

# MIPI CSI-2カメラと USB/GigEカメラの比較

西田祐矢

MIPI CSI-2 インタフェースはローコストなエンベデッドマシンビジョンシステムに最適なインタフェースである。USBやGigE、CoaXPressといったマシンビジョンに慣れ親しんだ人にとってはあまり馴染みのないインタフェースかもしれないので、どのように開発を進めるべきかという指針を解説する。

## MIPI CSI-2 インタフェースとは？

MIPIとはMobile Industry Processor Interfaceの略、CSI-2とはCamera Serial Interfaceの略で、MIPI CSI-2 インタフェースはMIPIアライアンスで規格化されている。カメラからのデータフローはLVDS(低電圧作動信号)で出力される。データ通信はCSIトランスミッターとCSIレシーバーで構成され一方通行でカメラ(CSIトランスミッター

側)からCSIレシーバー側に送られる。データレーンは最大4レーン、それにクロックレーンが足されている。カメラを制御するためのコントロールインタフェース(CCI / Camera Control Interface)は相互通信でI2Cのシリアル通信方式である。I2Cはマスターとスレーブ(カメラ側)をSCLとSDAという2本のラインで接続している。SCLとはシリアルクロックでマスターからスレーブ(カメラ)に一方通行

なのに対し、SDAとはデータクロックでマスターとスレーブ(カメラ)は双方向通信をしている。これらのCCIの通信は400kHzでやり取りされるのでレスポンスは非常に速い。以下に図としてまとめる。

## MIPI CSI-2 インタフェースの強みとは？

MIPI CSI-2インタフェースそのものの説明は以上で、そもそもMIPI CSI-2にどのような利点があるかを説明していく。

### 利点1：低遅延リアルタイム画像処理

1つ目の利点としては、ローコストにリアルタイム画像処理ができるという点があげられる。MIPI CSI-2カメラを用いれば、接続したPC(例：Jetsonやラズベリーパイ(通称：ラズパイ))が持つ各種画像処理アクセラレータ、ISP (Image Signal Processor)、DSP (Digital Signal Processor)へ専用MIPIデータラインを経由してダイレクトに画像データを取込む事ができる。これがUSBやGigE、CoaXPressのマシンビジョンインタフェースの場合は汎用的なバスを利用するため、PCIeバスを占有するフレームグラバボードが必要だったり、接続構成次第ではPC内

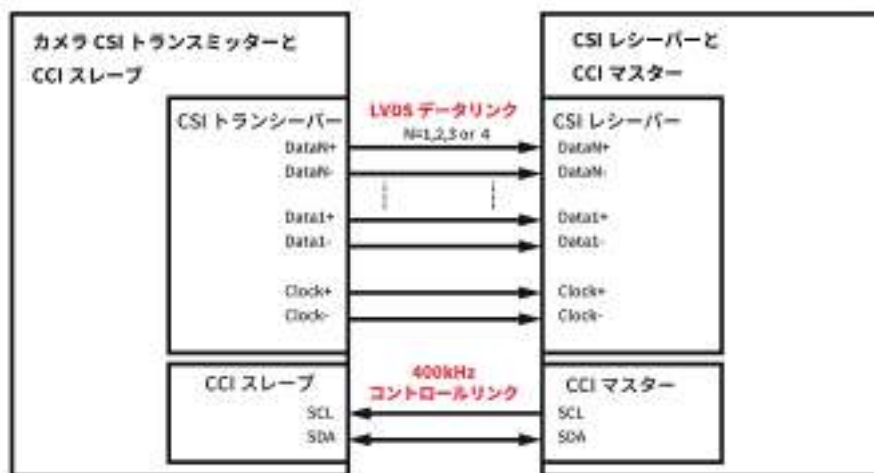


図1 MIPI CSI-2カメラの回路図を確認する場合は図1の通り、データレーン4つ+クロックレーン1つ、SCL & SDAレーン1つというのが基本で、それ以上にどのようなピンがデバイスやレシーバー側に実装されているかという視点で見ればわかりやすい。

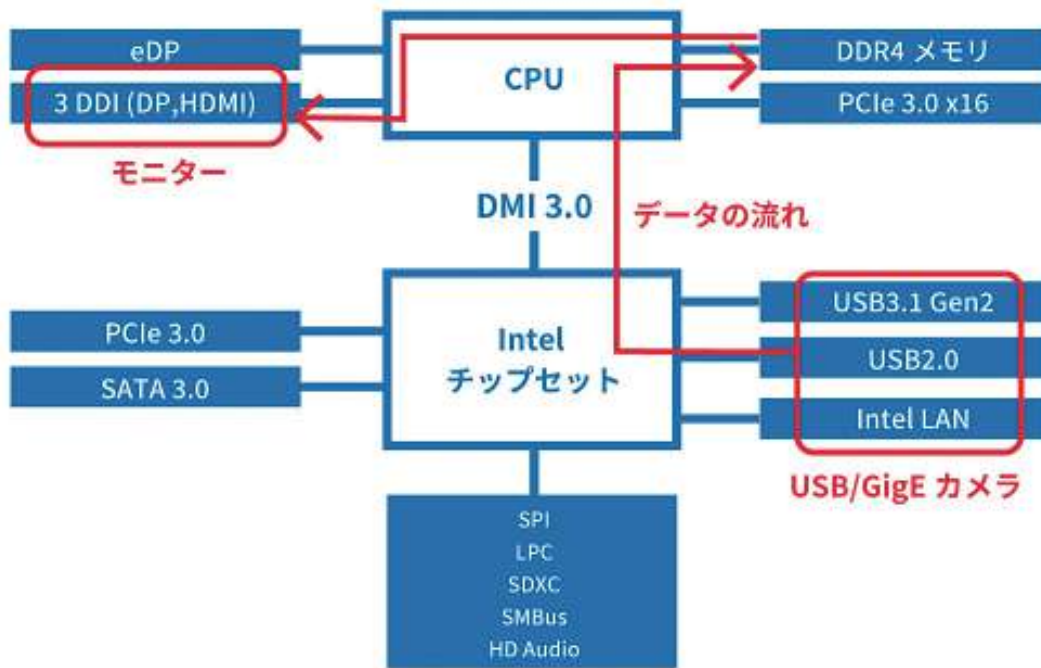


図2 Intelチップセットのブロック図。

でのデータフローや帯域を考慮する必要が発生する。一般的なIntelチップセットのブロック図で説明してみる。図2はUSBやGigEカメラの画像データを取込んでモニターに表示する、というだけのプロセスである。インタフェース→チップセット→DMI→CPU→メモリ→CPU→モニターという経路を辿っていることがわかる。しかし、メモリに入力されるまでにいくつかのボトルネックがあるのがわかる。例えばDMI転送回路でのデータ帯域、データ取込に関わるCPUのコア数やスレッド数などである。カメラに対してどのような挙動をさせたいか、PC側でどのような画像処理や機械学習や推論を走らせたいか、プログラムの組み方次第でPCハードウェアの必要スペックは大きく変わってくるのだ。話が少し逸れるが、高速マシンビジョンシステムかつリアルタイム画像処理を構築するアプリケーションでCPUでの処理が間に合わない場合は、PCIe インタフェース

のGPUやFPGAボードを増設して高速な機械学習や推論処理をしなければならない場合がある。その場合は帯域や処理負荷上それらがCPUに直結できるようなチップセット(例: Xeonなど)が望ましく、PCサーバ側のハードウェアコストや選定に検討すべき事項や把握しておくべき技術的要素が多くハードルは結構高い。MIPI CSI-2カメラをJetsonやラズパイに接続した場合は、ISPやDSPにダイレクトに画像データが転送されるので取込レイテンシーとデータ変換のCPU負荷を回避ができ、それらの要素を検討する必要がなくなる。このように、MIPIカメラを用いると検討事項やボトルネックを削減しながら低遅延リアルタイム画像処理を導入することができる。

#### 利点2: ローコストである点

もう一つの利点としては、カメラそのものが非常にローコストであるという点である。産業用USBカメラや産

業用GigEカメラなどに比べてローコストにできる理由は、カメラのハードウェア側に通信IFに変換するためのチップやカメラ制御用のFPGAが搭載されていないからである。カメラ側のデータストリームはセンサーからの信号をLVDSで直接出力するので、カメラは必要最低限のパーツで構成することができている。JetsonなどのPCはMIPI CSI-2インタフェースとISPがダイレクトに直結されており、ISP側でベイヤー変換・色補正・色空間変換・ホワイトバランス・レンズ補正・画像データ圧縮(H.264/H.256など)などのハードウェアアクセラレーション処理ができる。つまり、Jetsonなどのエッジコンピュータに実装されたアクセラレータに処理を任せるとMIPIカメラはローコストを達成しているという事である。

#### 注意点や他のインタフェースに比較した時のデメリット

MIPIのカメラを用いたエンベデッド



図3 産業用カメラ、産業用UVCカメラ、MIPIカメラと大きく3つに分類した時の開発ノウハウイメージ。

ビジョンシステムの構築は、カメラメーカーがドライバーやSDKを提供しているUSBやGigEカメラと比較してハードルは高い。上記で示したようにカメラはLVDSデータ信号を垂れ流しているだけなので、PC側でこういったデータなのか、こういったデバイスなのかを定義した上で適切にデータ処理する必要がある。既にそれらの定義が実装されているJetsonであればドライバーなどデバイスの制御環境は整っているが、自前の回路でMIPIコントローラに接続して制御する場合はデバイスとして制御する方法から把握しなければならない。

また、実際の開発に当たってはカメラ制御用のソフトウェア制御にてオープンソース関連(v4l2、Gstreamer、libargus)の知識も必要になってくる。またカメラ制御用のオープンソースは、オープンソースという性質上動作保証やサポートがなく自ら情報を収集しなければならない。これらの背景から、数台程度を運用したいといったケースでは、“ドライバーやSDKが準備されサポートが受けられるUSBやGigEなどの汎用的なインタフェースの産業用カメラ”を選択するというのが開発工数を含めたトータルコストで見ても多

い。自社開発のMIPIエンベデッドシステムが十分な販売量が見込める場合やそれらの開発資産や知識があるかどうか着手前に検討されたい。また、カメラを産業用カメラとして利用する場合は以下のようなトラブルのリスクがある。

- ・カメラがハングアップする(=止まっている)
- ・カメラへのコマンドが通らない(=ソフトウェア側からは送信されている)
- ・カメラから出力されたフレームがPCに届いていない(=カメラからは出力されている?)
- ・カメラの画像にノイズがのることがあり画像処理がうまくいかない

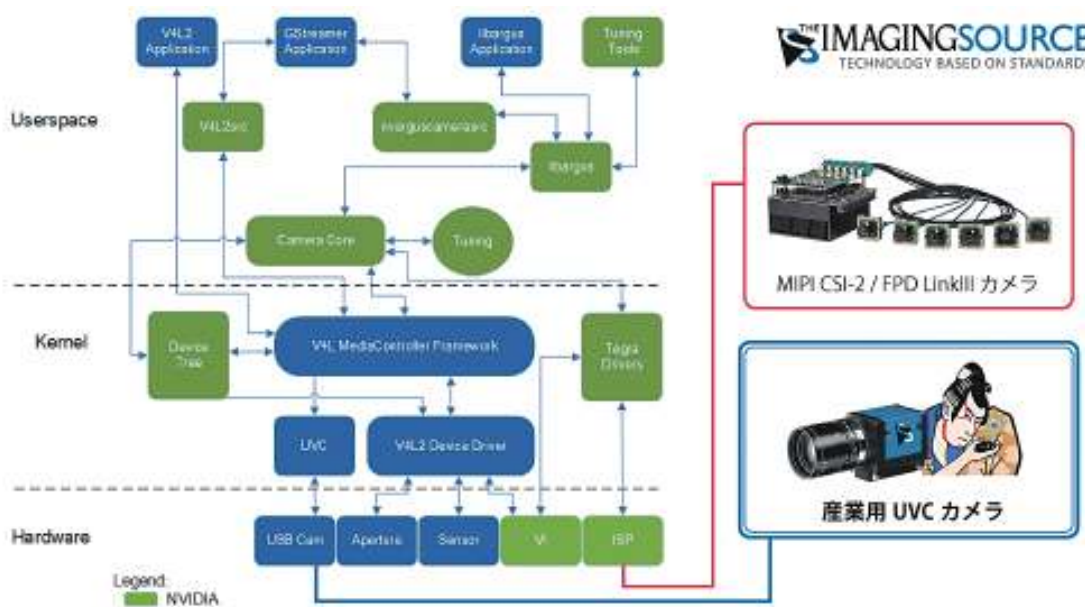
MIPIのカメラがローコストであるという理由で、カメラには必要最低限の構成でLVDS出力するのみと説明したが、そのデメリットとしてはカメラ自身が自らの状況を把握するような仕組みなどは規格化されていないため「フレーム落ちが出ているかどうかソフト側では把握できない」「デバイス設定の応答を把握できない」「デバイス設定のエラーを認識できない」などソフト側から状態を把握できないケースも多く“カメラに対する制御”と“カメラからの応答”に関連するトラブルは、状況か

ら推察しながら自ら解決しなければならない。プロトコル層からアプリケーション層までカメラ制御や安定動作を考慮して規格化されているGenicam準拠のUSB3 VisionカメラやGigEVisionカメラはそのようなトラブルシューティングやエラー処理の準備がある。産業用USB/GigEカメラは、ハードウェア的に無駄なものが搭載されて高価になっているわけではないという事である。エンベデッドビジョンシステムは、自動機つまり人の手を離れて自律動作するアプリケーションに使われるが、採用する場合は上記のような通信&制御トラブルにシステムの一環として対応する技術力を保有していなければならない。

### 産業用UVCカメラをMIPI CSI-2の踏み台にする方法

上記の通りMIPI CSI-2インタフェースカメラはローコストではあるものの少量での検討では非合理的な場合があったり、周辺知識含めて非常にハードルが高かったり、トラブルシューティングやデバッグをしなければならないという側面がある。しかし量産時のメリットも見逃せない。そこで堅実な方法としては“産業用UVCカメラ”からまずは導入してみるという方法である。産業用UVCカメラから着手する

図4 Jetsonソフトウェアアーキテクチャ。ISPがMIPIカメラ部分である。



のは、「将来的にエンベデッドビジョンシステムを大量生産するという展望があるが今ではない、ただノウハウは蓄積しておきたいし、ノウハウが他のビジョンシステムに実践的にすぐに応用もできるようにもしておきたい」という人に向いている。いわゆる産業用カメラ、産業用UVCカメラ、MIPIカメラと大きく3つに大別した時に開発ノウハウに係るイメージを図3に示した。

産業用UVCカメラの一部はUSB3Visionにも対応しておりカメラ側には安定動作に必要なFPGAが搭載されている。それらの機能はドライバーやSDKでサポートされておりユーザーアプリケーションからコントロールが可能である。一方の産業用UVCカメラはそれらのドライバーやSDKを利用できる事に加え、汎用UVCカメラ(=WEBカメラ)としてLinuxのv4l2で制御ができUSB3VisionデバイスとしてはAravisでも動作をさせることができる。これらのオープンソースはMIPI

でカメラ制御する場合のアプリケーションレベル(ユーザスペース)と共通している。図4はJetsonのソフトウェアアーキテクチャであるが、ISPがMIPIカメラ部分である。Tegra Driverを使って最終的にはGStreamerなどのオープンソースで画像取得ができる。例えば手軽に各種画像や動画データ、カメラデバイスなどを制御するオープンソースライブラリGStreamerで説明するが、下記コマンドはJetsonが持つISPを経由してデータ取得するMIPI CSI用ソースnvguscamerasrcの例である。gst-launch-1.0 nvguscamerasrc ! videoconvert ! ximagesink

これをUVCデバイスなどv4l2用ソースのv4l2srcで置き換えると下記コマンドとなる。gst-launch-1.0 v4l2src ! videoconvert ! ximagesink 要はUVC

カメラもMIPIもGStreamerやv4l2で制御ができるという事である。つまり、産業用UVCカメラを選択しGStreamerとv4l2での開発資産を構築しておけばそこから産業用カメラやMIPI CSI-2エンベデッドカメラシステムへとシフトしたりする時にはハードルが低くスムーズなのである。産業用UVCカメラを採用していれば、開発資産を活かして将来的に用途に応じた適切なカメラを使い分ける事ができ拡張性や自由度が大きくなる。日本にない技術を世界からというテーマで最先端技術を輸入し産業用UVCカメラを取り扱っているアルゴでは、それらのカメラ制御プログラムを属人的にサポートしており、カメラデバイスやインタフェース、レンズに至るまでユーザーのアプリケーションや用途をベースにそれらをゼロから選定している。

著者紹介

西田祐矢氏は、アルゴの営業技術担当。URL:www.argocorp.com

VSDJ