

ニューラルソフト有限公司

(公開資料)	企画書	検認		照査	作成
	計画・設計書(草案) 検査計画・要領書 検査成績書	-	-	-	市来 博記
表 題	リアルノイド オリジン 制御系ハードウェア開発計画書(草案)				
副 題	RealNoids-Origin Control Hardware Development Plan(Draft)				
キ ー ワ ー ド	リアルノイド/オリジン/制御/ハードウェア/ RealNoids/Origin/Control/Hardware				
参 照 / 添 付 資 料					
管 理 番 号	件 名	改 定 履 歴			
000-10-03-0000	ヒューマノイド研究開発	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z			

目次

1. はじめに	3
2. 用語と略語の定義	3
3. 概要	4
4. 基本方針	4
5. 構成	6
6. 機能と要件、及び開発項目	10
6.1 思考・感情統合制御器	10
6.2 視覚・聴覚統合制御器	10
6.3 運動統合制御器	10
6.4 節編成制御器	15
6.5 指編成制御器	16
6.6 関節制御器	17
6.7 指制御器	18
7. 体制	19
8. スケジュール	19
9. 予算	19
10. 問題点	20
11. おわりに	20

1. はじめに

本書は、ロボバイオ リアルノイド オリジン（等身大ヒューマノイド）の制御を実現するためのハードウェアの構成と機能、及び開発範囲を定義するものである。（駆動系に使用するモータ、モータ ドライバ、圧力センサ、温度センサについては、駆動系ハードウェア開発計画書に記載する。）

2. 用語と略語の定義

本書で使用する用語と略語を表 2-1 に示す。

表 2-1 用語と略語の一覧

用語・略語	説明
オリジン	ロボバイオ リアルノイド オリジン（等身大ヒューマノイド）

3. 概要

オリジンの制御系は、ヒューマノイドの思考・判断と運動を制御するための系であり、以下に分類される制御器とそれらを接続する通信網で構成される。

- 統合制御器
思考・判断や運動を統合制御する機能を有する。
- 節編成制御器
節編成単位で運動制御を行う機能を有する。
- 関節制御器
関節単位で運動制御を行う機能を有する。
- 複合節編成制御器
複数節編成単位で運動制御を行う機能を有する。
- 節編成軸制御器
節編成の全軸単位で運動制御を行う機能を有する。

オリジンの制御系の性能は、人と同等の判断能力と運動性能を実現する為に必要な速度と安定性が求められる。

4. 基本方針

オリジンの制御器の構成と機能を決定するための基本方針を示す。

- 運動制御の基本方式(モード)として、次の3種の方式を想定する。
 - 軸角度制御モード (緊急危険回避時の高速運動等が対象)
軸の目標角度と速度のみ指定する補間/補正制御なし制御。
 - 分節遅延補正制御モード (歩行、走行、物を掴む等の運動)
フィード フォワード + 遅延フィード バック制御 (詳細はソフトウェア開発計画書に記載)
 - 分節修正制御モード (演奏や舞等の運動が対象)
フィード フォワード + 停止 + 分節修正 + 当該分節からの再制御
分節(分割された運動経路)は直線のみを想定する。
- 制御器の種類は次の7種とする。
 - 思考・感情統合制御器
 - 視覚・聴覚統合制御器
 - 運動統合制御器 (運動統合制御器と節/指編成制御器を同期させるシンクロナイザーを含む)
 - 節編成制御器 (脊柱部、右/左腕部、右/左脚部)
 - 関節制御器 (腰、胸、首、肩、肘、手首、股、膝、足首、つま先)
 - 指編成制御器 (右/左手部：複合節編成制御器)
 - 指制御器 (1~5指部：節編成軸制御器)
- 制御器の主従関係は次の通りとする。
 - 思考・感情統合制御器 > 運動統合制御器 > 節/指編成制御器 > 関節/指制御器
 - └ > 視覚・聴覚統合制御器

- 各制御器は安価に入手可能なマイコンと周辺回路とする。(日本/米国/英国/豪州/台湾製を選定する。)
- 統合制御器は、下位制御器からの割り込み信号がアクティブになった時間(カウンタ)を 250 マイクロ秒の精度で記憶できる性能を有してを想定する。
- 運動統合制御器と節/指編成制御器は、4 ミリ秒周期に対向制御器間で運動制御に関するデータを送受信できる性能を有していることを想定する。
- 節/指編成制御器と関節/指制御器は、4 ミリ秒周期に対向制御器間で運動制御に関するデータを送受信できる性能を有していることを想定する。
- 関節制御器は、500 マイクロ秒周期に関節を構成する軸(最大 3 軸)の位置/速度/トルク制御を実行できる性能を有していることを想定する。
- 指制御器は、500 マイクロ秒周期に指を構成する軸(最大 4 軸)の位置/速度/トルク制御を実行できる性能を有していることを想定する。

5. 構成

オリジンの制御器、制御対象となるモータ、制御判断の基準となるセンサの構成を図 5-1 と図 5-2 に示す。(図中の制御器、モータ、センサの位置は、実装時のおおよその配置の位置を表す。)

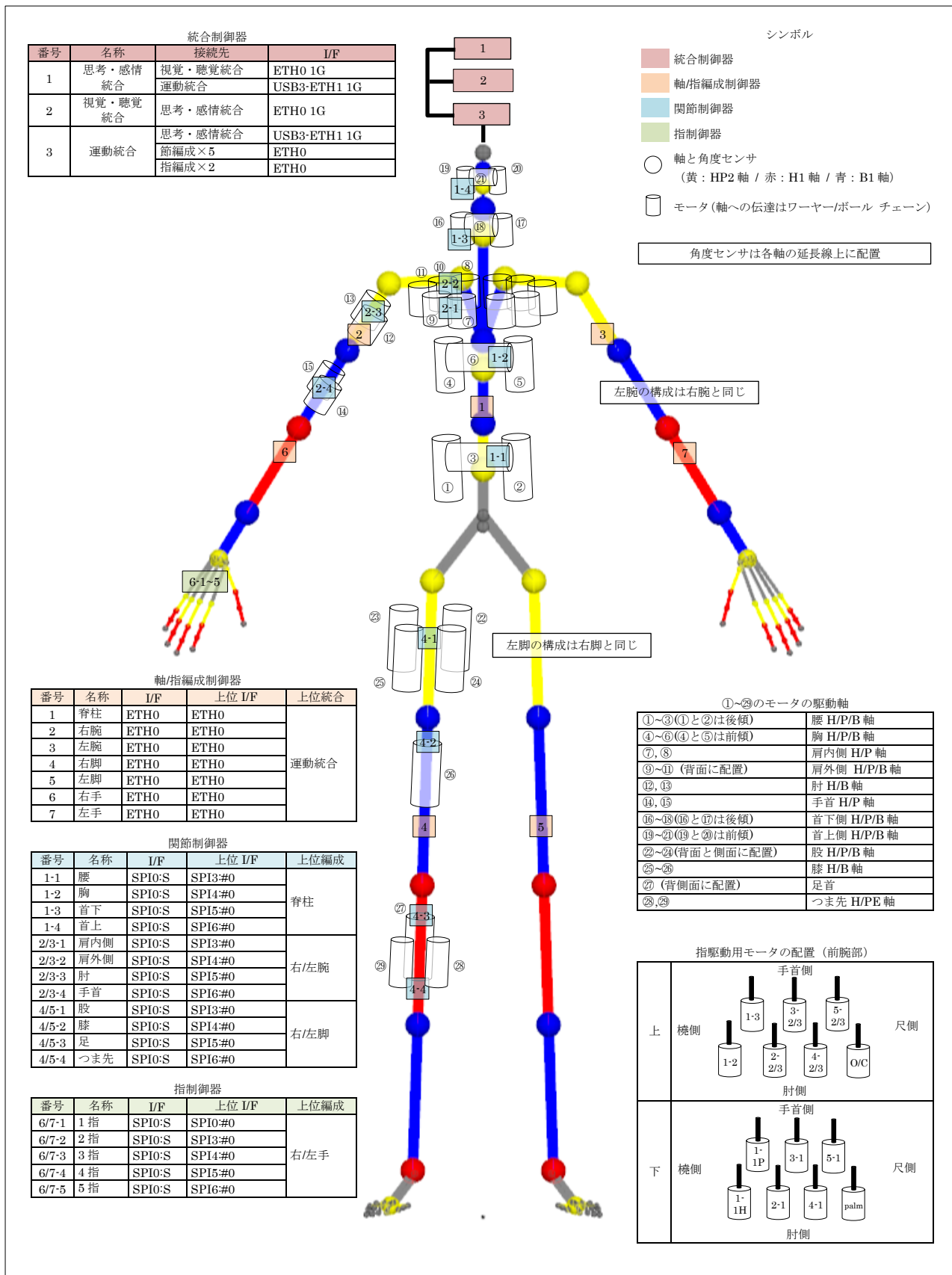


図 5-1 制御器/モータ/センサーの構成

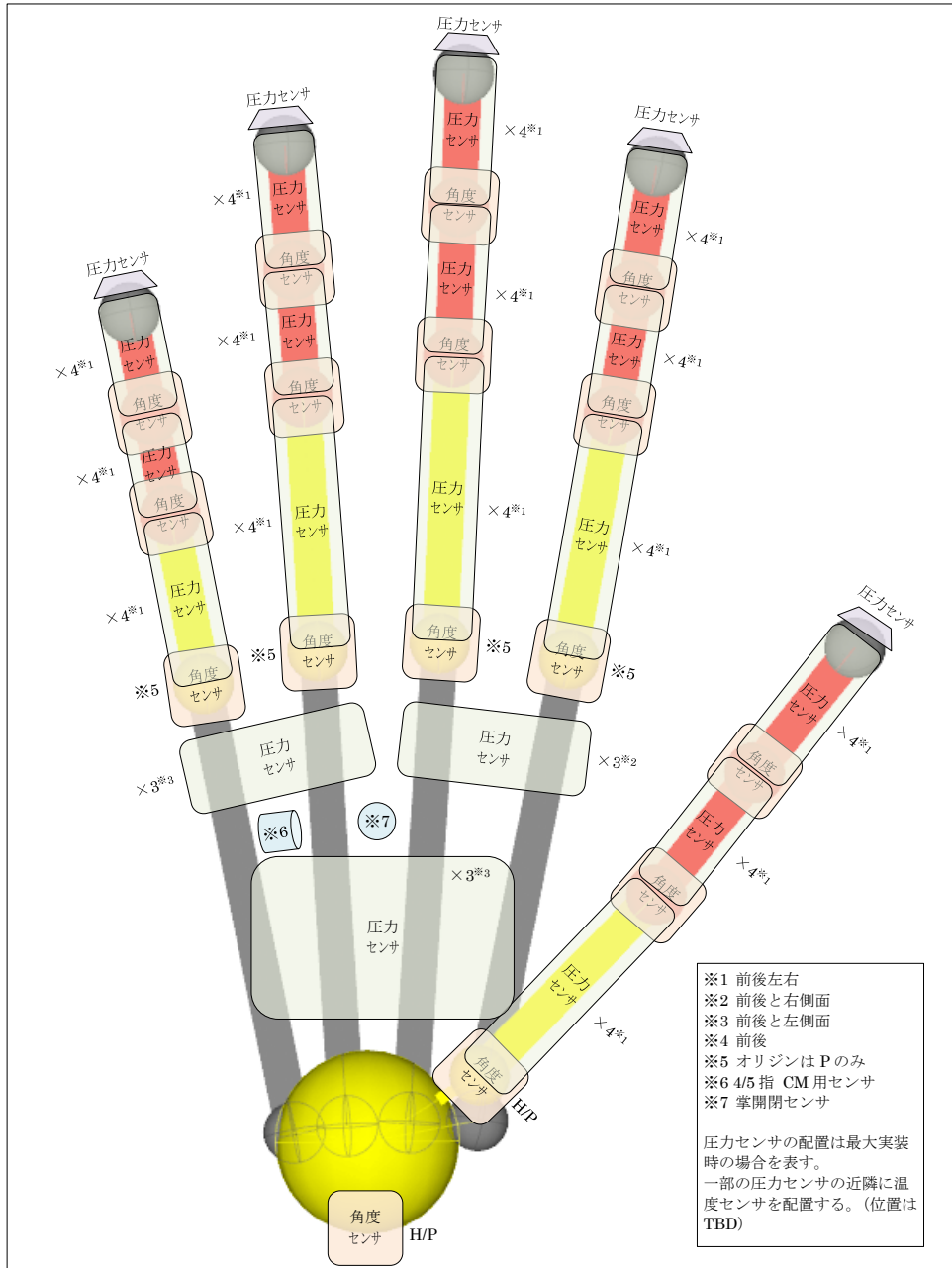


図 5-2 手のセンサーの構成

オリジンの制御器の接続図を図 5-3 に示す。

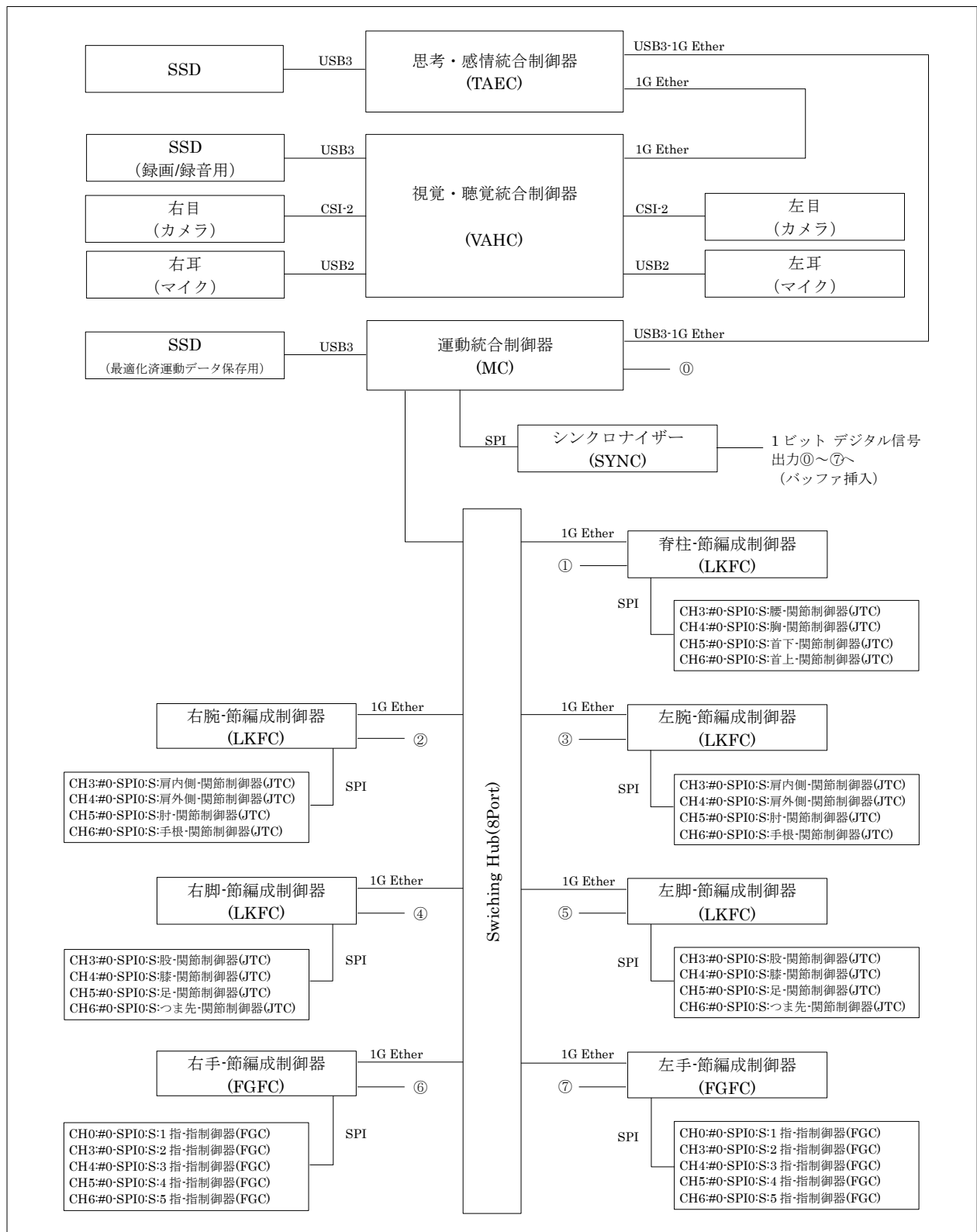


図 5-3 オリジン制御器の接続図

制御器接続の階層構造と通信方式、及び各制御器に使用するマイコンを図 5-4 に示す。

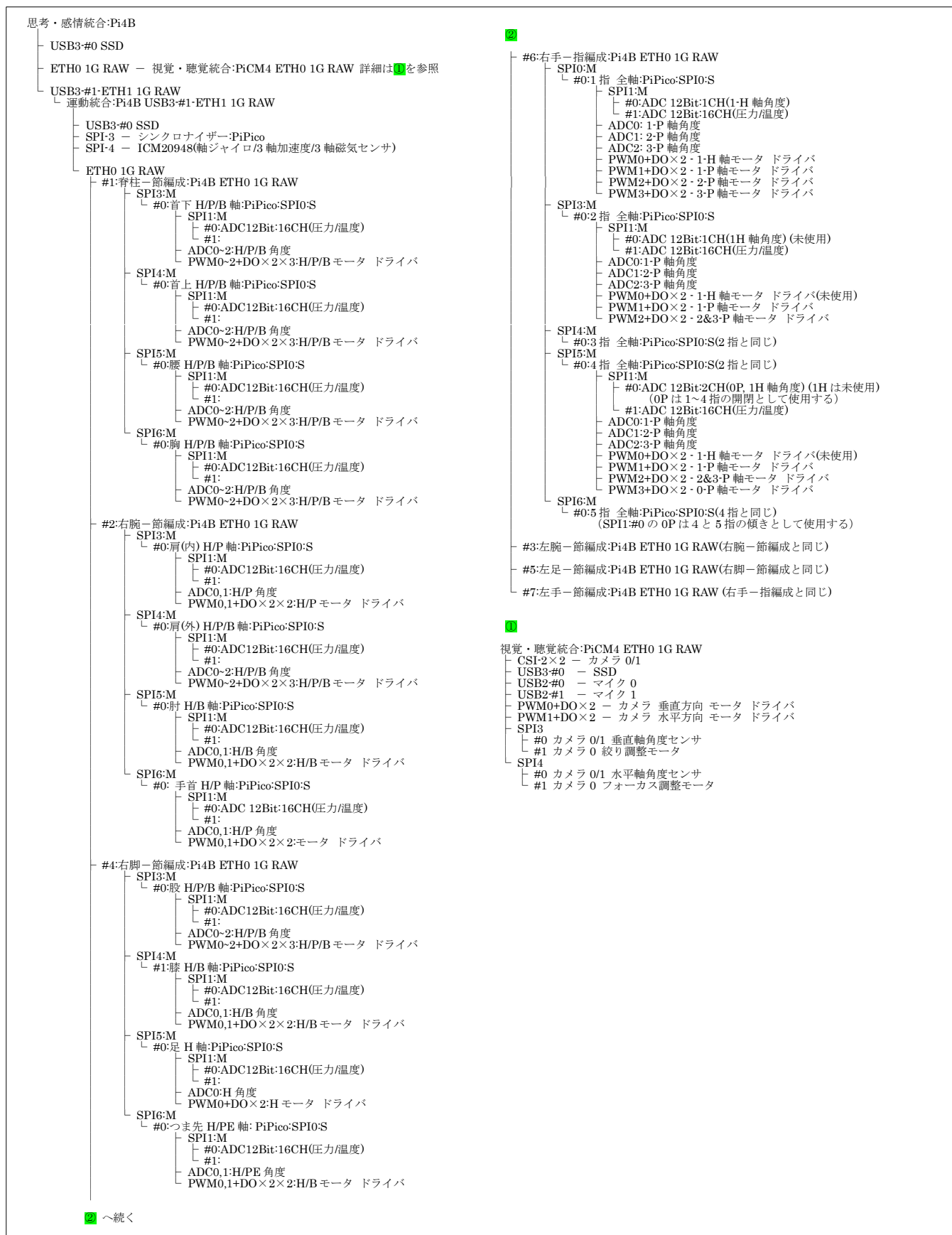


図 5-4 オリジンの制御器構成

6. 機能と要件、及び開発項目

次章以降に各制御器の機能と要件、及び開発項目を示す。

6.1 思考・感情統合制御器

思考・感情統合制御器は、自己の感情とボディの状態、及び外界からの情報から意志を決定し、目的を達成するまでの行程を分解して、思考と行動を進展させる機能を有する。本制御器の機能は、Raspberry Pi 4 Model B 上に実装される。(OS は Raspberry Pi OS 64Bit(Bullseye)を想定する。)但し、本制御器はオリジン プロジェクトの開発範囲外である為、開発作業は発生しない。

6.2 視覚・聴覚統合制御器

視覚・聴覚統合制御器は、ステレオ カメラからの画像情報とステレオ マイクからの音声情報から外界に存在する物体の位置、概形、動きを特定する機能と、音声情報から言語に関するデータを分離後、音素を配列データ化して、語/句/節(日本語ベースの原始言語)を特定する機能を有する。本制御器の機能は、Raspberry Pi Compute Module4 上に実装される。(OS は Raspberry Pi OS 64Bit(Bullseye)を想定する。)本制御器はオリジン プロジェクトの開発範囲外であるが、以下の項目の調査を行う。

(1) カメラからの画像入力

- (A) 入力形式
- (B) 画像データへのアクセス方法
- (C) 静止画生データ保存時の負荷
- (D) H264 形式入力データ保存時の負荷
- (E) 生データ入力&H264 ハードウェア エンコード保存時の負荷

(2) マイクからの音声入力

- (A) 入力形式
- (B) 音声データへのアクセス方法
- (C) 音声生データ保存時の負荷
- (D) 生データ入力&H264(AAC)エンコード保存時の負荷

6.3 運動統合制御器

運動統合制御器は思考・感情統合制御器、又は、無線通信によるリモート操作器からの指示により、全身の運動計画を生成して、その実行を制御(最適化と補正処理を含む)する機能を有する。本制御器の機能は、Raspberry Pi 4 Model B 上に実装される。(OS は Raspberry Pi OS 64Bit(Bullseye)を想定する。)

運動統合制御器の機能実現に対してハードウェアに求められる性能は、浮動小数点演算(浮動小数点⇒整数を含む)、及び有線/無線 LAN、SPI 通信の実効スループットである。本制御器の独自回路の開発作業は、ほぼ発生しないが、以下の項目の確認を行う。

(1) 1 編成 1 分節の目標姿勢決定に必要な処理時間(単精度/倍精度)

参考 約 10 年前の CPU i7-3930K 3.2GHz 搭載の PC で倍精度演算した場合、連続モードで 8 マイクロ秒、通常モード(初期姿勢からの演算)で 35 マイクロ秒程度。

(2) ETHER 1G(RAW)による通信の実効スループットと負荷

軸角度制御モードを想定し、図 6-1 の通信シーケンス実行時の周期精度と品質、及び CPU 負荷を確認する。(運動統合制御器の通信頻度と通信量は全てのモードで共通であることを想定している。)

運動統合制御器 - 節/指編成制御器間の ETHER 1G(RAW)による通信データ量の試算を表 6-1 に示す。

表 6-1 運動統合制御器 - 節/指編成制御器間の通信量の試算

節編成制御器	信号規格	方式	通信速度	プロトコル	トランザクション					
					分類	名称	データ量(バイト)		信号伝送時間(μs)	
							下り	上り		
脊柱	ETHER 1G-T	全二重	1Gbps	1Gbps	モード変更	通常/状態データ	8+64(2) ¹	8+14+180+4	1.648	
						休止/状態データ	8+64(2)	8+14+180+4	1.648	
						スリープ/状態データ	8+64(2)	8+14+180+4	1.648	
						システム更新/状態データ	8+64(2)	8+14+180+4	1.648	
						リセット/状態データ	8+64(2)	8+14+180+4	1.648	
					運動制御	運動データ/状態データ	8+14+100+4	8+14+180+4	1.648	
						節編成制御器 PG(0~)	8+14+1500+4	-	12.208	
						節編成制御器 PG Ack	-	8+14+1500+4	12.208	
						関節制御器 PG(0~)	8+14+1500+4	-	12.208	
						関節制御器 PG Ack	-	8+14+1500+4	12.208	
						システム更新	関節制御器書き込み	8+64(32)	-	0.576
							関節制御器書き込み Ack	-	8+64(32)	0.576
							関節制御器書き込み Stat	-	8+64(32)	0.576
							システム更新モード終了	8+64(32)	-	0.576
システム更新モード終了 Ack	-	8+64(32)	0.576							
右/左腕	脊柱と同じ									
右/左脚	#									
右/手	ETHER 1G-T	全二重	1Gbps	1Gbps	モード変更	脊柱のモード変更と同じ				
					運動制御	運動データ/状態データ	8+14+160+4	8+14+252+4	2.224	
					システム更新	脊柱のシステム更新と同じ(但し、関節制御器を指制御器に読み替えること。)				

運動統合制御器 - 節/指編成制御器間で全二重通信となる運動データと状態データの詳細(暫定)を表 6-2 ~表 6-5 に示す。

表 6-2 運動統合制御器 - 脊柱-節編成制御器間の通信データ

運動データ (下り方向)						状態データ (上り方向)						
分類	項目			ビットサイズ	バイトサイズ	分類	項目			ビットサイズ	バイトサイズ	
	軸角度	分節					種別	種別				
-				16	2	-				16	2	
腰関節	H	種別			16	2	H	種別		16	2	
		コマンド	コマンド	コマンド	8	8		位置		16	4	44
		制御速度	制御速度 0	制御トルク 0	8			速度		16	4	
		目標位置	制御速度 1	制御トルク 1	16			P	Hと同じ	32	4	
		予備	制御速度 2	制御トルク 2	16			B	Hと同じ	32	4	
		予備	制御速度 3	制御トルク 3	16			AD	16点×16Bit	256	32	
P	Hと同じ			64	8		H	腰関節 Hと同じ	32	4		
胸関節	B	Hと同じ			64	8	P	"	32	4	44	
		Hと同じ			64	8	B	"	32	4		
		H	腰関節 Hと同じ		64	8	AD	腰関節 ADと同じ	256	32		
首下関節	H	"			64	8	H	胸関節 Hと同じ	32	4	44	
		"			64	8	P	"	32	4		
		"			64	8	B	"	32	4		
首上関節	B	Hと同じ			64	8	AD	胸関節 ADと同じ	256	32		
		"			64	8	H	首下関節 Hと同じ	32	4	44	
		"			64	8	P	"	32	4		
"			64	8	B	"	32	4				
-		チェック サム		16	2	-		チェック サム	16	2		

*1 8 ⇒ プリアンプル / 64 ⇒ 最低パケットサイズ / 括弧内は実際のデータ サイズ

表 6-3 運動統合制御器 — 腕一節編成制御器間の通信データ

運動データ (下り方向)						
分類	項目			ビット サイズ	バイト サイズ	
	軸角度	分節				
—	種別			16	2	
肩(内) 関節	H	コマンド	コマンド	コマンド	8	8
		制御速度	制御速度 0	制御トルク 0	12	
		目標位置	制御速度 1	制御トルク 1	12	
		予備	制御速度 2	制御トルク 2	12	
		予備	制御速度 3	制御トルク 3	12	
		予備	制御速度 3	制御トルク 3	12	
肩(外) 関節	P	Hと同じ		64	8	
	B	未使用		64	8	
肘 関節	H	肩内側関節 Hと同じ		64	8	
	P	未使用		64	8	
手首 関節	H	肩内側関節 Hと同じ		64	8	
	P	未使用		64	8	
—	チェック	サム		16	2	

状態データ (上り方向)						
分類	項目			ビット サイズ	バイト サイズ	
	軸角度	分節				
—	種別			16	2	
肩(内) 関節	H	位置			16	4
		速度			16	
		Hと同じ			32	
		未使用			32	
肩(外) 関節	AD	16点×16Bit		256	32	
		H	肩(内)関節 Hと同じ		32	4
		P	"		32	4
		B	"		32	4
肘 関節	AD	肩(内)関節 ADと同じ		256	32	
		H	肩(内)関節 Hと同じ		32	4
		P	未使用		32	4
		B	肩(内)関節 Hと同じ		32	4
手首 関節	AD	肩(内)関節 ADと同じ		256	32	
		H	肩(内)関節 Hと同じ		32	4
		P	"		32	4
		B	未使用		32	4
—	チェック	サム		16	2	

表 6-4 運動統合制御器 — 脚一節編成制御器間の通信データ

運動データ (下り方向)						
分類	項目			ビット サイズ	バイト サイズ	
	軸角度	分節				
—	種別			16	2	
股 関節	H	コマンド	コマンド	コマンド	8	8
		制御速度	制御速度 0	制御トルク 0	12	
		目標位置	制御速度 1	制御トルク 1	12	
		予備	制御速度 2	制御トルク 2	12	
		予備	制御速度 3	制御トルク 3	12	
		予備	制御速度 3	制御トルク 3	12	
膝 関節	P	Hと同じ		64	8	
	B	未使用		64	8	
足 関節	H	肩内側関節 Hと同じ		64	8	
	P	未使用		64	8	
つま先 関節	H	肩内側関節 Hと同じ		64	8	
	PE	"		64	8	
—	チェック	サム		16	2	

状態データ (上り方向)						
分類	項目			ビット サイズ	バイト サイズ	
	軸角度	分節				
—	種別			16	2	
股 関節	H	位置			16	4
		速度			16	
		Hと同じ			32	
		未使用			32	
膝 関節	AD	16点×16Bit		256	32	
		H	股関節 Hと同じ		32	4
		P	未使用		32	4
		B	股関節 Hと同じ		32	4
足 関節	AD	股関節 ADと同じ		192	24	
		H	股関節 Hと同じ		32	4
		P	未使用		32	4
		B	未使用		32	4
つま先 関節	AD	股関節 ADと同じ		192	24	
		H	股関節 Hと同じ		32	4
		PE	"		32	4
		B	未使用		32	4
—	チェック	サム		16	2	

表 6-5 運動統合制御器 — 手一指編成制御器間の通信データ

運動データ (下り方向)							
分類	項目			ビット サイズ	バイト サイズ		
	軸角度	分節					
—	ヘッダ			16	2		
1 指	1H	コマンド	コマンド	コマンド	8	8	
		制御速度	制御速度 0	制御トルク 0	8		
		目標位置	制御速度 1	制御トルク 1	16		
		予備	制御速度 2	制御トルク 2	16		
		予備	制御速度 3	制御トルク 3	16		
	1P	1Hと同じ			64	8	
	2P	"			64	8	
	3P	"			64	8	
	2 指	1H	未使用			64	8
		1P	1 指の 1P と同じ			64	8
2/3P		1 指の 2P と同じ			64	8	
E		未使用			64	8	
3 指	1H	未使用			64	8	
	1P	1 指の 1P と同じ			64	8	
	2/3P	1 指の 2P と同じ			64	8	
4 指	E	未使用			64	8	
	1H	未使用			64	8	
	1P	1 指の 1P と同じ			64	8	
	2/3P	1 指の 2/3P と同じ			64	8	
5 指	E	1 指の 1P と同じ(4~5 指の開閉)			64	8	
	1H	未使用			64	8	
	1P	1 指の 1P と同じ			64	8	
	2/3P	1 指の 2/3P と同じ			64	8	
—	チェック サム			16	2		

状態データ (上り方向)						
分類	項目			ビット サイズ	バイト サイズ	
	軸角度	分節				
—	ヘッダ			16	2	
1 指	1H	位置		16	4	
		速度		16	4	
	1P	1Hと同じ			32	4
	2P	"			32	4
	3P	"			32	4
	AD	圧力×13		208	32	
		温度×3		48		
2 指	1H	未使用			32	4
	1P	1 指 1P と同じ			32	4
	2P	"			32	4
	3P	"			32	4
3 指	AD	1 指 AD と同じ			256	32
	2 指と同じ			384	48	
4 指	1H	未使用			32	4
	1P	1 指 1P と同じ			32	4
	2P	"			32	4
	3P	"			32	4
	E	"			32	4
AD	1 指 AD と同じ			256	32	
5 指	4 指と同じ			416	52	
—	チェック サム			16	2	

運動統合制御器—節/指編成制御器—関節/指制御器間の運動制御シーケンスの概要を図 6-1 に示す。

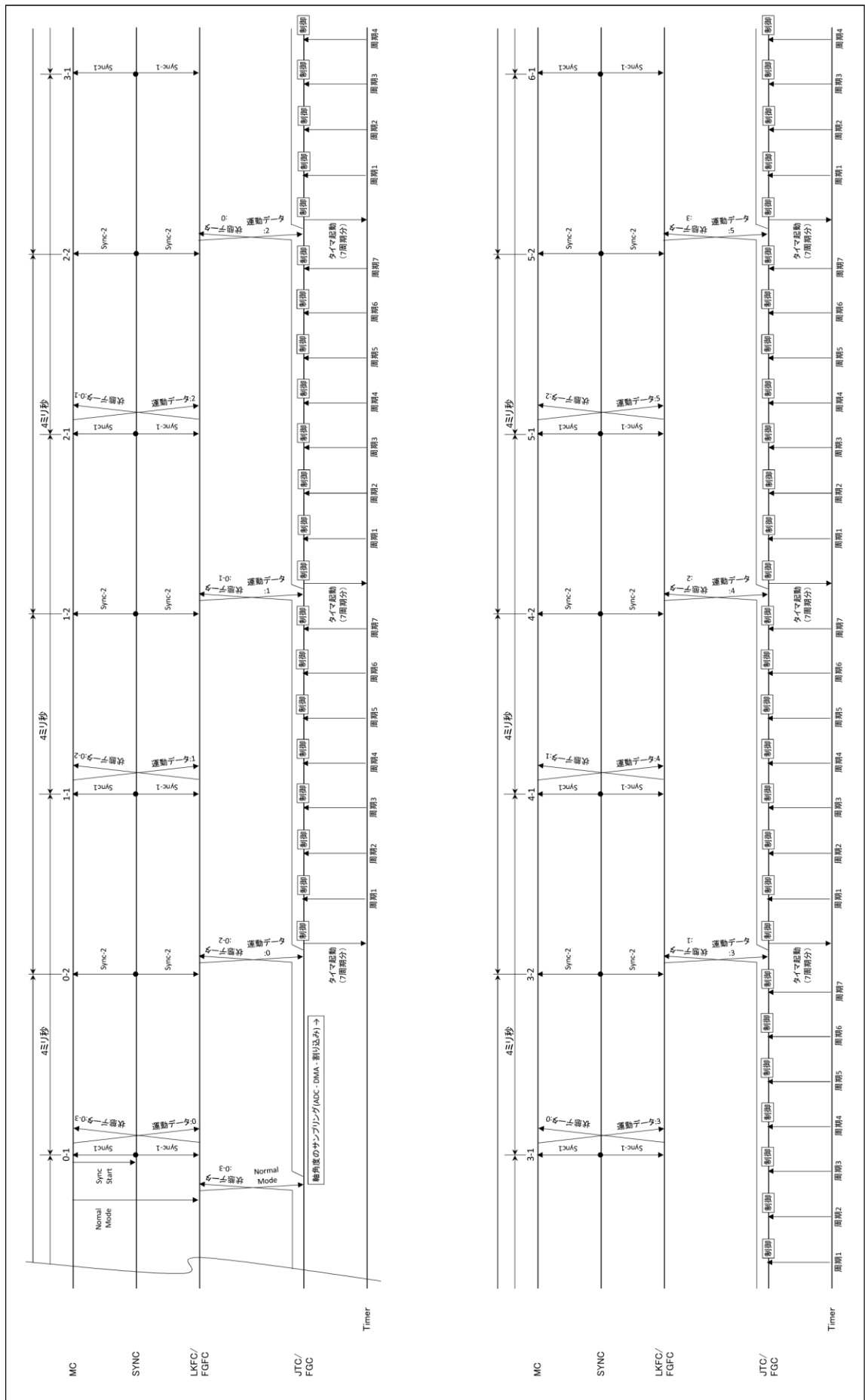


図 6-1 運動制御シーケンスの概要

6.4 節編成制御器

節編成制御器は、脊柱、腕、脚の部位単位の運動を制御する機能を有する。本制御器の機能は、Raspberry Pi 4 Model B 上に実装される。(OS は Raspberry Pi OS 32Bit(Bullseye)を想定する。)

(1) 機能概要

(A) 運動データ指示の分配

運動統合制御器から 4 ミリ秒周期で受信する運動データを配下の関節制御器(最大 4 台)へ分配する。

(B) 状態データの収集

配下の関節制御器(最大 4 台)から 4 ミリ秒周期で受信する状態データを運動統合制御器に送信する。

(2) 開発項目

節編成制御器の機能実現に対してハードウェアに求められる性能は、運動統合制御器間の有線 LAN と関節制御器(最大 4 台)間の SPI(3~6)通信の実効スループットである。本制御器の独自回路の開発作業は、ほぼ発生しないが、以下の項目の確認を行う。

(A) ETHER 1G(RAW)による通信の実効スループット

(B) SPI(3~6)の 1/2/4/8MHz の通信速度における最大ケーブル長 (テスト用接続ボード + CAT5e STP LAN ケーブルを使用)

(C) SPI(1MHz) + LTC6820 + 1.5m のケーブル(CAT5e STP LAN ケーブルを使用)における通信品質

(D) SPI(3~6)に各 1 台のテスト用関節制御器を接続した環境で、4 ポート同時送受信(ダミー データ)した場合の通信の品質

(E) (D)の環境における 4 ミリ秒周期の実行精度

(F) (D)の環境で、テスト用運動統合制御器から 4 ミリ秒周期に運動データ(ダミー データ)を受信した際の応答性能と、テスト用関節制御器間通信の品質と 4 ミリ秒周期の実行精度

(G) (F)の環境における CPU 負荷

節編成制御器 — 関節制御器間で全二重通信となる運動データと状態データのデータ量(暫定)を表 6-6 に示す。

表 6-6 節編成制御器 — 関節制御器間の通信量の試算

運動データ (下り方向)				
分類	項目	ビット サイズ	バイト サイズ	
—	種別	16	2	
軸 1	制御データ	64	8	24
軸 2	〃	64	8	
軸 3	〃	64	8	
—	チェック サム	16	2	
				28

状態データ (上り方向)				
分類	項目	ビット サイズ	バイト サイズ	
—	ヘッダ	16	2	
軸 1	位置/速度	32	4	44
軸 2	〃	32	4	
軸 3	〃	32	4	
AD	16 点×16 ビット	256	32	48
—	チェック サム	16	2	

6.5 指編成制御器

指編成制御器は、手(1~5 指)の運動を制御する機能を有する。本制御器の機能は、Raspberry Pi 4 Model B 上に実装される。(OS は Raspberry Pi OS 32Bit(Bullseye)を想定する。)

(1) 機能概要

(A) 運動データの分配

運動統合制御器から 4 ミリ秒周期で受信する運動データを配下の指制御器(5 台)へ分配する。

(B) 状態データの収集

配下の指制御器(5 台)から 4 ミリ秒周期で受信する状態データを運動統合制御器に送信する。

(2) 開発項目

指編成制御器の機能実現に対してハードウェアに求められる性能は、運動統合制御器間の有線 LAN と 5 台の関節制御器間の SPI(0,3~6)通信の実効スループットである。本制御器の回路の開発作業は、ほぼ発生しないが、以下の項目の確認を行う。

(A) SPI(0,3~6)に各 1 台のテスト用指制御器を接続した環境で、5 ポート同時送受信(ダミー データ)した場合の通信の品質

(B) (A)の環境における 4 ミリ秒周期の実行精度

(C) (A)の環境で、テスト用運動統合制御器から 4 ミリ秒周期に運動データ(ダミー データ)を受信した際の応答性能と、テスト用指制御器間通信の品質と 4 ミリ秒の周期の実行精度

(D) (C)の環境における CPU 負荷

指編成制御器 — 指制御器間で全二重通信となる運動データと状態データのデータ量(暫定)を表 6-7 に示す。

表 6-7 指編成制御器 — 指制御器(1 指~5 指)間の通信量の試算

運動データ (下り方向)				
分類	項目	ビット サイズ	バイト サイズ	
—	ヘッダ	16	2	36
軸 1	制御データ	64	8	
軸 2	〃	64	8	
軸 3	〃	64	8	
軸 4	〃	64	8	
—	チェック サム	16	2	

1~3 指の状態データ (上り方向)				
分類	項目	ビット サイズ	バイト サイズ	
—	ヘッダ	16	2	52
軸 1	位置/速度	32	4	
軸 2	〃	32	4	
軸 3	〃	32	4	
軸 4	〃	32	4	
AD	16 点×16 ビット	256	32	48
—	チェック サム	16	2	

4,5 指の状態データ (上り方向)				
分類	項目	ビット サイズ	バイト サイズ	
—	ヘッダ	16	2	56
軸 1	位置/速度	32	4	
軸 2	〃	32	4	
軸 3	〃	32	4	
軸 4	〃	32	4	
軸 5	〃	32	4	52
AD	16 点×16 ビット	256	32	
—	チェック サム	16	2	

(注) 運動データ中の

6.6 関節制御器

関節制御器は、関節単位の運動を制御する機能を有する。本制御器の機能は、Raspberry Pi Pico 上に実装される。

(1) 機能概要

(A) 運動データの受領と状態の通知

SPI スレーブとして 4 ミリ秒周期に節編成制御器から運動データを受信すると同時に、保持している最新の状態データを節編成制御器へ送信する。(運動データの受信前に必ずユニット開始の受信が発生する。運動データと状態データのデータ量については表 6-6 を参照のこと。)

(B) モータの制御

500 マイクロ秒周期に最新の運動データと状態データから設定値を算出して、1 軸当たり H/W PWM 信号(電圧又は電流制御用)×1 とデジタル信号(モード変更用)×2 をモータ ドライバに出力することで最大 3 軸分のモータの制御を行う。

(C) アナログ データの収集

(ア) MPU 内蔵 ADC による収集

MPU 内蔵の ADC を使用して、最大 3 軸分の角度情報 (8 ビット オーバー サンプリング→実効: 10~11 ビット) を収集する。

(イ) SPI I/F 経由の収集

SPI 接続の外部 ADC から 16 点のデータ (1 点当たり 12 ビット/500 マイクロ秒毎に 4 点読み出す) を収集する。

(2) 開発項目

(A) Raspberry Pi Pico 開発(評価)環境の構築

(B) SPI0 スレーブ受信の評価

(C) MPU 内蔵 ADC による軸角度読み取りの評価

(D) SPI1 接続の外部 ADC からのアナログ データ読み取りの評価

(E) PWM 信号とデジタル信号によるモータ ドライバ設定の評価

(F) 周期性確保の可否の確認

次の処理を実行して、節編成制御器への応答性能と、モータ ドライバへの設定周期(500 マイクロ秒)の精度と CPU 負荷を確認する。

(ア) MPU Core0 & DMA(SPI0&ADC)

- 節編成制御器からの運動データ受信と節編成制御器への状態データ送信 (4 ミリ秒周期)
- 3 軸分の角度サンプル データ転送 (ダブル バッファ) と LPF/間引き処理

(イ) MPU Core1 & DMA(SPI1)

- 最新の運動データと軸角度から設定値を算出してモータ ドライバに設定
- SPI 接続の外部 ADC からアナログ データを受信 (500 マイクロ秒毎に 4 点) と LPF 処理

(B)(D)の項目は、SPI クロック 1/2/4/8MHz で確認を行う。

(C)(D)(E)(F)の項目はテスト接続ボードを使用する。

(E)の項目はオシロを使用する。

6.7 指制御器

指制御器は、指単位の運動を制御する機能を有する。本制御器の機能は、Raspberry Pi Pico 上に実装される。

(1) 機能概要

(A) 運動データの受領と状態の通知

SPI スレーブとして 4 ミリ秒周期に指編成制御器から運動データを受信すると同時に、保持している最新の状態データを指編成制御器へ送信する。(運動データの受信前に必ずユニット開始の受信が発生する。運動データと状態データのデータ量については表 6-7 を参照のこと。)

(B) モータの制御

500 マイクロ秒周期に最新の運動データと状態データから設定値を算出して、1 軸当たり H/W PWM 信号(電圧又は電流制御用)×1 とデジタル信号(モード変更用)×2 をモータ ドライバに出力することで最大 4 軸分のモータの制御を行う。

(C) アナログ データの収集

(ア) MPU 内蔵 ADC による収集

MPU 内蔵の ADC を使用して、3 軸分の角度情報 (8 ビット オーバー サンプリング→実効 : 10~11 ビット) を収集する。

(イ) SPI I/F 経由の収集

SPI1:#0 接続の外部 ADC から 1 軸分の角度情報(12 ビット→実効 : 8~9 ビット)を収集する。

SPI1:#1 接続の外部 ADC から 16 点のデータ (1 点当たり 12 ビット/500 マイクロ秒毎に 4 点読み出す) を収集する。

(2) 開発項目

(A) SPI1:#0 接続の外部 ADC からの 1 軸分の角度読み取りの評価

(B) 周期性確保の可否の確認

次の処理を実行して、指編成制御器への応答性能とモータ ドライバへの設定周期の精度と CPU 負荷を確認する。

(ア) MPU Core0 & DMA(SPI0&ADC)

- 指編成制御器からの運動データ受信と指編成制御器への状態データ送信 (4 ミリ秒周期)
- 3 軸分の角度サンプル データ転送 (ダブル バッファ) と LPF/間引き処理

(イ) MPU Core1 & DMA(SPI1)

- SPI1:#0 接続の外部 ADC から 1 軸分の角度を取得
- 最新の運動データと軸角度から設定値を算出してモータ ドライバに設定
- SPI1:#1 接続の外部 ADC からアナログ データを受信 (500 マイクロ秒当たり 4 点)

(A)の項目は、SPI クロック 1/2/4/8MHz で確認を行う。

(B)の項目はテスト接続ボードを使用する。

7. 体制

制御系ハードウェアの開発はプロジェクト シリーズ責任者（一人）が行う。

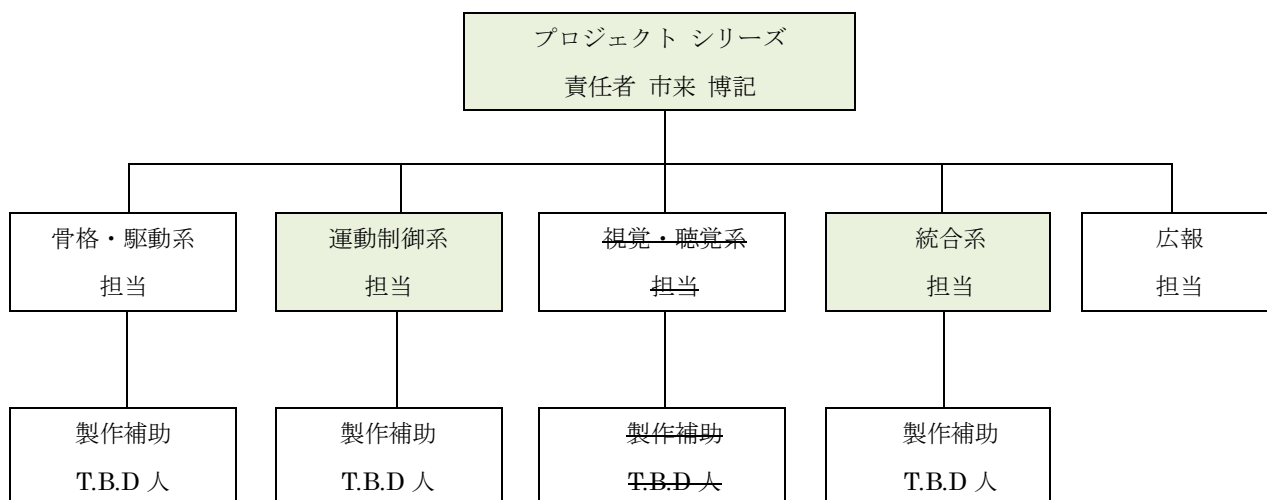


図 7-1 開発人員体制図

8. スケジュール

非公開

9. 予算

非公開

10. 問題点

現時点（2021年7月）で承知している問題点と対策を表 10-1 に示す。

表 10-1 制御系ハードウェア開発における問題点

区分	内容	対策
MPU ボード	Ethernet の伝送路の強制切替は可能か？	調査する。(切替時にネゴシエーションが実施されるのは NG)
	使用しない H/W 機能が多い。	不要な割り込み等が発生しないように無効に設定する。
通信	モータのノイズが通信に影響しないか？	シールド付きケーブルを採用して実験で確認する。
ツール	オシロがないと確認できない項目がある。	レンタル又は購入を検討する。

11. おわりに

経験不足の為、実験しなければ分からないことが多いが、一つひとつクリアにして確実に開発作業を進める。

ニューラルソフト有限公司

改定履歴	改 定 内 容	検 認	照 査	作 成
初期作成 21/7/5		—	—	市来 博記
B 21/12/25	文書ファイルのプロパティを設定した。	—	—	市来 博記
C 22/1/27	節編成制御器と指編成制御器の運動データ補正機能を削除した。(不要と思われる為) 関節制御器と指制御器の構成を変更した。	—	—	市来 博記
D 22/3/26	運動統合制御器配下にシンクロナイザーを導入した。 首の節編成と腰の節編成を結合した。(脊柱編成制御器とした。) 手の1指の3P軸を独立軸とした。	—	—	市来 博記
E 22/3/28	思考・感情統合制御器、視覚・聴覚統合制御器、運動統合制御器接続用イーサネット ハブを構成から削除した。 思考・感情統合制御器－視覚・聴覚統合制御器 ：内蔵 Ether 1G I/F を1対1接続 思考・感情統合制御器－運動統合制御器 ：USB3-Ether 1G アダプターを1対1接続 に変更した。 表 6-1 から表 6-5 の通信量試算の為の通信データ フォーマット(案)と、図 6-1 の運動制御用通信シーケンス(案)を変更した。	—	—	市来 博記
F 22/4/23	表 6-1 のモード変更の休止/状態データを削除した。	—	—	市来 博記
G 22/4/30	表紙のフォーマットを変更した。 管理番号を採番した。	—	—	市来 博記