

公立はこだて未来大学 2018 年度 システム情報科学実習  
グループ報告書

Future University-Hakodate 2018 System Information Science Practice  
Group Report

プロジェクト名

身体拡張筋電インタフェース -ASHURA-

**Project Name**

Body expansion interface using electromyogram -ASHURA-

グループ名

グループ A

**Group Name**

Group A

プロジェクト番号/Project No.

21-A

プロジェクトリーダー/Project Leader

1016088 豊見城 凧 Nagi Tomishiro

グループリーダー/Group Leader

1016113 米山裕樹 Hiroki Yoneyama

グループメンバー/Group Member

1016020 安部日菜子 Hinako Abe

1016079 岩崎香奈 Kana Iwasaki

1016086 鈴木智子 Tomoko Suzuki

1016088 豊見城 凧 Nagi Tomishiro

1016038 山本陽也 Hisaya Yamamoto

1016113 米山裕樹 Hiroki Yoneyama

指導教員

櫻沢 繁 安井 重哉 高木 清二 辻 義人

**Advisor**

Shigeru Sakurazawa Shigeya Yasui Seiji Takagi Yoshihito Tsuji

提出日

2019 年 1 月 16 日

**Date of Submission**

JANUARY 16, 2019

## 概要

義手とは、怪我や疾患によって上肢もしくは手首を失った患者が装着する人工の手である。その中でも、我々は表面筋電位によって動作制御される筋電義手に着目した。昨年度まで開発されてきた試装着用の筋電義手には、軽量化による握力の低下が問題点として残されていた。そこで我々は、握力や把持力の向上のために、義手表面の素材及び動力機構について改善する事にした。義手表面には柔らかい素材を使用することを決定した。理由は、人間の手に近づけるためである。今までの筋電義手だと、掴む際に義手と物との接触面積が極端に少なかった。しかし、人の手に近づけることで、接触面積を増やし、物を安定して持ち上げることができる。結果、把持力の向上と人間の手に近い感触を得ることができた。また、DC モータを遊星ギア付きステッピングモータに変えることにより義手自体の握力を上げることができた。

これらの改善の効果を評価するために、DC モータを遊星ギア付きステッピングモータに変えることにより義手自体の握力を上げることができた。これらの改善の効果を評価するために、前腕欠損者の方に協力してもらい、実際に義手を装着して使用していただき、使用感や機能面についての意見を伺った。実験の内容を以下に示す。

1. 異なる形の積み木とボールを箱の中に移動させるタスク
2. 箱をロープで結ばせるタスク
3. 未開封の 500g のペットボトルを持ち上げ蓋を開け、コップに注水するタスク

これら 3 つのタスクを行っていただき、それぞれの達成度を調べた。タスク 1 では、小さい物体や薄い物体は持ち上げにくいことが分かった。原因は、義手には感覚がないため見ながら位置を調整しなければならないためである。タスク 2 では、手のひらと親指の間でしっかりとロープを握っており、今年の義手に比べ物を縛りやすくなっていることが分かった。タスク 3 では、ペットボトルの容量が 500g までであれば、持ち上げられることができた。また、蓋をあけ、注水はできたものの、問題点が多くみられた。特に、ペットボトルを押さえる時に、指を曲げる順番や、義手表面の摩擦力の少なさなどの問題が挙げられた。以上の結果から、把持力や動作については改善され、また、昨年度の筋電義手よりも物を持ち上げやすくなっているという感想が得られたので、握力、把持力の向上は達成したといえる。新たな問題点としては、物を持ち上げる際に手首の旋回が必要なことや、ワイヤの固定方法などが出てきたため、義手本体以外にも見直しが必要である。

キーワード 筋電義手, 接触面積, 握力

(※文責: 豊見城凧)

# Abstract

Artificial hands are hands made by people worn by patients who have lost their upper limbs or wrists due to injury or disease. Among them, we focused on myoelectric prosthetic hands whose movement was controlled by the surface myoelectric potential. For myoelectric arms wearing trial wear that had been developed until the last fiscal year, reduction in grip strength due to weight saving was left as a problem. Therefore, in order to improve grip strength and gripping power, we decided to improve the material of the artificial surface and power mechanism. We decided to use a soft material on the surface of the artificial limb. The reason is to bring it closer to human hands. If it is a conventional myoelectric arms, the contact area between the artificial limb and the object was extremely small when gripping. However, by bringing it closer to the hand of a person, it is possible to increase the contact area and to stably lift the object. As a result, it was possible to improve the gripping force and obtain a feel close to a human hand. Also, by changing the DC motor to a stepping motor with planetary gear, we could raise the grip strength of the prosthesis itself.

In order to evaluate the effect of these improvements, we asked people with forearm deficits to cooperate, actually wear a myoelectric arms and use it, and hear opinions on feeling of use and function. The contents of the experiment are shown below.

1. Task to move different shaped blocks and balls into the box.
2. Task to tie a box with a rope.
3. Task to lift the unopened 500 g PET bottle, open the lid and pour water into the cup.

Three tasks were conducted, and the degree of achievement of each was examined. In task 1, small objects and thin objects were found to be difficult to lift. The reason is that the position has to be adjusted while watching as the artificial hand has no sense. In Task 2, we grasped the rope securely between the palm and the thumb, and we found that it was easier to tie the compare to last year's prosthesis. In task 3, if the content of PET bottle was up to 500 g, it could be lifted. Also, although the lid was opened and water injection was possible, many problem points were seen. Particularly, when pressing a plastic bottle, problems such as the order of bending the fingers and the less frictional force on the artificial surface. Based on the above results, it was found that the gripping force and motion were improved and the impression that it was easier to lift the object than the myoelectric muscle prosthesis of last year was obtained, so it can be said that the improvement of the grip strength and the gripping power were achieved. As a new problem, since it is necessary to turn the wrist when lifting the object and the method of fixing the wire, etc, it is necessary to review other besides the main body of the artificial arm.

**Keyword** The myoelectric arm, Contact area, Grip strength

(※ Responsibility for wording: Hinako Abe)

# 目次

第 1 章	はじめに	1
1.1	背景	1
1.2	目的	2
1.3	従来型の筋電義手の特徴	2
1.4	従来型の筋電義手の問題点	3
1.5	課題	3
第 2 章	プロジェクトの概要	4
2.1	問題の設定	4
2.2	課題の設定	5
2.3	到達レベル	5
2.4	課題の割り当て	6
2.5	計測機器	7
2.5.1	アクティブ電極	7
2.6	筋電位計測回路	7
2.6.1	差動増幅回路	8
2.6.2	ハイパスフィルタ	8
2.6.3	非反転増幅回路	9
2.6.4	半波整流回路	9
2.6.5	積分回路	9
2.7	制御回路	10
2.7.1	Arduino	10
2.7.2	L6470 使用ステッピングモータードライブレキット	10
2.7.3	ロータリエンコーダ	10
2.7.4	ポテンショメータ	10
2.7.5	制御プログラム	10
2.7.6	DC/DC コンバータ	11
2.7.7	3 端子レギュレータ	11
2.8	モータ	11
2.8.1	サーボモータ	11
2.8.2	ステッピングモータ	11
2.9	コーティング素材	12
2.9.1	Ecoflex	12
2.9.2	義体用シリコン	12
2.9.3	発泡ウレタン	12
2.10	外部部品	12
2.10.1	ソケット	12
2.10.2	アームライナー	13

2.10.3	3D プリンタ	13
2.10.4	レーザー加工機	13
2.10.5	Autodesk Fusion360	13
2.10.6	Autodesk Eagle	14
2.10.7	ワイヤの固定方法	14
<b>第 3 章</b>	<b>課題解決のプロセス</b>	<b>15</b>
3.1	前期プロトタイプ製作	15
3.1.1	従来型技術の問題点	15
3.1.2	考えた対応策	15
3.1.3	前期プロトタイプの仕様	16
3.2	前期プロトタイプの問題点	16
3.2.1	手袋に関する問題点	17
3.2.2	サーボモータに関する問題点	17
3.2.3	親指の曲がる角度に関する問題点	17
3.2.4	ワイヤの固定方法に関する問題点	17
3.3	後期プロトタイプの仕様	17
3.3.1	評価実験	18
3.3.2	被験者	18
3.4	各々の課題の概要とプロジェクト内における位置づけ	19
3.4.1	米山裕樹	19
3.4.2	豊見城凧	19
3.4.3	安部日菜子	20
3.4.4	岩崎香奈	21
3.4.5	鈴木智子	22
3.4.6	山本陽也	23
3.5	担当課題と他の課題の連携内容	24
3.5.1	米山裕樹	24
3.5.2	豊見城凧	25
3.5.3	安部日菜子	25
3.5.4	岩崎香奈	25
3.5.5	鈴木智子	25
3.5.6	山本陽也	26
<b>第 4 章</b>	<b>結果</b>	<b>27</b>
4.1	最終成果物	27
4.1.1	中間発表までの問題点	27
4.1.2	中間発表までの問題点の解決方針	27
4.1.3	最終成果物の仕様	28
4.2	評価実験	28
4.2.1	評価実験の概要と成果	28
4.2.2	評価実験における問題点	29

<b>第 5 章</b>	<b>まとめ</b>	<b>31</b>
5.1	前期終了時における進捗と問題点 . . . . .	31
5.2	後期における問題への対応 . . . . .	32
5.3	今後の展望 . . . . .	32
<b>第 6 章</b>	<b>評価フィードバック</b>	<b>34</b>
6.1	成果発表会 . . . . .	34
6.1.1	良かった点 . . . . .	34
6.1.2	悪かった点 . . . . .	34
6.2	まとめ . . . . .	35
<b>付録 A</b>	<b>課題解決のための新規獲得技術</b>	<b>36</b>
<b>付録 B</b>	<b>活用した講義</b>	<b>37</b>
<b>付録 C</b>	<b>制御プログラム</b>	<b>38</b>
<b>付録 D</b>	<b>制御プログラム</b>	<b>50</b>
<b>参考文献</b>		<b>57</b>

# 第 1 章 はじめに

## 1.1 背景

義手とは、怪我や疾患によって上肢もしくは手首を失った者が装着する人工の手である。一般的に、身体的外観を補うことを目的とした装飾義手、肩甲骨や肩関節などの動作を手の開閉に変換する能動義手など、機能や操作方法によって種類がある。また、能動義手の中には、筋肉の収縮時に生じる表面筋電位を利用して動作制御を行うものがある。これは筋電義手と呼ばれている。筋電義手は他の能動義手に比べ、物を掴み続ける力、すなわち把持力が強く、さらに軽量である。そのため、多様な場面で利用することができる。また、装着者の意思を即座に反映させた動作が可能である。

しかし、日本での筋電義手の普及は十分ではない。その理由として、以下の 2 点が挙げられる。第一に、筋電義手が高価であることである。第二に、筋電義手の購入に際して補助金支給に制限が設けられていることである。

厚生労働省は、国から補償金を受けられる条件を 2 つ定めている。第一に、両方の上肢を手関節以上で失った、もしくは片方の上肢を手関節以上で失うとともに、もう一方の上肢の機能をすべて失った、またはこれに準じた状態になったことにより、障害 (補償) 給付を受けた方、または受けると見込まれる者で、以下の 5 つの要件をすべて満たす者である [1]。

1. 手先装置の開閉操作に必要な強さの筋電信号を検出できる者
2. 筋電義手を使用するための判断力がある者
3. 筋電義手を使用するための十分な筋力がある者
4. ソケットの装着が可能な断端である者
5. 肩、肘の関節の機能に著しい障害がない者

第二に、片方の上肢を手関節以上で失ったことにより、障害 (補償) 給付を受けた者、または受けると見込まれる者で、以下の 3 つの要件をすべて満たす者である。

1. 就労中 (休職中を含む) の者で、筋電義手の装着により就労時の作業の質の向上や作業の種類拡大などが見込まれる、または申請時には就労していないが、筋電義手装着後に就労が予定されている者 (公共就業安定所への求職申込など就職活動中の者を含む) で、筋電義手の装着により就労時の作業の質の向上や作業の種類拡大などが見込まれる、または、非切断肢側の上肢、または手指に一定以上の障害があるため、筋電義手を使用しなければ社会生活ができないと認められる者
2. 筋電義手の装着訓練、試用着用期間における指導等、適合判定を実施する医療機関で、筋電義手の装着訓練を修了し、試用着用期間を経過している者
3. 上記で述べた第一の 5 つの要件をすべて満たす者で、筋電義手を継続して使用することが可能な者

上記の通り、補償金を受けるには要件を満たす必要がある。しかし、そのための訓練にかかる費用は補助の対象外である。補助を得ることができなかった場合、購入者は訓練にかかる費用に加え、義手や義手を付けるために必要なソケットなどの費用をすべて負担しなければならない。

さらに、義手は消耗品であり、耐用年数があるため、買い替えが必要となる。このように多大な費用がかかるなどリスクが高すぎるため、購入に踏み切ることができない問題がある。そこで、2016年度のプロジェクトでは、購入者のリスクを軽減すること、訓練前に筋電義手の使用感を試すことの2つを目的に、試装着用の筋電義手を製作した。成果として、実際に前腕を欠損されている方につけてもらい、動作実験を行うことができ、講評を得ることができた。さらに、この実験より、2つの問題点があがった。第一に、筋電義手本体が実際の重量よりも重く感じてしまったり、モータや基板などの部品が掌部分に集中してしまい、重心が先端に偏ってしまったという点である。第二に、物の形状によっては持てない場合や滑ってしまうという点である。2016年度のプロジェクト活動で挙がった問題の解決のために、2017年度のプロジェクトでは重量の原因であるモータなどを腰に移動し、手の先端にかかる負担を減らすために軽い筋電義手を開発することを目的としていた。しかし、軽量化を通して、以下の問題点が明らかになった。第一に、軽量化による義手の耐久性の低下である。第二に、軽量化したことで、モータの数が減り、握力が低下したことである。このことから、グループ A は物を掴む能力に注目し、筋電義手の握力、把持力を向上させる取り組みに着手した。

(※文責: 山本陽也)

## 1.2 目的

これまで述べたように、筋電義手は機能性に優れているが、本体が高価であり、購入のための補助金支給には制限が設けられている。これらを踏まえ、グループ A では、筋電義手の使用に興味を持つ人が、事前に筋電義手に触れる機会を設けることが必要だと考えた。そこで、試装着用の筋電義手を製作することで、筋電義手をより多くの人に使ってもらい、前腕欠損者の動作の幅を広げたいと考えた。その手段として、筋電義手の普及のために、試装着用として製作されてきた従来の筋電義手の性能向上を試みる。これまで、ASHURA プロジェクトでは、筋電義手の製作、および、筋電義手の性能を向上させる試みが行われてきた。2017年度の性能評価実験を通して、2017年度の筋電義手では、軽量化による耐久性と握力の低下が問題点として明らかになり、物を掴むという点に関して、十分な性能ではないことが考えられた。このことから、私たちは物を掴む能力に注目し、筋電義手の性能を向上させる取り組みに着手した。

(※文責: 岩崎香奈)

## 1.3 従来型の筋電義手の特徴

2016年度のプロジェクトでは、筋電義手に興味を持つ人が筋電義手の装着感や使用感を気軽に体験できるようにすることを目的とし、試装着に特化した筋電義手の開発が行われた。前腕欠損者に試装着を依頼した結果、以下の2点の課題が明らかになった。1つ目は装着した際に筋電義手本体が人の腕の重量よりもはるかに重く感じられること、2つ目は物の形状によっては持てない場合や滑ってしまう場合があるということである。それを踏まえ、2017年度のプロジェクトでは、筋電義手が重いという問題を解決することを目的とし、筋電義手を軽量化することで使用者の腕の負担が少ない筋電義手の開発が行われた。その結果、筋電義手の性能評価実験の、一定の高さで30秒間腕を維持するタスクにて、2016年度より軽量化されたとの回答が得られた。ペットボトルを

持ち上げるタスクでは、170g まで持ち上げられることが出来ていた。170g 以上のペットボトルを持ち上げられなかった原因として、素材を変更したことによる握力の低下が挙げられる。このことから、2017 年度の筋電義手には、軽量であるが、その一方で、素材を変更したことにより、握力が弱いという特徴が挙げられる。

(※文責: 鈴木智子)

## 1.4 従来型の筋電義手の問題点

筋電義手を軽量化した一方で、持つことに関する性能が低下した。2017 年度までの取り組みにより、実際に筋電義手を装着してもらうことと、筋電義手を軽量化することができている。問題点として、物の形状によっては持てない場合や滑ってしまう場合があることや、軽量化により握力が弱くなってしまふということが残されている。2017 年度に行われた評価実験のペットボトルを持ち上げるタスクを行った結果、170g のペットボトルまで持ち上げることができていた。握力が弱くなってしまった原因として、2016 年度よりも軽い素材である ABS 樹脂のフィラメントで筋電義手のパーツを製作したことと、ワイヤの張りや強度が弱かったこと、ワイヤと義手の間に摩擦力が生じることで、指の屈曲、伸展を妨げていたことが挙げられる。ワイヤは指の屈曲、伸展のために用いており、サーボモータから指先に通し、サーボモータの回転角度によって指の屈曲、伸展を制御している。ペットボトルからの注水のタスクでは、被験者および実験装置が濡れるのを防ぐために水の代わりに玩具用品の BB 弾を用いた。評価実験では、筋電義手の手首の部分にペットボトルの重さが集中し、意図しない手首の回転が発生した。そのため、本タスクは達成できなかった。このことから、軽量化することで持つことに関する性能が低下したといえる。

(※文責: 鈴木智子)

## 1.5 課題

2017 年度では筋電義手を軽量化したが、握力が低下する問題が発生した。1.4 節で挙げられた問題点のうち、握力の向上と、物の形状によっては滑ってしまう場合があることの解決を課題とした。これは、軽量化していない場合でも例年十分な握力が得られておらず、さらに、握力を向上させることで筋電義手の性能向上に大きく繋がると考えたためである。また、物を持ち上げやすくするために物の形状によっては滑ってしまう場合があることに関して、どんな形状の物でも滑らず持ち上げるために、滑りやすさの改善をすることとした。

(※文責: 鈴木智子)

## 第 2 章 プロジェクトの概要

### 2.1 問題の設定

1.4 節では，物の形状や材質によっては持てない場合や滑ってしまう場合があること，軽量化により耐久性と握力が弱くなってしまふことが従来の筋電義手の問題点として挙げられた．本プロジェクトでは，握りこむ力である握力や，物を握ったときに離さないようにする力である把持力が弱いことの改良を目指すこととした．形状や材質によって持てないことや滑ることについては，握力や把持力を向上させることで改良できると考えた．握力や把持力の弱さを問題に設定した理由として，以下の二点が挙げられる．第一に，2016 年度の評価実験での掴んだ物を特定の場所へ移動させるタスクにおいて，掴んだ物が滑り落ちてしまっていたためである．第二に，この評価実験において被験者から頂いた「モータをもっと強くして欲しい」という握力の強さに関する要望に対し，2017 年度のプロジェクトでは改良がされていなかったためである．

(※文責: 岩崎香奈)

## 2.2 課題の設定

2.1 節では、握力や把持力の弱さが問題として挙げられ、物の形状や材質によって持てない点や滑る点の改良を試みることにした。この問題を解決するために、以下の具体策が提案された。

- (1) 柔らかい素材を義手に取り付け、接触面積を増やすことで滑りにくくする。
- (2) ワイヤの張りを強くし、摩擦の小さいものに変更する。
- (3) モータをよりトルクの強いものにし、握力を向上させる。

(1) を課題として設定した理由は、2017 年度の筋電義手は表面が硬く滑りやすい素材 (図 2.1) でできていたため、人間の肉のように柔らかい素材を表面に取り付けることで、接触面積を増やすことができると考えたためである (図 2.2)。(2) を課題として設定した理由は、ワイヤと義手の間で摩擦力が生じることによって、モータの力がうまく伝わらず、握力が弱くなっていたためである。さらに、ワイヤが細すぎると義手の素材を削ってしまい、義手自体の耐久性が下がってしまう。このことを踏まえて、最適なワイヤを選び、張りを強くする構造を考える必要がある。(3) を課題として設定した理由は、モータのねじりのモーメントであるトルクを強くすることにより、ワイヤを引っ張る力を強くすることができるためである。



図 2.1 表面が硬い場合の接触面積

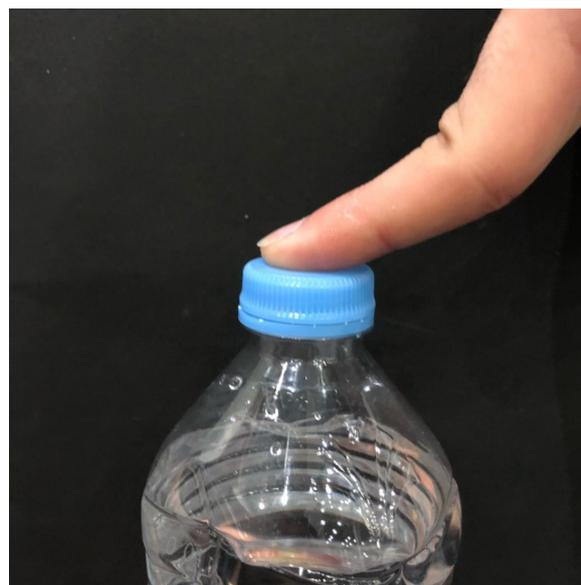


図 2.2 表面が柔らかい場合の接触面積

(※文責: 岩崎香奈)

## 2.3 到達レベル

本グループでは、筋電義手の使用に興味を持っている前腕欠損者が、筋電義手を装着することで行える動作の幅を広げることを目的とし、試装着用の筋電義手の性能向上に取り組んだ。そのために、握力を向上させ、物を持ちやすくすることを目標とした。目標設定に際して、以下のレベルを設定した (表 2.1)。100 点満点のうち、筋電義手として動作した場合は 60 点、動作や素材の安定性が確保されていた場合は 70 点、物を掴む際の滑りやすさが解消されていた場合は 80 点、170g

のペットボトルより重い物を持ち上げることができた場合は 90 点，十分な握力が得られた場合は 100 点とした．170g に設定した理由としては，2017 年度の筋電義手の性能評価実験で，170g までのペットボトルを持ち上げることができていたためである．100 点と評価する際の十分な握力とは，一般的なペットボトル飲料水の重さである 500g のペットボトルを持ち上げられる性能とする．

表 2.1 達成目標とその点数

達成目標	点数 (点)
筋電義手として動作した場合	60
動作や素材の安定性が確保されていた場合	70
物を掴む際の滑りやすさが解消されていた場合	80
170g のペットボトルより重い物を持ち上げることが出来ていた場合	90
十分な握力が得られた場合	100

(※文責: 岩崎香奈)

## 2.4 課題の割り当て

筋電義手を製作するにあたり，外装班と回路班の二つに分かれた．前期では，外装班は，3D プリンタを用いた義手の製作，義手に取り付ける柔らかい素材の製作を行った．回路班は，回路と電極の製作，サーボモータの制御を行った．また，サーボモータと義手を繋げて実際に筋電位を計測し，義手の動作を制御した．前期の各課題の割り当ては以下の通りである．

- 外装班は岩崎，鈴木，山本，回路班は米山，豊見城，安部に分かれた．

### <外装班>

- 3D プリンタを用いて義手の部品の出力，組み立て．
- 義手表面に取り付けるシリコン樹脂やウレタン樹脂を選定するための人差し指を 4 本製作．
- 2 種類のシリコン樹脂と 1 種類のウレタン樹脂を人差し指に取り付ける．
- 義手全体に選定したシリコン樹脂の取り付け．
- 3D プリンタでサーボモータを入れるための箱の部品の出力，行い組み立て．
- ワイヤのカバーの調節．

### <回路班>

- 筋電位計測回路についての学習．
- ユニバーサル基板で筋電位計測回路を製作．
- 筋電位計測回路から出た信号を Arduino で受け取り，サーボモータの PWM 制御．
- 電極の製作．

後期では中間発表の評価をもとに外装班は，3D プリンタを用いた前期よりも一回り小さい義手の製作，義手に取り付ける柔らかい素材の制作を行った．回路班は，筋電位計測回路を銅板で製作，ステッピングモータの制御，サーボモータの制御を行った．また，ステッピングモータとサー

## Body expansion interface using electromyogram -ASHURA-

ボモータを義手につなげて、実際に筋電位を計測し、義手の動作を制御した。後期の各課題の割り当ては以下のとおりである。

- 外装班と回路班は前期と同様に分かれた。

### <外装班>

- 3D プリンタを用いて義手の部品の出力を行い組み立て。
- 薄くした発泡ウレタンを義手の指全体と手のひらに張り付け。
- 張り付けた発泡ウレタンの上に義体用シリコンを手のひら側に張り付ける。
- ソケットと義手を繋ぐ手首を 3D プリンタで出力。
- ステッピングモータに取り付けるボビンを 3D プリンタで出力。
- レーザー加工機でステッピングモータを入れるための箱を製作。
- ワイヤのカバーの調節を行った。

### <回路班>

- 銅版で筋電位計測回路を製作。
- 筋電位計測回路から出た信号を Arduino で受け取り、サーボモータの PWM 制御。
- ステッピングモータの調節。
- 電極の製作。

(※文責: 安部日菜子)

## 2.5 計測機器

### 2.5.1 アクティブ電極

筋電位を計測する際に、グループで製作したアクティブ電極を使用した。アクティブ電極を使うことで、電極側の入力の抵抗を皮膚の抵抗より十分に大きくすることにより、ノイズを小さくすることが可能である。

(※文責: 安部日菜子)



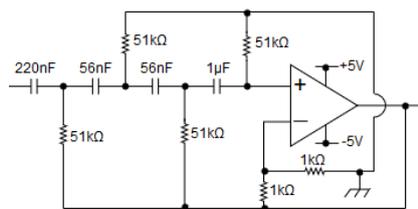


図 2.5 四次ハイパスフィルタ

### 2.6.3 非反転増幅回路

非反転増幅回路は，入力電圧のプラスマイナスを逆転させずに，出力電圧を増幅することのできる回路である．

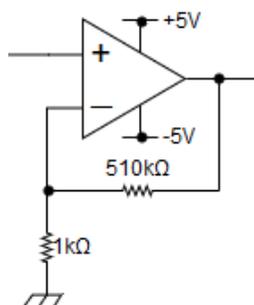


図 2.6 非反転増幅回路

(※文責: 豊見城風)

### 2.6.4 半波整流回路

半波整流回路は，交流電圧のマイナスの電圧のみを削除し，プラスの電圧のみの信号を出力させるものである．今回はダイオードを利用して半波整流を実現している．

(※文責: 豊見城風)

### 2.6.5 積分回路

積分回路は，入力された電圧を時間で積分して出力する回路である．また，ローパスフィルタの役割もしており，カットオフ周波数の高い周波数を減衰し，低い周波数を通過させることによりノイズをカットする役割を持っている．

(※文責: 豊見城風)

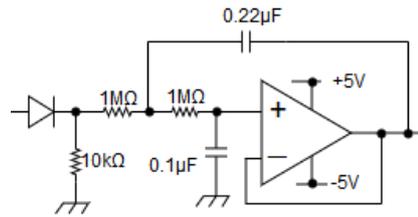


図 2.7 積分回路

## 2.7 制御回路

### 2.7.1 Arduino

Arduino とは AVR マイコンと呼ばれる小さな基板である。Arduino は C++ に似た Arduino 言語でプログラミングすることができる。今回は筋電位計測回路で得た信号に基づいて、Arduino を用いてサーボモータとステッピングモータを PWM 制御した。

(※文責: 安部日菜子)

### 2.7.2 L6470 使用ステッピングモータドライブレキット

L6470 使用ステッピングモータドライブレキットとは、バイポーラスステッピングモータドライブ IC の L6470 を使用したモータドライブレキットのことである。マイコンの SPI インタフェースで 2 層バイポーラスステッピングモータの正転、逆転、加速、減速、ブレーキ、速度制御、位置制御ができる。

(※文責: 安部日菜子)

### 2.7.3 ロータリエンコーダ

ロータリエンコーダとは、入力軸の回転の変位を内蔵した格子円盤を基準としてデジタル信号として出力する角位置センサである。

(※文責: 山本陽也)

### 2.7.4 ポテンシオメータ

ポテンシオメータとは、構造は可変抵抗そのものであるが、回転角度を検出することができる可変抵抗である。今回は炭素皮膜可変抵抗 B カーブ 50k Ω を用いて、回転角度を検出した。

(※文責: 山本陽也)

### 2.7.5 制御プログラム

Arduino 言語を使って、サーボモータの角度制御を行った。また、サーボモータを PWM 制御というパルス幅を変調することでサーボモータの回転を調節する制御方式で制御した。更に、Arduino で L6470 使用ステッピングモータドライブキットを手軽に制御できる公開されたプログラム [2] を一部変更して、ステッピングモータ角度制御を行った。

(※文責: 安部日菜子)

### 2.7.6 DC/DC コンバータ

DC/DC コンバータとは、ある直流電源を別の電圧の直流電源に変換するための装置である。これを用いた理由として、以下の 2 点が挙げられる。第一に、使用していたバッテリーが電圧 7.2V であり、サーボモータの電圧の 5V に下げる必要があるためである。第二に、筋電計測回路では±5V の両電源を必要としており、+5V から±5V に変換する必要があるためである。

(※文責: 山本陽也)

### 2.7.7 3 端子レギュレータ

3 端子レギュレータとは、入力端子 (IN)、出力端子 (OUT)、グラウンド端子 (GND) の 3 つの端子を持った電子回路の一種で、出力される電圧・電流を常に一定に保つように制御する回路のことである。

(※文責: 安部日菜子)

## 2.8 モータ

### 2.8.1 サーボモータ

サーボモータはモータ軸の回転角度/回転速度を回転検出器で検出し、モータドライバへフィードバックするため、モータ軸とモータドライバとの誤差が 0 になる。また、ステッピングモータに比べ高速回転時にも高いトルクを維持でき、さらにトルク変動に対し、エンコーダと連携した電流フィードバックを行うため、急な負荷変動に強い脱調が起こらないことが長所に挙げられる。短所としては、高い電源電圧必要になることとモータ・ドライバともに制御・機構部が複雑であるため、コスト高となること。またトルク変動に対し追従する際に遅れが発生することも短所として挙げられる。前期のプロトタイピングではすべての指に一つずつサーボモータを使用し、後期では中指と薬指に一つずつ使用した。

(※文責: 米山裕樹)

## 2.8.2 ステッピングモータ

ステッピングモータはドライバとセットで使用している。ステッピングモータはバイポーラ型とユニポーラ型があり、今回は同じ大きさでもより強い力を得られるバイポーラ型のステッピングモータを使用した。ステッピングモータはモータドライバへ入力されるパルス信号に応じ、モータの回転軸が回転する仕組みになっている。ステッピングモータ 1 パルスの入力に対し回転する軸の角度が決められている。例えば 1 パルスで  $0.9^\circ$  回転するステッピングモータを  $90^\circ$  回転させるには、100 パルスの信号を入力する必要がある。信号の周波数に比例して回転速度が速くなり、周波数を高めるとモータ回転は速くなり、低くすると回転が遅くなる。長所としては、静止時にトルクが最大となる、低い電源電圧にて駆動可能、構造がシンプルであり、低コストである、指令パルスに対し動作するため遅れが発生しにくいことが挙げられる。短所は負荷変動により脱調が起こることと高回転での運用は苦手なことである。脱調の対策として、ポテンションメータと一緒に用いてステッピングモータをサーボ制御することで脱調により乗じるずれを修正している。

(※文責: 米山裕樹)

## 2.9 コーティング素材

### 2.9.1 Ecoflex

Ecoflex とは、非常に柔らかく、耐久力の強い、付加タイプ・シリコーン樹脂ゴムである。室温で硬化して収縮はごくわずかである。A 液と B 液を 1 : 1 の割合で混ぜる。作業可能時間は 30 分であり、硬化時間は 4 時間である。

(※文責: 山本陽也)

### 2.9.2 義体用シリコン

義体用シリコンとは、実際の義肢などの作成にも使われている、質感は人間の肌に近いキャスト用のシリコーンである。A 液と B 液があり、1 : 1 の割合で混ぜる。作業可能時間は 20 分であり、硬化時間は 40~60 分である。ただし、 $120\sim 180^\circ\text{C}$  で硬化時間が短縮される。

(※文責: 山本陽也)

### 2.9.3 発泡ウレタン

発泡ウレタンとは、ポリウレタンに発泡剤を加え重合させたものである。非常に軽量で、断熱性、防音性にすぐれ、硬質のものから軟質のものまでバリエーションに富んでいる。A 液と B 液があり、5 : 2 の割合で混合し攪拌する。攪拌後、数十秒で発泡して膨れ上がる。10 分以上放置してから脱型する。

(※文責: 山本陽也)

## 2.10 外部部品

### 2.10.1 ソケット

ソケットは断端を収納し、義手と接続する部分である。ヒトと機械のインタフェースとなる部分であり、力を義手に伝達する。結合の良さが重要となるため、断端の形状とユーザーの動きにぴったりと合うよう個別に製作される。

(※文責: 山本陽也)

### 2.10.2 アームライナー

アームライナーは断端(切断部)をソックスのように覆い、断端の軟部組織と硬いソケットの間の摩擦などを防ぐ第2の皮膚のようなカバーである。ぴったりとフィットさせるためには、ここにあったアームライナーを選択する必要がある。

(※文責: 山本陽也)

### 2.10.3 3D プリンタ

本グループが用いた3Dプリンタは2種類ある。

1つは、大学工房内に設置されているMakerbot社製の3Dプリンタ”Makerbot Repliator 2X”である。この3Dプリンタは熱溶解積層法(FDM法)と呼ばれる方式が用いられている。ABS樹脂(Natural White, True Black, True White)などの熱可塑性樹脂を細長い線状にしたフィラメントを加熱して溶かし、ステッピングモータを用いた、押し出し機で押し出しながら積層する。

もう1つは、高木研究室が所有するFormlabs社製の3Dプリンタ”Form 2”である。この3Dプリンタは紫外線によって硬化する樹脂を用いた光造形法(STL法)と呼ばれる方法が用いられている。液体状の光硬化型樹脂にレーザー光をあて、一層分の樹脂を硬化させる。一層目が硬化したらステージを下げ、次の層にレーザー光をあてる。この作業を繰り返しながら積層する。

(※文責: 山本陽也)

### 2.10.4 レーザー加工機

本グループが用いたレーザー加工機は、大学工房内に設置されているUniversal Laser Systems社製のミドルクラスレーザー加工機”VLSプラットフォーム”である。低価格と高性能を両立したミドルクラスレーザー加工機でより高い汎用性がある。素材や加工内容に応じた最適な出力の発振器を選択することが可能である。

(※文責: 山本陽也)

### 2.10.5 Autodesk Fusion360

Autodesk Fusion360 は、Autodesk 社が提供しているクラウドベースの高機能 3D コンピュータ支援設計 (CAD) ソフトのことである。通常の 3DCAD ソフトは数十万円以上の販売価格がほとんどであるが、Fusion360 は毎月数千円で利用することができる。ただし、学生もしくは非営利・趣味用途であれば、無料で利用することができ、3D モデラやクリエ板によく使われている。Fusion360 ではスケッチを描いて、それを押し出したり、回転させるなどして立体のデータを製作することができる。また、シェル機能やスカルプト機能を備えているため、複雑な形状の物を簡単に作ることができる。本ソフトでは、STL 形式ファイルの出力が可能である。さらに、STL 形式ファイルを編集することもできる。作業スペースは”モデル”とし、”作成”のなかの”基準フィーチャを作成”を選択する。そのあと”挿入”から”メッシュを挿入”で STL ファイルを選択する。そして、”修正”の”メッシュ”のなかの”メッシュから BRep”を選択することで編集することができるようになる。基準フィーチャを終了してからでないでないと保存が出来ないので注意が必要である。3D プリンタで印刷を行う際に、本ソフトを活用した。

(※文責: 山本陽也)

### 2.10.6 Autodesk Eagle

Autodesk Eagle は、Autodesk 社が提供している電子設計自動化 (EDA) ソフトウェアのことである。Fusion360 と同様に、毎月数千円で利用することができ、学生であれば 3 年間無償で利用することができる。プリント基板 (PCB) を製作するために、回路図、コンポーネント配置、PCB 配線、包括的なライブラリコンテンツをシームレスに連携できる。

(※文責: 山本陽也)

### 2.10.7 ワイヤの固定方法

前期では金属のボビンを使ってワイヤをモータに固定していた。しかしこの固定方法ではワイヤを張っている状態でモータに固定することが難しいこと、時間が経つと張りが弱くなってしまふこと、ワイヤが伸びてしまふことなどの理由からワイヤが緩んでしまふ、義手の動作が安定しなかつた。そのため、後期では特殊な結び方をするこゝでワイヤが緩んでも締め直すことが出来るようにした。結び方はもやい結び、バタフライノット、神戸ヒッチを組み合わせたもので、動滑車の法則により締めるときは人力の約三倍で締めることが出来る。

(※文責: 米山裕樹)

## 第 3 章 課題解決のプロセス

### 3.1 前期プロトタイプの製作

#### 3.1.1 従来型技術の問題点

2016 年度のプロジェクトで製作された筋電義手では、重量が重いこと、物の形状や材質によって持てない場合や滑ってしまう場合があることの 2 点が問題点として残されていた。その結果を踏まえ、2017 年度のプロジェクトで製作された筋電義手では、軽量化に取り組んだが、一方で握力が弱くなってしまったことが問題点として残されていた。

(※文責: 岩崎香奈)

#### 3.1.2 考えた対応策

従来型技術では、物の形状や材質によっては滑ってしまうこと、軽量化により握力が弱くなってしまふことが問題点として残されていた。軽量化と握力の両立については、本学のプロジェクト学習の時間内では実現不可能であると考えた。そこで本プロジェクトでは、握力や把持力を向上させ、物の形状や材質によって滑ることを改良することを問題として設定した。この問題を解決するために、以下の対応策を考えた。

- (1) 柔らかい素材を義手に取り付け、接触面積を増やすことで滑りにくくする。
- (2) ワイヤの張りを強くし、摩擦の小さいものに変更する。
- (3) モータをよりトルクの強いものにし、握力を向上させる。

これらに対し前期では、次の手法で取り組んだ。(1) は、柔らかく弾力性のある素材を手袋状に加工し取り付けることで解決できると考え、義体用シリコン、人肌のゲル、Ecoflex を候補に挙げた。それぞれ人差し指に取り付け比較し、表面の粘性や柔らかさの観点から Ecoflex を選定した。義手の型を取り、Ecoflex を流し固めて手袋状の外装を製作した。(2) はワイヤと義手の摩擦が大きいことが原因でモータの力が弱くなってしまふという問題の解決を目指した。素材や太さの異なる 4 種類のワイヤをそれぞれの指に取り付け、実際に義手を動かした際にワイヤと義手の間に摩擦が生じないか、ワイヤが細すぎて義手の素材を削ってしまわないかどうかを検証した。また、ワイヤの固定が容易であるか、モータの力により伸びてしまわないかという点についても比較した。(3) は、従来使用していたモータよりトルクの大きいステッピングモータを使用し、サーボ制御をすることでトルクの強いサーボモータを自作することを考えた。そこでステッピングモータを数種類比較し、トルクの大きさ、入手しやすさ、重さ、制御のしやすさの観点から最適なものを選定した。選定したステッピングモータを制御し、義手の指を動かすことができた。

(※文責: 岩崎香奈)

### 3.1.3 前期プロトタイプの仕様

従来型の問題点を踏まえ、前期では、InMoov 社が公開している義手の 3D モデル [3] をもとに、工房に設置されている 3D プリンタを使用し、義手本体を ABS 樹脂で出力した。Ecoflex を手袋型に加工し、義手を覆うことで接触面積を増やした。指の関節部分はネジを適当な長さに切り、ネジとナットで固定した。適切なワイヤを選定するために、それぞれの指に素材や太さの異なる 4 種類のワイヤを通し、サーボモータに接続した。ワイヤは、モータ上のボビンにネジを取り付け、オーバルスリーブで固定した。3DCAD ソフトでサーボモータを収納する箱を製作し、3D プリンタで出力した。従来型と同様のサーボモータを使用しているため、義手の握力や把持力は変化していない。ワイヤカバーにはブレーキアウターケーブルを用いた。回路はユニバーサル基板を使用した。前腕屈筋群の筋電位を計測し、力を入れると義手が閉じ、力を抜くと義手が開く動作をするように制御した。



図 3.1 前期プロトタイプ

(※文責: 岩崎香奈)

## 3.2 前期プロトタイプの問題点

前期プロトタイプを製作して、実際に筋電位で義手を動かすことができた。しかし、前期プロトタイプの問題点として、以下の 4 点が挙げられた。1 つ目は、シリコン樹脂で製作した手袋が重くて、見た目が悪い点である。2 つ目は、サーボモータのトルクが弱いという点である。3 つ目は、親指の曲がる角度が原因で物が持ちにくいという点である。4 つ目は、ワイヤの固定方法が原因でワイヤが緩んでしまうという点がある。

(※文責: 安部日菜子)

### 3.2.1 手袋に関する問題点

手袋を作る際に、シリコン樹脂を均等に流すことができず、一部が薄くなって破けてしまったり、型を作る際に緑色の粘土を使用したため、型に粘土が付着してしまい、手袋が緑色になってしまったりして見た目が悪くなってしまった。また、手袋が予想していたよりもはるかに重くなってしまった。このことから、後期では軽くて柔らかい素材である発泡ウレタンと、滑りにくくするために凹凸のついた義体用シリコンを義手の必要な部分に取り付けることにした。

(※文責: 安部日菜子)

### 3.2.2 サーボモータに関する問題点

前期で使用したサーボモータではワイヤを引き続ける力が弱く、握力の向上ができなかった。しかし、前期で使用したサーボモータよりもトルクの強いモータを使用すると、高価で入手しにくいという問題点がある。トルクとはねじりのモーメントのことである。このことから、サーボモータを自作しようと考え、安価でトルクが強く、パルスを変更するだけで制御ができるステッピングモータを使用することに決定した。

(※文責: 安部日菜子)

### 3.2.3 親指の曲がる角度に関する問題点

InMoov 社の公開している義手の構造上、親指と人差し指の間がなかったため、筋電位で義手を動かした際に親指と人差し指の間に物を挟むことができず、物が持ちにくくなってしまった。このことから、後期では人差し指と親指の間の角度を約 90 度にし、物を挟む範囲を広げることにした。

(※文責: 安部日菜子)

### 3.2.4 ワイヤの固定方法に関する問題点

前期では、ワイヤとして使用した PE ラインをサーボモータとオーバースリーブを使って繋げていたが、何度か義手を動かしているうちに PE ラインが緩んでしまい、義手がすぐに動かなくなってしまった。このことから、PE ラインをオーバースリーブで固定するのではなく、もやい結び、バタフライノット、神戸ヒッチという 3 種類の結び方を組み合わせて、PE ラインが緩んだとしても、また締め直すことを可能とした。

(※文責: 安部日菜子)

## 3.3 後期プロトタイプの様

前期の問題点を踏まえ、物を掴む際に特に力が必要な親指、人差し指、小指にステッピングモータを使用し、中指と薬指にはサーボモータを使用した。また、前期ではシリコン樹脂を手袋型に

して義手の表面に柔らかい素材を取り付けたが、後期では、柔らかくて軽い素材である発泡ウレタンを義手の指全体と手のひらに張り付け、滑りにくくするために発泡ウレタンの上に凹凸のついた義体用シリコンを手のひら側全体に張り付けた。更に、物を持ちやすくするために、InMoov 社が公開している 3D モデルの親指の角度を変更し、人差し指との角度を約 90 度にした。義手を動かすためのワイヤには、前期と同様に PE ラインを使用して、PE ラインをカバーするためにブレーキアウターケーブルを使用した。筋電位計測回路は、ユニバーサル基板ではなく、銅版を使うことによって小型化した。前期では前腕屈筋群のみから筋電位をとっていたが、後期では前腕伸筋群からも筋電位をとることにした。前腕屈筋群の信号を受け取ると義手が閉じ、前腕伸筋群の信号を受け取ると義手が開くように動作制御した。また、前腕屈筋群と前腕伸筋群どちらからも信号がない時は、義手の開閉を止めるようにした。

(※文責: 安部日菜子)

### 3.3.1 評価実験

後期で製作したプロトタイプのパフォーマンスを評価するために実験を行った。被験者に、個人所有の能動義手と本プロジェクトで製作した筋電義手のそれぞれを装着し、以下の 3 つのタスクを行うよう指示し、どちらが物を持ちやすいか比較を行ってもらった (表 3.1)。1 つ目のタスクは、直径 10cm 以下の球体、各辺が 5cm 以下の四角推と四角柱の積み木を掴み、持ち上げ、深さ 10cm 程度の箱の中に移動させるタスクを行った。2 つ目のタスクは、各辺が 5cm から 15cm の箱をロープで縛るタスクを行った。3 つ目のタスクは、未開封の 500g のペットボトルを持ち上げ蓋を開け、コップに注ぐタスクを行った。ペットボトルの中身が飛散するのを防ぐため、水の代わりにスライムを用いた。それぞれのタスクにおいて、写真及びビデオによる撮影を行いながら、動作の様子を観察した。その後、製作した筋電義手の装着感や使用感、性能についての聞き取り調査を行った。実験装置の移動や休憩時間を含め、被験者の拘束時間は 90 分から 2 時間程度であった。

表 3.1 評価実験の内容

タスク	実験内容
1 つ目のタスク	直径 10cm 以下の球体、各片が 5cm 以下の四角推と四角柱の積み木を持ち上げ、深さ 10cm 程度の箱の中に移動させる。
2 つ目のタスク	各片が 5cm から 15cm の箱をロープで縛る。
3 つ目のタスク	未開封の 500g のペットボトルを持ち上げて蓋を開け、コップに注ぐ。

(※文責: 岩崎香奈)

### 3.3.2 被験者

一昨年度に、株式会社馬場義肢製作所の紹介で被験者になっていただいた方に今年度も協力してもらった。欠損部は、左前腕で手首の付け根辺りから先が欠損している。普段は能動義手を使用しており、日常生活で問題なく使用している。

(※文責: 山本陽也)

## 3.4 各々の課題の概要とプロジェクト内における位置づけ

### 3.4.1 米山裕樹

米山 裕樹の担当課題は以下の通りである。

<前期>

5月

現在研究されている筋電義手の問題点について分析をし、その結果から班員で話し合い、グループ A の目的を決定する。また、筋電義手の製作に必要な材料の決定し注文をした。

6月

担当教員とその研究室の方の協力のもと、筋電測定回路についての知識と技術を習得する。後期使用する予定のステッピングモータの試運転し、その性能の確認と後期に使えるかの検討をした。ブレットボードで筋電位計測回路を製作した。筋電位からサーボモータを動かすプログラムの作成。

7月

義手に使用するユニバーサル基板による筋電位測定回路の作成。筋電位計測回路とモータと義手をあわせて筋電義手を製作した。中間発表で使用するポスターとスライドの作成。中間報告書の作成

8月

中間は発表から得られたフィードバックをを分析し、問題点を抽出した。その結果から、後期の課題を設定した。

<後期>

9月

筋電計測回路に使用する基盤を小型化するために EAGLE を使った基盤作成の方法を研究室の方に指導してもらい、基盤加工機と EAGLE の使用方法を教わった。

10月

基板加工機を使った筋電計測回路の作成

11月

ステッピングモータとサーボモータの制御プログラムの作成。そして、完成した成果物を使って評価実験を二回行ったが、二回とも失敗に終わった。

12月

評価実験とその結果の分析。最終発表向けの練習と本番の発表。最終報告書の作成

1月

最終報告書の作成

(※文責: 米山裕樹)

### 3.4.2 豊見城凧

豊見城凧の担当課題は以下の通りである。

<前期>

5月

アイデア出しを行い、テーマを決定を行った。担当教員の指導のもとプロジェクトにおける基礎知識を習得した。また、プロジェクトの筋電義手の現状における問題点についての分析を行い、具体的な目標を設定した。

6月

担当教員やその研究所の先輩方のもと、電極や筋電義手を制御する回路、ユニバーサル基盤の基礎製作技術を習得した。その後、実際に製作に移り、使用可能な段階になるまで製作を行った。

7月

筋電位で実際に製作したプロトタイプを動かす、最終調整を行った。その後、中間発表に向けてスライドの製作を行い、最後に中間報告書を作成した。

8月

中間発表会での評価から、後期で改善すべき問題の検討を行った。

9月

義手に使用するモータとギアの選定を行い、検討した。

<後期>

10月

評価実験を行うことを決定し、実際に強力していただける前腕欠損者の方にお会いした。また、遊星ギアを取り付けたためモータの減速比が大きく望んでいる回転速度より遅かったため、ギア比の検討を行った。

11月

当時使用していたギアボックスに金属加工を行い、必要としている減速比を得ることができた。また、ワイヤに使用する釣り糸の選定を行い、米山とともに実際に釣り具店へ行き、確認した。電極とグラウンドの製作も行った。

12月

モータを固定するため金属部品の加工を行った。更に、成果物の評価実験を行った。また、最終発表に向けて原稿の製作を行った。

(※文責: 豊見城凧)

### 3.4.3 安部日菜子

安部日菜子の担当課題は以下の通りである。

<前期>

5月

アイデア出しを行い、テーマを決定した。現在の筋電義手の課題分析を行った。筋電位計測回路の知識を習得した。

6月

筋電位計測回路と電極の製作を行った。ユニバーサル基板の設計図を制作した。また、中間発表で使用するポスターの制作も行った。

7月

## Body expansion interface using electromyogram -ASHURA-

筋電位で義手を動かした。中間発表の練習を行った。最後に中間報告書を作成した。

8月

中間発表会での評価から、後期で改善すべき問題の検討を行った。ロータリエンコーダを使って脱調を検出し、ステッピングモータを制御することを決めた。

<後期>

10月

評価実験の内容を決めた。また、実際に被験者の方と会い、評価実験を円滑に進めるためのコミュニケーションをとった。実験計画書と承諾書の作成を行った。

11月

製作した筋電義手の評価実験を2回行った。筋電位計測回路の電源を製作した。

12月

製作した筋電義手の評価実験を行った。成果発表会のための原稿とポスターの作成を行った。また、最終報告書の作成を行った。

(※文責: 安部日菜子)

### 3.4.4 岩崎香奈

岩崎香奈の担当課題は以下の通りである。

<前期>

5月

アイデア出しを行い、テーマを決定した。担当教員に指導していただき、義手製作における基礎知識を習得した。また、筋電義手の現状における問題点についての分析を行い、具体的な目標を設定した。外装の素材について意見を出し合い、義体用シリコン、人肌のゲル、Ecoflexを選定した。

6月

工房職員の方に協力していただき、3Dプリンタの使い方を学習し義手のパーツの出力を行った。出力したパーツをそれぞれ研磨した後、義手の組み立てを行った。使用するワイヤの候補を絞り、指ごとに異なるものを取り付けた。型取り材で作った型に、選定した外装の素材を流し固め、人差し指に取り付けた。その結果から、Ecoflexを使用することを決定し、手袋型の外装を製作した。

7月

外装の手袋の破れた部分にEcoflexを塗布し、接着を行った。ワイヤの長さを調整し、ブレーキアウターケーブルをワイヤカバーとして取り付けた。モータとワイヤをつなぎ固定し、義手の動作を確認した。ポスターと発表原稿を作成し、中間発表会を行った。また、中間報告書を作成した。

<後期>

9月

グループ内で話し合い、後期の目標について再確認した。親指の曲がる角度に問題があったため、親指の構造を変更することを決めた。また、前期で製作した義手が大きすぎたため、各指の長さを短くすることにし、3DCADソフトを用いて各指の3Dモデルを修正した。被験者の方が使用できるよう、左手のパーツを3Dプリンタで印刷した。使用するワイヤをPEラインとフロロカーボンの2種類に決めた。性能評価実験の内容について話し合った。

10月

ワイヤカバーの素材について話し合い、PTFE チューブを使うことに決めた。ワイヤは、PTFE チューブとの摩擦力の面からフロロカーボンを使うことにした。外装の素材について新たに発泡ウレタンを使うことを検討し、試作をした。指を固定していたネジの代わりに、3D プリンタに用いる 3mm のフィラメントを使い、義手を組み立てた。被験者の方と話し、性能評価実験の詳細について決め、実験計画書と承諾書を作成した。馬場義肢製作所を見学し、実際の義肢製作の工程について学んだ。

11月

発泡ウレタンを薄く固め、パーツごとに切り分けた。3D プリンタのサポート材を型にし、表面が凹凸状になるように義体用シリコンを固めた。発泡ウレタンを指全体と手のひらにつけ、手のひら側の表面に義体用シリコンを貼り付け外装を作製した。義手の性能評価実験のための準備をした。PTFE チューブが柔らかすぎてモータの力に負け、曲がってしまったため、ワイヤカバーをブレーキアウターケーブルに変更した。また、フロロカーボンが伸びてしまったため、PE ラインを用いることにした。1 回目の評価実験では、ワイヤカバーの切り口でワイヤが切れてしまった。そこで、ワイヤカバーの切り口に PTFE チューブを取り付けることでワイヤを保護した。2 回目の評価実験では、モータに誤動作が発生し、様々な問題が明らかになった。ワイヤの固定方法を変え、調整可能な機構にした。ポスターと発表原稿を作成し、成果発表会の準備を行った。

12月

改良した義手の最終調整を行い、3 度目の評価実験を行った。成果発表会を行い、評価シートの意見をまとめた。最終報告書の分担をし、作成作業を進めた。

(※文責: 岩崎香奈)

### 3.4.5 鈴木智子

鈴木智子の担当課題は以下の通りである。

<前期>

5月

筋電義手についての分析を行い、アイデア出しをし、目的を設定した。その後、目的達成のために問題を分析し、解決方法の策を立てた。その後、課題を割り当て、義手に取り付ける素材の調査を行った。

6月

3D プリンタ使用のための知識習得をし、中指、人差し指数本の製作を行った。調査した素材、人肌のゲル・義体用シリコン・ECOFLEX を義手に試用し、素材を ECOFLEX に決定した。その後、製作した義手のパーツを組み立てた。手袋の製作方法を検討し、決定した。ECOFLEX で柔らかい手袋の製作、予備として義体用シリコンの手袋を製作した。

7月

サーボモータを固定する箱の 3D モデルを設計し、3D プリンタを用いて出力した。ワイヤカバーを取り付け、筋電義手の製作が完了した。その後、中間発表のスライド、原稿、ポスター製作をし、準備を行った。

8月

中間成果発表会での評価フィードバックを受け、製作した筋電義手の問題を分析し、今後の計画

を練った。ワイヤを最適なものへの変更，手の大きさの変更・親指の角度の変更・素材の変更を決定した。

<後期>

10月

後期の活動内容の計画を行い，おおよそのスケジュールを立てた。親指の曲がる角度を修正指の長さを短くすることを決定した。義手の外側に着ける素材について，新たな候補に挙げた発泡ウレタンを人差し指につけ，使用できるか試みた。評価実験の具体的な内容を決め，実験計画書と承諾書の作成を行った。その後，被験者の方と話し，馬場技師製作所で義手や義足に使用されているソケットの製作方法についての話を聞いた。

11月

義手に着ける素材や方法の案を絞り，発泡ウレタンの上に薄く凹凸を付けた義体用シリコンを付けることを決定した。また，前期の中間発表で指摘があった，手の甲に付ける意味はあるのか，という意見を取り入れ，手の甲側を薄く，手のひら側を厚くする型の取り方を試行錯誤し，決定した。その後，義手に取り付ける発泡ウレタンと義体用シリコンをそれぞれの指の数用意した。また，モータ系を固定するための箱を設計し，出力したが，強度に問題があったため，レーザーカッターで製作した。

12月

評価実験を行い，その結果をまとめた。最終成果発表に向け，スライド，発表原稿，ポスターの準備を行った。

(※文責: 鈴木智子)

### 3.4.6 山本陽也

山本陽也の担当課題は以下の通りである。

<前期>

5月

アイデア出しを行い，グループのテーマを決め，義手の手袋に用いる素材の選定をし，義手の問題点の分析，および義手のパーツの印刷を行った。

6月

引き続き義手のパーツの印刷を行い，印刷し終わった義手のパーツを組み立て，手袋の素材を決定し，手袋を製作した。

7月

手袋をはめた義手と回路基板とサーボモータの結合をした。また中間発表の準備，および報告書の製作を行った。

8月

中間成果発表会でのフィードバックより，後期に何をどうするか決めた。ワイヤの変更，サイズの変更，親指の角度の調整を決定し後期が始まるまでの課題とした。

<後期>

10月

親指の角度調整のため、Fusion360 でデータを修正した。ワイヤを通すチューブとして、柔らかい PTFE チューブを使用することを決めた。左手のパーツの印刷をした。前期の Ecoflex の問題点を受け、後期は発泡ウレタンを使うことに決めた。発泡ウレタンを指につけるための型を作るためのモデルを印刷した。前腕欠損者の方とお会いした。馬場義肢製作所を訪問し、ソケットやライナーなど、初めて見るものがあったり、さまざまなことを学ぶことができた。

11 月

発泡ウレタンを熱を加えることによって変形できる粘土をつかって、義手の指の形に合わせて型をとった。指の表面に指紋のような凹凸を付けるため、3D プリンタでプリントした際にできるサポート材を用い、義体用シリコンを薄くながすことで製作した。大学を訪れた中学生に向けてのプロジェクト発表のためのスライドを製作した。ギアのパーツを 3D プリンタとレーザー加工機で製作した。ソケットと義手を繋ぐ手首を設計、印刷した。PTFE チューブが、モータの力に耐えきれず、曲がってしまうことが分かったため、急遽チューブをブレーキアウターケーブルに変更し、通しなおした。さらに、ワイヤもフロロカーボンでは伸びてしまうため、PE ラインに変更した。評価実験の準備を行った。製作したモノが、当初の目的を達成しているか評価実験 1 回目、2 回目を行った。1 回目の実験では、ワイヤの角度が急に変わってしまっていたため、そこで摩擦によりワイヤがすり減ってしまって切れてしまった。そこで急に角度が変わってしまう部分に PTFE チューブを入れることで摩擦を減らすことにした。2 回目の実験では、ロータリエンコーダを用いたことによってステッピングモータが設定していた角度にまで戻らず、一定回数指を開閉するとワイヤがずれてしまっていた。さらに、レーザー加工機で製作したボビンの隙間にワイヤが挟まってしまった。そのため、急遽高木研究室が所有する STL 法を用いている 3D プリンタをお借りし、強度がある一体型のボビンを製作した。

12 月

評価実験の 3 回目を行った。3 回目の実験では、過去 2 回の実験の問題点をなくし実験が始まってからは問題なく動作した。成果発表のために発表原稿を覚えたりなどの準備を行った。成果発表を行い、周りからのフィードバックを得ることができた。報告書を作成した。

(※文責: 山本陽也)

## 3.5 担当課題と他の課題の連携内容

### 3.5.1 米山裕樹

前期では回路基板の制作を安部が行い、それを半田付けした。安部とサーボモータの設定とそれを制御するプログラムを Arduino で製作した。後期では評価実験から得られた、フィードバックから問題点の分析し改善案を出した。その後夏休みに研究室の方に教えていただき、EAGLE でプリント基板を作成した。その後、安倍と豊見城と協力してステッピングモータでサーボモータを自作した。また同時にモータの制御プログラム変更し指を途中でも止められるようにした。

(※文責: 米山裕樹)

### 3.5.2 豊見城凧

前期では安部と米山とともに筋電位計測回路と電極の制作した。その後、ワイヤの選定を行った。後期では、電極の製作を行ったほか、ステッピングモータとモータを固定する部品をグループが必要としている形、強さに加工し、それを安部と米山が制御を行った。また、山本と協力し、ボビンの製作を行った。最後に、岩崎、鈴木、山本が作った義手と、米山と安部が調整したサーボモータ、筋電位計測回路を繋げた。

(※文責: 豊見城凧)

### 3.5.3 安部日菜子

前期では、米山、豊見城と筋電位計測回路の製作を行い、ユニバーサル基板の設計図を制作し、米山が半田付けを行った。また、米山とサーボモータを動かす設定を行った。最後に岩崎、鈴木、山本が製作した義手と、ユニバーサル基板、サーボモータを繋げた。後期では、グループの中で評価実験について話し合い、実験計画書と承諾書の作成を行った。また、米山が製作した筋電位計測回路の電源の製作を行った。最後に、岩崎、鈴木、山本が作った義手と、豊見城が改造したステッピングモータと、米山が調整したサーボモータ、筋電位計測回路を繋げた。

(※文責: 安部日菜子)

### 3.5.4 岩崎香奈

前期では、鈴木、山本と義手のパーツの出力と組み立てを行い、ワイヤを通して義手を製作した。また、鈴木、山本と義手の型をとり、シリコーン樹脂を型に流し固めて手袋を製作した。それを米山、豊見城、安部が製作した回路基板とサーボモータに結合した。後期では、鈴木、山本と義手を組み立て、発泡ウレタンと義体用シリコンで外装の製作を行った。米山、豊見城、安部が製作した回路基板とモータを接続し、モータのボビンに義手のワイヤを結んで固定した。

(※文責: 岩崎香奈)

### 3.5.5 鈴木智子

岩崎、山本と義手のパーツを印刷し、部品の調整、組み立てを行い、義手のワイヤを通して義手を製作した。また、岩崎、山本と素材の決定方法や型の取り方を決定し、素材を人差し指に取り付けて実験を行い、素材を決定し、手袋の製作を行った。また、米山、豊見城、安部が製作した回路基板を取り付け、筋電義手に付ける柔らかい手袋の製作を行った。後期では、山本、岩崎と義手の組み立てを行った。山本、岩崎と義手に取り付ける物の検討をし、発泡ウレタンと義体用シリコンの製作を行った。その後、米山、安部、豊見城が製作した回路基板やモータなどにワイヤ、チューブ、ボビンなどを取り付け、筋電義手の製作を行った。

(※文責: 鈴木智子)

### 3.5.6 山本陽也

鈴木，岩崎と義手の各パーツの印刷と組み立てを行い，組み立てた義手にワイヤを通し，かつシリコーン樹脂でコーティングをした．また，米山，豊見城および安部が製作した回路基板とサーボモータを結合した．後期では，発泡ウレタンでカバーを作り，凹凸を付けたシリコーン樹脂をその表面に付着した．また，米山，豊見城および安部が製作した回路基板とステッピングモータ，サーボモータを PE ラインを使って結合するためのボビンとなるパーツの製作をレーザー加工機や 3D プリンタで製作した．

(※文責: 山本陽也)

## 第 4 章 結果

### 4.1 最終成果物

#### 4.1.1 中間発表までの問題点

中間発表により製作したプロトタイプにはいくつかの問題点がみられた。その中で主な問題点は 4 つある。1 つ目は、筋電義手に取り付けているシリコン樹脂が重い。2 つ目は、筋電義手の親指の角度が物を掴むということに対して不適切である。3 つ目は、握力自体が今年の筋電義手と同等であり、目的である把持力の向上がなされていない。4 つ目は、ワイヤの固定の仕方が不十分である。この 4 つの問題点を解決することを最終報告までの課題とし取り組んだ。

(※文責: 豊見城 凧)

#### 4.1.2 中間発表までの問題点の解決方針

上記の問題点を解決するにあたって、後期ではこれらの解決方針を決定した。1 つ目の外装は軽量化しながら、摩擦力の向上に取り組んだ。手のひら側のみ発泡ウレタンと薄く加工した義体用シリコンの二重構造の部品を取り付け、義体用シリコンには凹凸をつけることにより摩擦力を向上させることにした。2 つ目は、親指の角度を 45 度から 90 度に変更し、義手の動作を、開閉のみではなく、手の甲側に力を入れると開き、力を抜くとその状態で一時停止、手のひら側に力を入れると閉まるという動作にすることで、物を掴む際の加減を調整しやすくすることとした。3 つ目は、昨年は DC モータを使用していたが、今回は中指と薬指には DC モータを使用し、物を掴む際に最も力が必要な親指と人差し指、小指に遊星ギア付きステッピングモータを使用することにより握力を向上させることとした。4 つ目は、前期ではワイヤの固定方法が不安定だったため、義手の動作がぎこちなかったため、ワイヤを固定するにあたって適切な部品等を製作、探し出すこととした。後期はこの 4 点に重点を置き作業を行った。

(※文責: 豊見城 凧)

### 4.1.3 最終成果物の仕様

上記の解決方針に沿って、各自で作業を行った。製作した成果物の仕様は以下のとおりである。Arduino と自作基板により、手の甲側に力を入れると開き、力を抜くとその状態で一時停止、手のひら側に力を入れると閉まるという動作制御を行っている。モータは中指と薬指には DC モータを使用し、物を掴む際に最も力が必要な親指、人差し指、小指に遊星ギア付きステッピングモータを使用することにより握力を向上させた。ワイヤは PE ラインを使用し、モータとの接続は 3D プリンタで製作したボビンに巻き付け、引っ張る際の摩擦をできる限り減らした。指先はワイヤがモータの力でほつれないようにきつく縛り、義手とワイヤの接触部分には PTFE チューブを使用することで、摩擦を減らしつつ、ワイヤの保護も行った。また、指の曲げる速度を指ごとに変えることにより物を掴みやすくした。

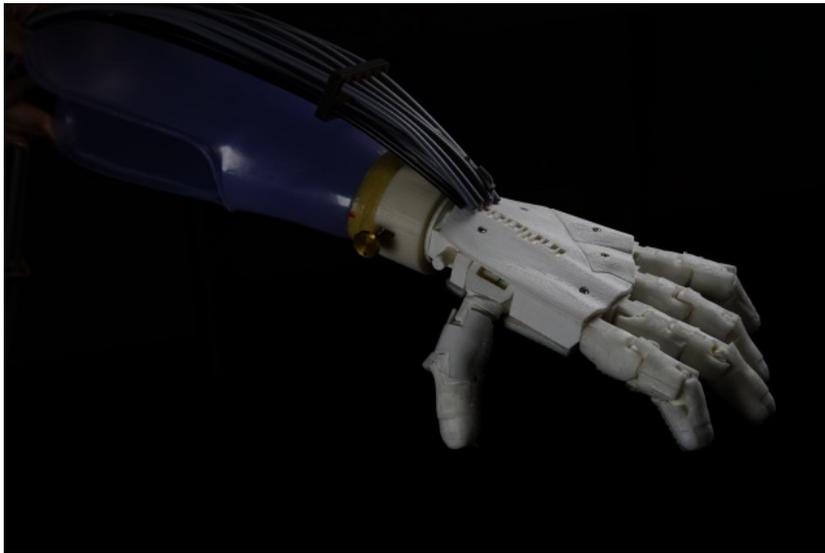


図 4.1 最終成果物の仕様

(※文責: 豊見城凧)

## 4.2 評価実験

### 4.2.1 評価実験の概要と成果

今年度製作した筋電義手を前腕欠損者に装着していただき、筋電義手の性能評価実験を行った。実験では、まず、日頃使用している能動義手で実験を行い、その後、製作した筋電義手で実験を行った。検証のために 3 つのタスクを行った。

まず、積み木の移動のタスクを行った (図 4.2)。球体は問題なく掴み、移動させることが出来ていた。しかし、小さい積み木では掴むのに時間がかかっており、薄い長方形の積み木では積み木が倒れてしまうため、持ち上げが困難であった。

次に、箱をロープで縛るタスクを行った (図 4.3)。ロープを強く保持することが出来ており、結ぶ作業も固く結ぶことが出来ていた。タスクにかかった時間は能動義手、筋電義手どちらも 1 分 9 秒であった。

最後に、ペットボトルの持ち上げ・開封・コップへの注水のタスクを行った (図 4.4)。500g の

## Body expansion interface using electromyogram -ASHURA-

ペットボトルを持ち上げることが出来ていた。開封については、持ち上げた状態ではペットボトルごと回転してしまったため不可能であったが、机上に置いた状態での開封は可能であった。その後のコップへの注水も可能であった。

総括すると、ロープを握ることが出来ていたことやペットボトルを持ち上げ、開封できたことから、昨年度と比較して、把持力、握力が向上したことが分かった。



図 4.2 積み木の移動のタスク



図 4.3 箱をロープで縛るタスク



図 4.4 ペットボトルのタスク

(※文責: 鈴木智子)

### 4.2.2 評価実験における問題点

それぞれのタスクで問題点が見受けられた。まず、積み木の移動のタスクで小さい積み木や薄い長方形の積み木の移動が困難であった理由として、手首の回転が不可能であるため、掴むために最適な角度から掴むことが出来なかったことが考えられる。さらに、積み木に触った際に、その感覚が得られないため、どの程度掴んでいるのかを目視で確認する必要がある。目視で確認する際、掴む対象の大きさが小さいと確認が困難であったことも考えられる。次に、箱をロープで縛るタスクで時間がかかっていた原因として、腕に力を入れてから義手が動作するまでに時間差があったことが問題点として挙げられる。ペットボトルの持ち上げ・開封・注水のタスクでの、持ち上げたままでの開封ができなかった原因として、親指の曲がる角度に問題があったため、正常に握ることが出

## Body expansion interface using electromyogram -ASHURA-

来ておらず、掴む力がペットボトルに伝わらなかった点が挙げられる。さらに、手のひらの面積が大きく、ペットボトルと密着させて持ち上げられなかったことも原因として挙げられる。すべてのタスクで共通した問題として、腕を動かしづらかった問題があった。原因として以下の2点が挙げられる。1つ目は、モータや制御系の部品を長机上に置いた状態での実験だったため、ブレーキアウターケーブルが体に密着していなかった点である。2つ目は、チューブが固く、重いため、腕の可動域が狭くなった点である。

(※文責: 鈴木智子)

## 第 5 章 まとめ

### 5.1 前期終了時における進捗と問題点

グループ A は、試装着用の筋電義手の性能を向上させるために活動してきた。前期では、最初にプロジェクト全体で櫻沢先生に筋電位計測回路の仕組みについて講義を受けた。次に昨年の評価実験の動画等を見ながら課題の分析を行った。その結果、昨年プロジェクトで製作された筋電義手には以下の 2 つの問題があることがわかった。一つが、軽量化による耐久性の低下である。もう一つが、モータの数を減らしたことによる握力、把持力の低下である。この 2 つの問題点のうちグループ A は握力、把持力の低下に注目し、物を掴みやすい筋電義手を製作することを目的とし、握力、把持力の向上を目標とした。そこで、義手の表面に柔らかい素材をつけることと、トルクの強いモータを自作することで、それらの課題を解決することにした。そして、そこから外装班と回路班に分かれて活動した。外装班は義手に取り付ける素材を選定し、義手より一回り大きい型を製作した。さらに、そこに Ecoflex を流し込み硬化させることで義手に素材を取り付けた。Ecoflex とはシリコーン樹脂の一種である。回路班は筋電位計測回路をブレッドボードで製作し、その後ユニバーサル基板で筋電位計測回路を小型化した。そして筋電位計測回路からの信号でサーボモータを制御できるように Arduino をプログラミングした。最後に、Ecoflex で覆われた筋電義手の開発を行った。Ecoflex で覆った理由は、物を持った際に滑りにくくするために、筋電義手と掴む物との接触面積を広くすることを目的としたためである。その結果、滑りやすさを改善することができたと考えられる。

プロトタイプ製作を行って明らかになった問題点として、以下の 5 点が挙げられた。1 つ目は、Ecoflex を義手に取り付ける方法である。義手よりも一回り大きいサイズの型を作りそこに義手と Ecoflex を入れて柔らかい筋電義手を作ったが、義手の固定方法がテープだったためしっかり固定することができなかった。そのため、義手がずれてしまい、極端に厚い部分と薄い部分が出てしまった。その結果、Ecoflex が破れる、指の運動を阻害する、といった問題が起こった。2 つ目は、筋電義手につけた Ecoflex が重すぎたことである。弾力をもたせるために Ecoflex の手袋を厚くした結果、義手自体が重くなった。これにより筋電義手が扱いづらくなった。3 つ目は、現在のサーボモータではトルクが足りないことである。そのため、さらに大きなモータを使う必要があった。しかし、サーボモータに使われている DC モータは高価なため、十分なトルクを得られるサイズのモータを購入することが難しい。4 つ目は親指の曲がる方向に問題があった。現在、人差し指に対する親指の角度が 45 度程度でしか曲げることができない。しかしその状態では、掴む、つまむといった動作をすることが難しく、特に、大きなものが掴みにくくなっている。5 つ目は、ワイヤの張りが弱かったことが挙げられる。これはワイヤを張っている状態でモータに固定することが難しいこと、時間が経つと張りが弱くなってしまうこと、ワイヤが伸びてしまうことが原因である。以上のことが前期のプロジェクトで明らかになった問題点として挙げられる。

(※文責: 米山裕樹)

## 5.2 後期における問題への対応

後期では上記で発見した問題に加え新たな問題が指摘された。それは、プロジェクトで製作している従来の義手では筋電位が高いときは手を閉じ、筋電位が低いときは手を開くといった機構になっているため、手の開閉を途中で止めることができなかったことである。しかし、モータのトルクを上げたことで、従来の機構ではものを掴んだとき、握りつぶしてしまう可能性がある。そのため、後期では筋電位を前腕屈筋群に加え、前腕伸筋群からも計測することで手の開閉を途中で止められるように機構を変更した。

前期の活動を通して明らかになった5つの問題点について以下のように対応した。1つ目の問題が、Ecoflexを義手に取り付ける際に、極端に厚い部分と薄い部分ができってしまったことだ。その結果、Ecoflexが破れる、指の運動を阻害する、といった問題が起こった。これはEcoflexの加工が難しいため、より加工が簡単な発泡ウレタンを使うことで解決した。また、発泡ウレタンはEcoflexよりも軽いため、2つ目の問題である義手の過重量も解決した。3つ目の問題がモータのトルク不足だ。前期で使用したサーボモータではトルクが足りず十分な握力を得ることができなかった。そこで前期より大型のサーボモータを購入しようと考えた。しかしサーボモータに使用されているDCモータは高価なため、前期より大型のサーボモータを買うのは難しいため、安価なステッピングモータでサーボモータを自作することで、より大きなトルクを得ることができた。4つ目の問題が、親指の曲がる方向にあった。前期では、人差し指に対する親指の角度が45度程度しかなかったため、物を掴むときにものにうまく力が伝わっていなかった。そこで人差し指に対する親指の角度を90度程度に変更することで力がうまく伝わるようになった。5つ目の問題が、ワイヤの張りが弱かったことだ。これはワイヤを張っている状態でモータに固定することが難しいこと、時間が経つと張りが弱くなってしまうこと、ワイヤが伸びてしまうことが原因として挙げられる。そのため、後期では特殊な結び方をすることでワイヤが緩んでも、締め直すことができるようにした。これにより、義手の動作が安定した。

(※文責: 米山裕樹)

## 5.3 今後の展望

今回のプロジェクトでは試装着用の筋電義手の握力向上と把持力の向上を目的に活動してきた。把持力は十分な向上が見られた。それに対し握力は必ずしも十分な向上が見られなかった。結果的に十分な握力を得ることはできなかったのさらなる握力の向上が必要になる。今回のプロジェクトでは3Dプリンタで義手を出力している。3DプリンタのモデルにはInMoov社が公開しているヒューマノイドロボットの手の部分を使用している。しかし、この手の構造に問題があり、ものを掴む動作が難しくなっていること、ワイヤと義手との摩擦が大きいため、モータから指への力の伝達率が悪く、モータのトルクを上げても義手の握力の向上に繋がりにくいことがわかった。このことから義手の構造を変更することでこれらの問題を解決する必要があると考えられる。今後、日常的に筋電義手を使うことを考えると小型化、軽量化、携帯性の向上が必要になると考えられる。しかし、現在のプロジェクトで製作した義手は、市販されている筋電義手に比べ、指の繊細な動きや握力といった能力に劣っているため、小型化、軽量化、携帯性の向上は、筋電義手の指の繊細な動きや握力といった能力が向上した後からでも良いと考えられる。現在では多くの機器が手持ちの

## Body expansion interface using electromyogram -ASHURA-

スマートフォンと連動して，スマートフォンから便利な情報を得ている．この機能を今後実装することで，さらに便利な義手になると考えられる．今後も筋電義手の本グループの取り組みが活かされ，改良が続けられていくことを期待している．また，筋電義手の普及に向けて，今後も取り組み続けてほしいと考えている．

(※文責: 米山裕樹)

## 第 6 章 評価フィードバック

### 6.1 成果発表会

2018 年 12 月 7 日 (金) に 2018 年度プロジェクト学習の発表会が開催された。発表会では、1 年間のプロジェクト活動をもとにプロジェクト全体のテーマと各グループの活動をスライドで説明し、その後、グループごとに分かれ制作したプロトタイプの実演を行った。その際に聴衆の方々に発表内容の評価をして頂いた。その結果、52 枚の評価シートを頂き、そのうちの 14 枚はグループ A の評価であった。また、14 枚中 3 枚だけ点数が未回答であったため、11 枚の評価シートを集計した。10 点満点のうち、グループ A に対する評価の平均点は 8.64 点、全体的な発表技術の平均点は 8.23 点であった。全体的な発表内容の平均点は 8.70 点であった。以下に良かった点と悪かった点の一覧を示す。

(※文責: 鈴木智子)

#### 6.1.1 良かった点

- 実演が丁寧で分かりやすかった。
- 生き物の成長を見ているような気持ちになった。面白かった。
- 物を握りやすい指の順番などを考えられていて面白かった。
- 前期に見た義手はまだ改善点たくさんあると感じたものだったのが義手としてしっかりものが持てるまで改善されていた。
- 目標をはっきりしており実現できていて素晴らしい。
- 成果物の完成度も高く、質疑応答もしっかりしていて良かったと思う。
- ハキハキしていてとても分かりやすかった。
- 内容がよくわかりやすく、実際に触れてみることで詳しく分かった。
- 目的が明確でよかった。
- 前期や昨年度との違いが明確で分かりやすかった。
- 研究の流れについて理解できた。
- 実際の物を見れて良かった
- 筋電義手のこれからの期待できる。
- 成果物の完成度も高く、質疑応答もしっかりしていて良かったと思う。
- 研究の流れについて理解できた。もう少し義手の動きを見たかった。

(※文責: 鈴木智子)

#### 6.1.2 悪かった点

- 物をつかむ所の実演が見たかった。
- 義手をもう少したくさんの方が体験した感想が欲しかった。

- ペットボトルを持ち上げる様子を見たかった。

(※文責: 鈴木智子)

## 6.2 まとめ

これらのコメントをまとめると、良かった点として筋電義手の説明や活動内容についての説明方法が分かりやすく、評価が高かった。悪かった点をまとめると、実際に筋電義手を動かすときに物を持ち上げる動作が必要だったということが挙げられる。これらの改良点として、発表練習にもっと時間を設けることが挙げられる。以上の評価内容を踏まえ、私たちの評価を5段階評価で表すと4である。望ましい点として、他のプロジェクトと比較すると、やりたいことが明確であり、前期や昨年度と成果物の違いをしっかりと報告できていたと感じたことが挙げられる。その一方、制御機器の配線を綺麗することができなかったという問題点が挙げられる。これらのことから、総合評価として4と考えられる。

(※文責: 安部日菜子)

## 付録 A 課題解決のための新規獲得技術

プロジェクト学習を通して獲得した技術は、3D プリンタで加工を行う技術、回路を製作する技術の2つである。

(※文責: 安部日菜子)

## 付録 B 活用した講義

情報処理演習 II, 人体生理学, 櫻沢先生による回路の講義

(※文責: 山本陽也)

## 付録 C 制御プログラム

以下に Arduino の制御プログラムを示す

---

```

1  #include <SPI.h>
2  //ピン番号宣言
3
4
5  int ENCval;//可変抵抗に使う
6
7
8
9  //変数の宣言
10
11
12
13 // ピン定義。
14 #define PIN_SPLMOSI 11
15 #define PIN_SPLMISO 12
16 #define PIN_SPLSCK 13
17 #define PIN_SPLSS 10
18 #define PIN_BUSY 9
19
20 void setup()
21 {
22
23
24   Serial.begin(9600);//シリアル通信速度設定
25   pinMode(PIN_SPLMOSI, OUTPUT);
26   pinMode(PIN_SPLMISO, INPUT);
27   pinMode(PIN_SPLSCK, OUTPUT);
28   pinMode(PIN_SPLSS, OUTPUT);
29   pinMode(PIN_BUSY, INPUT);
30   SPI.begin();
31   SPI.setDataMode(SPI_MODE3);
32   SPI.setBitOrder(MSBFIRST);
33   digitalWrite(PIN_SPLSS, HIGH);
34
35   L6470_resetdevice();//L6470 リセット
36   L6470_setup();//L6470 を設定
37
38 }
39
40
41 void loop(){
42   int a = analogRead(0);
43   int b = analogRead(1);
44   int ENCval=analogRead(2);
45   int rangeS=310;
46   int rangeL=800;//700?
47   Serial.print(a,DEC);
48   Serial.print(" ");
49   Serial.print(b,DEC);
50   Serial.print(" ");
51   Serial.print(ENCval,DEC);
52   Serial.println(" ");

```

```

53
54   if(a>=900&&ENCval<rangeL){
55       L6470_run(0,30000); //指定方向に連続回転
56
57       } else if(300>=a&&b>=800&&rangeS<ENCval){
58       L6470_run(1,30000); //指定方向に連続回転
59
60       } else{
61       L6470_hardstop(); //回転停止、保持トルクあり
62       }
63
64 }
65
66 void L6470_setup(){
67 L6470_setparam_acc(0x50); //[R, WS] 加速度 default 0x08A (12bit) (14.55*val+14.55[step/s
68 ^2])
69 L6470_setparam_dec(0x50); //[R, WS] 減速度 default 0x08A (12bit) (14.55*val+14.55[step/s
70 ^2])
71 L6470_setparam_maxspeed(0x50); //[R, WR]最大速度 default 0x041 (10bit) (1あり 5.25*val
72 +15.25[step/s])
73 L6470_setparam_minspeed(0x01); //[R, WS]最小速度 default 0x000 (1+12bit) (0.238*val[step/s
74 ])
75 L6470_setparam_fsspd(0x3ff); //[R, WR]μ ステップからフルステップへの切替点速度 default 0
76 x027 (10bit) (15.25*val+7.63[step/s])
77 L6470_setparam_kvalhold(0x50); //[R, WR]停止時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
78 L6470_setparam_kvalrun(0x50); //[R, WR]定速回転時励磁電圧 default 0x29 (8bit) (Vs[V]*val
79 /256)
80 L6470_setparam_kvalacc(0x50); //[R, WR]加速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
81 L6470_setparam_kvaldec(0x50); //[R, WR]減速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
82
83 L6470_setparam_stepmood(0x03); //ステップモード default 0x07 (1+3+1+3bit)
84 }
85
86 void fulash(){
87   Serial.print("0x");
88   Serial.print( L6470_getparam_abspos(),HEX);
89   Serial.print("_");
90   Serial.print("0x");
91   Serial.println( L6470_getparam_speed(),HEX);
92 }
93
94 /*ver 1.00 2013/4/24*/
95 /*ver 1.01 2013/12/14 コメント追記*/
96
97 /*L6470 コントロール コマンド
98 引数-----
99 dia 1:正転 0:逆転,
100 spd (20bit)(0.015*spd[step/s])
101 pos (22bit)
102 n_step (22bit)
103 act 1:絶対座標をマーク 0:絶対座標リセット
104 msec ミリ秒
105 val 各レジスタに書き込む値
106 -----
107 L6470_run(dia,spd); //指定方向に連続回転
108 L6470_stepclock(dia); //指定方向にstep ピンのクロックで回転
109 L6470_move(dia,n_step); //指定方向に指定数ステップする
110 L6470_goto(pos); //指定座標に最短でいける回転方向で移動
111 L6470_gotodia(dia,pos); //回転方向を指定して指定座標に移動
112 L6470_gountil(act,dia,spd); //指定した回転方向に指定した速度で回転し、スイッチの
113 ON で急停止と座標処理

```

Body expansion interface using electromyogram -ASHURA-

```

107 L6470_relesesw(act, dia); //スイッチがOFFに戻るまで最低速度で回転し、停止と座標処理
108 L6470_gohome(); //座標原点に移動
109 L6470_gomark(); //マーク座標に移動
110 L6470_resetpos(); //絶対座標リセット
111 L6470_resetdevice(); //L6470 リセット
112 L6470_softstop(); //回転停止、保持トルクあり
113 L6470_hardstop(); //回転急停止、保持トルクあり
114 L6470_softhiz(); //回転停止、保持トルクなし
115 L6470_hardhiz(); //回転急停止、保持トルクなし
116 L6470_getstatus(); //status レジスタの値を返す (L6470_getparam_status();と同じ)
117
118 L6470_busydelay(msec); //busy フラグが HIGH になってから、指定ミリ秒待つ。
119
120 レジスタ書き込みコマンド
121 L6470_setparam_abspos(val); //[R, WS]現在座標default 0x000000 (22bit)
122 L6470_setparam_elpos(val); //[R, WS]コイル励磁の電氣的位置default 0x000 (2+7bit)
123 L6470_setparam_mark(val); //[R, WR]マーク座標default 0x000000 (22bit)
124 //ありませんL6470_spped //[R] 現在速度read only (20bit)
125 L6470_setparam_acc(val); //[R, WS] 加速度default 0x08A (12bit) (14.55*val+14.55[step/s
    ^2])
126 L6470_setparam_dec(val); //[R, WS] 減速度default 0x08A (12bit) (14.55*val+14.55[step/s
    ^2])
127 L6470_setparam_maxspeed(val); //[R, WR]最大速度default 0x041 (10bit) (15.25*val+15.25[
    step/s])
128 L6470_setparam_minspeed(val); //[R, WS]最小速度default 0x000 (1+12bit) (0.238*val+[step/
    s])
129 L6470_setparam_fsspd(val); //[R, WR]μ ステップからフルステップへの切替点速度default 0x027
    (10bit) (15.25*val+7.63[step/s])
130 L6470_setparam_kvalhold(val); //[R, WR]停止時励磁電圧default 0x29 (8bit) (Vs[V]*val/256)
131 L6470_setparam_kvalrun(val); //[R, WR]定速回転時励磁電圧default 0x29 (8bit) (Vs[V]*val
    /256)
132 L6470_setparam_kvalacc(val); //[R, WR]加速時励磁電圧default 0x29 (8bit) (Vs[V]*val/256)
133 L6470_setparam_kvaldec(val); //[R, WR]減速時励磁電圧default 0x29 (8bit) (Vs[V]*val/256)
134 L6470_setparam_intspd(val); //[R, WH]逆起電力補償切替点速度default 0x0408 (14bit) (0.238*
    val[step/s])
135 L6470_setparam_stslp(val); //[R, WH]逆起電力補償低速時勾配default 0x19 (8bit) (0.000015*
    val[% s/step])
136 L6470_setparam_fnslpacc(val); //[R, WH]逆起電力補償高速時加速勾配default 0x29 (8bit)
    (0.000015*val[% s/step])
137 L6470_setparam_fnslpdec(val); //[R, WH]逆起電力補償高速時減速勾配default 0x29 (8bit)
    (0.000015*val[% s/step])
138 L6470_setparam_ktherm(val); //[R, WR]不明default 0x0 (4bit) (0.03125*val+1)
139 //ありませんL6470_adcout //[R] read only (5bit) ADC による逆起電力補償の大きさかな?
140 L6470_setparam_ocdth(val); //[R, WR]過電流しきい値default 0x8 (4bit) (375*val+375[mA])
141 L6470_setparam_stallth(val); //[R, WR]失速電流しきい値? default 0x40 (7bit) (31.25*val
    +31.25[mA])
142 L6470_setparam_stepmood(val); //[R, WH]ステップモードdefault 0x07 (1+3+1+3bit)
143 L6470_setparam_alareen(val); //[R, WS]有効アラームdefault 0xff (1+1+1+1+1+1+1+1bit)
144 L6470_setparam_config(val); //[R, WH]各種設定default 0x2e88 (3+3+2+1+1+1+1+1+3bit)
145 //L6470_status //[R]状態read only (16bit)
146
147 [R]:読み取り専用
148 [WR]:いつでも書き換え可
149 [WH]:書き込みは出力がハイインピーダンスの時のみ可
150 [WS]:書き換えはモータが停止している時のみ可
151
152
153 レジスタ読み込みコマンド (返り値 long 型)
154 L6470_getparam_abspos();
155 L6470_getparam_elpos();
156 L6470_getparam_mark();

```

```

157  L6470_getparam_speed();
158  L6470_getparam_acc();
159  L6470_getparam_dec();
160  L6470_getparam_maxspeed();
161  L6470_getparam_minspeed();
162  L6470_getparam_fsspd();
163  L6470_getparam_kvalhold();
164  L6470_getparam_kvalrun();
165  L6470_getparam_kvalacc();
166  L6470_getparam_kvaldec();
167  L6470_getparam_intspd();
168  L6470_getparam_stslp();
169  L6470_getparam_fnslpacc();
170  L6470_getparam_fnslpdec();
171  L6470_getparam_ktherm();
172  L6470_getparam_adcout();
173  L6470_getparam_ocdth();
174  L6470_getparam_stallth();
175  L6470_getparam_stepmood();
176  L6470_getparam_alareen();
177  L6470_getparam_config();
178  L6470_getparam_status();
179  */
180
181
182
183  void L6470_setparam_abspos(long val){L6470_transfer(0x01,3,val);}
184  void L6470_setparam_elpos(long val){L6470_transfer(0x02,2,val);}
185  void L6470_setparam_mark(long val){L6470_transfer(0x03,3,val);}
186  void L6470_setparam_acc(long val){L6470_transfer(0x05,2,val);}
187  void L6470_setparam_dec(long val){L6470_transfer(0x06,2,val);}
188  void L6470_setparam_maxspeed(long val){L6470_transfer(0x07,2,val);}
189  void L6470_setparam_minspeed(long val){L6470_transfer(0x08,2,val);}
190  void L6470_setparam_fsspd(long val){L6470_transfer(0x15,2,val);}
191  void L6470_setparam_kvalhold(long val){L6470_transfer(0x09,1,val);}
192  void L6470_setparam_kvalrun(long val){L6470_transfer(0x0a,1,val);}
193  void L6470_setparam_kvalacc(long val){L6470_transfer(0x0b,1,val);}
194  void L6470_setparam_kvaldec(long val){L6470_transfer(0x0c,1,val);}
195  void L6470_setparam_intspd(long val){L6470_transfer(0x0d,2,val);}
196  void L6470_setparam_stslp(long val){L6470_transfer(0x0e,1,val);}
197  void L6470_setparam_fnslpacc(long val){L6470_transfer(0x0f,1,val);}
198  void L6470_setparam_fnslpdec(long val){L6470_transfer(0x10,1,val);}
199  void L6470_setparam_ktherm(long val){L6470_transfer(0x11,1,val);}
200  void L6470_setparam_ocdth(long val){L6470_transfer(0x13,1,val);}
201  void L6470_setparam_stallth(long val){L6470_transfer(0x14,1,val);}
202  void L6470_setparam_stepmood(long val){L6470_transfer(0x16,1,val);}
203  void L6470_setparam_alareen(long val){L6470_transfer(0x17,1,val);}
204  void L6470_setparam_config(long val){L6470_transfer(0x18,2,val);}
205
206  long L6470_getparam_abspos(){return L6470_getparam(0x01,3);}
207  long L6470_getparam_elpos(){return L6470_getparam(0x02,2);}
208  long L6470_getparam_mark(){return L6470_getparam(0x03,3);}
209  long L6470_getparam_speed(){return L6470_getparam(0x04,3);}
210  long L6470_getparam_acc(){return L6470_getparam(0x05,2);}
211  long L6470_getparam_dec(){return L6470_getparam(0x06,2);}
212  long L6470_getparam_maxspeed(){return L6470_getparam(0x07,2);}
213  long L6470_getparam_minspeed(){return L6470_getparam(0x08,2);}
214  long L6470_getparam_fsspd(){return L6470_getparam(0x15,2);}
215  long L6470_getparam_kvalhold(){return L6470_getparam(0x09,1);}
216  long L6470_getparam_kvalrun(){return L6470_getparam(0x0a,1);}
217  long L6470_getparam_kvalacc(){return L6470_getparam(0x0b,1);}

```

## Body expansion interface using electromyogram -ASHURA-

```
218 long L6470_getparam_kvaldec(){return L6470_getparam(0x0c,1);}
219 long L6470_getparam_intspd(){return L6470_getparam(0x0d,2);}
220 long L6470_getparam_stslp(){return L6470_getparam(0x0e,1);}
221 long L6470_getparam_fnslpacc(){return L6470_getparam(0x0f,1);}
222 long L6470_getparam_fnslpdec(){return L6470_getparam(0x10,1);}
223 long L6470_getparam_ktherm(){return L6470_getparam(0x11,1);}
224 long L6470_getparam_adcout(){return L6470_getparam(0x12,1);}
225 long L6470_getparam_ocdth(){return L6470_getparam(0x13,1);}
226 long L6470_getparam_stallth(){return L6470_getparam(0x14,1);}
227 long L6470_getparam_stepmood(){return L6470_getparam(0x16,1);}
228 long L6470_getparam_alareen(){return L6470_getparam(0x17,1);}
229 long L6470_getparam_config(){return L6470_getparam(0x18,2);}
230 long L6470_getparam_status(){return L6470_getparam(0x19,2);}
231
232
233 void L6470_run(int dia,long spd){
234     if(dia==1)
235         L6470_transfer(0x51,3,spd);
236     else
237         L6470_transfer(0x50,3,spd);
238 }
239 void L6470_stepclock(int dia){
240     if(dia==1)
241         L6470_transfer(0x59,0,0);
242     else
243         L6470_transfer(0x58,0,0);
244 }
245 void L6470_move(int dia,long n_step){
246     if(dia==1)
247         L6470_transfer(0x41,3,n_step);
248     else
249         L6470_transfer(0x40,3,n_step);
250 }
251 void L6470_goto(long pos){
252     L6470_transfer(0x60,3,pos);
253 }
254 void L6470_gotodia(int dia,int pos){
255     if(dia==1)
256         L6470_transfer(0x69,3,pos);
257     else
258         L6470_transfer(0x68,3,pos);
259 }
260 void L6470_gountil(int act,int dia,long spd){
261     if(act==1)
262         if(dia==1)
263             L6470_transfer(0x8b,3,spd);
264         else
265             L6470_transfer(0x8a,3,spd);
266     else
267         if(dia==1)
268             L6470_transfer(0x83,3,spd);
269         else
270             L6470_transfer(0x82,3,spd);
271 }
272 void L6470_relesesw(int act,int dia){
273     if(act==1)
274         if(dia==1)
275             L6470_transfer(0x9b,0,0);
276         else
277             L6470_transfer(0x9a,0,0);
278     else
```

Body expansion interface using electromyogram -ASHURA-

```

279     if(dia==1)
280         L6470_transfer(0x93,0,0);
281     else
282         L6470_transfer(0x92,0,0);
283 }
284 void L6470_gohome(){
285     L6470_transfer(0x70,0,0);
286 }
287 void L6470_gomark(){
288     L6470_transfer(0x78,0,0);
289 }
290 void L6470_resetpos(){
291     L6470_transfer(0xd8,0,0);
292 }
293 void L6470_resetdevice(){
294     L6470_send_u(0x00);//nop 命令
295     L6470_send_u(0x00);
296     L6470_send_u(0x00);
297     L6470_send_u(0x00);
298     L6470_send_u(0xc0);
299 }
300 void L6470_softstop(){
301     L6470_transfer(0xb0,0,0);
302 }
303 void L6470_hardstop(){
304     L6470_transfer(0xb8,0,0);
305 }
306 void L6470_softhiz(){
307     L6470_transfer(0xa0,0,0);
308 }
309 void L6470_hardhiz(){
310     L6470_transfer(0xa8,0,0);
311 }
312 long L6470_getstatus(){
313     long val=0;
314     L6470_send_u(0xd0);
315     for(int i=0;i<=1;i++){
316         val = val << 8;
317         digitalWrite(PIN_SPL_SS, LOW); // ~SS イネーブル。
318         val = val | SPI.transfer(0x00); // アドレスもしくはデータ送信。
319         digitalWrite(PIN_SPL_SS, HIGH); // ~SS ディスエーブル
320     }
321     return val;
322 }
323
324 void L6470_transfer(int add,int bytes,long val){
325     int data[3];
326     L6470_send(add);
327     for(int i=0;i<=bytes-1;i++){
328         data[i] = val & 0xff;
329         val = val >> 8;
330     }
331     if(bytes==3){
332         L6470_send(data[2]);
333     }
334     if(bytes>=2){
335         L6470_send(data[1]);
336     }
337     if(bytes>=1){
338         L6470_send(data[0]);
339     }

```

Body expansion interface using electromyogram -ASHURA-

```

340 }
341 void L6470_send(unsigned char add_or_val){
342     while(!digitalRead(PIN_BUSY)){
343     } //BESYが解除されるまで待機
344     digitalWrite(PIN_SPLSS, LOW); // ~SS イネーブル。
345     SPI.transfer(add_or_val); // アドレスもしくはデータ送信。
346     digitalWrite(PIN_SPLSS, HIGH); // ~SS ディスエーブル。
347 }
348 void L6470_send_u(unsigned char add_or_val){//busyを確認せず送信するため用
349     digitalWrite(PIN_SPLSS, LOW); // ~SS イネーブル。
350     SPI.transfer(add_or_val); // アドレスもしくはデータ送信。
351     digitalWrite(PIN_SPLSS, HIGH); // ~SS ディスエーブル。
352 }
353 void L6470_busydelay(long time){//BESYが解除されるまで待機
354     while(!digitalRead(PIN_BUSY)){
355     }
356     delay(time);
357 }
358 long L6470_getparam(int add,int bytes){
359     long val=0;
360     int send_add = add | 0x20;
361     L6470_send_u(send_add);
362     for(int i=0;i<=bytes-1;i++){
363         val = val << 8;
364         digitalWrite(PIN_SPLSS, LOW); // ~SS イネーブル。
365         val = val | SPI.transfer(0x00); // アドレスもしくはデータ送信。
366         digitalWrite(PIN_SPLSS, HIGH); // ~SS ディスエーブル
367     }
368     return val;
369 }
370
371 #include <SPI.h>
372 //ピン番号宣言
373
374
375 int ENCval;//可変抵抗に使う
376
377
378 //変数の宣言
379
380
381
382 // ピン定義。
383 #define PIN_SPLMOSI 11
384 #define PIN_SPLMISO 12
385 #define PIN_SPLSCK 13
386 #define PIN_SPLSS 10
387 #define PIN_BUSY 9
388
389 void setup()
390 {
391     Serial.begin(9600);//シリアル通信速度設定
392     pinMode(PIN_SPLMOSI, OUTPUT);
393     pinMode(PIN_SPLMISO, INPUT);
394     pinMode(PIN_SPLSCK, OUTPUT);
395     pinMode(PIN_SPLSS, OUTPUT);
396     pinMode(PIN_BUSY, INPUT);
397     SPI.begin();
398     SPI.setDataMode(SPI_MODE3);
399     SPI.setBitOrder(MSBFIRST);
400     digitalWrite(PIN_SPLSS, HIGH);

```

```

401
402   L6470_resetdevice(); //L6470 リセット
403   L6470_setup(); //L6470 を設定
404
405 }
406
407
408 void loop(){
409   int a = analogRead(0);
410   int b = analogRead(1);
411   int ENCval=analogRead(2);
412   int rangeS=300;
413   int rangeL=800;
414   Serial.print(a,DEC);
415   Serial.print(" ");
416   Serial.print(b,DEC);
417   Serial.print(" ");
418   Serial.print(ENCval,DEC);
419   Serial.println(" ");
420
421   if(a>=900&&ENCval<rangeL){
422     L6470_run(0,30000); //指定方向に連続回転
423
424   } else if(300>=a&&b>=800&&rangeS<ENCval){
425     L6470_run(1,30000); //指定方向に連続回転
426
427   } else{
428     L6470_hardstop(); //回転停止、保持トルクあり
429   }
430 }
431
432 void L6470_setup(){
433   L6470_setparam_acc(0x50); //[R, WS] 加速度 default 0x08A (12bit) (14.55*val+14.55[step/s
434     ^2])
435   L6470_setparam_dec(0x50); //[R, WS] 減速度 default 0x08A (12bit) (14.55*val+14.55[step/s
436     ^2])
437   L6470_setparam_maxspeed(0x50); //[R, WR]最大速度 default 0x041 (10bit) (1あり 5.25*val
438     +15.25[step/s])
439   L6470_setparam_minspeed(0x01); //[R, WS]最小速度 default 0x000 (1+12bit) (0.238*val[step/s
440     ])
441   L6470_setparam_fsspd(0x3ff); //[R, WR]μ ステップからフルステップへの切替点速度 default 0
442     x027 (10bit) (15.25*val+7.63[step/s])
443   L6470_setparam_kvalhold(0x50); //[R, WR]停止時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
444   L6470_setparam_kvalrun(0x50); //[R, WR]定速回転時励磁電圧 default 0x29 (8bit) (Vs[V]*val
445     /256)
446   L6470_setparam_kvalacc(0x50); //[R, WR]加速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
447   L6470_setparam_kvaldec(0x50); //[R, WR]減速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
448
449   L6470_setparam_stepmood(0x03); //ステップモード default 0x07 (1+3+1+3bit)
450 }
451
452 void fulash(){
453   Serial.print("0x");
454   Serial.print( L6470_getparam_abspos(),HEX);
455   Serial.print(" ");
456   Serial.print("0x");
457   Serial.println( L6470_getparam_speed(),HEX);
458 }
459
460 #include <SPI.h>
461 //ピン番号宣言

```

## Body expansion interface using electromyogram -ASHURA-

```
456
457
458 int ENCval;//可変抵抗に使う
459
460
461
462 //変数の宣言
463
464
465
466 //ピン定義。
467 #define PIN_SPL_MOSI 11
468 #define PIN_SPL_MISO 12
469 #define PIN_SPL_SCK 13
470 #define PIN_SPL_SS 10
471 #define PIN_BUSY 9
472
473 void setup()
474 {
475
476     Serial.begin(9600);//シリアル通信速度設定
477     pinMode(PIN_SPL_MOSI, OUTPUT);
478     pinMode(PIN_SPL_MISO, INPUT);
479     pinMode(PIN_SPL_SCK, OUTPUT);
480     pinMode(PIN_SPL_SS, OUTPUT);
481     pinMode(PIN_BUSY, INPUT);
482     SPI.begin();
483     SPI.setDataMode(SPI_MODE3);
484     SPI.setBitOrder(MSBFIRST);
485     digitalWrite(PIN_SPL_SS, HIGH);
486
487     L6470_resetdevice();//L6470 リセット
488     L6470_setup();//L6470 を設定
489
490 }
491
492
493
494 void loop(){
495     int a = analogRead(0);
496     int b = analogRead(1);
497     int ENCval=analogRead(2);
498     int rangeS=310;
499     int rangeL=800;//700?
500     Serial.print(a,DEC);
501     Serial.print(" ");
502     Serial.print(b,DEC);
503     Serial.print(" ");
504     Serial.print(ENCval,DEC);
505     Serial.println(" ");
506
507     if(a>=900&&ENCval<rangeL){
508         L6470_run(0,30000); //指定方向に連続回転
509
510     } else if(300>=a&&b>=800&&rangeS<ENCval){
511         L6470_run(1,30000); //指定方向に連続回転
512
513     } else{
514         L6470_hardstop();//回転停止、保持トルクあり
515     }
516
```

Body expansion interface using electromyogram -ASHURA-

```

517 }
518
519 void L6470_setup(){
520 L6470_setparam_acc(0x50); //[R, WS] 加速度 default 0x08A (12bit) (14.55*val+14.55[step/s
    ^2])
521 L6470_setparam_dec(0x50); //[R, WS] 減速度 default 0x08A (12bit) (14.55*val+14.55[step/s
    ^2])
522 L6470_setparam_maxspeed(0x50); //[R, WR]最大速度 default 0x041 (10bit) (1あり 5.25*val
    +15.25[step/s])
523 L6470_setparam_minspeed(0x01); //[R, WS]最小速度 default 0x000 (1+12bit) (0.238*val[step/s
    ])
524 L6470_setparam_fsspd(0x3ff); //[R, WR]μ ステップからフルステップへの切替点速度 default 0
    x027 (10bit) (15.25*val+7.63[step/s])
525 L6470_setparam_kvalhold(0x50); //[R, WR]停止時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
526 L6470_setparam_kvalrun(0x50); //[R, WR]定速回転時励磁電圧 default 0x29 (8bit) (Vs[V]*val
    /256)
527 L6470_setparam_kvalacc(0x50); //[R, WR]加速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
528 L6470_setparam_kvaldec(0x50); //[R, WR]減速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
529
530 L6470_setparam_stepmood(0x03); //ステップモード default 0x07 (1+3+1+3bit)
531 }
532
533 void fulash(){
534 Serial.print("0x");
535 Serial.print( L6470_getparam_abspos(),HEX);
536 Serial.print("_");
537 Serial.print("0x");
538 Serial.println( L6470_getparam_speed(),HEX);
539 }
540
541 #include<Servo.h>
542 Servo servo1;
543 Servo servo2;
544 void setup(){
545 servo1.attach(8);
546 servo2.attach(9);
547 Serial.begin(9600);
548
549 }
550 int wdata8=120,wdata9=140;
551 void loop(){
552
553
554 int a = analogRead(0);
555 int b = analogRead(1);
556 Serial.print(analogRead(0));
557 Serial.print("_");
558 Serial.print(analogRead(1));
559 Serial.print("_");
560 if(a>=900&&wdata8<120){
561 wdata8 =wdata8+1;
562 } else if(300>=a&&b>=800&&wdata8>0){
563 wdata8 =wdata8-1;
564 } else{
565 }
566
567 if(a>=900&&wdata9<140){
568 wdata9 =wdata9+1;
569 } else if(300>=a&&b>=800&&wdata9>0){
570 wdata9 =wdata9-1;
571 } else{

```

## Body expansion interface using electromyogram -ASHURA-

```
572 }
573
574
575 Serial.print(wdata8);
576 Serial.print(" ");
577 Serial.println(wdata9);
578 servo1.write(wdata8);
579 servo2.write(wdata9);
580 delay(20);
581 }
582
583 #include <SPI.h>
584 //ピン番号宣言
585
586
587 int ENCval;//可変抵抗に使う
588
589 //変数の宣言
590
591
592
593 // ピン定義。
594 #define PIN_SPI_MOSI 11
595 #define PIN_SPI_MISO 12
596 #define PIN_SPI_SCK 13
597 #define PIN_SPI_SS 10
598 #define PIN_BUSY 9
599
600 void setup()
601 {
602
603
604   Serial.begin(9600);//シリアル通信速度設定
605   pinMode(PIN_SPI_MOSI, OUTPUT);
606   pinMode(PIN_SPI_MISO, INPUT);
607   pinMode(PIN_SPI_SCK, OUTPUT);
608   pinMode(PIN_SPI_SS, OUTPUT);
609   pinMode(PIN_BUSY, INPUT);
610   SPI.begin();
611   SPI.setDataMode(SPI_MODE3);
612   SPI.setBitOrder(MSBFIRST);
613   digitalWrite(PIN_SPI_SS, HIGH);
614
615   L6470_resetdevice();//L6470 リセット
616   L6470_setup();//L6470 を設定
617
618 }
619
620
621 void loop(){
622   int a = analogRead(0);
623   int b = analogRead(1);
624   int ENCval=analogRead(2);
625   int rangeS=270;
626   int rangeL=580;//700?
627   Serial.print(a,DEC);
628   Serial.print(" ");
629   Serial.print(b,DEC);
630   Serial.print(" ");
631   Serial.print(ENCval,DEC);
632   Serial.println(" ");
```

```

633
634   if(a>=800&&ENCval<rangeL){
635       L6470_run(0,30000); //指定方向に連続回転
636
637       } else if(300>=a&&b>=800&&rangeS<ENCval){
638       L6470_run(1,30000); //指定方向に連続回転
639
640       } else{
641       L6470_hardstop(); //回転停止、保持トルクあり
642       }
643
644 }
645 }
646
647 void L6470_setup(){
648 L6470_setparam_acc(0x40); //[R, WS] 加速度 default 0x08A (12bit) (14.55*val+14.55[step/s
        ^2])
649 L6470_setparam_dec(0x40); //[R, WS] 減速度 default 0x08A (12bit) (14.55*val+14.55[step/s
        ^2])
650 L6470_setparam_maxspeed(0x40); //[R, WR]最大速度 default 0x041 (10bit) (1あり 5.25*val
        +15.25[step/s])
651 L6470_setparam_minspeed(0x01); //[R, WS]最小速度 default 0x000 (1+12bit) (0.238*val[step/s
        ])
652 L6470_setparam_fsspd(0x3ff); //[R, WR]μ ステップからフルステップへの切替点速度 default 0
        x027 (10bit) (15.25*val+7.63[step/s])
653 L6470_setparam_kvalhold(0x40); //[R, WR]停止時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
654 L6470_setparam_kvalrun(0x40); //[R, WR]定速回転時励磁電圧 default 0x29 (8bit) (Vs[V]*val
        /256)
655 L6470_setparam_kvalacc(0x40); //[R, WR]加速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
656 L6470_setparam_kvaldec(0x40); //[R, WR]減速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
657
658 L6470_setparam_stepmood(0x03); //ステップモード default 0x07 (1+3+1+3bit)
659 }
660
661 void fulash(){
662     Serial.print("0x");
663     Serial.print( L6470_getparam_abspos(),HEX);
664     Serial.print("_");
665     Serial.print("0x");
666     Serial.println( L6470_getparam_speed(),HEX);
667 }

```

---

## 付録 D 制御プログラム

以下に「北の国から電子工作 (仮)」で公開されているプログラムを示す。

---

```

1  #include <SPI.h>
2  #include <MsTimer2.h>
3
4  // ピン定義。
5  #define PIN_SPL_MOSI 11
6  #define PIN_SPL_MISO 12
7  #define PIN_SPL_SCK 13
8  #define PIN_SPL_SS 10
9  #define PIN_BUSY 9
10
11 void setup()
12 {
13   delay(1000);
14
15   //ステッピングモータ用のピンの準備
16   pinMode(PIN_SPL_MOSI, OUTPUT);
17   pinMode(PIN_SPL_MISO, INPUT);
18   pinMode(PIN_SPL_SCK, OUTPUT);
19   pinMode(PIN_SPL_SS, OUTPUT);
20   pinMode(PIN_BUSY, INPUT);
21
22   //SPI 通信の開始宣言
23   SPI.begin();
24   SPI.setDataMode(SPI_MODE3); //SCK の立ち上がりでデータを送受信
25   SPI.setBitOrder(MSBFIRST); //MSB から送信
26   Serial.begin(9600);
27   digitalWrite(PIN_SPL_SS, HIGH);
28
29   L6470_resetdevice(); //L6470 リセット
30   L6470_setup(); //L6470 を設定
31
32   MsTimer2::set(100, fulash); //シリアルモニター用のタイマー割り込み
33   MsTimer2::start();
34   delay(2000);
35
36 }
37
38 void loop(){
39   //テスト駆動
40   L6470_move(1,1600); //指定方向に指定数ステップする
41   L6470_busydelay(5000); //busy フラグが HIGH になってから、指定ミリ秒待つ。
42   L6470_run(0,10000); //指定方向に連続回転
43   delay(6000);
44   L6470_softstop(); //回転停止、保持トルクあり
45   L6470_busydelay(5000);
46   L6470_goto(0x6789); //指定座標に最短でいける回転方向で移動
47   L6470_busydelay(5000);
48   L6470_run(0,0x4567);
49   delay(6000);
50   L6470_hardhiz(); //回転急停止、保持トルクなし
51 }
52

```

Body expansion interface using electromyogram -ASHURA-

```

53 void L6470_setup(){
54   L6470_setparam_acc(0x40); //[R, WS] 加速度 default 0x08A (12bit) (14.55*val+14.55[step/s
      ^2])
55   L6470_setparam_dec(0x40); //[R, WS] 減速度 default 0x08A (12bit) (14.55*val+14.55[step/s
      ^2])
56   L6470_setparam_maxspeed(0x40); //[R, WR]最大速度 default 0x041 (10bit) (15.25*val+15.25[
      step/s])
57   L6470_setparam_minspeed(0x01); //[R, WS]最小速度 default 0x000 (1+12bit) (0.238*val[step/s
      ])
58   L6470_setparam_fsspd(0x3ff); //[R, WR]μ ステップからフルステップへの切替点速度 default 0
      x027 (10bit) (15.25*val+7.63[step/s])
59   L6470_setparam_kvalhold(0x20); //[R, WR]停止時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
60   L6470_setparam_kvalrun(0x20); //[R, WR]定速回転時励磁電圧 default 0x29 (8bit) (Vs[V]*val
      /256)
61   L6470_setparam_kvalacc(0x20); //[R, WR]加速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
62   L6470_setparam_kvaldec(0x20); //[R, WR]減速時励磁電圧 default 0x29 (8bit) (Vs[V]*val/256)
63
64   L6470_setparam_stepmood(0x03); //ステップモード default 0x07 (1+3+1+3bit)
65 }
66
67 void fulash(){
68   Serial.print("0x");
69   Serial.print( L6470_getparam_abspos(),HEX);
70   Serial.print("_");
71   Serial.print("0x");
72   Serial.println( L6470_getparam_speed(),HEX);
73 }
74
75 /*ver 1.00 2013/4/24*/
76 /*ver 1.01 2013/12/14 コメント追記*/
77
78 /*L6470 コントロール コマンド
79 引数-----
80 dia 1:正転 0:逆転,
81 spd (20bit)(0.015*spd[step/s])
82 pos (22bit)
83 n_step (22bit)
84 act 1:絶対座標をマーク 0:絶対座標リセット
85 msec ミリ秒
86 val 各レジスタに書き込む値
87 -----
88 L6470_run(dia,spd); //指定方向に連続回転
89 L6470_stepclock(dia); //指定方向にstep ピンのクロックで回転
90 L6470_move(dia,n_step); //指定方向に指定数ステップする
91 L6470_goto(pos); //指定座標に最短でいける回転方向で移動
92 L6470_gotodia(dia,pos); //回転方向を指定して指定座標に移動
93 L6470_gountil(act,dia,spd); //指定した回転方向に指定した速度で回転し、スイッチの
      ONで急停止と座標処理
94 L6470_relesesw(act,dia); //スイッチがOFFに戻るまで最低速度で回転し、停止と座標処理
95 L6470_gohome(); //座標原点に移動
96 L6470_gomark(); //マーク座標に移動
97 L6470_resetpos(); //絶対座標リセット
98 L6470_resetdevice(); //L6470 リセット
99 L6470_softstop(); //回転停止、保持トルクあり
100 L6470_hardstop(); //回転急停止、保持トルクあり
101 L6470_softhiz(); //回転停止、保持トルクなし
102 L6470_hardhiz(); //回転急停止、保持トルクなし
103 L6470_getstatus(); //status レジスタの値を返す (L6470_getparam_status());と同じ)
104
105 L6470_busydelay(msec); //busy フラグが HIGH になってから、指定ミリ秒待つ。
106

```

Body expansion interface using electromyogram -ASHURA-

```

107 レジスタ書き込みコマンド
108 L6470_setparam_abspos(val); //[R, WS]現在座標default 0x000000 (22bit)
109 L6470_setparam_elpos(val); //[R, WS]コイル励磁の電氣的位置default 0x000 (2+7bit)
110 L6470_setparam_mark(val); //[R, WR]マーク座標default 0x000000 (22bit)
111 //ありませんL6470_spped //[R] 現在速度read onry (20bit)
112 L6470_setparam_acc(val); //[R, WS] 加速度default 0x08A (12bit) (14.55*val+14.55[step/s
    ^2])
113 L6470_setparam_dec(val); //[R, WS] 減速度default 0x08A (12bit) (14.55*val+14.55[step/s
    ^2])
114 L6470_setparam_maxspeed(val); //[R, WR]最大速度default 0x041 (10bit) (15.25*val+15.25[
    step/s])
115 L6470_setparam_minspeed(val); //[R, WS]最小速度default 0x000 (1+12bit) (0.238*val+[step/
    s])
116 L6470_setparam_fsspd(val); //[R, WR]μ ステップからフルステップへの切替点速度default 0x027
    (10bit) (15.25*val+7.63[step/s])
117 L6470_setparam_kvalhold(val); //[R, WR]停止時励磁電圧default 0x29 (8bit) (Vs[V]*val/256)
118 L6470_setparam_kvalrun(val); //[R, WR]定速回転時励磁電圧default 0x29 (8bit) (Vs[V]*val
    /256)
119 L6470_setparam_kvalacc(val); //[R, WR]加速時励磁電圧default 0x29 (8bit) (Vs[V]*val/256)
120 L6470_setparam_kvaldec(val); //[R, WR]減速時励磁電圧default 0x29 (8bit) (Vs[V]*val/256)
121 L6470_setparam_intspd(val); //[R, WH]逆起電力補償切替点速度default 0x0408 (14bit) (0.238*
    val[step/s])
122 L6470_setparam_stslp(val); //[R, WH]逆起電力補償低速時勾配default 0x19 (8bit) (0.000015*
    val[% s/step])
123 L6470_setparam_fnslpacc(val); //[R, WH]逆起電力補償高速時加速勾配default 0x29 (8bit)
    (0.000015*val[% s/step])
124 L6470_setparam_fnslpdec(val); //[R, WH]逆起電力補償高速時減速勾配default 0x29 (8bit)
    (0.000015*val[% s/step])
125 L6470_setparam_ktherm(val); //[R, WR]不明default 0x0 (4bit) (0.03125*val+1)
126 //ありませんL6470_adcout //[R] read onry (5bit) ADCによる逆起電力補償の大きさかな?
127 L6470_setparam_ocdth(val); //[R, WR]過電流しきい値default 0x8 (4bit) (375*val+375[mA])
128 L6470_setparam_stallth(val); //[R, WR]失速電流しきい値? default 0x40 (7bit) (31.25*val
    +31.25[mA])
129 L6470_setparam_stepmood(val); //[R, WH]ステップモードdefault 0x07 (1+3+1+3bit)
130 L6470_setparam_alareen(val); //[R, WS]有効アラームdefault 0xff (1+1+1+1+1+1+1+1bit)
131 L6470_setparam_config(val); //[R, WH]各種設定default 0x2e88 (3+3+2+1+1+1+1+1+3bit)
132 //L6470_status //[R]状態read onry (16bit)
133
134 [R]:読み取り専用
135 [WR]:いつでも書き換え可
136 [WH]:書き込みは出力がハイインピーダンスの時のみ可
137 [WS]:書き換えはモータが停止している時のみ可
138
139
140 レジスタ読み込みコマンド (返り値 long 型)
141 L6470_getparam_abspos();
142 L6470_getparam_elpos();
143 L6470_getparam_mark();
144 L6470_getparam_speed();
145 L6470_getparam_acc();
146 L6470_getparam_dec();
147 L6470_getparam_maxspeed();
148 L6470_getparam_minspeed();
149 L6470_getparam_fsspd();
150 L6470_getparam_kvalhold();
151 L6470_getparam_kvalrun();
152 L6470_getparam_kvalacc();
153 L6470_getparam_kvaldec();
154 L6470_getparam_intspd();
155 L6470_getparam_stslp();
156 L6470_getparam_fnslpacc();

```

Body expansion interface using electromyogram -ASHURA-

```

157  L6470_getparam_fnslpdec();
158  L6470_getparam_ktherm();
159  L6470_getparam_adcout();
160  L6470_getparam_ocdth();
161  L6470_getparam_stallth();
162  L6470_getparam_stepmood();
163  L6470_getparam_alareen();
164  L6470_getparam_config();
165  L6470_getparam_status();
166  */
167
168
169
170  void L6470_setparam_abspos(long val){L6470_transfer(0x01,3,val);}
171  void L6470_setparam_elpos(long val){L6470_transfer(0x02,2,val);}
172  void L6470_setparam_mark(long val){L6470_transfer(0x03,3,val);}
173  void L6470_setparam_acc(long val){L6470_transfer(0x05,2,val);}
174  void L6470_setparam_dec(long val){L6470_transfer(0x06,2,val);}
175  void L6470_setparam_maxspeed(long val){L6470_transfer(0x07,2,val);}
176  void L6470_setparam_minspeed(long val){L6470_transfer(0x08,2,val);}
177  void L6470_setparam_fsspd(long val){L6470_transfer(0x15,2,val);}
178  void L6470_setparam_kvalhold(long val){L6470_transfer(0x09,1,val);}
179  void L6470_setparam_kvalrun(long val){L6470_transfer(0x0a,1,val);}
180  void L6470_setparam_kvalacc(long val){L6470_transfer(0x0b,1,val);}
181  void L6470_setparam_kvaldec(long val){L6470_transfer(0x0c,1,val);}
182  void L6470_setparam_intspd(long val){L6470_transfer(0x0d,2,val);}
183  void L6470_setparam_stslp(long val){L6470_transfer(0x0e,1,val);}
184  void L6470_setparam_fnslpacc(long val){L6470_transfer(0x0f,1,val);}
185  void L6470_setparam_fnslpdec(long val){L6470_transfer(0x10,1,val);}
186  void L6470_setparam_ktherm(long val){L6470_transfer(0x11,1,val);}
187  void L6470_setparam_ocdth(long val){L6470_transfer(0x13,1,val);}
188  void L6470_setparam_stallth(long val){L6470_transfer(0x14,1,val);}
189  void L6470_setparam_stepmood(long val){L6470_transfer(0x16,1,val);}
190  void L6470_setparam_alareen(long val){L6470_transfer(0x17,1,val);}
191  void L6470_setparam_config(long val){L6470_transfer(0x18,2,val);}
192
193  long L6470_getparam_abspos(){return L6470_getparam(0x01,3);}
194  long L6470_getparam_elpos(){return L6470_getparam(0x02,2);}
195  long L6470_getparam_mark(){return L6470_getparam(0x03,3);}
196  long L6470_getparam_speed(){return L6470_getparam(0x04,3);}
197  long L6470_getparam_acc(){return L6470_getparam(0x05,2);}
198  long L6470_getparam_dec(){return L6470_getparam(0x06,2);}
199  long L6470_getparam_maxspeed(){return L6470_getparam(0x07,2);}
200  long L6470_getparam_minspeed(){return L6470_getparam(0x08,2);}
201  long L6470_getparam_fsspd(){return L6470_getparam(0x15,2);}
202  long L6470_getparam_kvalhold(){return L6470_getparam(0x09,1);}
203  long L6470_getparam_kvalrun(){return L6470_getparam(0x0a,1);}
204  long L6470_getparam_kvalacc(){return L6470_getparam(0x0b,1);}
205  long L6470_getparam_kvaldec(){return L6470_getparam(0x0c,1);}
206  long L6470_getparam_intspd(){return L6470_getparam(0x0d,2);}
207  long L6470_getparam_stslp(){return L6470_getparam(0x0e,1);}
208  long L6470_getparam_fnslpacc(){return L6470_getparam(0x0f,1);}
209  long L6470_getparam_fnslpdec(){return L6470_getparam(0x10,1);}
210  long L6470_getparam_ktherm(){return L6470_getparam(0x11,1);}
211  long L6470_getparam_adcout(){return L6470_getparam(0x12,1);}
212  long L6470_getparam_ocdth(){return L6470_getparam(0x13,1);}
213  long L6470_getparam_stallth(){return L6470_getparam(0x14,1);}
214  long L6470_getparam_stepmood(){return L6470_getparam(0x16,1);}
215  long L6470_getparam_alareen(){return L6470_getparam(0x17,1);}
216  long L6470_getparam_config(){return L6470_getparam(0x18,2);}
217  long L6470_getparam_status(){return L6470_getparam(0x19,2);}

```

Body expansion interface using electromyogram -ASHURA-

```

218
219
220 void L6470_run(int dia,long spd){ //指定方向に連続回転
221     if(dia==1)
222         L6470_transfer(0x51,3,spd);
223     else
224         L6470_transfer(0x50,3,spd);
225 }
226
227 void L6470_stepclock(int dia){ //指定方向に step ピンのクロックで回転
228     if(dia==1)
229         L6470_transfer(0x59,0,0);
230     else
231         L6470_transfer(0x58,0,0);
232 }
233
234 void L6470_move(int dia,long n_step){ //指定方向に指定数ステップする
235     if(dia==1)
236         L6470_transfer(0x41,3,n_step);
237     else
238         L6470_transfer(0x40,3,n_step);
239 }
240
241 void L6470_goto(long pos){ //指定座標に採点で行ける回転方向で移動
242     L6470_transfer(0x60,3,pos);
243 }
244
245 void L6470_gotodia(int dia,int pos){ //回転方向を指定して指定座標に移動
246     if(dia==1)
247         L6470_transfer(0x69,3,pos);
248     else
249         L6470_transfer(0x68,3,pos);
250 }
251
252 void L6470_gountil(int act,int dia,long spd){ //指定した回転方向に指定した速度で回転しス
    イッチの ON で急停止と座標処理
253     if(act==1)
254         if(dia==1)
255             L6470_transfer(0x8b,3,spd);
256         else
257             L6470_transfer(0x8a,3,spd);
258     else
259         if(dia==1)
260             L6470_transfer(0x83,3,spd);
261         else
262             L6470_transfer(0x82,3,spd);
263 }
264
265 void L6470_relesesw(int act,int dia){ //スイッチが
    OFF に戻るまで最低速度で回転し、停止と座標処理
266     if(act==1)
267         if(dia==1)
268             L6470_transfer(0x9b,0,0);
269         else
270             L6470_transfer(0x9a,0,0);
271     else
272         if(dia==1)
273             L6470_transfer(0x93,0,0);
274         else
275             L6470_transfer(0x92,0,0);
276 }

```

```

277
278 void L6470_gohome(){ //座標原点に移動
279     L6470_transfer(0x70,0,0);
280 }
281
282 void L6470_gomark(){ //マーク座標に移動
283     L6470_transfer(0x78,0,0);
284 }
285
286 void L6470_resetpos(){ //絶対座標リセット
287     L6470_transfer(0xd8,0,0);
288 }
289
290 void L6470_resetdevice(){ //L6470 リセット
291     L6470_send_u(0x00); //nop 命令
292     L6470_send_u(0x00);
293     L6470_send_u(0x00);
294     L6470_send_u(0x00);
295     L6470_send_u(0xc0);
296 }
297
298 void L6470_softstop(){ //回転停止、保持トルクあり
299     L6470_transfer(0xb0,0,0);
300 }
301
302 void L6470_hardstop(){ //回転急停止、保持トルクあり
303     L6470_transfer(0xb8,0,0);
304 }
305
306 void L6470_softhiz(){ //回転停止、保持トルクなし
307     L6470_transfer(0xa0,0,0);
308 }
309
310 void L6470_hardhiz(){ //回転急停止、保持トルクなし
311     L6470_transfer(0xa8,0,0);
312 }
313
314 long L6470_getstatus(){ //status レジスタの値を返す
315     long val=0;
316     L6470_send_u(0xd0);
317     for(int i=0;i<=1;i++){
318         val = val << 8;
319         digitalWrite(PIN_SPLSS, LOW); // ~SS イネーブル。
320         val = val | SPI.transfer(0x00); // アドレスもしくはデータ送信。
321         digitalWrite(PIN_SPLSS, HIGH); // ~SS ディスエーブル
322     }
323     return val;
324 }
325
326 void L6470_transfer(int add,int bytes,long val){
327     int data[3];
328     L6470_send(add);
329     for(int i=0;i<=bytes-1;i++){
330         data[i] = val & 0xff;
331         val = val >> 8;
332     }
333     if(bytes==3){
334         L6470_send(data[2]);
335     }
336     if(bytes>=2){
337         L6470_send(data[1]);

```

## Body expansion interface using electromyogram -ASHURA-

```
338 }
339 if(bytes>=1){
340     L6470_send(data[0]);
341 }
342 }
343
344 void L6470_send(unsigned char add_or_val){
345     while(!digitalRead(PIN_BUSY)){
346     } //BESYが解除されるまで待機
347     digitalWrite(PIN_SPLSS, LOW); // ~SS イネーブル。
348     SPI.transfer(add_or_val); // アドレスもしくはデータ送信。
349     digitalWrite(PIN_SPLSS, HIGH); // ~SS ディスエーブル。
350 }
351
352 void L6470_send_u(unsigned char add_or_val){//busyを確認せず送信するため用
353     digitalWrite(PIN_SPLSS, LOW); // ~SS イネーブル。
354     SPI.transfer(add_or_val); // アドレスもしくはデータ送信。
355     digitalWrite(PIN_SPLSS, HIGH); // ~SS ディスエーブル。
356 }
357
358 void L6470_busydelay(long time){//BESYが解除されるまで待機
359     while(!digitalRead(PIN_BUSY)){
360     }
361     delay(time);
362 }
363
364 long L6470_getparam(int add,int bytes){
365     long val=0;
366     int send_add = add | 0x20;
367     L6470_send_u(send_add);
368     for(int i=0;i<=bytes-1;i++){
369         val = val << 8;
370         digitalWrite(PIN_SPLSS, LOW); // ~SS イネーブル。
371         val = val | SPI.transfer(0x00); // アドレスもしくはデータ送信。
372         digitalWrite(PIN_SPLSS, HIGH); // ~SS ディスエーブル
373     }
374     return val;
375 }
```

---

## 参考文献

- [1] 厚生労働省「義肢等補装具費支給制度のご案内」  
<https://www.mhlw.go.jp/shinseiboshu/denshishinesei/dl/140513-01.pdf>  
(最終アクセス日：2018/7/19)
- [2] 北の国から電子工作 [pinelify.blog.fc2.com/blog-entry-1.html](http://pinelify.blog.fc2.com/blog-entry-1.html) (最終アクセス日:2018/10/30)
- [3] InMoov <http://inmoov.fr/> (最終アクセス日：2018/7/18)