



IoTEdge開発の手引

RaspberryPIベース

NXP ARM-SOCベース

マイコンベース



バックグラウンド

- RaspberryPIは現世代のPCと言え、数多くのシステムのPocがRaspberryPiで開発され、製品にも使われ始めています。廉価という強力な武器に加え、RaspberryPI用のアプリ(Windows版などより安い事が多い)もあり、開発環境はタダだし、開発自体も容易で安くできるというイメージ(これ重要)が流布しています。
- 加えてCM3+など、動作可能温度が -20°C ~ 70°C という産機向けのSOMモジュールもリリースされ、アプリオリエンテッドのCarryボードを作ることによって、完全な産機向けソリューションとして利用できるようになりました。これでモジュール自体の製造保証が10年に満たずとも、ソフト互換を確保できるようになり、この点でもPCと同様のポジションを有したと言えるのでしよう。

※ 日本国内向けに開発・製造したって勝てるわけ無いです。

モジュールモデル1

RaspberryPIをIoTのエッジとして使うケース

CPUモジュール: RaspberryPI CM3+ (-25C to +80C.)

1GByteLPDDR2 8GByteeMMC HDMI MIPI-CSI UARTx2 他

キャリアボード仕様

PoE受電機能付きLAN USB-HOST GROVEx3

HDMI LVDS変換対応 LTE通信モジュール

ケース

板金でセミカスタム対応可能

※ IOTEdge-RasCM3+として開発中



上LTE通信モジュール例

右板金ケース例



モジュールモデル2

パネコン型

タッチパネル付きLCDとセットとなるよう、HDMIをLVDSなどに変換し、BOX型と同様に放熱は強化する。

通信IFとしてLANに加え、RS485(ModbusRTUなど)、CAN(CAN Openなど)、EtherCATなどが付加できるスロットを有する。



左記のようにパネルの後方にBOX型板金ケースがマウントされる形状となる。
ケース自体はモデル1の板金ケースと同系である。



本当にRaspberryPIで良かった？

- 但し、企画製品の仕様や品質基準によりRaspberryPIより、TIやNXPなどのARM-SOCでボード設計し日本で製造する方が良いと思われるケースや、IoTEdgeシステムとしてRaspberryPIで開発されてはいるが、本当は電池駆動で作るべきで困っているなど、問題は多々あります。
 1. ターゲットの要求サイズが小さく、CM3でもSBCタイプでも実装困難な場合、特殊形状で基板を合わせる必要がある場合。
 2. 放熱が難しく温度特性が悪いシステムでは、RaspberryPIの発熱量は大きな問題となる。
 3. 10年以上置きっぱなしなどタフな環境仕様が要求される場合。
 4. 電池駆動で3年以上などの運用動作が必要なシステム。



モジュールモデル3

NXPの最新のSOCの場合

NXPの最新のSOCであるi.MX8Mmini/nano/Plusが推奨ですが、RasPIに比べ開発費も製品価格も高くはなりますが、

- 1) 14nmのCortexA53(Max1.8GHz)で、RaspberryPIに比べ発熱は低くヒートシンク無も可能で、小型基板には向いている。
- 2) マイコンコアとしてCortexA4/M7を内蔵し、IO専用マイコン処理をしたい場合などに向いている。
- 3) NXPは自動車用などでは15年の製造保証を明記している珍しい会社であるが、これによりボードを10年保証することが可能
- 4) SOM型モジュールを用意しており、RasPIのようにCarryだけ開発しても良いし、SOMレイアウトを使って1枚に集積も可能です。



モジュールモデル4

IoTEdgeはマイコンのほうが良いのでは？

- 1) RaspberryPIでも他のARM-SOCでも、電池駆動ができないとは言わないが、何千mAのリチウム電池で1日だろう。単3一個で3年を期待するシステムは多く、ソフトウェアも単純でいいものも多い。
- 2) 特定の物質の状態をチェックし一定量になったら通知するとか、1時間とか1日に一度見ればいい対象(水道メーターなど)であれば、基本止まっていて一定時間毎にWakeUP&センス&必要時に通信するシステムは、RaspberryPIもLinuxも不要でしょう。
- 3) マイコンなら温度環境は -40°C ~ 105°C 、DRAMレスでノイズにも強く、WakeUPに遅くても1msから速いと $1\mu\text{s}$ とかで、基本止まっていて良く、動作電流自体もプロセッサの100分の1とかである。

Carryなどによるハードウェア拡張

- 写真は一般的なCarryの例です。
 - 1) HDMIではなくLVDSなど一般的なLCDに接続する。
 - 2) USBポートをUSB-HUBチップにより拡張する。
 - 3) GPS/LTEなどを拡張する。
 - 4) ProfiBus・EtherCATなどのフィールドバスを拡張する。
 - 5) Groveなどで市販センサーを拡張
 - 6) PoEでACアダプタを不要にする





OSのチューニングなどソフト拡張

私達は長年、ハードウェアとユーザーの要望に合わせて、LinuxだけでなくRealTimeOSなどもチューニングしてきました。

ドライバーの追加やカスタマイズに加え、LinuxカーネルへのRealTimeパッチによるRealTime性の補強なども行ってきました。

Linuxが普通にRealTimeOS化しますし、丈夫にもできます。

何故かRaspberryPIにおいては、特に日本ではそのまま使うケースが多く、例えばパーティションを分けてシステム部をReadOnlyにする、などさえやっていないケースが見受けられます。

これは本質的にRaspberryPIというシステムの趣旨に反しており、日本のエンジニアは能力が低いです、と言っているようなものです。

ハードとOS部をセットでチューニングすることで、RaspberryPIをもっと使えるようにします。