



メリディアン計画

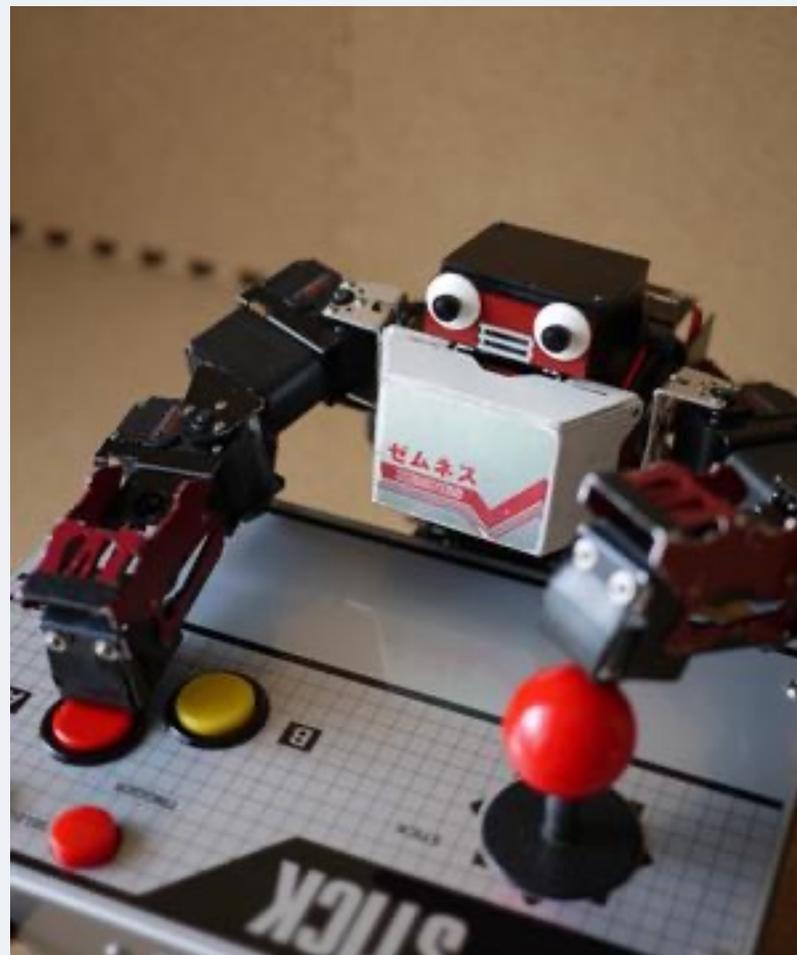
ROSとヒューマノイドをつなぐシステム「Meridian」の提案

ROSCon JP 2022

2022.10.19

Izumi Ninagawa

Ninagawa Izumi
二名川 和泉



汎用人型決戦遊具ゼムネス

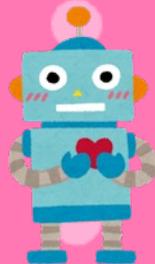


競技用動的人体模型スコブル

HOBBY HUMANOID ROBOT

MEETS

ROS





urdfviz - RViz

Time

ROS Time: 1666014540.73 ROS Elapsed: 1764.22 Wall Time: 1666014540.75 Wall Elapsed: 1764.22 Experimental

Reset Left-Click: Rotate Middle-Click: Move X/Y Right-Click/Mouse Wheel: Zoom, Shift: More options 31 fps

```
roscore http://...
* /rosversion: 1.15.14
NODES
auto-starting new master
process[master]: started with pid [91237]
ROS_MASTER_URI=http://nina-ubuntu-nba:11311/

setting /run_id to bd324230-4e1f-11ed-8add-f
df4cfef8a9a
process[rosout-1]: started with pid [91251]
started core service [/rosout]
```

Meridian_Console_v22.0730

Axis Monitor			Sensor Monitor							
4.850	R0	-0.330	L0	0.000	ac_x	0.000	ac_y	0.000	ac_z	
4.100	R1	-0.960	L1	0.000	gr_x	0.000	gr_y	0.000	gr_z	
5.090	R2	11.070	L2	0.000	mg_x	0.000	mg_y	0.000	mg_z	
39.950	R3	-0.130	L3	0.000	temp_int16					
-27.480	R4	-20.050	L4	0.000	rol	0.000	pit	0.000	yaw	
5.070	R5	11.340	L5							SetView
7.990	R6	-1.090	L6							
-21.460	R7	-2.530	L7							
20.010	R8	-12.850	L8							
-5.070	R9	15.990	L9							
-2.900	R10	-3.900	L10							
0.000	R11	0.000	L11							
0.000	R12	0.000	L12							
0.000	R13	0.000	L13							
0.000	R14	0.000	L14							

Command

ESP32-> ->ROSL

Power ESP32<- <-ROSL

<-Dense

Control Pad Monitor 0

Lx Lx Ly Ly Rx Rx Ry Ry

Lz Lz Rz Rz

Message

This PC's IP address is 192.168.1.11

UDP data receiving from 192.168.1.23

ERROR COUNT ESP-PC:0 PC-ESP:5 ESP-TS:195 TSV_Deleg:1

ERROR RATE ESP-PC:0.01% PC-ESP:0.01% ESP-TS:0.12% TsvSKIP:1.40% ESPSKIP:1.32%

SKIP COUNT TsvSKIP:1124 ESPSKIP:1035 PCSKIP:105 PCF:70231 BOARDF:~9400 0Hz

192.168.1.11

ROS_MASTER_URI=http://localhost:11311

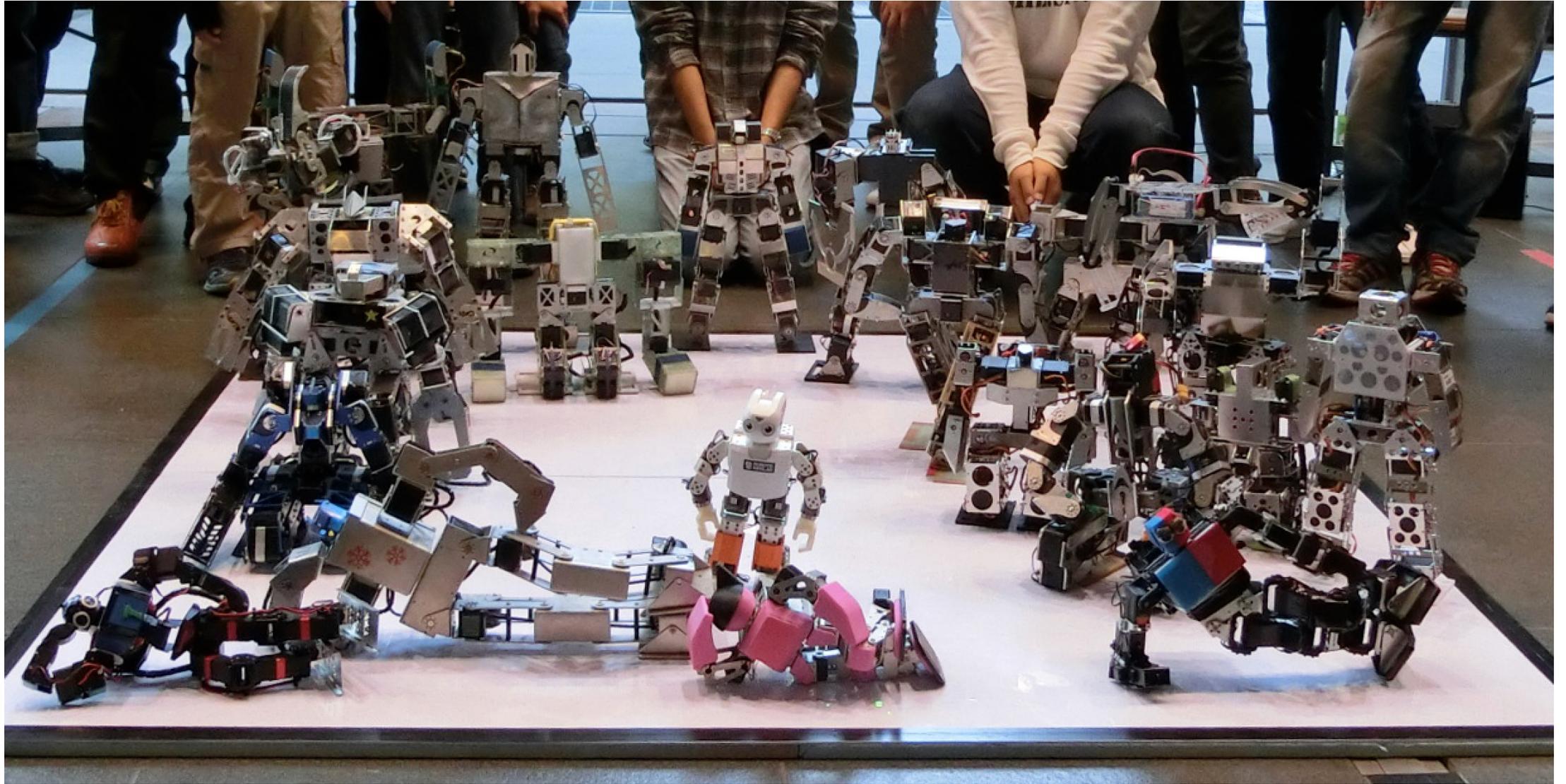
process[robot_state_publisher-1]: started with pid [90772]

process[rviz-2]: started with pid [90772]

<当研究のモチベーション>

**ロボットの研究はとても面白いので
誰もが楽しくヒューマノイドの制御を学べる
開発環境をつくりたい**

**学びと研究のエンターテインメント化
ロボット制御技術開発の”民主化”**



2016年早稲田大学理工展 ROBO-ONE認定大会

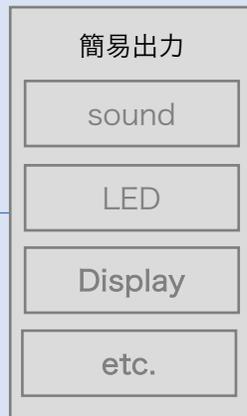
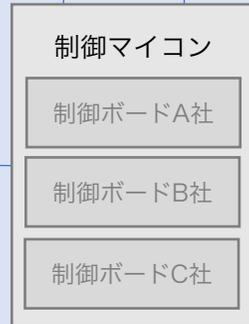
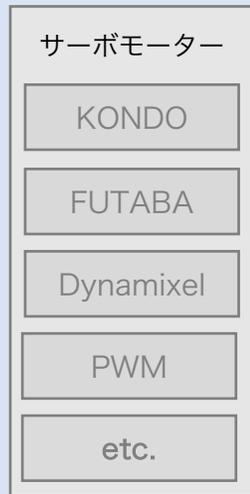
理想のヒューマノイド開発環境とは？

- ✓ 人体のような**のびやかな運動**をホビーロボットで再現できる。
- ✓ **計算制御**の学び・実装が楽しくできる。
- ✓ 各種**シミュレーター、ビジュアライザー**を使うことができる。
- ✓ **各社のコマンドサーボ**を混ぜて使える。
- ✓ センサもリモコンも**自由に組み合わせ**て使える。
- ✓ モーション作成中に**配線が絡まない**千切れない。
- ✓ **電源バチンで即起動**する。
- ✓ **データ**もぜんぶ取れる。
- ✓ **メタバースと連携**できる。
- ✓ **ROSの資産**も使える。
- ✓ システムが**非ブラックボックス**で、**可読性**が高い。
- ✓ **チュートリアル**が充実している。
- ✓ 情報交換しながら**みんなで楽しく開発**できる。

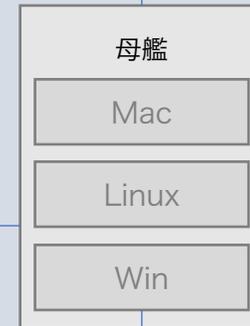


ホビロボのシステム

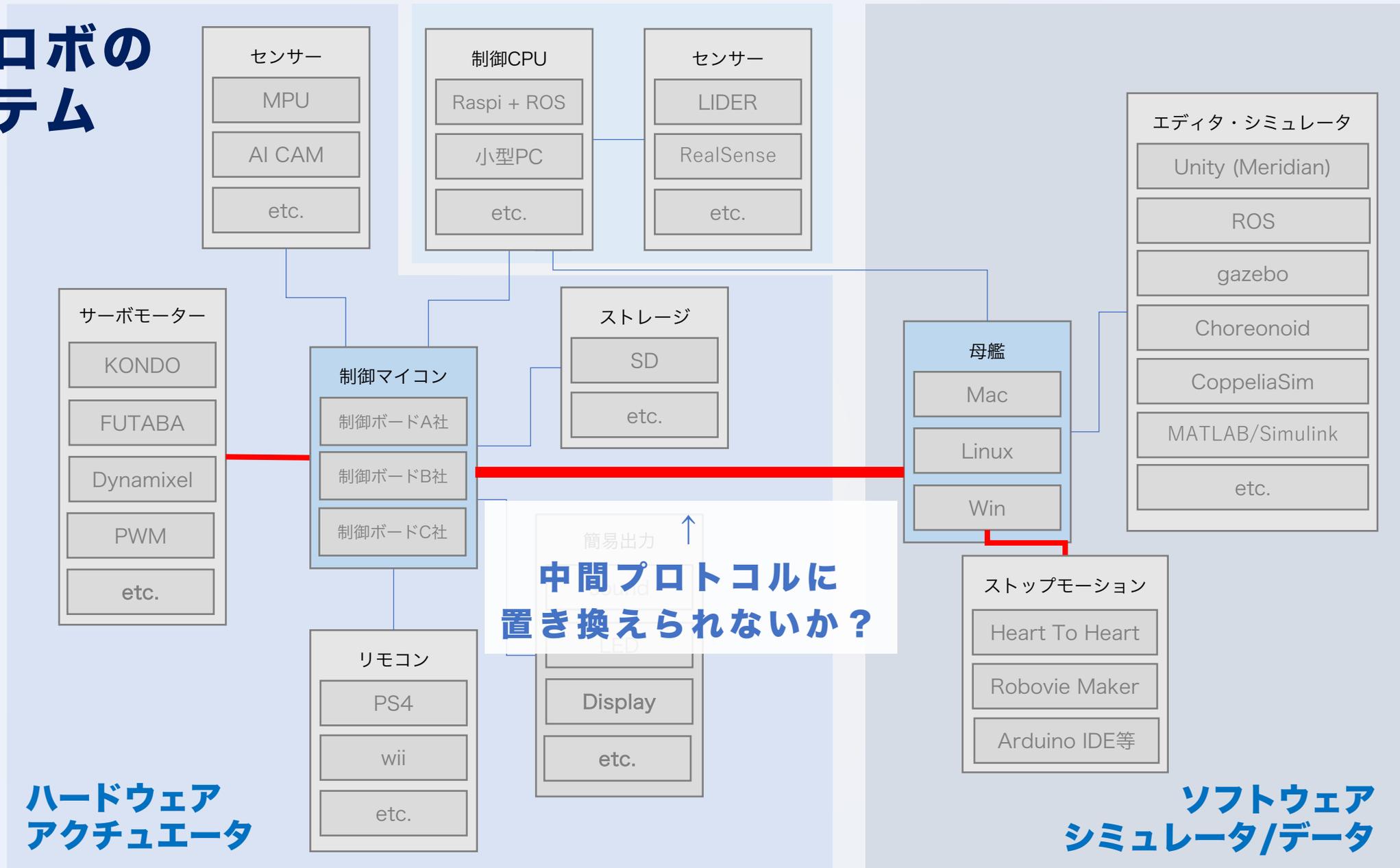
ハードウェア アクチュエータ



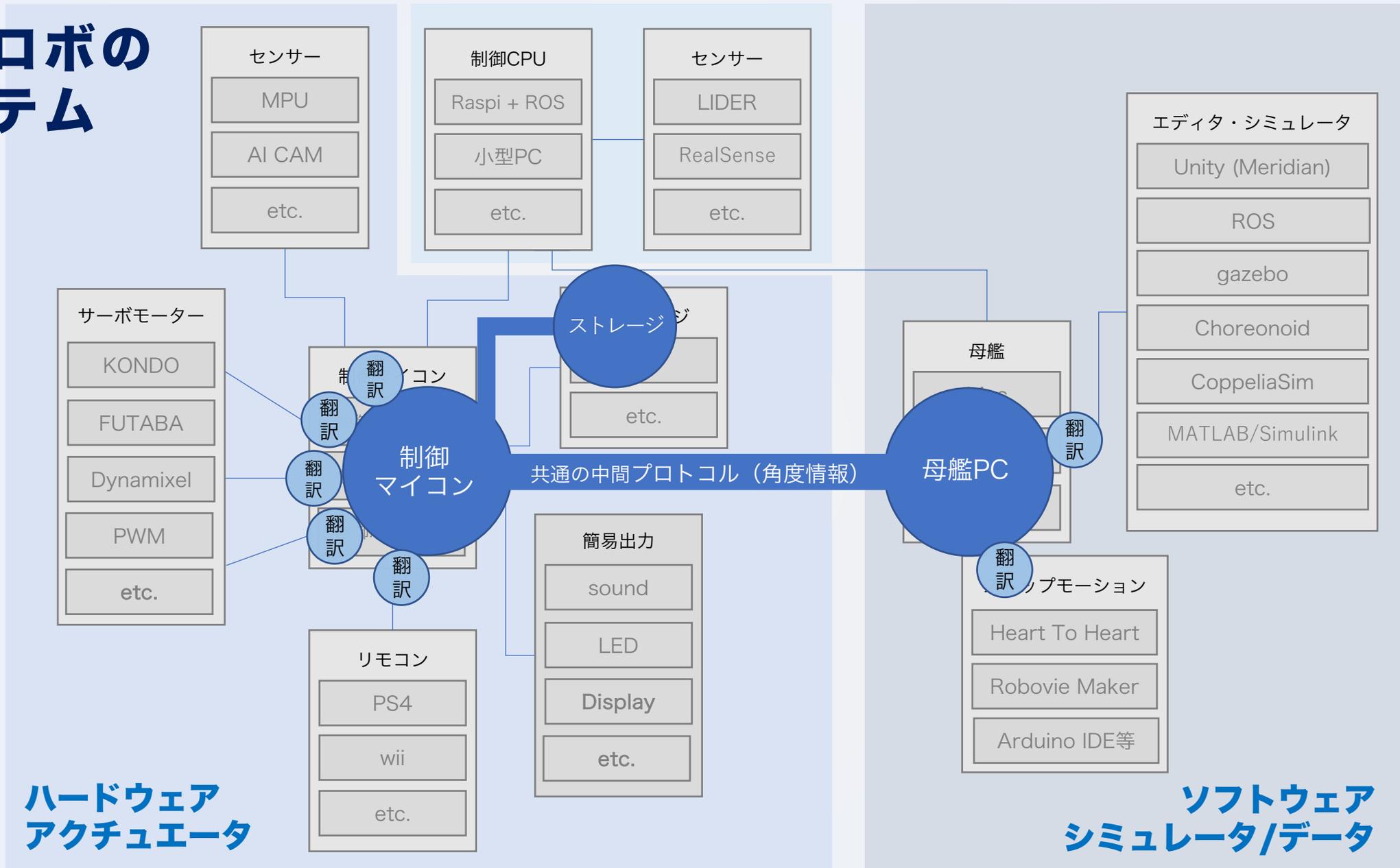
ソフトウェア シミュレータ/データ



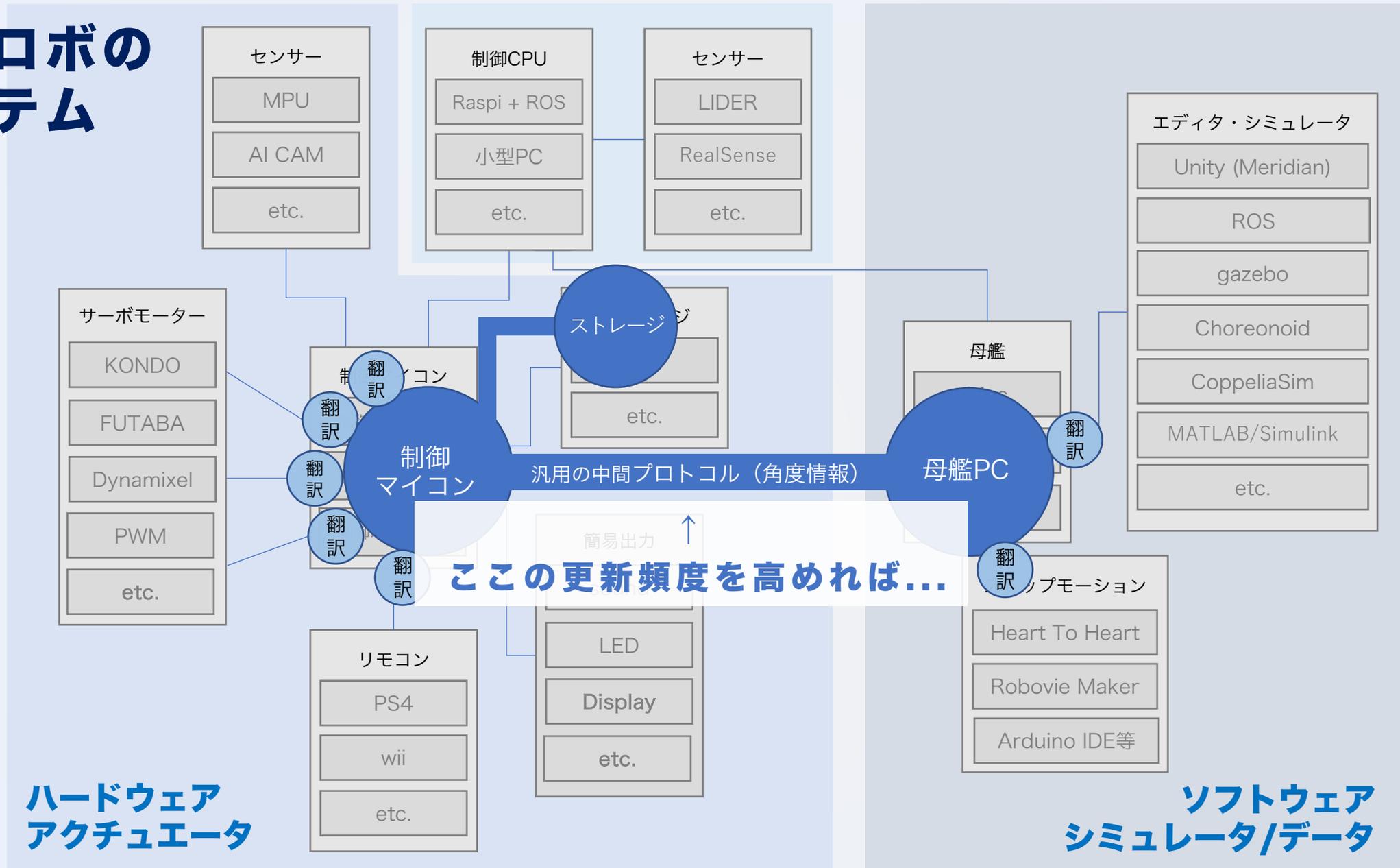
ホビロボのシステム



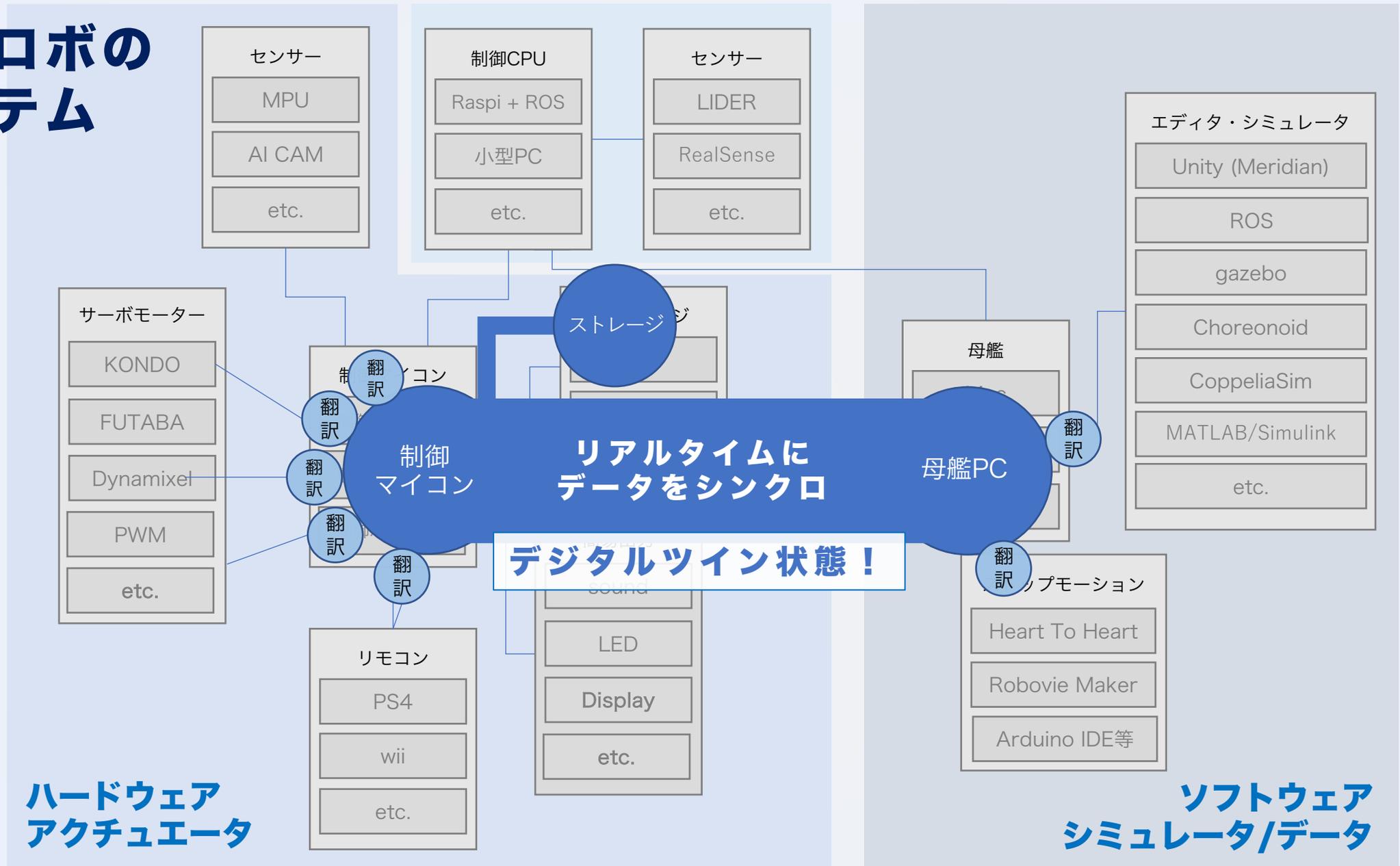
ホビロボのシステム



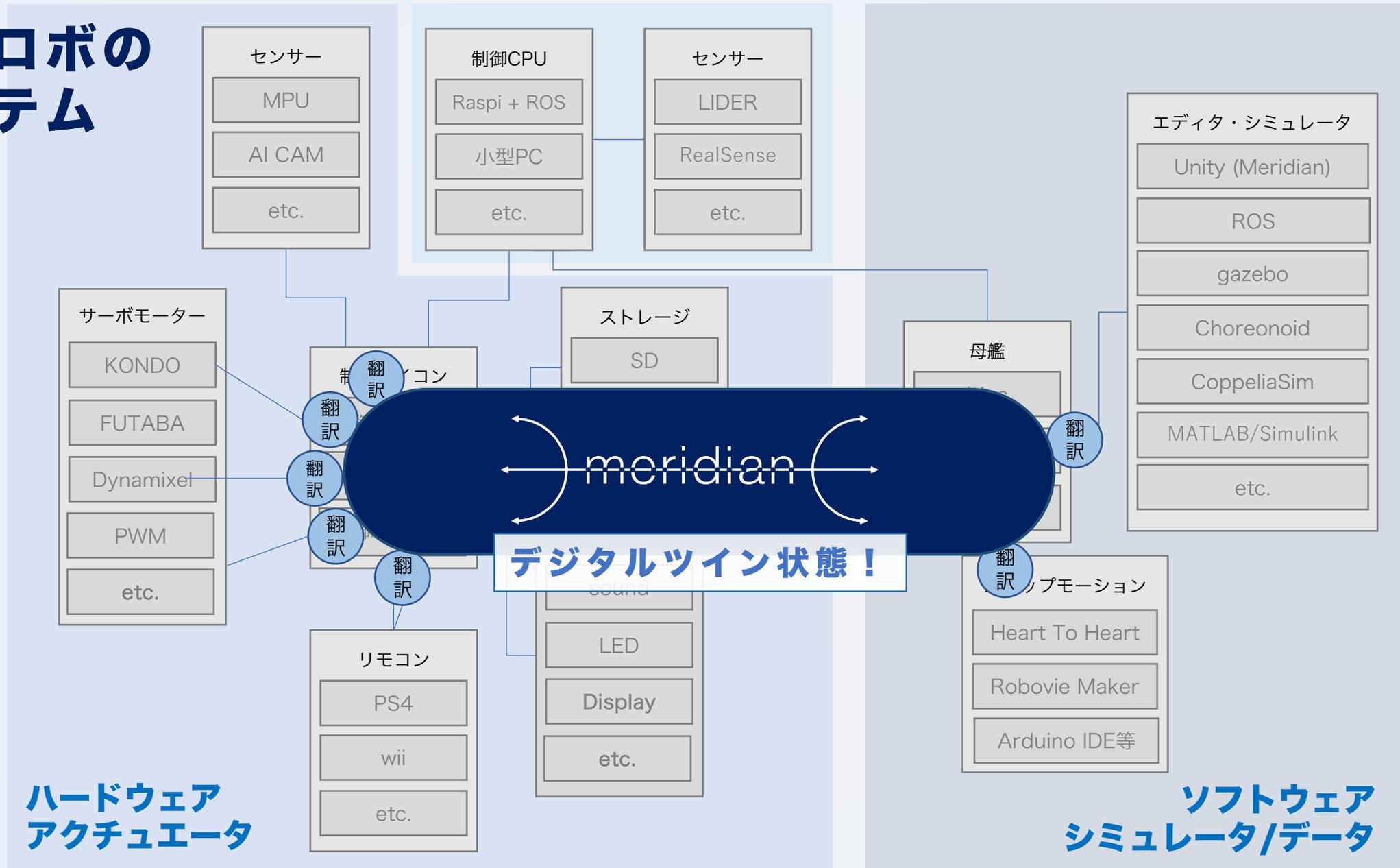
ホビロボのシステム



ホビロボのシステム



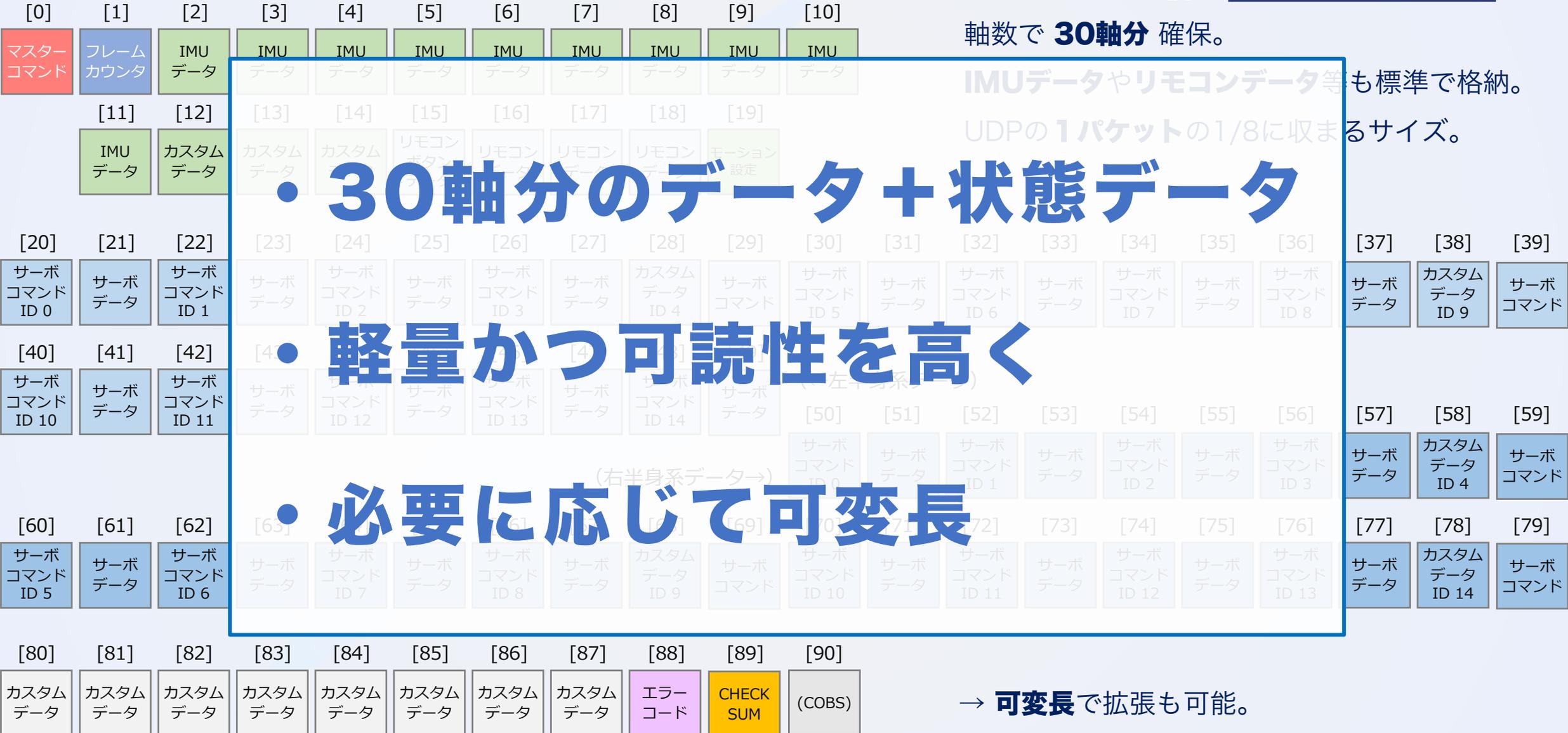
ホビロボのシステム





実現のために行ってきたこと

Meridim 90 配列 (中間プロトコルとして流すもの)



16ビット x 90個 = 180バイトのデータ。

軸数で **30軸分** 確保。

IMUデータやリモコンデータ等も標準で格納。

UDPの1パケットの1/8に収まるサイズ。

30軸分のデータ + 状態データ

軽量かつ可読性を高く

必要に応じて可変長

→ **可変長**で拡張も可能。

Meridim 90 配列 (中間プロトコルとして流すもの)



Index	内容	数値の意味	型	備考
0	Master_command	サイズフォーマットとコマンド	short	デフォルトは90
1	フレームカウンタ	フレーム同期確認用インクリメント	short	-30,000→+29,999
2	IMU:acc_x	加速度x	short	
3	IMU:acc_y	加速度y	short	
4	IMU:acc_z	加速度z	short	
5	IMU:gyro_x	ジャイロx	short	
6	IMU:gyro_y	ジャイロy	short	
7	IMU:gyro_z	ジャイロz	short	
8	IMU:mag_x	磁気コンパスx	short	
9	IMU:mag_y	磁気コンパスy	short	
10	IMU:mag_z	磁気コンパスz	short	
11	IMU:temp	温度	short	
12	IMU:ROLL	DMP推定ロール軸(小数点2位までx100)	short	
13	IMU:PITCH	DMP推定ピッチ軸(小数点2位までx100)	short	
14	IMU:YAW	DMP推定ヨー軸(小数点2位までx100)	short	
15	Controller Button	ボタンデータ1	short	
16	Controller Stick Left	ボタンアナログ1	short	
17	Controller Stick Right	ボタンアナログ2	short	
18	L2, R2 Analog	ボタンアナログ3	short	
19	モーション設定			モーションのフレーム数とイー징ング
20	[ID L0] Servo Command	0:poweroff 1:poweron etc...	short	ペンダー情報も入れ、値の書式を指定可能に
21	[ID L0] Servo Data	(degree)小数点2位までx100	short	
22	[ID L1] Servo Command		short	
23	[ID L1] Servo Data		short	
24	[ID L2] Servo Command	58 [ID R4] Servo Command	short	
25	[ID L2] Servo Data	59 [ID R4] Servo Data	short	
26	[ID L3] Servo Command	60 [ID R5] Servo Command	short	
27	[ID L3] Servo Data	61 [ID R5] Servo Data	short	
28	[ID L4] Servo Command	62 [ID R6] Servo Command	short	
29	[ID L4] Servo Data	63 [ID R6] Servo Data	short	
30	[ID L5] Servo Command	64 [ID R7] Servo Command	short	
31	[ID L5] Servo Data	65 [ID R7] Servo Data	short	
32	[ID L6] Servo Command	66 [ID R8] Servo Command	short	
33	[ID L6] Servo Data	67 [ID R8] Servo Data	short	
34	[ID L7] Servo Command	68 [ID R9] Servo Command	short	
35	[ID L7] Servo Data	69 [ID R9] Servo Data	short	
36	[ID L8] Servo Command	70 [ID R10] Servo Command	short	
37	[ID L8] Servo Data	71 [ID R10] Servo Data	short	
38	[ID L9] Servo Command	72 [ID R11] Servo Command	short	
39	[ID L9] Servo Data	73 [ID R11] Servo Data	short	
40	[ID L10] Servo Command	74 [ID R12] Servo Command	short	
41	[ID L10] Servo Data	75 [ID R12] Servo Data	short	
42	[ID L11] Servo Command	76 [ID R13] Servo Command	short	
43	[ID L11] Servo Data	77 [ID R13] Servo Data	short	
44	[ID L12] Servo Command	78 [ID R14] Servo Command	short	
45	[ID L12] Servo Data	79 [ID R14] Servo Data	short	
46	[ID L13] Servo Command	80 free	short	
47	[ID L13] Servo Data	81 free	short	
48	[ID L14] Servo Command	82 free	short	
49	[ID L14] Servo Data	83 free	short	
50	[ID R0] Servo Command	84 free	short	
		85 free	short	
		86 free	short	
		87 free	short	
※88	ERROR CODE	エラーコードを格納	short	アドレスはデータ長さ-2, 内容定義は次シート
※89	CHECK SUM		short	アドレスはデータ長さ-1, 内容定義は次シート
90	COBS	ユーザー使用不可	short	COBS処理等で使用予定
91	COBS	ユーザー使用不可	short	COBS処理等で使用予定

配列の内容

Meridim 90 Index	役割	コマンド詳細
Index[0]	マスターコマンド	30-735 コマンドの長さ指定。デフォルトは値90 0 全サーボオフ 1 全サーボコマンド受付 (PCからの操作受付) 2 全サーボコマンド受付 (Teensy等未端ボードでの制御処理を優先) 3 ヘルステック1 (内容未定だが各種状態を全て取得してレポートするシーケンスへ移行) ?EEPROM書き込み信号関連 10002 センサのヨー軸を現在値センターに 10003 トリムモードオン
Index[1]	フレームカウンタ	フレーム同期確認カウンタ・30,000→+29,999
Index[15]	ボタンデータ	bit 0 bool: select bit 1 bool: stick_R_push bit 2 bool: stick_L_push bit 3 bool: start bit 4 bool: ↑ (UP) bit 5 bool: → (RIGHT) bit 6 bool: ↓ (DOWN) bit 7 bool: ← (LEFT) bit 8 bool: L2 bit 9 bool: R2 bit 10 bool: L1 bit 11 bool: R1 bit 12 bool: △ (UP) bit 13 bool: ○ (RIGHT) bit 14 bool: × (DOWN) bit 15 bool: □ (LEFT)
Index[16]	左スティック	下位8bit char:stickR_Left -127~127 上位8bit char:stickR_Right
Index[17]	右スティック	下位8bit char:stickL_Left -127~127 上位8bit char:stickL_Right
Index[18]	L2, R2 Analog	下位8bit char: L2 Analog 0~255 上位8bit char: R2 Analog
Index[19]	モーション設定	停止時間とイーディングモードの組み合わせを指定。当面的はこの生値が以下のサーボ位置移動に費やすフレーム数
Index[20]	サーボID 0汎化コマンド	0 サーボパワーオフ、位置情報取得 (degree) 1 サーボパワーオン、サーボ位置指定 & 位置情報取得 (degree) 2 サーボパワーオフ、位置情報取得 (radian) 3 サーボパワーオン、サーボ位置指定 & 位置情報取得 (radian) 4 トルク(スピード) 設定 (電流制限?) 5 トルク(スピード) 設定 (電流制限?) 6 7 バネ硬さ設定 (ストレッチ・コンプライ) 8 9 10 11 KONDO KRSサーボ 1001 サーボパワーオフ、位置情報取得 (KONI) 1002 サーボパワーオン、サーボ位置指定 & 位置情報取得 (KONI) 1003 サーボパワーオフ、位置情報取得 (KONI) 1004 KONDO RS485サーボ 2001 サーボパワーオフ、位置情報取得 (KONI) 2002 サーボパワーオン、サーボ位置指定 & 位置情報取得 (KONI) 2003 サーボパワーオフ、位置情報取得 (KONI) 2004 FUTABA TTLサーボ 3001 サーボパワーオフ、位置情報取得 (FUT#) 3002 サーボパワーオン、サーボ位置指定 & 位置情報取得 (FUT#) 3003 サーボパワーオフ、位置情報取得 (FUT#) 3004 FUTABA RS485サーボ 4001 サーボパワーオフ、位置情報取得 (FUT#) 4002 サーボパワーオン、サーボ位置指定 & 位置情報取得 (FUT#) 4003 サーボパワーオフ、位置情報取得 (FUT#) 4004
Index[21]	サーボID 0 データ	short デフォルトはdegree:小数点2位までx100
Index[LEN-2]	ERROR CODE/CLOCK	bit 35 bool: PCのESP32からのUDP受信エラー (0:エラーなし、1:エラー検出) bit 34 bool: ESP32のPCからのUDP受信エラー (0:エラーなし、1:エラー検出) bit 33 bool: TeensyのESPからのSPI受信エラー (0:エラーなし、1:エラー検出) bit 32 bool: ESP32のTeensyからのSPI受信エラー (0:エラーなし、1:エラー検出) 実装せず bit 31 bool: 未定義 bit 30 bool: ESP32のPCからのUDPフレームスキップエラー (0:エラーなし、1:エラー検出) bit 9 bool: TeensyのESP経由PCからのUDPフレームスキップエラー (0:エラーなし、1:エラー検出) bit 8 bool: PC受信のカウントのスキップ検出(hi0-7の数値を前回受信と比較) bit 0-7 bool: Teensy等の未端ボードでのサーボ信号受信エラー 0:エラーなし、1-99:メッセージ未定義、 100~255:エラーサーボID (0-14系[0-15], 15-29系[0-15])
Index[LEN-1]	CHECK SUM	short アドレスはデータ長さ-1, Index[LEN-2]までの数値を合計したものをビット反転したshort型

配列で使う コマンド (定義はすり合わせ つつアップデート)

送信時は
指示出し / 情報発信

データの
書き換え処理

Meridim 90 配列

**デバイス
B**

**1パケット以内の
一つのデータ形式が
デバイス間を
ぐるぐると循環**

受信時は
命令/情報の受信

受信時は
指示受け/情報受信

**デバイス
A**

Meridim 90 配列

データの
書き換え処理

送信時は
指示出し / 情報発信

**Meridim90という
データパッケージが決まった。**



**このデータをどうやって
ホビーロボット環境に循環させるか。**

ボードのプロトタイピング

UART高速通信に超強いTeensy4.0

600MHzの
ARM Cortex-M7プロセッサ



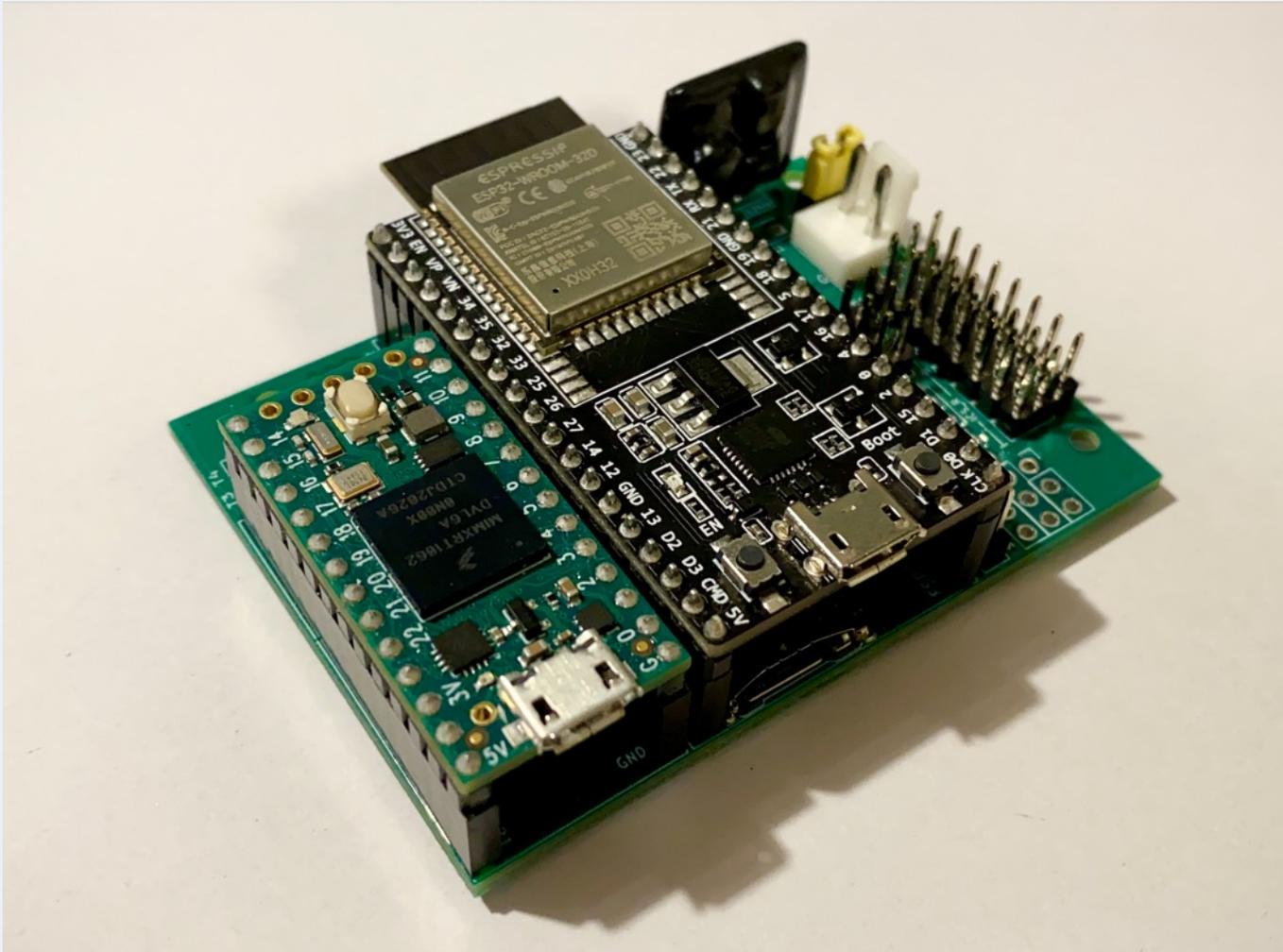
+



Wi-Fi 2.4GHz IEEE 802.11b/g/n
Bluetooth V4.2 BR/EDR BLE
デュアルコア 240MHZ CPU

無線通信に強いESP32-DevkitC

ボードのプロトタイピング



Meridian Board Type.K ver1.2

■ Teensy 4.0

600MHz,UART4ch

SPI通信とDMAで連携

■ ESP32 DevKitC

240MHz,Wifi,Bluetooth

サーボ自身の通信性能にもよるが、
5~10msで送受信可能。

データの 流れ

コマンドサーボ

命令
実行

状態
データ
返信

センシング
データ

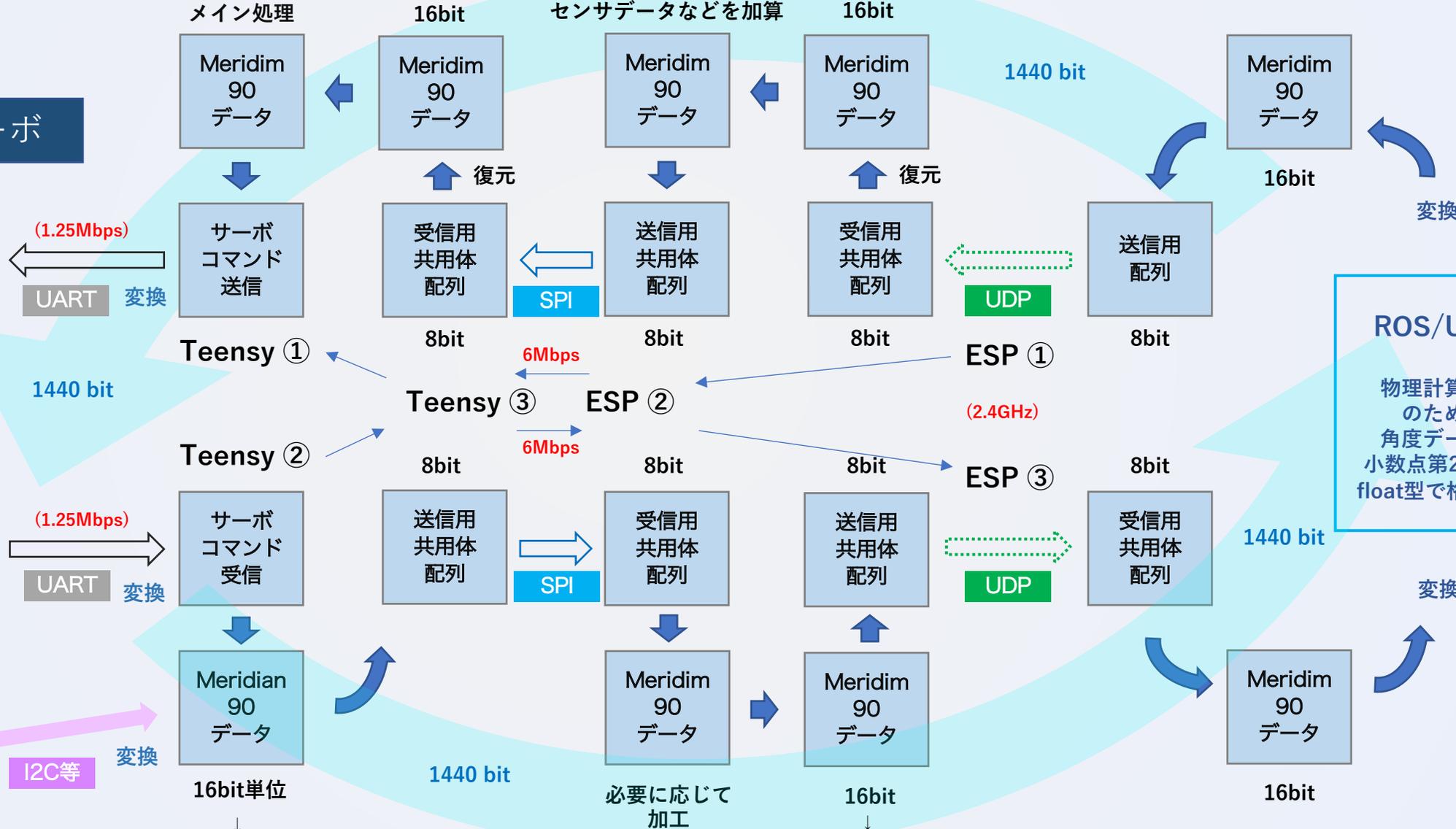
Teensy4.0

ESP32

PC/ROS/Unity

スタンドアロン時
計算処理や
モーションなどの
メイン処理

必要に応じて
リモコンデータや
センサデータなどを加算



ROS/Unity
物理計算処理
のために
角度データは
小数点第2までの
float型で格納保持

運用イメージ

Teensy4.0

ESP32

PC/ROS/Unity

コマンドサーボ

命令
実行

(1.25Mbps)
UART 変換

サーボ
コマンド
送信

受信用
共用体
配列

SPI

送信用
共用体
配列

UDP

送信用
共用体
配列

受信用
共用体
配列

競技などでは
PCとロボットを
切り離し
スタンドアロン
で運用

開発段階では
PCとシミュレータ
による環境を利用し
実機検証は
リモートブレイン

ROS/Unity

物理計算処理
のために
角度データは
小数点第2までの
float型で格納保持

状態
データ
返信

(1.25Mbps)
UART 変換

サーボ
コマンド
受信

送信用
共用体
配列

SPI

受信用
共用体
配列

UDP

受信用
共用体
配列

1440 bit

変換

センシング
データ

I2C等

変換

16bit単位

1440 bit

必要に応じて
加工

16bit

Meridim
90
データ

16bit

変換

(Teensy - PC Serial Monitor)

(ESP32 - PC Serial Monitor)

Meridian 2021.12.11

**ROSと
Meridianの
関係性**

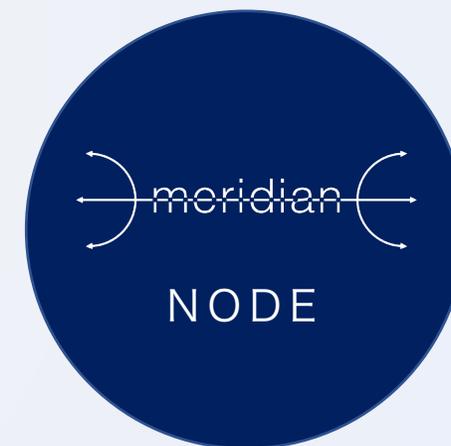
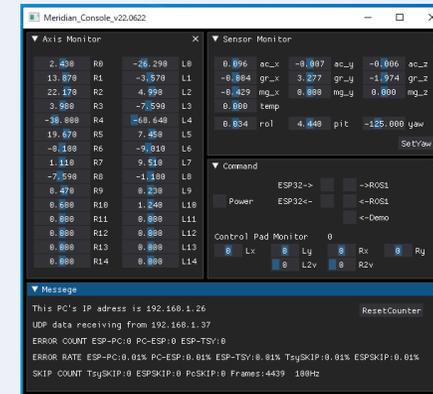
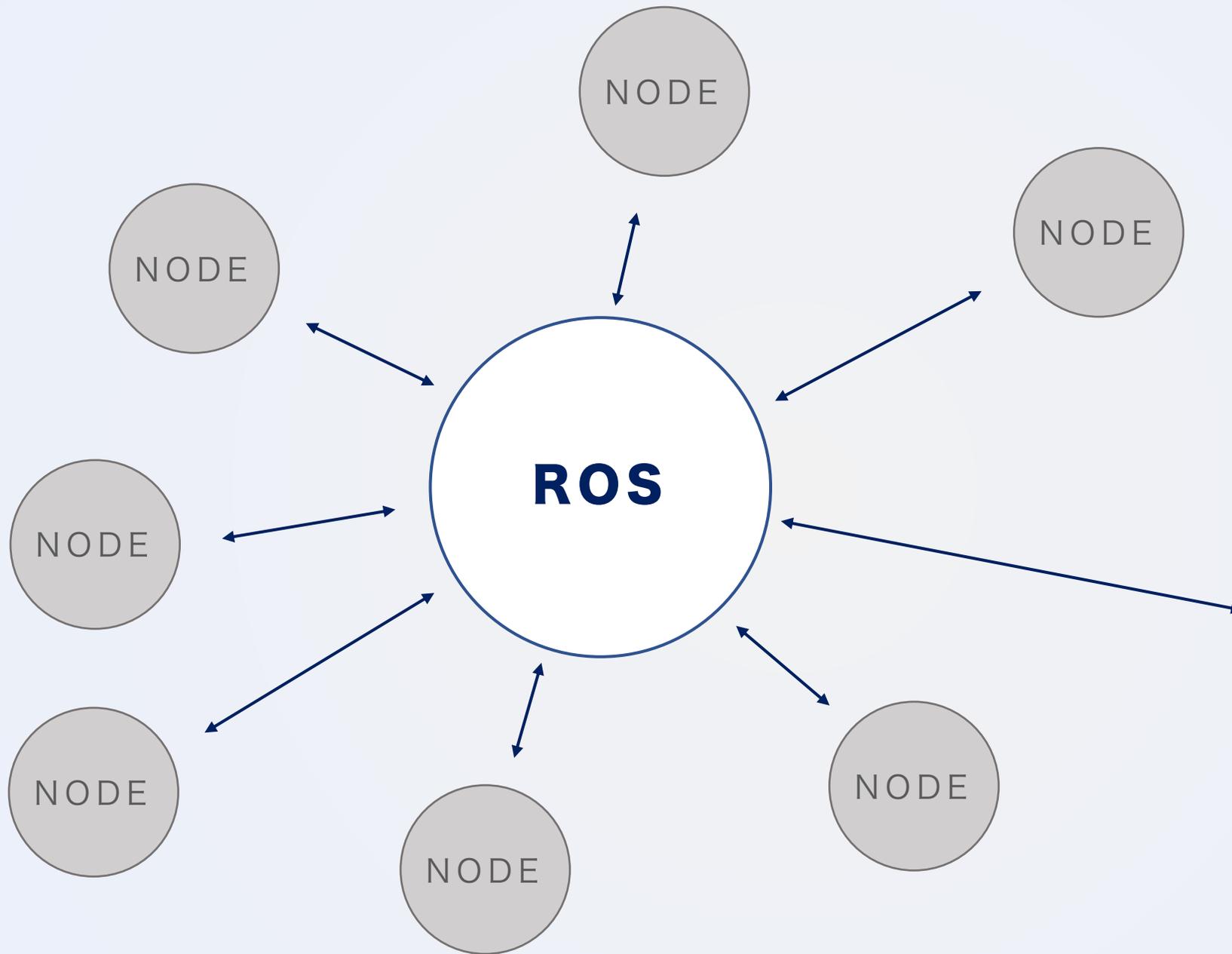
Meridian Console

The screenshot displays the Meridian Console v22.0622 interface, which is divided into several functional panels:

- Axis Monitor:** A table showing 15 rows of data for axes R0 through R14, with corresponding L0 through L14 values.
- Sensor Monitor:** Displays real-time sensor data including acceleration (ac_x, ac_y, ac_z), gravity (gr_x, gr_y, gr_z), magnetometer (mg_x, mg_y, mg_z), temperature (temp), roll (rol), pitch (pit), and yaw (yaw).
- Command:** A control panel with buttons for 'Power', 'ESP32->', '<-ROS1', '<-Demo', and 'Control Pad Monitor'. It also features a 'Control Pad Monitor' section with buttons for Lx, Ly, Rx, Ry, L2v, and R2v.
- Message:** A log window showing system status, including IP address (192.168.1.26), UDP data reception (192.168.1.37), error counts, error rates, and skip counts. It also includes a 'ResetCounter' button.

主な機能：

- 配列データの可視化
- コマンドの入出力
- **ROSの出版購読**



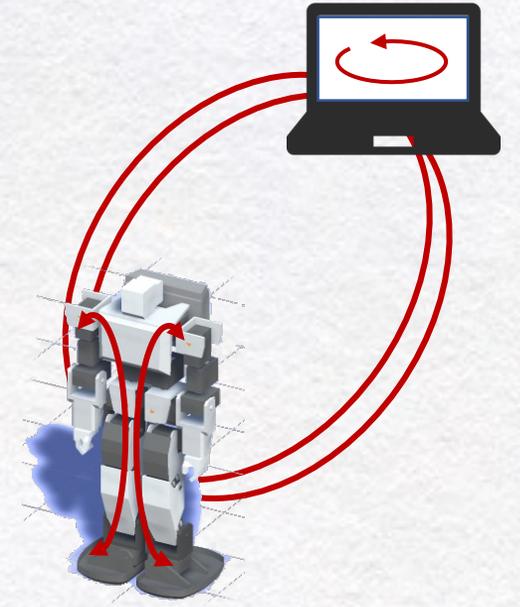
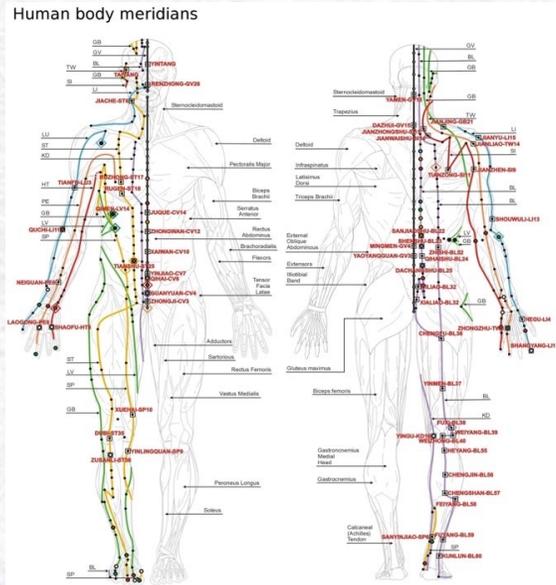
**ROSから見ると
Meridianは
ひとつのノード見える**

由来

メリディアンとは「**経絡**」のこと。

古代中国の医学において

気や血液の通り道。



仮説

ヒューマノイドにあてはめると、
情報信号の流れが経絡に該当すると捉える。

この流れを整え、

ロボットの開発・学習環境を健全に保つ。



- LITE -

PCでモーション計算 → UDP通信(100Hz) → 実機で再生



- ✓ **180byteのコンパクトなデータ配列**
- ✓ **1/100秒単位でロボットの状態情報を無線同期**
- ✓ **ROSの環境ではNODEとしてふるまう**
- ✓ **基本的になんにでもつながる**
- ✓ **オープンソースで運用が柔軟**

協力者紹介

ありがとうございます



二名川
@Ninagawa123

実機テスト フィジビリ検証/応用開発



holypong
@automo_emo



みっちー@メタバース・ロボッ...
@Dream_Drive



ぷくたい (音楽記号のロボット)
@RC_pukupukutaiy



キルロボ
@kirurobo



山口辰久
@qzy13700



カナ
@1573hsvydvh



PONDA
@PONDA_make



せんこー
@senkou_777

Twitter等での アドバイスやヒント



FrostyDesign
@FrostyDesign_JP



梨甘 🍌🍇🐟🥕🌱
@nashi_37



でべ
@devemin



Zak@Chiba
@zaklab



魔女みならい
@witch_kazumin



Hideaki Tai
@hideakitai_



勇者ロボ建造Vtuber
華姉(かあね)りあん/ロボット建造師石...
@BRLab



まさゆき 🇯🇵 愛知
@mayusaki3



realteck bot
@realteck_bot

※ 他にも**プリメイドハックのみなさん**の公開情報なども参考にさせていただいています。

Meridian NEWS

～ Meridian計画の進捗報告～



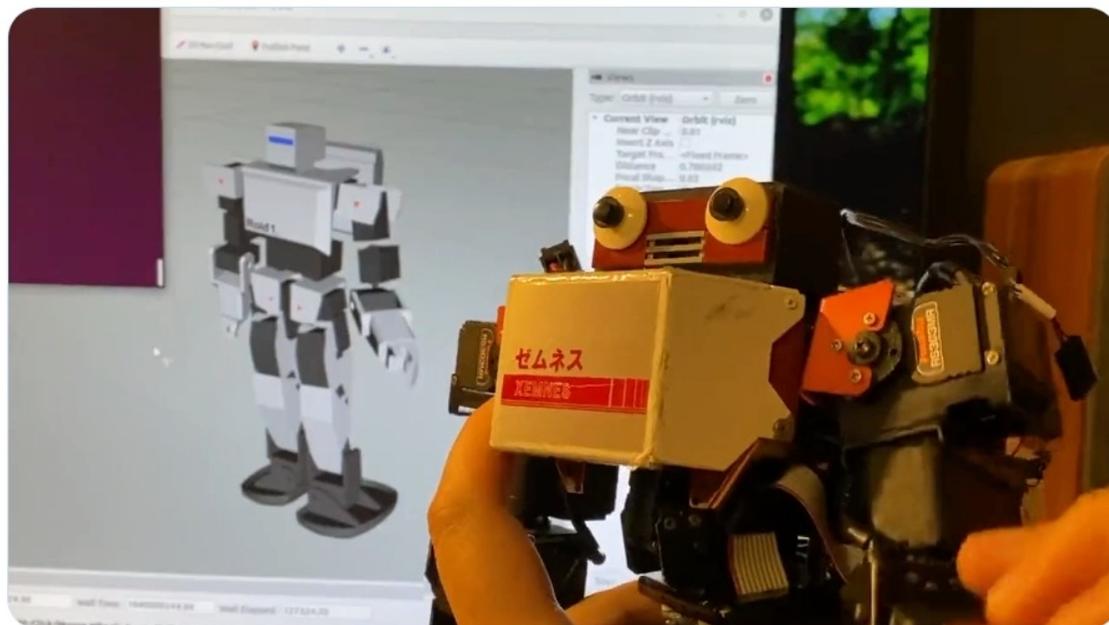


二名川

@Ninagawa123

ロボットメイカーにながわ。

各社サーボへの対応



FUTABAサーボ(RS30xシリーズ)

**FUTABAのTTLコマンドサーボを
Meridianに翻訳し、ROSとRvizで表示。**

参考URL



みっちー@メタバース・ロボッ...

@Dream_Drive

<https://dream-drive.net/2017/09/24/6557/>

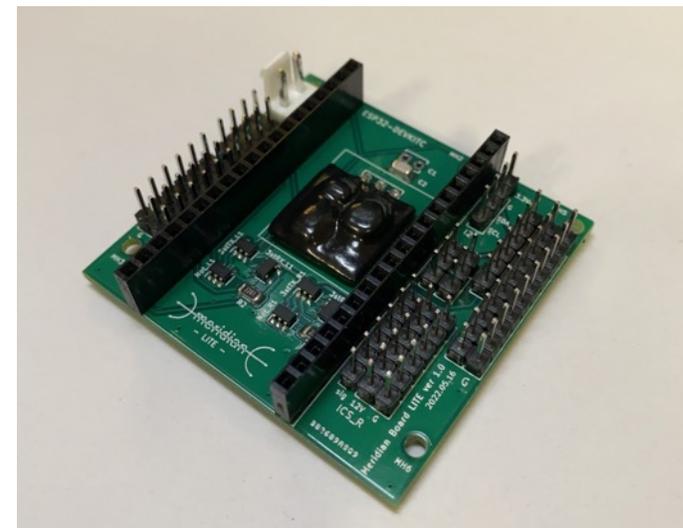
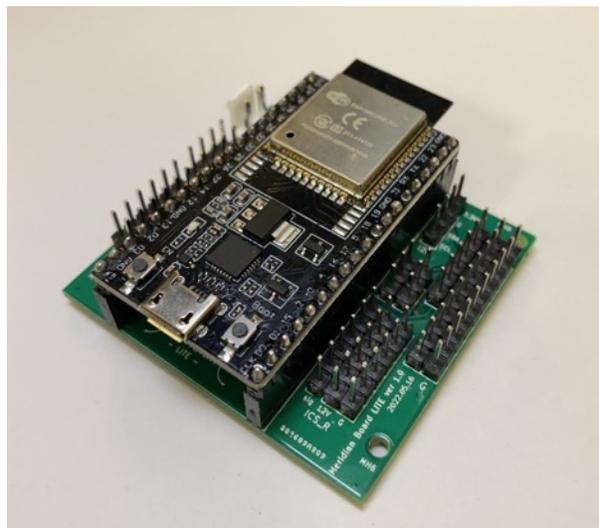
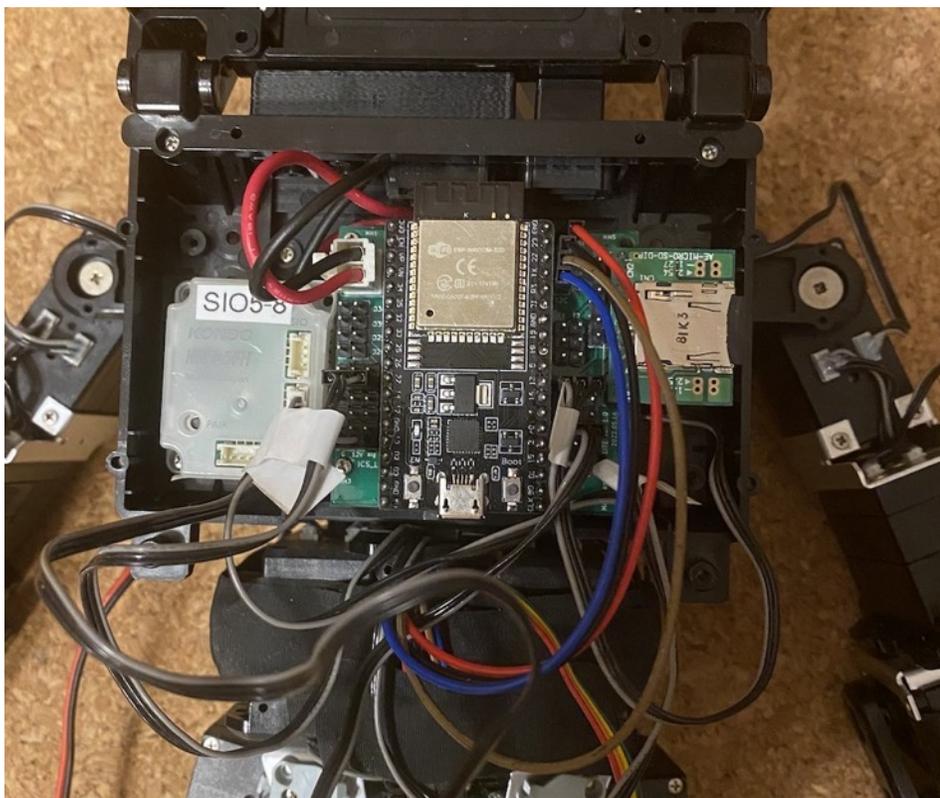


二名川

@Ninagawa123

ロボットメイカーにながわ。

対応ボードの開発



ESP32 1枚で動く

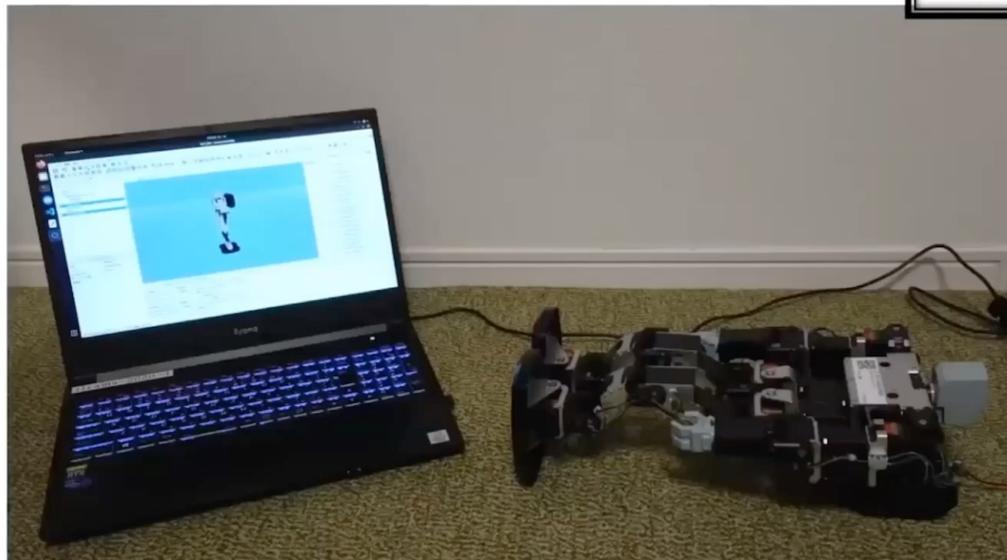
Meridian Board - LITE - が完成

- ✓ 22軸を100Hzで送受信（ただし一部データ補正あり）
- ✓ リモコンはKRC-5FH利用想定
- ✓ 9軸センサBNO055接続可能



ChoreonoidからMeridianへの通信

14



**Fusion360でCADデータをURDF化し
meridianとchoreonoid をROS経由で連携**

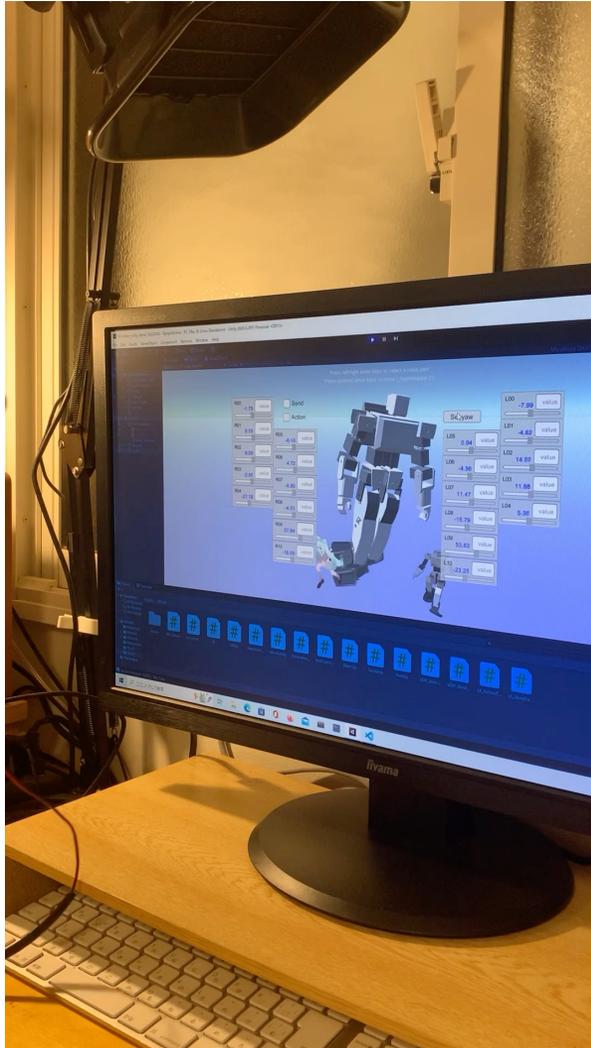


キルロボ
@kirurobo



二名川
@Ninagawa123
ロボットメイカーにながわ。

CG用モーションデータの活用



CGアバター表示用のモーションファイル



PC上でロボットの関節に変換



結果をUDP(100Hz)で送信



ロボット実機で再生



holypong
@automo_emo

公開技術の移植



Guero
@doctorguero

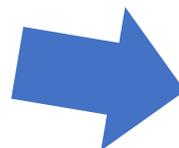
新しい動画編集ソフトを手に入れたので練習も兼ねて最近実験したロボットの解説動画を作ってみました。



youtube.com

Biped robot that does not fall.

An overview of balance control technology (upper body vertical control) that applies postural reflexes and the hardware configuration of the robot ...



公開されている
研究成果を
スムーズに
移植・再現



holypong
@automo_emo

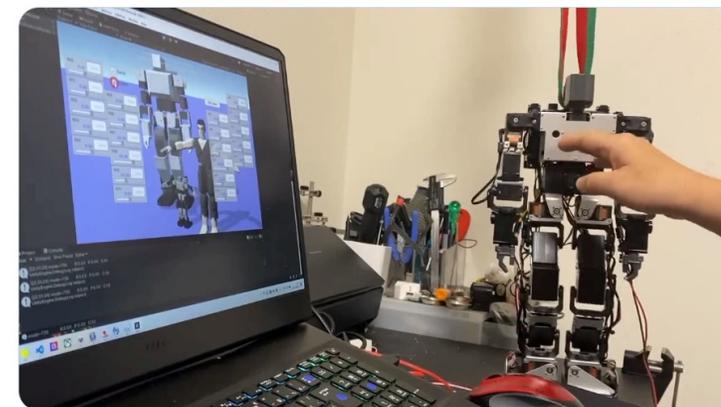
UVC（上体垂直制御）アルゴリズムをPC上のUnityに移植

デジタルツイン化により、リアルロボットを叩いたときのIMU値をPCがみて転倒回避処理をバーチャルロボットに実行、動きがリアルロボットにトレースされ転倒回避が実現する

通信周期はわずか0.01秒に過ぎない

「蒸着」より早い

#Meridian計画





みっちー@メタバース・ロボッ...
@Dream_Drive

VR空間とリアル空間の連携



ロボット実機の動き



Meridian



(MIDI信号化)



仮想空間

VRChatサーバー上の
ワールドに反映

VISION

誰もが二足歩行ロボットの研究を
楽しめるやさしい社会の実現

**同じような所を掘ってる方
ごいっしょにいかがですか？**

**先行研究を進められている方、
ぜひお知恵をお貸しくください。**

#Meridian計画

 **note**



二名川
@Ninagawa123

ご静聴ありがとうございました。

<https://note.com/ninagawa123/n/nb768563591be>

<https://twitter.com/Ninagawa123>



Discordにも部室があります。