

# カメラ付き小型サブディスプレイ開発についての報告書

電子情報工学専攻 小池 亮太郎

## 1. はじめに

オンライン授業において、先生と生徒との間における目線の関係は重要である。通常の PC に装着あるいは埋め込んであるカメラの位置はディスプレイに対して上か下が標準である。この状態で、ビデオ通話を介した対話において目線の位置をカメラのある上、又は下に向けてしまうと、一方の人は目線が自分の見ている画面に対して正面を向いてない状態に見えてしまう。

このような背景の元、今回の演習目標として以上の問題を解決する装置の制作を設定した。この装置は、PC ディスプレイの中央にカメラがあったとしても邪魔にならず、目線を合わせることができるようになることを目的としている。この装置が必要となる環境は、PC を用いたビデオ通話/会議を想定している。装置はディスプレイとカメラとしての役割である装置 B と、もう一方はドングルおよび通信ホストとし、ユーザ PC との接続を担う装置 A の2つを想定している。

それぞれの装置は無線で相互通信し、PC は装置 B をカメラと外部接続ディスプレイとして認識できる。さらに装置 B のカメラを隠すように小型でベゼル部を抑えたディスプレイを配置する。PC のディスプレイ中央に装置 B のカメラが来るように設置し、装置 B のディスプレイには PC に映っているはずの裏側の画面を出力すれば違和感なくカメラを中央に設置することができるはずである。

## 2. 概要

### 2.1 装置 A について

装置 A は入力に、PC からの映像信号(HDMI)と装置 B からのカメラ映像を受け付ける。出力には装置 B へトリミングされた映像信号と PC へカメラ映像を渡す。PC が常に外部ディスプレイにつながっている場合には更に、入力された映像信号をそのまま出力するパススルーとしての機能を有する必要があるため、出力に映像信号(HDMI)が追加されることになる。以上の事から、これらの機能実装において FPGA 等を用いた回路設計による装置作成は上記の要件達成を満たすのに現実的ではない。そのため、小型でリアルタイム OS を動作させることができ、ソフトウェアからのアプローチで機能実装を実現させる作業領域を持ったデバイスを選定すると、RaspberryPi4model B(8GB-RAM)が適任である。

### 2.2 装置 B について

装置 B は入力に、適切にトリミングされた映像信号を受け付ける。出力には、装置 A へ送るカメラ映

像と自信に付いたディスプレイへの映像出力である。以上の事から、これらの実装をスムーズに実現させるのに RaspberryPiZeroW が適任であると判断する。

### 2.3 装置の全体像

装置 A、装置 B とそれを機能させる環境の全体を図 1 に示す。

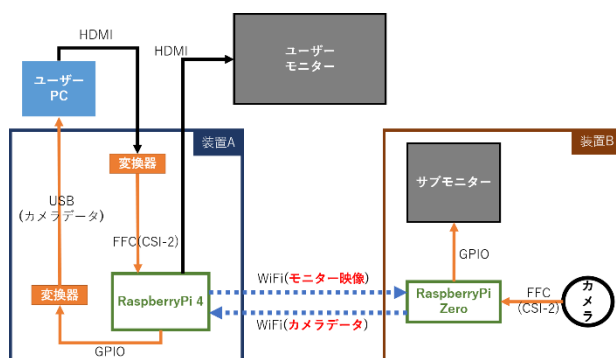


図 1 ユーザ PC と装置 A、装置 B との接続関係と接続形式

ユーザ PC からの HDMI 入力には変換器を用いて RaspberryPi4 にある CSI-2 端子へつながる。また、ユーザ PC へは USB 接続がされるが、その経路は GPIO からシリアル変換器を通して通信される。装置 A-B 間の通信は双方向性が必要なので TCP 通信を用いる。装置 B の OS は軽量化のため X システムを起動しない状態であるため、サブディスプレイへの出力はフレームバッファに割り込んで直接描画させるような実装になる。

## 3. 所要技術について

### 3.1 フレームバッファへの割り込みについて

フレームバッファへの割り込みは装置 B において使用した手法である。装置 B はサブディスプレイとして市販のものを採用した。性能表は付録 1 を参照。

フレームバッファとは、Linux Kernel 2.2 から追加されたバッファ領域で、純粋に画面表示領域として機能している。同時に VRAM(Video RAM)と混同されることがあるが、厳密には VRAM はビデオ回路用のメモリ全体を指す。

フレームバッファは、通常はメインメモリの一部として割り付けられており、直接、あるいはグラフィックス回路の描画機能等を使って、ここに画面イメージが書き込まれる。書き込まれたイメージは、DAC (Digital to Analog Converter) によって、ディスプレイのタイミングに合せたビデオ信号に変換され画面に表示される。 [1]

実装目線では、このフレームバッファは/dev/fb0 のメモリデバイスとして使用できるため、OS がディスプレイを認識さえしていれば自由に読み込み・書き込みが行える。そのため、OpenCV や OpenGL から新しいウィンドウを立ち上げて描画するといった回りくどい設計をせずに実装ができる。また、今回の様に動作デバイスのスペックによる制約により、X システムが起動できない等の状況では非常に役立つ機能である。

### 3.2 TCP/IP ソケット通信について

装置 A と装置 B は双方向通信を行っており、これは TCP/IP プロトコルによって実装されている。Wlan の規格は RaspberryPiZero の規格上、IEEE 802.11n の 2.4GHz 帯域を使用している。装置 A が AP 化されており、装置 B は起動時に自動的に装置 A の IP を探して接続する流れとなっている。

装置 A、装置 B における画像送受信のプロセスはほとんど同じで、図 2 に従って送信・受信を行っている。ただし、画像データについてはリサイズと圧縮を行ってからクライアントに送信している。圧縮は jpeg 圧縮を用い、メモリ上でファイル形式を変更している。圧縮後の画像のバイナリサイズは常に同じにならないため、送信するメッセージの先頭にバイナリサイズをヘッダ情報として付加している。

受信するときはヘッダにあるバイナリサイズを元にデコードすることによって画像を得ることができる。

TCP/IP			
	クライアント		サーバー
1	初期化	socket	初期化
2			待ち受けポートを指定
3			待ち受け開始
4			接続待ち
5	接続	connect	
6	送信→	send	
7			→受信
8			←送信
9	受信←	recv	
10	切断	shutdown	切断
11	終了	close	終了

図 2 TCP/IP プロトコルの送受信プロセス

## 4. 問題点

### 4.1 HDMI の CSI 入力における解像度の異常

HDMI からの入力を CSI を通して受信しているが、そのインターフェースブリッジとして東芝の TC358743XBG(付録 2)IC を用いている。これにより、UVC プロトコルで転送できる形に機械的に高速変換することができる。

通常、このチップセットは HDMI-CSI 変換を行う

ため、デバイスドライバは必要ない。しかし、実際に信号を受け取る際、解像度が強制的に固定化されてしまう。図 1 でのユーザー PC からはデュアルディスプレイのような見え方をしており、ユーザーモニターに関しては任意の解像度が設置されているが、装置 A への出力解像度は 1280x720(16:9)となってしまう。装置 A から見た入力解像度は 1920x1080(16:9)となっているため、要求する画像データは正常に取得できる。

この問題を解決するために、装置 A から・ユーザー PC からのアプローチを行ったが、いずれも失敗に終わった。アプローチ方法はいくつかあり、まず強制的に出力・入力解像度を変更する方法である。PC からは xrandr コマンドを、装置 A からは v4l2-ctl を用いていくつかの手順を経て設定を行ったが、映像信号が出力されなくなるなどの不具合に見舞われた。

現時点での解消方法は見つかっていない。

### 4.2 シリアル通信時でのデバイスモードの異常

装置 A とユーザー PC 間では USB 接続によるシリアル通信を予定していた。シリアル通信は GPIO を介して USB-RS232C 変換である FTDI 社の IC が搭載された変換器を用いている。

RaspberryPi は Peripheral mode(周辺機器モード)に対応しており、OTG 機能によって通信が可能になる。しかし、OTG 機能におけるデバイスツリー一覧のウェブカメラモードに対応したドライバについて不安定であることと、GPIO を使ったシリアル通信による OTG 機能はサポートしていないことから、現状での実装は厳しいものとなっている。

解決策として GPIO-OTG ではなく、USB-OTG を試すことにより問題を解消することができるかもしれない。

## 5. まとめ

リモート通話において画面中央にカメラが存在することによって目線問題を解決することができるというのが今回演習で作成したデバイスの目的であった。今回の期間で本デバイスでは以下の機能を実装した。

- (1)装置 A と装置 B 間の無線データ交信の実装
- (2)HDMI 入力からカメラ入力への変換機能の実装
- (3)フレームバッファへの介入により任意のディスプレイに描画する機能の実装

開発したデバイスは自律動作はまだできず、試作の試作といった段階である。また、完全な機能実装に至っておらず、不安定な部分も多いのが現状である。全機能の実装と自律化及び安定化が今後の課題である。

## 6. 付録

- (1)<https://www.switch-science.com/catalog/5901/>
- (2)<https://toshiba.semicon-storage.com/jp/semiconductor/product/interface-bridge-ics-for-mobile-peripheral-devices/hdmir-interface-bridge-ics/detail.TC358743XBG.html/>

## 参考文献

[1] 鈴木直美:フレームバッファ,<https://pc.watch.impress.co.jp/docs/article/980303/key20.htm>