

第9回NCESシンポジウム

衛星搭載ソフトウェアの構築技術 ～ Software Defined Satellite(SDS)に向けた要素技術～

名古屋大学 大学院情報学研究科
附属組込みシステム研究センター
研究員 高田光隆

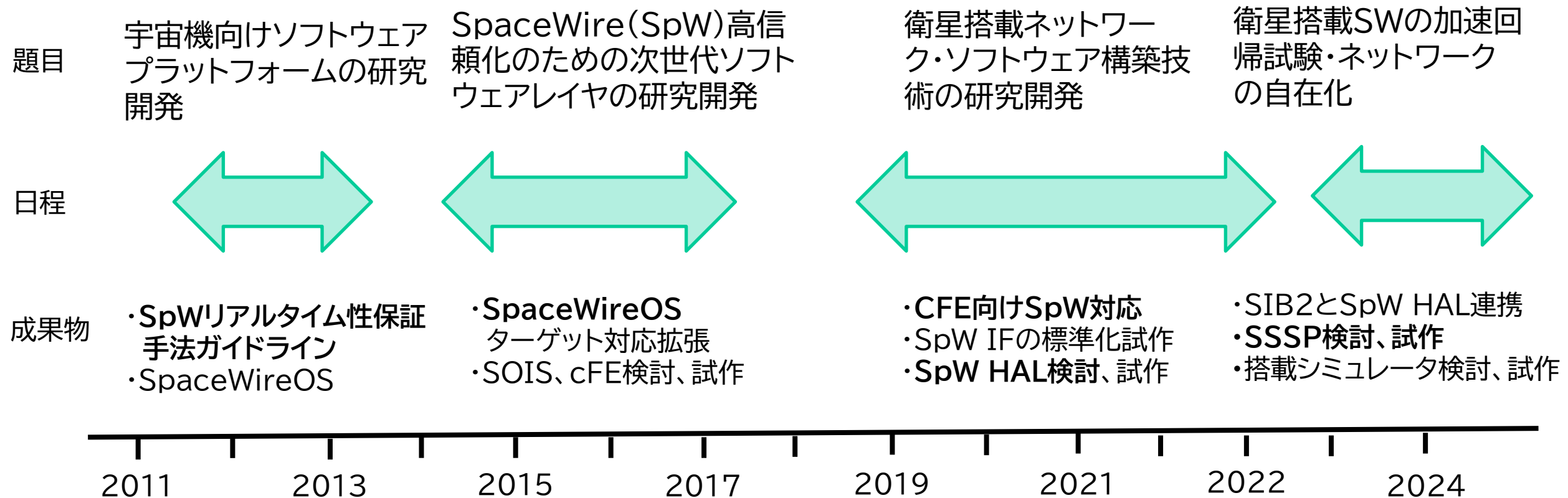
目次

- NCESとJAXAとの共同研究の取り組み
- 背景
 - 近くて遠い宇宙開発がより身近に
 - Software Defined SatelliteとSatellite as a Service
 - 衛星開発プロセスの向上
- 衛星加速試験のシミュレータエンジン
 - シミュレータの要件
 - Spacecraft System Simulation Protocol
 - 通信シミュレータの比較
- まとめ、その他

NCESとJAXAとの共同研究の取り組み

JAXAとの協力協定に基づき

スペースソフトウェアプラットフォーム連携研究開発拠点 をNCESに設置(2011)



背景: 近くて遠い宇宙開発がより身近に

これまで(単品開発)

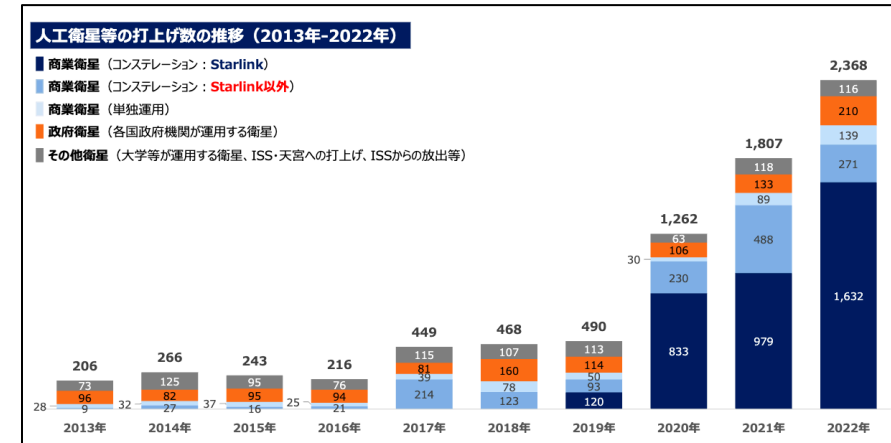
- 科学技術衛星などプロジェクトごとの開発
- 放射線対策などの宇宙グレード品による高価な部材を利用
- 採用メーカーごとに提供されるOSやミドルウェアの上でミッションアプリケーションを開発

現在、これから(量産開発)

- 国内でもJAXAだけでなく、民間によるロケット開発が進んでいる
- 打ち上げの相乗り、ISSからの放出など大学や民間によるSmallSat(超小型衛星)やコンステレーション衛星により打ち上げ数が急伸
- 車載など民生量産品を使って衛星コストを低減
- 民間の衛星ビジネス(衛星データ、衛星通信)が立ち上がり



ソフトウェア開発の向上とサービスがより重要



内閣府: 宇宙輸送を取り巻く環境認識と将来像より

Software Defined Satellite(SDS)時代に向けて

- 衛星サービス/SataaS (Satellite-as-a-Service)
 - 衛星放送事業、気象観測
 - 測位 (GPS、準天頂)
 - 地球観測 (光学、熱赤外、マイクロ波)



単機能の提供から、サービスを提供(SataaS)する基盤づくりへ

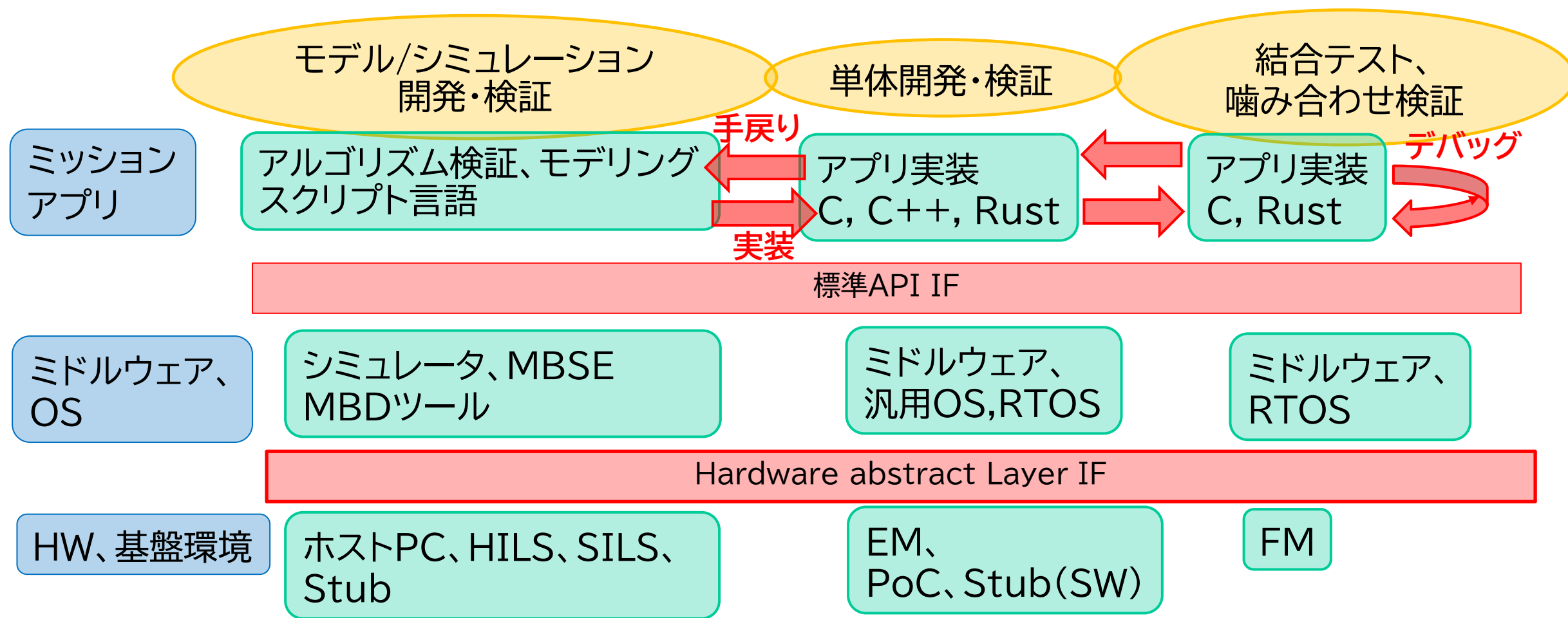
- SDS はSataaSを実現するための基盤技術
 - **SDSは衛星のハードウェアとソフトウェアの分離 (標準化)**
 - 軌道上で衛星の機能を**柔軟に変更**することを可能にする (OTA)
 - 衛星オペレータは顧客ニーズに応じた**サービスをカスタマイズ (PF)**



実際はまだまだソフトウェア開発の効率化余地は残っている

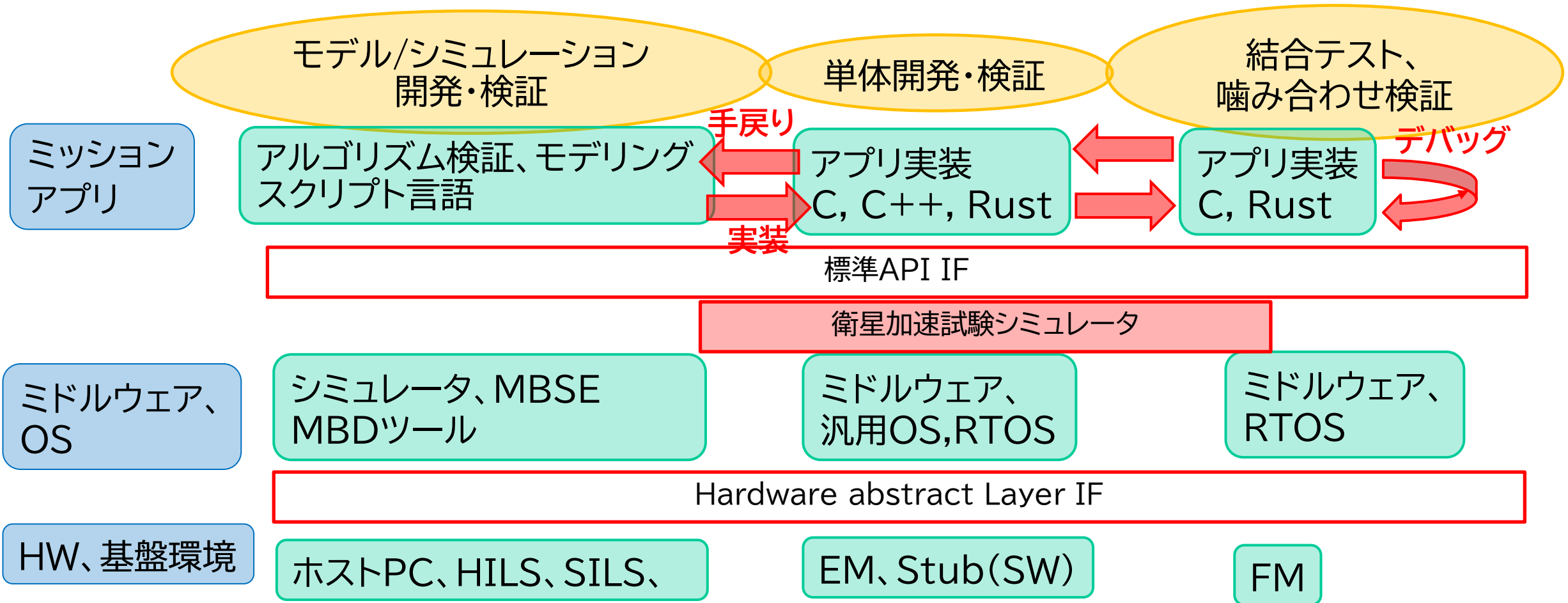
標準化による手戻りしやすい環境づくり

- 標準API IFを提案し、OS・ミドルウェアでも利用できるアプリケーション開発環境
- 標準API、HAL IFに対応したターゲットドライバ、ミドルウェア、**シミュレーション**の構築



標準化による手戻りしやすい環境づくり

- 標準API IFを提案し、OS・ミドルウェアでも利用できるアプリケーション開発環境
- 標準API、HAL IFに対応したターゲットドライバ、ミドルウェア、**シミュレーション**の構築



衛星加速試験に向けてのシミュレータエンジンの検討

要件・要求

1. SILS環境（HILS含む）でのプログラムロジック・タイミング検証（主にコンポーネント間の通信試験）
2. コンポーネント間は衛星の実プロトコルをEthernetでカプセル化を行う
3. SILS実行時はコンポーネント間で時刻同期を行うことで、コンポーネント間の通信処理（シーケンス）の再現性を保てること
4. カプセル化で通信データにシミュレーション時刻情報を付加することで、シミュレーション実行時の通信データをキャプチャし、別途通信の検証が行えること
5. シミュレーションの加速実行

Spacecraft System Simulation Protocol(SSSP)

- 概要

- ADU(Application Data Unit)にPCAPヘッダを付与してUDP/TCPで伝送
- ADUにはSpace Packet Protocol(CCSDS) と SpaceWireを想定
- PCAPに送信時のシミュレーション時刻を格納する

- 特徴

- SpaceWire 伝送プロトコルの可視化
- 実装が簡易のため、実搭載機器に近い環境、シミュレータ模擬環境の双方で利用できる (HILS/SILS)
- WireShark/tcpdumpなど既存のネットワークツールを利用

まとめ

- 衛星加速シミュレータ
 - 実機コンポーネントの組み合わせテストをする前に、フロントローディングとしてPC（シミュレーション）で時刻を含めたシーケンスの確認
 - SSSPの検討・実装では通信データをPCAPでカプセル化することで、簡便なシミュレーション時刻の管理とEthernet上で通信データを流せるようになることがわかった

Contacts

Please feel free to ask us if you have any questions.



Center for Embedded Computing Systems,
Nagoya University Tel : 052-789-4228 Fax:
052-789-4273

URL: <http://www.nces.i.nagoya-u.ac.jp/>

email: nces-office@nces.i.nagoya-u.ac.jp

NCES SpaceWire

検索

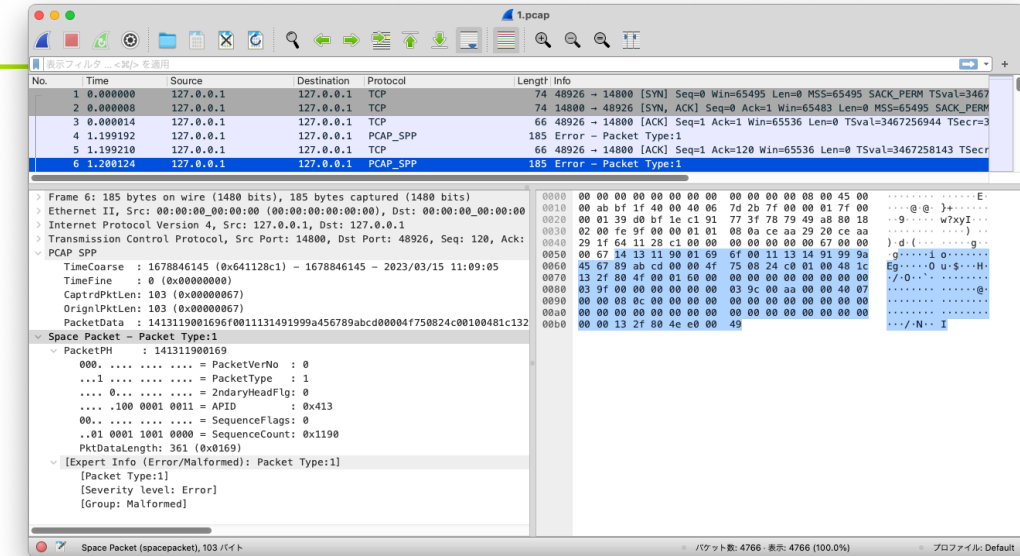
補足資料

SSSPと他の通信シミュレータの比較

	目的	構成	時刻管理	ログ	可視化
SSSP	SIL/HILによるアプリ検証	PCAP(転送データ) NCコマンド(転送処理) ソフトウェア実装(転送データ、処理) JSON(ノード)	なし(時刻マスターが必要)	PCAP	WireShark
NC3	モデル、シナリオ検証	シナリオスクリプト モデルアプリ	NC3(シナリオスクリプト)	NCフォーマット	NetAnim
OMNET++	モデルとシミュレーション(SILとして利用できるが、リアルタイム向けでは無い)	トポロジー(NED言語) モジュール間通信(メッセージパッシング) シミュレーションモデル実装(C++)	シミュレーションカーネル(イベントスケジューリングと実行)	ベクトルファイル(時系列) スカラーファイル(統計情報)	専用ツール

SSSPの実装例

- 実装プロトタイプとして**pcap-nc**が公開
 - <https://github.com/matsuzaki-keiichi/pcap-nc>
 - ncコマンドをベースにSSSPを伝送する
- WireSharkのプラグインも開発

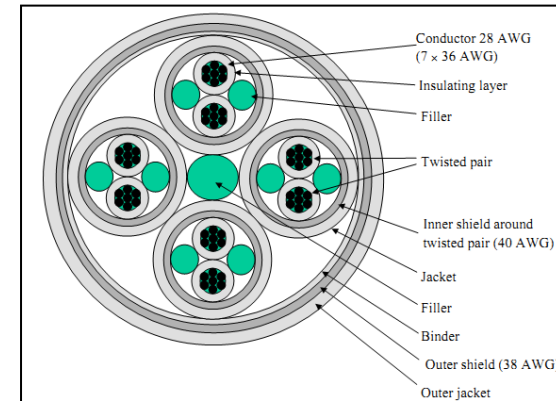


- NCESでは既存研究成果（搭載ソフトウェア）をベースにSSSPを適用
 - pcap-ncとの相互疎通通信を確認
 - アプリの通信IFは既存の研究成果（標準API IF）を利用
- アプリケーションを**実装環境**から**シミュレーション環境**へのシームレスな移行が可能



SpaceWireの特徴(1/2)

- 宇宙機内の機器間データ通信インターフェース規格
 - ESA（欧州宇宙機関）がIEEE1355規格をカスタマイズして提案(1994) (ECSS-E-ST-50-12C)
 - LVDSによるノイズに強い高速シリアルライン
- 幅広い転送速度（2-400Mbps）
 - データ転送レートが可変
 - Gbps向けのSpaceFibre規格も登場
- 可変長のパケットサイズ
 - データサイズの規定は無い
- 簡単なプロトコル（省リソース, 省電力）
 - FPGA実装のノードやルータも多数
 - 上位プロトコル(RMAP, SpaceWire-D, SpaceWire-R)があり、HW化の研究
- 中大型衛星では採用が進んでいる（大型のSmallSat(8U以上)でも搭載が始まっている）



SpaceWireの特徴(2/2)

- タイムコードによるノード間の時刻送信・同期機能
- メッシュ型のネットワーク
 - 冗長経路を持つことで通信の障害耐性が強い
 - ノードを組み上げていくときに柔軟なネットワークが不可欠
- ワームホールルーティング ルータを介して接続
 - ルータ内のルーティング遅延がほとんどない
 - しかし一度**パケット衝突が発生すると、ネットワークをブロック**する恐れ

大きな
問題点

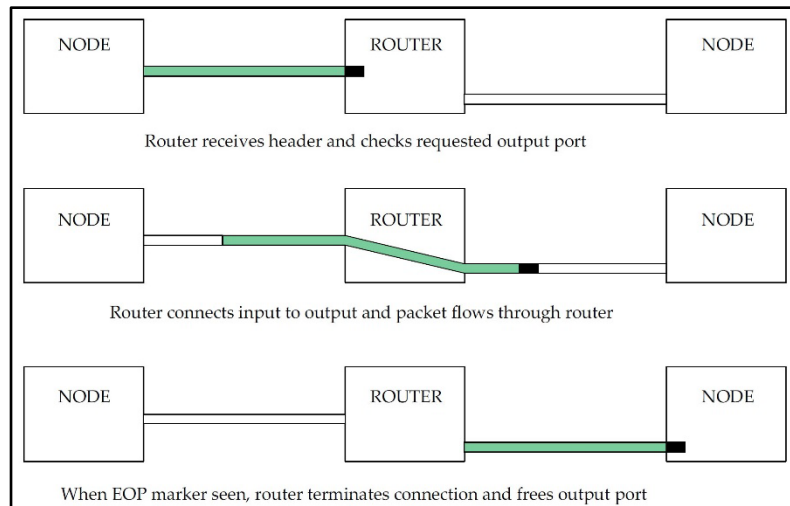


図.ワームホールルーティングの例(ECSS-E-ST-50-12Cより)

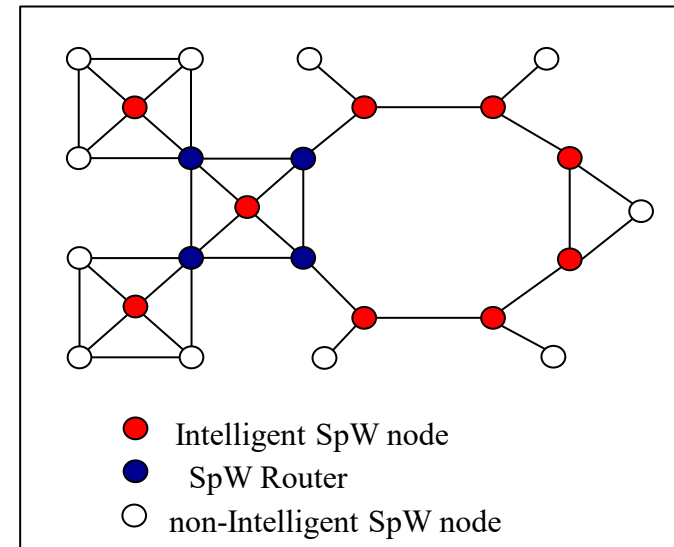


図.SpaceWireネットワーク構成例

SpaceWireの主な上位層プロトコル

RMAP : Remote memory access protocol(ECSS-E-ST-50-51C)

- SpaceWireの上位レイヤに位置する通信プロトコル
- ネットワーク越しのノードに対して, メモリマップドI/O機能でアクセスできる仕組みをそなえたプロトコル

CCSDS: Consultative Committee for Space Data System

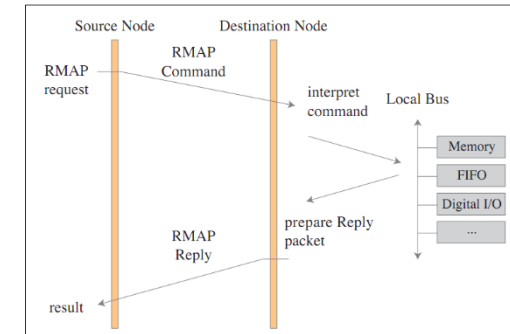
- 宇宙データシステム諮問委員会で決められたプロトコル
- 主に地上系とのコマンド・テレメトリ通信で利用されている
- SpaceWire-PTPとも呼ばれる(ECSS-E-ST-50-53C)

SpaceWire-D : Deterministic(本研究で改良案を提案)

- タイムコードを利用した時分割通信方式
- 時分割されたスロットに設計で予め送信するパケットを決定

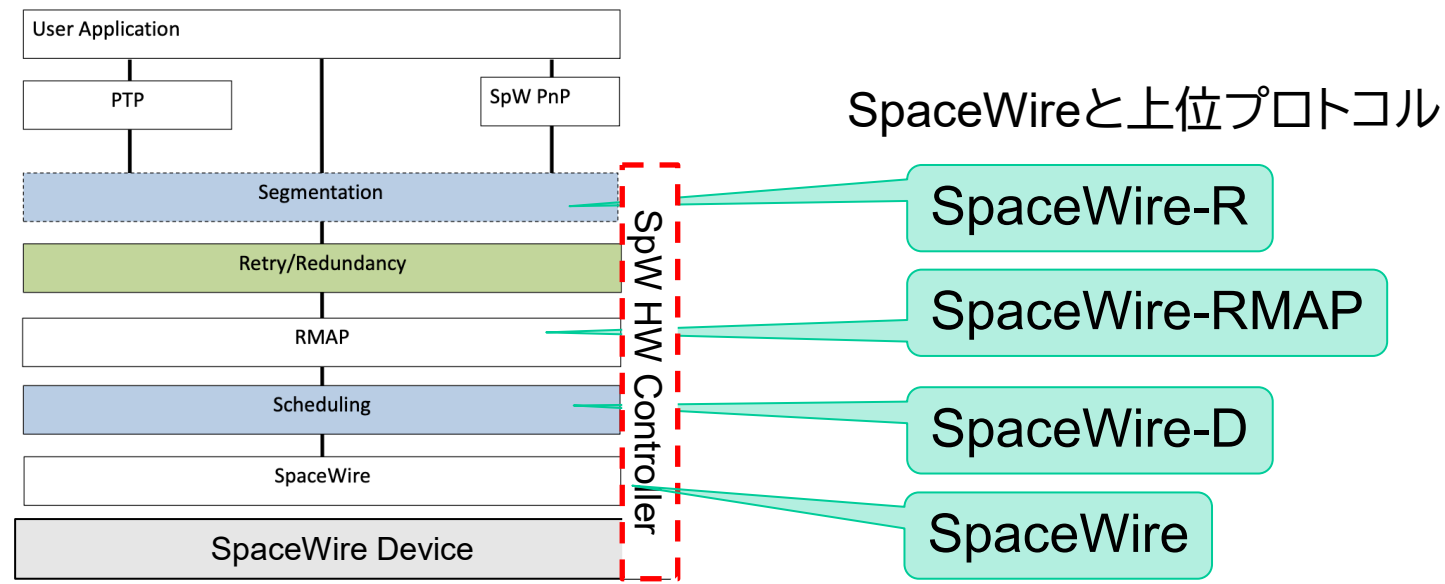
SpaceWire-R : Reliability(JAXAが提案)

- トランスポート層(一部セッション層)の役割を担う
- データの多重化, データの分割／復元, フロー制御
- シーケンスチェック, ハートビート(オプション)



SpaceWire導入の現状とハードウェアの進化

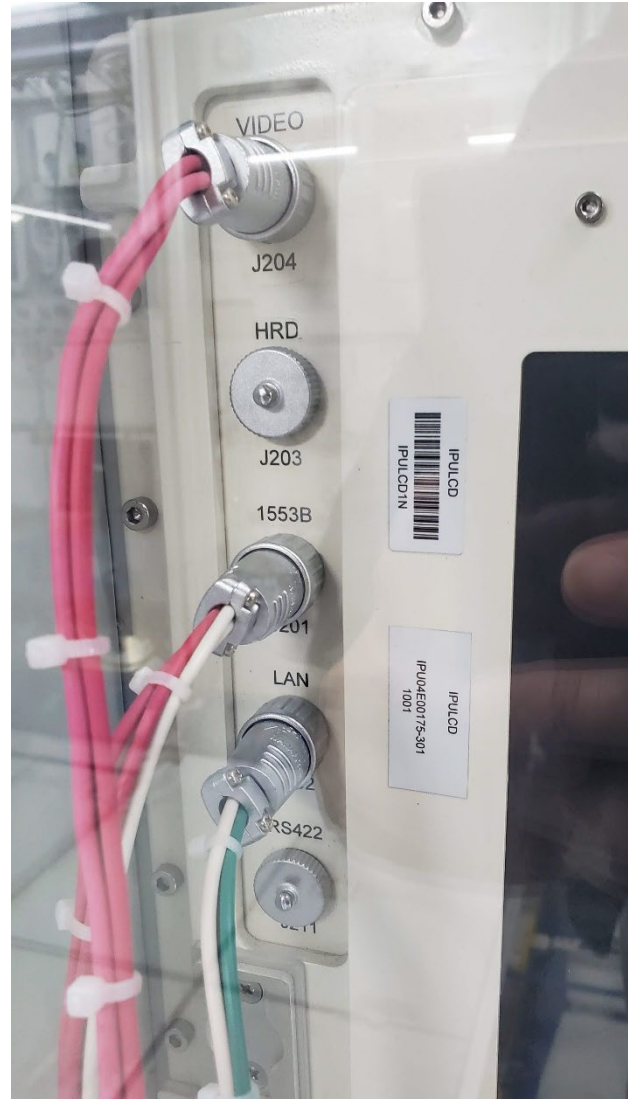
- SpaceWireは衛星内データ通信プロトコルとして利用
 - 2-400Mbps(次世代プロトコルのSpaceFibreではGbpsを実現)
 - イプシロン、ひさき、あらせなどのデータバスIF
 - JERG-2-432でオンボードサブネットワーク設計標準が公開
- 上位プロトコルが存在し、プロトコルスタックのハードウェア実装も



SpW-D Protocol Draft B よりプロトコルスタックの構成図を引用し、一部追記

主な衛星内通信プロトコル

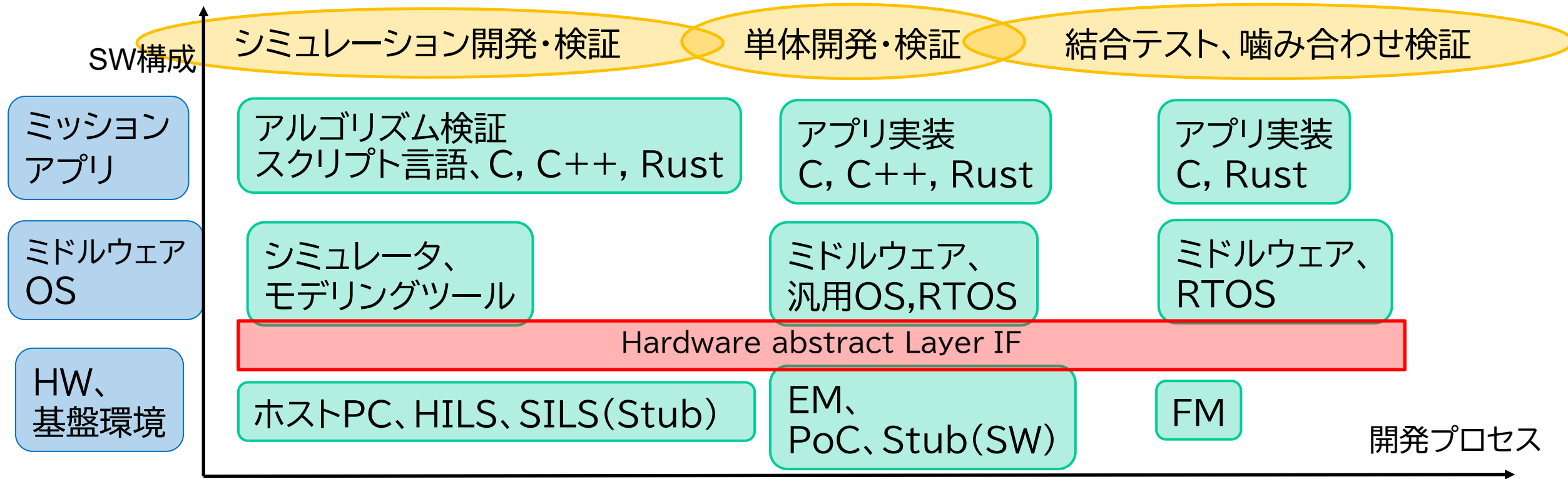
プロトコル	速度	JAXA規定	主な用途
MIL-STD-1553B	-1Mbps	JERG-2-431	オンボード通信(バス系)
SpaceWire	2-400Mbps	JERG-2-432	
CAN	-1Mbps	-	
RS422	-10Mbps (1.6Mbps)	-	オンボード、センサ、アクチュエータ
SPI	-50Mbps	-	センサ、アクチュエータ
I2C	-1Mbps	-	
PC-104(ISA,PCI)	480Kbps		CubeSat
Ethernet	100Mbps	-	ISSなどのモジュール



かがみはら航空宇宙博物館にあるISSきぼうモジュールのコネクタ

衛星開発プロセスとソフトウェア

- 衛星開発ではシミュレーションによる開発・検証が必須
- シミュレーションで検証したソフトウェアはそのまま利用することは少ない
- ミドルウェア、デバイスドライバIFが異なることで生じるアプリケーションの再開発
- NW設定ミスなど噛み合わせ時の他SpWデバイスとの結合確認による手戻り



SpaceWire ミドルウェアの開発実績

SpaceWireOS(ISC2013,宇科連56,57)

- MHI製SOI –SOC向けに名大で開発
- TOPPERS/ASP,HRP2
- RMAP, SpaceWire-D
- **SpaceWire HAL実装確認のため同一デバイス(SpacePi)のTOPPERS環境へ**



@シマフジ電気

SpaceWire RMAP Library(ISC2022,宇科連66)

- POSIX API向け(C++)、地上でのPoC利用
- RAW, RMAP, SpaceWire-R
- **SpaceWire HAL実装確認のため同一デバイス(SpacePi)のLinux環境へ**
- <https://github.com/yuasatakayuki/SpaceWireRMAPLibrary>



@シマフジ電気

SBN(cFS)(FSW2017,2018)

- POSIX, RTOS(TOPPERS/HRP2を移植)
- MHI製SOI –SOC, SpacePi
- SpaceWire-RAW対応実装と通信確認を行った

