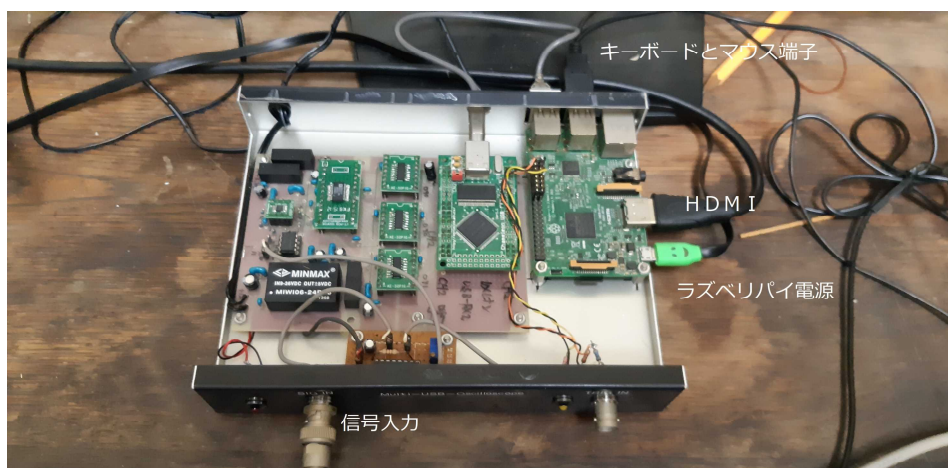


写真1 このように配線し、電源を入れるとLCD画面は写真2となる。

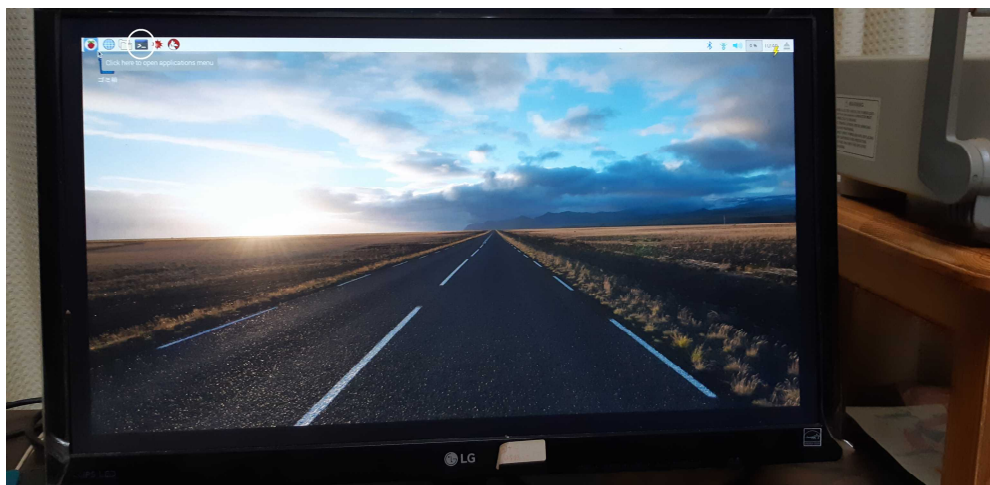


写真2 スタート画面である。左上端の白丸のアイコン (LXTerminal) をクリックする。Linuxが駆動し、写真3の画面となる。中央のウィンドウが実行領域となる。

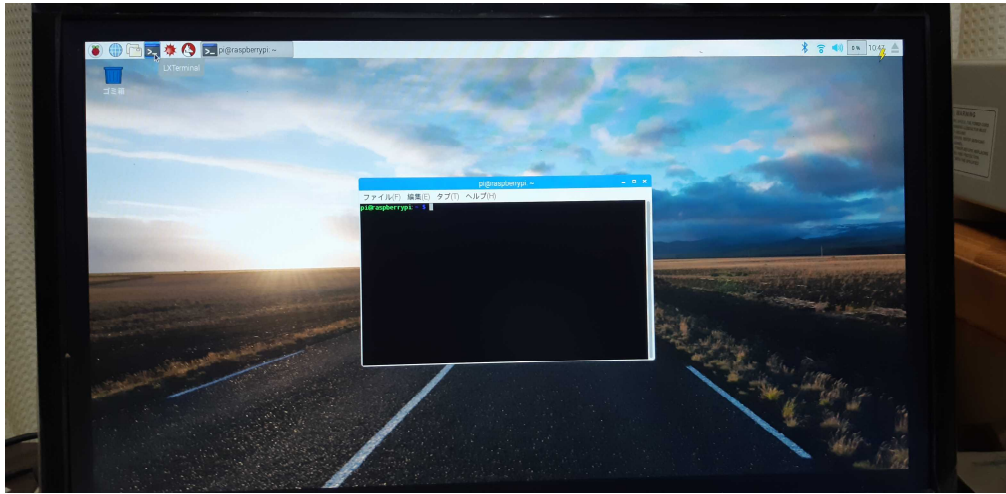


写真3 Linux実行画面

ここで、ルート・ディレクトリを表示させると、写真4のように、幾つものサブ・ディレクトリが表示される。

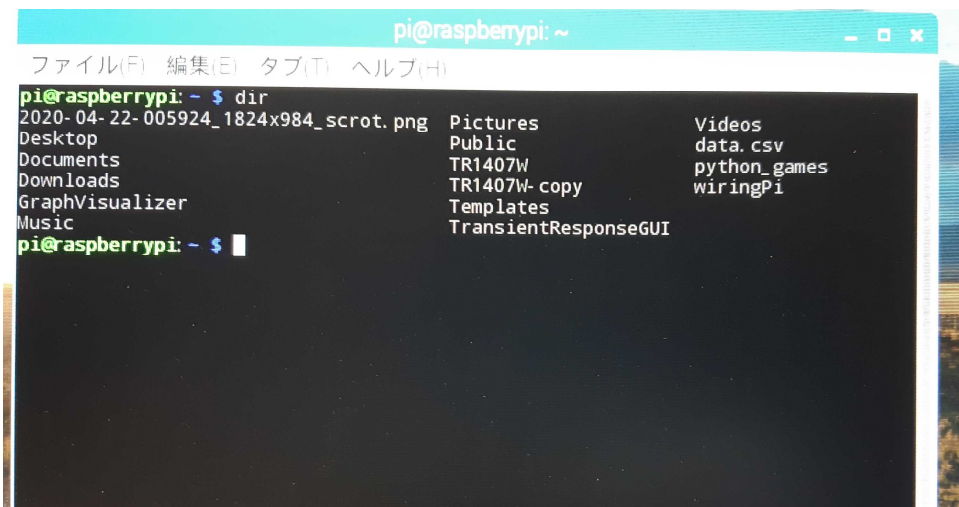


写真4 ルート・ディレクトリの内容

1. 「通常のUSBオシロスコープ」として使用する場合。つまりモード1としての動作  
**TR1407Wのディレクトリに入る。**

```
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi: ~ $ dir
2020-04-22-005924_1824x984_scrrot.png  Pictures          Videos
Desktop                               Public            data.csv
Documents                             TR1407W          python_games
Downloads                             TR1407W-copy    wiringPi
GraphVisualizer                       Templates
Music                                  TransientResponseGUI

pi@raspberrypi: ~ $ cd TR1407W
pi@raspberrypi: ~/TR1407W $ dir
Readme.txt ソース一式 ラズベリーパイ実験用オシロスコープ説明書.txt
pi@raspberrypi: ~/TR1407W $ cd ソース一式
pi@raspberrypi: ~/TR1407W/ソース一式 $ dir
include pld pld_src src 配布ソフト
pi@raspberrypi: ~/TR1407W/ソース一式 $ cd src
pi@raspberrypi: ~/TR1407W/ソース一式/src $ dir
Makefile bench_in.c bench_in.ihx build osc_gl osc_gl.c
pi@raspberrypi: ~/TR1407W/ソース一式/src $ sudo chmod +x build
pi@raspberrypi: ~/TR1407W/ソース一式/src $ sudo ./build
osc_gl.c: In function 'WinCreate':
osc_gl.c:198:2: warning: implicit declaration of function 'graphics_get_display_size' [-Wimplicit-function-declaration]
    success = graphics_get_display_size(0, &width, &height);
               ^
osc_gl.c: In function 'main':
osc_gl.c:566:2: warning: implicit declaration of function 'bcm_host_init' [-Wimplicit-function-declaration]
    bcm_host_init();
               ^
osc_gl.c: In function 'thread2':
osc_gl.c:864:31: warning: passing argument 3 of 'usb_bulk_read' discards 'volatile' qualifier from pointer target type
    ret = usb_bulk_read(dh, EP6, buf[area], FByte, TIMEOUT);
                               ^
In file included from osc_gl.c:7:0:
/usr/include/usb.h:306:5: note: expected 'char *' but argument is of type 'volatile unsigned char *'
    int usb_bulk_read(usb_dev_handle *dev, int ep, char *bytes, int size,
               ^
osc_gl.c:879:1: warning: control reaches end of non-void function [-Wreturn-type]
}
^
osc_gl.c:851:5: warning: 'dev' is used uninitialized in this function [-Wuninitialized]
    dev=USB_find(bus, dev);
               ^
pi@raspberrypi: ~/TR1407W/ソース一式/src $ sudo cycfx2prog prg: bench_in.ihx
Using ID 04b4:8613 on 001.006.
Putting 8051 into reset.
Programming 8051 using "bench_in.ihx".
pi@raspberrypi: ~/TR1407W/ソース一式/src $ sudo ./osc_gl
```

写真5 ディレクトリTR1407Wに入ったならば、画面のように進行させていく。

- (1) ディレクトリTR1407W→ソース一式→srcに入っていく。
- (2) sudo chmod +x build リターン
- (3) sudo ./build リターン
- (4) sudo cycfx2prog prg: bench\_in.ihx リターン
- (5) sudo ./osc\_gl オプション リターン



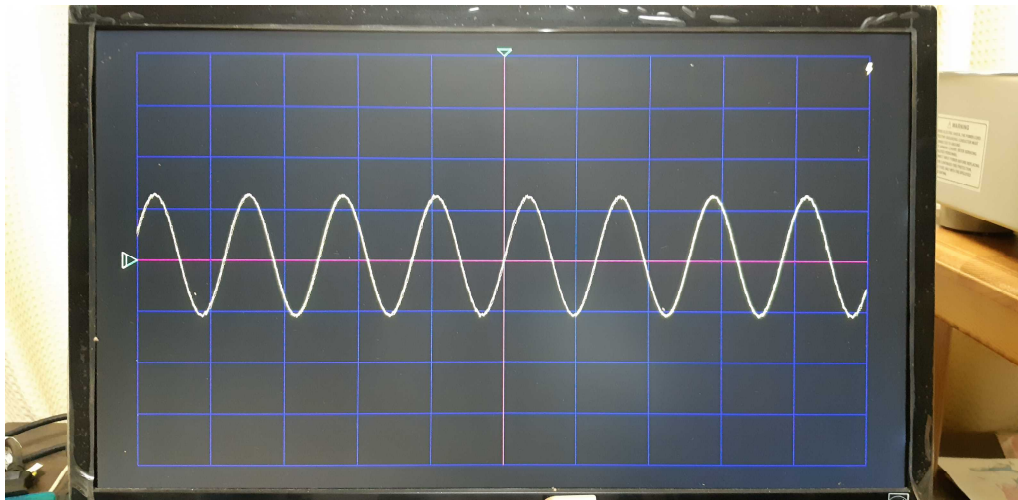


写真6 `sudo ./osc_g1` オプション リターンで測定画面となり、被測定信号が描写され続ける。発振器からのサイン波を入力端子に入れている。横軸スケール、縦軸スケール、トリガレベル、その他はデフォルト値となっているので、必要に応じて設定する必要がある。実行の終了は **cntl** キー + **C** キーのダブル押し。これで `Linux` モードに戻る。

表4 オシロスコープ用プログラムで設定できる内容

設定項目	設定できる値	オプション	記述例
時間レンジ [s]	1 $\mu$ s, 2 $\mu$ s, 5 $\mu$ s, 10 $\mu$ s, 20 $\mu$ s, 50 $\mu$ s, 100 $\mu$ s(デフォルト値), 200 $\mu$ s, 500 $\mu$ s, 1 ms, 2 ms, 5 ms, 10 ms, 20 ms, 50 ms, 100 ms, 200 ms, 500 ms	-x*	-x10us
電圧レンジ(縦軸)	100 mV, 200 mV, 500 mV, 1 V, 2 V(デフォルト値), 5 V	-v*	-v2v
トリガ・レベル	デフォルト値は0.0. $\pm 10$ Vの間で小数点以下まで設定可	-t*	-t-2.53
トリガ・エッジ検出方法	1: 立ち上がり(デフォルト値) 0: 立ち下がり	-e*	-e1
オフセット・レベル	デフォルト値は0.0. $\pm 10$ Vの間で小数点以下まで設定可	-y*	-y-5.50
トリガ・ポジション	デフォルト値は0.0. $\pm 5$ の範囲	-p*	-p-3.0
サンプリング・クロック	1 MHz, 2 MHz(デフォルト値)	-s*	-s1MHz

表1 設定のオプション `sudo ./osc_g1 *** **` リターンの\*\*\*の所に記述例に習って入力する。

外部トリガモードで動作するオシロスコープとして使用する場合

トリガ入力端子にトリガ信号を入力させる。トリガ信号としては矩形波が望ましい。かつ、電圧振幅は被測定信号振幅より大きくしておく必要がある。被測定信号と、このトリガ信号は加算された信号として処理されていく。トリガのオプションでトリガ電圧の正負、大きさに電圧レベルを設定すれば、このトリガレベルでトリガが確実にかかることになる。

測定例として、写真7に過渡減衰波形とその発現矩形波をトリガパルスにした場合の測定表示画面を示している。

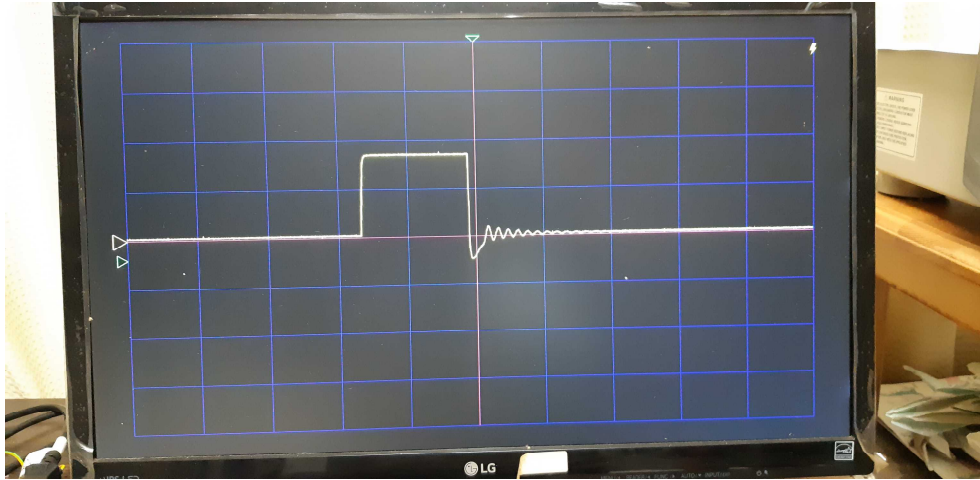


写真7 測定例

2. 積算機能を有するUSBオシロスコープとして使用する場合。つまりモード2としての動作。モード2での動作をさせるには、ルート/ディレクトリからTR1407 W-c o p yのディレクトリに入る。

「s i g i n」端子に被測定信号を、「t r i g i n」端子にトリガ信号を接続する。モード1での同じ手順を踏んでいく。写真8の通り。

被測定信号モデルとしては、既報の場合と同じモデルとした。即ち、LC共振回路においてスイッチング時に生じる過渡減衰波形を測定し抽出する。写真7で示している中心部から右側に流れている波形である。これに雑音をわざと混入させる。つまり、過渡現象波形+雑音を被測定信号とする。その様子が写真10に示されている。

```

ファイル(F) 編集(E) タブ(I) ヘルプ(H)
pi@raspberrypi: ~ $ dir
2020-04-22-005924_1824x984_scrot.png  Pictures          Videos
Desktop                               Public            data.csv
Documents                             TR1407W          python_games
Downloads                              TR1407W-copy     wiringPi
Graphicalizer                         Templates
Music                                  TransientResponseGUI

pi@raspberrypi: ~ $ cd TR1407W-copy
pi@raspberrypi: ~/TR1407W-copy $ dir
Readme.txt ソース一式 ラズベリーパイ実験用オシロスコープ説明書.txt
pi@raspberrypi: ~/TR1407W-copy $ cd ソース一式
pi@raspberrypi: ~/TR1407W-copy/ソース一式 $ dir
include pld pld_src src 配布ソフト
pi@raspberrypi: ~/TR1407W-copy/ソース一式 $ cd src
pi@raspberrypi: ~/TR1407W-copy/ソース一式/src $ dir
-21-3      2023-7-7  4-21-5  bench_in.c  osc_gl      sss        wiringPi
2023-7-3   2023-7-8  4-22-1  bench_in.ihx osc_gl.c    test
2023-7-5   4-21-1   4-22-2  build       sample      test1
2023-7-6   4-21-2   Makefile example     ss          test2
pi@raspberrypi: ~/TR1407W-copy/ソース一式/src $ sudo chmod +x build
pi@raspberrypi: ~/TR1407W-copy/ソース一式/src $ sudo ./build
osc_gl.c: In function 'WinCreate':
osc_gl.c: 236:2: warning: implicit declaration of function 'graphics_get_display_size' [-Wimplicit-function-declaration]
    success = graphics_get_display_size(0, &width, &height);
    ^
osc_gl.c: In function 'main':
osc_gl.c: 612:2: warning: implicit declaration of function 'bcm_host_init' [-Wimplicit-function-declaration]
    bcm_host_init();
    ^
osc_gl.c: 408:6: warning: variable 'ch' set but not used [-Wunused-but-set-variable]
    int ch; //=====
    ^
osc_gl.c: In function 'thread2':
osc_gl.c: 1026:31: warning: passing argument 3 of 'usb_bulk_read' discards 'volatile' qualifier from pointer target type
    ret = usb_bulk_read(dh, EP6, buf+area, FByte, TIMEOUT);
    ^
In file included from osc_gl.c: 8: 0:
/usr/include/usb.h: 306:5: note: expected 'char *' but argument is of type 'volatile unsigned char *'
    int usb_bulk_read(usb_dev_handle *dev, int ep, char *bytes, int size,
    ^
osc_gl.c: 1041:1: warning: control reaches end of non-void function [-Wreturn-type]
}
^
osc_gl.c: 1013:5: warning: 'dev' is used uninitialized in this function [-Wuninitialized]
    dev=USB_find(bus, dev);
    ^
pi@raspberrypi: ~/TR1407W-copy/ソース一式/src $ sudo cycfx2prog prg:bench_in.ihx
Using ID 04b4:8613 on 001.006.
Putting 8051 into reset.
Programming 8051 using "bench_in.ihx".
pi@raspberrypi: ~/TR1407W-copy/ソース一式/src $ sudo ./osc_gl

```

写真8 LCD画面設定部。ディレクトリ「TR1407W-copy」に入り、後はモード1での手順と同じく進めていく。

```

pi@raspberrypi: ~/TR1407W-copy/ソース一式/src $ sudo ./osc_gl -x500us -t5v
Please input sum_counter = 2000
Please Input data_file_name = test1
RaspberryPI_Oscilloscope_Start

```

写真9 `sudo ./osc_gl` オプション リターンとすると、モード2では積算回数 (`sum_counter =` ) とデータ保存ファイル名 (`data_file_name =` ) を入力し、リターンキーを押すと、写真10のごとくの測定実行画面表示となる。

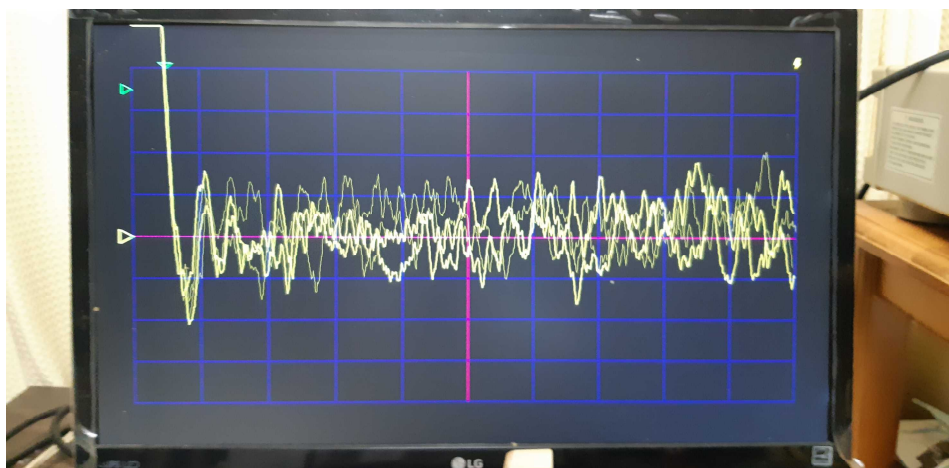


写真10 測定中

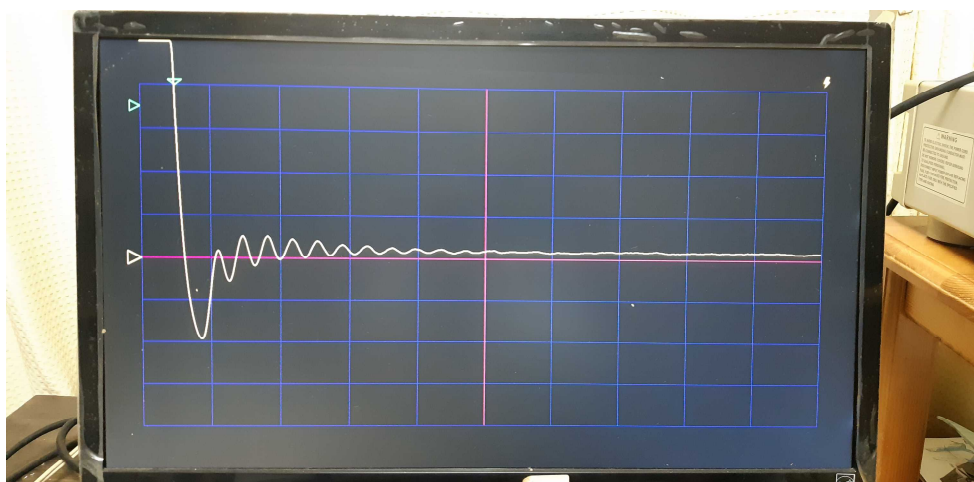


写真11 測定が終了し、結果を表示

入力した積算回数だけ測定すると結果グラフが表示され、実行停止となる。結果の測定値は指名したファイルに保存されている。この画面を終了したければ、`cntl`キー+`C`キーのダブル押しをする。結果を見て、測定条件指定オプションの表1を参考として、満足のいくグラフが得られるようにすれば良い。

### 参考文献

本ホームページの「研究結果の紹介 その2」で紹介しているものを列記する。

- (1) 48. 「マルチUSBオシロスコープの紹介」、2020年5月21日。
- (2) 47. 「USBオシロスコープの複製（2Mサンプル/s）の続編－微弱な過渡現象波形も測定できるUSBオシロスコープに改良」、2019年5月1日。
- (3) 46. 「USBオシロスコープの複製（2Mサンプル/s）」、  
2016年12月19日。