RaspberryPi活用6⑤

RaspberryPiのカメラモジュールを以下のように接続する。



「設定」→「Raspberry Piの設定」→「カメラ」を有効する。

Raspberry Pi の設定					
システム	インターフェイス	パフォーマンス	ローカライゼーション		
カメラ:		④ 有効	○ 無効		
SSH:		④ 有効	○ 無効		
VNC:		④ 有効	○ 無効		
SPI:		● 有効	○ 無効		

映像は、Raspiのモニタ端子に接続されたモニタに映し出される。 (VNCのネット経由での映像は出ない)



動画と静止画を保存するフォルダを準備する。 フォルダ名をCameraで作成する。

pi@raspberrypi:~ \$ mkdir Camera

情報によれば、raspistillの動画はRaspiHDMIモニタに映し出され、プレビュー時間内(ディフォルト5s)の最終フレームを保存される。

pi@raspberrypi:~ \$ cd Camera/
pi@raspberrypi:~/Camera \$ raspistill -o image.jpg

pi@raspberrypi:~/Camera \$ ls
image.jpg
pi@raspberrypi:~/Camera \$

カメラを上下、左右反対に使用する場合は \$raspistill -vf -hf -o image.jpg とする。

イメージビューワにて確認する。ファイルを開くアイコンから保存したimage.jpgを 開く。



動画記録はディフォルトでH264フォーマットの5秒間の映像が記録される。-tオ プションで10000とすれば10秒間の撮影となる。 記録中の動画は先ほど同様HDMIコネクタ出力モニターに映る。

\$ raspivid -t 10000 -o video.h264

Raspiに直接接続されたモニタに10秒間の映像が映るとともに、video.h264の ファイル名で記録される。

pi@raspberrypi:~/Camera \$ raspivid -t 10000 -o video.h264
pi@raspberrypi:~/Camera \$ ls video*
video.h264
pi@raspberrypi:~/Camera \$

omxplayerを使ってvideo.h264を実行すれば、同様にコネクタ出力モニターに19 20x1080p(ディフォルト解像度)の録画された動画が表示される。

```
pi@raspberrypi:~/Camera $ omxplayer video.h264
Video codec omx-h264 width 1920 height 1080 profile 100 fps 25.000000
Subtitle count: 0, state: off, index: 1, delay: 0
V:PortSettingsChanged: 1920x1080@25.00 interlace:0 deinterlace:0 anaglyph:0 par:
1.00 display:0 layer:0 alpha:255 aspectMode:0
have a nice day ;)
pi@raspberrypi:~/Camera $
```

VNCクライアント画面には映らないので、直接接続したモニター画面でチェックのこと。

VNCクライアント画面で再生させるにはVLCプレイヤーが使用できる。

ただし、VNCでの画像はリアルな再生は望めない。

バージョンを重ねるうち、VLCメディアプレイヤーは標準でインストールされるようになった。



カメラモジュールの接続

「再生」→「再生」

VLCメディアプレ	イヤー	_ 🗆 🗙		
メディア (<u>M</u>) 再生 (<u>L</u>) オーディオ (<u>A</u>) ビ	デオ (V) 字幕 (T)	ツール (<u>0</u>) »		
			VLCメディア	プレイヤー
	メディ	ア (M) 再生 (L) オーディオ (<u>A</u>)	ビデオ (V) 字
		タイ	´ トノレ (!)	>
		チャ	· プター (<u>C</u>)	>
		プロ	Ιグラム (<u>P</u>)	>
		カス	タムブックマーク	(<u>B</u>) >
		速度	E (<u>E</u>)	>
		@少し	,先に進む (<u>J</u>)	
		🖻 少し	/前に戻る (<u>K</u>)	
		指定	医時間に移動 (工)	Ctrl+T
		▶ 再生	E (P)	
		■ 停止	= (<u>S</u>)	

カメラモジュールの接続

「ファイル」→「追加」 選択

		メディアを開く	_ = ×	
🖻 ファイル (E) 💊 ディスク (D)	🚏 ネットワーク (<u>N</u>)	キャプチャーデバイ、<>	
ファイル選択 以下のリン	R ストとボタンでローカ	ルファイルを選択することだ	が可能です。	
		ひとつ、または複数の	ファイルを選択します。	_ 0
	Look in:	nome/pi/Camera	• \$ \$ \$	🔳
字幕7	Computer	Name	✓ Size Ty	pe Date
	🍙 pi	video h264	18.7 MB h2	64 File 201
□詳細設♬				
		c		•
Sec.	File <u>n</u> ame: video	p.h264		<u>O</u> pen
	Files of type: メデ	ィアファイル (*.3g2 *.3gp *	.3gp2 *.3gpp *.amv *.asf *.a [,] •	Cancel

カメラモジュールの接続

再生をクリック

		メディアを開く		_ = ×
ファイル (E)	S ディスク (D)	早ネットワーク(N)	ロ キャプチャ	ーデバイ、〈〉
ファイル選択 以下のリストの	とボタンでローカ	ルファイルを選択するこ	ことが可能です。	
/home/pi/Ca	amera/video.h26	64		▶ 追加
			-	一削除
字幕ファイル	レを使用 (<u>T</u>)			
				参照
〕詳細設定オプ	ションの表示 (M)	γ		
		Ŧ	ī生 (P) ~ [キ	ヤンセル (<u>C</u>)

raspividコマンドで解像度640×480で10秒程度、ビットレート10Mbit/s(ネットの10Base程度)の動画ファイルを作成するには.

raspivid に以下のオプションが用意されている。

\$ raspivid -o video.h264 -t 10000 -w 640 -h 480 -b 10000000 とタイプインする。H264はmp4形式だが、拡張子をh264にしないと記録されない。

pi@raspberrypi:~/Camera \$ raspivid -o video.h264 -t 10000 -w 640 -h 480
 -b 10000000

VLCにて同様に動画を確認する。



VLCはデータを変換する機能をもつ。 「メディア」→「変換/保存」を選択

	VLCメディアプレイヤ
メディア (<u>M</u>) 再生 (<u>L</u>) オーディオ (<u>A</u>)	ビデオ (V) 字幕 (T) ツ
ファイルを開く… (<u>F</u>)	Ctrl+O
▶ 複数のファイルを開く (<u>0</u>)	Ctrl+Shift+O
🗎 ディレクトリを開く (<u>l</u>)	Ctrl+F
ディスクを開く (D)…	Ctrl+D
譬 ネットワークストリームを開く (ℕ)	Ctrl+N
■ キャプチャーデバイスを開く (C)…	Ctrl+C
クリップボードからURLを開く (<u>L</u>)	Ctrl+V
最近アクセスしたメディア (<u>R</u>)	>
プレイリストファイルの保存 (<u>F</u>)	Ctrl+Y
変換 / 保存 (<u>R</u>)	Ctrl+R
((*)) ストリーム (<u>S</u>)	Ctrl+S
プレイリストの最後で終了	
▶終了 (Q)	Ctrl+Q

「追加」ボタンで対象のファイルを選択する。

メディア (M) 再生 (L) オーディオ (A) ビデオ	(V) 字幕 (T) ツール (<u>0</u>) 表示 (<u>1</u>) ヘルプ (<u>H</u>)
	メディアを開く – ロ ×
🖹 ファイル (E) 🥥 ディスク	(<u>D</u>) 🚏 ネットワーク (<u>N</u>) 📑 キャプチャーテ< >
ファイル選択 以下のリストとボタンでロ	ーカルファイルを選択することが可能です。
/home/pi/video.h264	追加
	削除
□ 空草ファイ ルを使田 (T)	
□ 子幕ノア1ルを使用 (<u>1</u>)	

「変換/保存」ボタンをクリック

変換 / 保存 (<u>o</u>)	~ キャンセル (<u>C</u>)

「プロファイル」から変換形式を選んで適当なファイル名にして実行する。



VLCデータ変換はVLCプレイヤーがサポートするものしかないので、Windowsマシンの助けを借りてWindowsソフトのXmedia Recodeで変換するのがお勧め。そのためにはWinSCPを使用してRaspiからデータをwindowsマシンの適当なフォルダにコピーする。下記はWinSCPを起動画面。

WinSCP	► ログイン - WinSCP ■ 新しいサイト	セッション	
		転送プロトコル(F) SFTP ▼ ホスト名(H) 192.168.137.55 ユーザ名(U)	ボート番号(R) 22 🚔 パスワード(P)
		pi (保存(S) ▼	••••••••••••••••••••••••••••••••••••
		E E D Ø A	開じる ヘルプ(H)

ここではWindowsマシンのディスクトップにコピーする。

💑 pi - pi@192.168.137.5	55 - WinSCF								- • ×
ローカル(L) マーク(M)	ファイル(F)	コマンド(C) セッシ	≠ョン(S) オプション(O) リ	モート(R) ヘルプ(H)				
🕀 💦 📚 同期 🧊 🦑		🖗 🍙 ≠	- • 転送設定 デフォルト		• 🧬 •				
📮 pi@192.168.137.55 🚅 新しいセッション									
ニー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・								余。	
山 アップロード 山 7 編集 🗙 🚮 🕞 プロパティ 🗳 🕞 匣 🖃 🔽 🛛 😭 🖗 🗩 🕞 🖗 🖓 編集 🗙 🚮 🕞 プロパティ 🗳 🕞 匣 💌									
C:¥Users¥yamaoka¥Desk	ctop				/home/pi				
名前	サイズ	種類	更新日時		名前	サイズ	更新日時	パーミッ	所有者
👪		ひとつ上のディ	2016/10/28 15:18:42		월		1970/01/01 9:39:04	rwxr-xr-x	root
) Microsoft Office 2013		ファイル フォル	2016/05/25 13:54:33		퉬 Desktop		2016/09/23 12:52:39	rwxr-xr-x	рі
🚼 Tera Term.lnk	2 KB	ショートカット	2016/05/25 16:11:59) Documents		2016/10/28 14:59:44	rwxr-xr-x	pi
🛃 Untitled.BMP	4,771 KB	ビットマップ イ	2016/10/24 9:12:06)) Downloads		2016/09/23 13:03:29	rwxr-xr-x	pi
ਙ਼ੀ xfinder.exe - ≶∃−	1 KB	ショートカット	2016/10/19 11:48:12		퉬 mjpg-streamer		2016/10/27 14:40:17	rwxr-xr-x	pi
記ウイルスバスター ク	2 KB	ショートカット	2016/07/28 9:25:06		퉬 Music		2016/09/23 13:03:29	rwxr-xr-x	рі
🛃 無題.bmp	8,641 KB	ビットマップ イ	2016/10/26 12:25:20		Pictures		2016/09/23 13:03:29	rwxr-xr-x	pi
					퉬 pressure		2016/10/26 10:18:02	rwxr-xr-x	pi
					퉬 Public		2016/09/23 13:03:29	rwxr-xr-x	pi
					python_games		1970/01/01 9:39:05	rwxr-xr-x	pi
) Templates		2016/09/23 13:03:29	rwxr-xr-x	pi
					퉬 Videos		2016/09/23 13:03:29	rwxr-xr-x	рі
					퉬 wiringPi		2016/10/26 17:02:36	rwxr-xr-x	pi
					퉬 Work_py		2016/10/28 11:02:10	rwxr-xr-x	pi
) WorkC		2016/10/28 11:23:10	rwxr-xr-x	рі
					git-proxy.sh	1 KB	2016/10/26 12:05:41	rwxr-xr-x	рі
					🔛 image.jpg	4,114 KB	2016/10/28 15:54:06	rw-rr	рі
					video.h264	19,986	2016/10/28 12:30:02	rw-rr	рі
0 B (全 13,415 KB 中) /	0個目(全)	6 ファイル中)	1 非表示	Ę	19,986 KB (全 24,099 KB	中)/1個	目 (全 17 ファイル中)		20 非表示
								SFTP-3	0:01:06
								H H	

Xmedia Recodeを起動してファイルを開くでvideo.h264を選択する。

Vite de Dese de	ファイル(F) 編集(E) オプション(O) ヘルプ(H)
XMedia Recode	
	ディスク ファイル リストに リストか エンコー
	クロシアイルを開く
	形式 リスト

🗄 XMedia Rec	ode 3.3.5.4						
ファイル(F) 🕯	編集(E) オプS	ション(0) へ	レプ(H)				
	1	-	- 1	0	2	?	0
ティスク フ:	לע ארדיק	KNE UX	トか エンコー		ム 更新の確		7 終了
	名前	チャプター	再生時間		映像		音声
	video.h264	0	00:00:00		MPEG-4 AVC / H.264	25.0	
形式映像	音声トラック1	字幕 クロップ	⁷ /プレビュー リスト				
一般			_	lux			

カメラモジュールの接続

	名前 video.h264	チャプター 0	再生時間 00:00:00	映像 MPEG-4 AVC / H.264 2	音声 5.0
 形式 映像 一般 プロファイル: 形式: ファイル拡張・ 	 音声トラック 1 カスタム MPEG 1 子: mpeg 	字幕 クロップ	*/ブレビュー リスト ・ ・ ・ ・ ・	Mux Mux レート (Byte/s): 0 パケットサイズ (Byte): 2048	V V
- 出力ストリーム ◎ 映像のみ ◎ 映像 + 看	」形式 ◎ 1	音声のみ			

適当なフォルダを指定して「開く」



形式には多種のデータ形式が用意されている。

XMedia Recode 3.3.5.4				
ファイル(F) 編集(E) オプション(O) ヘルプ(H)				
ディスク ファイル リストに リストか エンコー ホーム				
:	名前	チャプター	再生時間	映像
	video.h264	0	00:00:00	MPEC
形式映像	音声トラック1	字幕 クロッ:	ブ/ブレビュー リスト	
──般				Mux
プロファイル:	<u> </u>		•	Mux レート (By
形式:	MPEG 1		- 🖈	パケットサイズ
ファイル拡張子	: MP3 MP4		^	
一出カストリームチ	MPEG 1 MPEG 2 TET MPEG TS (D MPI	VB TS)		
◎ 映像のみ	MXF (Mater	ial eXchange Fo	rmat)	
◎ 映像 + 音詞				
☑ 映像と音声	OGM OGV Opus RealMedia iをli Speex Audi SubRip sub SVCD	o (SPX) title		
出力方法: [SWF			
保存先:	C:¥VCD	Audio)		

WinSCPでRaspbrryに書き戻す前にWindowsアプリ側のVLC media playerで確認する。



カメラモジュールを使ってmjpeg-streamerに使用する。 バージョンを重ねるうちカメラモジュールは接続されると ディフォルトで /dev/video0 に登録されるようになった。 したがって先の作成した stream.sh がそのまま使える。



pi@raspberrypi:~ \$ cd /dev
pi@raspberrypi:/dev \$ ls video*
video0 video10 video11 video12
pi@raspberrypi:/dev \$

ここでは複数のカメラを配置した場合を紹介する。

ちなみにカメラモジュールデバイス情報は

\$ v4l2-ctl -list-device で確認できる。

また、カメラモジュールの動画フォーマットは \$ v4l2-ctl -list-formats で確認できる。

pi@raspberrypi:~ \$	v4l2-ctllist-formats
ioctl: VIDIOC_ENUM	_FMT
Index	: 0
Type	: Video Canture

Index 1 および 5 にそれぞれ YUYV, MJPGフォーマットが確認できる。

Index Type Pixel Name	: Format: ;	1 Video Capture 'YUYV' YUYV 4:2:2
Index	:	5
Type	:	Video Capture
Pixel	Format:	'MJPG' (compressed)
Name	:	Motion-JPEG

Webカメラを接続追加する。



pi@raspberrypi:/dev \$ ls video*
video0 video1 video10 video11 video12 video2
pi@raspberrypi:/dev \$

新たに video1 と video2が現れるが、video1が追加されたカメラになる。

さらにWebカメラを接続追加する。

このとき、カメラも電力を使用するので電源容量には余裕のあるものを使用すること。 省電力を考慮してギリギリの電源供給の場合、カメラを認識しない場合がある。

pi@raspberrypi:/dev \$ ls video*
video0 video1 video10 video11 video12 video2 video3 video4
pi@raspberrypi:/dev \$

ドライバは video3 と video4が追加される。 ペアで追加される理由は不明。

注意:

どのカメラがどのドライバに割り当てられるか不明である。

今回のように順番に接続した場合は識別できるが、カメラ接続状態から立ち上げた場合の割振りは不明である。

複数カメラを使用する場合、別途特定ルーチン、あるいは設定が必要となるだろう。

./stream.sh は 3台のカメラのための./stream.shを用意する。

pi@raspberrypi:~ \$ cd mjpg-streamer/				
<pre>pi@raspberrypi:~/mjpg-streamer</pre>	\$	ср	<pre>stream.sh</pre>	stream0.sh
<pre>pi@raspberrypi:~/mjpg-streamer</pre>	\$	ср	<pre>stream.sh</pre>	stream1.sh
<pre>pi@raspberrypi:~/mjpg-streamer</pre>	\$	ср	<pre>stream.sh</pre>	stream3.sh
pi@raspberrypi:~/mjpg-streamer	\$			

./stream0.shの編集。

pi@raspberrypi:~/mjpg-streamer \$ nano stream0.sh

#! /bin/sh

PORT="8080" SIZE="640x480" F_RATE="60" MJPG_STREAMER="/home/pi/mjpg-streamer" sudo "\$MJPG_STREAMER/mjpg_streamer" \ -i "\$MJPG_STREAMER/input_uvc.so -f \$F_RATE -r \$SIZE -d /dev/video0 -y -n" \ -o "\$MJPG_STREAMER/input_http.so -w \$MJPG_STREAMER/www -p \$PORT"

動画サイズを640x480。 デバイスは /dev/video0。動画フォーマットは YUYV として "-y"。

./stream1.shの編集。

pi@raspberrypi:~/mjpg-streamer \$ nano stream1.sh

#! /bin/sh
PORT="8082"
SIZE="640x480"
F_RATE="60"
MJPG_STREAMER="/home/pi/mjpg-streamer"
sudo "\$MJPG_STREAMER/mjpg_streamer" \
-i "\$MJPG_STREAMER/input_uvc.so -f \$F_RATE -r \$SIZE -d /dev/video1 -n" \
-o "\$MJPG_STREAMER/output_http.so -w \$MJPG_STREAMER/www -p \$PORT"

動画サイズを640x480。デバイスは /dev/video1。使用するBuffaroカメラは MJPEGフォームしかないので"-y"をはずす。 ポートNoは、競合を防ぐため "8082"

./stream3.shの編集。

pi@raspberrypi:~/mjpg-streamer \$ nano stream3.sh

```
#! /bin/sh
PORT="8084"
SIZE="640x480"
F_RATE="60"
MJPG_STREAMER="/home/pi/mjpg-streamer"
sudo "$MJPG_STREAMER/mjpg_streamer" \
-i "$MJPG_STREAMER/input_uvc.so -f $F_RATE -r $SIZE -d /dev/video3 -n" \
-o "$MJPG_STREAMER/output_http.so -w $MJPG_STREAMER/www -p $PORT"
```

動画サイズを640x480。デバイスは /dev/video3。使用するカメラはMJPEG、 YUYVフォームともにあるが "-y"を入れ YUYV とする。 ポートNoは、競合を防ぐため "8084"

./stream0.sh を起動

<pre>pi@raspberrypi:~/mjpg-streamer \$./stream0.sh</pre>				
MJPO	G Streamer Version: s	svn rev: 3:172M		
i:	Using V4L2 device.:	/dev/video0		
i:	Desired Resolution:	640 x 480		
i:	Frames Per Second.:	60		
i:	Format:	YUV		
i:	JPEG Quality:	80		
0:	www-folder-path:	<pre>/home/pi/mjpg-streamer/www/</pre>		
0:	HTTP TCP port:	8080		
o:	username:password.:	disabled		
0:	commands:	enabled		

./stream1.sh を起動

pi@raspberrypi:~/mjpg-st	treamer \$./stream1.sh
MJPG Streamer Version: s	svn rev: 3:172M
i: Using V4L2 device.:	/dev/video1
i: Desired Resolution:	640 x 480
i: Frames Per Second.:	60
i: Format	MJPEG
o: www-folder-path:	<pre>/home/pi/mjpg-streamer/www/</pre>
o: HTTP TCP port:	8082
<pre>o: username:password.:</pre>	disabled
o: commands:	enabled

./stream3.sh を起動

<pre>pi@raspberrypi:~/mjpg-streamer \$./stream3.sh</pre>				
MJPG Streamer Version: svn rev: 3:172M				
Using V4L2 device.:	/dev/video3			
Desired Resolution:	640 x 480			
Frames Per Second.:	60			
Format:	YUV			
JPEG Quality:	80			
www-folder-path:	/home/pi/mjpg-streamer/www/			
HTTP TCP port:	8084			
username:password.:	disabled			
commands:	enabled			
	Aspberrypi:~/mjpg-st Streamer Version: S Using V4L2 device.: Desired Resolution: Frames Per Second.: Format JPEG Quality WWW-folder-path: HTTP TCP port username:password.: commands			

3箇所のmjpeg-streamによる stream動画が確認できる。



課題:連続した5秒動画を個別の10ファイルで保存する。

シェルスクリプトを使い、以下のサンプルファイルを作成せよ (条件)

- ・640x480解像度で5秒動画。
- ・動画のファイル名は識別できるファイル名とすること。
- ・Recordフォルダ内に当日のフォルダ名がわかるように保存すること。
- ・systemd.serviceを使い電源投入時に自動起動させる。

課題:連続した5秒動画を個別に10個保存する。

Recordフォルダを作成後、record.shスクリプトファイルを作成する。

pi@raspberrypi:~/Camera \$ mkdir Record pi@raspberrypi:~/Camera \$ nano record.sh

```
#! /bin/sh
today=$(date ~+%y%m%d~)
dirname=/home/pi/Camera/Record/record$today
 f [ ! -d "$dirname" ]; then
         mkdir $dirname
for i in `seq 0_9`
do
         file=file<mark>$today$i</mark>
         raspivid -o $dirname/$file.h264 -t 5000 -w 640 -h 480
         echo i=<mark>$</mark>i
done
```

課題:連続した5秒動画を個別の10ファイルで保存する。

パーミションを変更する。

pi@raspberrypi:~/Camera \$ chmod 755 record.sh

実行すると

./Recordフォルダにrecord181101フォルダが作成されてフォルダ内に10個の ファイルが保存される。

再度実行時には上書きされる。

pi@raspberrypi:~/Camera \$./record.sh

pi@raspberrypi: //Camera \$ ls ./Record/ record181101 pi@raspberrypi: //Camera \$ ls ./Record/record181101/ file1811010.h264 file1811013.h264 file1811016.h264 file1811019.h264 file1811011.h264 file1811014.h264 file1811017.h264 file1811012.h264 file1811015.h264 file1811018.h264 pi@raspberrypi: //Camera \$ 課題:連続した5秒動画を個別の10ファイルで保存する。

記録時刻は

\$ ls - l にて確認可能

pi@raspberrypi:~/Camera \$ ls -l	./Record/record181101/			
合計 3104				
-rw-rr 1 pi pi 348441 11月	1 16:14 file1811010.h264			
-rw-rr 1 pi pi 322404 11月	1 16:15 file1811011.h264			
-rw-rr 1 pi pi 312015 11月	1 16:15 file1811012.h264			
-rw-rr 1 pi pi 321380 11月	1 16:15 file1811013.h264			
-rw-rr 1 pi pi 337066 11月	1 16:15 file1811014.h264			
-rw-rr 1 pi pi 322362 11頁	1 16:15 file1811015.h264			
-rw-rr 1 pi pi 307182 11頁	1 16:15 file1811016.h264			
-rw-rr 1 pi pi 309491 11頁	1 16:15 file1811017_h264			
-rw-rr 1 pi pi 297509 11月	1 16:15 file1811018.h264			
-rw-rr 1 pi pi 279788 11月	1 16:15 file1811019.h264			
pi@raspberrypi:~/Camera \$				