

マルチUSBオシロスコープの紹介 (別名 微弱な過渡現象信号も測定できる USBオシロスコープのための追記)

2020年5月20日

1. はじめに

2014年出版の参考文献(1)に掲載されていた、高速のサンプリング特性を持ったUSBオシロスコープを、2016年に著者はほぼ忠実に複製した。その内容を著者のホームページで公開した⁽²⁾。動作は良好であった。

その後、著者の研究において、雑音に埋もれてもいる微弱過渡減衰信号を取り扱う必要が生じた。複製したUSBオシロスコープそのままではどうにもならない。そこで、このUSBオシロスコープでも微弱過渡減衰信号を取り扱うことが出来るように改良を加えた。その改良版のUSBオシロスコープ(使用しているラズベリーパイは「Raspberr y P i 2 M o d e l B」であった)については、2019年に著者のホームページで公開した⁽³⁾。

改良版を「微弱な過渡現象信号も測定できるUSBオシロスコープ」と報告文では呼称していたが、長すぎるので、簡略化も考えて、以降では「マルチUSBオシロスコープ」と呼称を変更しよう。

USBオシロスコープの複製は3台を作った。1台は手元に、2台は共同研究先に提供した。研究課題の中途終了とともに、1台が著者の手元に返却されてきた。元々手元にあった1台は、研究目的に合うように、既に「マルチUSBオシロスコープ」に改良済みであった。戻って来たのは、単なる「USBオシロスコープ」。このままでは、宝の持ち腐れになると思い、この返却されてきた「USBオシロスコープ」を「マルチUSBオシロスコープ」に改良することにした。自分の参考文献(3)を手引きにすれば良いので、変更は容易であると考えていた。その通り、ハード部の変更は、比較的簡単に済んだ。が使用を開始してみると、**プログラムのコンパイル時に予想もしないエラーが発生し、実行プログラムが作成されないことが分かった**。しかし、色々試行錯誤した結果、正常に動作する2台目の「マルチUSBオシロスコープ」とすることが出来た。

原因はよく分からないが、ラズベリーパイで使用するOSであるRaspbianをインストールするインストーラーとも言える「**NOOBS**」のバージョンに起因したものであると考えている。最近入手できる「NOOBS_v3」では、実行するプログラム(プログラム名 o s c _ g l)のコンパイル時にエラーが発生(Open-GLに関するものらしい)するが、昔使用していた「NOOBS_v1」ならば、このエラーは出ず、コンパイルは正常に行われるのである。なを、ここで使用しているラズベリーパイは「Raspberr y P i 3 M o d e l B」である。

複製した「USBオシロスコープ」、或いは「マルチUSBオシロスコープ」を、現時点で正常に動作させるためには、以下の手順が必要のようである。ただし、使用しているラズベリーパイは「Raspberr y P i 2 M o d e l B」か「Raspberr y P i 3 M o d e l B」とする。

2. 必要なファイルなどのダウンロード

提供している圧縮フォルダをダウンロードし解凍し、当面適当な場所に保存しておく。図1に示しているように4つのフォルダが入っている。

(1)「NOOBS_v1」

ラズベリーパイ財団が提供していた古いバージョンのOSインストーラー。

(2)「wiringPi」

ラズベリーパイのGPIO関連のプログラムを作るためのwiringpi社のライブラリ。

(3)「TR1407W」

参考文献(1)で紹介されているUSBオシロスコープ動作のためのプログラム類。

(4)「TR1407W-copy」

著者が「TR1407W」を母親にして、微弱過渡現象波形等を測定できるように編集し直したプログラム類。実行プログラム(o s c _ g l)以外の内容は「TR1407W」と全く同じである。

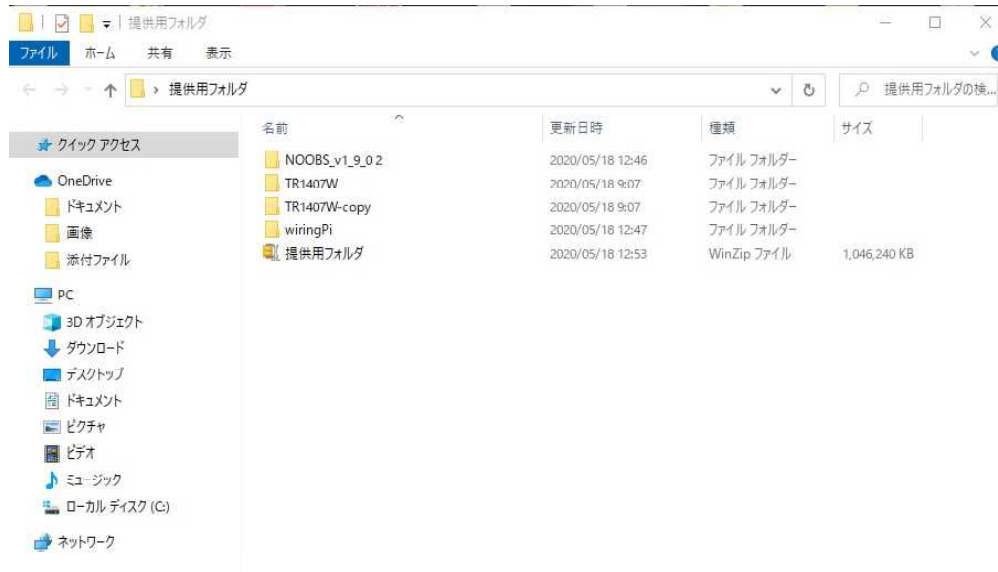


図1 提供している圧縮ファイルを解凍する。必要とする4つのフォルダがある。

3. ラズベリーパイの立ち上げ

- (1) パソコンに **Raspbian** の OS のインストーラー「**NOOBS_v1**」を用意する。著者が所有し、提供するフォルダ「NOOBS_v1」の内容は図2。
- (2) パソコンにマイクロSDを挿入する。使用済みの場合はフォーマットをした方が良い。
- (3) 図2に示している全フォルダ、全ファイルをマイクロSDカードにコピーする。
- (4) パソコンからマイクロSDカードを取り出し、ラズベリーパイに差し込む。

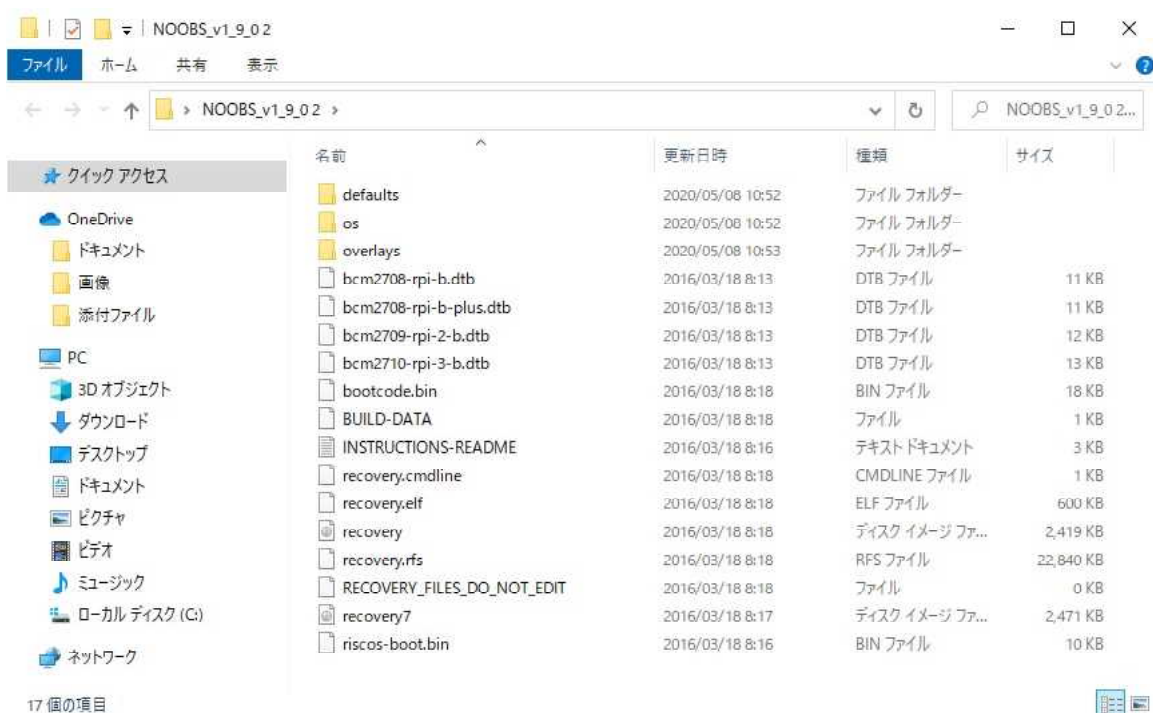


図2 フォルダ「NOOBS_v1」の内容。

(5) 参考文献(4)などに従って、Raspberry Piの初期設定などを行うが、**まだインターネットは未接続としておくこと**。理由は分からないが、この段階からインターネット接続がしてあると、後で、プログラムのコンパイルエラーを起こしてしまう。

(6) ラズベリパイの電源を入れると、図3の初期画面が表示される。左端のチェックボックスにチェックを入れ、能動となった左端の「install」をクリックすると、インストールが開始される。数十分はかかる。気長に待とう。

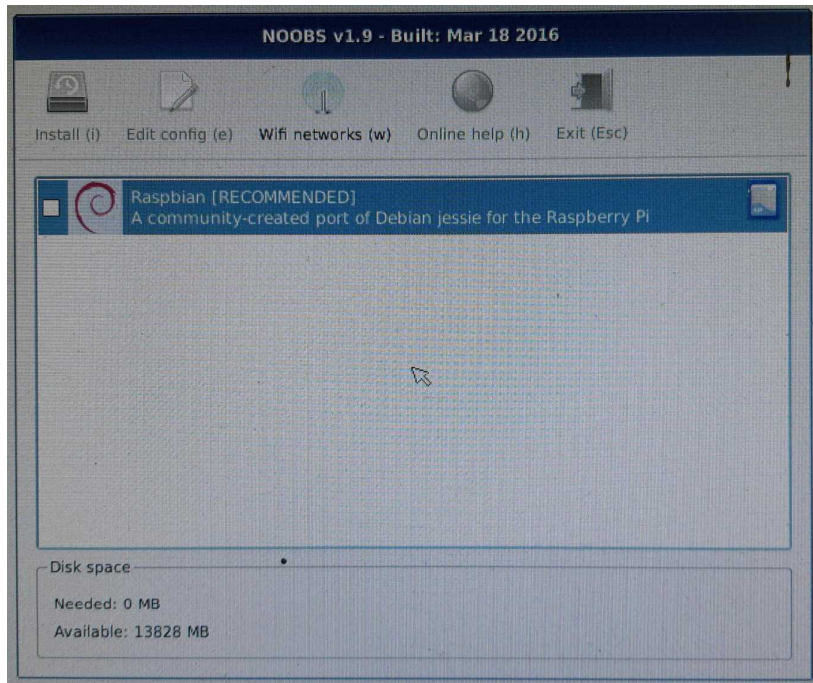


図3 Raspbianのインストールの初期画面

(7) OSのインストールが終了すると図4の画面となる。「OK」ボタンをクリックすると、ディスプレイ画面は暗転し、ラズベリパイの再始動が開始される。



図4 インストールの終了画面。

(8) 再始動が終了し、図5の画面となる。ラズベリパイのシステムが正常に立ち上がったことになる。

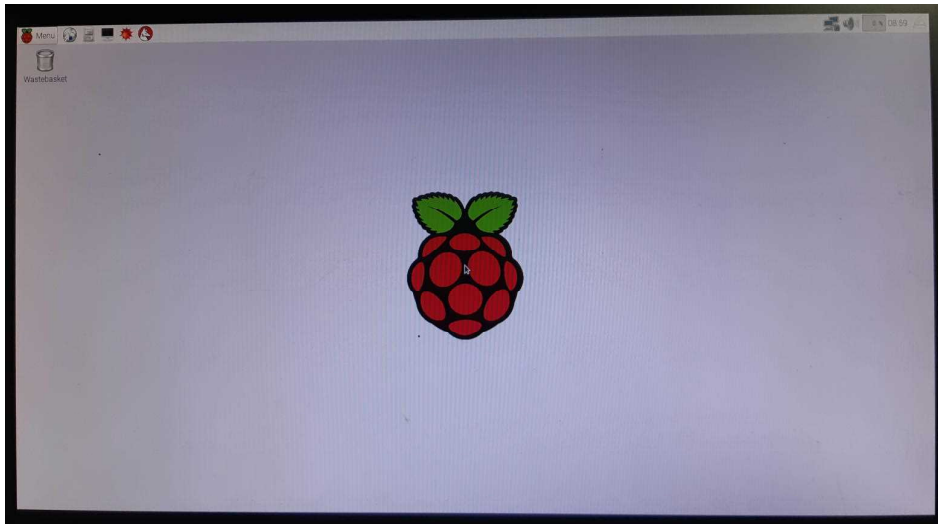


図5 始動画面

(9) 本システムを日本語版、日本語も使用できるように設定しておいた方が便利である。そのためには図5の画面で、左上部の「ターミナル」アイコンをクリックし、開いたターミナル画面で

```
$ sudo raspi-config ↵          (↵=リターン記号)
```

とする。後は、参考文献(4)を参考として「Raspi-configの設定」(41頁から50頁)を手引きに必要な項目について処理をしていく。終わったら、

(10) インターネットの接続をする。

(11) 次に、「Raspberry Piを日本語化しよう」(69頁から72頁)を手引きとして、処理を行っていく。

(12) そして、ラズベリーパイを再始動しよう。図6の如く、左上隅のアイコンが英語の「Wastebasket」から日本語の「ゴミ箱」に変わっている。

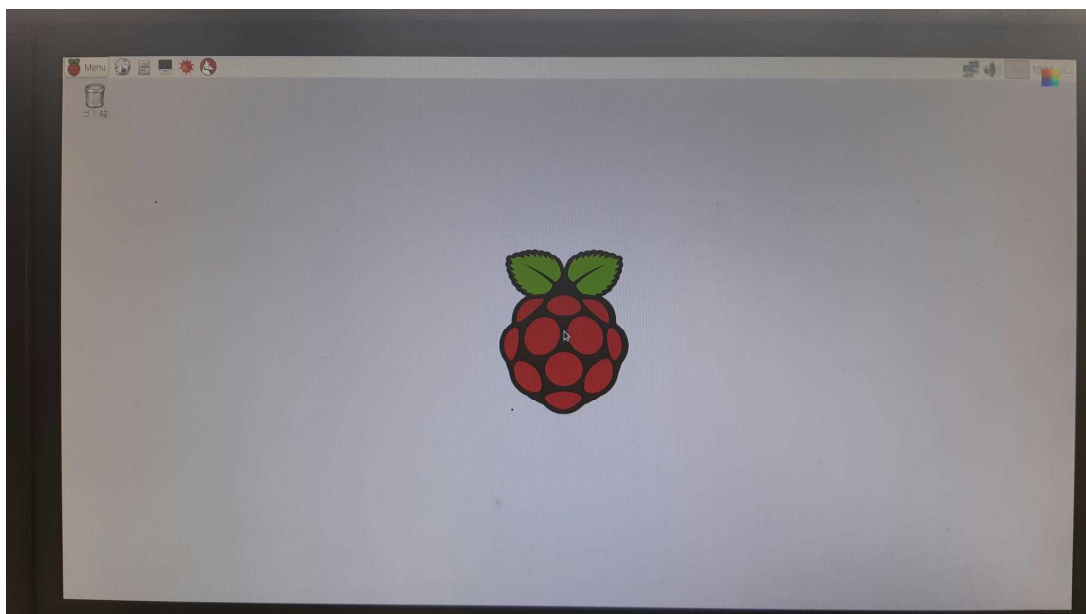


図6 日本語が使用できるようにしての始動画面。

以上で、ラズベリーパイをNOOBS_v1のOSでの立ち上げが完了したことになる。以降、ラズベリーパイを自由に使用できることになる。以降、参考文献(2)及び(3)を手引きにすれば、通常のUSBオシロスコープとしての動作及び微弱過渡現象波形測定可能USBオシロスコープとしての動作をさせることが出来る。

これを機会に以下で、「マルチUSBオシロスコープ」について、詳細に解説をしよう。今後において、大事な参考文献として書き残して置きたいからでもある。

3. 回路図と概観

図7に、「マルチUSBオシロスコープ」の回路図を示している。参考文献(2)に掲載している回路図に若干の変更を加えている。参考文献(2)の回路図では入力段に10:1の減衰回路を入れるのを忘れていた。ここで、1MΩの可変抵抗を用いて付加した。減衰率を微調整できるようにしている。参考文献(3)で紹介しているように、本装置を「マルチ」として利用できるようにするために、ゲート回路部を付加したことを、この回路図中で明記した。また、参考文献の(1)、(2)ではラズベリパイのGPIO端子は全く未使用であったが、マルチ動作ではラズベリパイのGPIO端子の幾つかに働いてもらうので、その部分も描いている。

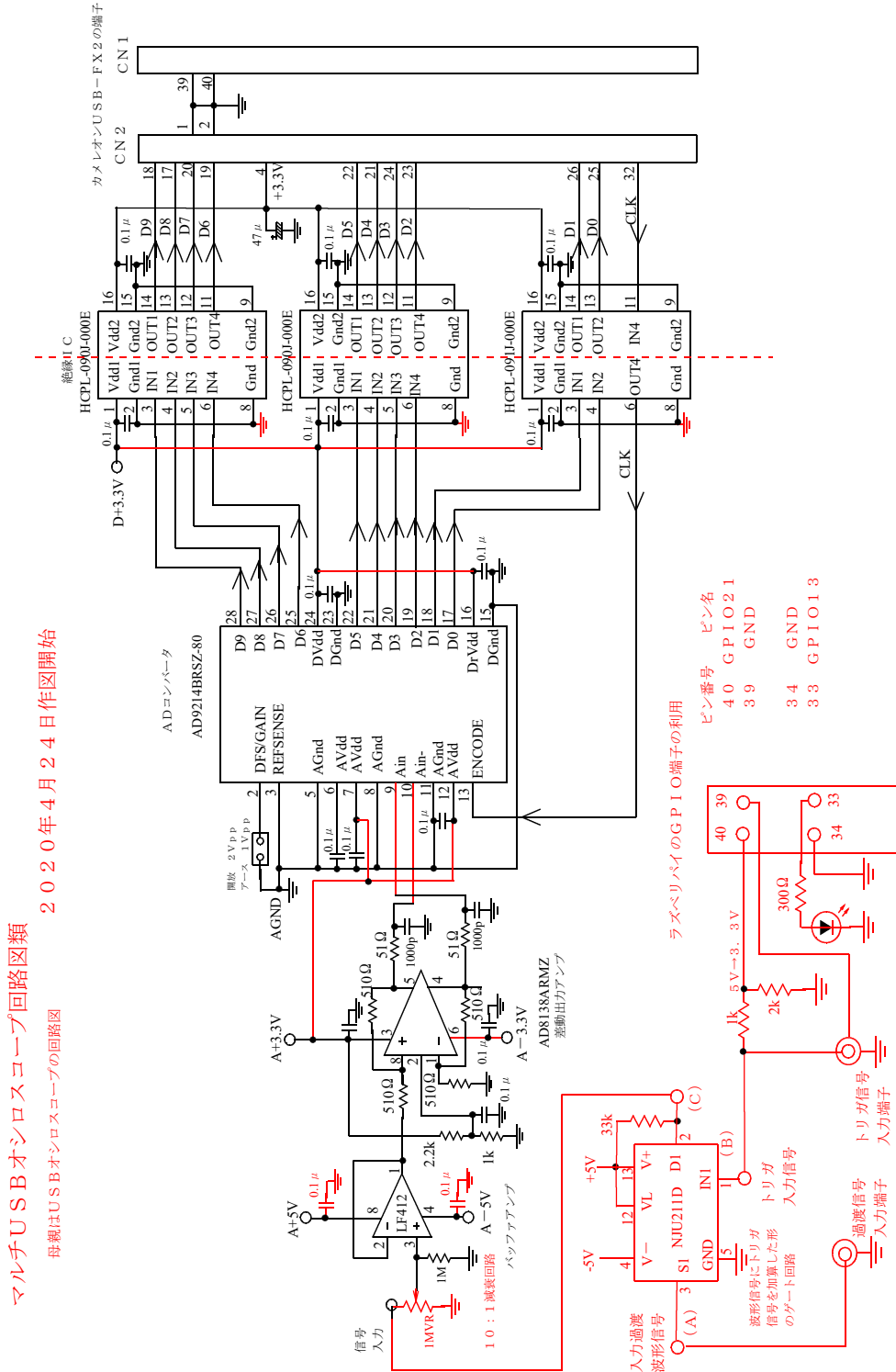


図7 「マルチUSBオシロスコープ」の回路図

4. マルチUSBとしての使用方法

図9に、基本的な使用における本システムの結合ブロック図を示している。

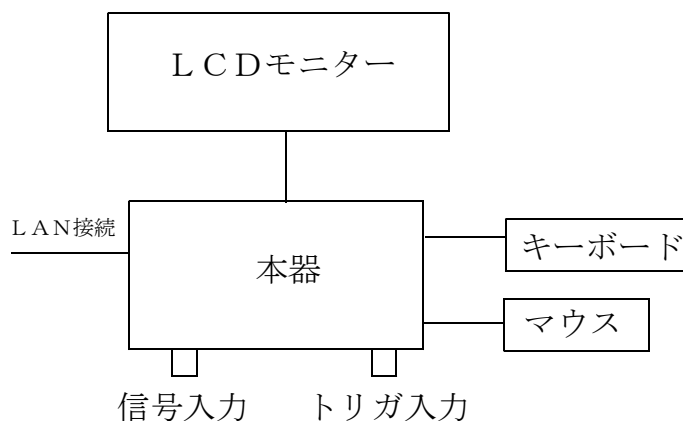


図9 基本的な結合ブロック図。

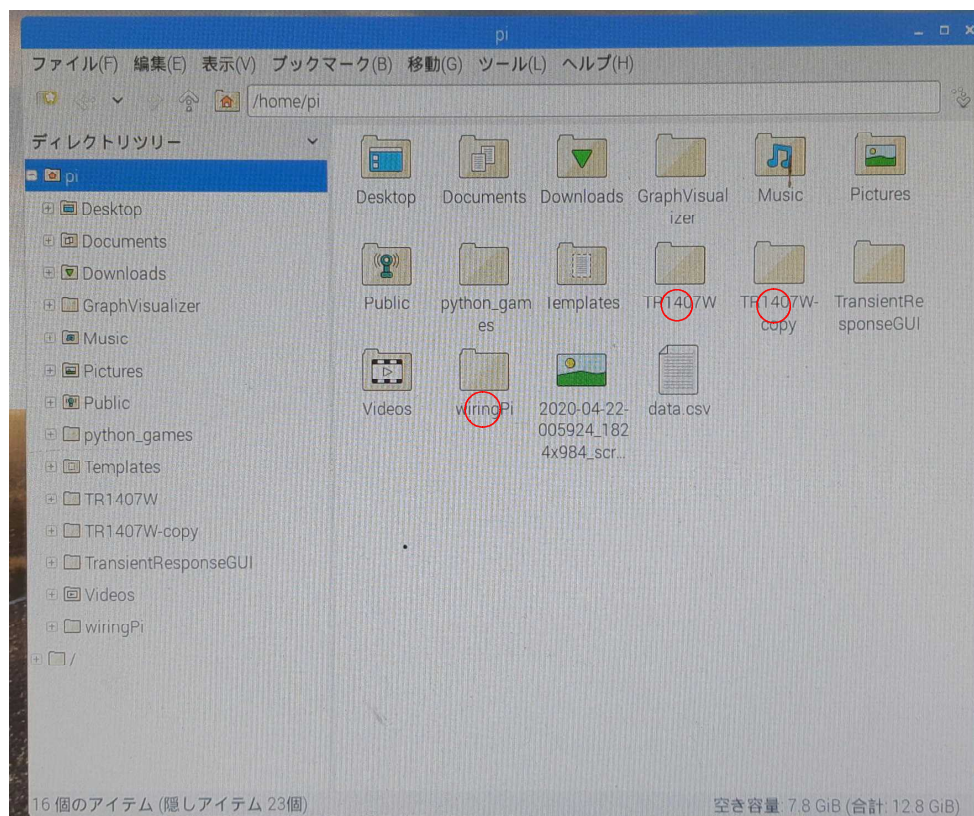


図10 ラズベリーパイを始動し、ディスプレイ画面に /home/pi のディレクトリを表示している。本システムで使用するために、解凍した残りの3つのフォルダ (TR1407W、TR1407W-copy、wiringpi) をここに配置する。赤丸で明示している。

4.1 普通のおシロスコープとしての使い方

図9で、信号入力端子に、測定したい信号を入力する。トリガ入力端子は未使用なので、何も接続する必要はない、解放のままで良い。タスクバーの左上にあるアイコン「L X T e r m i n a l」をクリックし、ターミナル画面を開く。以後この画面中でLinuxを用いて処理をしていく。その手順を参考文献(2)で紹介している内容のほぼ複写で以下に記述する。

(1) 最新バージョンの確認

```
$ sudo apt-get update ↵ (↵=リターン記号)
```

(2) libusbのインストール

```
$ sudo apt-get install libusb++ ↵
```

注意 文献では libusb ↵ だけであったが、現在では、インストールできない。ファイル名に変更が行われたらしい。上記の様に拡張子が必要。余分なファイルもインストールされる様であるが、致し方がない。Linuxに詳しくれば、libusbファイルの中から、本システムで必要なものだけを見つけ出せるのであろうが。

(3) cycfx2progソースコードのラズベリパイ版を入手(任意)

```
$ sudo apt-get install cycfx2prog ↵
```

(4) コンパイラの入手(任意)

```
$ sudo apt-get install sdcc ↵
```

(5) 現在のディレクトリを確認

```
$ dir ↵
```

サブディレクトリが列記される。その中にディレクトリ「TR1407W」が表示されているはず。(万一、そうでなければ、インストールした「TR1407W」をこの位置まで移動すること。)

(6) ディレクトリ「TR1407W」に、移動する。

```
$ cd TR1407W ↵
```

(7) ディレクトリ内の内容の確認

```
$ dir ↵
```

ディレクトリの内容が列記される。多分内容は

```
ソース一式 Readme ラズベリパイ実験用オシロスコープ説明書
```

(8) ディレクトリ「ソース一式」に、移動する。

```
$ cd ソース一式 ↵
```

(9) ディレクトリ内の内容の確認

```
$ dir ↵
```

ディレクトリの内容が列記される。多分内容は

```
include pld pld_src src 配布ソフト
```

(10) ディレクトリ「src」に、移動する。

```
$ cd src ↵
```

(11) ディレクトリ内の内容の確認

```
$ dir ↵
```

ディレクトリの内容が列記される。多分内容は

```
bench_in.c bench_in.ihx build  
Makefile osc_gl osc_gl.c
```

(12) 以降で、実行権限が必要となるので、以下のコマンドを入力する。

```
$ sudo chmod +x build
```


(13) 余計なメッセージがなければ、よし。以下のコマンドで、プログラムのコンパイルを実行する。

```
$ sudo ./build ↵  
画面にメッセージが列記され、コンパイル完了。
```

(14) `usbtest` をアンロードさせる。(必要ないかも?)

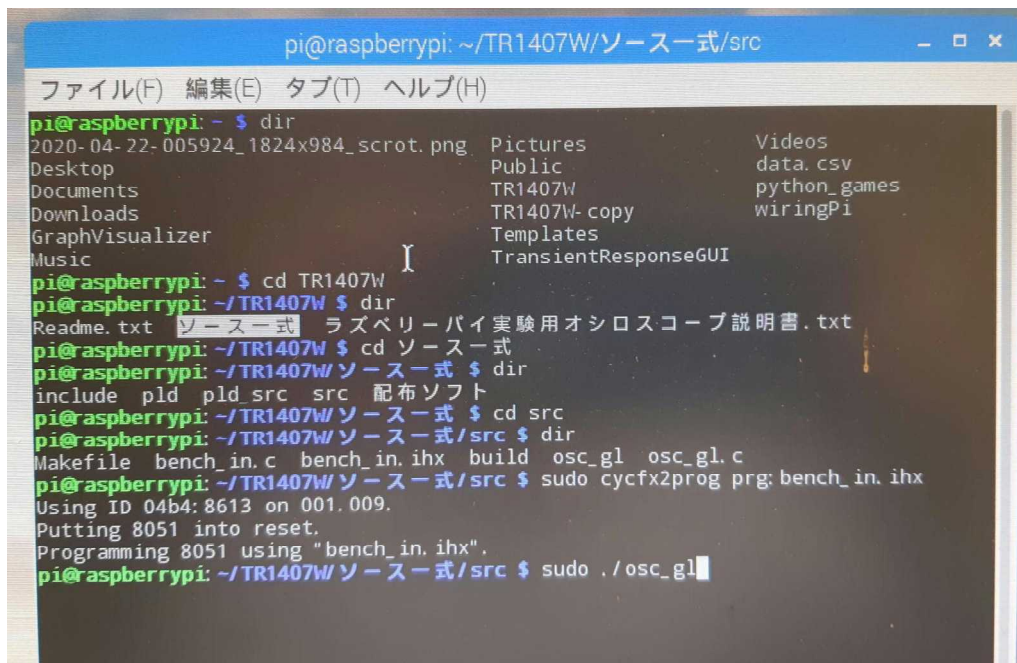
```
$ sudo modprobe -r usbtest ↵
```

(15) FX2ファームをダウンロードする。ファームのバイナリファイルのある「`bench_in.ihx`」のあるディレクトリで、以下のコマンドを実行する。

```
$ sudo cycfx2prog prg:bench_in.ihx ↵
```

(17) オシロソフトの実行

```
$ sudo ./osc_gl -** (起動オプション 表参照)  
オシロ画面が描写され、被測定信号も描写され続ける。
```



```
pi@raspberrypi: ~/TR1407W/ソース式/src  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi: ~ $ dir  
2020-04-22-005924_1824x984_screenshot.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 ソース式  
pi@raspberrypi: ~/TR1407W/ソース式 $ dir  
include pld pld_src src 配布ソフト  
pi@raspberrypi: ~/TR1407W/ソース式 $ cd src  
pi@raspberrypi: ~/TR1407W/ソース式/src $ dir  
Makefile bench_in.c bench_in.ihx build osc_gl osc_gl.c  
pi@raspberrypi: ~/TR1407W/ソース式/src $ sudo cycfx2prog prg:bench_in.ihx  
Using ID 04b4:8613 on 001.009.  
Putting 8051 into reset.  
Programming 8051 using "bench_in.ihx".  
pi@raspberrypi: ~/TR1407W/ソース式/src $ sudo ./osc_gl
```

図11 ターミナルモードとして、ディレクトリ「TR1407W」に移り、その後、参考文献(1)、或いは(2)に従って、被測定信号であるサイン波を表示するための手順を示している。最後の命令

```
$ sudo ./osc_gl エンターキー
```

とすれば、画面は次の図12の如くとなる。ただし、サイン波の周波数、電圧振幅に依存しているので適当な値に設定すれば良い。プログラムの実行時のパラメーター、オプションなどの設定の説明については、参考文献(1)などに譲る。

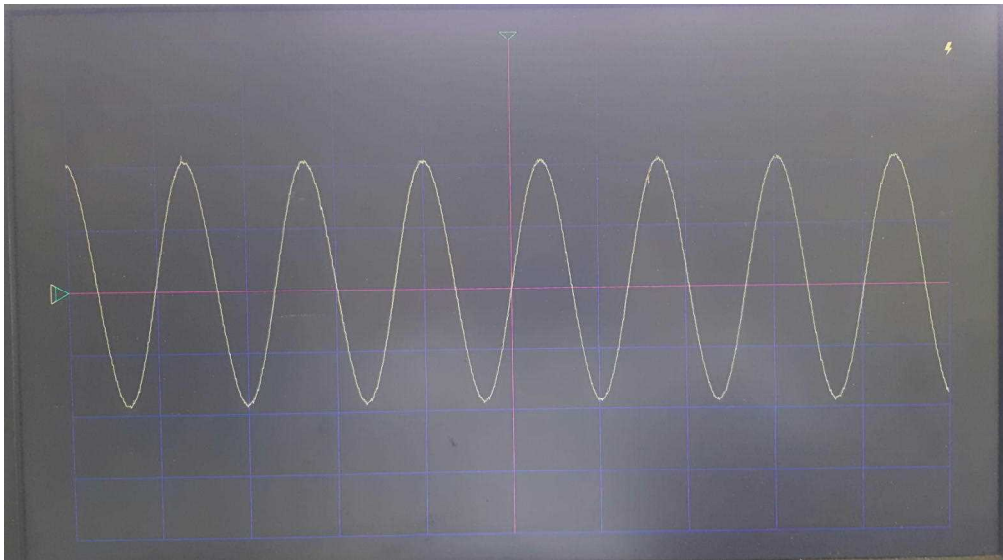


図 1 2 サイン波がグラフ画面に表示され続けられる。こうなれば、今までの処理手順などが正常であったことを確認できる。今後、通常の USB オシロスコープとして使用できよう。動作の正常終了はコントロールキーと C 文字キーを同時に押すことでできる。そして、グラフ画面は消え、次の図 1 3 の画面となる。測定パラメータなどが表示される。

オシロ動作の終了

「ctrl」+「c」で終了できる。

(18) コンソール画面の終了

\$ exit ↵

で、コンソール画面を終了する。

(19) タスクバー左上の「Menu」をクリックし、順次終了手順を行う。

(20) 各部分の電源を切る。

```

pi@raspberrypi: ~/TR1407W/ソース式/src
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi: ~ $ dir
2020-04-22-005924_1824x984_screenshot.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 ソース式
pi@raspberrypi: ~/TR1407W/ソース式 $ dir
include pld pld_src src 配布ソフト
pi@raspberrypi: ~/TR1407W/ソース式 $ cd src
pi@raspberrypi: ~/TR1407W/ソース式/src $ dir
Makefile bench_in.c bench_in.ihx build osc_gl osc_gl.c
pi@raspberrypi: ~/TR1407W/ソース式/src $ sudo cycfx2prog prg: bench_in.ihx
Using ID 04b4:8613 on 001.009.
Putting 8051 into reset.
Programming 8051 using "bench_in.ihx".
pi@raspberrypi: ~/TR1407W/ソース式/src $ sudo ./osc_gl
RaspberryPi_Oscilloscope_Start
SCLK = 2MHz
S/s = 1MHz
Time/div = 100us
V/div = 2v
offset = 0.00v
trgedge = 1
trglevel = 0.00v
trgpos = 0.0
pix_x = 1000
^Cpi@raspberrypi: ~/TR1407W/ソース式/src $

```

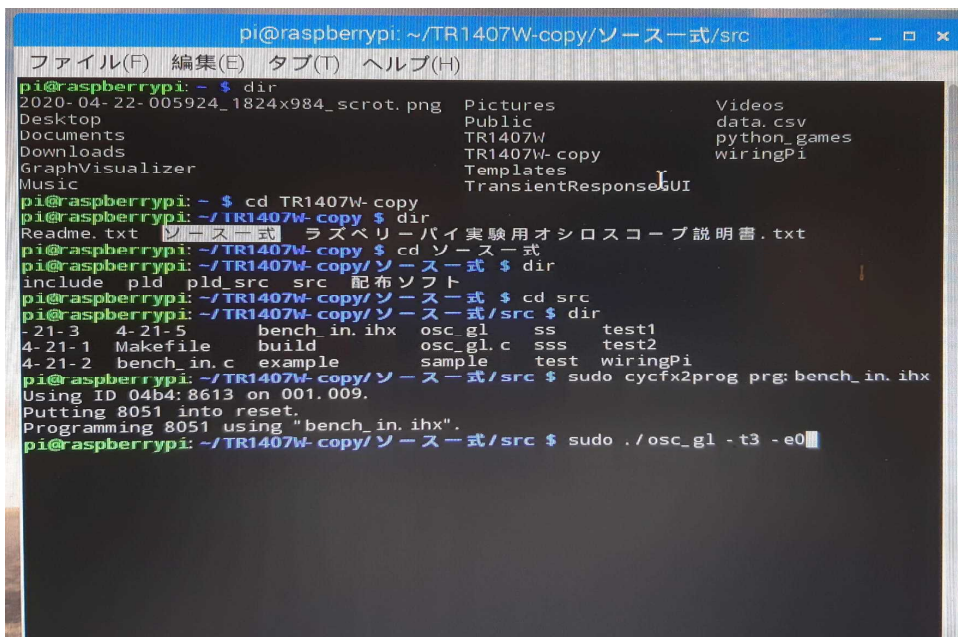
図 1 3 各種の設定パラメータが表示され、ターミナルモードでの入力可能モードとなる。

再動作の手順

- (21) 各部分の電源を入れる。
- (22) コンソール画面を開く。
- (23) ディレクトリ「src」へ移動する。
- (24) `$ sudo chmod +x build` (必要ないかも?)
- (25) `$ sudo cycfx2prog prg:bench_in.ihx`
- (26) `$ sudo ./osc_gl -**` (起動オプション 表参照) ↵

4.2 微弱な過渡現象波形などの測定可能なオシロスコープとしての使い方

参考文献(3)で紹介している使用方法である。図9の配置で、信号入力端子に被測定信号を入力。トリガ入力端子には現象を生起しているトリガ信号を入力する。トリガ信号はLレベルは0V、Hレベルは+5Vが理想値。その後、4.1で記述しているターミナルモードでの処理を行う。ただし「TR1407w」は「TR1407W-copy」に変更する。図15にそれらの手順を示している。なを、ここではモデル信号として、参考文献(3)で使用していたLC共振回路における過渡減衰波形を同じく用いている。



```
pi@raspberrypi: ~ $ dir
2020-04-22-005924_1824x984_screenshot.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 ソース式
pi@raspberrypi: ~/TR1407W-copy/ソース式 $ dir
include  pld  pld_src  src  配布ソフト

pi@raspberrypi: ~/TR1407W-copy/ソース式 $ cd src
pi@raspberrypi: ~/TR1407W-copy/ソース式/src $ dir
-21-3  4-21-5  bench_in.ihx  osc_gl  ss  test1
4-21-1  Makefile  build  osc_gl.c  sss  test2
4-21-2  bench_in.c  example  sample  test  wiringPi

pi@raspberrypi: ~/TR1407W-copy/ソース式/src $ sudo cycfx2prog prg:bench_in.ihx
Using ID 04b4:8613 on 001.009.
Putting 8051 into reset.
Programming 8051 using "bench_in.ihx".
pi@raspberrypi: ~/TR1407W-copy/ソース式/src $ sudo ./osc_gl -t3 -e0
```

図14 ターミナルモードで、ディレクトリ「TR1407W-copy」へ移る。注意:「TR1407W」ではない。後は、手順は図13と同じ。最後に命令

`$ sudo ./osc_gl -t3 -e0` エンターキー

で、実行される。が、`-t3 -e0` はオプションであり、なくても良い。詳細は参考文献(1)を参照すること。エンターキーを押すと、積算回数を聞いてくる(Please Input Sum_counter=)。図17を参照。希望する整数値を入力する。続いて、保存ファイル名を聞いてくる(Please Input data_file_name=)。希望するファイル名を入力する。画面は図15の如くのグラフ画面となり、入力信号の描写がトリガ信号が入力する度に行われる。

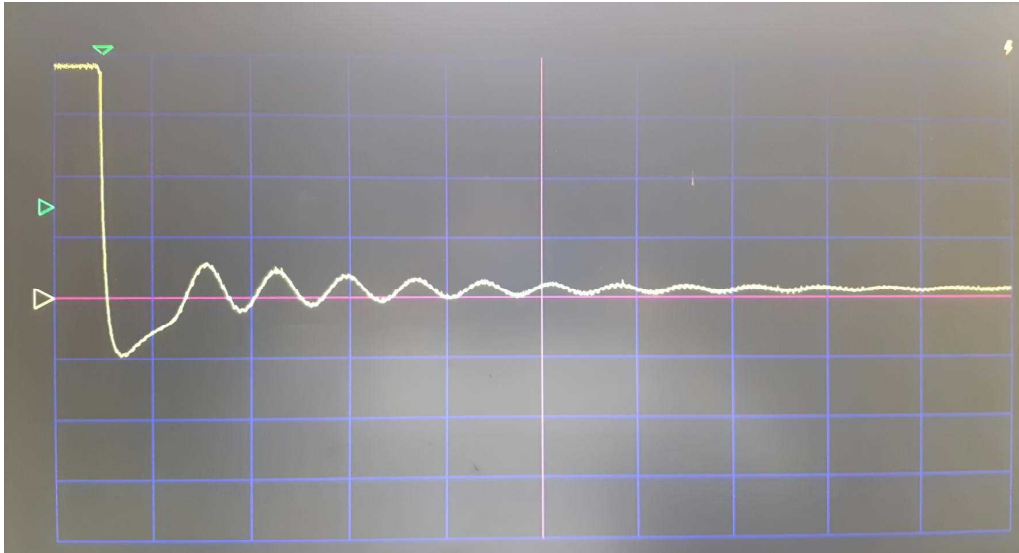


図15 トリガ信号+過渡減衰波形の描写画面。何故こうなのかは参考文献(3)を読むこと。このように安定して描写させるためには、オプションパラメータを適値に設定する必要がある。トライ&エラーである。入力した積算回数に達すると、積算は終了し、次の図16のように、積算平滑した波形が表示されて停止する。

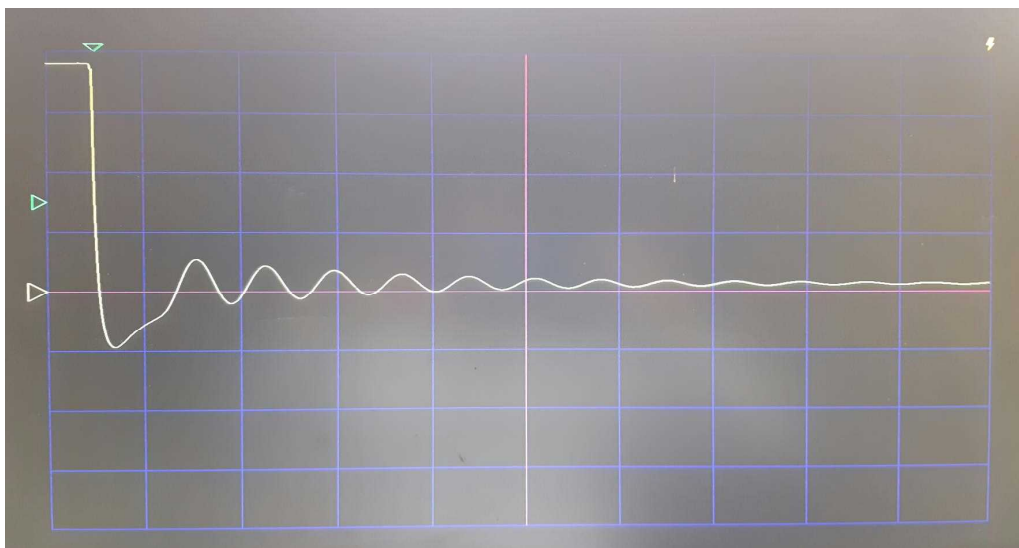


図16 図15のグラフ曲線状にあった、細かい雑音成分が消滅し、綺麗なサイン型の減衰波形が得られていることが分かって。効能の程は参考文献(3)を参照。


```

pi@raspberrypi: ~/TR1407W-copy/ソース式/src
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi: ~ $ dir
2020-04-22-005924_1824x984_screenshot.png Pictures Videos
Desktop Public data.csv
Documents TR1407W python_games
Downloads TR1407W-copy wiringPi
GraphVisualizer Templates
Music TransientResponseGUI
pi@raspberrypi: ~ $ cd TR1407W-copy
pi@raspberrypi: ~/TR1407W-copy $ dir
Readme.txt ソース式 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 ソース式
pi@raspberrypi: ~/TR1407W-copy/ソース式 $ dir
include pld pld_src src 配布ソフト
pi@raspberrypi: ~/TR1407W-copy/ソース式 $ cd src
pi@raspberrypi: ~/TR1407W-copy/ソース式/src $ dir
-21-3 4-21-5 bench_in.ihx osc_gl ss test1
4-21-1 Makefile build osc_gl.c sss test2
4-21-2 bench_in.c example sample test wiringPi
pi@raspberrypi: ~/TR1407W-copy/ソース式/src $ sudo cycfx2prog prg: bench_in.ihx
Using ID 04b4:8613 on 001.009.
Putting 8051 into reset.
Programming 8051 using "bench_in.ihx".
pi@raspberrypi: ~/TR1407W-copy/ソース式/src $ sudo ./osc_gl -t3 -e0
Please input sum_counter = 1000
Please Input data_file_name = 4-22-1
RaspberryPI_Oscilloscope_Start
SCLK = 2MHz
S/s = 1MHz
Time/div = 100us
V/div = 2v
offset = 0.00v
trgedge = 0
trglevel = 3.00v
trgpos = -4.5
pix_x = 1000
^Cpi@raspberrypi: ~/TR1407W-copy/ソース式/src $

```

図 1 7 測定終了後、コントロールキー+C文字キーの同時押しで、プログラムの実行は終了し、命令入力モード（プロンプトモード）に戻る。

参考文献

- (1) 「每秒 2 M サンプル取り込み！ 実験用 USB オシロスコープ」、渡辺 研、トランジスタ技術、CQ出版社、2014年7月号。
- (2) 「USB オシロスコープの複製」、金野茂男、2016年12月、著者のホームページで公開。
- (3) 「微弱な過渡現象信号も測定できる USB オシロスコープ」、金野茂男、2019年3月、追記同年5月。
- (4) 「名刺サイズの魔法のパソコン Raspberry Pi で遊ぼう！ 改訂第4番」、林和孝、株式会社ラトルズ、2015年8月。